

High availability of SAP HANA on Azure VMs on Red Hat Enterprise Linux

Article • 04/06/2023

For on-premises development, you can use either HANA System Replication or use shared storage to establish high availability for SAP HANA. On Azure virtual machines (VMs), HANA System Replication on Azure is currently the only supported high availability function. SAP HANA Replication consists of one primary node and at least one secondary node. Changes to the data on the primary node are replicated to the secondary node synchronously or asynchronously.

This article describes how to deploy and configure the virtual machines, install the cluster framework, and install and configure SAP HANA System Replication. In the example configurations, installation commands, instance number **03**, and HANA System ID **HN1** are used.

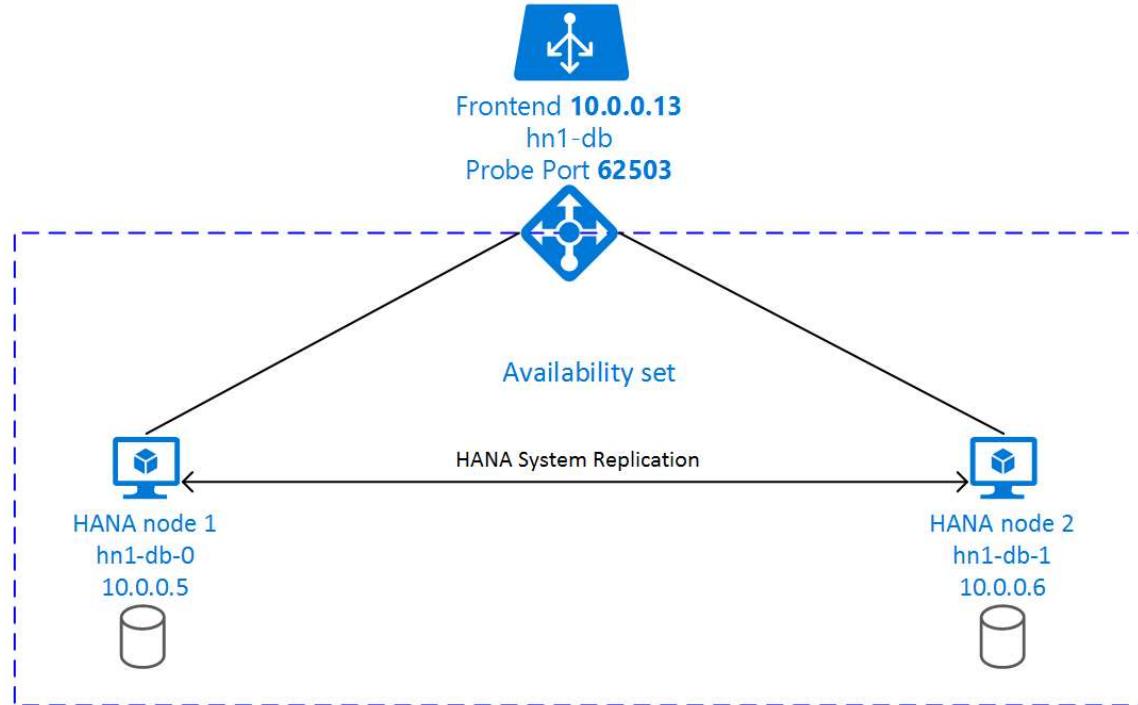
Read the following SAP Notes and papers first:

- SAP Note [1928533](#) , which has:
 - The list of Azure VM sizes that are supported for the deployment of SAP software.
 - Important capacity information for Azure VM sizes.
 - The supported SAP software, and operating system (OS) and database combinations.
 - The required SAP kernel version for Windows and Linux on Microsoft Azure.
- SAP Note [2015553](#) lists prerequisites for SAP-supported SAP software deployments in Azure.
- SAP Note [2002167](#) has recommended OS settings for Red Hat Enterprise Linux
- SAP Note [2009879](#) has SAP HANA Guidelines for Red Hat Enterprise Linux
- SAP Note [3108302](#) has SAP HANA Guidelines for Red Hat Enterprise Linux 9.x
- SAP Note [2178632](#) has detailed information about all monitoring metrics reported for SAP in Azure.
- SAP Note [2191498](#) has the required SAP Host Agent version for Linux in Azure.
- SAP Note [2243692](#) has information about SAP licensing on Linux in Azure.
- SAP Note [1999351](#) has additional troubleshooting information for the Azure Enhanced Monitoring Extension for SAP.
- [SAP Community WIKI](#) has all required SAP Notes for Linux.
- [Azure Virtual Machines planning and implementation for SAP on Linux](#)

- Azure Virtual Machines deployment for SAP on Linux (this article)
- Azure Virtual Machines DBMS deployment for SAP on Linux
- SAP HANA system replication in pacemaker cluster
- General RHEL documentation
 - High Availability Add-On Overview
 - High Availability Add-On Administration
 - High Availability Add-On Reference
 - HANA scale-up system replication with RHEL HA add-on
- Azure specific RHEL documentation:
 - Support Policies for RHEL High Availability Clusters - Microsoft Azure Virtual Machines as Cluster Members
 - Installing and Configuring a Red Hat Enterprise Linux 7.4 (and later) High-Availability Cluster on Microsoft Azure
 - Install SAP HANA on Red Hat Enterprise Linux for Use in Microsoft Azure

Overview

To achieve high availability, SAP HANA is installed on two virtual machines. The data is replicated by using HANA System Replication.



SAP HANA System Replication setup uses a dedicated virtual hostname and virtual IP addresses. On Azure, a load balancer is required to use a virtual IP address. The presented configuration shows a load balancer with:

- Front-end IP address: 10.0.0.13 for hn1-db
- Probe Port: 62503

Deploy for Linux

The Azure Marketplace contains an image for Red Hat Enterprise Linux 7.4 for SAP HANA that you can use to deploy new virtual machines.

