SAP HANA High Availability with HANA System Replication and RHEL Cluster on Cisco UCS

Based on SAP HANA 2.0 with Red Hat Enterprise Linux 7.6 and Cisco Unified Computing System 4.0

Contents

Overview

System replication is one of the high-availability mechanisms that SAP HANA offers. With this feature, all data is replicated to a configured secondary SAP HANA system that is on standby until an event occurs that requires switching the active system from the current primary system to the secondary system. This activity is called the takeover process. To enhance data availability, this process should be integrated with a high-availability cluster solution that can control the takeover process and help ensure seamless access to data.

This document describes how to configure SAP HANA system replication in a scale-up environment in a high-availability cluster with supported Red Hat Enterprise Linux (RHEL) releases in deployments with SAP HANA scale-up appliances and SAP HANA Tailored Datacenter Integration (TDI) infrastructure based on Cisco Unified Computing System™ (Cisco UCS®). This configuration includes SAP HANA appliances based on Cisco UCS C-Series Rack Servers and TDI solutions based on Cisco UCS B-Series Blade Servers.

Audience

This document is intended for IT architects, engineers, and administrators who are responsible for configuring and deploying SAP HANA and associated high-availability configurations. It assumes that the reader has knowledge of Cisco UCS, Linux, and SAP HANA and their deployment scenarios and high-availability concepts.

Reference architecture

The solution presented in this document is based on Cisco UCS. This solution applies to SAP HANA systems with persistence configured through the use of internal disks or an external enterprise storage array. The solution provides a general approach to a single-system (scale-up) design with SAP HANA system replication and automated failover.

All Cisco UCS hardware listed in the Certified and Supported SAP HANA Hardware Directory can be used in SAP HANA system replication scenarios.

Document reference

This document is based on Red Hat's knowledgebase article Automated SAP HANA System Replication in Scale-Up in Pacemaker Cluster. It focuses on the configuration steps specific to Cisco UCS.

Document scope

This document does **not** cover the preparation of an RHEL system for SAP HANA installation or the SAP HANA installation procedure. For more information about these topics, refer to SAP Note 2009879 - SAP HANA Guidelines for Red Hat Enterprise Linux (RHEL). For more information about SAP HANA high availability, refer to SAP Note 2407186 - How-To Guides and White Papers for SAP HANA High Availability.

Supported configurations

For information about supported configurations, see Support Policies for RHEL High Availability Clusters - Management of SAP HANA in a Cluster.

Preparing for installation

SAP HANA primary and secondary systems must be installed and configured in accordance with SAP Notes 2009879 and 2292690. Both systems must have the same SAP system ID (SID) and instance number. All software release versions, including the versions of the SAP HANA software on the designated secondary system, must match the versions on the primary system.

Prerequisites

Verify that the SAP HANA systems meet the prerequisites listed in Chapter 4, Planning, in How to Perform SAP HANA System Replication and follows the network recommendations provided in that document. Verify that both systems are independently up and running.

Subscriptions and repositories

In addition to existing RHEL 7 server and RHEL for SAP HANA repositories to which the systems may already be subscribed, verify that the systems also are subscribed to RHEL high-availability repositories. Refer to Red Hat knowledgebase articles at https://access.redhat.com/solutions/45930 and https://access.redhat.com/solutions/2318061 for instructions.

The following output shows an example of the repositories enabled with RHEL for SAP Solutions 7.6:

```
[root@cishana01 ~]# yum repolist
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
repo id                                          repo name
status
rhel-7-server-eus-rpms/x86_64                    Red Hat Enterprise Linux 7 Server - Extended
Update Support (RPMs)                              24,842
rhel-7-server-rpms/x86_64                        Red Hat Enterprise Linux 7 Server (RPMs)
24,539
rhel-ha-for-rhel-7-server-eus-rpms/x86_64        Red Hat Enterprise Linux High Availability
(for RHEL 7 Server) - Extended Update Support (RPMs    635
rhel-ha-for-rhel-7-server-rpms/x86_64            Red Hat Enterprise Linux High Availability
(for RHEL 7 Server) (RPMs)                             632
rhel-sap-hana-for-rhel-7-server-eus-rpms/x86_64 RHEL for SAP HANA (for RHEL 7 Server)
Extended Update Support (RPMs)                         55
rhel-sap-hana-for-rhel-7-server-rpms/x86_64      Red Hat Enterprise Linux for SAP HANA (RHEL
7 Server) (RPMs)                                     53
```

Configuring SAP HANA system replication

The following example shows how to set up system replication between two nodes running SAP HANA.

This example uses the following configuration:

```
SID: CIS
Instance number: 00
Node 1 fully qualified domain name (FQDN): cishana01.ciscolab.local
Node 2 FQDN: cishana02.ciscolab.local
Node 1 HANA site name: DC1
Node 2 HANA site name: DC2
SAP HANA SYSTEM user password: <HANA_SYSTEM_PASSWORD>
```

```
SAP HANA administrative user: cisadm
```

Verify that both systems can resolve the FQDNs of both systems without problems. To verify that FQDNs can be resolved even without the use of Domain Name System (DNS), you can place them in /etc/hosts as in the following example:

```
[root@cishana01 ~]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
## Admin Network
#
192.168.76.201  cishana01m.ciscolab.local cishana01m
192.168.76.202  cishana02m.ciscolab.local cishana02m
#
#Sysrep network
#
192.168.224.201  cishana01.ciscolab.local cishana01
192.168.224.202  cishana02.ciscolab.local cishana02
```

For system replication to work, the SAP HANA **log_mode** variable must be set to **normal**. You can verify this setting as the SAP HANA **system** user by using the following command on both nodes:

```
cisadm@cishana01:/usr/sap/CIS/HDB00> hdbsql -u system -p <HANA_SYSTEM_PASSWORD> -i 00
"select value from "SYS"."M_INIFILE_CONTENTS" where key='log_mode'"
VALUE
"normal"
1 row selected (overall time 204.048 msec; server time 187.837 msec)
```

Configure SAP HANA primary node

SAP HANA system replication will work only after initial backup has been performed. You should back up the systems based on your existing best practices. The example used here shows how to back up systems to a file.

The following command creates an initial backup in the default path /usr/sap/CIS/HDB00/backup/data. In the default multicontainer systems, that is, SAP HANA tenant database systems, SYSTEMDB and all tenant databases must be backed up.

```
cisadm@cishana01:/usr/sap/CIS/HDB00>  hdbsql -i 00 -u system -p <HANA_SYSTEM_PASSWORD> -d
SYSTEMDB "BACKUP DATA USING FILE ('/usr/sap/CIS/HDB00/backup/data/SYSTEMDB/')"
0 rows affected (overall time 4815.587 msec; server time 4814.583 msec).

cisadm@cishana01:/usr/sap/CIS/HDB00>  hdbsql -i 00 -u system -p <HANA_SYSTEM_PASSWORD> -d
CIS "BACKUP DATA USING FILE ('/usr/sap/CIS/HDB00/backup/data/DB_CIS/')"
0 rows affected (overall time 4520.414 msec; server time 4518.852 msec)
```

After the initial backup, initialize the replication using the following command:

```
cisadm@cishana01:/usr/sap/CIS/HDB00> hdbnsutil -sr_enable --name=DC1
nameserver is active, proceeding ...
successfully enabled system as system replication source site done.
```

