Cluster Server Agent for SAP HANA Installation and Configuration Guide

Linux

8.0.2

VERITAS™

Veritas InfoScale™ Availability Agents

Last updated: 2023-06-29

Legal Notice

Veritas Technologies LLC
2625 Augustine Drive
Santa Clara, CA 95054
http://www.veritas.com

Technical Support

Technical Support maintains support centers globally. All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policies. For information about our support offerings and how to contact Technical Support, visit our website:
https://www.veritas.com/support

You can manage your Veritas account information at the following URL:
https://my.veritas.com

If you have questions regarding an existing support agreement, please email the support agreement administration team for your region as follows:

Worldwide (except Japan)	CustomerCare@veritas.com
Japan	CustomerCare_Japan@veritas.com

Documentation

Make sure that you have the current version of the documentation. Each document displays the date of the last update on page 2. The latest documentation is available on the Veritas website:
https://sort.veritas.com/documents

Documentation feedback

Your feedback is important to us. Suggest improvements or report errors or omissions to the documentation. Include the document title, document version, chapter title, and section title of the text on which you are reporting. Send feedback to:
infoscaledocs@veritas.com

You can also see documentation information or ask a question on the Veritas community site:
http://www.veritas.com/community/

Veritas Services and Operations Readiness Tools (SORT)

Veritas Services and Operations Readiness Tools (SORT) is a website that provides information and tools to automate and simplify certain time-consuming administrative tasks. Depending on the product, SORT helps you prepare for installations and upgrades, identify risks in your datacenters, and improve operational efficiency. To see what services and tools SORT provides for your product, see the data sheet:
https://sort.veritas.com/data/support/SORT_Data_Sheet.pdf

Contents

Chapter 1

Introducing the agent for SAP HANA

This chapter includes the following topics:

About the Cluster Server agent for SAP HANA

The Cluster Server agents monitor specific resources within an enterprise application. They determine the status of resources and start or stop them according to external events.

The Cluster Server agent for SAP HANA provides high availability for SAP HANA instances where the data is replicated with SAP HANA System Replication. The agent brings an SAP HANA instance online, monitors the instance, and takes the instance offline. It also monitors the system processes and the server state, and shuts down the server in case of a failover.

The agent currently supports the following configurations where data is replicated with SAP HANA System Replication:

- Two replicated, single-node SAP HANA database instances.

- Three replicated, single-node SAP HANA database instances, where one instance is in GCO and the other two instances are in the local cluster. Multitier and multitarget configurations are also supported.
- Two SAP HANA instances in the Scale-Out deployment.

The agent also supports SAP HANA tenant databases in Scale-Up environments.

Supported software

For information on the software versions that the Cluster Server agent for SAP HANA supports, see the Veritas Services and Operations Readiness Tools (SORT) site: https://sort.veritas.com/agents.

How the agent makes SAP HANA highly available

The Cluster Server agent for SAP HANA continuously monitors the SAP HANA instance processes to verify that they function properly.

The agent provides the following levels of application monitoring:

- Primary or Basic monitoring
  This mode has Process check and Health check monitoring options. With the default Process check option, the agent verifies that the SAP HANA instance processes are present in the process table. Process check cannot detect whether processes are in the hung or stopped states.
  In case of System Replication, the agent checks the process table and status using the `landscapeHostConfiguration.py` script.
  For an SAP HANA tenant database, the agent retrieves the list of processes by using the `sapcontrol` utility, and verifies that the tenant database process is in the running state.
- Secondary or Detail monitoring
  In this mode, the agent runs a utility to verify the status of the SAP HANA instance. The agent detects application failure, if the monitoring routine reports an improper function of the SAP HANA instance processes. When this application failure occurs, the SAP HANA instance service group fails over to another node.

In addition to these levels of application monitoring, the agent for SAP HANA is Intelligent Monitoring Framework (IMF) aware and uses Asynchronous Monitoring Framework (AMF) kernel driver for IMF notification.

Thus, the agent ensures high availability for SAP HANA instances.

Features of the agent

The Cluster Server agent for SAP HANA has the following features:

- Support for validation of attributes that are based on agent functions
  The agent can validate attributes in each agent function before the actual data processing starts.
- Support for First Failure Data Capture (FFDC)
  In case of a fault, the agent generates a huge volume of the debug logs that enable troubleshooting of the fault.
- Support for Fast First Level Monitor (FFLM)
  The agent maintains PID files based on search patterns to expedite the monitoring process.
- Support for external user-supplied monitor utilities
  The agent enables the user-specified monitor utilities to be plugged in, in addition to the built-in monitoring logic. This enables administrators to completely customize the monitoring of the application.
- Support for intelligent resource monitoring and poll-based monitoring
  The agent supports the VCS Intelligent Monitoring Framework feature. IMF allows the agent to register the resources to be monitored with the IMF notification module to receive immediate notification of resource state changes without having to periodically poll the resources.
- Support for distributed HANA databases
  When the SAP HANA database content is distributed across multiple nodes, the agent can query a table that is partitioned across multiple nodes. When this `hdbsql`-based query is successful, the agent reports that the resource is online.

SAP HANA agent functions

The agent consists of resource type declarations and agent executables. The agent executables are organized into online, offline, monitor, and clean functions.

Online

The online function performs the following tasks:

- Performs a preliminary check to ensure that the SAP HANA instance is not online on the specified node in the cluster.
- Runs the `cleanipc` utility if the value of ProcMon is **sapstartsrv**.

Otherwise, kills all the relevant SAP HANA processes that remain due to an earlier unclean shutdown.

- Starts the sapstartsrv process for the Web-based SAP Management console.
- Starts the SAP HANA instance using the `sapcontrol` command.
  For SAP HANA Scale-Out deployment, when the service group for SAP HANA instance is brought online on all the nodes, the `StartSystem` function of the `sapcontrol` command is executed. Else, the `StartWait` function of the `sapcontrol` command is executed.
- Decides whether to make the current node as the secondary node by running the `hdbnsutil -sr_state` command on the current or the remote node.
  If the current node needs to be made secondary, the `hdbnsutil -sr_register` command is used to perform the registration to primary.
  In case of deployments with three nodes, the agent registers to the primary or the secondary nodes based on the value of the Topology attribute.

  Note: The user should execute the `-sr_register` command manually before bringing the database instance online in a SAP HANA Scale-Out deployment in any of these cases:

  - In a cluster with three or more nodes

  - When the IsDistributedSystem attribute is set to 1

- Ensures that the instance is fully initialized.
- Starts the tenant database in a scale-up environment by using the `ALTER SYSTEM START DATABASE` command. The agent performs this task if the tenant database that is specified in the TenantDatabaseName attribute is not already running.
  If the RestartLimit attribute is set to a non-zero value, VCS can restart the tenant database when it is unexpectedly taken offline. You can also override the RestartLimit attribute at resource level.

Warning: The agent cannot determine the status of System Replication when the primary and secondary instances are offline simultaneously. The agent brings the SAP HANA instance up with its previous status of System Replication.

Offline

The offline function performs the following tasks:

- Checks if the SAP HANA instance is already offline.

- Executes the `sapcontrol` utility to stop the relevant instance processes.
  In a Scale-Out deployment, when the service group for the SAP HANA instance is taken offline on all the nodes, the `StopSystem` function of the `sapcontrol` utility is executed. Otherwise, the `StopWait` function of the `sapcontrol` utility is executed.
- Stops the tenant database in a scale-up environment by using `ALTER SYSTEM STOP DATABASE` command. The agent performs this task if the tenant database that is specified in the TenantDatabaseName attribute is not already stopped.
- Waits for the SAP HANA instance to go offline successfully.
- Ensures that no relevant SAP HANA processes are running. If any processes remain, the operation kills the remaining processes using a SIGKILL signal.
- Executes the `cleanipc` utility for resources where the ProcMon attribute is set to **sapstartsrv**.

Monitor

The monitor function monitors the state of the SAP HANA instance on all nodes in the cluster. The function performs the following tasks:

- Depending upon the search criteria that the ProcMon attribute specifies, the monitor function scans the process table to verify that the SAP HANA instance processes are running. For more information about setting the ProcMon attribute:
  See "Monitoring an SAP HANA instance" on page 18.
  The agent also supports IMF in the first level check. IMF enables intelligent resource monitoring. You can use the MonitorFreq key of the IMF attribute to specify the frequency at which the agent invokes the monitor function.
  For an SAP HANA tenant database in a Scale-Up environment, the agent retrieves the list of processes by using the `sapcontrol` utility, and verifies that the tenant database process is in the running state.
- If the LevelTwoMonitorFreq attribute is greater than 0, the monitor function performs a thorough health check of the SAP HANA instance using the `sapcontrol` utility.
- For System Replication, the agent checks the process table and the status of instance using the `landscapeHostConfiguration.py` script. The status of `landscapeHostConfiguration.py` overrides the status of the monitor operation.
- For a SAP HANA Scale-Out deployment, if IsDistributedSystem is set to 0 and if the SAP HANA instance is online on only one node, the `landscapeHostConfiguration.py` script returns the status as ERROR and the monitor operation reports the instance as offline.

- If the LevelTwoMonitorFreq attribute is enabled and the RunHdbSqlQuery attribute is set, the agent executes a SELECT query on a HANA database table using an `hdbsql` command.
  See "Executing `hdbsql` query for database table" on page 45.
- The monitor function executes a custom monitor utility.
  See "Executing a customized monitoring program" on page 44.

Clean

The clean function performs the following tasks:

- Sends a SIGINT signal to the `sapstart` process, if the process exists. Otherwise, the function sends a SIGINT signal to all running processes that are relevant to the specified SAP HANA instance.
- Ensures that no relevant SAP HANA processes are running. If any processes are running, the operation kills all the remaining processes using the SIGKILL signal.
- Executes the `cleanipc` utility for resources with the sapstartsrv ProcMon.
- Stops the tenant database by using `ALTER SYSTEM STOP DATABASE` command if the tenant database that is specified in the TenantDatabaseName attribute is not already stopped.

imf_init

This function initializes the SAP HANA agent to interface with the AMF kernel driver, which is the IMF notification module for the agent for SAP HANA. This function runs when the agent starts up.

imf_getnotification

This function gets notifications about resource state changes. This function runs after the agent initializes with the AMF kernel module. This function continuously waits for notification and takes action on the resource upon notification.

imf_register

This function registers or unregisters resource entities with the AMF kernel module. This function runs for each resource after the resource goes into a steady online or offline state.

Typical SAP HANA configuration in a VCS cluster or GCO

A typical SAP HANA configuration in a VCS cluster or Global Cluster Option (GCO) has the following characteristics:

- VCS is installed and configured.
- The SAP HANA instance binaries are installed locally on all nodes or on shared disks, according to SAP documentation.
- The Cluster Server agent for SAP HANA is installed on all nodes.

Figure 1-1 depicts a typical SAP HANA System Replication configuration on two systems.

Figure 1-1 SAP HANA System Replication configuration

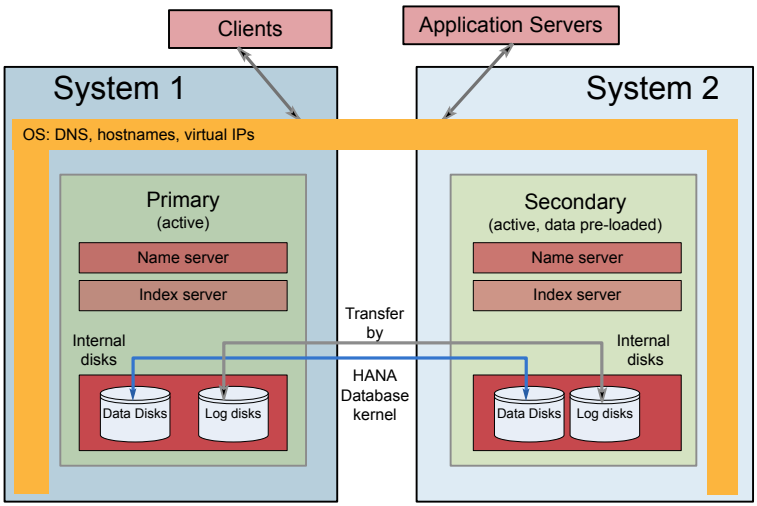
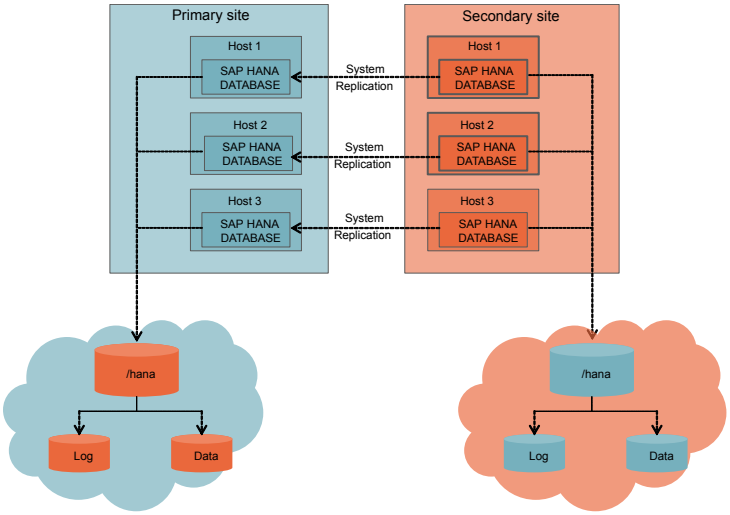


Figure 1-2 depicts a typical SAP HANA System Replication Scale-Out configuration on three systems.

Figure 1-2 SAP HANA System Replication Scale-Out configuration



Setting up SAP HANA in a VCS cluster

Perform the following tasks to set up SAP HANA in a cluster:

1. Set up a VCS cluster.
2. Install and configure SAP HANA for high availability.

   See “About configuring SAP HANA for high availability ” on page 19.
3. Install the Cluster Server agent for SAP HANA.

   See “Installing the agent ” on page 24.
4. Configure the service groups for SAP HANA.

   See “About configuring service groups for SAP HANA” on page 76.

Chapter 2

Installing and configuring SAP HANA for high availability

This chapter includes the following topics:

About SAP HANA

SAP HANA is a modern, in-memory database and a platform that can be deployed on premise or cloud. The SAP HANA platform is a flexible data source and in-memory data platform that allows users to analyze large volumes of data in real time. It also acts as a development platform that provides infrastructure and tools for building high-performance applications based on SAP HANA Extended Application Services (SAP HANA XS).

It is the foundation of various SAP HANA editions, such as the SAP HANA Platform Edition, which provides core database technology, and the SAP HANA Enterprise Edition, which bundles additional components for data provisioning. The SAP HANA Platform Edition integrates a number of SAP components, such as the SAP HANA database, SAP HANA studio, and SAP HANA clients.

About SAP HANA Scale-Out system

SAP HANA Scale-Out system is a system containing several hosts, having one instance per host. These hosts can be configured as active worker hosts or idle standby hosts. The server software is based on a flexible architecture that enables a distributed installation and balances the load across various hosts. The server software is installed in a shared file system and the file system should be mounted on all hosts that are part of the system.

A distributed system is essential when it is required to:

- Scale SAP HANA either by increasing the RAM for a single server or by adding hosts to the system to deal with larger workloads.
  This allows you to go beyond the limits of a single physical server.
- Implement high availability and failover scenarios.
  Individual hosts in a distributed system have different roles, such as master, worker, slave, and standby, depending on the tasks.

The SAP HANA database life cycle manager can be used to install an SAP HANA multiple-host system in one of the program interfaces and with a combination of parameter specification methods. For more information about installing a multiple-host system, see the related SAP documentation.

About SAP HANA System Replication

SAP HANA replicates data to a secondary SAP HANA system using its standard SAP HANA feature. Data is constantly loaded on the secondary system to minimize the recovery time objective (RTO). System replication is flexible and is used for fault tolerance and disaster recovery to achieve high availability. The data pre-load option can be used for fault recovery to enable a quicker takeover than the host auto failover.

You can build a solution with a single-node system or a scale-out system.

SAP HANA processes

The following table lists the SAP HANA processes.

Table 2-1 SAP HANA processes

Process	Functions
indexserver	Contains actual data stores and engines for processing the data.

Table 2-1 SAP HANA processes *(continued)*

Process	Functions
preprocessor	Analyzes text data using an index server and extracts information based on the text-search capabilities.
nameserver	Owns information about the topology of the SAP HANA system.
xsengine	Provides application and application developers with an access to the SAP HANA Database using a consumption model, which is exposed through HTTP.
compileserver	Compiles stored procedures and programs.
statisticsserver	Collects information related to system performance, status, and resource usage.
sapwebdisp_hdb	The native SAP HANA Web Dispatcher service.
hdbwebdispatcher	The native SAP HANA Web Dispatcher service.

Monitoring an SAP HANA instance

The monitor operation performs the process-level check to ensure proper functioning of an SAP HANA instance.

The ProcMon attribute specifies the processes that must be running successfully for a particular SAP HANA instance. The monitor operation uses this list of processes to scan the process table, and verify that the processes are running successfully. Setting the ProcMon attribute is optional.

The following are the valid values of the ProcMon attribute.

- sapstartsrv
- hdbstatisticsserver
- hdbpreprocessor
- hdbcompileserver
- hdbxsengine
- sapwebdisp_hdb
- hdbwebdispatcher

By default, the agent monitors the sapstart, hdb.sap, and hdbnameserver processes.

The sapstartsrv process should not be specified with any other *hdb* processes.

The monitor operation takes a snapshot of the running processes table. The operation compares the processes that the ProcMon attribute specifies, to the set of running UNIX processes. If any process is missing, the operation declares the SAP HANA instance as offline and bypasses further monitor operations.

Note: For SAP HANA Scale-Out system, it is recommended to set the ProcMon attribute value to sapstartsrv or keep the attribute value blank.

About configuring SAP HANA for high availability

The guidelines for configuring SAP HANA for high availability are as follows:

- In a service group, keep the single point of failure as minimal as possible and watch the application startup time.
- Assign a virtual hostname to the component within the switchover environment. For more information, refer to the *SAP HANA Administration Guide*.
- Based on the expected failover time, configure the reconnection parameters for all software components and enable its automatic reconnection.
- Configure sapcpe to copy the instance-specific executables and binaries from a central file system to the instance-executable directory, during the instance startup.

HA-DR for SAP HANA multitier and multitarget System Replication configurations

This section describes the InfoScale high availability and disaster recovery (HA-DR) solution for a SAP HANA global cluster. The solution is specifically designed for a Global Cluster Option (GCO)-enabled InfoScale cluster with three replicated, single-node SAP HANA database instances. Two of those instances are configured in the local cluster and the third instance is configured in the remote cluster.

The InfoScale HA-DR solution supports the multitier and the multitarget system replication configurations:

- Multitier system replication
- Multitarget system replication

When the node that hosts the primary SAP HANA instance fails, InfoScale performs a takeover operation, which promotes one of the secondary nodes as the new primary node. When the old primary node comes back online, it registers itself as

a secondary to one of the other nodes in the configuration based on the Topology attribute value of the SAP HANA agent.

The SAP HANA agent uses the following attribute values to determine certain actions during failback in case of multitier or multitarget configurations:

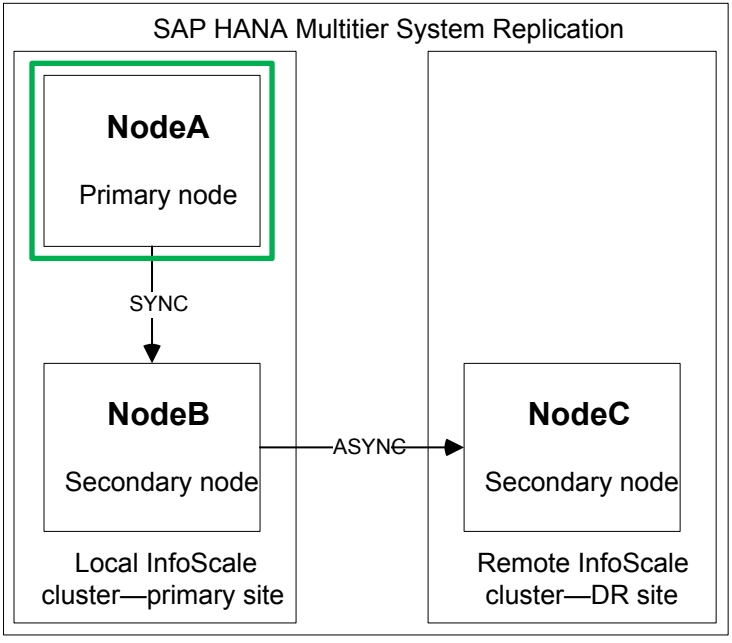
- ReplicationModeTuples—to determine the replication mode
- Topology—to decide the instance to which the new secondary node should be registered

See "SAP HANA agent attributes" on page 30.

Multitier system replication

Figure 2-1 depicts a three-node SAP HANA multitier System Replication configuration. The local InfoScale cluster at the primary site contains two nodes, NodeA and NodeB. The remote InfoScale cluster at the secondary or the DR site contains the third node, NodeC. NodeA hosts the primary SAP HANA instance. NodeB hosts the secondary SAP HANA instance at tier 2. NodeC hosts another secondary SAP HANA instance at tier 3. Data is replicated from NodeA to NodeB and from NodeB to NodeC.

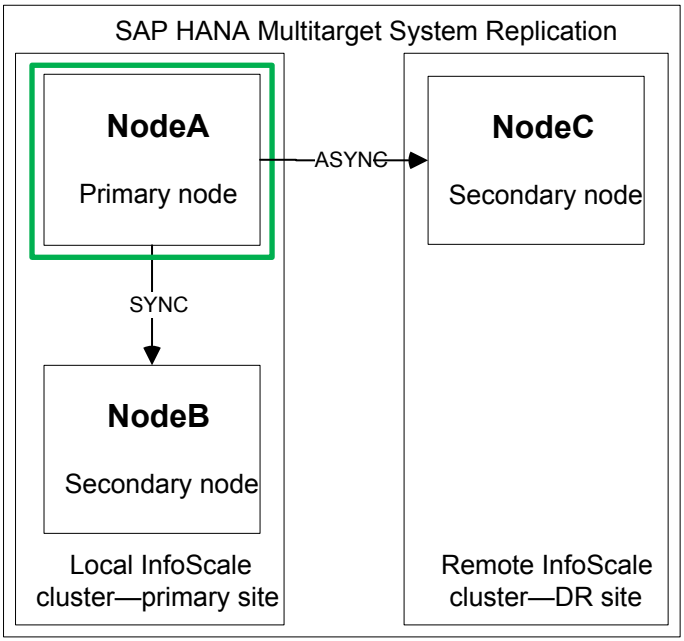
Figure 2-1 Multitier system replication



Multitarget system replication

Figure 2-2 depicts a three-node SAP HANA multitarget System Replication configuration. The local InfoScale cluster at the primary site contains two nodes, NodeA and NodeB. The remote InfoScale cluster at the DR site contains the third node, NodeC. NodeA hosts the primary SAP HANA instance. NodeB hosts the secondary SAP HANA instance at tier 2. NodeC also hosts another secondary SAP HANA instance at tier 2. Data is replicated from NodeA to NodeB and from NodeA to NodeC.