Deploy with a template

You can use one of the quickstart templates that are on GitHub to deploy all the required resources. The template deploys the virtual machines, the load balancer, the availability set, and so on. To deploy the template, follow these steps:

1. Open the [database template](#) on the Azure portal.
2. Enter the following parameters:
 - **Sap System ID:** Enter the SAP system ID of the SAP system you want to install. The ID is used as a prefix for the resources that are deployed.
 - **Os Type:** Select one of the Linux distributions. For this example, select **RHEL 7**.
 - **Db Type:** Select **HANA**.
 - **Sap System Size:** Enter the number of SAPS that the new system is going to provide. If you're not sure how many SAPS the system requires, ask your SAP Technology Partner or System Integrator.
 - **System Availability:** Select **HA**.
 - **Admin Username, Admin Password or SSH key:** A new user is created that can be used to sign in to the machine.
 - **Subnet ID:** If you want to deploy the VM into an existing VNet where you have a subnet defined the VM should be assigned to, name the ID of that specific subnet. The ID usually looks like `/subscriptions/<subscription ID>/resourceGroups/<resource group name>/providers/Microsoft.Network/virtualNetworks/<virtual network name>/subnets/<subnet name>`. Leave empty, if you want to create a new virtual network

Manual deployment

1. Create a resource group.
2. Create a virtual network.
3. Create an availability set.
Set the max update domain.
4. Create a load balancer (internal). We recommend [standard load balancer](#).
 - Select the virtual network created in step 2.
5. Create virtual machine 1.
Use at least Red Hat Enterprise Linux 7.4 for SAP HANA. This example uses the [Red Hat Enterprise Linux 7.4 for SAP HANA image](#). Select the availability set created in step 3.
6. Create virtual machine 2.
Use at least Red Hat Enterprise Linux 7.4 for SAP HANA. This example uses the [Red Hat Enterprise Linux 7.4 for SAP HANA image](#). Select the availability set created in step 3.
7. Add data disks.

ⓘ Important

Floating IP is not supported on a NIC secondary IP configuration in load-balancing scenarios. For details see [Azure Load balancer Limitations](#). If you need additional IP address for the VM, deploy a second NIC.

ⓘ Note

When VMs without public IP addresses are placed in the backend pool of internal (no public IP address) Standard Azure load balancer, there will be no outbound internet connectivity, unless additional configuration is performed to allow routing to public end points. For details on how to achieve outbound connectivity see [Public endpoint connectivity for Virtual Machines using Azure Standard Load Balancer in SAP high-availability scenarios](#).

1. To set up standard load balancer, follow these configuration steps:
 - a. First, create a front-end IP pool:

- i. Open the load balancer, select **frontend IP pool**, and select **Add**.
 - ii. Enter the name of the new front-end IP pool (for example, **hana-frontend**).
 - iii. Set the **Assignment** to **Static** and enter the IP address (for example, **10.0.0.13**).
 - iv. Select **OK**.
 - v. After the new front-end IP pool is created, note the pool IP address.
- b. Create a single back-end pool:
- i. Open the load balancer, select **Backend pools**, and then select **Add**.
 - ii. Enter the name of the new back-end pool (for example, **hana-backend**).
 - iii. Select **NIC** for Backend Pool Configuration.
 - iv. Select **Add a virtual machine**.
 - v. Select the virtual machines of the HANA cluster.
 - vi. Select **Add**.
 - vii. Select **Save**.
- c. Next, create a health probe:
- i. Open the load balancer, select **health probes**, and select **Add**.
 - ii. Enter the name of the new health probe (for example, **hana-hp**).
 - iii. Select **TCP** as the protocol and port **62503**. Keep the **Interval** value set to **5**.
 - iv. Select **OK**.
- d. Next, create the load-balancing rules:
- i. Open the load balancer, select **load balancing rules**, and select **Add**.
 - ii. Enter the name of the new load balancer rule (for example, **hana-lb**).
 - iii. Select the front-end IP address, the back-end pool, and the health probe that you created earlier (for example, **hana-frontend**, **hana-backend** and **hana-hp**).
 - iv. Increase idle timeout to 30 minutes
 - v. Select **HA Ports**.
 - vi. Increase the **idle timeout** to 30 minutes.
 - vii. Make sure to **enable Floating IP**.
 - viii. Select **OK**.

For more information about the required ports for SAP HANA, read the chapter [Connections to Tenant Databases](#) in the [SAP HANA Tenant Databases](#) guide or [SAP Note 2388694](#).

 **Important**

Do not enable TCP timestamps on Azure VMs placed behind Azure Load Balancer. Enabling TCP timestamps will cause the health probes to fail. Set parameter `net.ipv4.tcp_timestamps` to 0. For details see [Load Balancer health probes](#). See also SAP note [2382421](#).

Install SAP HANA

The steps in this section use the following prefixes:

- [A]: The step applies to all nodes.
- [1]: The step applies to node 1 only.
- [2]: The step applies to node 2 of the Pacemaker cluster only.

1. [A] Set up the disk layout: **Logical Volume Manager (LVM)**.

We recommend that you use LVM for volumes that store data and log files. The following example assumes that the virtual machines have four data disks attached that are used to create two volumes.

List all of the available disks:

```
ls /dev/disk/azure/scsi1/lun*
```

Example output:

```
/dev/disk/azure/scsi1/lun0  /dev/disk/azure/scsi1/lun1
/dev/disk/azure/scsi1/lun2  /dev/disk/azure/scsi1/lun3
```

Create physical volumes for all of the disks that you want to use:

```
sudo pvcreate /dev/disk/azure/scsi1/lun0
sudo pvcreate /dev/disk/azure/scsi1/lun1
```

```
sudo pvcreate /dev/disk/azure/scsi1/lun2  
sudo pvcreate /dev/disk/azure/scsi1/lun3
```

Create a volume group for the data files. Use one volume group for the log files and one for the shared directory of SAP HANA:

```
sudo vgcreate vg_hana_data_HN1 /dev/disk/azure/scsi1/lun0  
/dev/disk/azure/scsi1/lun1  
sudo vgcreate vg_hana_log_HN1 /dev/disk/azure/scsi1/lun2  
sudo vgcreate vg_hana_shared_HN1 /dev/disk/azure/scsi1/lun3
```

Create the logical volumes. A linear volume is created when you use `lvcreate` without the `-i` switch. We suggest that you create a striped volume for better I/O performance, and align the stripe sizes to the values documented in [SAP HANA VM storage configurations](#). The `-i` argument should be the number of the underlying physical volumes and the `-I` argument is the stripe size. In this document, two physical volumes are used for the data volume, so the `-i` switch argument is set to **2**. The stripe size for the data volume is **256KiB**. One physical volume is used for the log volume, so no `-i` or `-I` switches are explicitly used for the log volume commands.

Important

Use the `-i` switch and set it to the number of the underlying physical volume when you use more than one physical volume for each data, log, or shared volumes. Use the `-I` switch to specify the stripe size, when creating a striped volume.

