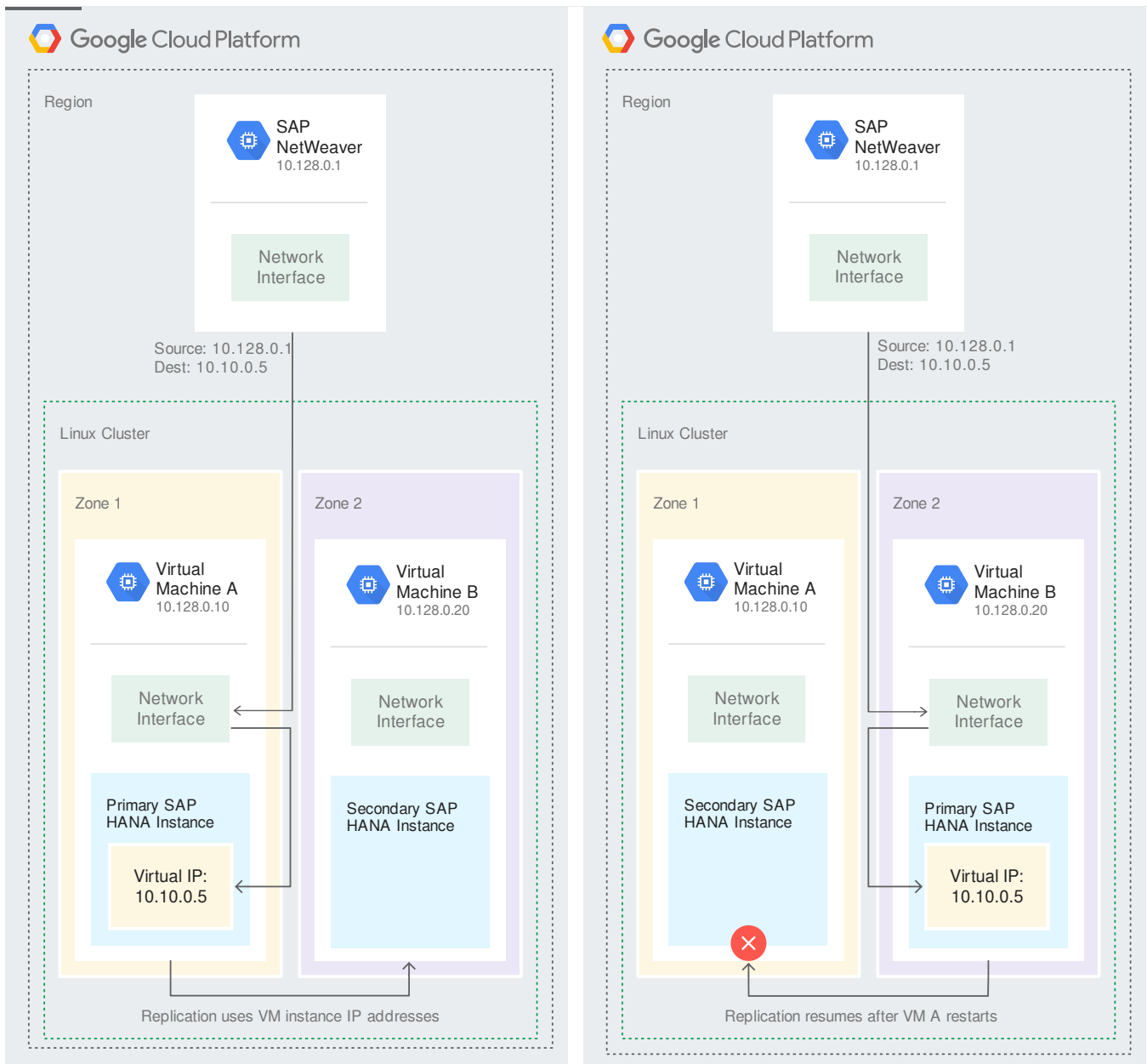


This guide shows you how to deploy a performance-optimized SUSE Linux Enterprise Server (SLES) cluster for a single host SAP HANA scaleup system on Google Cloud. You complete a configuration file template and use Cloud Deployment Manager to deploy a cluster and two SAP HANA systems that incorporate best practices from Compute Engine and SAP.

One of the SAP HANA systems functions as the primary, active system and the other functions as a secondary, standby system. Each SAP HANA system is deployed on a Compute Engine VM within the same region, ideally in different zones.



The deployed cluster includes the following functions and features:

- The Pacemaker high-availability cluster resource manager.
- A GCP fencing mechanism.
- The SUSE high-availability pattern.
- The SUSE SAPHanaSR resource agent package.
- Synchronous system replication.
- Memory preload.
- Automatic restart of the failed instance as the new secondary instance.