Verify that initialization is showing the current node as primary and that SAP HANA is running on it.

```
cisadm@cishana01:/usr/sap/CIS/HDB00> hdbnsutil -sr_state

System Replication State
~~~~~~~~~~~~~~~~~~~~~~~~

online: true

mode: primary
operation mode: primary
site id: 1
site name: DC1


is source system: true
is secondary/consumer system: false
has secondaries/consumers attached: false
is a takeover active: false

Host Mappings:
~~~~~~~~~~~~~~


Site Mappings:
~~~~~~~~~~~~~~
DC1 (primary/)

Tier of DC1: 1

Replication mode of DC1: primary

Operation mode of DC1:

done.
```

Configure SAP HANA secondary node

Verify that you can ping the primary node on the system replication network at the configured IP address. Also verify that the network ports are available. The port **<instance number>+1**, used for system replication communication, must be free and open on the firewall to allow seamless communication between the systems.

You need to register the secondary node to the now-running primary node. Before proceeding, shut down SAP HANA on the secondary node using the following command:

```
cisadm@cishana02:/usr/sap/CIS/HDB00> HDB stop
```

Copy the SAP HANA system PKI SSFS_RH2.KEY and SSFS_RH2.DAT files from the primary node to the secondary node:

```
cisadm@cishana02:/usr/sap/CIS/HDB00> scp
root@cishana01:/usr/sap/CIS/SYS/global/security/rsecssfs/key/SSFS_CIS.KEY
/usr/sap/CIS/SYS/global/security/rsecssfs/key/SSFS_CIS.KEY
The authenticity of host 'cishana01 (192.168.224.201)' can't be established.
ECDSA key fingerprint is SHA256:s6jDynevzrOWTTa2K8PNOPxbu3OUTwjO0C0s1Umggew.
ECDSA key fingerprint is MD5:04:20:0f:33:60:b7:86:00:64:56:9b:f4:22:b6:ed:56.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'cishana01,192.168.224.201' (ECDSA) to the list of known hosts.
root@cishana01's password:
SSFS_CIS.KEY                                                  100%  187
109.2KB/s   00:00

cisadm@cishana02:/usr/sap/CIS/HDB00> scp
root@cishana01:/usr/sap/CIS/SYS/global/security/rsecssfs/data/SSFS_CIS.DAT
/usr/sap/CIS/SYS/global/security/rsecssfs/data/SSFS_CIS.DAT
root@cishana01's password:
SSFS_CIS.DAT                                                  100% 2960
2.3MB/s   00:00
```

To register the secondary node, use the following command:

```
cisadm@cishana02:/usr/sap/CIS/HDB00> hdbnsutil -sr_register --
remoteHost=cishana01.ciscolab.local --remoteInstance=00 --replicationMode=syncmem –
operationMode=logreplay --name=DC2
adding site ...
--operationMode not set; using default from global.ini/[system_replication]/operation_mode:
logreplay
nameserver cishana02.ciscolab.local:30001 not responding.
collecting information ...
updating local ini files ...
done.
```

Note: Verify that the correct primary host name is used when you register the secondary system. Refer to SAP Note 2510253 for more information.

Note: Choose the replication mode and operating mode based on your use case. Refer to Chapters 3.2.1 and 3.2.2 of How to Perform SAP HANA System Replication.

Start SAP HANA on the secondary node:

```
cisadm@cishana02:/usr/sap/CIS/HDB00> HDB start
```

Verify that the secondary node is running and that the mode is **syncmem**. The output should look similar to the output shown here:

```
cisadm@cishana02:/usr/sap/CIS/HDB00> hdbnsutil -sr_state

System Replication State
~~~~~~~~~~~~~~~~~~~~~~~~

online: true

mode: syncmem
operation mode: logreplay
site id: 2
site name: DC2


is source system: false
is secondary/consumer system: true
has secondaries/consumers attached: false
is a takeover active: false
active primary site: 1

primary masters: cishana01.ciscolab.local

Host Mappings:
~~~~~~~~~~~~~~

cishana02.ciscolab.local -> [DC2] cishana02.ciscolab.local
cishana02.ciscolab.local -> [DC1] cishana01.ciscolab.local


Site Mappings:
~~~~~~~~~~~~~~
DC1 (primary/primary)
    |---DC2 (syncmem/logreplay)

Tier of DC1: 1
Tier of DC2: 2

Replication mode of DC1: primary
Replication mode of DC2: syncmem

Operation mode of DC1: primary
Operation mode of DC2: logreplay

Mapping: DC1 -> DC2
done.
```

To verify the current state of SAP HANA system replication, run the following command as the SAP HANA administrative user on the current primary SAP HANA node:

```
cisadm@cishana01:/usr/sap/CIS/HDB00/exe/python_support> python systemReplicationStatus.py
| Database | Host                    | Port  | Service Name | Volume ID | Site ID | Site
Name | Secondary               | Secondary | Secondary | Secondary | Secondary     |
Replication | Replication | Replication    |
```

```
|          |                          |       |             |           |         |
| Host                    | Port      | Site ID   | Site Name | Active Status | Mode
| Status      | Status Details |
| -------- | ------------------------ | ----- | ---------- | --------- | ------- | -------
-- | ------------------------ | --------- | --------- | --------- | ------------- | --------
--- | ----------- | -------------- |
| SYSTEMDB | cishana01.ciscolab.local | 30001 | nameserver   |         1 |       1 | DC1
| cishana02.ciscolab.local |     30001 |         2 | DC2       | YES           | SYNCMEM
| ACTIVE      |                |
| CIS      | cishana01.ciscolab.local | 30007 | xsengine     |         2 |       1 | DC1
| cishana02.ciscolab.local |     30007 |         2 | DC2       | YES           | SYNCMEM
| ACTIVE      |                |
| CIS      | cishana01.ciscolab.local | 30003 | indexserver  |         3 |       1 | DC1
| cishana02.ciscolab.local |     30003 |         2 | DC2       | YES           | SYNCMEM
| ACTIVE      |                |

status system replication site "2": ACTIVE
overall system replication status: ACTIVE

Local System Replication State
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

mode: PRIMARY
site id: 1
site name: DC1
```

Configuring SAP HANA in a Pacemaker cluster

Now install the configuration tools required to set up the cluster and then create a fencing configuration.