See [SAP HANA VM storage configurations](#) for recommended storage configurations, including stripe sizes and number of disks.

```
sudo lvcreate -i 2 -I 256 -l 100%FREE -n hana_data vg_hana_data_HN1  
sudo lvcreate -l 100%FREE -n hana_log vg_hana_log_HN1  
sudo lvcreate -l 100%FREE -n hana_shared vg_hana_shared_HN1  
sudo mkfs.xfs /dev/vg_hana_data_HN1/hana_data  
sudo mkfs.xfs /dev/vg_hana_log_HN1/hana_log  
sudo mkfs.xfs /dev/vg_hana_shared_HN1/hana_shared
```

Create the mount directories and copy the UUID of all of the logical volumes:

```
sudo mkdir -p /hana/data/HN1
sudo mkdir -p /hana/log/HN1
sudo mkdir -p /hana/shared/HN1
# Write down the ID of /dev/vg_hana_data_HN1/hana_data,
# /dev/vg_hana_log_HN1/hana_log, and /dev/vg_hana_shared_HN1/hana_shared
sudo blkid
```

Create fstab entries for the three logical volumes:

```
sudo vi /etc/fstab
```

Insert the following line in the /etc/fstab file:

```
/dev/disk/by-uuid/<UUID of /dev/mapper/vg_hana_data_HN1-hana_data>
/hana/data/HN1 xfs defaults,nofail 0 2
/dev/disk/by-uuid/<UUID of /dev/mapper/vg_hana_log_HN1-hana_log>
/hana/log/HN1 xfs defaults,nofail 0 2
/dev/disk/by-uuid/<UUID of /dev/mapper/vg_hana_shared_HN1-hana_shared>
/hana/shared/HN1 xfs defaults,nofail 0 2
```

Mount the new volumes:

```
sudo mount -a
```

2. [A] Set up the disk layout: Plain Disks.

For demo systems, you can place your HANA data and log files on one disk. Create a partition on /dev/disk/azure/scsi1/lun0 and format it with xfs:

```
sudo sh -c 'echo -e "n\nn\nn\nn\nnw\n" | fdisk /dev/disk/azure/scsi1/lun0'
sudo mkfs.xfs /dev/disk/azure/scsi1/lun0-part1
```

```
# Write down the ID of /dev/disk/azure/scsi1/lun0-part1  
sudo /sbin/blkid  
sudo vi /etc/fstab
```

Insert this line in the /etc/fstab file:

```
/dev/disk/by-uuid/<UUID> /hana xfs defaults,nofail 0 2
```

Create the target directory and mount the disk:

```
sudo mkdir /hana  
sudo mount -a
```

3. [A] Set up host name resolution for all hosts.

You can either use a DNS server or modify the /etc/hosts file on all nodes. This example shows you how to use the /etc/hosts file. Replace the IP address and the hostname in the following commands:

```
sudo vi /etc/hosts
```

Insert the following lines in the /etc/hosts file. Change the IP address and hostname to match your environment:

```
10.0.0.5 hn1-db-0  
10.0.0.6 hn1-db-1
```

4. [A] RHEL for HANA configuration

Configure RHEL as described in the following notes:

- [2447641 - Additional packages required for installing SAP HANA SPS 12 on RHEL 7.X](#)

- [2292690 - SAP HANA DB: Recommended OS settings for RHEL 7](#)
- [2777782 - SAP HANA DB: Recommended OS Settings for RHEL 8](#)
- [2455582 - Linux: Running SAP applications compiled with GCC 6.x](#)
- [2593824 - Linux: Running SAP applications compiled with GCC 7.x](#)
- [2886607 - Linux: Running SAP applications compiled with GCC 9.x](#)

5. [A] Install the SAP HANA

To install SAP HANA System Replication, see [Automating SAP HANA Scale-Up System Replication using the RHEL HA Add-On](#).

- Run the **hdblcm** program from the HANA DVD. Enter the following values at the prompt:
 - Choose installation: Enter **1**.
 - Select additional components for installation: Enter **1**.
 - Enter Installation Path [/hana/shared]: Select Enter.
 - Enter Local Host Name [..]: Select Enter.
 - Do you want to add additional hosts to the system? (y/n) [n]: Select Enter.
 - Enter SAP HANA System ID: Enter the SID of HANA, for example: **HN1**.
 - Enter Instance Number [00]: Enter the HANA Instance number. Enter **03** if you used the Azure template or followed the manual deployment section of this article.
 - Select Database Mode / Enter Index [1]: Select Enter.
 - Select System Usage / Enter Index [4]: Select the system usage value.
 - Enter Location of Data Volumes [/hana/data/HN1]: Select Enter.
 - Enter Location of Log Volumes [/hana/log/HN1]: Select Enter.
 - Restrict maximum memory allocation? [n]: Select Enter.
 - Enter Certificate Host Name For Host '...' [...]: Select Enter.
 - Enter SAP Host Agent User (sapadm) Password: Enter the host agent user password.
 - Confirm SAP Host Agent User (sapadm) Password: Enter the host agent user password again to confirm.
 - Enter System Administrator (hdbadm) Password: Enter the system administrator password.
 - Confirm System Administrator (hdbadm) Password: Enter the system administrator password again to confirm.
 - Enter System Administrator Home Directory [/usr/sap/HN1/home]: Select Enter.
 - Enter System Administrator Login Shell [/bin/sh]: Select Enter.
 - Enter System Administrator User ID [1001]: Select Enter.

- Enter ID of User Group (sapsys) [79]: Select Enter.
- Enter Database User (SYSTEM) Password: Enter the database user password.
- Confirm Database User (SYSTEM) Password: Enter the database user password again to confirm.
- Restart system after machine reboot? [n]: Select Enter.
- Do you want to continue? (y/n): Validate the summary. Enter y to continue.

6. [A] Upgrade the SAP Host Agent.

Download the latest SAP Host Agent archive from the [SAP Software Center](#) and run the following command to upgrade the agent. Replace the path to the archive to point to the file that you downloaded:

```
sudo /usr/sap/hostctrl/exe/saphostexec -upgrade -archive <path to SAP Host Agent SAR>
```

7. [A] Configure firewall

Create the firewall rule for the Azure load balancer probe port.

```
sudo firewall-cmd --zone=public --add-port=62503/tcp  
sudo firewall-cmd --zone=public --add-port=62503/tcp --permanent
```

Configure SAP HANA 2.0 System Replication

The steps in this section use the following prefixes:

- [A]: The step applies to all nodes.
- [1]: The step applies to node 1 only.
- [2]: The step applies to node 2 of the Pacemaker cluster only.