Figure 2-2 Multitarget system replication



Installing SAP HANA using virtual hostname

SAP HANA can be installed in the high-availability environment using virtual hostname. To install SAP HANA using virtual hostname, perform the following steps:

To install SAP HANA using virtual hostname

1. In the master DVD, navigate to the directory where the hdblcm tool is present.
2. Launch hdblcm using the following command:

   ```
   hdblcm --hostname=VirtualHostName
   ```

3. From the installation, select the installation type based on the usage type of system you are planning to install.

Note: SAP only supports SAP HANA installations done by a SAP HANA hardware partner, or by an SAP certified engineer qualified as "SAP Certified Technology Specialist - SAP HANA Installation" on SAP HANA certified hardware and successfully verified with the SAP HANA hardware configuration check tool.

Chapter 3

Installing, removing, and upgrading the agent for SAP HANA

This chapter includes the following topics:

Before you install the Cluster Server agent for SAP HANA

You must install the Cluster Server agent for SAP HANA on all the systems that will host SAP HANA service groups.

Ensure that you meet the following prerequisites to install the agent for SAP HANA.

- Install and configure Cluster Server.
  For more information on installing and configuring Cluster Server, refer to the Cluster Server installation and configuration guides.
- Install the latest version of the ACC library on each system in the cluster that runs the agent.

The ACC library contains common, reusable functions that perform tasks, such as process identification, logging, and system calls.
See "Installing the ACC library" on page 24.

Installing the ACC library

Install the ACC library on each system in the cluster that runs the agent.

To install the ACC library

1. Log in as a superuser.
2. Download the ACC library from the Veritas Services and Operations Readiness Tools (SORT) site (https://sort.veritas.com/agents).
3. Navigate to the `rpms` directory.
4. Install the package.

   ```
   # rpm -ivh VRTSacclib-version-GENERIC.noarch.rpm
   ```

   Enter **Yes** if asked to confirm overwriting of files in the existing package.

Installing the agent

Install the agent for SAP HANA on each node in the cluster.

To install the agent in a VCS environment

1. Download the agent from the Veritas Services and Operations Readiness Tools (SORT) site:

   https://sort.veritas.com/agents
2. Uncompress the file to a temporary location, for example: `/tmp`.
3. Log in as a superuser.
4. Install the package on Linux as follows:

   ```
   # rpm -ihv VRTSsapnw-agentVersion-GENERIC.noarch.rpm
   ```
5. Import the agent type configuration file.

   See "Importing the agent types files in a VCS environment" on page 29.

Removing the agent

You must uninstall the agent for SAP HANA from a cluster while the cluster is active.

Note: The SAPHDB agent is combined with the package for the SAPNW agent. Removing the SAPHDB agent removes the SAPNW agent.

To uninstall the agent in a VCS environment

1. Log in as a superuser.
2. Set the cluster configuration mode to read-write by running the following command from any node in the cluster:

   ```
   # haconf -makerw
   ```

3. Remove all SAP HANA (SAPHDB) and SAP NetWeaver (SAPNW) resources from the cluster. Run the following command to verify that all resources have been removed:

   ```
   # hares -list Type=SAPHDB

   # hares -list Type=SAPNW
   ```

4. Remove the agent type from the cluster configuration by running the following command from any node in the cluster:

   ```
   # hatype -delete SAPHDB

   # hatype -delete SAPNW
   ```

   Removing the agent's type file from the cluster removes the include statement for the agent from the `main.cf` file, but the agent's type file is not removed from the cluster configuration directory. You can remove the agent's type file later from the cluster configuration directory.

5. Save these changes. Then set the cluster configuration mode to read-only by running the following command from any node in the cluster:

   ```
   # haconf -dump -makero
   ```

6. Use the platform's native software management program to remove the agent for SAP HANA from each node in the cluster.

   Run the command to uninstall the agent:

   ```
   # rpm -e VRTSsapnw
   ```

Removing the ACC library

Perform the following steps to remove the ACC library.

To remove the ACC library

1. Ensure that all agents that use ACC library are removed.
2. Run the following command to remove the ACC library package:

```
# rpm -e VRTSacclib
```

Upgrading the agent in a VCS environment

Perform the following steps to upgrade the agent with minimal disruption, in a VCS environment.

To upgrade the agent in a VCS environment

1. Persistently freeze the service groups that host the application.

```
# hagrp -freeze groupName -persistent
```

2. Freeze all the groups that belong to the SAPNW or SAPHDB agent.
3. Stop the SAPHDB agent.

```
# haagent -stop SAPHDB -force -sys sysName
```

4. Stop the SAPNW agent, for some resources may belong to the SAPNW agent.

```
# haagent -stop SAPNW -force -sys sysName
```

5. Upgrade the agents.

```
Linux # rpm -Uvh VRTSsapnw-agentVersion-GENERIC.noarch.rpm
```

6. If the preonline trigger is configured, update the script as applicable to your SAP HANA deployment.

   For Scale-Up/single-node deployments: See “Configuring SAPHDB preonline script” on page 57.

   For Scale-Out deployments: See “Configuring SAPHDB preonline script” on page 82.

7. Verify that the values of the OnlineTimeout and the OfflineTimeout attributes are appropriately set.

```
# hatype -disp SAPHDB -attr OfflineTimeout OnlineTimeout
```

8. Start the SAPHDB agent.

```
# haagent -start SAPHDB -sys sysName
```

9 Start the SAPNW agent.

```
# haagent -start SAPNW -sys sysName
```

10 Unfreeze the service groups that belong to the SAPNW or SAPHDB agent.

```
# hagrp -unfreeze groupName -persistent
```

Chapter 4

Configuring the agent for SAP HANA

This chapter includes the following topics:

About configuring the Cluster Server agent for SAP HANA

After installing the Cluster Server agent for SAP HANA, you must import the agent type configuration file. After importing this file, review the attributes table that describes the resource type and its attributes, and then create and configure SAP HANA resources.

Note: When SAP HANA instances are configured in the VCS environment, ensure that you start and stop the instances only in the VCS environment.

To view the sample agent type definition and service groups configuration:

See "About sample configurations for the agents for SAP HANA" on page 91.

Importing the agent types files in a VCS environment

To use the agent for SAP HANA, you must import the agent types file into the cluster.

You can import the agent types file using the Cluster Manager (Java Console) or using the command line interface (CLI).

To import the agent types file using Cluster Manager (Java Console)

1. Start the Cluster Manager and connect to the cluster on which the agent is installed.
2. Click **File > Import Types**.
3. In the **Import Types** dialog box, select the following file:

   ```
   /etc/VRTSagents/ha/conf/SAPHDB/SAPHDBTypes.cf
   ```

4. Click **Import**.
5. Save the VCS configuration.

   The SAP agent types file is now imported to the VCS engine.

   You can now create SAP HANA resources. For additional information about using the Cluster Manager (Java Console), refer to the *Cluster Server Administrator's Guide*.

To import the agent types file using the CLI:

1. Log on to any one of the systems in the cluster as the superuser.
2. Run the following command:

   ```
   /etc/VRTSagents/ha/conf/SAPHDB/SAPHDBTypes.cmd
   ```

   The SAP HANA agent type is now imported to the VCS engine.

   You can now create SAP HANA resources.

SAP HANA agent attributes

Table 4-1 Required attributes for configuring an SAP HANA instance

Required attributes	Description
EnvFile	Absolute path to the file that must be sourced with the UNIX shell. The ksh, sh, and csh shell environments are supported. Before executing SAP scripts for the online, offline, monitor, and clean operations, you must source this file to set the environment. For RHEL7 or later and SLES12 or later, you must specify the user-generated environments file for the SAP HANA instance in this attribute. **Note:** Ensure that the syntax of this file is in accordance with the user shell that the SAPSID user specifies. See "Generating an environment file for SAP" on page 43. Type and dimension: string-scalar Default value: No default value Example: `/usr/sap/BBV/HDB00/hdbenv.sh`
InstProfile	Full path to the SAP HANA Instance profile. The *SAPSID* is found in the `/usr/sap/`*`SAPSID`*`/SYS/profile` directory. The value of the instance is `SAPSID_`*`InstName_hostname`*. The hostname must resolve into a valid IP address that is used to cluster the SAP HANA instance. Type and dimension: string-scalar Default value: No default value Example: `/usr/sap/BBV/SYS/profile/BBV_HDB00_saphana1`
EnableReverseReplication	Specifies whether the SAPHDB service group must be brought online on the replication node after the takeover service group is brought online on the primary node. The replication node is the old primary node where the SAP HANA instance was running before the takeover operation. **Note:** To enable this feature, the EnableReverseReplication and the RegistrationOfSecondary attributes must both be set to 1. The postonline trigger must also be configured for the takeover service group. Type and dimension: boolean-scalar Default value: 0 Example: 1 **Note:** Do not set this attribute to 1 for a multitier or a multitarget System Replication configuration.

Table 4-1 Required attributes for configuring an SAP HANA instance *(continued)*

Required attributes	Description
RunHdbSqlQuery	Specifies whether the SAPHDB agent must execute the hdbsql query during level-two monitoring. If enabled, the agent executes a SELECT query on vcsschema.MY_TABLE by using an hdbsql command. Type and dimension: boolean-scalar Default value: 1 Example: 0
TakeoverWithHandshake	Specifies whether the SAPHDB agent must perform a Takeover with Handshake by adding the --suspendPrimary option to the takeover command. The Takeover with Handshake is only performed when the replication state is active. Type and dimension: boolean-scalar Default value: 1 Example: 0

Table 4-2 Optional attributes for configuring an SAP HANA instance

Optional attribute	Description
ResLogLevel	Detail level of logs that the agent must record for the resource. The valid values are: ■ ERROR: Only logs error messages. ■ WARN: Logs above plus warning messages. ■ INFO: Logs above plus informational messages. ■ TRACE: Logs above plus trace messages. TRACE is very verbose and should only be used during initial configuration or for troubleshooting and diagnostic operations. Type and dimension: string-scalar Default value: INFO Example: TRACE **Note:** The use of the ResLogLevel attribute is deprecated from VCS version 6.2 onwards. You must use the LogDbg attribute instead of the ResLogLevel attribute to enable debug logs for the ACCLib-based agents, when the ACCLib version is 6.2.0.0 or later. The agent captures the first failure data of the unexpected events and automatically logs debug messages in their respective agent log files.

Table 4-2 Optional attributes for configuring an SAP HANA instance *(continued)*

Optional attribute	Description
LogDbg	For ACCLib-based agents, you must use the LogDbg resource type attribute to enable the debug logs when the ACCLib version is 6.2.0.0 or later and the VCS version is 6.2 or later. Set the LogDbg attribute to DBG_5 to enable debug logs for the ACCLib-based agent. By default, setting the LogDbg attribute to DBG_5 enables debug logs for all SAP HANA resources in the cluster. If debug logs must be enabled for a specific SAP HANA resource, override the LogDbg attribute. For more information on how to enable debug logs, See "To enable debug logs for all resources of type SAPHDB" on page 89. Type and dimension: keylist Default value: No default value For more information on how to use the LogDbg attribute, refer to the *Cluster Server Administrator's Guide*.
MonitorProgram	Absolute path name of an external, user-supplied monitor executable. See "Executing a customized monitoring program" on page 44. Type and dimension: string-scalar Default value: No default value Example 1: `/usr/sap/PI1/HDB00/work/myMonitor.sh` Example 2: `/usr/sap/PI1/HDB00/work/myMonitor.sh arg1 arg2`

Table 4-2 Optional attributes for configuring an SAP HANA instance *(continued)*

Optional attribute	Description
ProcMon	List of SAP processes to monitor. The entries in this list are separated using space and can be specified in any order. Your configuration must have at least two resources for each SAP HANA instance. One of the resources must contain the `sapstartsrv` process as a ProcMon attribute value. In the other resource, this attribute may be left blank or it may contain other processes like `hdb`, but it must *not* contain `sapstartsrv`. Before you specify a value, review the information about how the monitor operation uses this attribute. See "Monitoring an SAP HANA instance" on page 18. Type and dimension: string-scalar Default value: No default value Example: hdbxsengine **Note:** For SAP HANA Scale-Out system, Veritas recommends that you either set the ProcMon attribute value to `sapstartsrv` or keep it blank.
LevelTwoMonitorFreq	Frequency at which the agent for this resource type must perform the second-level or detailed monitoring. You can also override the value of this attribute at the resource level. The value indicates the number of monitor cycles after which the agent will monitor the SAP HANA instance in detail. For example, the value 5 indicates that the agent will monitor the SAP HANA instance in detail after every five online monitor intervals. The sapcontrol binary is used during the second-level monitoring. Type and dimension: integer-scalar Default value: 0 Example: 1 `# /opt/VRTSvcs/bin/hatype -modify SAPHDB LevelTwoMonitorFreq 1`

Table 4-2 Optional attributes for configuring an SAP HANA instance *(continued)*

Optional attribute	Description
IsDistributedSystem	Specifies whether the Scale-Out mode for the SAP HANA database is distributed: ■ **0**: SAP HANA Scale-Out is not distributed. ■ **1**: SAP HANA Scale-Out is distributed. If a single system goes down when this attribute is set to 1, the entire SAP HANA database is failed over to the other site. **Note:** Veritas recommends that when this attribute is set to 1, you set the SAPHDB type-level attribute LevelTwoMonitorFreq to 1, so that the agent verifies the HANA database status using the `hdbsql` query. Type and dimension: boolean-scalar Default value: 0 Example: 1
DBUser	SAP HANA database user name. This user must have the administrative privileges to perform the insert, delete, update, and select operations on `vcsschema.MY_TABLE` in the `SystemDB` database. ■ If hdbuserstore is configured, you can use this attribute to specify the hdbuserstore key for the database user. ■ If hdbuserstore is not configured, you must provide the appropriate value in the DBPassword attribute. You must specify this attribute in any of the following conditions: ■ When IsDistributedSystem is set to 1. ■ In a scale-up environment with tenant databases, when LevelTwoMonitorFreq is enabled. Type and dimension: string-scalar Default value: No default value Example: vcsuser

Table 4-2 Optional attributes for configuring an SAP HANA instance *(continued)*

Optional attribute	Description
DBPassword	Encrypted password value for the SAP HANA database user. Use `vcsencrypt` to encrypt the password value. ■ If hdbuserstore is configured, this attribute must be empty. ■ If hdbuserstore is not configured, this attribute must be specified when IsDistributedSystem is set to 1. Type and dimension: string-scalar Default value: No default value Example: (encrypted)
TenantDatabaseName	Name of the SAP HANA tenant databases to be monitored. A single SAPHDB agent resource can be used to monitor multiple tenant databases; the database names should be separated by spaces. When this attribute is specified, you must specify the DBUser attribute. The DBUser attribute value should be the name of a user who has DATABASE ADMIN privileges to start and stop the tenant database. If hdbuserstore is configured, you can use the DBUser attribute to the hdbuserstore key for the database user. **Note:** Also, when hdbuserstore is not configured, you must specify the appropriate value in the DBPassword attribute. Type and dimension: string-scalar Default value: No default value Example: **BW9 ERP TDB3**
IMF	Determines if the agent must perform the intelligent resource monitoring. You can also override the value of this attribute at the resource level. See "About the keys of the IMF attribute" on page 41.
IMFRegList	Specifies the ordered list of attributes whose values are registered with the IMF notification module. The attribute values can be overridden at the resource level.

Table 4-2 Optional attributes for configuring an SAP HANA instance *(continued)*

Optional attribute	Description
SystemReplication	Specifies whether the System Replication is enabled for the SAP HANA database. The valid values are: ■ 1 - SAP HANA System Replication is enabled. ■ 0 - SAP HANA System Replication is disabled. Type and dimension: boolean-scalar Default value: 0 Example: 1
SystemReplicationMode	System Replication mode for the SAP HANA database. The valid values are `async`, `sync`, and `syncmem`. You need to provide a value for this attribute only in case of a two-node configuration. For a multitier or a multitarget configuration, add the appropriate value in the ReplicationModeTuples attribute. Type and dimension: string-scalar Default value: No default value Example: sync
TakeOverInInactiveState	Specifies whether the System Replication takeover should be forcibly performed for the SAP HANA Database when the Replication link state is inactive. Refer to the *SAP Note 2063657: HANA System Replication takeover decision guideline*. Valid values are: ■ 1: SAP HANA System Replication takeover is forcibly enabled. ■ 0: SAP HANA System Replication takeover is not forcibly enabled. Type and dimension: boolean-scalar Default value: 0 Example: 1

Table 4-2 Optional attributes for configuring an SAP HANA instance *(continued)*

Optional attribute	Description
DifferenceInTimestamp	Specifies whether to perform the takeover operation for the SAP HANA Database only when the time difference in seconds between the shippedLogPosTimestamp and shippedSavepointTimestamp attribute is less than the value of the DifferenceInTimestamp attribute. The following are the minimum and maximum values that can be specified for the DifferenceInTimestamp attribute: ■ Minimum value: 0 ■ Maximum value: 65535 Refer to the *SAP Note 2063657: HANA System Replication takeover decision guideline*. Type and dimension: integer-scalar Default value: 65535 Example: 900
HANAScaleOutSupport	Specifies whether the ScaleOut mode is enabled for the SAP HANA Database. Valid values are: ■ 1 - SAP HANA ScaleOut is enabled. ■ 0 - SAP HANA ScaleOut is not enabled. Default value: 0 Example: 1
RegistrationOfSecondary	Specifies whether the automatic re-registration of a secondary SAP HANA instance to its primary is enabled. Valid values are: ■ 1 - Registration of a secondary SAP HANA instance to its primary is enabled. ■ 0 - Registration of a secondary SAP HANA instance to its primary is not enabled. Default value: 1 Example: 1

Table 4-2 Optional attributes for configuring an SAP HANA instance *(continued)*

Optional attribute	Description
UseSystemD	SystemD is a system and a service manager for Linux operating systems. It helps manage applications across Linux distributions that support the SystemD feature. When the UseSystemD attribute is set to 1, on SLES 12 or RHEL 7 distributions, the agent starts the SAP HANA processes in system.slice. When this attribute is set to 0, a typical online function starts the SAP processes in user.slice. Type and dimension: boolean-scalar Default value: 0 Example: 1
CustomQuery	When LevelTwoMonitorFreq is enabled, the agent runs the hdbsql query for the tenant database with the DBUser credentials that are provided. ■ In an SAP HANA scale-out configuration, the agent executes the `SELECT * FROM VCSVIEW` query. ■ In an SAP HANA scale-out configuration with tenant databases, you can replace the default query with a custom query. To do so, you must specify the custom query as part of this attribute value. **Note:** When you specify this attribute, you must also specify the DBUser attribute. The database user that is specified in that attribute must have the necessary privileges to run the custom query. Type and dimension: string-scalar Example: `SELECT * FROM MY_TABLE` Default value: No default value
GetReplicationStatusTimeout	During the Online operation, the agent runs the GetReplicationStatusInfo action entry point. The GetReplicationStatusInfo action identifies the replication status by checking the availability of the primary HANA instance. In a scale-up environment, if the primary HANA instance is not available, this task may cause the Online operation to time out. If the GetReplicationStatusTimeout attribute is set to a non-zero value, the agent uses this value as a timeout value within which to execute the GetReplicationStatusInfo action. Default value: 0 Example: 30

Table 4-2 Optional attributes for configuring an SAP HANA instance *(continued)*

Optional attribute	Description
ReplicationModeTuples	Specifies the SAP HANA System Replication modes to be used between the current node and each of its secondary nodes. Its value is in the form of tuples that contain two elements. The first element indicates a secondary or a target node, and the second element indicates the replication mode between the current node and that target node. The valid replication modes are **sync**, **syncmem**, and **async**. This attribute is used when two or more nodes can act as the secondary node. Default value: No default value Example: Consider that NodeA is the current node and the possible target nodes are: NodeB with synchronous replication and NodeC with asynchronous replication. The value of this attribute should be set to **node_B=sync, node_C=async**.
Topology	Helps the InfoScale agent to decide which SAP HANA instance to register to during a failback operation. The valid values for this attribute and the corresponding agent behaviors are as follows: ■ If set to **multitier_strict**, the agent follows the multitier topology and ensures that none of the instances have more than one target at a time. ■ If set to **multitier_preferred**, the agent follows the multitier topology over the multitarget topology. The agent takes this action when a secondary instance has both the options available and the value of following either of the topologies is the same. ■ If set to **multitarget_strict**, the agent follows the multitarget topology and ensures that the configuration does not have more than two tiers at a time. ■ If set to **multitarget_preferred**, the agent follows the multitarget topology over the multitier topology. The agent takes this action when a secondary instance has both the options available and the value of following either of the topologies is the same. This attribute is used when two nodes can act as the secondary node. You need to ensure that this attribute is set to the same value across all the nodes. Default value: No default value Example: **multitier_preferred**

Table 4-2 Optional attributes for configuring an SAP HANA instance *(continued)*

Optional attribute	Description
WaitToShutdownOldPrimary	After the sr_takeover operation for a SAP HANA database, the agent stops the old primary instance and then proceeds to bring the TakeOverIP resource online. In the case of an SAP HANA scale-out configuration, you can specify the number of seconds to wait for the old primary instance to be shut down. After this duration, the agent proceeds to bring the TakeOverIP resource online. Type and dimension: integer-scalar Default value: 60 Example: 30
WaitForPrimarySGOnline	Specifies the duration in seconds that the preonline trigger on the secondary must wait before the online operation for the corresponding service group on the primary is initiated. This attribute is used only if the online operation of the secondary service group is initiated before that of the primary. To use this attribute effectively, perform both these activities: ■ Configure the HA/DR provider hook. See “Integrating HA/DR provider hooks with the SAPHDB agent” on page 72. ■ Set the OnlineTimeout attribute at the resource level. Type and dimension: integer-scalar Default value: 120 Example: 60

Table 4-2 Optional attributes for configuring an SAP HANA instance *(continued)*

Optional attribute	Description
CheckSyncStateBeforeRoleChange	Specifies whether to perform a takeover by bringing the SAP HANA IP resource online. The agent takes the following actions based on the value of this attribute: ■ If set to 1 and if the HANA databases are in sync, the agent performs the takeover. ■ If set to 1 and if the HANA databases are not in sync, the agent does not perform the takeover. ■ If set to 0, the agent forcefully performs the takeover, regardless of the synchronization status of the HANA databases. If the SAP HANA IP resource is brought online on the Secondary and if this attribute is set to 1, the IP resource is taken offline. Configure the HA/DR provider hook to use this attribute effectively. See "Integrating HA/DR provider hooks with the SAPHDB agent" on page 72. Type and dimension: boolean-scalar Default value: 0 Example: 1

About the keys of the IMF attribute

The IMF type-level attribute uses the following keys:

Table 4-3 IMF attribute keys

Key	Description
Mode	Define this attribute to enable or disable intelligent resource monitoring. Valid values are as follows: ■ 0—Does not perform intelligent resource monitoring ■ 1—Performs intelligent resource monitoring for offline resources and performs poll-based monitoring for online resources ■ 2—Performs intelligent resource monitoring for online resources and performs poll-based monitoring for offline resources ■ 3—Performs intelligent resource monitoring for both online and for offline resources. **Note:** The agent for SAP HANA supports intelligent resource monitoring for online resources only. Hence, Mode should be set to either 0 or 2. Default: 2
MonitorFreq	This key value specifies the frequency at which the agent invokes the monitor agent function. The value of this key is an integer. You can set this key to a non-zero value for cases where the agent requires to perform both poll-based and intelligent resource monitoring. If the value is 0, the agent does not perform poll-based process check monitoring. After the resource registers with the AMF kernel driver, the agent calls the monitor agent function as follows: ■ After every (MonitorFreq x MonitorInterval) number of seconds for online resources ■ After every (MonitorFreq x OfflineMonitorInterval) number of seconds for offline resources Default: 5
RegisterRetryLimit	If you enable intelligent resource monitoring, the agent invokes the imf_register agent function to register the resource with the AMF kernel driver. The value of the RegisterRetryLimit key determines the number of times the agent must retry registration for a resource. If the agent cannot register the resource within the limit that is specified, then intelligent monitoring is disabled until the resource state changes or the value of the Mode key changes. Default: 3

Generating an environment file for SAP

Veritas recommends that you use a custom-generated environment file to configure the EnvFile attribute of the agent.