Install Pacemaker configuration tools

Install the RHEL High Availability Add-On software packages along with the fence agent that you require by using the following command on both the primary and secondary nodes:

```
[root@cishana01 ~]# yum install pcs pacemaker fence-agents-cisco-ucs fence-agents-ipmilan
fence-agents-redfish fence-agents-sbd
```

```
Installed:
  fence-agents-cisco-ucs.x86_64 0:4.2.1-11.el7_6.8              fence-agents-ipmilan.x86_64 0:4.2.1-11.el7_6.8
  fence-agents-redfish.x86_64 0:4.2.1-11.el7_6.8                fence-agents-sbd.x86_64 0:4.2.1-11.el7_6.8
  pacemaker.x86_64 0:1.1.19-8.el7_6.5                           pcs.x86_64 0:0.9.165-6.el7_6.2

Dependency Installed:
  OpenIPMI-modalias.x86_64 0:2.0.23-2.el7_6.1                     audit-libs-python.x86_64 0:2.8.4-4.el7
  checkpolicy.x86_64 0:2.5-8.el7                                  cifs-utils.x86_64 0:6.2-10.el7
  clufter-bin.x86_64 0:0.77.1-1.el7                               clufter-common.noarch 0:0.77.1-1.el7
  corosync.x86_64 0:2.4.3-4.el7                                   corosynclib.x86_64 0:2.4.3-4.el7
  fence-agents-common.x86_64 0:4.2.1-11.el7_6.8                   ipmitool.x86_64 0:1.8.18-7.el7
  libcgroup.x86_64 0:0.41-20.el7                                  libldb.x86_64 0:1.3.4-1.el7
  libqb.x86_64 0:1.0.1-7.el7                                      libsemanage-python.x86_64 0:2.5-14.el7
  libtalloc.x86_64 0:2.1.13-1.el7                                 libtevent.x86_64 0:0.9.36-1.el7
  libwbclient.x86_64 0:4.8.3-6.el7_6                              net-snmp-libs.x86_64 1:5.7.2-38.el7_6.2
  overpass-fonts.noarch 0:2.1-1.el7                               pacemaker-cli.x86_64 0:1.1.19-8.el7_6.5
  pacemaker-cluster-libs.x86_64 0:1.1.19-8.el7_6.5                pacemaker-libs.x86_64 0:1.1.19-8.el7_6.5
  perl-TimeDate.noarch 1:2.30-2.el7                               pexpect.noarch 0:2.3-11.el7
  policycoreutils-python.x86_64 0:2.5-29.el7_6.1                  psmisc.x86_64 0:22.20-15.el7
  python-IPy.noarch 0:0.75-6.el7                                  python-clufter.noarch 0:0.77.1-1.el7
  python-requests.noarch 0:2.6.0-5.el7_6                          python-urllib3.noarch 0:1.10.2-5.el7
  resource-agents.x86_64 0:4.1.1-12.el7_6.23                      ruby.x86_64 0:2.0.0.648-35.el7_6
  ruby-irb.noarch 0:2.0.0.648-35.el7_6                            ruby-libs.x86_64 0:2.0.0.648-35.el7_6
  rubygem-bigdecimal.x86_64 0:1.2.0-35.el7_6                      rubygem-io-console.x86_64 0:0.4.2-35.el7_6
  rubygem-json.x86_64 0:1.7.7-35.el7_6                            rubygem-psych.x86_64 0:2.0.0-35.el7_6
  rubygem-rdoc.noarch 0:4.0.0-35.el7_6                            rubygems.noarch 0:2.0.14.1-35.el7_6
  samba-client-libs.x86_64 0:4.8.3-6.el7_6                        samba-common.noarch 0:4.8.3-6.el7_6
  samba-common-libs.x86_64 0:4.8.3-6.el7_6                        setools-libs.x86_64 0:3.3.8-4.el7

Dependency Updated:
  policycoreutils.x86_64 0:2.5-29.el7_6.1

Complete!
```

Create the cluster

First start the pcsd daemon.

The following commands start the pcsd service and enable pcsd at system start. Run these commands on each node in the cluster:

```
[root@cishana01 ~]# systemctl start pcsd.service
[root@cishana01 ~]# systemctl enable pcsd.service
Created symlink from /etc/systemd/system/multi-user.target.wants/pcsd.service to
/usr/lib/systemd/system/pcsd.service.
```

Now authenticate the cluster nodes.

The Pacemaker-libs package installed as a dependency in the previous step creates the **hacluster** user and group. The password for user hacluster should be the same on each node:

```
[root@cishana01 ~]# passwd hacluster
Changing password for user hacluster.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

The following command authenticates pcs to the pcs daemon on the nodes in the cluster.

On one node of the cluster, authenticate the nodes that will constitute the cluster with the following command. After you run this command, you will be prompted for a username and a password. Specify **hacluster** as the username and the key in the password set earlier.

Use the FQDN of the nodes associated with defined system replication network.

```
[root@cishana01 ~]# pcs cluster auth cishana01.ciscolab.local cishana02.ciscolab.local
Username: hacluster
Password:
cishana01.ciscolab.local: Authorized
cishana02.ciscolab.local: Authorized
```

Run the following command as the root user on either node to configure the cluster infrastructure and start the cluster. In the example here, **sap_hana** is chosen as the cluster name.

```
[root@cishana01 ~]# pcs cluster setup --name sap_hana cishana01.ciscolab.local
cishana02.ciscolab.local
Destroying cluster on nodes: cishana01.ciscolab.local, cishana02.ciscolab.local...
cishana01.ciscolab.local: Stopping Cluster (pacemaker)...
cishana02.ciscolab.local: Stopping Cluster (pacemaker)...
cishana02.ciscolab.local: Successfully destroyed cluster
cishana01.ciscolab.local: Successfully destroyed cluster

Sending 'pacemaker_remote authkey' to 'cishana01.ciscolab.local', 'cishana02.ciscolab.local'
cishana01.ciscolab.local: successful distribution of the file 'pacemaker_remote authkey'
cishana02.ciscolab.local: successful distribution of the file 'pacemaker_remote authkey'
Sending cluster config files to the nodes...
cishana01.ciscolab.local: Succeeded
cishana02.ciscolab.local: Succeeded

Synchronizing pcsd certificates on nodes cishana01.ciscolab.local,
cishana02.ciscolab.local...
cishana01.ciscolab.local: Success
cishana02.ciscolab.local: Success
Restarting pcsd on the nodes in order to reload the certificates...
cishana01.ciscolab.local: Success
cishana02.ciscolab.local: Success
```

Start and enable cluster services on both nodes:

```
[root@cishana01 ~]# pcs cluster start --all
cishana01.ciscolab.local: Starting Cluster (corosync)...
cishana02.ciscolab.local: Starting Cluster (corosync)...
cishana01.ciscolab.local: Starting Cluster (pacemaker)...
cishana02.ciscolab.local: Starting Cluster (pacemaker)...

[root@cishana01 ~]# systemctl enable pacemaker
Created symlink from /etc/systemd/system/multi-user.target.wants/pacemaker.service to
/usr/lib/systemd/system/pacemaker.service.
[root@cishana01 ~]# systemctl enable corosync
Created symlink from /etc/systemd/system/multi-user.target.wants/corosync.service to
/usr/lib/systemd/system/corosync.service.

[root@cishana02 ~]# systemctl enable pacemaker
Created symlink from /etc/systemd/system/multi-user.target.wants/pacemaker.service to
/usr/lib/systemd/system/pacemaker.service.
```

```
[root@cishana02 ~]# systemctl enable corosync
Created symlink from /etc/systemd/system/multi-user.target.wants/corosync.service to
/usr/lib/systemd/system/corosync.service.
```

A check on the cluster state shows that STONITH devices need to be configured next:

```
[root@cishana01 ~]# pcs status
Cluster name: sap_hana

WARNINGS:
No stonith devices and stonith-enabled is not false

Stack: corosync
Current DC: cishana02.ciscolab.local (version 1.1.19-8.el7_6.5-c3c624ea3d) - partition with
quorum
Last updated: Tue Jan 21 04:59:36 2020
Last change: Mon Jan 20 16:46:25 2020 by hacluster via crmd on cishana01.ciscolab.local

2 nodes configured
0 resources configured

Online: [ cishana01.ciscolab.local cishana02.ciscolab.local ]