1. [A] Configure firewall

Create firewall rules to allow HANA System Replication and client traffic. The required ports are listed on [TCP/IP Ports of All SAP Products](#). The following commands are just an example to allow HANA 2.0 System Replication and client traffic to database SYSTEMDB, HN1 and NW1.

```
sudo firewall-cmd --zone=public --add-port=40302/tcp --permanent
sudo firewall-cmd --zone=public --add-port=40302/tcp
sudo firewall-cmd --zone=public --add-port=40301/tcp --permanent
sudo firewall-cmd --zone=public --add-port=40301/tcp
sudo firewall-cmd --zone=public --add-port=40307/tcp --permanent
sudo firewall-cmd --zone=public --add-port=40307/tcp
sudo firewall-cmd --zone=public --add-port=40303/tcp --permanent
sudo firewall-cmd --zone=public --add-port=40303/tcp
sudo firewall-cmd --zone=public --add-port=40340/tcp --permanent
sudo firewall-cmd --zone=public --add-port=40340/tcp
sudo firewall-cmd --zone=public --add-port=30340/tcp --permanent
sudo firewall-cmd --zone=public --add-port=30340/tcp
sudo firewall-cmd --zone=public --add-port=30341/tcp --permanent
sudo firewall-cmd --zone=public --add-port=30341/tcp
sudo firewall-cmd --zone=public --add-port=30342/tcp --permanent
sudo firewall-cmd --zone=public --add-port=30342/tcp
```

2. [1] Create the tenant database.

If you're using SAP HANA 2.0 or MDC, create a tenant database for your SAP NetWeaver system. Replace **NW1** with the SID of your SAP system.

Execute as <hanasid>adm the following command:

```
hdbsql -u SYSTEM -p "passwd" -i 03 -d SYSTEMDB 'CREATE DATABASE NW1 SYSTEM
USER PASSWORD "passwd"'
```

3. [1] Configure System Replication on the first node:

Backup the databases as <hanasid>adm:

```
hdbsql -d SYSTEMDB -u SYSTEM -p "passwd" -i 03 "BACKUP DATA USING FILE
('initialbackupSYS')"
hdbsql -d HN1 -u SYSTEM -p "passwd" -i 03 "BACKUP DATA USING FILE
('initialbackupHN1')"
hdbsql -d NW1 -u SYSTEM -p "passwd" -i 03 "BACKUP DATA USING FILE
('initialbackupNW1')"
```

Copy the system PKI files to the secondary site:

```
scp /usr/sap/HN1/SYS/global/security/rsecssfs/data/SSFS_HN1.DAT hn1-db-1:/usr/sap/HN1/SYS/global/security/rsecssfs/data/
scp /usr/sap/HN1/SYS/global/security/rsecssfs/key/SSFS_HN1.KEY hn1-db-1:/usr/sap/HN1/SYS/global/security/rsecssfs/key/
```

Create the primary site:

```
hdbsnutil -sr_enable --name=SITE1
```

4. [2] Configure System Replication on the second node:

Register the second node to start the system replication. Run the following command as <hanasid>adm:

```
sapcontrol -nr 03 -function StopWait 600 10
hdbsnutil -sr_register --remoteHost=hn1-db-0 --remoteInstance=03 --
replicationMode=sync --name=SITE2
```

5. [1] Check replication status

Check the replication status and wait until all databases are in sync. If the status remains UNKNOWN, check your firewall settings.

```
sudo su - hn1adm -c "python
/usr/sap/HN1/HDB03/exe/python_support/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 | hn1-db-0 | 30301 | nameserver |           | 1 |       1 |
SITE1    | hn1-db-1  | 30301 |           |           | SITE2 | YES
| SYNC      | ACTIVE    |           |           |           |           |
```

```

# | HN1      | hn1-db-0 | 30307 | xsengine   |           2 |           1 |
SITE1 | hn1-db-1 | 30307 |           2 | SITE2 | YES
| SYNC     | ACTIVE    |           |           |           |
# | NW1      | hn1-db-0 | 30340 | indexserver |           2 |           1 |
SITE1 | hn1-db-1 | 30340 |           2 | SITE2 | YES
| SYNC     | ACTIVE    |           |           |           |
# | HN1      | hn1-db-0 | 30303 | indexserver |           3 |           1 |
SITE1 | hn1-db-1 | 30303 |           2 | SITE2 | YES
| SYNC     | ACTIVE    |           |           |           |
#
# status system replication site "2": ACTIVE
# overall system replication status: ACTIVE
#
# Local System Replication State
# ~~~~~
#
# mode: PRIMARY
# site id: 1
# site name: SITE1

```

Configure SAP HANA 1.0 System Replication

The steps in this section use the following prefixes:

- [A]: The step applies to all nodes.
- [1]: The step applies to node 1 only.
- [2]: The step applies to node 2 of the Pacemaker cluster only.

1. [A] Configure firewall

Create firewall rules to allow HANA System Replication and client traffic. The required ports are listed on [TCP/IP Ports of All SAP Products](#). The following commands are just an example to allow HANA 2.0 System Replication. Adapt it to your SAP HANA 1.0 installation.

```

sudo firewall-cmd --zone=public --add-port=40302/tcp --permanent
sudo firewall-cmd --zone=public --add-port=40302/tcp

```

2. [1] Create the required users.

Run the following command as root. Make sure to replace bold strings (HANA System ID **HN1** and instance number **03**) with the values of your SAP HANA installation:

```
PATH="$PATH:/usr/sap/HN1/HDB03/exe"  
hdbsql -u system -i 03 'CREATE USER hdbhasync PASSWORD "passwd"'  
hdbsql -u system -i 03 'GRANT DATA ADMIN TO hdbhasync'  
hdbsql -u system -i 03 'ALTER USER hdbhasync DISABLE PASSWORD LIFETIME'
```

3. [A] Create the keystore entry.

Run the following command as root to create a new keystore entry:

```
PATH="$PATH:/usr/sap/HN1/HDB03/exe"  
hdbuserstore SET hdbhaloc localhost:30315 hdbhasync passwd
```

4. [1] Back up the database.

Back up the databases as root:

```
PATH="$PATH:/usr/sap/HN1/HDB03/exe"  
hdbsql -d SYSTEMDB -u system -i 03 "BACKUP DATA USING FILE ('initial-backup)"
```

If you use a multi-tenant installation, also back up the tenant database:

```
hdbsql -d HN1 -u system -i 03 "BACKUP DATA USING FILE ('initialbackup)"
```

5. [1] Configure System Replication on the first node.

Create the primary site as <hanasid>adm:

```
su - hdbadm  
hdbnsutil -sr_enable --name=SITE1
```

6. [2] Configure System Replication on the secondary node.

Register the secondary site as <hanasid>adm:

```
HDB stop  
hdbsnutil -sr_register --remoteHost=hn1-db-0 --remoteInstance=03 --  
replicationMode=sync --name=SITE2  
HDB start
```

Create a Pacemaker cluster

Follow the steps in [Setting up Pacemaker on Red Hat Enterprise Linux in Azure](#) to create a basic Pacemaker cluster for this HANA server.