To generate the environment file for SAP applications on non-SystemD platforms

1. Log in as an SAP administrator.

   For example:

   ```
   su - pi1adm
   ```

2. Capture the environment information in a file with the `env` command.

   For example:

   ```
   env > /home/pi1adm/sappi1env.env
   ```

3. Update this file according to the user shell environment of the SAP administrator.

   For example, if the generated file contains the environment for the `bash` shell and the SAP administrator uses the C shell, update the file as follows:

   - Edit the `sappi1env.env` file to add the "setenv" string at the beginning of each line.
   - Replace each instance of the "=" character with a space.

4. Ensure that the file permissions are set appropriately for the SAP administrator.

   ```
   chmod a+x sappi1env.env
   ```

5. Copy the `sappi1env.env` file to a shared directory and set the EnvFile attribute value to this file path.

   This file is then used as the environment file for the SAP instance.

To generate the environment file for SAP applications on SystemD-enabled platforms

1. Log in as an SAP administrator.

   For example:

   ```
   su - pi1adm
   ```

2. Capture the environment information in a file with the `env` command.

   For example:

   ```
   env > /home/pi1adm/sap_sysd.env
   ```

3. Ensure that the file permissions are set appropriately for the SAP administrator.

   ```
   chmod a+x sappi1env.env
   ```

4. Copy the `sap_sysd.env` file to a shared directory and set the SystemDEnvFile attribute value to this file path.

   This file is then used as the environment file for the SAP instance.

Executing a customized monitoring program

The monitor function can execute a customized monitoring utility to perform an additional SAP HANA instance state check.

The monitor function executes the utility specified in the MonitorProgram attribute if the following conditions are satisfied:

- The specified utility is a valid executable file.
- The first-level process check indicates that the SAP HANA instance is online.
- The LevelTwoMonitorFreq attribute is either set to 0 or 1, and the second-level check indicates that the SAP HANA instance is online.
- The LevelTwoMonitorFreq attribute is set to greater than 1, but the second-level check is deferred for this monitoring cycle.

The monitor function interprets the utility exit code as follows:

110 or 0	SAP HANA instance is online
100 or 1	SAP HANA instance is offline
99	SAP HANA instance is unknown

Any other value	SAP HANA instance is unknown

To ensure that the customized utility is always available to the agent, Veritas recommends storing the file in a shared directory that is available on an online node.

Note: If the IsDistributedSystem attribute is set to 1, the user-specified external monitor program is executed regardless of the status received from `landscapeHostConfiguration.py`.

Executing `hdbsql` query for database table

When the TenantDatabaseName attribute is not specified, if you enable the LevelTwoMonitorFreq attribute and set the RunHdbSqlQuery attribute, ensure that you create a schema named `vcsschema` that contains a table named `MY_TABLE`.

- In case of a HANA scale-out deployment, MY_TABLE can be partitioned across multiple nodes.
- In case of a HANA scale-up deployment, MY_TABLE can be created only on a single node.

The agent executes a `SELECT` query on `vcsschema.MY_TABLE` by using an `hdbsql` command.

If MY_TABLE is partitioned in a HANA scale-out deployment, the agent can query the partitioned table only if the IsDistributedSystem attribute is enabled.

For the `hdbsql` query to run, you must perform the following tasks:

1. Create a schema with the name `vcsschema`.

   ```
   CREATE SCHEMA vcsschema;
   ```

2. Create a table with data partitioned across multiple servers when IsDistributedSystem is enabled. You can customize the table creation query according to the number of servers. For example:

   ```
   CREATE COLUMN TABLE vcsschema.MY_TABLE (a INT, b INT, c INT,
   PRIMARY KEY (a,b)) PARTITION BY HASH (a, b) PARTITIONS
   GET_NUM_SERVERS()
   ```

   Alternatively, create a table on a single node. For example:

   ```
   CREATE COLUMN TABLE vcsschema.MY_TABLE (a INT, b INT, c INT,
   PRIMARY KEY (a,b))
   ```

3. Insert test data into the table. For example:

```
INSERT INTO vcsschema.MY_TABLE VALUES (1,1,1);

INSERT INTO vcsschema.MY_TABLE VALUES (2,1,1);
```

4. Verify the partitions, if any. For example:

   ```
   SELECT * FROM M_TABLE_LOCATIONS where table_name='MY_TABLE' and
   SCHEMA_NAME='VCSSCHEMA'
   ```

   Sample output:

   ```
   HOST,PORT,SCHEMA_NAME,TABLE_NAME,PART_ID,LOCATION
   "saphanavm1",30003,"VCSSCHEMA","MY_TABLE",1,"saphanavm1:30003"
   "saphanavm2",30003,"VCSSCHEMA","MY_TABLE",2,"saphanavm2:30003"
   "saphanavm3",30003,"VCSSCHEMA","MY_TABLE",3,"saphanavm3:30003"
   "saphanavm4",30003,"VCSSCHEMA","MY_TABLE",4,"saphanavm4:30003"
   ```

5. Verify that the SELECT query works correctly.

   ```
   SELECT * FROM vcsschema.MY_TABLE
   ```

   Sample output:

   ```
   A,B,C
   2,1,1
   1,1,1
   ```

6. The user that is configured to perform the VCS query must have administrative privileges on `vcsschema`.

   In scale-up environments with tenant databases, the user must also have the necessary privileges to start and stop the tenant databases.

   Sample command to create a user:

   ```
   CREATE USER vcsuser PASSWORD Axp14ph46
   ```

   Sample commands to create a role and grant the appropriate persmissions:

   ```
   CREATE ROLE vcs_role

   GRANT SELECT ON SCHEMA vcsschema TO vcs_role

   GRANT INSERT ON SCHEMA vcsschema TO vcs_role

   GRANT UPDATE ON SCHEMA vcsschema TO vcs_role

   GRANT DELETE ON SCHEMA vcsschema TO vcs_role

   GRANT vcs_role TO vcsuser WITH ADMIN OPTION;
   ```

Enabling the agent to support IMF

By default, the SAP HANA agent is enabled to support IMF. In order to enable the SAP HANA agent to support IMF, you must make the following configuration changes to the attributes of the agent:

- AgentFile: Set the AgentFile attribute to **Script60Agent**
- IMF Mode: Set the IMF Mode attribute to **2**
- IMFRegList: Update the IMFRegList attribute

The following sections provide more information on the commands you can use to make these configuration changes, depending on whether VCS is in a running state or not.

If VCS is in a running state

To enable the SAP HANA resource for IMF when VCS is in a running state:

1. Make the VCS configuration writable.

```
# haconf -makerw
```

2. Run the following command to update the AgentFile attribute.

```
# hatype -modify SAPHDB AgentFile\
/opt/VRTSvcs/bin/Script60Agent
```

3. Run the following commands to add the IMF attributes:

```
# haattr -add -static  SAPHDB IMF -integer -assoc Mode 0 \
MonitorFreq 1 RegisterRetryLimit 3

# haattr -add -static SAPHDB IMFRegList -string -vector
```

4 Run the following command to update the IMF attribute.

```
# hatype -modify SAPHDB IMF Mode num MonitorFreq num
RegisterRetryLimit num
```

For example, to enable intelligent monitoring of online resources, with the MonitorFreq key set to 5, and the RegisterRetryLimit key is set to 3, run the following command:

```
# hatype -modify SAPHDB IMF Mode 2 MonitorFreq 5 \
RegisterRetryLimit 3
```

Note: The valid values for the Mode key of the IMF attribute are 0 (disabled) and 2 (online monitoring).

5 Run the following command to update the IMFRegList attribute:

```
# hatype -modify SAPHDB IMFRegList InstProfile
```

6 Save the VCS configuration.

```
# haconf -dump -makero
```

7 If the SAP HANA agent is running, restart the agent.

For information on the commands you can use to restart the agent, see Restarting the agent.

Restarting the agent

To restart the agent:

1 Run the following command to stop the agent forcefully:

```
# haagent -stop <resourceType> -force -sys <systemName>
```

Note: Stopping the agent forcefully eliminates the need to take the resource offline.

2 Run the following command to start the agent:

```
# haagent -start <resourceType> -sys <systemName>
```

If VCS is not in a running state

To change the SAPHDB type definition file when VCS is not in a running state:

1. Update the AgentFile attribute.

   ```
   static str AgentFile = "/opt/VRTSvcs/bin/Script60Agent"
   ```

2. Update the IMF attribute.

   The valid values for the Mode key of the IMF attribute are 0 (disabled) and 2 (online monitoring).

   ```
   static int IMF{} = { Mode=num, MonitorFreq=num,
   RegisterRetryLimit=num }
   ```

   For example, to update the IMF attribute such that the Mode key is set to 2, the MonitorFreq key is set to 5, and the RegisterRetryLimit key is set to 3:

   ```
   static int IMF{} = { Mode=2, MonitorFreq=5, RegisterRetryLimit=3
   }
   ```

3. Update the IMFRegList attribute.

   ```
   static str IMFRegList[] = { InstProfile }
   ```

Disabling intelligent resource monitoring

To disable intelligent resource monitoring

1. Make the VCS configuration writable.

   ```
   # haconf -makerw
   ```

2. To disable intelligent resource monitoring for all the resources of a certain type, run the following command:

   ```
   # hatype -modify SAPHDB IMF -update Mode 0
   ```

3. To disable intelligent resource monitoring for a specific resource, run the following command:

   ```
   # hares -override resource_name IMF
   ```

   ```
   # hares -modify resource_name IMF -update Mode 0
   ```

4. Save the VCS configuration.

   ```
   # haconf -dump -makero
   ```

Chapter 5

Configuring service groups for Scale-Up/single-node SAP HANA deployments

This chapter includes the following topics:

About configuring service groups for SAP HANA

Configuring the SAP HANA service group involves creating the SAP service group, its resources, and defining attribute values for the configured resources. You must have administrator privileges to create and configure a service group.

You can configure the service groups using one of the following:

- The Cluster Manager (Java console)
- Veritas Infoscale Operations Manager
- The command line

Before configuring the service groups for SAP HANA

Before you configure the SAP HANA service group, you must:

- Verify that the Cluster Server components are installed and configured on all nodes in the cluster where you will configure the service group.
  For more information on installing the components, refer to the *InfoScale Availability Installation Guide*.
- Verify that SAP HANA is installed and configured identically on all nodes in the cluster.
- Verify that the Cluster Server agent for SAP HANA is installed on all nodes in the cluster.
  See "Installing the agent " on page 24.

Configuring service groups for SAP HANA

The following sections describe how to configure service groups within the same cluster and in the GCO cluster.

Configuring service groups for SAP HANA within the same cluster

While setting up a cluster, you must always ensure that the cluster has some spare capacity to handle the SAP HANA failover scenarios. For example, in case of a back end database failure, the cluster must be able to run another database instance in conjunction with other running applications.

The cluster should be able to provide application failover by encapsulating the resources required for an application into a service group. A service group is a virtualized application that can switch between the cluster nodes. It contains a set of dependent resources, such as disk groups, disk volumes, file systems, IP addresses, NIC cards, and dependent application processes. It also includes logic about the dependencies between the application components.

These service groups should thus be configured such that the cluster can start, stop, monitor, and switch the service groups between the nodes, depending upon the server faults or resource faults. An administrator should also be proactively able

to move a service group between cluster nodes to perform preventative maintenance or apply patches.

Note:

Perform all the steps in the following procedure to ensure that the service group configuration is completed successfully.

To add a service group for SAP HANA when GCO is not enabled

1. Create a service group for SAP HANA.

   For example:

   ```
   # hagrp -add BBV_HDB00_REP_SG
   ```

   ```
   # hagrp -modify BBV_HDB00_REP_SG Parallel 1
   ```

   For more details on creating a service group, refer to the *Cluster Server Administrator's Guide*.

2. Modify SystemList attribute for the group, to add systems.

   For example:

   ```
   # hagrp -modify BBV_HDB00_REP_SG SystemList sysA 0 sysB 1
   ```

3. Create resources for sapstartsrv process in the service group.

   For example:

   ```
   # hares -add BBV_HDB00_REP_sapstartsrv SAPHDB BBV_HDB00_REP_SG
   ```

4. Create SAPHDB resources for SAP HANA.

   For example:

   ```
   # hares -add BBV_HDB00_REP_RES SAPHDB BBV_HDB00_REP_SG
   ```

   Based on the SAP HANA instance you are clustering, modify the resource attributes. For more information on agent attributes,

   See "SAP HANA agent attributes" on page 30.

5. Create resource dependencies for the SAP HANA resource.

   The SAPHDB resource depends on the SAPHDB resource with the sapstartsrv process.

   For example:

   ```
   # hares -link BBV_HDB00_REP_RES BBV_HDB00_REP_sapstartsrv
   ```

6 Verify the final resource dependencies for the SAP HANA service group.

```
# hares -dep
```

For example:

```
Group               Parent               Child
APHDB-BBVHDB00     BBV_HDB00_REP_RES    BBV_HDB00_REP_sapstartsrv
```

7 Create a service group for the SAP HANA IP resource.

For example:

```
# hagrp -add SAP_IP_REP
```

8 Modify the SystemList attribute for the group to add systems.

For example:

```
# hagrp -modify SAP_IP_REP SystemList sysA 0 sysB 1
```

9 Modify the PreOnline attribute of the SAP HANA IP group.

For example:

```
# hagrp -modify SAP_IP_REP PreOnline 1
```

10 Create IP resources in the service group.

For example:

```
# hares -add SAP_IP_RES IP SAP_IP_REP
```

Note: You must set the RestartLimit attribute for the configured IP resource. You can override the attribute values as follows:

```
# /opt/VRTSvcs/bin/hares -override SAP_IP_RES RestartLimit

# /opt/VRTSvcs/bin/hares -modify SAP_IP_RES RestartLimit 2
```

---

11 Create group dependencies for SAP HANA and SAP HANA IP.

For example:

```
# hagrp -link SAP_IP_REP BBV_HDB00_REP_SG online local hard
```

12 Remove all the entries from the AutoStartList attribute of the service group.

13 Bring the service group online on the cluster node where the HANA instance is the primary.

14 Bring the service group online on the other cluster node where the HANA instance is the secondary.

Configuring service groups for SAP HANA within the GCO cluster

Ensure that you meet the following prerequisites to configure service group for SAP HANA under the GCO cluster.

- Configure GCO using the `gcoconfig` command or using CPI.
- Maintain same service group and resource names on all sites of the cluster.
- Passwordless authentication should be configured between all nodes in the clusters.

Perform the following steps on site 1 to add a service group for SAP HANA

1. Create a service group for SAP HANA.

   For example:

   ```
   # hagrp -add SAPHANASR_TEST_SG
   ```

   For more details on creating a service group, refer to the *Cluster Server Administrator's Guide*.

2. Modify the SystemList attribute for the group, to add systems.

   For two replicated, single-node SAP HANA database instances,

   ```
   # hagrp -modify SAPHANASR_TEST_SG SystemList sys1 0
   ```

   For three replicated, single-node SAP HANA database instances,

   ```
   # hagrp -modify SAPHANASR_TEST_SG SystemList sys1 0 sys3 1
   ```

   Similarly, run the following command on site 2 to modify the SystemList attribute for the group.

   ```
   # hagrp -modify SAPHANASR_TEST_SG SystemList sys2 0
   ```

3. Create resources for the sapstartsrv process in the service group.

   For example:

   ```
   # hares -add RES_SAPHANA_SR_SAPSTARTSRV SAPHDB SAPHANASR_TEST_SG
   ```

4. Create SAPHDB resources for SAP HANA.

   For example:

   ```
   # hares -add BBV_HDB00_REP_RES SAPHDB SAPHANASR_TEST_SG
   ```

   Based on the SAP HANA instance you are clustering, modify the resource attributes. For more information on agent attributes,

   See "SAP HANA agent attributes" on page 30.

5 Create resource dependencies for the SAP HANA resource.

The SAPHDB resource depends on the SAPHDB resource with the sapstartsrv process.

For example:

```
# hares -link RES_SAPHANA_SR RES_SAPHANA_SR_SAPSTARTSRV
```

6 Verify the final resource dependencies for the SAP HANA service group.

```
# hares -dep
```

For example:

```
Group                Parent              Child
ClusterService       gcoip               gconic
ClusterService       wac                 gcoip
SAPHANASR_TEST_SG    RES_SAPHANA_SR      RES_SAPHANA_SR_SAPSTARTSRV
```

7 Create a service group for the SAP HANA IP resource.

For example:

```
# hagrp -add SAPHANA_SR_NETWORK
```

8 Modify the SystemList attribute for the group to add systems.

For two replicated, single-node SAP HANA database instances,

```
# hagrp -modify SAPHANA_SR_NETWORK SystemList sys1 0
```

For three replicated, single-node SAP HANA database instances,

```
# hagrp -modify SAPHANA_SR_NETWORK SystemList sys1 0 sys3 1
```

Similarly, run the following command on site 2 to modify the SystemList attribute.

```
# hagrp -modify SAPHANA_SR_NETWORK SystemList sys2 0
```

9 Modify the PreOnline attribute of the SAP HANA IP group.

For example:

```
# hagrp -modify SAPHANA_SR_NETWORK PreOnline 1
```

10 Create IP resources in the service group.

For example:

```
# hares -add RES_SAPHANA_MUM_IP IP SAPHANA_SR_NETWORK
```

Note: You must set the RestartLimit attribute for the configured IP resource. You can override the attribute values as follows:

```
/opt/VRTSvcs/bin/hares -override RES_SAPHANA_MUM_IP RestartLimit
```

```
/opt/VRTSvcs/bin/hares -modify RES_SAPHANA_MUM_IP RestartLimit 2
```

---

11 Create group dependencies for SAP HANA and SAP HANA IP.

For example:

```
# hagrp -link SAPHANA_SR_NETWORK SAPHANASR_TEST_SG online local
hard
```

```
[root@saphanasr2 config]#hagrp -dep
#Parent              Child              Relationship
SAPHANA_SR_NETWORK   SAPHANASR_TEST_SG  online local hard
```

12 Configure the service group with IP resource configured as global.

For example:

```
# hagrp -modify SAPHANA_SR_NETWORK ClusterList saphana_site1 0
saphana_site2 1
```

13 Set the ClusterFailOverPolicy attribute as Manual. In the Auto mode, the agent takes the decision of takeover, whereas manual intervention is required for the actions in the Manual mode.

```
# hagrp -modify SAPHANA_SR_NETWORK ClusterFailOverPolicy Manual
```

Note: If the ClusterFailOverPolicy attribute is set as Auto, the PreOnline trigger can run multiple times on the same system resulting in the execution of the `-sr_takeover` command on the system several times. This may result in an unexpected behavior.

---

14 Perform all the steps in this procedure on site 2.

Note: The value of the OnlineTimeOut attribute of the SAPHDB agent type should be large enough to accommodate the replication time taken either for starting the SAP HANA instance or performing the takeover operation. If the replication delays the starting of the SAP HANA instance or the takeover operation, and the time exceeds the value specified by the OnlineTimeOut attribute, then the SAP HANA instance should be started outside VCS or the takeover operation should be performed.

For GCO, the local group name and resource names for the SAPHDB resources should be the same across the sites. Additionally, the service group for the IP resource can only be the parent service group for the SAPHDB resource.

Configuring SAPHDB preonline script

In a clustered environment, the SAP administrator installs and configures the SAP HANA System Replication. The SAP HANA System Replication has the following requisites:

- If a primary replication site fails, then the secondary replication site must take over the role as primary replication.
- If the primary replication site is switched over manually to a secondary replication site, then the secondary replication site must take over the role of primary replication and the primary replication site is brought down.

The SAPHDB preonline script facilitates proper SAP HANA System Replication takeover behavior. The existing VCS preonline script calls the SAPHDB preonline script.

Note: The preonline script must be configured for a service group for the IP resource.

The SAPHDB preonline script performs the following tasks:

- If the service group for which the script is running does not have child service group for the SAPHDB resources, the script returns the control back to the VCS preonline script.
- If the service group is for the IP resource and has SAPHDB resources in its child service group, the script determines whether the target node is the secondary replication site and performs the takeover action. The script also ensures that the online operation does not execute the VCS preonline script again.

