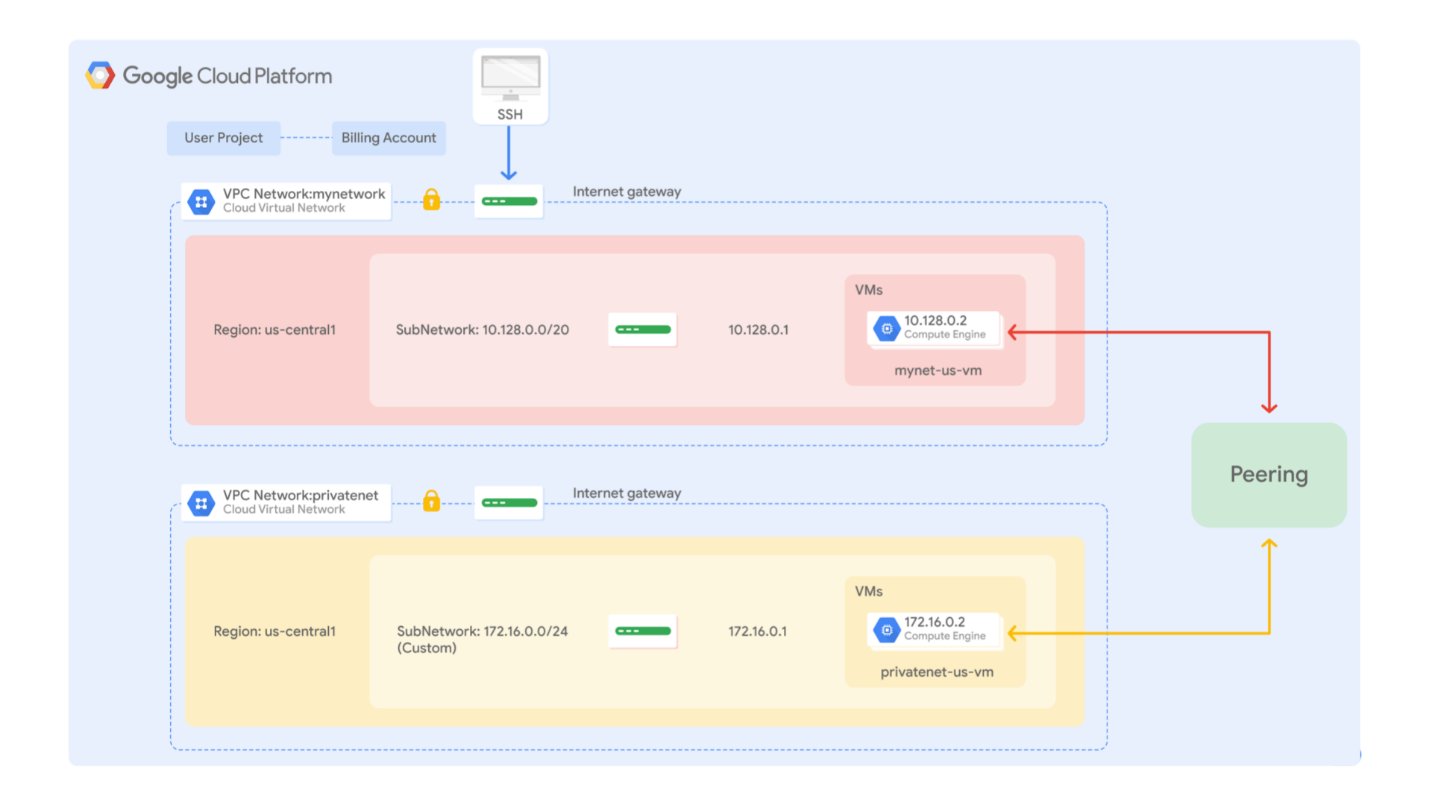
Configuring VPC Network Peering

1 hourFree

Rate Lab

**Overview**

In this lab, you configure VPC network peering between two networks. Then you verify private communication between two VMs in those networks, as illustrated in this diagram:



VPC network peering allows you to build SaaS (Software as a service) ecosystems in Google Cloud, which makes services available privately across different VPC networks within and across organizations. This allows workloads to communicate in private RFC 1918 space.

VPC network peering gives you several advantages over using external IP addresses or VPNs to connect networks, including:

* **Network Latency:** Public IP networking results in higher latency than private networking.
* **Network Security:** Service owners do not need to have their services exposed to the public internet and deal with its associated risks.
* **Network Cost:** Google Cloud charges [egress bandwidth pricing](https://cloud.google.com/compute/pricing#internet_egress) for networks using external IPs to communicate, even if the traffic is within the same zone. If, however, the networks are peered, they can use internal IPs to communicate and save on those egress costs. [Regular network pricing](https://cloud.google.com/compute/pricing#network) still applies to all traffic.

Objectives

In this lab, you learn how to perform the following tasks:

* Explore connectivity between non-peered VPC networks
* Configure VPC network peering
* Verify private communication between peered VPC networks
* Delete VPC network peering

**Before you click the Start Lab button**

Read these instructions. Labs are timed and you cannot pause them. The timer, which starts when you click Start Lab, shows how long Cloud resources will be made available to you.