If you need a scale-out system with standby hosts for SAP HANA automatic host failover, use the [SAP HANA Scale-Out System with SAP HANA Host Auto-Failover Deployment Guide](/solutions/sap/docs/sap-hana-ha-scaleout-deployment-guide) (/solutions/sap/docs/sap-hana-ha-scaleout-deployment-guide) instead.

To deploy a SAP HANA system without a Linux high-availability cluster or standby hosts, use the [SAP HANA Deployment Guide](/solutions/sap/docs/sap-hana-deployment-guide) (/solutions/sap/docs/sap-hana-deployment-guide).

This guide is intended for advanced SAP HANA users who are familiar with Linux high-availability configurations for SAP HANA.

Before you create the SAP HANA high availability cluster, make sure that the following prerequisites are met:

- You or your organization has a Google Cloud account and you have created a project for the SAP HANA deployment. For information about creating Google Cloud accounts and projects, see [Setting up your Google account](/solutions/sap/docs/sap-hana-deployment-guide#setting-up-your-google-account) (/solutions/sap/docs/sap-hana-deployment-guide#setting-up-your-google-account) in the SAP HANA Deployment Guide.
- The SAP HANA installation media is stored in a Cloud Storage bucket that is available in your deployment project and region. For information about how to upload SAP HANA installation media to a Cloud Storage bucket, see [Downloading SAP HANA](/solutions/sap/docs/sap-hana-deployment-guide#creating_a_cloud_storage_bucket_for_the_sap_hana_installation_files) (/solutions/sap/docs/sap-hana-deployment-guide#creating_a_cloud_storage_bucket_for_the_sap_hana_installation_files) in the SAP HANA Deployment Guide.
- If you are using [VPC internal DNS](/compute/docs/internal-dns) (/compute/docs/internal-dns), the value of the `VmDnsSetting` variable in your project metadata must be either `GlobalOnly` or `ZonalPreferred` to enable the resolution of the node names across zones. The default setting of `VmDnsSetting` is `ZonalOnly`. For more information, see:
 - [Configuring DNS names for your project or instances](/compute/docs/internal-dns#setting-zonal-dns) (/compute/docs/internal-dns#setting-zonal-dns)
 - [Querying custom metadata](/compute/docs/storing-retrieving-metadata#querying_custom_metadata) (/compute/docs/storing-retrieving-metadata#querying_custom_metadata)
 - [Setting project-wide custom metadata](/compute/docs/storing-retrieving-metadata#projectwide) (/compute/docs/storing-retrieving-metadata#projectwide)

to avoid unintentionally exposing your VM instance to the internet, follow these recommendations:

Use a NAT gateway.

[Create firewall rules](#) (/vpc/docs/using-firewalls#creating_firewall_rules) that block all external access that you don't require.

When you create your VMs:

- Specify a network tag for each VM for use in routing and firewall rules. If you use the Deployment Manager templates that Google Cloud provides, specify a tag with **networkTag: [TAG]**.
- Create the VMs without an external IP. If you use the Deployment Manager templates that Google Cloud provides, specify **publicIP: No**.

For security purposes, create a new network. You can control who has access by adding firewall rules or by using another access control method.

If your project has a default VPC network, don't use it. Instead, create your own VPC network so that the only firewall rules in effect are those that you create explicitly.

During deployment, VM instances typically require access to the internet to download Google's monitoring agent. If you are using one of the SAP-certified Linux images that are available from Google Cloud, the VM instance also requires access to the internet in order to register the license and to access OS vendor repositories. A configuration with a NAT gateway and with VM network tags supports this access, even if the target VMs do not have external IPs.

To set up networking:

1. Go to Cloud Shell.

[Go to Cloud Shell](https://console.cloud.google.com/?cloudshell=true) (https://console.cloud.google.com/?cloudshell=true)

2. To create a new network in the custom subnetworks mode, run:

where `[YOUR_NETWORK_NAME]` is the name of the new network. The network name can contain only lowercase characters, digits, and the dash character (-).

Specify `--subnet-mode custom` to avoid using the default auto mode, which automatically creates a subnet in each Compute Engine region. For more information, see [Subnet creation mode](/vpc/docs/vpc#subnet-ranges) (`/vpc/docs/vpc#subnet-ranges`).

3. Create a subnetwork, and specify the region and IP range:

where:

- `[YOUR_SUBNETWORK_NAME]` is the new subnetwork.
- `[YOUR_NETWORK_NAME]` is the name of the network you created in the previous step.
- `[REGION]` is the region where you want the subnetwork.
- `[YOUR_RANGE]` is the IP address range, specified in CIDR format (https://wikipedia.org/wiki/Classless_Inter-Domain_Routing), such as `10.1.0.0/24`. If you plan to add more than one subnetwork, assign non-overlapping CIDR IP ranges for each subnetwork in the network. Note that each subnetwork and its internal IP ranges are mapped to a single region.