- If the replication status on the secondary replication site is not active and if the following conditions are met, then the takeover operation is rejected on the secondary replication site. An attempt is made to restart the primary replication instance.
  - The preload_column_tables value is set to false, and:
    - The TakeOverInInactiveState attribute of the SAPHDB resource is set to false.
    - The value of the DifferenceInTimestamp attribute of the SAPHDB resource is less than the time difference in seconds between the shippedLogPosTimestamp and shippedSavepointTimestamp values.
  - The OperationMode is logreplay and the preload_column_tables value is set to true, and:
    - The TakeOverInInactiveState attribute of the SAPHDB resource is set to false.
    - The old primary database is shut down.

To accomplish this failover behavior, you must configure the VCS preonline script.

To configure the VCS preonline script

1 Create a symlink for the preonline script to the monitor script.

```
cd /opt/VRTSagents/ha/bin/SAPHDB ln -s \
/opt/VRTSagents/ha/bin/SAPHDB/monitor preonline
```

Note: You need to create this link only if the package installer has failed to create it.

2 Navigate to the `$VCS_HOME/bin/triggers` directory.

3 If the VCS preonline trigger script is already present, add the following lines to the main preonline trigger script to integrate the call to the SAPHDB preonline trigger:

```
# SAPHDB specific preonline code: START
# If preonline trigger needs to be run for more than one agent,
# then copy this snippet into the main preonline trigger.

#-------------------
# Define variables..
#-------------------
my $sCmd = '/opt/VRTSagents/ha/bin/SAPHDB/preonline';
my $sResLogLevel = 'TRACE'; # Define logging level..
my @lsCmdArgs = ( @ARGV, $sResLogLevel ); # Insert logging level..
my $sArgs = join ( ' ', @lsCmdArgs );
my $iSAPHDBExitCode = undef;

#-----------------------------------------
# Pass control to preonline, if it exists..
#-----------------------------------------
if ( -x $sCmd ) {
  VCSAG_LOG_MSG ("I", "Preonline Cmd [$sCmd] Args [$sArgs]", 15031);
  system ( $sCmd, @lsCmdArgs );
  $iSAPHDBExitCode = $? >> 8; # Capture exit code..
  VCSAG_LOG_MSG ("I", "Preonline Cmd [$sCmd] Exited with \
[$iSAPHDBExitCode]", 15031);
  exit 0 unless ( $iSAPHDBExitCode == 1 );
}

#
# SAPHDB specific preonline code: STOP
#
if (defined $ARGV[3]) {
   system("$vcs_home/bin/hagrp -online -nopre $ARGV[1] \
-sys $ARGV[0] -checkpartial $ARGV[3]");
   exit;
}
system("$vcs_home/bin/hagrp -online -nopre $ARGV[1] -sys $ARGV[0]");
exit;
```

4 If the VCS preonline trigger script is not present, do the following:

- Pick the sample preonline script from the `/etc/VRTSagents/ha/conf/SAPHDB` directory and copy it in the `$VCS_HOME/bin/triggers` directory.
- Ensure that the file is executable and accessible to the root user.

5 For the service group, set the preonline flag to True.

```
hagrp -modify service_group PreOnline 1 -sys system
```

The preonline script is now configured to facilitate the SAP HANA replication behavior. To configure the logging level used in the preonline script, you can set the ResLogLevel attribute in the preonline wrapper. You can then view the logs in the VCS engine log, `/var/VRTSvcs/log/engine_A.log`.

Note: Once the preonline trigger is configured, you may see unexpected behavior while manually switching or performing online operations on the SAP HANA instance group. This behavior is a result of the control logic within the preonline trigger that performs the replication takeover, if needed. For system maintenance, if you prefer to perform manual operations on the service groups, you can do so by disabling the preonline trigger.

To disable the preonline trigger, use the following command:

```
# hagrp -modify service_group PreOnline 0 -sys system
```

Configuring SAPHDB postonline script

When the primary and the standby instances of an SAP HANA setup are both part of a single VCS cluster, the SAPHDB postonline script facilitates setting up reverse replication by registering the old primary as the standby. The existing VCS postonline script calls the SAPHDB postonline script. To enable this feature, the EnableReverseReplication and the RegistrationOfSecondary attributes must both be set to 1.

Note: The postonline script must be configured for a service group for the IP resource. However, in case of multitier and multitarget System replication configurations, the postonline script must not be configured.

The SAPHDB postonline script performs the following tasks:

- If the service group for which the script is running does not have child service group for the SAPHDB resources, the script exits.

- If the service group is for the IP resource and has SAPHDB resources in its child service group, the script determines whether the target node is the secondary node and initiates reverse replication.
- If the SAPHDB service group state on the secondary node is ONLINE or STARTING, the agent does not take any action.
- If the SAPHDB service group state on the secondary node is OFFLINE, the agent brings the SAPHDB service group online on the secondary node.
- If the SAPHDB service group state on the secondary node is FAULTED, the agent clears the fault and then brings the SAPHDB service group online on the secondary node.
- If the SAPHDB service group state on the secondary node is STOPPING, the agent waits for the service group to go offline. It waits for the duration specified by the OfflineTimeout attribute, and then brings the SAPHDB service group online on the secondary node.

To configure the VCS postonline script

1 Create a symlink for the postonline script to the monitor script.

```
cd /opt/VRTSagents/ha/bin/SAPHDB ln -s \
/opt/VRTSagents/ha/bin/SAPHDB/monitor postonline
```

Note: You need to create this link only if the package installer has failed to create it.

2 Navigate to the `$VCS_HOME/bin/triggers` directory.

3 If the VCS postonline trigger script is already present, add the following lines to the main postonline trigger script to integrate the call to the SAPHDB postonline trigger:

```
# Add the SAPHDB Trigger Call here….
#-------------------
# Define variables..
#-------------------
my $sCmd = '/opt/VRTSagents/ha/bin/SAPHDB/postonline';
my $sResLogLevel = 'TRACE'; # Define logging level..
my @lsCmdArgs = ( @ARGV, $sResLogLevel ); # Insert logging level..
my $sArgs = join ( ' ', @lsCmdArgs );
my $iSAPHDBExitCode = undef;

#-----------------------------------------
# Pass control to postonline, if it exists..
#-----------------------------------------
if ( -x $sCmd ) {
  VCSAG_LOG_MSG ("I", "Postonline Cmd [$sCmd] Args [$sArgs]", 15033);
  system ( $sCmd, @lsCmdArgs );
  $iSAPHDBExitCode = $? >> 8; # Capture exit code..
  VCSAG_LOG_MSG ("I", "Postonline Cmd [$sCmd] Exited with
    [$iSAPHDBExitCode]", 15034);
  exit 0 unless ( $iSAPHDBExitCode );
}

#
# SAPHDB specific postonline code: STOP
#
exit;
```

4 If the VCS postonline trigger script is not present, do the following:

- Pick the sample postonline script from the `/etc/VRTSagents/ha/conf/SAPHDB` directory and copy it to the `$VCS_HOME/bin/triggers` directory.
- Ensure that the file is executable and accessible to the root user.

5 For the service group, add POSTONLINE for the TriggersEnabled attribute.

```
hagrp -modify serviceGroupName TriggersEnabled -add POSTONLINE
```

The postonline script is now configured to facilitate reverse replication.

To configure the logging level used in the postonline script, you can set the ResLogLevel attribute in the postonline wrapper. You can then view the logs in the VCS engine log, `/var/VRTSvcs/log/engine_A.log`.

To disable the postonline trigger

Run the following command on any node in the cluster:

```
# hagrp -modify serviceGroupName TriggersEnabled -delete
POSTONLINE
```

SAP HANA System Replication takeover decision guidelines

Agent performs the below operations as part of the preonline trigger while deciding to perform the takeover operation on the current node. These operations are decided as per the *SAP Note 2063657: HANA System Replication takeover decision guideline*.

If the replication status on the secondary replication site—where the preonline trigger is being executed—is not active and if the following conditions are met, the takeover operation is rejected on the node on which the preonline trigger is getting executed. An attempt is made to restart its primary replication instance.

- The preload_column_tables value is set to false, and:
  - The TakeOverInInactiveState attribute of the SAPHDB resource is set to false.
  - The value of the DifferenceInTimestamp attribute of the SAPHDB resource is less than the time difference in seconds between the shippedLogPosTimestamp and shippedSavepointTimestamp values.
- The OperationMode is logreplay and the preload_column_tables value is set to true, and:
  - The TakeOverInInactiveState attribute of the SAPHDB resource is set to false.
  - The old primary database is shut down.

The values of preload_column_tables, shippedLogPosTimestamp , and shippedSavepointTimestamp are determined using the `hdbcons` command.

Based on the SAP replication takeover decision guidelines, Veritas recommends that you set the RetryLimit attribute as follows:

- If pre-loading is active, set the value of RetryLimit to 0; if it is inactive set it to 1.

- After the takeover operation, the system returns to normal performance only after the loading of the tables is complete, depending on your application configuration.
  - In OLTP scenarios like SAP ERP or SAP CRM, several tables need to be accessed concurrently. Therefore, the system might be available, but with poor performance, until the loading of the tables is complete. In such scenarios, performing a takeover may let the system return to a normal speed of operation faster. In comparison, restarting the primary site database and waiting for the tables to be reloaded, may keep the system performance low for longer. In this case, set the RetryLimit attribute to 0.
  - In OLAP scenarios like SAP BW, the load varies more and the application may operate with acceptable performance even while the tables are being loaded. In such scenarios, performing a takeover may not reduce the duration of the downtime in comparison to restarting the primary site, if the latter is possible. In this case, set the RetryLimit attribute to 0.

Note: Takeover with Handshake is performed only for scale-up configurations with SAP HANA 2.0 SPS4 and later.

Takeover with Handshake

Takeover with Handshake ensures that all the redo log data that is sent is written to the disk on the secondary system. This operation is ideal for a safe, planned takeover, while the primary is still running. Any new writing transactions on the primary are suspended.

Takeover with Handshake avoids:

- Data loss, as the log is available on the secondary before the takeover is triggered.
- Split-brain situations, as the former primary is suspended.

The agent performs a takeover with handshake in a scale-up configuration, only if the following conditions are met:

- The SAP HANA version is 2.0 SPS 04 or later.
- The primary instance is running.
- The replication state is active.
- The TakeOverWithHandshake attribute is set to 1.

If Takeover with Handshake is not successful, the agent attempts a Takeover without Handshake.

Make a note of the following:

- When Takeover with Handshake is performed, the time taken to shut down an old, suspended primary increases considerably. Ensure that you update the OnlineTimeout and the OfflineTimeout values accordingly.
- When a Takeover without Handshake times out, a partial takeover is possible. In such a scenario, perform the following tasks sequentially to restore the previous configuration:
  - Manually take the takeover service group offline on the secondary.
  - Re-register the secondary with the old primary,
  - Bring the takeover service group online on the old primary.

Note: Takeover with Handshake is not performed in case of three-node setups.

Events and responses for System Replication setups with three replicated, single-node SAP HANA database instances

Consider a SAP HANA configuration with three replicated, single-node SAP HANA database instances. Two instances, node A and node B, are configured in the local cluster. The third instance, node C, is configured in the remote cluster.

The following configurations are supported:

- **Multitarget**: System Replication is set up with the SAP HANA instance on node A configured as the primary and the instances on node B and node C configured as its secondaries. The resultant replication chains are A → B and A → C.
- **Multitier**: System Replication is set up with the SAP HANA instance on node A configured as the primary, the instance on node B configured as the secondary to node A, and the one on node C configured as secondary to node B. The resultant replication chain is A → B → C.

Table 5-1 Common failover or takeover scenarios and the corresponding responses

Scenario	Consequent events in a multitarget configuration (A → B, A → C)	Consequent events in a multitier configuration (A → B → C)
The IP group is switched from node A to node B.	The SAP HANA instance on node B becomes the primary. The agent takes the SAP HANA service group offline on node A. Node B replicates data to node C.	The SAP HANA instance on node B becomes the primary. The agent takes the SAP HANA service group offline on node A. Node B replicates data to node C.
The SAP HANA instance on node A faults.	VCS performs the SAP HANA System Replication takeover operation on node B. The instance on node B becomes the primary. Node B replicates data to node C.	VCS performs the SAP HANA System Replication takeover operation on node B. The instance on node B becomes the primary. Node B replicates data to node C.
The SAP HANA instance on node B faults.	VCS does not take any action. Node A continues to replicate data to node C.	Node C registers to node A. Node A begins to replicate data to node C.
The SAP HANA instance on node C faults.	VCS does not take any action. Node A continues to replicate data to node B.	VCS does not take any action. Node A continues to replicate data to node B.
An unexpected primary site failure occurs.	VCS moves the IP to the node at the other site, and makes that node the primary.	VCS moves the IP to the node at the other site, and makes that node the primary.
The IP group is switched from node A to node C.	VCS performs the SAP HANA System Replication takeover operation on node C. Node C begins to replicate data to node B.	VCS performs the SAP HANA System Replication takeover operation on node C. VCS does not change the replication configuration. The administrator must take the appropriate action to reconfigure the replication from node C to node B.

Table 5-1 Common failover or takeover scenarios and the corresponding responses *(continued)*

Scenario	Consequent events in a multitarget configuration (A → B, A → C)	Consequent events in a multitier configuration (A → B → C)
The IP resource that is configured for SAP HANA goes down.	VCS attempts to restart the IP resource. **Note:** The RestartLimit attribute value for the IP resource must be greater than 1. If VCS is not able to restart the IP resource, it performs the takeover operation on the target node.	VCS attempts to restart the IP resource. **Note:** The RestartLimit attribute value for the IP resource must be greater than 1. If VCS is not able to restart the IP resource, it performs the takeover operation on the target node.

Considerations for multitier and multitarget configurations

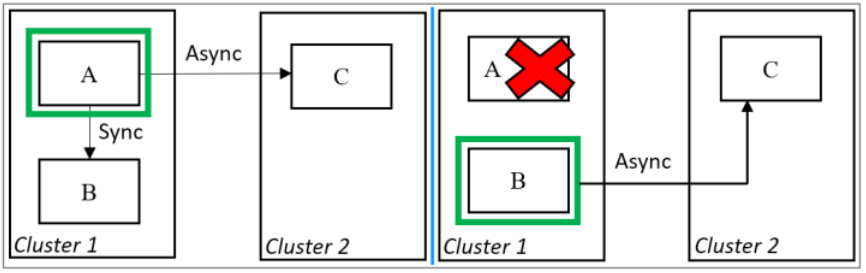
When the Topology attribute is set as follows, you must set the corresponding System Replication attribute values appropriately for the failover and failback operations to complete successfully:

Topology attribute value	Actions for the System Replication configuration
■ `multitarget_strict` ■ `multitarget_preferrerd` ■ `multitier_preferred`	Set the `global.ini/[system_replication]/register_secondaries_on_takeover` paramater to `true` so that the old secondary can automatically register itself to the new primary.

Topology attribute value	Actions for the System Replication configuration
▪ `multitier_strict` ▪ `multitier_preferred` ▪ `multitarget_preferred`	▪ Set `[system_replication]/enable_log_retention` to `force_on_takeover` on all the systems. During the takeover operation on a secondary system, if `enable_log_retention` is set to `force_on_takeover`, its value is changed to `force`. Consequently, the log is retained from the time that the takeover operation begins until it is explicitly disabled. ▪ Re-register all the required systems until the landscape is fully functional again. ▪ Reset `[system_replication]/enable_log_retention` to `force_on_takeover` on the system where the takeover operation was performed before the original configuration was reestablished. ▪ If you want to reorder your systems, enable log retention propagation. Log retention without propagation only affects the direct neighbors. To enable log retention propagation, set the following parameter in `global.ini`: `[system_replication]/propagate_log_retention = on`. If you want to propagate log retention between all systems in a system replication landscape, ensure that this parameter is set on all the systems in the landscape. **Note:** When you configure these attributes, ensure that you follow the guidelines that are mentioned in the SAP article: Log Retention and Multitarget System Replication. This article describes both the attributes, **enable_log_retention** and **propagate_log_retention**.

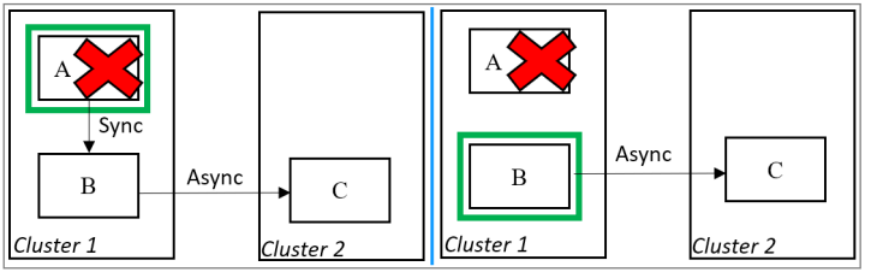
Sample use cases

- Primary node failure
  - Multitarget configuration



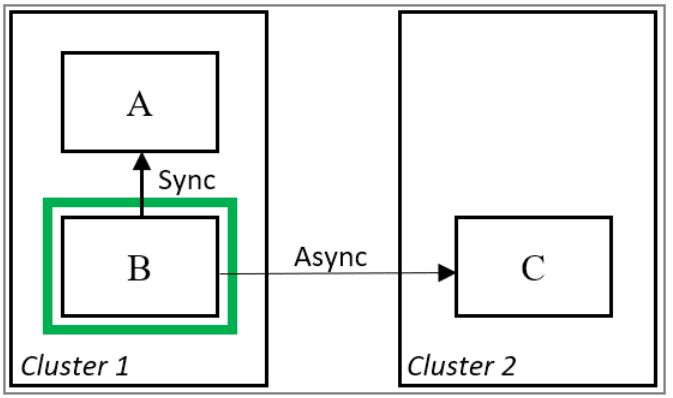
The primary, node A, faults. When InfoScale detects the failure, it performs a takeover operation and node B takes over as the new primary node. Veritas recommends that you set `global.ini/[system_replication]/register_secondaries_on_takeover` to `true`, so that node C can automatically register itself to the new primary, node B.

  - Multitier configuration



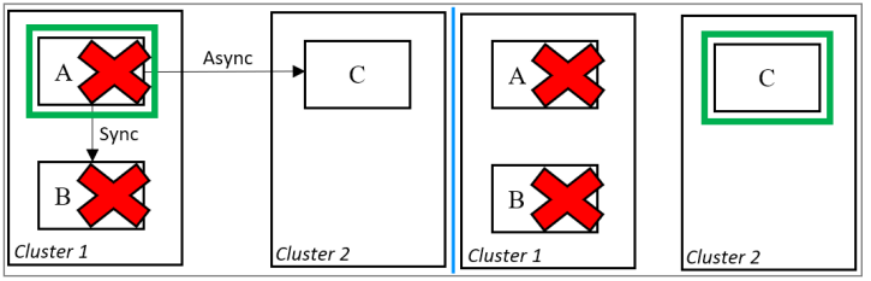
The primary, node A, faults. When InfoScale detects the failure, it performs a takeover operation and node B takes over as the new primary node. Node C continues to be a secondary to node B.

- Failback after primary node recovery—recommended failback method



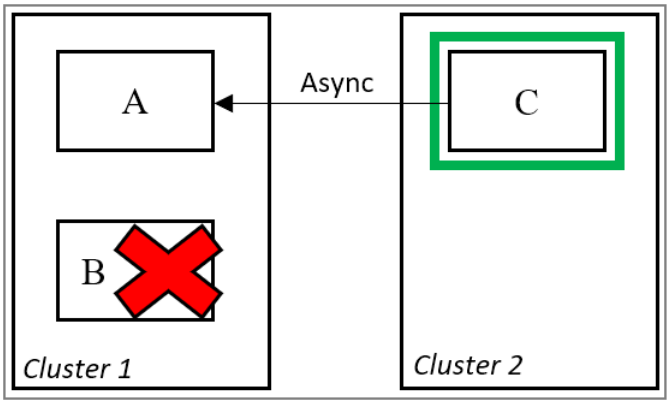
The old primary, node A, is healthy again and is able to join the cluster. When the SAP HANA database must be brought online on node A, it registers itself to node B as a secondary. In this case, the Topology attribute is set to `multitarget_preferred`.

- Primary site failure

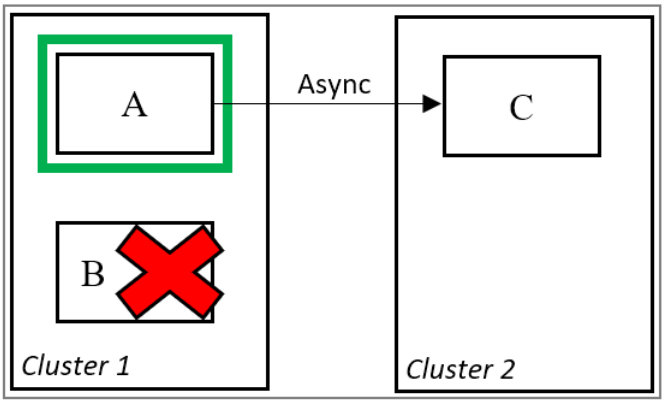


In a multitarget configuration, the primary site, Cluster 1, faults. When InfoScale detects the failure, it performs a takeover operation, and node C at the DR site (Cluster 2) takes over as the new primary node.

- Failback after primary site recovery—recommended failback method
  - Step 1: Failback of node A

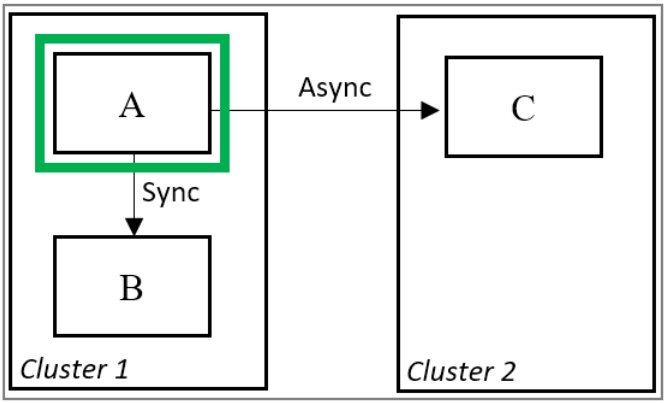


The old primary, node A, is healthy again and is able to join the cluster. The SAP HANA database service group must be brought online on node A. It then registers itself to node C as a secondary.



At this point you must restore the primary SAP HANA database at the primary site (Cluster 1). You must perform a switch operation from node C to node A, and then bring the SAP HANA database online on node C. Thereafter, the replication direction will be reversed.