No resources


Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
```

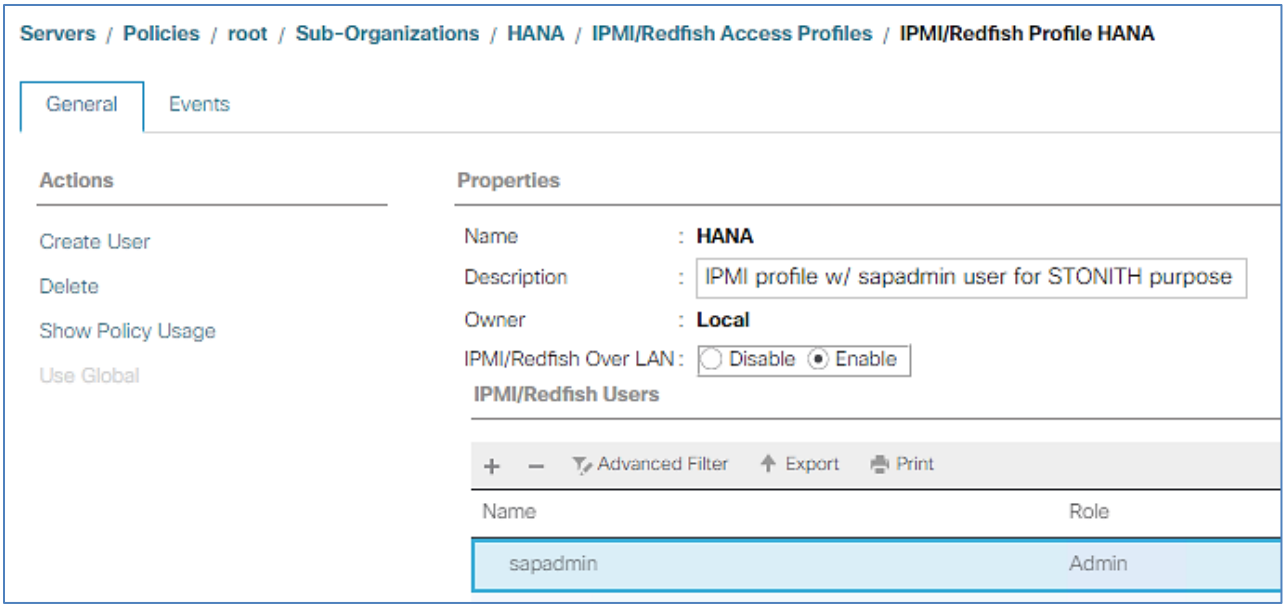
Prepare to fence

STONITH is an acronym for " shoot the other node in the head." STONITH protects data from being corrupted by rogue nodes or concurrent access. It also plays a role in the event that a clustered service cannot be stopped. In this case, the cluster uses STONITH to force the whole node offline, thereby making it safe to start the service elsewhere. Intelligent Platform Management Interface (IPMI) runs on the baseboard management controller (BMC) of the server.

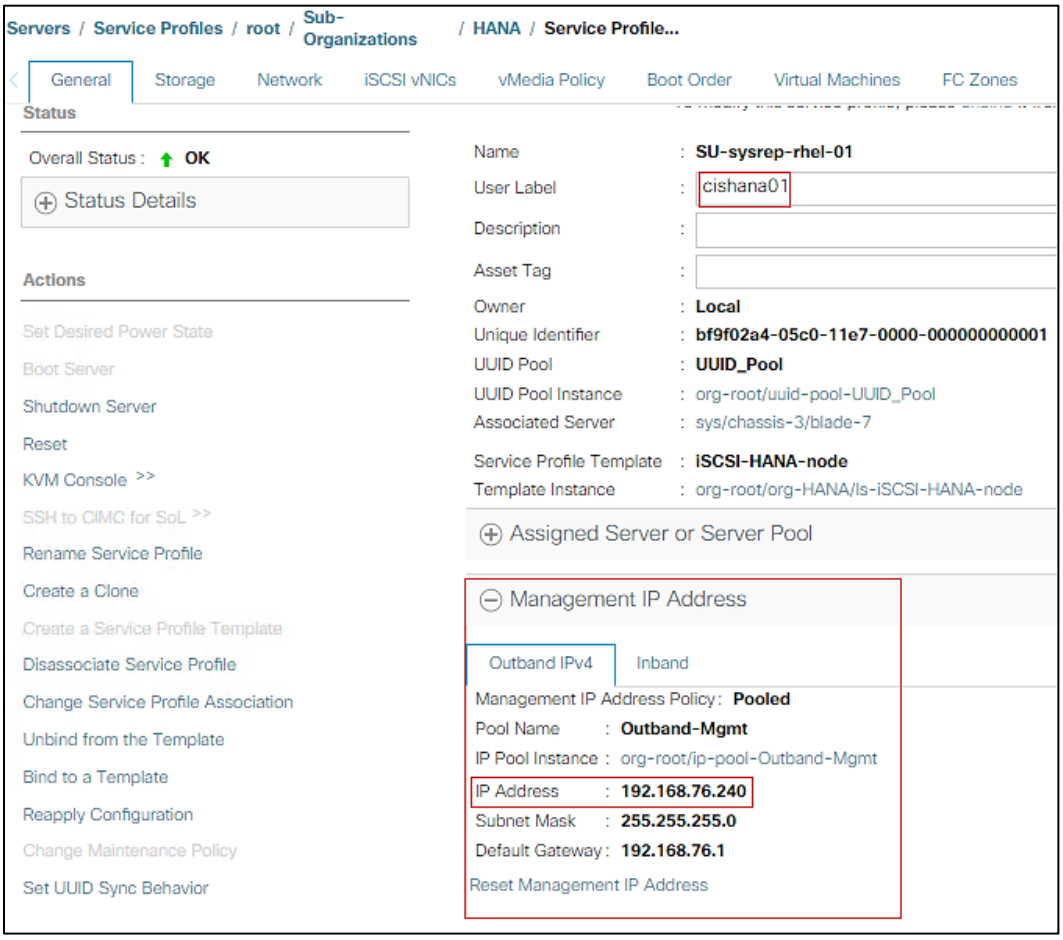
This section discusses two cases: one using Cisco UCS B- and C-Series servers managed by Cisco UCS, and one using standalone Cisco UCS C-Series servers not managed by Cisco UCS.

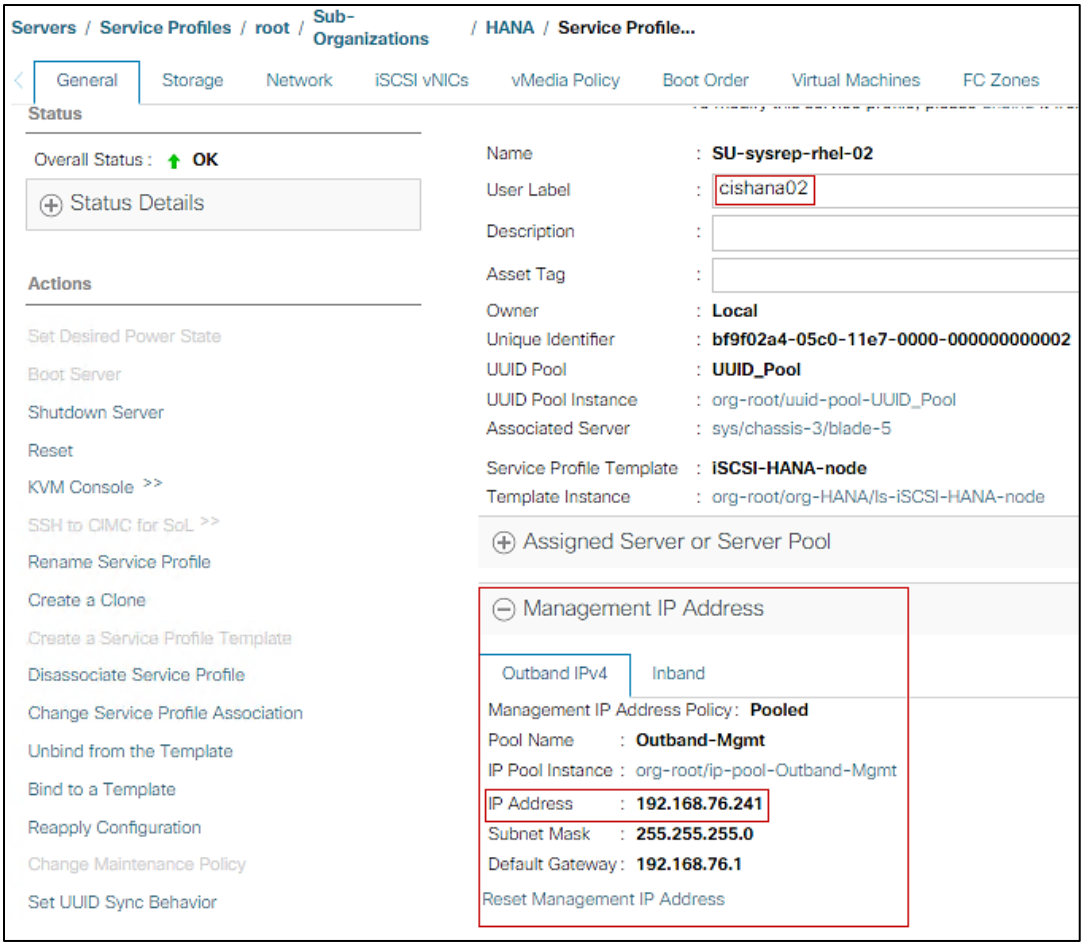
Case 1: Cisco UCS B- and C-Series servers managed by Cisco UCS

With Cisco UCS B- and C-Series servers managed by Cisco UCS, you use IPMI access profiles for access over the out-of-band management IP network configured for the node. An example of an IPMI Redfish profile that SAP HANA created with admin role user **sapadmin** is shown here:



Determine the IPMI address from the Cisco UCS service profile of the nodes:





Add those entries to the /etc/hosts file and synchronize the host file to the other node:

```
[root@cishana01 ~]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
## Admin Network
#
192.168.76.201  cishana01m.ciscolab.local cishana01m
192.168.76.202  cishana02m.ciscolab.local cishana02m
#
#Sysrep network
#
192.168.224.201  cishana01.ciscolab.local cishana01
192.168.224.202  cishana02.ciscolab.local cishana02
#
#IPMI network
#
192.168.76.240  cishana01-ipmi
192.168.76.241  cishana02-ipmi
#
```

Test the IPMI connectivity on both nodes.

Verify that the out-of-band network IP addresses are accessible from the admin network defined for the nodes with the appropriate routing configured in the host. In the current example, both the admin network of the nodes and the out-of-band network are on the same Layer 3 network.

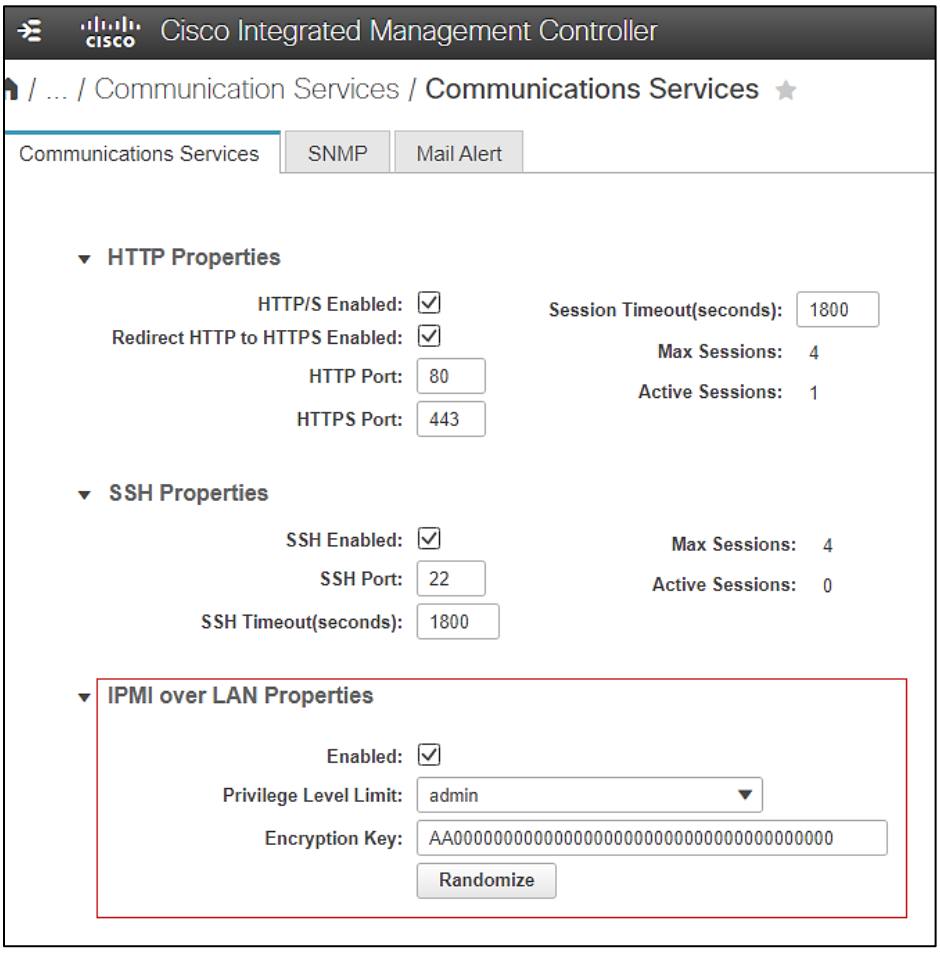
With IPMI user sapadmin created for the IMPI Redfish profile, verify IPMI connectivity on both nodes:

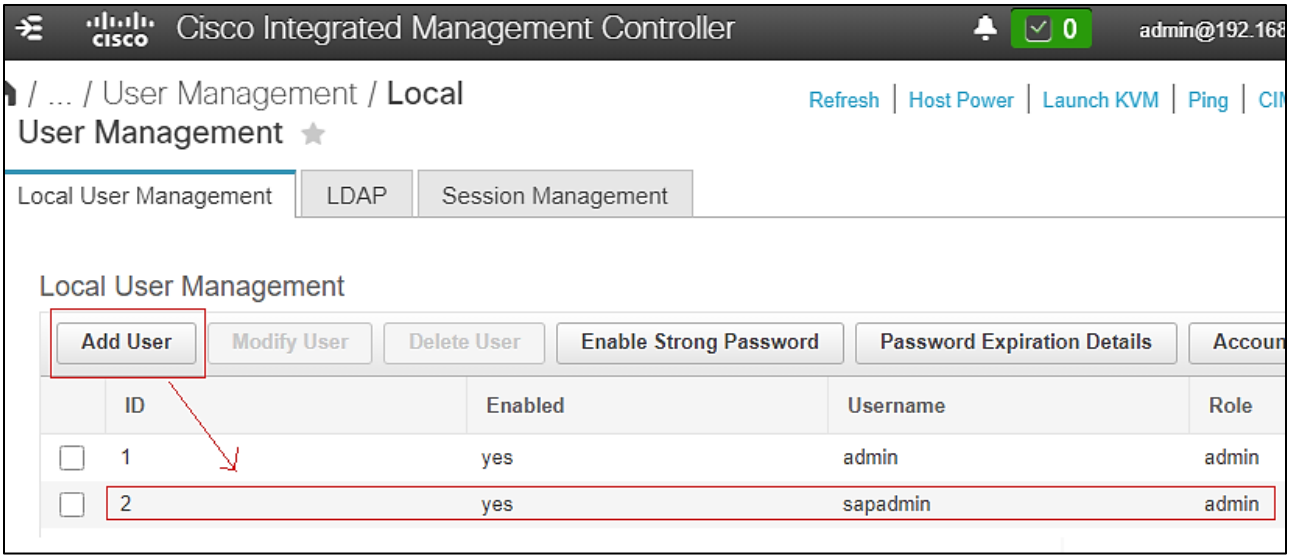
```
[root@cishana01 ~]# ipmitool -H cishana01-ipmi -U sapadmin -P <IPMI_USER_PASSWORD> -I
lanplus power status
Chassis Power is on