4. Optionally, repeat the previous step and add additional subnetworks.

If you intend to create one or more VMs that will not have public IP addresses, you must create a NAT gateway so that your VMs can access the Internet to download Google's monitoring agent.

If you intend to assign an external public IP address to your VM, you can skip this step.

Important: Do not remove the public IP addresses from your new VMs until after the installation of your SAP software is complete and validated.

To create a NAT gateway:

1. Create a VM to act as the NAT gateway in the subnet you just created:

where:

- [YOUR_VM_NAME] is the name of the VM you are creating that want to use for the NAT gateway.
- [YOUR_ZONE] is the zone where you want the VM.
- [YOUR_IMAGE_FAMILY] and [YOUR_IMAGE_PROJECT] specify the image you want to use (/compute/docs/images#os-compute-support) for the NAT gateway.
- [YOUR_MACHINE_TYPE] is any supported machine type. If you expect high network traffic, choose a machine type with that has at least eight virtual CPUs.
- [YOUR_SUBNETWORK_NAME] is the name of the subnetwork where you want the VM.
- [YOUR_VM_TAG] is a tag that is applied to the VM you are creating. If you use this VM as a bastion host, this tag is used to apply the related firewall rule only to this VM.

2. Create a route that is tagged so that traffic passes through the NAT VM instead of the default Internet gateway:

where:

- [YOUR_ROUTE_NAME] is the name of the route you are creating.
- [YOUR_NETWORK_NAME] is the network you created.
- [YOUR_VM_NAME] is the VM you are using for your NAT gateway.
- [YOUR_ZONE] is the zone where the VM is located.
- [YOUR_TAG_NAME] is the tag on the route that directs traffic through the NAT VM.

3. If you also want to use the NAT gateway VM as a bastion host, run the following command. This command creates a firewall rule that allows inbound SSH access to this instance from the Internet:

where:

- [YOUR_NETWORK_NAME] is the network you created.
- [YOUR_VM_TAG] is the tag you specified when you created the NAT gateway VM. This tag is used so this firewall rule applies only to the VM that hosts the NAT gateway, and not to all VMs in the network.

By default, an *implied firewall rule* blocks incoming connections from outside your Virtual Private Cloud (VPC) network. To allow incoming connections, set up a firewall rule for your VM. After an incoming connection is established with a VM, traffic is permitted in both directions over that connection.

You can create a firewall rule to allow external access to specified ports, or to restrict access between VMs on the same network. If the `default` VPC network type is used, some additional default rules also apply, such as the `default-allow-internal` rule, which allows connectivity between VMs on the same network on all ports.

Depending on the IT policy that is applicable to your environment, you might need to isolate or otherwise restrict connectivity to your database host, which you can do by creating firewall rules.

Depending on your scenario, you can create firewall rules to allow access for:

- The default SAP ports that are listed in [TCP/IP of All SAP Products](https://help.sap.com/viewer/575a9f0e56f34c6e8138439eefc32b16/2.0/en-US/616a3c0b1cc748238de9c0341b15c63c.html) (<https://help.sap.com/viewer/575a9f0e56f34c6e8138439eefc32b16/2.0/en-US/616a3c0b1cc748238de9c0341b15c63c.html>)
- Connections from your computer or your corporate network environment to your Compute Engine VM instance. If you are unsure of what IP address to use, talk to your company's network administrator.

- Communication between VMs when, for example, your database server and application server are running on different VMs. To enable communication between VMs, you must create a firewall rule to allow traffic that originates from the subnetwork.
- SSH connections to your VM instance, including [SSH from the browser](https://cloud.google.com/compute/docs/ssh-in-browser) (<https://cloud.google.com/compute/docs/ssh-in-browser>).
- Connection to your VM by using a third-party tool in Linux. Create a rule to allow access for the tool through your firewall.

The following procedure is a simplified version of the instructions for creating firewall rules. For more detailed instructions, see [Virtual Private Cloud documentation](https://cloud.google.com/vpc/docs/using-firewalls#creating_firewall_rules) (/vpc/docs/using-firewalls#creating_firewall_rules).

