**Task 1. Create the VM instance**

Create a VPC network with some firewall rules and a VM instance that has no external IP address, and connect to the instance using an IAP tunnel.

Create a VPC network and firewall rules

First, create a VPC network for the VM instance and a firewall rule to allow SSH access.

1. In the Cloud Console, on the **Navigation menu** (Navigation menu icon), click **VPC network > VPC networks**.
2. Click **Create VPC Network**.
3. For **Name**, type **privatenet**.
4. For **Subnet creation mode**, click **Custom**.
5. In **New Subnet** specify the following, and leave the remaining settings as their defaults:

|  |  |
| --- | --- |
| **Property** | **Value (type value or select option as specified)** |
| Name | privatenet-us |
| Region | us-central1 |
| IPv4 address range | 10.130.0.0/20 |

**Note:** Don't enable **Private Google access** yet!

1. Click **Done**.
2. Click **Create** and wait for the network to be created.
3. In the left pane, click **Firewall**.
4. Click **Create Firewall Rule**.
5. Specify the following, and leave the remaining settings as their defaults:

|  |  |
| --- | --- |
| **Property** | **Value (type value or select option as specified)** |
| Name | privatenet-allow-ssh |
| Network | privatenet |
| Targets | All instances in the network |
| Source filter | IPv4 ranges |
| Source IPv4 ranges | 35.235.240.0/20 |
| Protocols and ports | Specified protocols and ports |

1. For **tcp**, click the checkbox and specify port **22**.
2. Click **Create**.

**Note:** In order to connect to your private instance using SSH, you need to open an appropriate port on the firewall. [IAP connections](https://cloud.google.com/iap/docs/using-tcp-forwarding" \t "\\_blank) come from a specific set of IP addresses (**35.235.240.0/20**). Therefore, you can limit the rule to this CIDR range.

Create the VM instance with no public IP address

1. In the Cloud Console, on the **Navigation menu** (Navigation menu icon), click **Compute Engine > VM instances**.
2. Click **Create Instance**.
3. Specify the following, and leave the remaining settings as their defaults:

|  |  |
| --- | --- |
| **Property** | **Value (type value or select option as specified)** |
| Name | vm-internal |
| Region | us-central1 |
| Zone | us-central1-c |
| Series | N1 |
| Machine type | n1-standard-1 (1vCPU, 3.75 GB memory) |
| Boot disk | Debian GNU/Linux 11 (bullseye) |

1. Click **Advanced options**.
2. Click **Networking**.
3. In **Network interfaces**, click the **default** network to edit it.
4. Specify the following, and leave the remaining settings as their defaults:

|  |  |
| --- | --- |
| **Property** | **Value (type value or select option as specified)** |
| Network | privatenet |
| Subnetwork | privatenet-us |
| External IPv4 address | None |

**Note:** The default setting for a VM instance is to have an ephemeral external IP address. This behavior can be changed with a policy constraint at the organization or project level. To learn more about controlling external IP addresses on VM instances, refer to the [external IP address documentation](https://cloud.google.com/compute/docs/ip-addresses/reserve-static-external-ip-address#disableexternalip).

1. Click **Done**.
2. Click **Create**, and wait for the VM instance to be created.
3. On the **VM instances** page, verify that the **External IP** of **vm-internal** is **None**.

Click **Check my progress** to verify the objective.

Create the VM instance

Check my progress

SSH to vm-internal to test the IAP tunnel

1. In the Cloud Console, click **Activate Cloud Shell** (Cloud Shell).
2. If prompted, click **Continue**.
3. To connect to **vm-internal**, run the following command:

gcloud compute ssh vm-internal --zone us-central1-c --tunnel-through-iap

Copied!

content\_copy

1. If prompted click **Authorize**
2. If prompted to continue, type **Y**.
3. When prompted for a passphrase, press **ENTER**.
4. When prompted for the same passphrase, press **ENTER**.

Did the command prompt change to @vm-internal?



True



False

1. To test the external connectivity of **vm-internal**, run the following command:

ping -c 2 www.google.com

Copied!

content\_copy

This should not work because **vm-internal** has no external IP address!

1. Wait for the ping command to complete.
2. To return to your Cloud Shell instance, run the following command:

exit

Copied!

content\_copy

**Note:** When instances do not have external IP addresses, they can only be reached by other instances on the network via a managed VPN gateway or via a Cloud IAP tunnel. Cloud IAP enables context-aware access to VMs via SSH and RDP without bastion hosts. To learn more about this, see the blog post [Cloud IAP enables context-aware access to VMs via SSH and RDP without bastion hosts](https://cloud.google.com/blog/products/identity-security/cloud-iap-enables-context-aware-access-to-vms-via-ssh-and-rdp-without-bastion-hosts" \t "_blank).