Implement the Python system replication hook SAPHanaSR

This is important step to optimize the integration with the cluster and improve the detection when a cluster failover is needed. It is highly recommended to configure the SAPHanaSR Python hook.

1. [A] Install the SAP HANA resource agents on **all nodes**. Make sure to enable a repository that contains the package. You don't need to enable additional repositories, if using RHEL 8.x HA-enabled image.

Bash

```
# Enable repository that contains SAP HANA resource agents  
sudo subscription-manager repos --enable="rhel-sap-hana-for-rhel-7-server-  
rpms"  
  
sudo yum install -y resource-agents-sap-hana
```

2. [A] Install the HANA "system replication hook". The hook needs to be installed on both HANA DB nodes.

 **Tip**

The Python hook can only be implemented for HANA 2.0.

- Prepare the hook as root.

Bash

```
mkdir -p /hana/shared/myHooks
cp /usr/share/SAPHanaSR/srHook/SAPHanaSR.py /hana/shared/myHooks
chown -R hn1adm:sapsys /hana/shared/myHooks
```

- Stop HANA on both nodes. Execute as <sid>adm:

Bash

```
sapcontrol -nr 03 -function StopSystem
```

- Adjust global.ini on each cluster node.

Bash

```
# add to global.ini
[ha_dr_provider_SAPHanaSR]
provider = SAPHanaSR
path = /hana/shared/myHooks
execution_order = 1

[trace]
ha_dr_saphanasr = info
```

- [A] The cluster requires sudoers configuration on each cluster node for <sid>adm. In this example that is achieved by creating a new file. Execute the commands as root.

Bash

```
sudo visudo -f /etc/sudoers.d/20-saphana
# Insert the following lines and then save
Cmnd_Alias SITE1_SOK    = /usr/sbin/crm_attribute -n
hana_hn1_site_srHook_SITE1 -v SOK -t crm_config -s SAPHanaSR
Cmnd_Alias SITE1_SFAIL = /usr/sbin/crm_attribute -n
hana_hn1_site_srHook_SITE1 -v SFAIL -t crm_config -s SAPHanaSR
Cmnd_Alias SITE2_SOK    = /usr/sbin/crm_attribute -n
hana_hn1_site_srHook_SITE2 -v SOK -t crm_config -s SAPHanaSR
Cmnd_Alias SITE2_SFAIL = /usr/sbin/crm_attribute -n
hana_hn1_site_srHook_SITE2 -v SFAIL -t crm_config -s SAPHanaSR
hn1adm ALL=(ALL) NOPASSWD: SITE1_SOK, SITE1_SFAIL, SITE2_SOK, SITE2_SFAIL
Defaults!SITE1_SOK, SITE1_SFAIL, SITE2_SOK, SITE2_SFAIL !requiretty
```

- [A] Start SAP HANA on both nodes. Execute as <sid>adm.

```
Bash
```

```
sapcontrol -nr 03 -function StartSystem
```

5. [1] Verify the hook installation. Execute as <sid>adm on the active HANA system replication site.

```
Bash
```

```
cdtrace
awk '/ha_dr_SAPHanaSR.*crm_attribute/ \
{ printf "%s %s %s %s\n",$2,$3,$5,$16 }' nameserver_*
# Example output
# 2021-04-12 21:36:16.911343 ha_dr_SAPHanaSR SFAIL
# 2021-04-12 21:36:29.147808 ha_dr_SAPHanaSR SFAIL
# 2021-04-12 21:37:04.898680 ha_dr_SAPHanaSR SOK
```

For more details on the implementation of the SAP HANA system replication hook see [Enable the SAP HA/DR provider hook](#).

Create SAP HANA cluster resources

Create the HANA topology. Run the following commands on one of the Pacemaker cluster nodes:

```
sudo pcs property set maintenance-mode=true

# Replace the bold string with your instance number and HANA system ID
sudo pcs resource create SAPHanaTopology_HN1_03 SAPHanaTopology SID=HN1
InstanceNumber=03 \
op start timeout=600 op stop timeout=300 op monitor interval=10 timeout=600 \
clone clone-max=2 clone-node-max=1 interleave=true
```

Next, create the HANA resources.

ⓘ Note

This article contains references to the term *slave*, a term that Microsoft no longer uses. When the term is removed from the software, we'll remove it from this article.

If building a cluster on **RHEL 7.x**, use the following commands:

```
# Replace the bold string with your instance number, HANA system ID, and the
# front-end IP address of the Azure load balancer.

sudo pcs resource create SAPHana_HN1_03 SAPHana SID=HN1 InstanceNumber=03
PREFER_SITE_TAKEOVER=true DUPLICATE_PRIMARY_TIMEOUT=7200
AUTOMATED_REGISTER=false \
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

sudo pcs resource create vip_HN1_03 IPAddr2 ip="10.0.0.13"
sudo pcs resource create nc_HN1_03 azure-lb port=62503
sudo pcs resource group add g_ip_HN1_03 nc_HN1_03 vip_HN1_03

sudo pcs constraint order SAPHanaTopology_HN1_03-clone then SAPHana_HN1_03-
master symmetrical=false
sudo pcs constraint colocation add g_ip_HN1_03 with master SAPHana_HN1_03-mas-
ter 4000

sudo pcs resource defaults resource-stickiness=1000
sudo pcs resource defaults migration-threshold=5000

sudo pcs property set maintenance-mode=false
```

If building a cluster on **RHEL 8.x**, use the following commands:

```
# Replace the bold string with your instance number, HANA system ID, and the
# front-end IP address of the Azure load balancer.

sudo pcs resource create SAPHana_HN1_03 SAPHana SID=HN1 InstanceNumber=03
PREFER_SITE_TAKEOVER=true DUPLICATE_PRIMARY_TIMEOUT=7200
AUTOMATED_REGISTER=false \
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 \
promotable notify=true clone-max=2 clone-node-max=1 interleave=true

sudo pcs resource create vip_HN1_03 IPAddr2 ip="10.0.0.13"
sudo pcs resource create nc_HN1_03 azure-lb port=62503
sudo pcs resource group add g_ip_HN1_03 nc_HN1_03 vip_HN1_03
```

```
sudo pcs constraint order SAPHanaTopology_HN1_03-clone then SAPHana_HN1_03-clone symmetrical=false  
sudo pcs constraint colocation add g_ip_HN1_03 with master SAPHana_HN1_03-clone 4000  
  
sudo pcs resource defaults update resource-stickiness=1000  
sudo pcs resource defaults update migration-threshold=5000  
  
sudo pcs property set maintenance-mode=false
```

ⓘ Important

It's a good idea to set `AUTOMATED_REGISTER` to `false`, while you're performing failover tests, to prevent a failed primary instance to automatically register as secondary. After testing, as a best practice, set `AUTOMATED_REGISTER` to `true`, so that after takeover, system replication can resume automatically.