To create a firewall rule:

1. In the Cloud Console, go to the **Firewall rules** page.

OPEN FIREWALL RULES (<https://console.cloud.google.com/networking/firewalls/list>)

2. At the top of the page, click **Create firewall rule**.

- In the **Network** field, select the network where your VM is located.
- In the **Targets** field, specify the resources on Google Cloud that this rule applies to. For example, specify **All instances in the network**. Or to limit the rule to specific instances on Google Cloud, enter tags in **Specified target tags**.
- In the **Source filter** field, select one of the following:
 - **IP ranges** to allow incoming traffic from specific IP addresses. Specify the range of IP addresses in the **Source IP ranges** field.
 - **Subnets** to allow incoming traffic from a particular subnetwork. Specify the subnetwork name in the following **Subnets** field. You can use this option to allow access between the VMs in a 3-tier or scaleout configuration.
- In the **Protocols and ports** section, select **Specified protocols and ports** and enter **tcp: [PORT_NUMBER]**.

3. Click **Create** to create your firewall rule.

The following instructions use the Cloud Deployment Manager to create a SLES Linux cluster with two SAP HANA systems: a primary single-host SAP HANA system on one VM instance and a standby SAP HANA system on another VM instance in the same Compute Engine region. The SAP HANA systems use synchronous system replication and the standby system preloads the replicated data.

You define configuration options for the SAP HANA high-availability cluster in a Deployment Manager configuration file template.

The following instructions use the Cloud Shell, but are generally applicable to the Cloud SDK.

1. Confirm that your current quotas for resources such as persistent disks and CPUs are sufficient for the SAP HANA systems you are about to install. If your quotas are insufficient, deployment fails. For the SAP HANA quota requirements, see [Pricing and quota considerations for SAP HANA](#) (/solutions/sap/docs/sap-hana-planning-guide#costs).

[Go to the quotas page](https://console.cloud.google.com/iam-admin/quotas) (https://console.cloud.google.com/iam-admin/quotas)

2. Open the Cloud Shell or, if you installed the Cloud SDK on your local workstation, open a terminal.

[Go to the Cloud Shell](https://console.cloud.google.com/?cloudshell=true) (https://console.cloud.google.com/?cloudshell=true)

3. Download the `template.yaml` configuration file template for the SAP HANA high-availability cluster to your working directory by entering the following command in the Cloud Shell or Cloud SDK:

4. Optionally, rename the `template.yaml` file to identify the configuration it defines.

5. Open the `template.yaml` file in the Cloud Shell code editor or, if you are using the Cloud SDK, the text editor of your choice.

To open the Cloud Shell code editor, click the pencil icon in the upper right corner of the Cloud Shell terminal window.

6. In the `template.yaml` file, update the property values by replacing the brackets and their contents with the values for your installation. The properties are described in the following table.

To create the VM instances without installing SAP HANA, delete or comment out all of the lines that begin with `sap_hana_`.

Property	Data type	Description
primaryInstanceName	String	The name of the VM instance for the primary SAP HANA system. Specify the name in lowercase letters, numbers, or hyphens.
secondaryInstanceName	String	The name of the VM instance for the secondary SAP HANA system. Specify the name in lowercase letters, numbers, or hyphens.
primaryZone	String	The zone in which the primary SAP HANA system is deployed. The primary and secondary zones must be in the same region.
secondaryZone	String	The zone in which the secondary SAP HANA system will be deployed. The primary and secondary zones must be in the same region.
instanceType	String	The type of Compute Engine virtual machine (/solutions/sap/docs/sap-hana-planning-guide#vm_types) that you need to run SAP HANA on. If you need a custom VM type, specify a predefined VM type (/solutions/sap/docs/sap-hana-planning-guide#vm_types) with a number of vCPUs that is closest to the number you need while still being larger. After deployment is complete, modify the number of vCPUs and the amount of memory (/solutions/sap/docs/modifying_vm_configurations).
subnetwork	String	The name of the subnetwork you created in a previous step. If you are deploying to a shared VPC, specify this value as [SHAREDVPC_PROJECT]/[SUBNETWORK] . For example, myproject/network1.
linuxImage	String	The name of the Linux operating- system image or image family that you are using with SAP HANA. To specify an image family, add the prefix <code>family/</code> to the family name. For example, <code>family/sles-12-sp3-sap</code> . To specify a specific image, specify only the image name. For the list of available image families, see the Images (https://console.cloud.google.com/compute/images) page in the Cloud console.
linuxImageProject	String	The Google Cloud project that contains the image you are going to use. This project might be your own project or a Google Cloud image project. For SLES, specify <code>suse-sap-cloud</code> . For a list of GCP image projects, see the Images (/compute/docs/images) page in the Compute Engine documentation.
sap_hana_deployment_bucket	String	The name of the GCP storage bucket in your project that contains the SAP HANA installation files that you uploaded in a previous step.
sap_hana_sid	String	The SAP HANA system ID. The ID must consist of three alphanumeric characters and begin with a letter. All letters must be uppercase.