**Task 2. Enable Private Google Access**

VM instances that have no external IP addresses can use Private Google Access to reach external IP addresses of Google APIs and services. By default, Private Google Access is disabled on a VPC network.

Create a Cloud Storage bucket

Create a Cloud Storage bucket to test access to Google APIs and services.

1. In the Cloud Console, on the **Navigation menu** (Navigation menu icon), click **Cloud Storage > Buckets**.
2. Click **Create**.
3. Specify the following, and leave the remaining settings as their defaults:

|  |  |
| --- | --- |
| **Property** | **Value (type value or select option as specified)** |
| Name | *Enter a globally unique name* |
| Location type | Multi-region |

1. Click **Create**. If prompted to enable public access prevention, ensure it is checked and cick **Confirm**. Note the name of your storage bucket.
2. Store the name of your bucket in an environment variable:

export MY\_BUCKET=<enter your bucket name here>

Copied!

content\_copy

1. Verify it with echo:

echo $MY\_BUCKET

Copied!

content\_copy

Copy an image file into your bucket

Copy an image from a public Cloud Storage bucket to your own bucket.

1. In Cloud Shell, run the following command:

gsutil cp gs://cloud-training/gcpnet/private/access.svg gs://$MY\_BUCKET

Copied!

content\_copy

1. In the Cloud Console, click your bucket name to verify that the image was copied.

You can click on the name of the image in the Cloud Console to view an example of how Private Google Access is implemented.

Access the image from your VM instance

Currently, which of your VM instances can access the image from your bucket?



vm-internal



Cloud Shell

Submit

1. In Cloud Shell, to try to copy the image from your bucket, run the following command:

gsutil cp gs://$MY\_BUCKET/\*.svg .

Copied!

content\_copy

This should work because Cloud Shell has an external IP address!

1. To connect to **vm-internal**, run the following command:

gcloud compute ssh vm-internal --zone us-central1-c --tunnel-through-iap

Copied!

content\_copy

1. If prompted, type **Y** to continue.
2. Store the name of your bucket in an environment variable:

export MY\_BUCKET=<enter your bucket name here>

Copied!

content\_copy

1. Verify it with echo:

echo $MY\_BUCKET

Copied!

content\_copy

1. Try to copy the image to **vm-internal**, run the following command:

gsutil cp gs://$MY\_BUCKET/\*.svg .

Copied!

content\_copy

This should not work: **vm-internal** can only send traffic within the VPC network because Private Google Access is disabled (by default).

1. Press **Ctrl+Z** to stop the request.

Enable Private Google Access

Private Google Access is enabled at the subnet level. When it is enabled, instances in the subnet that only have private IP addresses can send traffic to Google APIs and services through the default route (0.0.0.0/0) with a next hop to the default internet gateway.

1. In the Cloud Console, on the **Navigation menu** (Navigation menu icon), click **VPC network > VPC networks**.
2. Click **privatenet** to open the network.
3. Click **privatenet-us** to open the subnet.
4. Click **Edit**.
5. For **Private Google access**, select **On**.
6. Click **Save**.

Click **Check my progress** to verify the objective.

Create a Cloud Storage bucket and Enable Private Google Access

Check my progress

**Note:** Enabling Private Google Access is as simple as selecting **On** within the subnet!

1. In **Cloud Shell** for **vm-internal**, to try to copy the image to **vm-internal**, run the following command, replacing <your\_bucket\_name> with the name of your bucket:

gsutil cp gs://$MY\_BUCKET/\*.svg .

Copied!

content\_copy

This should work because **vm-internal**'s subnet has **Private Google Access** enabled!

1. To return to your Cloud Shell instance, run the following command:

exit

Copied!

content\_copy

**Note:**To view the eligible APIs and services that you can use with Private Google Access, see supported services in the [Private access options for services Guide](https://cloud.google.com/vpc/docs/private-access-options#pga-supported).

**Task 3. Configure a Cloud NAT gateway**

Although **vm-internal** can now access certain Google APIs and services without an external IP address, the instance cannot access the internet for updates and patches. Configure a Cloud NAT gateway, which allows **vm-internal** to reach the internet.

Try to update the VM instances

1. In **Cloud Shell**, to try to re-synchronize the package index, run the following:

sudo apt-get update

Copied!

content\_copy

The output should finish like this (**example output**):

...

Reading package lists... Done

This should work because **Cloud Shell** has an external IP address!

1. To connect to **vm-internal**, run the following command:

gcloud compute ssh vm-internal --zone us-central1-c --tunnel-through-iap

Copied!

content\_copy

1. If prompted, type **Y** to continue.
2. To try to re-synchronize the package index of **vm-internal**, run the following command:

sudo apt-get update

Copied!

content\_copy

This should only work for Google Cloud packages because **vm-internal** only has access to Google APIs and services!