This Qwiklabs hands-on lab lets you do the lab activities yourself in a real cloud environment, not in a simulation or demo environment. It does so by giving you new, temporary credentials that you use to sign in and access the Google Cloud Platform for the duration of the lab.

**What you need**

To complete this lab, you need:

* Access to a standard internet browser (Chrome browser recommended).
* Time to complete the lab.

***Note:*** If you already have your own personal GCP account or project, do not use it for this lab.

**Task 1. Explore connectivity between non-peered VPC networks**

Each Google Cloud project starts with the **default** network. In addition, **mynetwork**, **privatenet**, and **managementnet** have been created for you along with firewall rules to allow ICMP-SSH-RDP traffic and four VM instances.

**Verify VPC network peering requirements**

In a peered VPC network, no subnet IP range can overlap with another subnet IP range. Therefore, verify that the CIDR blocks of the subnets of **mynetwork** and **privatenet** are non-overlapping.

1. In the Cloud Console, on the **Navigation menu** (Navigation menu), click **VPC network** > **VPC networks**.
2. Examine the **IP addresses ranges** for the subnets of **mynetwork**.

The subnets of **mynetwork** fit within the 10.128.0.0/9 CIDR block. As new Google Cloud regions become available, new subnets in those regions are automatically added to this auto mode network using an IP range from that block.

1. Examine the **IP addresses ranges** for the subnets of **privatenet**.

The subnets of **privatenet** fit within the 172.16.0.0/24 and 172.20.0.0/24 CIDR blocks. They do not overlap with the 10.128.0.0/9 CIDR block of **mynetwork**.

You can configure VPC network peering between **mynetwork** and **privatenet** because their subnets' CIDR blocks are non-overlapping.

**Explore the connectivity between mynetwork and privatenet**

Before configuring VPC network peering, explore the current connectivity between **mynetwork** and **privatenet**.

You should be able to ping the internal and external IP addresses of privatenet-us-vm from mynet-us-vm.

True

False

1. On the **Navigation menu** (Navigation menu), click **VPC network** > **VPC network peering**. Notice that there is no peering connection.

You will return to this page to configure the VPC network peering connections.

1. On the **Navigation menu**, click **VPC network** > **Routes**. Notice that none of the routes have a peering connection as the **Next hop**.

You will return to this page after configuring the VPC network peering connection.

1. On the **Navigation menu**, click **VPC network** > **Firewall Rules**. Notice the allow **SSH** and **ICMP** firewall rules for **mynetwork** and **privatenet**.

These firewall rules have been created for you.

1. On the **Navigation menu**, click **Compute Engine** > **VM instances**. Notice the **mynet-eu-vm**, **mynet-us-vm**, **privatenet-us-vm**, and **managementnet-us-vm** instances.

These VM instances have been created for you.

1. Note the internal and external IP addresses for **privatenet-us-vm**.
2. For **mynet-us-vm**, click **SSH** to launch a terminal and connect.
3. To test connectivity to **privatenet-us-vm**'s external IP, run the following command, replacing **privatenet-us-vm**'s external IP:

ping -c 3 <Enter privatenet-us-vm's external IP here>

This should work!

1. To test connectivity to **privatenet-us-vm**'s internal IP, run the following command, replacing **privatenet-us-vm**'s internal IP:

ping -c 3 <Enter privatenet-us-vm's internal IP here>

This should not work, as indicated by a 100% packet loss! But why?

1. On the VM instances page, click **Columns**, and then select **Network**.

The **mynet-us-vm** and **privatenet-us-vm** instances are in the same zone (us-central1-a) but in different VPC networks (**mynetwork** and **privatenet**). Because VPC network peering has not been configured between those networks, private communication fails between the instances of those networks.

1. Close the SSH terminal to **mynet-us-vm**:

exit

**Task 2. Configure VPC network peering**

VPC network peering can be configured for different VPC networks within and across organizations. Configure the following peering connections in this project:

* **peering-1-2**: Peer **mynetwork** with **privatenet**
* **peering-2-1**: Peer **privatenet** with **mynetwork**

Each side of a peering association is set up independently. Peering is active only when the configuration from both sides matches.

**Create peering 1-2**

Peer **mynetwork** with **privatenet**.

1. In the Cloud Console, on the **Navigation menu** (Navigation menu), click **VPC network** > **VPC network peering**.
2. Click **Create connection**.
3. Read through the requirements.

You won't need the **project ID** because you are connecting to a VPC network within the same project.

1. Click **Continue**.
2. Specify the following, and leave the remaining settings as their defaults:

|  |  |
| --- | --- |
| **Property** | **Value (type value or select option as specified)** |
| Name | peering-1-2 |
| Your VPC Network | mynetwork |
| VPC network name | privatenet |

If you peered with a VPC network in another project, you would need the **Network admin** or **Project owner/editor** IAM role for that project to create a peering connection.

1. Click **Create**. Wait for the **Status** to change to **Waiting for peer network to connect.**

Because the other side has not been configured, the networks are not yet peered.