Property	Data type	Description
sap_hana_instance_number	Integer	The instance number, 0 to 99, of the SAP HANA system. The default is 0.
sap_hana_sidadm_password	String	The password for the operating system administrator. Passwords must be at least eight characters and include at least one uppercase letter, one lowercase letter, and one number.
sap_hana_system_password	String	The password for the database superuser. Passwords must be at least 8 characters and include at least one uppercase letter, one lowercase letter, and one number.
sap_hana_scaleout_nodes	Integer	The number of additional SAP HANA worker hosts that you need. Specify 0, because scaleout hosts are not supported in high availability configurations currently.
sap_vip	String	The floating IP address that is always assigned to the active SAP HANA instance. The IP address must be within the range of IP addresses that are assigned to your subnetwork.

The following example shows a completed configuration file, which directs the Deployment Manager to deploy a high-availability cluster with the primary SAP HANA system installed in the us-west1-a zone and the secondary SAP HANA system installed in the us-west1-b zone. Both systems will be installed on n1-highmem-96 VMs that are running the SLES 12 SP2 operating system.

7. Create the instances:

The above command invokes the Deployment Manager, which deploys the VMs, downloads the SAP HANA software from your storage bucket, and installs SAP HANA, all according to the specifications in your template.yaml file. The process takes approximately 10 to 15 minutes to complete. To check the progress of your deployment, follow the steps in the next section.

Verifying an SAP HANA HA cluster involves several different procedures:

- Checking the Stackdriver logs
- Checking the configuration of the VM and the SAP HANA installation
- Checking the SAP HANA system using SAP HANA Studio
- Performing a failover test

1. Open Stackdriver Logging to check for errors and monitor the progress of the installation.

★ **Note:** You might incur costs when completing this step in Stackdriver. For more information, see [Stackdriver pricing \(/stackdriver/pricing_v2\)](#).

[Go to Stackdriver Logging \(https://console.cloud.google.com/logs/viewer\)](https://console.cloud.google.com/logs/viewer)

2. On the Resources tab, select **Global** as your logging resource.

- If "INSTANCE DEPLOYMENT COMPLETE" is displayed, Deployment Manager processing is complete and you can proceed to the next step.

- If you see a quota error:
 - a. On the IAM & admin [Quotas](https://console.cloud.google.com/iam-admin/quotas) (https://console.cloud.google.com/iam-admin/quotas) page, increase any of your quotas that do not meet the SAP HANA requirements that are listed in the [SAP HANA Planning Guide](/solutions/sap/docs/sap-hana-planning-guide#quotas) (/solutions/sap/docs/sap-hana-planning-guide#quotas).
 - b. On the Deployment Manager [Deployments](https://console.cloud.google.com/dm/deployments) (https://console.cloud.google.com/dm/deployments) page, delete the deployment to clean up the VMs and persistent disks from the failed installation.
 - c. Rerun the Deployment Manager.

The screenshot shows the Stackdriver Logging interface. The left sidebar has 'Stackdriver Logging' selected. The main area shows a filter for 'Global' and 'All logs'. The log entries are as follows:

Timestamp	Source	Message
2018-05-08 09:35:09.526 PDT	example-ha-vm1	Deployment "Cluster: Adding STONITH devi..."
2018-05-08 09:35:11.125 PDT	example-ha-vm1	Deployment "Cluster: Adding virtual IP"
2018-05-08 09:35:12.718 PDT	example-ha-vm1	Deployment "Cluster: Configuring bootstr..."
2018-05-08 09:35:14.981 PDT	example-ha-vm1	Deployment "Cluster: Adding HANA nodes"
2018-05-08 09:35:16.398 PDT	example-ha-vm1	Deployment "INSTANCE DEPLOYMENT COMPLETE"

1. After the SAP HANA system deploys without errors, connect to each VM by using SSH. From the Compute Engine [VM instances page](https://console.cloud.google.com/compute/instances) (https://console.cloud.google.com/compute/instances), you can click the SSH button for each VM instance, or you can use your preferred SSH method.

The screenshot shows the 'VM instances' page in Google Cloud. The table below lists the instances:

Name	Zone	Internal IP	External IP	Connect
<input type="checkbox"/> <input checked="" type="checkbox"/> example-ha-vm1	us-central1-c	10.1.0.2	35.202.24.22	SSH
<input type="checkbox"/> <input checked="" type="checkbox"/> example-ha-vm2	us-central1-f	10.1.0.3	35.188.90.85	SSH
<input type="checkbox"/> <input checked="" type="checkbox"/> example-nat-gway-4-ha-cluster	us-central1-c	10.1.0.4	35.224.211.1	SSH

