**Multiple VPC Networks**

experimentLabschedule1 hour 10 minutesuniversal\_currency\_alt5 Creditsshow\_chartIntermediate

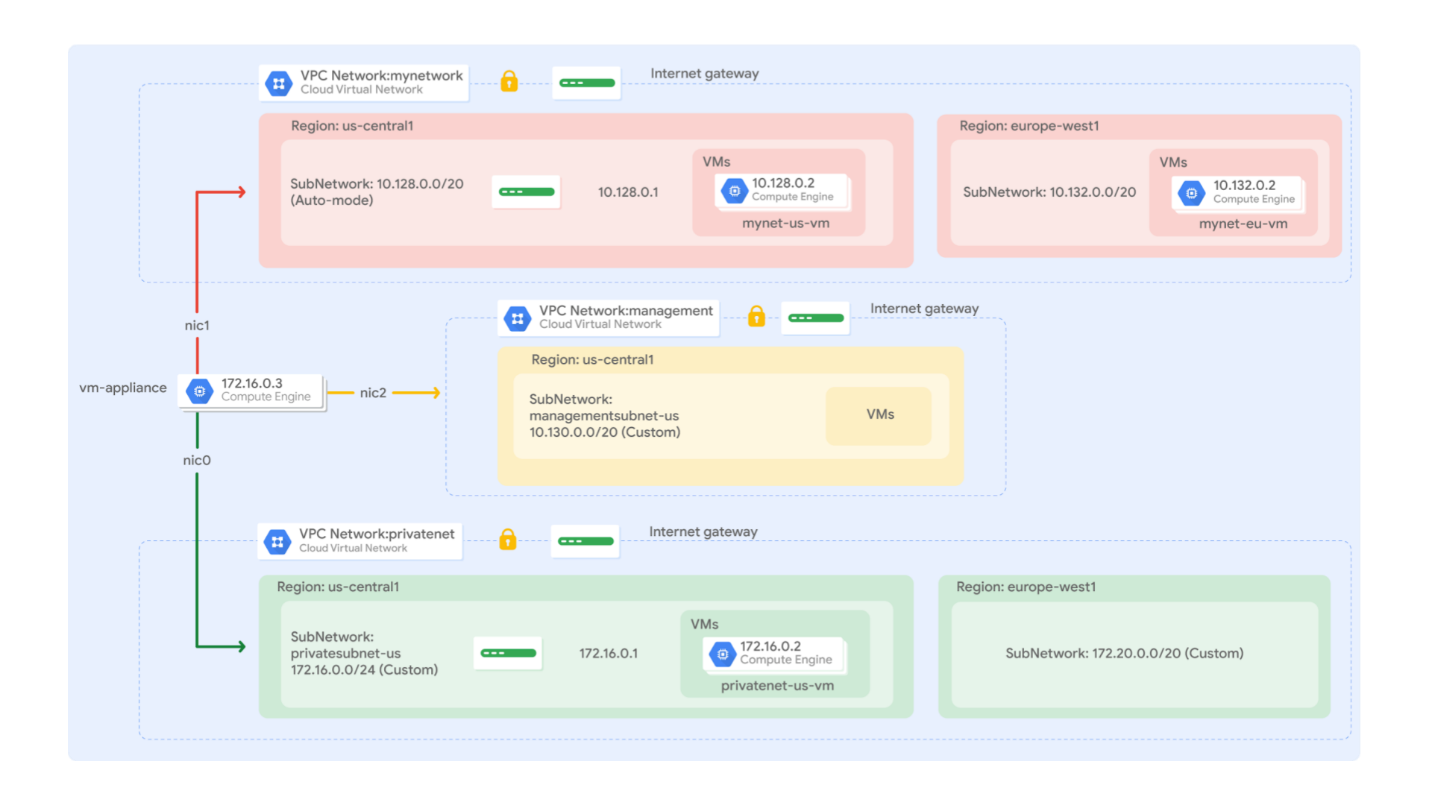
**GSP211**



**Overview**

Virtual Private Cloud (VPC) networks allow you to maintain isolated environments within a larger cloud structure, giving you granular control over data protection, network access, and application security.