  - Step 2: Failback of node B



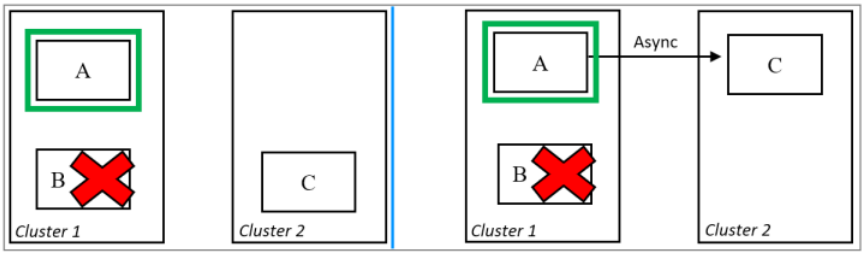
Node B is healthy again and is able to join the cluster. When the SAP HANA database is brought online on node B, it registers itself to node A as a secondary. In this case, the Topology attribute is set to `multitarget_preferred`. The initial configuration is restored.

Note: After a site failure, node C is the primary node. Node B joins the configuration and becomes the secondary to node C. At this point, if node A is brought online, it registers itself to node B. In this case, Topology is set to either multitier_preferred, multitier_strict, or multitarget_preferred, so the resultant replication chain is C → B → A.

In case of such an InfoScale configuration, the following limitations exist:

- If node C fails, the takeover operation may occur on node A instead of node B. In this event, node B does not register as a secondary to node A.
- You can perform a manual takeover node C to node A. In this case, node B does not register as a secondary to node A.

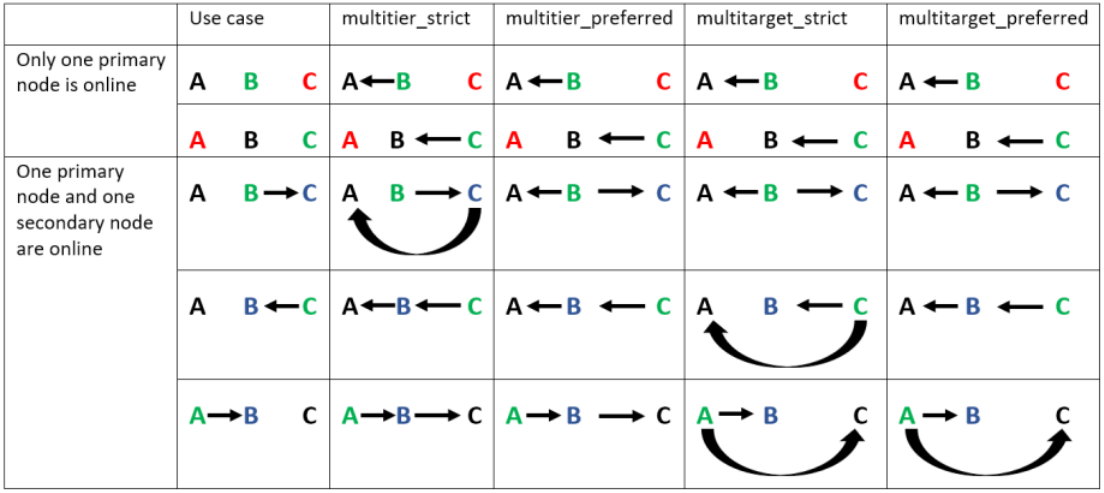
- Secondary node failure in a multitier configuration



The secondary, node B, faults. The replication links between A and B, and between B and C are both broken. When InfoScale detects the failure, it

establishes a replication link between node C and node A. Node C then becomes the direct secondary to node A.

Figure 5-1 Summary of failback use cases for multitarget and multitier configurations based on the Topology attribute value



Color reference for the summary of failback use cases:

- Green indicates a primary node.
- Red indicates a node in the FAULTED or the OFFLINE state.
- Blue indicates a secondary node.
- Black indicates a failback node that is joining the cluster and registering to either a primary or a secondary node.
  When all the other nodes are offline, or one or both of the secondary nodes are hung, this node comes online without registering to any other node.

Integrating HA/DR provider hooks with the SAPHDB agent

Integrating HA/DR provider hooks with the SAPHDB agent helps address the following scenarios:

- Handling split-brain conditions of SAP HANA databases.
- Allowing the failover or the switch operations of the SAP HANA IP service group when the SAP HANA databases are in sync.

- Allowing the SAP HANA IP resource to be brought online only on the Primary. If the resource is brought online on the Secondary, the agent takes it offline, if the HA/DR hooks provide are configured.
- In a scenario where the SAP HANA service group is in the FAULTED state on the Primary, the online operation may be triggered on the SAP HANA IP service group on the Secondary. In such a situation, if the databases are not in sync, the agent clears the fault on the Primary and attempts to bring the SAP HANA IP service group online there.

Note: The integration of HA/DR provider hooks with the SAPHDB agent is mandatory for two-node InfoScale clusters without GCO.

An InfoScale HA/DR solution for SAP HANA databases is effective only when VCS is running on the cluster nodes. If VCS is not running on any either of the two nodes, the agent cannot identify the true Primary during a takeover operation. However, InfoScale does perform an sr_takeover operation in certain situations, so that two SAP HANA instances that are online in an InfoScale cluster are never the Primary at the same time. An example of such a situation would be when both the SAP HANA instances go offline and then the instance on the older Secondary is brought online.

Perform the following tasks on the Primary and the Secondary to integrate the HA/DR provider hooks with the SAPHDB agent:

1. Copy the `/etc/VRTSagents/ha/conf/SAPHDB/SAPHDBSR.py` file to a location that is accessible to the ***SID*adm** user, for example, `/hana/shared/hooks`. The ***SID*adm** user must have read and execute permission on this python script file.
2. Update the `global.ini` file of the SAP HANA database with the following values.
   - HA/DR provider, path, and the SAP HANA IP service group name with the order of execution:

     ```
     [ha_dr_provider_SAPHDBSR]
     provider = SAPHDBSR
     path = /hana/shared/hooks
     takeover_group = SAP_HANA_IP_service_group_name
     execution_order = 1
     ```

   - Optionally, add the trace level:

```
[trace]
ha_dr_saphdbsr = debug
```

For details on configuring the HA/DR provider script, refer to the SAP documentation.

3. Add the administrator user for the SAP HANA IP service group.

   Perform the following steps sequentially:

   - Set the cluster configuration to the read-write mode.

     ```
     # haconf -makerw
     ```

   - Create a user with the administrator privileges.

     ```
     # hauser -add user_name -priv Administrator -group
     takeover_service_group_name
     ```

     Enter a password when prompted.

     This step can be performed on any one of the cluster nodes.

   - Log in with the ***SID*adm** user credentials.

     ```
     # su - SIDadm
     ```

   - Verify that the user has been created successfully.

     ```
     # /opt/VRTSvcs/bin/halogin user_name password_value
     ```

   - Reset the cluster configuration to the read-only mode.

     ```
     # haconf -dump -makero
     ```

   For details about managing VCS users from the command line, refer to the *Cluster Server Administrator's Guide*.

4. Configure the preonline and the postonline triggers for the SAP HANA IP service group.

   See "Configuring SAPHDB preonline script" on page 57.

   See "Configuring SAPHDB postonline script" on page 60.

5. Configure the preonline trigger for the SAP HANA service group.

   See "Configuring SAPHDB preonline script" on page 57.

6. Copy the `resstatechange` script from `/etc/VRTSagents/ha/conf/SAPHDB` to `$VCS_HOME$/bin/triggers`.

7. Configure the `resstatechange` trigger for the SAP HANA IP resource.

   Run the following commands sequentially:

   ```
   # haconf -makerw
   ```

```
# hares -modify SAP_HANA_IP_resource_name TriggersEnabled
RESSTATECHANGE

# hares -modify SAP_HANA_IP_resource_name TriggerPath
trigger_path_value

# haconf -dump -makero
```

8. Verify whether the HA/DR provider hook is working as expected, by performing the following tasks:
   - Perform a switch operation on the SAP HANA IP service group.
   - Verify that the UserStrGlobal attribute is set at the group level and that its value contains the name of the latest Primary.
     For example, `UserStrGlobal = "saphanaserver10;1;2022-04-07_12:09:12"`, where `saphanaserver10` is the hostname or the name of the latest Primary.

Chapter 6

Configuring service groups for SAP HANA Scale-Out systems

This chapter includes the following topics:

About configuring service groups for SAP HANA

Configuring the SAP HANA service group involves creating the SAP service group, its resources, and defining attribute values for the configured resources. You must have administrator privileges to create and configure a service group.

You can configure the service groups using one of the following:

- The Cluster Manager (Java console)
- Veritas Infoscale Operations Manager
- The command line

Before configuring the service groups for SAP HANA

Before you configure the SAP HANA service group, you must:

- Verify that the Cluster Server components are installed and configured on all nodes in the cluster where you will configure the service group.
  For more information on installing the components, refer to the *InfoScale Availability Installation Guide*.
- Verify that SAP HANA is installed and configured identically on all nodes in the cluster.
- Verify that the Cluster Server agent for SAP HANA is installed on all nodes in the cluster.
  See "Installing the agent " on page 24.

Configuring service groups for SAP HANA

The following section describes how to configure service groups in the GCO cluster.

Configuring service groups for SAP HANA within the GCO cluster

Ensure that you meet the following prerequisites to configure service group for SAP HANA under the GCO cluster.

- Configure GCO using the `gcoconfig` command or using CPI.
- Maintain same service group and resource names on all sites of the cluster.
- Passwordless authentication should be configured between all nodes in the clusters.

Note: If passwordless authentication is not configured between all the nodes of a cluster, after sr_takeover on the remote cluster, the agent cannot shut down the old primary SAP HANA instance automatically. You can manually shut down the service group for the old primary instance by using VCS commands. Alternatively, you can write custom a postonline trigger for the global Failover_IP service group to shut down the old primary instance.

Perform the following steps on site 1 to add a service group for SAP HANA

1 Create a service group for SAP HANA.

For example:

```
# hagrp -add SAPHANASR_TEST_SG
```

For more details on creating a service group, refer to the *Cluster Server Administrator's Guide*.

2 Modify the SystemList attribute for the group, to add systems.

```
# hagrp -modify SAPHANASR_TEST_SG SystemList sys1 0 sys2 1 sys3
2

# hagrp -modify SAPHANASR_TEST_SG Parallel 1
```

Similarly, run the following command on site 2 to modify the SystemList attribute for the group.

```
# hagrp -modify SAPHANASR_TEST_SG SystemList sys3 0 sys4 1 sys5
2

# hagrp -modify SAPHANASR_TEST_SG Parallel 1
```

3 Create SAPHDB resources for SAP HANA.

For example:

```
# hares -add BBV_HDB00_REP_RES SAPHDB SAPHANASR_TEST_SG
```

Based on the SAP HANA instance you are clustering, modify the resource attributes. For more information on agent attributes,

See "SAP HANA agent attributes" on page 30.

Note: The values of the OnlineTimeOut, MonitorTimeout, and OfflineTimeout attributes of the SAPHDB agent type should be large enough to accommodate the time required for starting, monitoring, and stopping the SAP HANA instance or performing takeover operation.

For example:

```
# hares -override BBV_HDB00_REP_RES OnlineTimeout

# hares -modify BBV_HDB00_REP_RES OnlineTimeout 900

# hares -override BBV_HDB00_REP_RES MonitorTimeout

# hares -modify BBV_HDB00_REP_RES MonitorTimeout 600

# hares -override BBV_HDB00_REP_RES OfflineTimeout

# hares -modify BBV_HDB00_REP_RES OfflineTimeout 600
```

---

4 Create a service group for SAP HANA.

For example:

```
# hagrp -add SAPHANASR_TEST_SG1
```

5 Modify the SystemList attribute for the group to add systems.

```
# hagrp -modify SAPHANASR_TEST_SG1 SystemList sys1 0 sys2 1 sys3
2

# hagrp -modify SAPHANASR_TEST_SG1 Parallel 1
```

Similarly, run the following command on site 2 to modify the SystemList attribute for the group.

```
# hagrp -modify SAPHANASR_TEST_SG1 SystemList sys3 0 sys4 1 sys5
2

# hagrp -modify SAPHANASR_TEST_SG1 Parallel 1
```

6 Create resources for the sapstartsrv process in the service group.

For example:

```
# hares -add RES_SAPHANA_SR_SAPSTARTSRV SAPHDB SAPHANASR_TEST_SG1
```

7 Create a service group for the SAP HANA IP resource.

For example:

```
# hagrp -add SAPHANA_SR_NETWORK
```

8 Modify the SystemList attribute for the group to add systems.

```
# hagrp -modify SAPHANA_SR_NETWORK SystemList sys1 0 sys2 1 sys3
2

# hagrp -modify SAPHANA_SR_NETWORK Parallel 2
```

Similarly, run the following command on site 2 to modify the SystemList attribute.

```
# hagrp -modify SAPHANA_SR_NETWORK SystemList sys4 0 sys5 1 sys6
2

# hagrp -modify SAPHANA_SR_NETWORK Parallel 2
```

9 Modify the PreOnline attribute of the SAP HANA IP group.

For example:

```
# hagrp -modify SAPHANA_SR_NETWORK PreOnline 1
```

10 Create IP resources in the service group.

For example:

```
# hares -add RES_SAPHANA_MUM_IP IP SAPHANA_SR_NETWORK
```

Note: You must set the RestartLimit attribute for the configured IP resource. You can override the attribute values as follows:

```
# /opt/VRTSvcs/bin/hares -override RES_SAPHANA_MUM_IP RestartLimit

# /opt/VRTSvcs/bin/hares -modify RES_SAPHANA_MUM_IP RestartLimit
2
```

---

11 Create group dependencies for SAP HANA and SAP HANA IP.

For example:

```
# hagrp -link SAPHANA_SR_NETWORK SAPHANASR_TEST_SG1 online local
hard

[root@saphanasr2 config]#hagrp -dep
Parent              Child              Relationship
SAPHANA_SR_NETWORK   SAPHANASR_TEST_SG  online local hard
```

12 Configure the service group with IP resource configured as global.

For example:

```
# hagrp -modify SAPHANA_SR_NETWORK ClusterList saphana_site1 0
saphana_site2 1
```

13 Set the ClusterFailOverPolicy attribute as Manual. In the Auto mode, the agent takes the decision of takeover, whereas manual intervention is required for the actions in the Manual mode.

```
# hagrp -modify SAPHANA_SR_NETWORK ClusterFailOverPolicy Manual
```

Note: If the ClusterFailOverPolicy attribute is set as Auto, the PreOnline trigger can run multiple times on the same system resulting in the execution of the `-sr_takeover` command on the system several times. This may result in an unexpected behaviour

---

14 Perform all the steps in this procedure on site 2.

Note: The value of the OnlineTimeOut attribute of the SAPHDB agent type should be large enough to accommodate the replication time taken either for starting the SAP HANA instance or performing the takeover operation. If the replication delays the starting of the SAP HANA instance or the takeover operation, and the time exceeds the value specified by the OnlineTimeOut attribute, then the SAP HANA instance should be started outside VCS or the takeover operation should be performed.

For GCO, the local group name and resource names for the SAPHDB resources should be the same across the sites. Additionally, the service group for the IP resource can only be the parent service group for the SAPHDB resource.

Configuring SAPHDB preonline script

In a clustered environment, the SAP administrator installs and configures the SAP HANA System Replication. The SAP HANA System Replication has the following requisites:

- If a primary replication site fails, then the secondary replication site must take over the role as primary replication.
- If the primary replication site is switched over manually to a secondary replication site, then the secondary replication site must take over the role of primary replication and the primary replication site is brought down.

The SAPHDB preonline script facilitates proper SAP HANA System Replication takeover behavior. The existing VCS preonline script calls the SAPHDB preonline script.

Note: The preonline script must be configured for a service group for the IP resource.

The SAPHDB preonline script performs the following tasks:

- If the service group for which the script is running does not have child service group for the SAPHDB resources, the script returns the control back to the VCS preonline script.
- If the service group is for the IP resource and has SAPHDB resources in its child service group, the script determines whether the target node is the secondary replication site and performs the takeover action. The script also ensures that the online operation does not execute the VCS preonline script again.

To accomplish this failover behavior, you must configure the VCS preonline script.

To configure the VCS preonline script

1 Create a symlink for the preonline script to the monitor script.

```
cd /opt/VRTSagents/ha/bin/SAPHDB ln -s \
/opt/VRTSagents/ha/bin/SAPHDB/monitor preonline
```

Note: You need to create this link only if the package installer has failed to create it.

2 Navigate to the `$VCS_HOME/bin/triggers` directory.

3 If the VCS preonline trigger script is already present, add the following lines to the main preonline trigger script to integrate the call to the SAPHDB preonline trigger:

```
# SAPHDB specific preonline code: START
# If preonline trigger needs to be run for more than one agent,
# then copy this snippet into the main preonline trigger.

#-------------------
# Define variables..
#-------------------
my $sCmd = '/opt/VRTSagents/ha/bin/SAPHDB/preonline';
my $sResLogLevel = 'TRACE'; # Define logging level..
my @lsCmdArgs = ( @ARGV, $sResLogLevel ); # Insert logging level..
my $sArgs = join ( ' ', @lsCmdArgs );
my $iSAPHDBExitCode = undef;

#-----------------------------------------
# Pass control to preonline, if it exists..
#-----------------------------------------
if ( -x $sCmd ) {
  VCSAG_LOG_MSG ("I", "Preonline Cmd [$sCmd] Args [$sArgs]", \
15031);
  system ( $sCmd, @lsCmdArgs );
  $iSAPHDBExitCode = $? >> 8; # Capture exit code..
  VCSAG_LOG_MSG ("I", "Preonline Cmd [$sCmd] Exited with \
[$iSAPHDBExitCode]", 15031);
  exit 0 unless ( $iSAPHDBExitCode == 1 );
}

#
# SAPHDB specific preonline code: STOP
#
if (defined $ARGV[3]) {
   system("$vcs_home/bin/hagrp -online -nopre $ARGV[1] \
-sys $ARGV[0] -checkpartial $ARGV[3]");
   exit;
}
system("$vcs_home/bin/hagrp -online -nopre $ARGV[1] \
-sys $ARGV[0]");
exit;
```

4 If the VCS preonline trigger script is not present, do the following:

- Pick the sample preonline script from the `/etc/VRTSagents/ha/conf/SAPHDB` directory and copy it in the `$VCS_HOME/bin/triggers` directory.
- Ensure that the file is executable and accessible to the root user.

5 For the service group, set the preonline flag to True.

```
#hagrp -modify service_group PreOnline 1 -sys system
```

The preonline script is now configured to facilitate the SAP HANA replication behavior. To configure the logging level used in the preonline script, you can set the ResLogLevel attribute in the preonline wrapper. You can then view the logs in the VCS engine log, `/var/VRTSvcs/log/engine_A.log`.

Note: Once the preonline trigger is configured, you may see unexpected behavior while manually switching or performing online operations on the SAP HANA instance group. This behavior is a result of the control logic within the preonline trigger that performs the replication takeover, if needed. For system maintenance, if you prefer to perform manual operations on the service groups, you can do so by disabling the preonline trigger.

To disable the preonline trigger, use the following command:

```
#hagrp -modify service_group PreOnline 0 -sys system
```

Chapter 7

Troubleshooting the agent for SAP HANA

This chapter includes the following topics:

Preliminary troubleshooting checks

If you face problems with the Cluster Server agent for SAP HANA, perform the following checks before further investigation:

- Use the correct software and operating system versions.
  Ensure that no issues arise due to incorrect software and operating system versions. For information on the software versions that the agent for SAP HANA supports, see the Veritas Services and Operations Readiness Tools (SORT) site: https://sort.veritas.com/agents.
- Meet prerequisites.
  Before installing the agent for SAP HANA, ensure that all the prerequisites are met. For example, you must install the ACC library on VCS before installing the agent for SAP HANA.
- Configure SAP HANA resources correctly.
  Before using SAP HANA resources, ensure that you configure the resources properly. For a list of attributes used to configure all SAP HANA resources, refer to the agent attributes.

Starting the SAP HANA instance outside a cluster

If you face problems while working with a resource, you must disable the resource within the cluster framework. A disabled resource is not under the control of the cluster framework, and so you can test the SAP HANA instance independent of the cluster framework. Refer to the cluster documentation for information about disabling a resource.

You can then restart the SAP HANA instance outside the cluster framework.

Note: While restarting the SAP HANA instance outside the cluster framework, use the same parameters as that configured for the VCS SAP resource.

To restart the SAP HANA instance outside the cluster framework

1. Log in as a superuser.
2. Ensure that the SAP database is up and running. Refer to the relevant database documentation or consult your database administrator for more information.
3. Use the *SAPSID*adm user to log in to the SAP HANA instance.

   ```
   $ su -SAPSIDadm
   ```

4 Start the SAP HANA instance to run the instance using the following commands:

```
$ sapcontrol -nr instNum -function StartService SAPSID
```

Note: You should run the above command on each system in case of SAP HANA Scale-Out.

Run the following command in case of Scale-Up/single-node SAP HANA deployments.

```
$ sapcontrol -nr instNum -function Start
```

Run the following command in case of Scale-Out SAP HANA deployments.

```
$ sapcontrol -nr instNum -function StartSystem HDB
```

5 Ensure that the SAP HANA instance is running successfully by running the grep command for InstName.

For example, for an SAP HANA instance:

```
$ ps -ef | grep instName
```

As a result all the processes listed in ProcMon, for the instance running on the system, must be displayed.

If the SAP HANA instance is working outside the cluster framework, you can log out of the resource. You can then attempt to restart the SAP HANA instance within the framework.

Reviewing log files

If you face problems while using SAP HANA or the agent for SAP HANA, use the log files described in this section to investigate the problems.

- SAP HANA log files
  The SAP log files are located in the `/usr/sap/SAPSID/instName/hostName` directory.
- Cluster log files
  The engine log file is located at `/var/VRTSvcs/log/engine_A.log`. For a long running cluster, the log files are rotated as `engine_B.log`, `engine_C.log`, and so on. The most-recent engine logs are present in the `engine_A.log` file.
- SAP HANA agent log files

The SAP HANA agent log file is located at `/var/VRTSvcs/log/SAPHDB_A.log`.

Using trace level logging

The ResLogLevel attribute controls the level of logging that is written in a cluster log file for each SAP HANA resource. You can set this attribute to TRACE, which enables very detailed and verbose logging.

If you set the ResLogLevel attribute to TRACE, a very high volume of messages are produced. Veritas recommends that you localize the ResLogLevel attribute for a particular resource.

Warning: You may consider temporarily increasing the timeout values for SAPHDB for debugging purposes. After the debugging process is complete, you can revert back to the original timeout values.

The LogDbg attribute should be used to enable the debug logs for the ACCLib-based agents when the ACCLIB version is 6.2.0.0 or later and the VCS version is 6.2 or later.