Make sure that the cluster status is ok and that all of the resources are started. It's not important on which node the resources are running.

ⓘ Note

The timeouts in the above configuration are just examples and may need to be adapted to the specific HANA setup. For instance, you may need to increase the start timeout, if it takes longer to start the SAP HANA database.

```
sudo pcs status  
  
# Online: [ hn1-db-0 hn1-db-1 ]  
#  
# Full list of resources:  
#  
# azure_fence      (stonith:fence_azure_arm):      Started hn1-db-0  
# Clone Set: SAPHanaTopology_HN1_03-clone [SAPHanaTopology_HN1_03]  
#           Started: [ hn1-db-0 hn1-db-1 ]  
# Master/Slave Set: SAPHana_HN1_03-master [SAPHana_HN1_03]  
#           Masters: [ hn1-db-0 ]  
#           Slaves: [ hn1-db-1 ]  
# Resource Group: g_ip_HN1_03
```

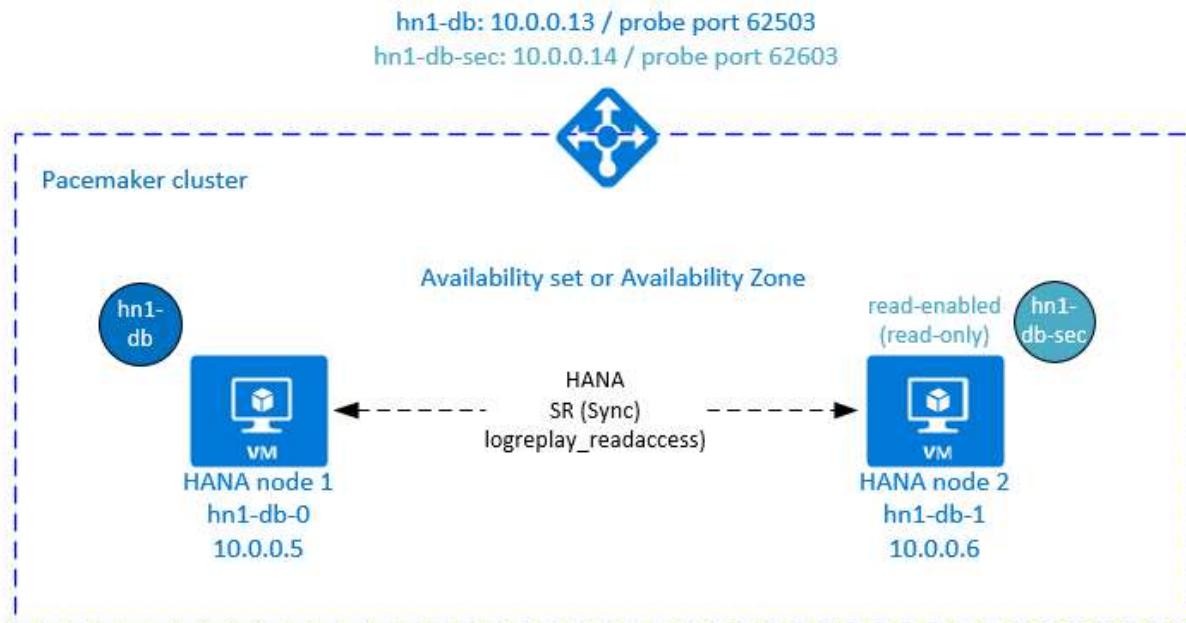
```
# nc_HN1_03 (ocf::heartbeat:azure-lb): Started hn1-db-0
# vip_HN1_03 (ocf::heartbeat:IPAddr2): Started hn1-db-0
```

Configure HANA active/read enabled system replication in Pacemaker cluster

Starting with SAP HANA 2.0 SPS 01 SAP allows Active/Read-Enabled setups for SAP HANA System Replication, where the secondary systems of SAP HANA system replication can be used actively for read-intense workloads. To support such setup in a cluster a second virtual IP address is required which allows clients to access the secondary read-enabled SAP HANA database. To ensure that the secondary replication site can still be accessed after a takeover has occurred the cluster needs to move the virtual IP address around with the secondary of the SAPHana resource.

This section describes the additional steps that are required to manage HANA Active/Read enabled system replication in a Red Hat high availability cluster with second virtual IP.

Before proceeding further, make sure you have fully configured Red Hat High Availability Cluster managing SAP HANA database as described in above segments of the documentation.



Additional setup in Azure load balancer for active/read-enabled setup

To proceed with additional steps on provisioning second virtual IP, make sure you have configured Azure Load Balancer as described in [Manual Deployment](#) section.

1. For **standard** load balancer, follow below additional steps on the same load balancer that you had created in earlier section.

- a. Create a second front-end IP pool:

- Open the load balancer, select **frontend IP pool**, and select **Add**.
- Enter the name of the second front-end IP pool (for example, **hana-secondaryIP**).
- Set the **Assignment** to **Static** and enter the IP address (for example, **10.0.0.14**).
- Select **OK**.
- After the new front-end IP pool is created, note the pool IP address.

- b. Next, create a health probe:

- Open the load balancer, select **health probes**, and select **Add**.
- Enter the name of the new health probe (for example, **hana-secondaryhp**).
- Select **TCP** as the protocol and port **62603**. Keep the **Interval** value set to 5, and the **Unhealthy threshold** value set to 2.
- Select **OK**.

- c. Next, create the load-balancing rules:

- Open the load balancer, select **load balancing rules**, and select **Add**.
- Enter the name of the new **load balancer rule** (for example, **hana-secondarylb**).
- Select the front-end IP address , the back-end pool, and the health probe that you created earlier (for example, **hana-secondaryIP**, **hana-backend** and **hana-secondaryhp**).
- Select **HA Ports**.
- Make sure to **enable Floating IP**.
- Select **OK**.