1. Press **Ctrl+C** to stop the request.

Configure a Cloud NAT gateway

Cloud NAT is a regional resource. You can configure it to allow traffic from all ranges of all subnets in a region, from specific subnets in the region only, or from specific primary and secondary CIDR ranges only.

1. In the Cloud Console, on the **Navigation menu** (Navigation menu icon), click **Network services > Cloud NAT**.
2. Click **Get started** to configure a NAT gateway.
3. Specify the following:

|  |  |
| --- | --- |
| **Property** | **Value (type value or select option as specified)** |
| Gateway name | nat-config |
| Network | privatenet |
| Region | us-central1 |

1. For **Cloud Router**, select **Create new router**.
2. For **Name**, type **nat-router**
3. Click **Create**.

**Note:** The NAT mapping section allows you to choose the subnets to map to the NAT gateway. You can also manually assign static IP addresses that should be used when performing NAT. Do not change the NAT mapping configuration in this lab.

1. Click **Create**.
2. Wait for the gateway's status to change to **Running**.

Click **Check my progress** to verify the objective.

Configure a Cloud NAT gateway

Check my progress

Verify the Cloud NAT gateway

It may take up to 3 minutes for the NAT configuration to propagate to the VM, so wait at least a minute before trying to access the internet again.

1. In **Cloud Shell** for **vm-internal**, to try to re-synchronize the package index of **vm-internal**, run the following command:

sudo apt-get update

Copied!

content\_copy

The output should finish like this (**example output**):

...

Reading package lists... Done

This should work because **vm-internal** is using the NAT gateway!

1. To return to your Cloud Shell instance, run the following command:

exit

Copied!

content\_copy

**Note:** The Cloud NAT gateway implements outbound NAT, but not inbound NAT. In other words, hosts outside of your VPC network can only respond to connections initiated by your instances; they cannot initiate their own, new connections to your instances via NAT.

**Task 4. Configure and view logs with Cloud NAT Logging**

[Cloud NAT logging](https://cloud.google.com/nat/docs/monitoring) allows you to log NAT connections and errors. When Cloud NAT logging is enabled, one log entry can be generated for each of the following scenarios:

* When a network connection using NAT is created.
* When a packet is dropped because no port was available for NAT.

You can opt to log both kinds of events, or just one or the other. Created logs are sent to Cloud Logging.

Enabling logging

If logging is enabled, all collected logs are sent to Cloud Logging by default. You can filter these so that only certain logs are sent.

You can also specify these values when you create a NAT gateway or by editing one after it has been created. The following directions show how to enable logging for an existing NAT gateway.

1. In the GCP Console, on the **Navigation menu** (Navigation menu icon), click **Network services** > **Cloud NAT**.
2. Click on the nat-config gateway and then click **Edit**.
3. Click the **Advanced configurations** dropdown to open that section.
4. Under **Stackdriver logging**, select **Translation and errors** and then click **Save**.

NAT logging in Cloud Operations

Now that you have set up Cloud NAT logging for the nat-config gateway, let's find out where we can view our logs.

1. Click on nat-config to expose its details. Then click on the **Logs** tab. Then click the link to **Cloud Logging**.
2. This will open a new tab with **Operations Logging**.

You will see that there aren't any logs yet—that's because we just enabled this feature for the gateway. **Keep this tab open** and return to your other GCP Console tab.

Generating logs

As a reminder, Cloud NAT logs are generated for the following sequences:

* When a network connection using NAT is created.
* When a packet is dropped because no port was available for NAT.

Let's connect the host to the internal VM again to see if any logs are generated.

1. In **Cloud Shell** for **vm-internal**, to try to re-synchronize the package index of **vm-internal**, run the following command:

gcloud compute ssh vm-internal --zone us-central1-c --tunnel-through-iap

Copied!

content\_copy

1. If prompted, type **Y** to continue.
2. Try to re-synchronize the package index of **vm-internal** by running the following:

sudo apt-get update

Copied!

content\_copy

The output should look like this (**example output**):

...

Reading package lists... Done

1. To return to your Cloud Shell instance, run the following command:

exit

Copied!

content\_copy

Let's see if opening up this connection revealed anything new in our logs.

Viewing Logs

* Return to your Operations Logging tab and in the navigation menu, click **Logs Explorer**.

You should see two new logs that were generated after connecting to the internal VM.

**Note:** You may need to wait for a few minutes. If you are still unable to see the logs, repeat step 1 to step 4, from the **Generating logs** section, and then refresh the logging page.

As we see, the logs give us details on the VPC network we connected to and the connection method we used. Feel free to expand different labels and details.