To localize ResLogLevel attribute for a resource

1. Identify the resource for which you want to enable detailed logging.
2. Localize the ResLogLevel attribute for the identified resource:

   ```
   # hares -local Resource_Name ResLogLevel
   ```

3. Set the ResLogLevel attribute to TRACE for the identified resource:

   ```
   # hares -modify Resource_Name ResLogLevel TRACE -sys SysA
   ```

4. Note the time before you begin to operate the identified resource.
5. Test the identified resource. The function reproduces the problem that you are attempting to diagnose.
6. Note the time when the problem is reproduced.
7. Set the ResLogLevel attribute back to INFO for the identified resource:

   ```
   # hares -modify Resource_Name ResLogLevel INFO -sys SysA
   ```

8 Save the configuration changes.

```
# haconf -dump
```

9 Review the contents of the log file. Use the time noted in Step 4 and Step 6 to diagnose the problem.

You can also contact Veritas support for more help.

To enable debug logs for all resources of type SAPHDB

Enable the debug log.

```
# hatype -modify SAPHDB LogDbg DBG_5
```

To override the LogDbg attribute at resource level

Override the LogDbg attribute at the resource level and enable the debug logs for the specific resource.

```
# hares -override hh LogDbg
# hares -modify hh LogDbg DBG_5
```

Using trace level logging for preonline trigger

While executing the preonline trigger, you can set the ResLogLevel attribute to TRACE, to enable detailed logging.

See "Configuring SAPHDB preonline script" on page 57.

To set the ResLogLevel attribute for preonline trigger

1 Go to the `$VCS_HOME/bin/triggers` directory.

2 Open the preonline file, and go to this section:

```
#-------------------
# Define variables..
#-------------------
my $sCmd = '/opt/VRTSagents/ha/bin/SAPHDB/preonline';
my $sResLogLevel = 'INFO'; # Define logging level..
my @lsCmdArgs = ( @ARGV, $sResLogLevel ); # Insert logging level..
my $sArgs = join ( ' ', @lsCmdArgs );
my $iExitCode = undef;
```

3 Edit the value of the ResLogLevel attribute:

```
#-------------------
# Define variables..
#-------------------
my $sCmd = '/opt/VRTSagents/ha/bin/SAPHDB/preonline';
my $sResLogLevel = 'TRACE'; # Define logging level..
my @lsCmdArgs = ( @ARGV, $sResLogLevel ); # Insert logging level..
my $sArgs = join ( ' ', @lsCmdArgs );
my $iExitCode = undef;
```

4 Save and close the preonline file.

You can view the logs in the VCS engine log at `/var/VRTSvcs/log/engine_A.log` and the agent log at `/var/VRTSvcs/log/SAPHDB_A.log`.

Appendix A

Sample Configurations for Scale-Up/single-node SAP HANA deployments

This appendix includes the following topics:

About sample configurations for the agents for SAP HANA

The sample configuration graphically depicts the resource types, resources, and resource dependencies within the service group. Review these dependencies carefully before configuring the agents for SAP HANA. For more information about these resource types, refer to the *Cluster Server Bundled Agents Reference Guide*.

Sample agent type definition

After importing the agent type file into the cluster, if you save the configuration on your system disk using the `haconf -dump` command, you can find the

SAPHDBTypes.cf file in the `/etc/VRTSvcs/conf/config` cluster configuration directory.

An excerpt from this file is as follows:

```
type SAPHDB (
    static int IMF{} = { Mode=2, MonitorFreq=5, RegisterRetryLimit=3 }
    static str IMFRegList[] = { InstProfile }
    static str AgentDirectory = "/opt/VRTSagents/ha/bin/SAPHDB"
    static str AgentFile = "/opt/VRTSvcs/bin/Script60Agent"
    static keylist SupportedActions = { GetReplicationStatusInfo,
      RunSrEnable }
    static str ArgList[] = { ResLogLevel, State, IState, EnvFile,
      InstProfile, ProcMon, SystemReplicationMode, SystemReplication,
      TakeOverInInactiveState, DifferenceInTimestamp,
      HANAScaleOutSupport, RegistrationOfSecondary, IsDistributedSystem,
      DBUser, DBPassword, TenantDatabaseName, CustomQuery,
      WaitToShutdownOldPrimary, GetReplicationStatusTimeout,
      MonitorProgram, UseSystemD, EnableReverseReplication,
      RunHdbSqlQuery, TakeoverWithHandshake, ReplicationModeTuples,
      Topology, WaitForPrimarySGOnline, CheckSyncStateBeforeRoleChange }
    static boolean AEPTimeout = 1
    str ResLogLevel = INFO
    str EnvFile
    str InstProfile
    str ProcMon
    str SystemReplicationMode
    boolean SystemReplication = 0
    boolean TakeOverInInactiveState = 0
    int DifferenceInTimestamp = 65535
    boolean HANAScaleOutSupport = 0
    boolean RegistrationOfSecondary = 1
    boolean IsDistributedSystem = 0
    str DBUser
    str DBPassword
    str TenantDatabaseName
    str CustomQuery
    int WaitToShutdownOldPrimary = 60
    int GetReplicationStatusTimeout
    str MonitorProgram
    boolean UseSystemD = 0
    boolean EnableReverseReplication = 0
    boolean RunHdbSqlQuery = 1
    boolean TakeoverWithHandshake = 1
```

```
    str ReplicationModeTuples{}
    str Topology
    int WaitForPrimarySGOnline = 60
    boolean CheckSyncStateBeforeRoleChange = 0
)
```

Sample SAP resource configuration

Given the number of possible SAP resource configurations, this section provides sample working examples that configure a specific SAP HANA instance for Add-In installations.

Sample resource configuration when GCO is not enabled

A sample excerpt from the `main.cf` file is shown below for SAP HANA resource configuration when GCO is not enabled.

```
include "types.cf"
include "SAPHDBTypes.cf"

cluster saphanarep (
   UserNames = { admin = hqrJqlQnrMrrPzrLqo }
   Administrators = { admin }
)

system sysA (
)

system sysB (
)

group BBV_HDB00_REP_SG (
   SystemList = { sysA = 0, sysB = 1 }
   Parallel = 1
)

SAPHDB BBV_HDB00_REP_RES (
   EnvFile = "/usr/sap/BBV/HDB00/hdbenv.sh"
   InstProfile @sysA = "/usr/sap/BBV/SYS
   /profile/BBV_HDB00_sysA"
   InstProfile @sysB = "/usr/sap/BBV/SYS
   /profile/BBV_HDB00_sysB"
   SystemReplicationMode = sync
```

```
   SystemReplication = 1
   EnableReverseReplication = 1
)

SAPHDB BBV_HDB00_REP_sapstartsrv (
   EnvFile = "/usr/sap/BBV/HDB00/hdbenv.sh"
   InstProfile @sysA = "/usr/sap/BBV/SYS
   /profile/BBV_HDB00_sysA"
   InstProfile @sysB = "/usr/sap/BBV/SYS
   /profile/BBV_HDB00_sysB"
   ProcMon = sapstartsrv
)

BBV_HDB00_REP_RES requires BBV_HDB00_REP_sapstartsrv


// resource dependency tree
//
//      group BBV_HDB00_REP_SG
//      {
//      SAPHDB BBV_HDB00_REP_RES
//          {
//          SAPHDB BBV_HDB00_REP_sapstartsrv
//          }
//      }


group SAP_IP_REP (
   SystemList = { sysA = 0, sysB = 1 }
   PreonlineTimeout = 60
   PreOnline = 1
   TriggersEnabled = { PREONLINE, POSTONLINE }
)

IP SAP_IP_RES (
   Device = eth0
   Address = "110.20.77.11"
   NetMask = "255.255.255.0"
)

NIC SAP_NIC_RES (
   Device = eth0
)
```

```
SAP_IP_RES requires SAP_NIC_RES

requires group BBV_HDB00_REP_SG online local hard


// resource dependency tree
//
//      group SAP_IP_REP
//      {
//      IP SAP_IP_RES
//      }
```

Sample resource configuration when GCO is enabled

When GCO is enabled on two nodes

An excerpt form the sample `main.cf` file:

Site 1

```
include "types.cf"
include "SAPHDBTypes.cf"

cluster saphana_Site1 (
   UserNames = { admin = hOPhOJoLPkPPnXPjOM }
   ClusterAddress = "110.120.62.18"
   Administrators = { admin }
)

remotecluster saphana_Site2 (
   ClusterAddress = "210.209.79.47"
)

heartbeat Icmp (
    ClusterList = { saphana_Site2 }
    Arguments @saphana_Site2 = { "210.209.79.47" }
)

system saphanasr2 (
)

group ClusterService (
    SystemList = { saphanasr2 = 0 }
```

```
    AutoStartList = { saphanasr2 }
    OnlineRetryLimit = 3
    OnlineRetryInterval = 120
)

Application wac (
    StartProgram = "/opt/VRTSvcs/bin/wacstart"
    StopProgram = "/opt/VRTSvcs/bin/wacstop"
    MonitorProcesses = { "/opt/VRTSvcs/bin/wac" }
    RestartLimit = 3
)

IP gcoip (
  Device = eth0
  Address = "110.120.62.18"
  NetMask = "255.255.252.0"
)

NIC gconic (
   Device = eth0
)

gcoip requires gconic
wac requires gcoip

// resource dependency tree
//
//      group ClusterService
//      {
//      Application wac
//          {
//          IP gcoip
//              {
//              NIC gconic
//              }
//          }
//      }

group SAPHANASR_TEST_SG (
    SystemList = { saphanasr2 = 0 }
    Parallel = 1
    ClusterFailOverPolicy = Auto
 )
```

```
 SAPHDB RES_SAPHANA_SR (
    EnvFile = "/usr/sap/XYZ/HDB01/hdbenv.sh"
    InstProfile = "/usr/sap/XYZ/SYS/profile/
    XYZ_HDB01_saphanasr2"
    SystemReplicationMode = sync
    SystemReplication = 1
 )

 SAPHDB RES_SAPHANA_SR_SAPSTARTSRV (
    EnvFile = "/usr/sap/XYZ/HDB01/hdbenv.sh"
    InstProfile = "/usr/sap/XYZ/SYS/profile/
    XYZ_HDB01_saphanasr2"
    ProcMon = sapstartsrv
 )

 RES_SAPHANA_SR requires RES_SAPHANA_SR_SAPSTARTSRV

  // resource dependency tree
  //
  //      group SAPHANASR_TEST_SG
  //      {
  //      SAPHDB RES_SAPHANA_SR
  //          {
  //          SAPHDB RES_SAPHANA_SR_SAPSTARTSRV
  //          }
  //      }

group SAPHANA_SR_NETWORK (
    SystemList = { saphanasr2 = 0 }
    ClusterList = { saphana_Site1 = 0,
    saphana_Site2 = 1 }
    Authority = 1
    ClusterFailOverPolicy = Auto
    PreOnline = 1
 )

IP RES_SAPHANA_SITE1_IP (
    Device = eth0
    Address = "110.209.70.42"
    NetMask = "255.255.252.0"
 )
```

```
NIC RES_SAPHANA_SITE1_NIC (
    Device = eth0
    )

requires group SAPHANASR_TEST_SG online local hard
RES_SAPHANA_SITE1_IP requires RES_SAPHANA_SITE1_NIC

// resource dependency tree
//
//      group SAPHANA_SR_NETWORK
//      {
//      IP RES_SAPHANA_SITE1_IP
//          {
//          NIC RES_SAPHANA_SITE1_NIC
//          }
//      }
```

Site 2

```
include "types.cf"
include "SAPHDBTypes.cf"

cluster saphana_Site2 (
    UserNames = { admin = ajkCjeJgkFkkIskEjh }
    ClusterAddress = "210.209.79.47"
    Administrators = { admin }
)

remotecluster saphana_Site1 (
    ClusterAddress = "110.120.62.18"
)

heartbeat Icmp (
    ClusterList = { saphana_Site1 }
    Arguments @saphana_Site1 = { "110.120.62.18" }
)

system saphanasr1 (
)

group ClusterService (
    SystemList = { saphanasr1 = 0 }
    AutoStartList = { saphanasr1 }
    OnlineRetryLimit = 3
```

```
    OnlineRetryInterval = 120
)

Application wac (
    StartProgram = "/opt/VRTSvcs/bin/wacstart"
    StopProgram = "/opt/VRTSvcs/bin/wacstop"
    MonitorProcesses = { "/opt/VRTSvcs/bin/wac" }
    RestartLimit = 3
)

IP gcoip (
    Device = eth0
    Address = "210.209.79.47"
    NetMask = "255.255.252.0"
)

NIC gconic (
    Device = eth0
)

gcoip requires gconic
wac requires gcoip

// resource dependency tree
//
//      group ClusterService
//      {
//      Application wac
//          {
//          IP gcoip
//              {
//              NIC gconic
//              }
//          }
//      }

group SAPHANASR_TEST_SG (
    SystemList = { saphanasr1 = 0 }
    Parallel = 1
    Authority = 1
)

SAPHDB RES_SAPHANA_SR (
```

```
    EnvFile = "/usr/sap/XYZ/HDB01/hdbenv.sh"
    InstProfile = "/usr/sap/XYZ/SYS/profile/
      XYZ_HDB01_saphanasr1"
    SystemReplicationMode = sync
    SystemReplication = 1
)

SAPHDB RES_SAPHANA_SR_SAPSTARTSRV (
    EnvFile = "/usr/sap/XYZ/HDB01/hdbenv.sh"
    InstProfile = "/usr/sap/XYZ/SYS/profile/
      XYZ_HDB01_saphanasr1"
    ProcMon = sapstartsrv
)

RES_SAPHANA_SR requires RES_SAPHANA_SR_SAPSTARTSRV

// resource dependency tree
//
//      group SAPHANASR_TEST_SG
//      {
//      SAPHDB RES_SAPHANA_SR
//          {
//          SAPHDB RES_SAPHANA_SR_SAPSTARTSRV
//          }
//      }

group SAPHANA_SR_NETWORK (
    SystemList = { saphanasr1 = 0 }
    ClusterList = { saphana_Site1 = 1, saphana_Site2 = 0 }
    ClusterFailOverPolicy = Auto
    PreOnline = 1
)

IP RES_SAPHANA_IP (
    Device = eth0
    Address = "210.209.79.178"
    NetMask = "255.255.252.0"
)

NIC RES_SAPHANA_NIC (
    Device = eth0
)
```

```
requires group SAPHANASR_TEST_SG online local hard
RES_SAPHANA_IP requires RES_SAPHANA_NIC

// resource dependency tree
//
//      group SAPHANA_SR_NETWORK
//      {
//      IP RES_SAPHANA_IP
//          {
//          NIC RES_SAPHANA_NIC
//          }
//      }
```

When GCO is enabled on three nodes

An excerpt form the sample `main.cf` file.

Site 1

```
include "types.cf"
include "SAPHDBTypes.cf"

cluster saphana_Site1 (
    UserNames = { admin = hqrJqlQnrMrrPzrLqo }
    ClusterAddress = "10.209.79.178"
    Administrators = { admin }
    HacliUserLevel = COMMANDROOT
    )

remotecluster saphana_Site2 (
    ClusterAddress = "10.209.79.179"
    )

heartbeat Icmp (
    ClusterList = { saphana_Site2 }
    Arguments @saphana_Site2 = { "10.209.79.179" }
    )

system saphanasr1 (
    )

system saphanasr2 (
    )

group BBV_HDB00_REP_SG (
```

```
    SystemList = { saphanasr1 = 0, saphanasr2 = 1 }
    Parallel = 1
    )

SAPHDB BBV_HDB00_REP_RES (
    ResLogLevel = TRACE
    EnvFile = "/usr/sap/CCV/HDB00/hdbenv.sh"
    InstProfile @saphanasr1 = "/usr/sap/CCV/SYS/profile/
      CCV_HDB00_saphanasr1"
    InstProfile @saphanasr2 = "/usr/sap/CCV/SYS/profile/
      CCV_HDB00_saphanasr2"
    SystemReplication = 1
    ReplicationModeTuples @saphanasr1 = { saphanasr2 = sync,
      saphanasr3 = async }
    ReplicationModeTuples @saphanasr2 = { saphanasr1 = sync,
      saphanasr3 = async }
    Topology = multitier_preferred
    )

SAPHDB BBV_HDB00_REP_sapstartsrv (
    Critical = 0
    ResLogLevel = TRACE
    EnvFile = "/usr/sap/CCV/HDB00/hdbenv.sh"
    InstProfile @saphanasr1 = "/usr/sap/CCV/SYS/profile/
      CCV_HDB00_saphanasr1"
    InstProfile @saphanasr2 = "/usr/sap/CCV/SYS/profile/
      CCV_HDB00_saphanasr2"
    ProcMon = sapstartsrv
    )

BBV_HDB00_REP_RES requires BBV_HDB00_REP_sapstartsrv

// resource dependency tree
//
// group BBV_HDB00_REP_SG
// {
// SAPHDB BBV_HDB00_REP_RES
//     {
//     SAPHDB BBV_HDB00_REP_sapstartsrv
//     }
// }

group ClusterService (
```

```
    SystemList = { saphanasr2 = 0, saphanasr1 = 0 }
    AutoStartList = { saphanasr2, saphanasr1 }
    OnlineRetryLimit = 3
    OnlineRetryInterval = 120
    )

Application wac (
    StartProgram = "/opt/VRTSvcs/bin/wacstart"
    StopProgram = "/opt/VRTSvcs/bin/wacstop"
    MonitorProcesses = { "/opt/VRTSvcs/bin/wac" }
    RestartLimit = 3
    )

IP gcoip (
    Device = eth0
    Address = "10.209.79.178"
    NetMask = "255.255.252.0"
    )

NIC gconic (
    Device = eth0
    )

gcoip requires gconic
wac requires gcoip

// resource dependency tree
//
// group ClusterService
// {
// Application wac
//     {
//     IP gcoip
//         {
//         NIC gconic
//         }
//     }
// }

group SAP_IP_REP (
    SystemList = { saphanasr1 = 0, saphanasr2 = 1 }
    PreonlineTimeout = 60
    ClusterList = { saphana_Site1 = 0, saphana_Site2 = 1 }
```

```
    Authority = 1
    ClusterFailOverPolicy = Auto
    PreOnline = 1
    )

IP SAP_IP_RES (
    Device = eth0
    Address = "10.209.79.177"
    NetMask = "255.255.252.0"
    )

requires group BBV_HDB00_REP_SG online local hard

// resource dependency tree
//
// group SAP_IP_REP
// {
// IP SAP_IP_RES
// }
```

Site 2

```
include "types.cf"
include "SAPHDBTypes.cf"

cluster saphana_Site2 (
    UserNames = { admin = cpqIpkPmqLqqOyqKpn }
    ClusterAddress = "10.209.79.179"
    Administrators = { admin }
    )

remotecluster saphana_Site1 (
    ClusterAddress = "10.209.79.178"
    )

heartbeat Icmp (
    ClusterList = { saphana_Site1 }
    Arguments @saphana_Site1 = { "10.209.79.178" }
    )

system saphanasr3 (
    )

group BBV_HDB00_REP_SG (
```

```
    SystemList = { saphanasr3 = 0 }
    )

SAPHDB BBV_HDB00_REP_RES (
    ResLogLevel = TRACE
    EnvFile = "/usr/sap/CCV/HDB00/hdbenv.sh"
    InstProfile = "/usr/sap/CCV/SYS/profile/
      CCV_HDB00_saphanasr3"
    ReplicationModeTuples  = { saphanasr1 = async,
      saphanasr2 = async }
    Topology = multitier_preferred
    )

SAPHDB BBV_HDB00_REP_sapstartsrv (
    ResLogLevel = TRACE
    EnvFile = "/usr/sap/CCV/HDB00/hdbenv.sh"
    InstProfile = "/usr/sap/CCV/SYS/profile/
      CCV_HDB00_saphanasr3"
    ProcMon = sapstartsrv
    )

BBV_HDB00_REP_RES requires BBV_HDB00_REP_sapstartsrv

// resource dependency tree
//
// group BBV_HDB00_REP_SG
// {
// SAPHDB BBV_HDB00_REP_RES
//     {
//     SAPHDB BBV_HDB00_REP_sapstartsrv
//     }
// }

group ClusterService (
    SystemList = { saphanasr3 = 0 }
    AutoStartList = { saphanasr3 }
    OnlineRetryLimit = 3
    OnlineRetryInterval = 120
    )

Application wac (
    StartProgram = "/opt/VRTSvcs/bin/wacstart"
    StopProgram = "/opt/VRTSvcs/bin/wacstop"
```

```
    MonitorProcesses = { "/opt/VRTSvcs/bin/wac" }
    RestartLimit = 3
    )

IP gcoip (
    Device = eth0
    Address = "10.209.79.179"
    NetMask = "255.255.252.0"
    )

NIC gconic (
    Device = eth0
    )

gcoip requires gconic
wac requires gcoip

// resource dependency tree
//
// group ClusterService
// {
// Application wac
//     {
//     IP gcoip
//         {
//         NIC gconic
//         }
//     }
// }

group SAP_IP_REP (
    SystemList = { saphanasr3 = 0 }
    ClusterList = { saphana_Site1 = 0, saphana_Site2 = 1 }
    ClusterFailOverPolicy = Auto
    PreOnline = 1
    )

IP SAP_IP_RES (
    Device = eth0
    Address = "10.209.79.177"
    NetMask = "255.255.252.0"
    )
```

```
requires group BBV_HDB00_REP_SG online local hard

// resource dependency tree
//
// group SAP_IP_REP
// {
// IP SAP_IP_RES
// }
```

Sample SAP HANA Database instance

An excerpt of the `main.cf` file for an SAP HANA Database instance is as follows:

```
SAPHDB BBV_HDB00_REP_RES (
   EnvFile = "/usr/sap/BBV/HDB00/hdbenv.sh"
   InstProfile @sysA = "/usr/sap/BBV/SYS/profile/BBV_HDB00_sysA"
   InstProfile @sysB = "/usr/sap/BBV/SYS/profile/BBV_HDB00_sysB"
   SystemReplicationMode = sync
   SystemReplication = 1
)

SAPHDB BBV_HDB00_REP_sapstartsrv (
   EnvFile = "/usr/sap/BBV/HDB00/hdbenv.sh"
   InstProfile @sysA = "/usr/sap/BBV/SYS/profile/BBV_HDB00_sysA"
   InstProfile @sysB = "/usr/sap/BBV/SYS/profile/BBV_HDB00_sysB"
   ProcMon = sapstartsrv
)
```