2. Change to the root user.

3. At the command prompt, enter `df -h`. Ensure that you see output that includes the `/hana` directories, such as `/hana/data`.

```
example-ha-vm1:~ # df -h
Filesystem                Size      Used Avail Use% Mounted on
devtmpfs                  308G         0   308G  0% /dev
tmpfs                     308G        54M   308G  1% /dev/shm
tmpfs                     308G        66M   308G  1% /run
tmpfs                     308G         0   308G  0% /sys/fs/cgroup
/dev/sda1                  30G        2.2G    26G  8% /
/dev/mapper/vg_hana-shared 614G        34G   581G  6% /hana/shared
/dev/mapper/vg_hana-data   938G        6.7G   931G  1% /hana/data
/dev/mapper/vg_hana-log    384G        5.2G   379G  2% /hana/log
/dev/mapper/vg_hana-sap    32G        233M    32G  1% /usr/sap
/dev/mapper/vg_hanabackup-backup 1.3T        5.0G   1.3T  1% /hanabackup
```

4. Check the status of the new cluster by issuing the following command:

You should see results similar to the following the example, in which `example-ha-vm1` is the master and `example-ha-vm2` is the slave:

```
example-ha-vm1:~ # crm status
Stack: corosync
Current DC: example-ha-vm1 (version 1.1.16-4.8-77ea74d) - partition with quorum
Last updated: Thu May 17 21:47:34 2018
Last change: Thu May 17 21:47:30 2018 by root via crm_attribute on example-ha-vm1

2 nodes configured
8 resources configured

Online: [ example-ha-vm1 example-ha-vm2 ]

Full list of resources:

STONITH-example-ha-vm1 (stonith:external/gcpstonith): Started example-ha-vm2
STONITH-example-ha-vm2 (stonith:external/gcpstonith): Started example-ha-vm1
Resource Group: g-vip
  rsc_vip_int (ocf::heartbeat:IPaddr2): Started example-ha-vm1
  rsc_vip_gcp (ocf::gcp:alias): Started example-ha-vm1
Clone Set: cln_SAPHanaTopology_HA1_HDB00 [rsc_SAPHanaTopology_HA1_HDB00]
  Started: [ example-ha-vm1 example-ha-vm2 ]
Master/Slave Set: msl_SAPHana_HA1_HDB00 [rsc_SAPHana_HA1_HDB00]
  Masters: [ example-ha-vm1 ]
  Slaves: [ example-ha-vm2 ]
```

5. Change to the SAP admin user by replacing **[SID]** in the following command with the [SID] value that you specified in the configuration file template.

6. Ensure that the SAP HANA services, such as `hdbnameserver`, `hdbindexserver`, and others, are running on the instance by entering the following command:

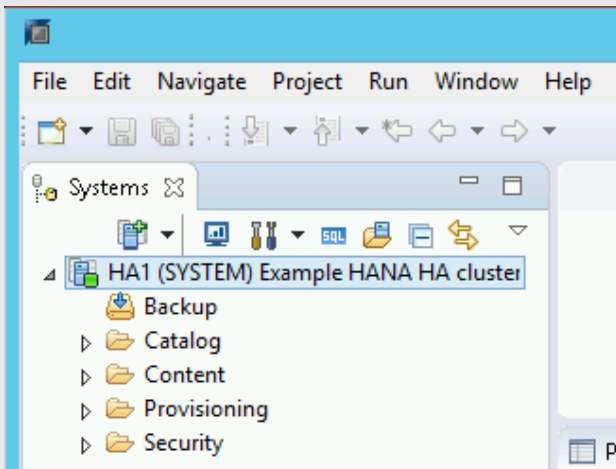
1. Connect to the HANA system by using SAP HANA Studio. When defining the connection, specify the following values:

- On the Specify System panel, specify the floating IP address as the Host Name.
- On the Connection Properties panel, for database user authentication, specify the database superuser name and the password that you specified for the `sap_hana_system_password` property in the `template.yaml` file.

For information from SAP about installing SAP HANA Studio, see [SAP HANA Studio Installation and Update Guide](#)

(<https://help.sap.com/viewer/a2a49126a5c546a9864aae22c05c3d0e/2.0.02/en-US>).

2. After SAP HANA Studio is connected to your HANA HA system, display the system overview by double-clicking the system name in the navigation pane on the left side of the window.



3. Under General Information on the Overview tab, confirm that:

- The Operational Status shows "All services started".
- The System Replication Status shows "All services are active and in sync".