[root@cishana01 ~]# ipmitool -H cishana02-ipmi -U sapadmin -P <IPMI_USER_PASSWORD> -I
lanplus power status
Chassis Power is on
```

Case 2: Standalone Cisco UCS C-Series servers not managed by Cisco UCS

With Cisco UCS C-Series standalone servers, configure IMPI over the LAN with admin privileges to manage the Cisco® Integrated Management Controller (IMC) using IPMI commands. Note the IMC IP addresses of the nodes for IPMI access and append these to the /etc/hosts file of the nodes.

Also create a local management user with admin privilege for IPMI access.





Use the following commands to test IPMI access with Cisco UCS C-Series standalone servers with IMPI over LAN enabled:

```
[root@cishana01 ~]# ipmitool -H cishana01-ipmi -I lanplus -U sapadmin -P
<IPMI_USER_PASSWORD> -y AA000000000000000000000000000000000000000000 chassis power status
Chassis Power is on


[root@cishana01 ~]# ipmitool -H cishana02-ipmi -I lanplus -U sapadmin -P
<IPMI_USER_PASSWORD> -y AA000000000000000000000000000000000000000000 chassis power status
Chassis Power is on
```

Note: cishana01-ipmi and cishana02-ipmi are the IMC IP addresses of cishana01 and cishana02 respectively. The sapadmin user in these commands is the admin-privileged IMC login user.

Create fence devices

Now create fence devices.

Case 1: Cisco UCS B- and C-Series servers managed by Cisco UCS

To configure STONITH devices in the cluster in the case of Cisco UCS managed servers, run the commands shown here:

```
[root@cishana01 ~]# pcs stonith create st_ipmi_node1 fence_ipmilan ipaddr="cishana01-ipmi"
login="sapadmin" passwd="<IPMI_USER_PASSWORD>" pcmk_host_list="cishana01.ciscolab.local"
lanplus=1

[root@cishana01 ~]# pcs stonith create st_ipmi_node2 fence_ipmilan ipaddr="cishana02-ipmi"
login="sapadmin" passwd="<IPMI_USER_PASSWORD>" pcmk_host_list="cishana02.ciscolab.local"
lanplus=1
```

Case 2: Standalone Cisco UCS C-Series servers not managed by Cisco UCS

In the case of Cisco UCS C-Series standalone nodes, use the following command format to create the STONITH devices:

```
[root@cishana01 ~]# pcs stonith create st_ipmi_node1 fence_ipmilan ipaddr="cishana01-ipmi"
login="sapadmin" passwd="<IPMI_USER_PASSWORD>"
hexadecimal_kg="AA00000000000000000000000000000000000000"
pcmk_host_list="cishana01.ciscolab.local" lanplus=1

[root@cishana01 ~]# pcs stonith create st_ipmi_node2 fence_ipmilan ipaddr="cishana02-ipmi"
login="sapadmin" passwd="<IPMI_USER_PASSWORD>"
hexadecimal_kg="AA00000000000000000000000000000000000000"
pcmk_host_list="cishana02.ciscolab.local" lanplus=1
```

Note: cishana01-ipmi and cishana02-ipmi are IMC IP addresses of cishana01 and cishana02 respectively. The sapadmin user in these commands is the admin-privileged IMC login user.

With standalone Cisco UCS C-Series Rack Server Software Release 4.0, you must include the hexadecimal key during the configuration.

A check on the cluster state shows that STONITH devices that are now part of cluster:

```
[root@cishana01 ~]# pcs status
Cluster name: sap_hana
Stack: corosync
Current DC: cishana02.ciscolab.local (version 1.1.19-8.el7_6.5-c3c624ea3d) - partition with
quorum
Last updated: Tue Jan 21 10:36:15 2020
Last change: Tue Jan 21 10:31:52 2020 by root via cibadmin on cishana01.ciscolab.local

2 nodes configured
2 resources configured

Online: [ cishana01.ciscolab.local cishana02.ciscolab.local ]

Full list of resources:

 st_ipmi_node1  (stonith:fence_ipmilan):        Started cishana01.ciscolab.local
 st_ipmi_node2  (stonith:fence_ipmilan):        Started cishana02.ciscolab.local

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
```

Install SAP HANA resource agents

The SAP HANA resource agents interface with Pacemaker to allow SAP instances to be managed in a cluster environment.

Install the resource agents package on both nodes as the root user. This process makes the resource agents SAPHana and SAPHanaTopology available for the landscape. While one gathers the topology

information of the configured landscape, the other manages instances configured in SAP HANA system replication.

```
[root@cishana01 ~]# yum install resource-agents-sap-hana

[root@cishana02 ~]# yum install resource-agents-sap-hana
```

```
Downloading packages:
resource-agents-sap-hana-4.1.1-12.el7_6.23.x86_64.rpm                    |  77 kB  00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : resource-agents-sap-hana-4.1.1-12.el7_6.23.x86_64                    1/1
  Verifying  : resource-agents-sap-hana-4.1.1-12.el7_6.23.x86_64                    1/1

Installed:
  resource-agents-sap-hana.x86_64 0:4.1.1-12.el7_6.23

Complete!
```

SAPHanaTopology resource

The SAPHanaTopology resource gathers status and configuration information about SAP HANA system replication on each node. In addition, it starts and monitors the local SAP Host Agent, which is required for starting, stopping, and monitoring the SAP HANA instances.

Create a clone of the SAPHanaTopology resource specific to the implementation, supplying the correct SID and instance number attributes as shown here on either node. In the example here, the name **SAPHanaTopology_CIS_00** is used for the clone resource created.

```
[root@cishana01 ~]# pcs resource create SAPHanaTopology_CIS_00 SAPHanaTopology SID=CIS
InstanceNumber=00 \
> op start timeout=600 \
> op stop timeout=300 \
> op monitor interval=10 timeout=600 \
> --clone clone-max=2 clone-node-max=1 interleave=true

Assumed agent name 'ocf:heartbeat:SAPHanaTopology' (deduced from 'SAPHanaTopology')
```

Note: The optional timeouts shown in the preceding example for the resource operations are examples only and may need to be adjusted depending on the actual SAP HANA setup (for example, large SAP HANA databases can take longer to start, and therefore the start timeout may have to be increased).

The command **crm_mon -A1** displays the collected information about the system replication status in the form of node attributes.