Sample SAP HANA Database instance with tenant databases

An excerpt of the `main.cf` file for an SAP HANA database instance with a system database and two tenant databases is as follows:

```
SAPHDB SAPHDB_RES (
    EnvFile = "/usr/sap/MLM/HDB00/hdbenv.sh"
    InstProfile @saphanavm10 = "/usr/sap/MLM/SYS/profile/
      MLM_HDB00_saphanavm10"
    InstProfile @saphanavm11 = "/usr/sap/MLM/SYS/profile/
      MLM_HDB00_saphanavm11"
    ProcMon = hdbindexserver
    SystemReplicationMode = sync
    SystemReplication = 1
    TakeOverInInactiveState = 1
    )
```

```
SAPHDB SAPHDB_TDB_BW9_RES (
    Critical = 0
    EnvFile = "/usr/sap/MLM/HDB00/hdbenv.sh"
    InstProfile @saphanavm10 = "/usr/sap/MLM/SYS/profile/
      MLM_HDB00_saphanavm10"
    InstProfile @saphanavm11 = "/usr/sap/MLM/SYS/profile/
      MLM_HDB00_saphanavm11"
    ProcMon = hdbindexserver
    SystemReplicationMode = sync
    SystemReplication = 1
    TakeOverInInactiveState = 1
    DBUser = dbuserkey
    TenantDatabaseName = BW9
    )

SAPHDB SAPHDB_TDB_ERP_RES (
    ResLogLevel = TRACE
    EnvFile = "/usr/sap/MLM/HDB00/hdbenv.sh"
    InstProfile @saphanavm10 = "/usr/sap/MLM/SYS/profile/
      MLM_HDB00_saphanavm10"
    InstProfile @saphanavm11 = "/usr/sap/MLM/SYS/profile/
      MLM_HDB00_saphanavm11"
    ProcMon = hdbindexserver
    SystemReplicationMode = sync
    SystemReplication = 1
    TakeOverInInactiveState = 1
    DBUser = SYSTEM
    DBPassword = blhNepEhcHdhEhfHg
    TenantDatabaseName = ERP2
    )

SAPHDB_TDB_BW9_RES requires SAPHDB_RES
SAPHDB_TDB_ERP_RES requires SAPHDB_RES


// resource dependency tree
//
//      group SAPHDB_SG
//      {
//      SAPHDB SAPHDB_TDB_BW9_RES
//          {
//          SAPHDB SAPHDB_RES
```

```
    //          }
    //      SAPHDB SAPHDB_TDB_ERP_RES
    //          {
    //          SAPHDB SAPHDB_RES
    //          }
    //      }
```

An excerpt of the `main.cf` file for a sample SAPHDB resource with multiple database tenants is as follows:

```
SAPHDB hana_pb6_res (
    ResLogLevel = TRACE
    EnvFile = "/usr/sap/MLM/HDB00/hdbenv.sh"
    InstProfile @sys14 = "/usr/sap/MLM/SYS/profile/MLM_HDB00_sys14"
    InstProfile @sys15 = "/usr/sap/MLM/SYS/profile/MLM_HDB00_sys15"
    ProcMon = hdbindexserver
    SystemReplicationMode = sync
    SystemReplication = 1
    DBUser = SYSTEM
    DBPassword = CMIoFQfIDiEIGiH
    TenantDatabaseName = "TDB1 TDB2"
)
```

Sample SAP HANA IP resource

A sample excerpt from the `main.cf` file is shown below for the SAP HANA IP resource.

```
IP SAP_IP_RES (
   Device = eth0
   Address = "110.20.77.11"
   NetMask = "255.255.255.0"
)
```

Sample SAP HANA IP service group for cloud environments

AWS

Sample configuration of resources for IP service group in AWS:

```
AWSIP AWSIP_HANA (
    NIC = eth0
    OverlayIP = "10.209.76.15"
    AWSBinDir = "/usr/local/bin"
```

```
    RouteTableIds = { rtb-fb97ac9d, rtb-f416eb8d, rtb-e48be49d }
)

IP HANA_IP (
    Device = eth0
    Address = "10.209.76.15"
    NetMask = "255.255.252.0"
)

NIC HANA_NIC (
    Device = eth0
)

AWSIP_HANA requires HANA_IP
HANA_IP requires HANA_NIC
```

Microsoft Azure

To configure SAPHB in Microsoft Azure, you must configure an AzureAuth service group and an IP service group in parallel.

Sample configuration of resources for parallel AzureAuth service group:

```
AzureAuth AZURE_AUTH (
    SubscriptionId = 2dfgg136-fgh6-40dd-b616-c1e9abdf1d63
    ClientId = 123456-d10a-4704-8986-beb86739104d
    SecretKey = fntPgnUnhTprQrqrnRonSlhPhrQpiNtrItpRhnGrrNklFngLs
    TenantId = 12345-0528-4308-brf03-6667d61dd0e3
)

Phantom PHANTOM (
)
```

Sample configuration of resources for failover IP service group:

```
AzureIP HANA_AZUREIP (
    NICDevice = eth0
    OverlayIP = "10.209.76.15"
    RouteTableResourceIds = { "/subscriptions/
      6940a326-abc6-40dd-b616-d3f9bbdf1d63/resourceGroups/azureRG/
      providers/Microsoft.Network/routeTables/azureroute1",
      "/subscriptions/6940a326-abc6-40dd-b616-d3f9bbdf1d63/
      resourceGroups/azureRG/providers/Microsoft.Network/
      routeTables/azureroute2" }
    AzureAuthResName = AZURE_AUTH
```

```
)

IP HANA_IP (
    Device = eth0
    Address = "10.209.76.15"
    NetMask = "255.255.252.0"
)

NIC HANA_NIC (
    Device = eth0
)

Proxy AUTH_PROXY (
    TargetResName = AZURE_AUTH
)

HANA_AZUREIP requires AUTH_PROXY
HANA_IP requires HANA_AZUREIP
AUTH_PROXY requires HANA_NIC
```

GCP

Sample configuration of resources for IP service group on GCP:

```
GoogleIP HANA_GoogleIP (
    Device = eth0
    OverlayIP = "10.209.76.15"
)

IP HANA_IP (
    Device = eth0
    Address = "10.209.76.15"
    NetMask = "255.255.252.0"
)

NIC HANA_NIC (
    Device = eth0
)

HANA_GoogleIP requires HANA_IP
HANA_IP requires HANA_NIC
```

Sample multitarget configuration

```
group HANA_SR_NETWORK (
    SystemList = { pundl360g10-8 = 0, pundl360g10-10 = 1 }
    PreonlineTimeout = 60
    ClusterList = { Mumbai_HANA = 0, Pune_HANA = 1 }
    ClusterFailOverPolicy = Auto
    TriggerPath = "bin/triggers/saphdb"
    TriggersEnabled = { PREONLINE }
    PreOnline = 1
    )

    IP hana_IP (
        Device = eth2
        Address = "10.221.81.185"
        NetMask = "255.255.240.0"
        RestartLimit = 2
        )

    requires group HDB_REP online local hard

    // resource dependency tree
    //
    //      group HANA_SR_NETWORK
    //      {
    //      IP hana_IP
    //      }

group HDB_REP (
    SystemList = { pundl360g10-8 = 0, pundl360g10-10 = 1 }
    UserStrGlobal = STARTED
    Parallel = 1
    Authority = 1
    )

    SAPHDB replication (
        EnvFile = "/usr/sap/MUL/HDB02/hdbenv.sh"
        InstProfile @pundl360g10-8 = "/usr/sap/MUL/SYS/
          profile/MUL_HDB02_pundl360g10-8"
        InstProfile @pundl360g10-10 = "/usr/sap/MUL/SYS/
          profile/MUL_HDB02_pundl360g10-10"
        SystemReplicationMode = sync
        SystemReplication = 1
```

```
        DBUser = SYSTEM
        DBPassword = "ab:c8:2f:59:0d:94:97:a4:a3:ef:f1:fa:
          17:80:67:cf:e6:47:71:2e:22:f1:12:4c:e1:8a:31:dd:
          4f:89:e5:2a"
        TenantDatabaseName = TDB1
        RunHdbSqlQuery = 0
        ReplicationModeTuples @pundl360g10-8 =
          { pundl360g10-9 = syncmem, pundl360g10-10 = syncmem }
        ReplicationModeTuples @pundl360g10-10 =
          { pundl360g10-9 = syncmem, pundl360g10-8 = syncmem }
        Topology = multitier_preferred
        )

    SAPHDB startsrv (
        EnvFile = "/usr/sap/MUL/HDB02/hdbenv.sh"
        InstProfile @pundl360g10-8 = "/usr/sap/MUL/SYS/
          profile/MUL_HDB02_pundl360g10-8"
        InstProfile @pundl360g10-10 = "/usr/sap/MUL/SYS/
          profile/MUL_HDB02_pundl360g10-10"
        ProcMon = sapstartsrv
        SystemReplicationMode = syncmem
        DifferenceInTimestamp = 1000
        )

    replication requires startsrv

    // resource dependency tree
    //
    //      group HDB_REP
    //      {
    //      SAPHDB replication
    //          {
    //          SAPHDB startsrv
    //          }
    //      }
```

Sample configuration with HA/DR provider hook

```
group SAP_IP_REP (
    SystemList = { sysA = 0, sysB = 1 }
    PreOnline = 1
    TriggersEnabled = { PREONLINE, POSTONLINE }
)
```

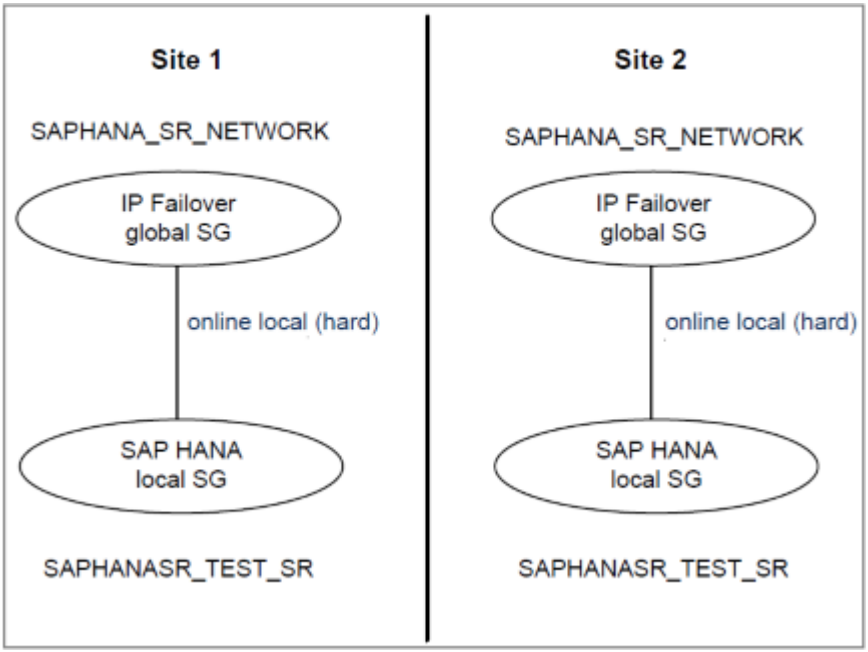
```
IP SAP_IP_RES (
    TriggersEnabled = { RESSTATECHANGE }
    Device = eth0
    Address = "110.20.77.11"
    NetMask = "255.255.255.0"
)

NIC SAP_NIC_RES (
    Device = eth0
)

SAP_IP_RES requires SAP_NIC_RES
requires group BBV_HDB00_REP_SG online local hard

// resource dependency tree
//
// group SAP_IP_REP
// {
// IP SAP_IP_RES
// }

group BBV_HDB00_REP_SG (
    SystemList = { sysA = 0, sysA = 1 }
    Parallel = 1
    TriggersEnabled = { PREONLINE }
)

SAPHDB BBV_HDB00_REP_RES (
    ResLogLevel = TRACE
    EnvFile = "/usr/sap/BBV/HDB02/hdbenv.sh"
    InstProfile @sysA = "/hana/shared/BBV/profile/BBV_HDB02_sysA"
    InstProfile @sysB = "/hana/shared/BBV/profile/BBV_HDB02_sysB"
    ProcMon = "hdbindexserver"
    SystemReplicationMode = sync
    SystemReplication = 1
    EnableReverseReplication = 1
    WaitForPrimarySGOnline = 120
    CheckSyncStateBeforeRoleChange = 1
 )

SAPHDB BBV_HDB00_REP_sapstartsrv (
    EnvFile = "/usr/sap/BBV/HDB02/hdbenv.sh"
```

```
    InstProfile @sysA = "/hana/shared/BBV/profile/BBV_HDB02_sysA"
    InstProfile @sysB = "/hana/shared/BBV/profile/BBV_HDB02_sysB"
    ProcMon = sapstartsrv
)

BBV_HDB00_REP_RES requires BBV_HDB00_REP_sapstartsrv

// resource dependency tree
//
//      group BBV_HDB00_REP_SG
//      {
//      SAPHDB BBV_HDB00_REP_RES
//          {
//          SAPHDB BBV_HDB00_REP_sapstartsrv
//          }
//      }
```

Sample service group dependency

Figure A-1 shows the sample service group dependency for SAP HANA when GCO is enabled.

Figure A-1 SAP HANA System Replication (GCO)



Sample resource dependency of SAP HANA IP service groups for cloud environments

Figure A-2 AWS

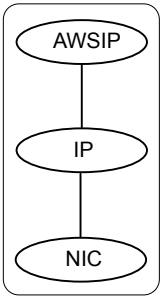


Figure A-3 Azure

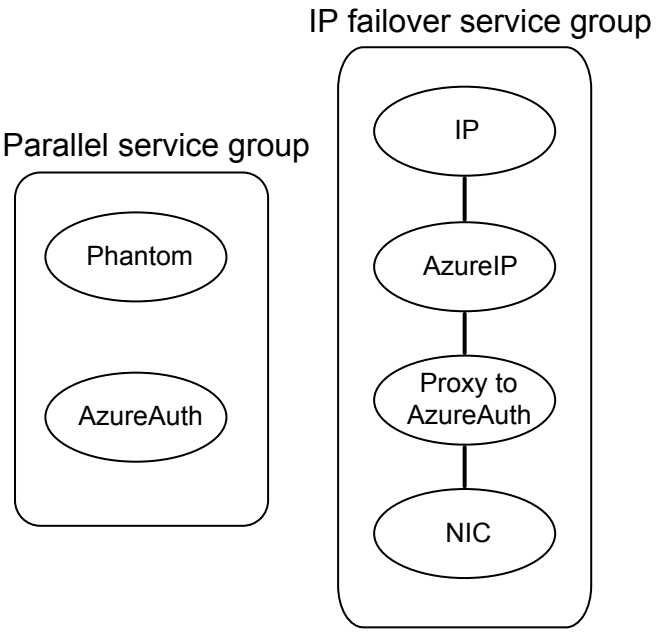
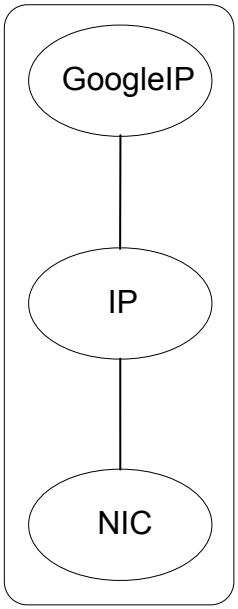


Figure A-4 GCP



Appendix B

Sample Configurations for Scale-Out systems

This appendix includes the following topics:

About sample configurations for the agents for SAP HANA

The sample configuration graphically depicts the resource types, resources, and resource dependencies within the service group. Review these dependencies carefully before configuring the agents for SAP HANA. For more information about these resource types, refer to the *Cluster Server Bundled Agents Reference Guide*.

Sample agent type definition

After importing the agent type file into the cluster, if you save the configuration on your system disk using the `haconf -dump` command, you can find the SAPHDBTypes.cf file in the `/etc/VRTSvcs/conf/config` cluster configuration directory.

An excerpt from this file is as follows:

```
type SAPHDB (
    static int IMF{} = { Mode=2, MonitorFreq=5, RegisterRetryLimit=3 }
    static str IMFRegList[] = { InstProfile }
    static str AgentDirectory = "/opt/VRTSagents/ha/bin/SAPHDB"
    static str AgentFile = "/opt/VRTSvcs/bin/Script60Agent"
    static keylist SupportedActions = { GetReplicationStatusInfo }
    static str ArgList[] = { ResLogLevel, State, IState, EnvFile,
      InstProfile, ProcMon, SystemReplicationMode, SystemReplication,
      TakeOverInInactiveState, DifferenceInTimestamp,
      HANAScaleOutSupport, RegistrationOfSecondary,
      IsDistributedSystem, DBUser, DBPassword, TenantDatabaseName,
      CustomQuery, WaitToShutdownOldPrimary,
      GetReplicationStatusTimeout, MonitorProgram, UseSystemD,
      EnableReverseReplication, RunHdbSqlQuery, TakeoverWithHandshake }
    static boolean AEPTimeout = 1
    str ResLogLevel = INFO
    str EnvFile
    str InstProfile
    str ProcMon
    str SystemReplicationMode
    boolean SystemReplication = 0
    boolean TakeOverInInactiveState = 0
    int DifferenceInTimestamp = 65535
    boolean HANAScaleOutSupport = 0
    boolean RegistrationOfSecondary = 1
    boolean IsDistributedSystem = 0
    str DBUser
    str DBPassword
    str TenantDatabaseName
    str CustomQuery
    int WaitToShutdownOldPrimary = 60
    int GetReplicationStatusTimeout
    str MonitorProgram
    boolean UseSystemD = 0
    boolean EnableReverseReplication = 0
    boolean RunHdbSqlQuery = 1
    boolean TakeoverWithHandshake = 1
)
```

Sample SAP resource configuration

Given the number of possible SAP resource configurations, this section provides sample working examples that configure a specific SAP HANA instance for Add-In installations.

Sample resource configuration when GCO is enabled

An excerpt form the sample main.cf file when GCO is configured.

Site 1:

```
include "types.cf"
include "CFSTypes.cf"
include "CRSResource.cf"
include "CSSD.cf"
include "CVMTypes.cf"
include "SAPHDBTypes.cf"
include "SAPNWTypes.cf"

cluster sapclus_primary (
        UserNames = { admin = gpqIpkPmqLqqOyqKpn }
        ClusterAddress = "10.209.62.60"
        Administrators = { admin }
        HacliUserLevel = COMMANDROOT
        )

remotecluster sapclus_secondary (
        ClusterAddress = "10.209.79.129"
        )

heartbeat Icmp (
        ClusterList = { sapclus_secondary }
        Arguments @sapclus_secondary = { "10.209.79.129" }
        )

system vcslx662-v09 (
        )

system vcslx662-v10 (
        )

system vcslx662-v11 (
        )
```

```
group ClusterService (
        SystemList = { vcslx662-v09 = 0, vcslx662-v10 = 1,
                       vcslx662-v11 = 2 }
        AutoStartList = { vcslx662-v09, vcslx662-v10,
                          vcslx662-v11 }
        OnlineRetryLimit = 3
        OnlineRetryInterval = 120
        )

        Application wac (
                StopProgram = "/opt/VRTSvcs/bin/wacstop"
                MonitorProcesses = { "/opt/VRTSvcs/bin/wac" }
                StartProgram = "/opt/VRTSvcs/bin/wacstart"
                RestartLimit = 3
                )

        IP webip (
                Address = "10.209.62.60"
                NetMask = "255.255.252.0"
                Device @vcslx662-v09 = eth0
                Device @vcslx662-v10 = eth3
                Device @vcslx662-v11 = eth0
                )

        NIC csgnic (
                Device @vcslx662-v09 = eth0
                Device @vcslx662-v10 = eth3
                Device @vcslx662-v11 = eth0
                )

        wac requires webip
        webip requires csgnic


        // resource dependency tree
        //
        //      group ClusterService
        //      {
        //      Application wac
        //          {
        //          IP webip
        //              {
```

```
        //              NIC csgnic
        //              }
        //          }
        //      }


group SAP_HANA_LOCAL_SG (
        SystemList = { vcslx662-v09 = 0, vcslx662-v10 = 1,
                       vcslx662-v11 = 2 }
        UserStrGlobal = STARTED
        Parallel = 1
        )

        SAPHDB SAP_HANA_LOCAL_RES (
                ResLogLevel = TRACE
                EnvFile = "/usr/sap/HDB/HDB00/hdbenv.sh"
                InstProfile @vcslx662-v09 =
                "/usr/sap/HDB/SYS/profile/HDB_HDB00_vcslx662-v09"
                InstProfile @vcslx662-v10 =
                "/usr/sap/HDB/SYS/profile/HDB_HDB00_vcslx662-v10"
                InstProfile @vcslx662-v11 =
                "/usr/sap/HDB/SYS/profile/HDB_HDB00_vcslx662-v11"
                SystemReplicationMode = async
                SystemReplication = 1
                TakeOverInInactiveState = 1
                HANAScaleOutSupport = 1
                MonitorTimeout = 600
                OnlineTimeout = 800
                OfflineTimeout = 600
                )

        // resource dependency tree
        //
        //      group SAP_HANA_LOCAL_SG
        //      {
        //      SAPHDB SAP_HANA_LOCAL_RES
        //      }

group SAP_HANA_TAKEOVER_IP (
        SystemList = { vcslx662-v09 = 0, vcslx662-v10 = 1,
                       vcslx662-v11 = 2 }
        Parallel = 2
        ClusterList = { sapclus_secondary = 1, sapclus_primary = 0 }
```

```
        Authority = 1
        PreOnline = 1
        )

        IP HANA_IP_RES (
                Address @vcslx662-v09 = "10.209.60.227"
                Address @vcslx662-v10 = "10.209.60.228"
                Address @vcslx662-v11 = "10.209.60.229"
                NetMask = "255.255.252.0"
                Device @vcslx662-v09 = eth0
                Device @vcslx662-v10 = eth3
                Device @vcslx662-v11 = eth0
                RestartLimit = 2
                )

        NIC HANA_NIC_RES (
                Device @vcslx662-v09 = eth0
                Device @vcslx662-v10 = eth3
                Device @vcslx662-v11 = eth0
                )

        requires group SAP_HANA_LOCAL_SG online local hard
        HANA_IP_RES requires HANA_NIC_RES

        // resource dependency tree
        //
        //      group SAP_HANA_TAKEOVER_IP
        //      {
        //      IP HANA_IP_RES
        //          {
        //          NIC HANA_NIC_RES
        //          }
        //      }

group SAP_SAPSTARTSRV_SG (
        SystemList = { vcslx662-v09 = 0, vcslx662-v10 = 1,
                       vcslx662-v11 = 2 }
        UserStrGlobal = FAULTED
        Parallel = 1
        OnlineRetryLimit = 10
        )

        SAPHDB SAP_HANA_LOCAL__SAPSTARTSRV_RES (
```

```
                Critical = 0
                ResLogLevel = TRACE
                EnvFile = "/usr/sap/HDB/HDB00/hdbenv.sh"
                InstProfile @vcslx662-v09 =
                "/usr/sap/HDB/SYS/profile/HDB_HDB00_vcslx662-v09 "
                InstProfile @vcslx662-v10 =
                "/usr/sap/HDB/SYS/profile/HDB_HDB00_vcslx662-v10"
                InstProfile @vcslx662-v11 =
                "/usr/sap/HDB/SYS/profile/HDB_HDB00_vcslx662-v11"
                ProcMon = sapstartsrv
                SystemReplicationMode = async
                SystemReplication = 1
                TakeOverInInactiveState = 1
                HANAScaleOutSupport = 1
                RegistrationOfSecondary = 0
                )

        // resource dependency tree
        //
        //      group SAP_SAPSTARTSRV_SG
        //      {
        //      SAPHDB SAP_HANA_LOCAL__SAPSTARTSRV_RES
        //      }

group cvm (
        SystemList = { vcslx662-v09 = 0, vcslx662-v10 = 1,
                       vcslx662-v11 = 2 }
        AutoFailOver = 0
        Parallel = 1
        AutoStartList = { vcslx662-v09, vcslx662-v10,
                          vcslx662-v11 }
        )

        CFSfsckd vxfsckd (
                )

        CVMCluster cvm_clus (
                CVMClustName = sapclus_primary
                CVMNodeId = { vcslx662-v09 = 0, vcslx662-v10 = 1,
                         vcslx662-v11 = 2 }
                CVMTransport = gab
                CVMTimeout = 200
                )
```

```
        CVMVxconfigd cvm_vxconfigd (
                Critical = 0
                CVMVxconfigdArgs = { syslog }
                )

        ProcessOnOnly vxattachd (
                Critical = 0
                PathName = "/bin/sh"
                Arguments = "- /usr/lib/vxvm/bin/vxattachd root"
                RestartLimit = 3
                )

        cvm_clus requires cvm_vxconfigd
        vxfsckd requires cvm_clus


        // resource dependency tree
        //
        //      group cvm
        //      {
        //      ProcessOnOnly vxattachd
        //      CFSfsckd vxfsckd
        //          {
        //          CVMCluster cvm_clus
        //              {
        //              CVMVxconfigd cvm_vxconfigd
        //              }
        //          }
        //      }
```