The screenshot shows the SAP HANA Overview tab for a system named 'HA1 (SYSTEM) Example HANA HA cluster' with version '10.1.0.24 00'. The 'General Information' section is expanded, showing the following details:

- Operational Status:** All services started (indicated by a green checkmark)
- System Usage:** Custom System
- Start Time of First Started Service:** May 15, 2018 6:59:11 PM
- Start Time of Most Recently Started Service:** May 15, 2018 6:59:33 PM
- System Replication Status:** All services are active and in sync (indicated by a green checkmark)

4. Confirm the replication mode by clicking the **System Replication Status** link under General Information. Synchronous replication is indicated by SYNCMEM in the REPLICATION_MODE column on the System Replication tab.

The screenshot shows the 'System Replication' tab in the SAP HANA console. It displays a table with the following data:

AB	HOST	AB	SECONDARY_HOST	AB	REPLICATION_MODE	AB	REPLICATION_STATUS
	example-ha-vm1		example-ha-vm2		SYNCMEM		ACTIVE
	example-ha-vm1		example-ha-vm2		SYNCMEM		ACTIVE

If any of the validation steps show that the installation failed:

1. Resolve the errors.
2. Delete the deployment from the [Deployments](https://console.cloud.google.com/dm/deployments) (https://console.cloud.google.com/dm/deployments) page.
3. Recreate the instances, as described in the last step of [the previous section](#) (#creating_a_high-availability_linux_cluster_with_sap_hana_installed).

To perform a failover test:

1. Connect to the primary VM by using SSH. You can connect from the Compute Engine [VM instances page](https://console.cloud.google.com/compute/instances) (<https://console.cloud.google.com/compute/instances>) by clicking the SSH button for each VM instance, or you can use your preferred SSH method.

2. Change to the root user.

3. At the command prompt, enter the following command:

The `ifconfig eth0 down` command triggers a failover by severing communications with the primary host.

4. Follow the progress of the failover in Stackdriver:

[Go to Stackdriver logging](https://console.cloud.google.com/logs/viewer) (<https://console.cloud.google.com/logs/viewer>)

The following example shows the log entries for a successful failover:

```
2018-05-24 13:07:12.841 PDT example-ha-vm1 rsc_vip_gcp "example-ha-vm1 has the correct IP address attached"
2018-05-24 13:07:48.967 PDT example-ha-vm2 gcp:stonith "Issuing reset of example-ha-vm1 in zone us-central1-c"
2018-05-24 13:08:57.453 PDT example-ha-vm2 gcp:stonith "Reset of example-ha-vm1 in zone us-central1-c complete"
2018-05-24 13:09:01.784 PDT example-ha-vm2 rsc_vip_gcp "Checking to see if example-ha-vm1 owns 10.1.0.24/32"
2018-05-24 13:09:02.407 PDT example-ha-vm2 rsc_vip_gcp "10.1.0.24/32 is attached to example-ha-vm1 - Removing all alias IP addresses from example-ha-vm1"
2018-05-24 13:09:12.021 PDT example-ha-vm2 rsc_vip_gcp "Adding 10.1.0.24/32 to example-ha-vm2"
2018-05-24 13:09:22.190 PDT example-ha-vm2 rsc_vip_gcp "Finished adding 10.1.0.24/32 to example-ha-vm2"
2018-05-24 13:09:35.874 PDT example-ha-vm2 rsc_vip_gcp "example-ha-vm2 has the correct IP address attached"
```