Configure HANA active/read enabled system replication

The steps to configure HANA system replication are described in [Configure SAP HANA 2.0 System Replication](#) section. If you are deploying read-enabled secondary scenario, while configuring system replication on the second node, execute following command as `hanasidadm`:

```
sapcontrol -nr 03 -function StopWait 600 10  
  
hdbnsutil -sr_register --remoteHost=hn1-db-0 --remoteInstance=03 --  
replicationMode=sync --name=SITE2 --operationMode=logreplay_readaccess
```

Adding a secondary virtual IP address resource for an active/read-enabled setup

The second virtual IP and the appropriate colocation constraint can be configured with the following commands:

```
pcs property set maintenance-mode=true  
  
pcs resource create secvip_HN1_03 ocf:heartbeat:IPAddr2 ip="10.40.0.16"  
  
pcs resource create secnc_HN1_03 ocf:heartbeat:azure-lb port=62603  
  
pcs resource group add g_secip_HN1_03 secnc_HN1_03 secvip_HN1_03  
  
pcs constraint location g_secip_HN1_03 rule score=INFINITY hana_hn1_sync_state  
eq SOK and hana_hn1_roles eq 4:S:master1:master:worker:master  
  
pcs constraint location g_secip_HN1_03 rule score=4000 hana_hn1_sync_state eq  
PRIM and hana_hn1_roles eq 4:P:master1:master:worker:master  
  
pcs property set maintenance-mode=false
```

Make sure that the cluster status is ok and that all of the resources are started. The second virtual IP will run on the secondary site along with SAPHana secondary resource.

```
sudo pcs status  
  
# Online: [ hn1-db-0 hn1-db-1 ]
```

```
#  
# Full List of Resources:  
#   rsc_hdb_azr_agt    (stonith:fence_azure_arm):      Started hn1-db-0  
#   Clone Set: SAPHanaTopology_HN1_03-clone [SAPHanaTopology_HN1_03]:  
#     Started: [ hn1-db-0 hn1-db-1 ]  
#   Clone Set: SAPHana_HN1_03-clone [SAPHana_HN1_03] (promotable):  
#     Masters: [ hn1-db-0 ]  
#     Slaves: [ hn1-db-1 ]  
#   Resource Group: g_ip_HN1_03:  
#     nc_HN1_03          (ocf::heartbeat:azure-lb):      Started hn1-db-0  
#     vip_HN1_03          (ocf::heartbeat:IPAddr2):      Started hn1-db-0  
#   Resource Group: g_secip_HN1_03:  
#     secnc_HN1_03        (ocf::heartbeat:azure-lb):      Started hn1-db-1  
#     secvip_HN1_03       (ocf::heartbeat:IPAddr2):      Started hn1-db-1
```

In next section, you can find the typical set of failover tests to execute.

Be aware of the second virtual IP behavior, while testing a HANA cluster configured with read-enabled secondary:

1. When you migrate **SAPHana_HN1_03** cluster resource to secondary site **hn1-db-1**, the second virtual IP will continue to run on the same site **hn1-db-1**. If you have set AUTOMATED_REGISTER="true" for the resource and HANA system replication is registered automatically on **hn1-db-0**, then your second virtual IP will also move to **hn1-db-0**.
2. On testing server crash, second virtual IP resources (**secvip_HN1_03**) and azure load balancer port resource (**secnc_HN1_03**) will run on primary server alongside the primary virtual IP resources. So, till the time secondary server is down, application that are connected to read-enabled HANA database will connect to primary HANA database. The behavior is expected as you do not want applications that are connected to read-enabled HANA database to be inaccessible till the time secondary server is unavailable.
3. During failover and fallback of second virtual IP address, it may happen that the existing connections on applications that uses second virtual IP to connect to the HANA database may get interrupted.

The setup maximizes the time that the second virtual IP resource will be assigned to a node where a healthy SAP HANA instance is running.

Test the cluster setup

This section describes how you can test your setup. Before you start a test, make sure that Pacemaker does not have any failed action (via `pcs status`), there are no unexpected location constraints (for example leftovers of a migration test) and that HANA is sync state, for example with `systemReplicationStatus`:

```
[root@hn1-db-0 ~]# sudo su - hn1adm -c "python /usr/sap/HN1/HDB03/exe/python_support/systemReplicationStatus.py"
```

Test the migration

Resource state before starting the test:

```
Clone Set: SAPHanaTopology_HN1_03-clone [SAPHanaTopology_HN1_03]
  Started: [ hn1-db-0 hn1-db-1 ]
Master/Slave Set: SAPHana_HN1_03-master [SAPHana_HN1_03]
  Masters: [ hn1-db-0 ]
  Slaves: [ hn1-db-1 ]
Resource Group: g_ip_HN1_03
  nc_HN1_03 (ocf::heartbeat:azure-lb):     Started hn1-db-0
  vip_HN1_03 (ocf::heartbeat:IPAddr2):      Started hn1-db-0
```

You can migrate the SAP HANA master node by executing the following command:

```
# On RHEL 7.x
[root@hn1-db-0 ~]# pcs resource move SAPHana_HN1_03-master
# On RHEL 8.x
[root@hn1-db-0 ~]# pcs resource move SAPHana_HN1_03-clone --master
```

If you set `AUTOMATED_REGISTER="false"`, this command should migrate the SAP HANA master node and the group that contains the virtual IP address to hn1-db-1.

Once the migration is done, the '`sudo pcs status`' output looks like this

```
Clone Set: SAPHanaTopology_HN1_03-clone [SAPHanaTopology_HN1_03]
  Started: [ hn1-db-0 hn1-db-1 ]
```

```
Master/Slave Set: SAPHana_HN1_03-master [SAPHana_HN1_03]
  Masters: [ hn1-db-1 ]
  Stopped: [ hn1-db-0 ]
Resource Group: g_ip_HN1_03
  nc_HN1_03 (ocf::heartbeat:azure-lb):     Started hn1-db-1
  vip_HN1_03 (ocf::heartbeat:IPAddr2):      Started hn1-db-1
```

The SAP HANA resource on hn1-db-0 is stopped. In this case, configure the HANA instance as secondary by executing this command:

```
[root@hn1-db-0 ~]# su - hn1adm

# Stop the HANA instance just in case it is running
hn1adm@hn1-db-0:/usr/sap/HN1/HDB03> sapcontrol -nr 03 -function StopWait 600
10
hn1adm@hn1-db-0:/usr/sap/HN1/HDB03> hdbnsutil -sr_register --remoteHost=hn1-
db-1 --remoteInstance=03 --replicationMod
e=sync --name=SITE1
```

The migration creates location constraints that need to be deleted again:

```
# Switch back to root
exit
[root@hn1-db-0 ~]# pcs resource clear SAPHana_HN1_03-master
```

Monitor the state of the HANA resource using 'pcs status'. Once HANA is started on hn1-db-0, the output should look like this

```
Clone Set: SAPHanaTopology_HN1_03-clone [SAPHanaTopology_HN1_03]
  Started: [ hn1-db-0 hn1-db-1 ]
Master/Slave Set: SAPHana_HN1_03-master [SAPHana_HN1_03]
  Masters: [ hn1-db-1 ]
  Slaves: [ hn1-db-0 ]
Resource Group: g_ip_HN1_03
  nc_HN1_03 (ocf::heartbeat:azure-lb):     Started hn1-db-1
  vip_HN1_03 (ocf::heartbeat:IPAddr2):      Started hn1-db-1
```

Test the Azure fencing agent

ⓘ Note

This article contains references to the term *slave*, a term that Microsoft no longer uses. When the term is removed from the software, we'll remove it from this article.