In this lab you create several VPC networks and VM instances, then test connectivity across networks. Specifically, you create two custom mode networks (**managementnet** and **privatenet**) with firewall rules and VM instances as shown in this network diagram:



The **mynetwork** network with its firewall rules and two VM instances (**mynet-us-vm** and **mynet-eu-vm**) have already been created for you for this lab.

**Objectives**

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

* Create custom mode VPC networks with firewall rules
* Create VM instances using Compute Engine
* Explore the connectivity for VM instances across VPC networks
* Create a VM instance with multiple network interfaces

**Setup and requirements**

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 Google Cloud resources will be made available to you.

This 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 Google Cloud for the duration of the lab.

To complete this lab, you need:

* Access to a standard internet browser (Chrome browser recommended).

**Note:** Use an Incognito or private browser window to run this lab. This prevents any conflicts between your personal account and the Student account, which may cause extra charges incurred to your personal account.

* Time to complete the lab---remember, once you start, you cannot pause a lab.

**Note:** If you already have your own personal Google Cloud account or project, do not use it for this lab to avoid extra charges to your account.

How to start your lab and sign in to the Google Cloud console

1. Click the **Start Lab** button. If you need to pay for the lab, a pop-up opens for you to select your payment method. On the left is the **Lab Details** panel with the following:
   * The **Open Google Cloud console** button
   * Time remaining
   * The temporary credentials that you must use for this lab
   * Other information, if needed, to step through this lab
2. Click **Open Google Cloud console** (or right-click and select **Open Link in Incognito Window** if you are running the Chrome browser).

The lab spins up resources, and then opens another tab that shows the **Sign in** page.

***Tip:*** Arrange the tabs in separate windows, side-by-side.

**Note:**If you see the **Choose an account** dialog, click **Use Another Account**.

1. If necessary, copy the **Username** below and paste it into the **Sign in** dialog.

student-01-8fb0c6312b68@qwiklabs.net

Copied!

content\_copy

You can also find the **Username** in the **Lab Details** panel.

1. Click **Next**.
2. Copy the **Password** below and paste it into the **Welcome** dialog.

hrVzJEEPDTfN

Copied!

content\_copy

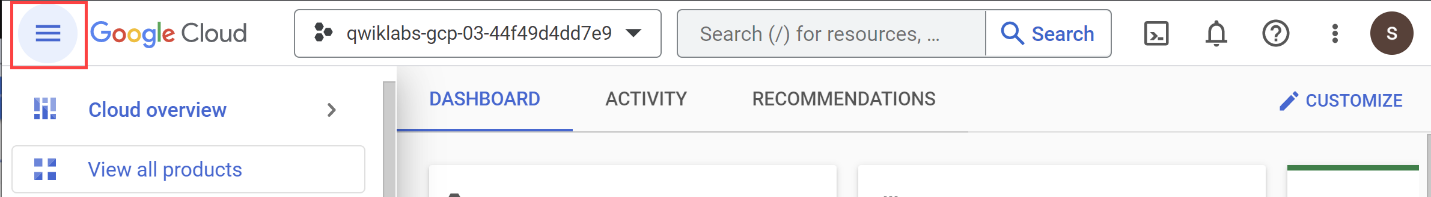
You can also find the **Password** in the **Lab Details** panel.

1. Click **Next**.

**Important:**You must use the credentials the lab provides you. Do not use your Google Cloud account credentials.**Note:**Using your own Google Cloud account for this lab may incur extra charges.

1. Click through the subsequent pages:
   * Accept the terms and conditions.
   * Do not add recovery options or two-factor authentication (because this is a temporary account).
   * Do not sign up for free trials.

After a few moments, the Google Cloud console opens in this tab.

**Note:** To view a menu with a list of Google Cloud products and services, click the **Navigation menu** at the top-left. 