Site 2:

```
include "types.cf"
include "CFSTypes.cf"
include "CRSResource.cf"
include "CSSD.cf"
include "CVMTypes.cf"
include "SAPHDBTypes.cf"

cluster sapclus_secondary (
        UserNames = { admin = GnoGniNkoJooMwoInl }
        ClusterAddress = "10.209.79.129"
        Administrators = { admin }
```

```
        HacliUserLevel = COMMANDROOT
        )

remotecluster sapclus_primary (
        ClusterAddress = "10.209.62.60"
        )

heartbeat Icmp (
        ClusterList = { sapclus_primary }
        Arguments @sapclus_primary = { "10.209.62.60" }
        )

system vcsesx004-v04 (
        )

system vcsesx004-v05 (
        )

system vcsesx004-v06 (
        )

group ClusterService (
        SystemList = { vcsesx004-v04 = 0, vcsesx004-v05 = 1,
                       vcsesx004-v06 = 2 }
        AutoStartList = { vcsesx004-v04, vcsesx004-v05,
                          vcsesx004-v06 }
        OnlineRetryLimit = 3
        OnlineRetryInterval = 120
        )

        Application wac (
                StartProgram = "/opt/VRTSvcs/bin/wacstart"
                StopProgram = "/opt/VRTSvcs/bin/wacstop"
                MonitorProcesses = { "/opt/VRTSvcs/bin/wac" }
                RestartLimit = 3
                )

        IP gcoip (
                Device = eth0
                Address = "10.209.79.129"
                NetMask = "255.255.252.0"
                )
```

```
        NIC gconic (
                Device = eth0
                )

        gcoip requires gconic
        wac requires gcoip

        // resource dependency tree
        //
        //      group ClusterService
        //      {
        //      Application wac
        //          {
        //          IP gcoip
        //              {
        //              NIC gconic
        //              }
        //          }
        //      }

group SAP_HANA_LOCAL_SG (
        SystemList = { vcsesx004-v04 = 0, vcsesx004-v05 = 1,
                       vcsesx004-v06 = 2 }
        UserStrGlobal = FAULTED
        Parallel = 1
        )

        SAPHDB SAP_HANA_LOCAL_RES (
                ResLogLevel = TRACE
                EnvFile = "/usr/sap/HDB/HDB00/hdbenv.sh"
                InstProfile @vcsesx004-v04 =
                "/usr/sap/HDB/SYS/profile/HDB_HDB00_vcsesx004-v04"
                InstProfile @vcsesx004-v05 =
                "/usr/sap/HDB/SYS/profile/HDB_HDB00_vcsesx004-v05"
                InstProfile @vcsesx004-v06 =
                "/usr/sap/HDB/SYS/profile/HDB_HDB00_vcsesx004-v06"
                SystemReplicationMode = async
                SystemReplication = 1
                TakeOverInInactiveState = 1
                HANAScaleOutSupport = 1
                MonitorTimeout = 600
                OnlineTimeout = 900
                OfflineTimeout = 600
```

```
        )

    // resource dependency tree
    //
    //      group SAP_HANA_LOCAL_SG
    //      {
    //      SAPHDB SAP_HANA_LOCAL_RES
    //      }

group SAP_HANA_TAKEOVER_IP (
        SystemList = { vcsesx004-v04 = 0, vcsesx004-v05 = 1,
                       vcsesx004-v06 = 2 }
        Parallel = 2
        ClusterList = { sapclus_secondary = 1, sapclus_primary = 0 }
        PreOnline = 1
        )

        IP HANA_IP_RES (
                Device = eth0
                Address @vcsesx004-v04 = "10.209.76.15"
                Address @vcsesx004-v05 = "10.209.76.70"
                Address @vcsesx004-v06 = "10.209.78.75"
                NetMask = "255.255.252.0"
                RestartLimit = 2
                )

        NIC HANA_NIC_RES (
                Device = eth0
                )

        requires group SAP_HANA_LOCAL_SG online local hard
        HANA_IP_RES requires HANA_NIC_RES


        // resource dependency tree
        //
        //      group SAP_HANA_TAKEOVER_IP
        //      {
        //      IP HANA_IP_RES
        //          {
        //          NIC HANA_NIC_RES
        //          }
        //      }
```

```
group SAP_SAPSTARTSRV_SG (
        SystemList = { vcsesx004-v04 = 0, vcsesx004-v05 = 1,
                       vcsesx004-v06 = 2 }
        UserStrGlobal = FAULTED
        Parallel = 1
        AutoStartList = { vcsesx004-v04, vcsesx004-v05,
                          vcsesx004-v06 }
        OnlineRetryLimit = 10
        )

        SAPHDB SAP_HANA_LOCAL__SAPSTARTSRV_RES (
                TriggerResRestart = 1
                Critical = 0
                EnvFile = "/usr/sap/HDB/HDB00/hdbenv.sh"
                InstProfile @vcsesx004-v04 =
                "/usr/sap/HDB/SYS/profile/HDB_HDB00_vcsesx004-v04"
                InstProfile @vcsesx004-v05 =
                "/usr/sap/HDB/SYS/profile/HDB_HDB00_vcsesx004-v05"
                InstProfile @vcsesx004-v06 =
                "/usr/sap/HDB/SYS/profile/HDB_HDB00_vcsesx004-v06"
                ProcMon = sapstartsrv
                SystemReplicationMode = async
                SystemReplication = 1
                TakeOverInInactiveState = 1
                HANAScaleOutSupport = 1
                RegistrationOfSecondary = 0
                )

        // resource dependency tree
        //
        //      group SAP_SAPSTARTSRV_SG
        //      {
        //      SAPHDB SAP_HANA_LOCAL__SAPSTARTSRV_RES
        //      }

group cvm (
        SystemList = { vcsesx004-v04 = 0, vcsesx004-v05 = 1,
                       vcsesx004-v06 = 2 }
        AutoFailOver = 0
        Parallel = 1
        AutoStartList = { vcsesx004-v04, vcsesx004-v05,
                          vcsesx004-v06 }
```

```
)

CFSfsckd vxfsckd (
        )

CVMCluster cvm_clus (
        CVMClustName = sapclus_secondary
        CVMNodeId = { vcsesx004-v04 = 0, vcsesx004-v05 = 1,
                 vcsesx004-v06 = 2 }
        CVMTransport = gab
        CVMTimeout = 200
        )

CVMVxconfigd cvm_vxconfigd (
        Critical = 0
        CVMVxconfigdArgs = { syslog }
        )

ProcessOnOnly vxattachd (
        Critical = 0
        PathName = "/bin/sh"
        Arguments = "- /usr/lib/vxvm/bin/vxattachd root"
        RestartLimit = 3
        )

cvm_clus requires cvm_vxconfigd
vxfsckd requires cvm_clus


// resource dependency tree
//
//      group cvm
//      {
//      ProcessOnOnly vxattachd
//      CFSfsckd vxfsckd
//          {
//          CVMCluster cvm_clus
//              {
//              CVMVxconfigd cvm_vxconfigd
//              }
//          }
//      }
```

Sample SAP HANA Database instance

An excerpt of the `main.cf` file for a SAP HANA database instance is as follows:

```
SAPHDB BBV_HDB00_REP_RES (
   ResLogLevel = TRACE
   EnvFile = "/usr/sap/BBV/HDB00/hdbenv.sh"
   InstProfile @sysA = "/usr/sap/BBV/SYS/profile/BBV_HDB00_sysA"
   InstProfile @sysB = "/usr/sap/BBV/SYS/profile/BBV_HDB00_sysB"
   SystemReplicationMode = sync
   SystemReplication = 1
)

SAPHDB BBV_HDB00_REP_sapstartsrv (
   Critical = 0
   ResLogLevel = TRACE
   EnvFile = "/usr/sap/BBV/HDB00/hdbenv.sh"
   InstProfile @sysA = "/usr/sap/BBV/SYS/profile/BBV_HDB00_sysA"
   InstProfile @sysB = "/usr/sap/BBV/SYS/profile/BBV_HDB00_sysB"
   ProcMon = sapstartsrv
)
```

An excerpt of the `main.cf` file for a SAP HANA database instance when the database content is distributed with no local standby is as follows:

```
SAPHDB SAP_HANA_LOCAL_RES (
    ResLogLevel = TRACE
    EnvFile = "/usr/sap/VRT/HDB00/hdbenv.sh"
    InstProfile @saphanavm5 = "/usr/sap/VRT/SYS/profile/
      VRT_HDB00_saphanavm5"
    InstProfile @saphanavm6 = "/usr/sap/VRT/SYS/profile/
      VRT_HDB00_saphanavm6"
    InstProfile @saphanavm7 = "/usr/sap/VRT/SYS/profile/
      VRT_HDB00_saphanavm7"
    SystemReplicationMode = async
    SystemReplication = 1
    TakeOverInInactiveState = 1
    HANAScaleOutSupport = 1
    RegistrationOfSecondary = 0
    IsDistributedSystem = 1
    DBUser = hanauser
    DBPassword = xxxx
)
```

Sample SAP HANA IP resource

A sample excerpt from the main.cf file is shown below for the SAP HANA IP resource.

```
IP IP_res (
  Device = eth0
  Address @hana1-ds1 = "10.209.141.191"
  Address @hana2-ds1 = "10.209.141.192"
  Address @hana3-ds1 = "10.209.141.193"
  NetMask = "255.255.252.0"
)
```

Sample SAP HANA IP service group for cloud environments

AWS

Sample configuration of resources for IP service group in AWS:

```
AWSIP AWSIP_HANA (
    NIC = eth0
    OverlayIP = "10.209.76.15"
    AWSBinDir = "/usr/local/bin"
    OverlayIP @hana1-ds1 = "10.209.141.191"
    OverlayIP @hana2-ds1 = "10.209.141.192"
    OverlayIP @hana2-ds1 = "10.209.141.193"
    RouteTableIds = { rtb-fb97ac9d, rtb-f416eb8d, rtb-e48be49d }
)

IP HANA_IP (
    Device = eth0
    Address @hana1-ds1 = "10.209.141.191"
    Address @hana2-ds1 = "10.209.141.192"
    Address @hana2-ds1 = "10.209.141.193"
    NetMask = "255.255.252.0"
)

NIC HANA_NIC (
    Device = eth0
)

AWSIP_HANA requires HANA_IP
HANA_IP requires HANA_NIC
```

Microsoft Azure

To configure SAPHB in Microsoft Azure, you must configure an AzureAuth service group and an IP service group in parallel.

Sample configuration of resources for parallel AzureAuth service group:

```
AzureAuth AZURE_AUTH (
    SubscriptionId = 2dfgg136-fgh6-40dd-b616-c1e9abdf1d63
    ClientId = 123456-d10a-4704-8986-beb86739104d
    SecretKey = fntPgnUnhTprQrqrnRonSlhPhrQpiNtrItpRhnGrrNklFngLs
    TenantId = 12345-0528-4308-brf03-6667d61dd0e3
)

Phantom PHANTOM (
)
```

Sample configuration of resources for failover IP service group:

```
AzureIP HANA_AZUREIP (
    NICDevice = eth0
    OverlayIP @hana1-ds1 = "10.209.141.191"
    OverlayIP @hana2-ds1 = "10.209.141.192"
    OverlayIP @hana2-ds1 = "10.209.141.193"
    RouteTableResourceIds = { "/subscriptions/
      6940a326-abc6-40dd-b616-d3f9bbdf1d63/resourceGroups/azureRG/
      providers/Microsoft.Network/routeTables/azureroute1",
      "/subscriptions/6940a326-abc6-40dd-b616-d3f9bbdf1d63/
      resourceGroups/azureRG/providers/Microsoft.Network/
      routeTables/azureroute2" }
    AzureAuthResName = AZURE_AUTH
)

IP HANA_IP (
    Device = eth0
    Address @hana1-ds1 = "10.209.141.191"
    Address @hana2-ds1 = "10.209.141.192"
    Address @hana2-ds1 = "10.209.141.193"
    NetMask = "255.255.252.0"
)

NIC HANA_NIC (
    Device = eth0
)
```

```
Proxy AUTH_PROXY (
    TargetResName = AZURE_AUTH
)

HANA_AZUREIP requires AUTH_PROXY
HANA_IP requires HANA_AZUREIP
AUTH_PROXY requires HANA_NIC
```

GCP

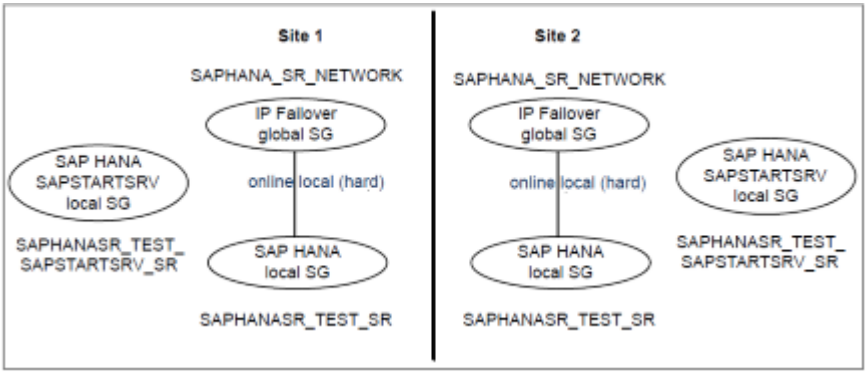
Sample configuration of resources for IP service group on GCP:

```
GoogleIP HANA_GoogleIP (
    Device = eth0
    OverlayIP @hana1-ds1 = "10.209.141.191"
    OverlayIP @hana2-ds1 = "10.209.141.192"
    OverlayIP @hana2-ds1 = "10.209.141.193"
)

IP HANA_IP (
    Device = eth0
    Address @hana1-ds1 = "10.209.141.191"
    Address @hana2-ds1 = "10.209.141.192"
    Address @hana2-ds1 = "10.209.141.193"
    NetMask = "255.255.252.0"
)

NIC HANA_NIC (
    Device = eth0
)

HANA_GoogleIP requires HANA_IP
HANA_IP requires HANA_NIC
```

Sample service group dependency

The following figure depicts the sample service group dependency for SAP HANA when GCO is enabled.

Figure B-1 SAP HANA System Replication (GCO)



Sample resource dependency of SAP HANA IP service groups for cloud environments

Figure B-2 AWS

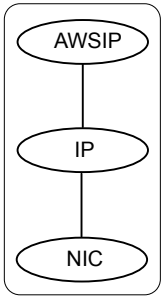


Figure B-3 Azure

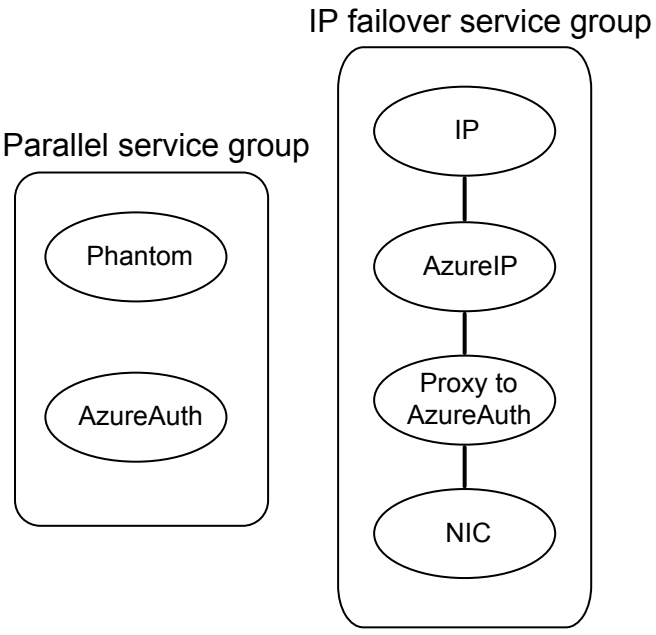
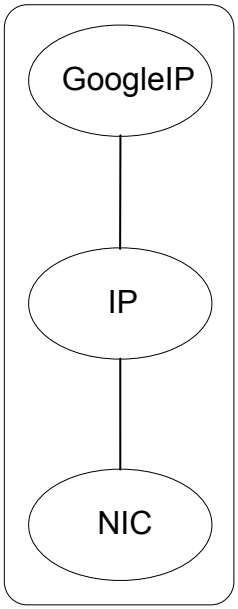


Figure B-4 GCP



Appendix C

Sample SAP HANA TakeOver IP service group for cloud environments

This appendix includes the following topics:

Sample SAP HANA TakeOver IP service group for cloud environments

AWS

Sample configuration of resources for IP service group in AWS:

```
AWSIP AWSIP_HDB (
    NIC = eth0
    OverlayIP = "10.209.76.15"
    AWSBinDir = "/usr/local/bin"
    RouteTableIds = { rtb-fb97ac9d, rtb-f416eb8d, rtb-e48be49d }

)
IP HDB_IP (
    Device = eth0
    Address = "10.209.76.15"
    NetMask = "255.255.252.0"
)
NIC HDB_NIC (
    AWSIP_HDB requires HDB_IP
```

```
    HDB_IP requires HDB_NIC


)


AWSIP_HDB requires HDB_IP
HDB_IP requires HDB_NIC
```

Sample configuration of resources for IP service group in AWS in case of network load balancer used

```
IP HDB_IP (
    Device = eth0
    Address = "10.209.76.15"
    NetMask = "255.255.252.0"
    )

NIC HDB_NIC (
    Device = eth0
    )


Process HDBsocat (
    PathName = "/usr/bin/socat"
    Arguments = "-U TCP-LISTEN:3600,backlog=10,fork,reuseaddr /dev/null"
    RestartLimit = 1
    )


  HDB_IP requires HDB_NIC
  HDBsocat requires HDB_IP
```

Microsoft Azure

To configure SAPHB in Microsoft Azure, you must configure an AzureAuth service group and an IP service group in parallel.

Sample configuration of resources for parallel AzureAuth service group:

```
AzureAuth AZURE_AUTH (
    SubscriptionId = 2dfgg136-fgh6-40dd-b616-c1e9abdf1d63
    ClientId = 123456-d10a-4704-8986-beb86739104d
    SecretKey = fntPgnUnhTprQrqrnRonSlhPhrQpiNtrItpRhnGrrNklFngLs
    TenantId = 12345-0528-4308-brf03-6667d61dd0e3
```

```
)

Phantom PHANTOM (
)
```

Sample configuration of resources for failover IP service group:

```
AzureIP HDB_AZUREIP (
    NICDevice = eth0
    OverlayIP = "10.209.76.15"
    RouteTableResourceIds = { "/subscriptions/
       6940a326-abc6-40dd-b616-d3f9bbdf1d63/resourceGroups/azureRG/
       providers/Microsoft.Network/routeTables/azureroute1",
       "/subscriptions/6940a326-abc6-40dd-b616-d3f9bbdf1d63/
       resourceGroups/azureRG/providers/Microsoft.Network/
       routeTables/azureroute2" }
    AzureAuthResName = AZURE_AUTH
)

IP HDB_IP (
    Device = eth0
    Address = "10.209.76.15"
    NetMask = "255.255.252.0"
)

NIC HDB_NIC (
    Device = eth0
)

Proxy AUTH_PROXY (
    TargetResName = AZURE_AUTH
)

HDB_AZUREIP requires AUTH_PROXY
HDB_IP requires HDB_AZUREIP
AUTH_PROXY requires HDB_NIC
```

Sample configuration of resources for IP service group in Azure in case of network load balancer used

```
IP HDB_IP (

    Device = eth0
    Address = "10.209.76.15"
```

```
    NetMask = "255.255.252.0"
    )
NIC HDB_NIC (
    Device = eth0
    )

Process HDBsocat (
    PathName = "/usr/bin/socat"
    Arguments = "-U TCP-LISTEN:3600,backlog=10,fork,reuseaddr /dev/null"
    RestartLimit = 1
    )

    HDB_IP requires HDB_NIC
    HDBsocat requires HDB_IP
```

GCP

Sample configuration of resources for IP service group on GCP:

```
GoogleIP HDB_GoogleIP (
    Device = eth0
    OverlayIP = "10.209.76.15"
)

IP HDB_IP (
    Device = eth0
    Address = "10.209.76.15"
    NetMask = "255.255.252.0"
)

NIC HDB_NIC (
    Device = eth0
)

HDB_GoogleIP requires HDB_IP
HDB_IP requires HDB_NIC
```