```
[root@cishana01 ~]# crm_mon -A1
Stack: corosync
Current DC: cishana02.ciscolab.local (version 1.1.19-8.el7_6.5-c3c624ea3d) - partition with
quorum
Last updated: Fri Jan 24 03:32:53 2020
Last change: Fri Jan 24 03:31:53 2020 by root via crm_attribute on cishana02.ciscolab.local

2 nodes configured
4 resources configured
```

```
Online: [ cishana01.ciscolab.local cishana02.ciscolab.local ]

Active resources:

 st_ipmi_node1  (stonith:fence_ipmilan):        Started cishana01.ciscolab.local
 st_ipmi_node2  (stonith:fence_ipmilan):        Started cishana02.ciscolab.local
 Clone Set: SAPHanaTopology_CIS_00-clone [SAPHanaTopology_CIS_00]
     Started: [ cishana01.ciscolab.local cishana02.ciscolab.local ]

Node Attributes:
* Node cishana01.ciscolab.local:
    + hana_cis_remoteHost              : cishana02.ciscolab.local
    + hana_cis_roles                   : 4:P:master1:master:worker:master
    + hana_cis_site                    : DC1
    + hana_cis_srmode                  : syncmem
    + hana_cis_version                 : 2.00.033.00.1535711040
    + hana_cis_vhost                   : cishana01.ciscolab.local
* Node cishana02.ciscolab.local:
    + hana_cis_remoteHost              : cishana01.ciscolab.local
    + hana_cis_roles                   : 4:S:master1:master:worker:master
    + hana_cis_site                    : DC2
    + hana_cis_srmode                  : syncmem
    + hana_cis_version                 : 2.00.033.00.1535711040
    + hana_cis_vhost                   : cishana02.ciscolab.local
```

SAPHana resource

The SAPHana resource in a primary-secondary configuration manages instances in primary and secondary nodes.

Create a clone of the SAPHana resource specific to the implementation, supplying the correct SID and instance number attributes as shown here on either node. In the example here, the name **SAPHana_CIS_00** is used for the clone resource created.

```
[root@cishana01 ~]# pcs resource create SAPHana_CIS_00 SAPHana SID=CIS InstanceNumber=00 \
> PREFER_SITE_TAKEOVER=true DUPLICATE_PRIMARY_TIMEOUT=7200 AUTOMATED_REGISTER=true \
> op start timeout=3600 \
> op stop timeout=3600 \
> op monitor interval=61 role="Slave" timeout=700 \
> op monitor interval=59 role="Master" timeout=700 \
> op promote timeout=3600 \
> op demote timeout=3600 \
> master notify=true clone-max=2 clone-node-max=1 interleave=true

Assumed agent name 'ocf:heartbeat:SAPHana' (deduced from 'SAPHana')
```

Note: The following optional parameters are of importance:

- AUTOMATED_REGISTER: When a takeover event occurs, should the former primary instance be registered as secondary? If you specify **false**, manual intervention will be needed. If you specify **true**, the former primary instance will be registered by the resource agent as the secondary instance.
- PREFER_SITE_TAKEOVER: Should the resource agent prefer to switch over to the secondary instance instead of restarting the primary instance locally? If you specify **true**, you prefer the

takeover to switch to the secondary site. If you specify **false**, you prefer to restart locally. If you specify **never**, under no circumstances should a takeover switch to the other node.

After the resource is started, it will add node attributes describing the current state of SAP HANA databases on the nodes as shown in the following example:

```
[root@cishana01 ~]# crm_mon -A1
Stack: corosync
Current DC: cishana02.ciscolab.local (version 1.1.19-8.el7_6.5-c3c624ea3d) - partition with
quorum
Last updated: Fri Jan 24 04:23:18 2020
Last change: Fri Jan 24 04:23:10 2020 by root via crm_attribute on cishana01.ciscolab.local

2 nodes configured
6 resources configured

Online: [ cishana01.ciscolab.local cishana02.ciscolab.local ]

Active resources:

 st_ipmi_node1  (stonith:fence_ipmilan):        Started cishana01.ciscolab.local
 st_ipmi_node2  (stonith:fence_ipmilan):        Started cishana02.ciscolab.local
 Clone Set: SAPHanaTopology_CIS_00-clone [SAPHanaTopology_CIS_00]
     Started: [ cishana01.ciscolab.local cishana02.ciscolab.local ]
 Master/Slave Set: SAPHana_CIS_00-master [SAPHana_CIS_00]
     Masters: [ cishana01.ciscolab.local ]
     Slaves: [ cishana02.ciscolab.local ]

Node Attributes:
* Node cishana01.ciscolab.local:
    + hana_cis_clone_state              : PROMOTED
    + hana_cis_op_mode                  : logreplay
    + hana_cis_remoteHost               : cishana02.ciscolab.local
    + hana_cis_roles                    : 4:P:master1:master:worker:master
    + hana_cis_site                     : DC1
    + hana_cis_srmode                   : syncmem
    + hana_cis_sync_state               : PRIM
    + hana_cis_version                  : 2.00.033.00.1535711040
    + hana_cis_vhost                    : cishana01.ciscolab.local
    + lpa_cis_lpt                       : 1579868590
    + master-SAPHana_CIS_00             : 150
* Node cishana02.ciscolab.local:
    + hana_cis_clone_state              : PROMOTED
    + hana_cis_op_mode                  : logreplay
    + hana_cis_remoteHost               : cishana01.ciscolab.local
    + hana_cis_roles                    : 4:S:master1:master:worker:master
    + hana_cis_site                     : DC2
    + hana_cis_srmode                   : syncmem
```

```
    + hana_cis_sync_state               : SOK
    + hana_cis_version                  : 2.00.033.00.1535711040
    + hana_cis_vhost                    : cishana02.ciscolab.local
    + lpa_cis_lpt                       : 30
    + master-SAPHana_CIS_00             : 100
```

Create virtual IP address resource

The cluster needs a virtual IP address to reach the primary instance of SAP HANA. Create the IPaddr2 resource as shown here:

```
[root@cishana01 ~]# pcs resource create vip_CIS_00 IPaddr2 ip="192.168.224.200"

Assumed agent name 'ocf:heartbeat:IPaddr2' (deduced from 'IPaddr2')
```

A check on the cluster status shows the addition of the cluster IP address resource:

```
[root@cishana01 ~]# pcs status
Cluster name: sap_hana
Stack: corosync
Current DC: cishana02.ciscolab.local (version 1.1.19-8.el7_6.5-c3c624ea3d) - partition with
quorum
Last updated: Thu Jan 30 02:44:06 2020
Last change: Thu Jan 30 02:43:39 2020 by root via crm_attribute on cishana02.ciscolab.local

2 nodes configured
7 resources configured

Online: [ cishana01.ciscolab.local cishana02.ciscolab.local ]

Full list of resources:

 st_ipmi_node1  (stonith:fence_ipmilan):        Started cishana01.ciscolab.local
 st_ipmi_node2  (stonith:fence_ipmilan):        Started cishana02.ciscolab.local
 Clone Set: SAPHanaTopology_CIS_00-clone [SAPHanaTopology_CIS_00]
     Started: [ cishana01.ciscolab.local cishana02.ciscolab.local ]
 Master/Slave Set: SAPHana_CIS_00-master [SAPHana_CIS_00]
     Masters: [ cishana01.ciscolab.local ]
     Slaves: [ cishana02.ciscolab.local ]
 vip_CIS_00     (ocf::heartbeat:IPaddr2):       Started cishana01.ciscolab.local