Resource state before starting the test:

```
Clone Set: SAPHanaTopology_HN1_03-clone [SAPHanaTopology_HN1_03]
  Started: [ hn1-db-0 hn1-db-1 ]
Master/Slave Set: SAPHana_HN1_03-master [SAPHana_HN1_03]
  Masters: [ hn1-db-1 ]
  Slaves: [ hn1-db-0 ]
Resource Group: g_ip_HN1_03
  nc_HN1_03 (ocf::heartbeat:azure-lb):     Started hn1-db-1
  vip_HN1_03 (ocf::heartbeat:IPAddr2):       Started hn1-db-1
```

You can test the setup of the Azure fencing agent by disabling the network interface on the node where SAP HANA is running as Master. See [Red Hat Knowledgebase article 79523](#) for a description on how to simulate a network failure. In this example we use the `net_breaker` script to block all access to the network.

```
[root@hn1-db-1 ~]# sh ./net_breaker.sh BreakCommCmd 10.0.0.6
```

The virtual machine should now restart or stop depending on your cluster configuration. If you set the `stonith-action` setting to off, the virtual machine is stopped and the resources are migrated to the running virtual machine.

After you start the virtual machine again, the SAP HANA resource fails to start as secondary if you set `AUTOMATED_REGISTER="false"`. In this case, configure the HANA instance as secondary by executing this command:

```
su - hn1adm

# Stop the HANA instance just in case it is running
hn1adm@hn1-db-1:/usr/sap/HN1/HDB03> sapcontrol -nr 03 -function StopWait 600
10
```

```
hn1adm@hn1-db-1:/usr/sap/HN1/HDB03> hdbnsutil -sr_register --remoteHost=hn1-db-0 --remoteInstance=03 --replicationMode=sync --name=SITE2

# Switch back to root and clean up the failed state
exit
# On RHEL 7.x
[root@hn1-db-1 ~]# pcs resource cleanup SAPHana_HN1_03-master
# On RHEL 8.x
[root@hn1-db-1 ~]# pcs resource cleanup SAPHana_HN1_03 node=<hostname on which the resource needs to be cleaned>
```

Resource state after the test:

```
Clone Set: SAPHanaTopology_HN1_03-clone [SAPHanaTopology_HN1_03]
  Started: [ hn1-db-0 hn1-db-1 ]
Master/Slave Set: SAPHana_HN1_03-master [SAPHana_HN1_03]
  Masters: [ hn1-db-0 ]
  Slaves: [ hn1-db-1 ]
Resource Group: g_ip_HN1_03
  nc_HN1_03 (ocf::heartbeat:azure-lb):     Started hn1-db-0
  vip_HN1_03 (ocf::heartbeat:IPAddr2):      Started hn1-db-0
```

Test a manual failover

Resource state before starting the test:

```
Clone Set: SAPHanaTopology_HN1_03-clone [SAPHanaTopology_HN1_03]
  Started: [ hn1-db-0 hn1-db-1 ]
Master/Slave Set: SAPHana_HN1_03-master [SAPHana_HN1_03]
  Masters: [ hn1-db-0 ]
  Slaves: [ hn1-db-1 ]
Resource Group: g_ip_HN1_03
  nc_HN1_03 (ocf::heartbeat:azure-lb):     Started hn1-db-0
  vip_HN1_03 (ocf::heartbeat:IPAddr2):      Started hn1-db-0
```

You can test a manual failover by stopping the cluster on the hn1-db-0 node:

```
[root@hn1-db-0 ~]# pcs cluster stop
```

After the failover, you can start the cluster again. If you set `AUTOMATED_REGISTER="false"`, the SAP HANA resource on the hn1-db-0 node fails to start as secondary. In this case, configure the HANA instance as secondary by executing this command:

```
[root@hn1-db-0 ~]# pcs cluster start
[root@hn1-db-0 ~]# su - hn1adm

# Stop the HANA instance just in case it is running
hn1adm@hn1-db-0:/usr/sap/HN1/HDB03> sapcontrol -nr 03 -function StopWait 600
10
hn1adm@hn1-db-0:/usr/sap/HN1/HDB03> hdbnsutil -sr_register --remoteHost=hn1-
db-1 --remoteInstance=03 --replicationMode=sync --name=SITE1

# Switch back to root and clean up the failed state
hn1adm@hn1-db-0:/usr/sap/HN1/HDB03> exit
# On RHEL 7.x
[root@hn1-db-1 ~]# pcs resource cleanup SAPHana_HN1_03-master
# On RHEL 8.x
[root@hn1-db-1 ~]# pcs resource cleanup SAPHana_HN1_03 node=<hostname on which
the resource needs to be cleaned>
```

Resource state after the test:

```
Clone Set: SAPHanaTopology_HN1_03-clone [SAPHanaTopology_HN1_03]
  Started: [ hn1-db-0 hn1-db-1 ]
Master/Slave Set: SAPHana_HN1_03-master [SAPHana_HN1_03]
  Masters: [ hn1-db-1 ]
  Slaves: [ hn1-db-0 ]
Resource Group: g_ip_HN1_03
  nc_HN1_03 (ocf::heartbeat:azure-lb):           Started hn1-db-1
  vip_HN1_03 (ocf::heartbeat:IPAddr2):           Started hn1-db-1
```

Test a manual failover

Resource state before starting the test:

```
Clone Set: SAPHanaTopology_HN1_03-clone [SAPHanaTopology_HN1_03]
  Started: [ hn1-db-0 hn1-db-1 ]
Master/Slave Set: SAPHana_HN1_03-master [SAPHana_HN1_03]
```

```
Masters: [ hn1-db-0 ]
Slaves: [ hn1-db-1 ]
Resource Group: g_ip_HN1_03
  nc_HN1_03 (ocf::heartbeat:azure-lb):     Started hn1-db-0
  vip_HN1_03 (ocf::heartbeat:IPAddr2):      Started hn1-db-0
```

You can test a manual failover by stopping the cluster on the hn1-db-0 node:

```
[root@hn1-db-0 ~]# pcs cluster stop
```

Next steps

- [Azure Virtual Machines planning and implementation for SAP](#)
- [Azure Virtual Machines deployment for SAP](#)
- [Azure Virtual Machines DBMS deployment for SAP](#)
- [SAP HANA VM storage configurations](#)