Activate Cloud Shell

Cloud Shell is a virtual machine that is loaded with development tools. It offers a persistent 5GB home directory and runs on the Google Cloud. Cloud Shell provides command-line access to your Google Cloud resources.

1. Click **Activate Cloud Shell** Activate Cloud Shell icon at the top of the Google Cloud console.

When you are connected, you are already authenticated, and the project is set to your **Project\_ID**, qwiklabs-gcp-03-3e1859c26f37. The output contains a line that declares the **Project\_ID** for this session:

Your Cloud Platform project in this session is set to qwiklabs-gcp-03-3e1859c26f37

gcloud is the command-line tool for Google Cloud. It comes pre-installed on Cloud Shell and supports tab-completion.

1. (Optional) You can list the active account name with this command:

gcloud auth list

Copied!

content\_copy

1. Click **Authorize**.

**Output:**

ACTIVE: \*

ACCOUNT: student-01-8fb0c6312b68@qwiklabs.net

To set the active account, run:

$ gcloud config set account `ACCOUNT`

1. (Optional) You can list the project ID with this command:

gcloud config list project

Copied!

content\_copy

**Output:**

[core]

project = qwiklabs-gcp-03-3e1859c26f37

**Note:**For full documentation of gcloud, in Google Cloud, refer to [the gcloud CLI overview guide](https://cloud.google.com/sdk/gcloud).

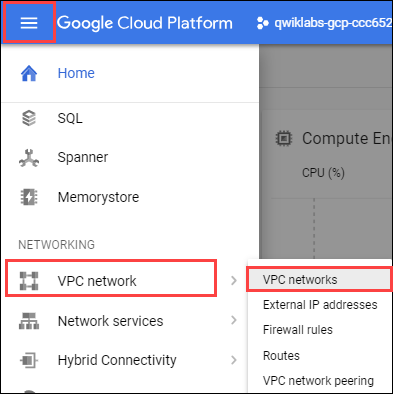
**Task 1. Create custom mode VPC networks with firewall rules**

Create two custom networks **managementnet** and **privatenet**, along with firewall rules to allow **SSH**, **ICMP**, and **RDP** ingress traffic.

Create the managementnet network

Create the **managementnet** network using the Cloud console.

1. In the Cloud console, navigate to **Navigation menu** (Navigation menu icon) > **VPC network** > **VPC networks**.



1. Notice the **default** and **mynetwork** networks with their subnets.

Each Google Cloud project starts with the **default** network. In addition, the **mynetwork** network has been premade as part of your network diagram.

1. Click **Create VPC Network**.
2. Set the **Name** to managementnet.
3. For **Subnet creation mode**, click **Custom**.
4. Set the following values, leave all other values at their defaults:

|  |  |
| --- | --- |
| **Property** | **Value (type value or select option as specified)** |
| Name | managementsubnet-us |
| Region | us-west1 |
| IPv4 range | 10.130.0.0/20 |

1. Click **Done**.
2. Click **EQUIVALENT COMMAND LINE**.

These commands illustrate that networks and subnets can be created using the Cloud Shell command line. You will create the **privatenet** network using these commands with similar parameters.

1. Click **Close**.
2. Click **Create**.

**Test Completed Task**

Click **Check my progress** to verify your performed task. If you have successfully created a managementnet network, you will see an assessment score.

Assessment Completed!

Create the managementnet network

Check my progress

*Assessment Completed!*

**Create the privatenet network**

Create the **privatenet** network using the Cloud Shell command line.

1. Run the following command to create the **privatenet** network:

gcloud compute networks create privatenet --subnet-mode=custom

Copied!

content\_copy

1. Run the following command to create the **privatesubnet-us** subnet:

gcloud compute networks subnets create privatesubnet-us --network=privatenet --region=us-west1 --range=172.16.0.0/24

Copied!

content\_copy

1. Run the following command to create the **privatesubnet-eu** subnet:

gcloud compute networks subnets create privatesubnet-eu --network=privatenet --region=europe-west1 --range=172.20.0.0/20

Copied!

content\_copy

**Test Completed Task**

Click **Check my progress** to verify your performed task. If you have successfully created a privatenet network, you will see an assessment score.