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
```

Create cluster constraints

For correct operation, you need to verify that the SAPHanaTopology resource is started before you start the SAPHana resource. You also need to verify that the virtual IP address is present on the node on which the primary resource of SAPHana is running. To achieve this, create two constraints:

- Constraint to start SAPHanaTopology before SAPHana resource
- Constraint to co-locate the IPaddr2 resource with the primary of SAPHana resource

Constraint to start SAPHanaTopology before SAPHana resource

Create the constraint that mandates the start order of these resources by using the following command:

```
[root@cishana01 ~]# pcs constraint order SAPHanaTopology_CIS_00-clone then SAPHana_CIS_00-master symmetrical=false

Adding SAPHanaTopology_CIS_00-clone SAPHana_CIS_00-master (kind: Mandatory) (Options: first-action=start then-action=start symmetrical=false)
```

Note: The **symmetrical=false** attribute specifies that you care only about the start of resources, and that they don't need to be stopped in reverse order.

Both resources (SAPHana and SAPHanaTopology) have the attribute **interleave=true**, which allows parallel startup of these resources on nodes. This attribute permits the SAPHana resource to start on any of the nodes as soon as SAPHanaTopology is running there. That is, despite the ordering, the system does not need to wait for all the nodes to start SAPHanaTopology.

Constraint to co-locate the IPaddr2 resource with the primary of SAPHana resource

Create the constraint to co-locate the IPaddr2 resource with the SAPHana resource that gets promoted as the primary:

```
[root@cishana01 ~]# pcs constraint colocation add vip_CIS_00 with master SAPHana_CIS_00-master 2000
```

Note: Here, the constraint is using a score of 2000 instead of the default **INFINITY**. This specification allows the IPaddr2 resource to be taken down by the cluster in the event that no primary is promoted in the SAPHana resource. Therefore, this address can still be used with tools such as SAP Management Console and SAP logical volume management (LVM), which can use this address to query the status information for the SAP instance.

Verify that STONITH is enabled

STONITH must be enabled for fencing to be functional in the cluster.

```
[root@cishana01 ~]# pcs property set stonith-enabled=true
```

A check for the property setting in the cluster information base (**cib**) should show the value set to **true**:

```
[root@cishana01 ~]# pcs cluster cib | grep stonith-enabled
        <nvpair id="cib-bootstrap-options-stonith-enabled" name="stonith-enabled" value="true"/>
```

Cluster behavior during takeover

With cluster configuration in place, the SAP HANA system is protected against accidental shutdown of processes by human error. Note that the starting and stopping of the instances are completely managed by the cluster, and manual intervention for these processes should be avoided unless the cluster is in maintenance or standby mode.

The flowchart in Figure 1 explains the cluster behavior when the SAP HANA system becomes unavailable.

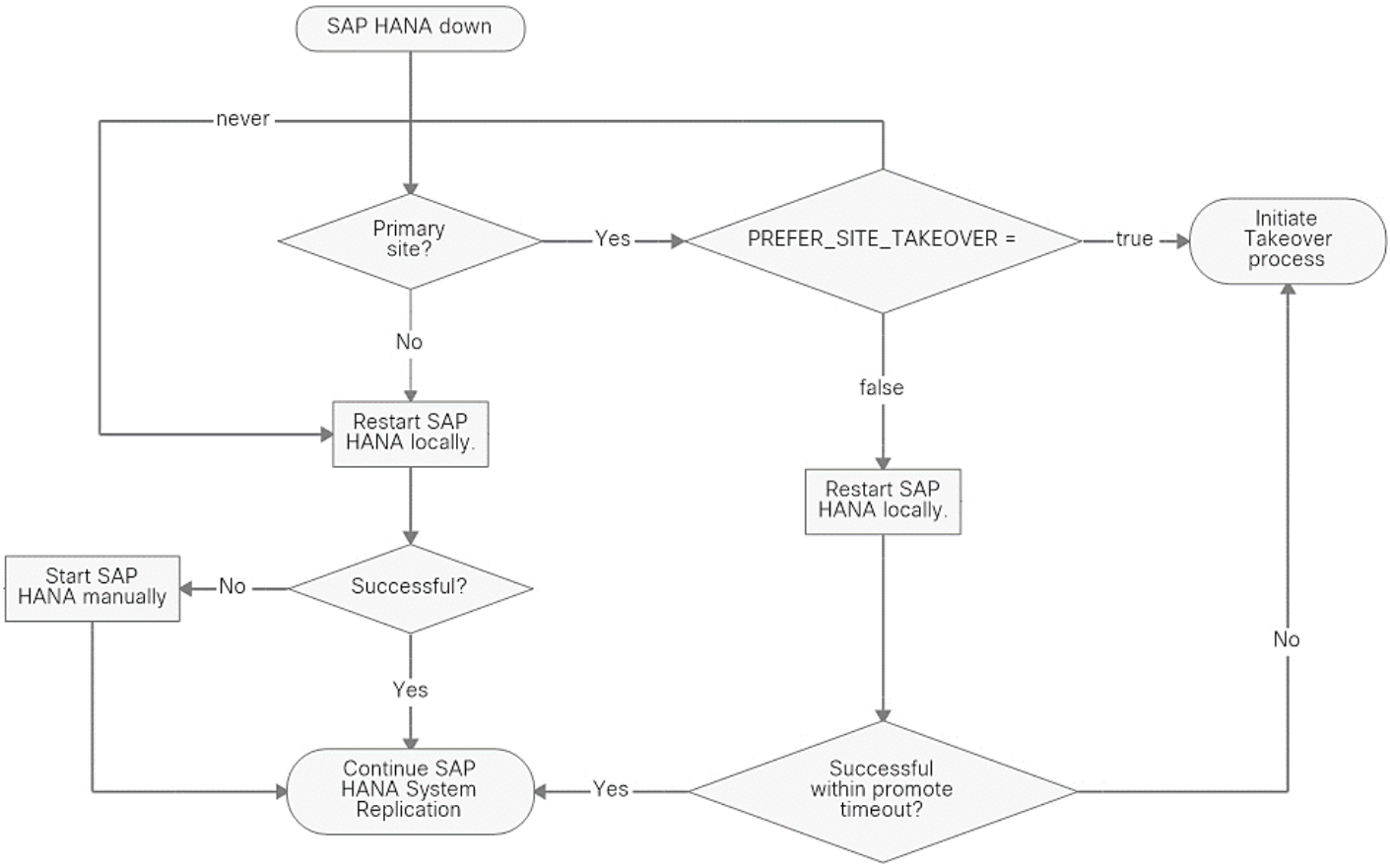


Figure 1.
Cluster behavior when the SAP HANA system is unavailable

Appendix: Disabling system replication

Refer to the following SAP help article for steps to disable system replication: https://help.sap.com/viewer/6b94445c94ae495c83a19646e7c3fd56/2.0.03/en-US/9a4a4cdcda454663ba0c75d180c7ed11.html.

For more information

Refer to the following resources for additional information:

- Configuring and managing RHEL 7 High Availability Add-On software
- Exploring concepts of RHEL high-availability clusters: Fencing and STONITH
- Configuring IPMI over LAN: Cisco UCS

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **https://www.cisco.com/go/offices**.

Printed in USA 219376-12 02/20