5. Reconnect to either host using SSH and change to the root user.

6. Enter `'crm status'` to confirm that the primary host is now active on the VM that used to contain the secondary host. Automatic restart is enabled in the cluster, so the stopped host will restart and assume the role of secondary host, as shown in the following screenshot.

```

example-ha-vm1:~ # crm status
Stack: corosync
Current DC: example-ha-vm2 (version 1.1.15-21.1-e174ec8) - partition with quorum
Last updated: Thu May 24 20:23:55 2018
Last change: Thu May 24 20:23:54 2018 by root via crm_attribute on example-ha-vm2

2 nodes configured
8 resources configured

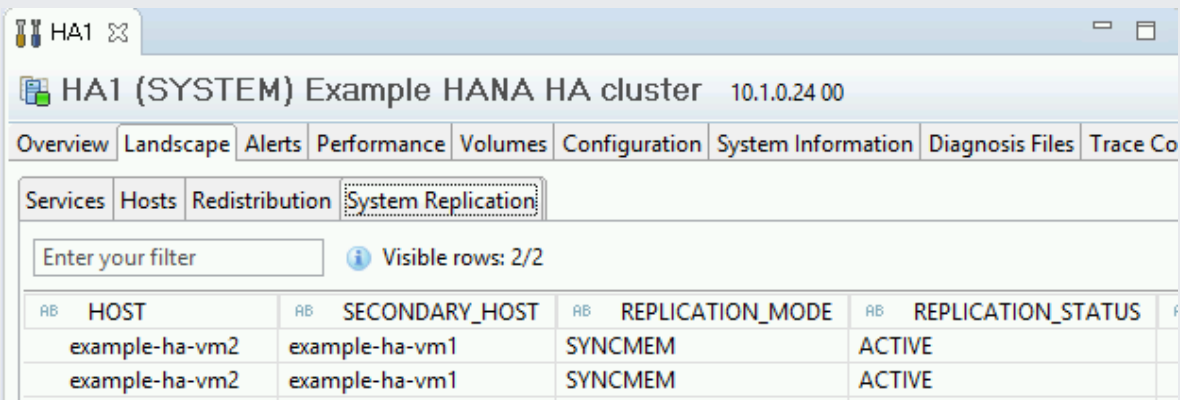
Online: [ example-ha-vm1 example-ha-vm2 ]

Full list of resources:

STONITH-example-ha-vm1 (stonith:external/gcpstonith): Started example-ha-vm2
STONITH-example-ha-vm2 (stonith:external/gcpstonith): Started example-ha-vm1
Resource Group: g-vip
  rsc_vip_int      (ocf::heartbeat:IPaddr2): Started example-ha-vm2
  rsc_vip_gcp      (ocf::gcp:alias): Started example-ha-vm2
Clone Set: cln_SAPHanaTopology_HA1_HDB00 [rsc_SAPHanaTopology_HA1_HDB00]
  Started: [ example-ha-vm1 example-ha-vm2 ]
Master/Slave Set: msl_SAPHana_HA1_HDB00 [rsc_SAPHana_HA1_HDB00]
  Masters: [ example-ha-vm2 ]
  Slaves: [ example-ha-vm1 ]

```

- In SAP HANA Studio, confirm that you are still connected to the system by double-clicking the system entry in the navigation pane to refresh the system information.
- Click the **System Replication Status** link to confirm that the primary and secondary hosts have switched hosts and are active.



RB	HOST	RB	SECONDARY_HOST	RB	REPLICATION_MODE	RB	REPLICATION_STATUS
	example-ha-vm2		example-ha-vm1		SYNCMEM		ACTIVE
	example-ha-vm2		example-ha-vm1		SYNCMEM		ACTIVE

If you created a NAT gateway, complete the following steps.

do not delete the external IP address from the VM instances that are running SAP HANA until you have completed cr network and a NAT gateway. After the external IP addresses are deleted, you can access the VM instances only thro T gateway.

1. Add tags to all instances, including the worker hosts:

2. Delete external IPs:

Optionally, you can set up Google's monitoring agent for SAP HANA, which collects metrics from SAP HANA and sends them to [Stackdriver Monitoring \(/monitoring/docs/\)](/monitoring/docs/). Stackdriver Monitoring allows you to create dashboards for your metrics, set up custom alerts based on metric thresholds, and more.

To monitor an HA cluster, you can install the monitoring agent on either a VM instance outside of the cluster or on each VM instance in the cluster.

If you install the monitoring agent on a VM instance outside of the HA cluster, you specify the floating IP address of the cluster as the IP address of the host instance to monitor. If you install the monitoring agent on each VM in the cluster, when configuring each monitoring agent, you specify the local IP address of the VM instance that is hosting the monitoring agent.

For more information on setting up and configuring Google's monitoring agent for SAP HANA, see the [SAP HANA Monitoring Agent User Guide \(/solutions/sap/docs/sap-hana-monitoring-agent-user-guide\)](/solutions/sap/docs/sap-hana-monitoring-agent-user-guide/).

Note that because these instructions don't use an external IP address for SAP HANA, you can only connect to the SAP HANA instances through the bastion instance using SSH or through the Windows server through SAP HANA Studio.

- To connect to SAP HANA through the bastion instance, connect to the bastion host, and then to the SAP HANA instance(s) by using an SSH client of your choice.
- To connect to the SAP HANA database through SAP HANA Studio, use a remote desktop client to connect to the Windows Server instance. After connection, manually install SAP HANA Studio (https://help.sap.com/hana/SAP_HANA_Studio_Installation_Update_Guide_en.pdf) and access your SAP HANA database.

Before using your SAP HANA instance, we recommend that you perform the following post-deployment steps. For more information, see SAP HANA Installation and Update Guide (http://help.sap.com/hana/SAP_HANA_Server_Installation_Guide_en.pdf).

1. Update the SAP HANA software with the latest patches.
 2. Install any additional components such as Application Function Libraries (AFL) or Smart Data Access (SDA).
 3. Configure and backup your new SAP HANA database. For more information, see the SAP HANA operations guide (/solutions/sap/docs/sap-hana-operations-guide#backup_and_recovery).
- For more information about VM administration of and monitoring, see the SAP HANA Operations Guide (</solutions/sap/docs/sap-hana-operations-guide>).