**Create peering 2-1**

Peer **privatenet** with **mynetwork**.

1. In the Cloud Console, return to the **VPC network peering** page.
2. Click **Create peering connection**.
3. Click **Continue**.
4. Specify the following, and leave the remaining settings as their defaults:

|  |  |
| --- | --- |
| **Property** | **Value (type value or select option as specified)** |
| Name | peering-2-1 |
| Your VPC Network | privatenet |
| VPC network name | mynetwork |

1. Click **Create**. Wait for the **Status** of both connections to change to **Connected.**

As expected, peering is active when the configuration from both sides matches.

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

Configure VPC Network Peering

Check my progress

**Task 3. Verify private communication between peered VPC networks**

Verify private RFC 1918 connectivity across **mynetwork** and **privatenet**.

**Verify routes between networks**

Verify that routes have been established between **mynetwork** and **privatenet**.

* In the Cloud Console, on the **Navigation menu** (Navigation menu), click **VPC network** > **Routes**. Notice that there is a route for each subnet in **mynetwork** with **peering-1-2** as the **Next hop**. Similarly, notice that there is a route for each subnet in **privatenet** with **peering-2-1** as the **Next hop**.

These routes were automatically created with the VPC peering connection.

**Note** User-configured routes are not propagated across peered networks. If you configure a route in a network to a destination in a VPC network, that destination will not be reachable from a peered network.

**Ping from mynetwork to privatenet**

Try to ping the internal IP of **privatenet-us-vm** from **mynet-us-vm**.

1. On the **Navigation menu**, click **Compute Engine** > **VM instances**.
2. Note the internal IP address for **privatenet-us-vm**.
3. For **mynet-us-vm**, click **SSH** to launch a terminal and connect.
4. To test connectivity to **privatenet-us-vm**'s internal IP, run the following command, replacing **privatenet-us-vm**'s internal IP:

ping -c 3 <Enter privatenet-us-vm's internal IP here>

This should work because of the route that was established by the peering connection.

1. Close the SSH terminal to **mynet-us-vm**:

exit

**Ping from privatenet to mynetwork**

Similarly, try to ping the internal IP of **mynet-us-vm** from **privatenet-us-vm**.

1. Note the internal IP address for **mynet-us-vm**.
2. For **privatenet-us-vm**, click **SSH** to launch a terminal and connect.
3. To test connectivity to **mynet-us-vm**'s internal IP, run the following command, replacing **mynet-us-vm**'s internal IP:

ping -c 3 <Enter mynet-us-vm's internal IP here>

This should also work because of the route that was established by the peering connection.

1. To test Compute Engine DNS across peered networks, run the following command:

ping -c 3 mynet-us-vm

The output should look like this (**do not copy; this is example output**):

ping: mynet-us-vm: Name or service not known

Compute Engine internal DNS names created in a network are not accessible to peered networks. The IP address of the VM should be used to reach the VM instances in peered network.

1. Close the SSH terminal to **privatenet-us-vm**:

exit

Now that you've verified the VPC Peering connection, you could stop both instances to remove their external IP addresses. This helps secure your instances and reduces your networking costs. You can still SSH to an instance without a public IP address using [Cloud IAP tunnels.](https://cloud.google.com/iap/docs/using-tcp-forwarding#tunneling_with_ssh)

**Task 4. Delete the VPC Peering Connection**

Delete the VPC Peering connection and verify the deletion.

**Delete the peering connection**

Delete the **peering-1-2** connection.

The overall peering connection will terminate when you delete the peering-1-2 connection.

True

False

1. On the **Navigation menu**, click **VPC network** > **VPC network peering**.
2. Select the **peering-1-2** connection.
3. Click **Delete**.
4. Click **Delete** to confirm the deletion. When the connection is deleted, notice the **Peer network is disconnected** status for **peering-2-1**.

Deleting one side of the peering connection terminates the overall peering connection.

**Verify the peering deletion**

Verify that **routes** no longer exist for the peering connection and that there is no private RFC 1918 connectivity across **mynetwork** and **privatenet**.

1. On the **Navigation menu**, click **VPC network** > **Routes**. Notice that the VPC Peering routes have disappeared.

Deleting one side of the VPC Peering connection removes all peering routes.

1. On the **Navigation menu**, click **Compute Engine** > **VM instances**.
2. Note the internal IP address for **privatenet-us-vm**.
3. For **mynet-us-vm**, click **SSH** to launch a terminal and connect.
4. To test connectivity to **privatenet-us-vm**'s internal IP, run the following command, replacing **privatenet-us-vm**'s internal IP:

ping -c 3 <Enter privatenet-us-vm's internal IP here>

This should not work as indicated by a 100% packet loss!

1. Close the SSH terminal to **mynet-us-vm**:

exit

**Task 5. Review**

In this lab, you configured VPC network peering between two networks (**privatenet** and **mynetwork**). Then you verified private RFC 1918 connectivity across **mynetwork** and **privatenet** by pinging VMs on their internal IP addresses within those networks. Finally, you deleted one side of the VPC network peering connection to demonstrate that this removes private RFC 1918 connectivity across those networks.