Assessment Completed!

Create the privatenet network

Check my progress

*Assessment Completed!*

1. Run the following command to list the available VPC networks:

gcloud compute networks list

Copied!

content\_copy

The output should look like this:

NAME: default

SUBNET\_MODE: AUTO

BGP\_ROUTING\_MODE: REGIONAL

IPV4\_RANGE:

GATEWAY\_IPV4:

NAME: managementnet

SUBNET\_MODE: CUSTOM

BGP\_ROUTING\_MODE: REGIONAL

IPV4\_RANGE:

GATEWAY\_IPV4:

...

**Note: default** and **mynetwork** are auto mode networks, whereas, **managementnet** and **privatenet** are custom mode networks. Auto mode networks create subnets in each region automatically, while custom mode networks start with no subnets, giving you full control over subnet creation

1. Run the following command to list the available VPC subnets (sorted by VPC network):

gcloud compute networks subnets list --sort-by=NETWORK

Copied!

content\_copy

The output should look like this:

NAME: default

REGION: us-west1

NETWORK: default

RANGE: 10.128.0.0/20

STACK\_TYPE: IPV4\_ONLY

IPV6\_ACCESS\_TYPE:

INTERNAL\_IPV6\_PREFIX:

EXTERNAL\_IPV6\_PREFIX:

...

**Note:**As expected, the **default** and **mynetwork** networks have subnets in [each region **(zones/regions may changes as per lab's requirements)**](https://cloud.google.com/compute/docs/regions-zones/#available) as they are auto mode networks. The **managementnet** and **privatenet** networks only have the subnets that you created as they are custom mode networks .

1. In the Cloud console, navigate to **Navigation menu** > **VPC network** > **VPC networks**.
2. You see that the same networks and subnets are listed in the Cloud console.

Create the firewall rules for managementnet

Create firewall rules to allow **SSH**, **ICMP**, and **RDP** ingress traffic to VM instances on the **managementnet** network.

1. In the Cloud console, navigate to **Navigation menu** (Navigation menu icon) > **VPC network** > **Firewall**.
2. Click **+ Create Firewall Rule**.
3. Set the following values, leave all other values at their defaults:

|  |  |
| --- | --- |
| **Property** | **Value (type value or select option as specified)** |
| Name | managementnet-allow-icmp-ssh-rdp |
| Network | managementnet |
| Targets | All instances in the network |
| Source filter | IPv4 Ranges |
| Source IPv4 ranges | 0.0.0.0/0 |
| Protocols and ports | Specified protocols and ports, and then *check* tcp, *type:* 22, 3389; and *check* Other protocols, *type:* icmp. |

**Note:**Make sure to include the **/0** in the **Source IPv4 ranges** to specify all networks.

1. Click **EQUIVALENT COMMAND LINE**.

These commands illustrate that firewall rules can also be created using the Cloud Shell command line. You will create the **privatenet**'s firewall rules using these commands with similar parameters.

1. Click **Close**.
2. Click **Create**.

**Test Completed Task**

Click **Check my progress** to verify your performed task. If you have successfully created firewall rules for managementnet network, you will see an assessment score.

Assessment Completed!

Create the firewall rules for managementnet

Check my progress

*Assessment Completed!*

Create the firewall rules for privatenet

Create the firewall rules for **privatenet** network using the Cloud Shell command line.

1. In Cloud Shell, run the following command to create the **privatenet-allow-icmp-ssh-rdp** firewall rule:

gcloud compute firewall-rules create privatenet-allow-icmp-ssh-rdp --direction=INGRESS --priority=1000 --network=privatenet --action=ALLOW --rules=icmp,tcp:22,tcp:3389 --source-ranges=0.0.0.0/0

Copied!

content\_copy

The output should look like this:

Creating firewall...done.

NAME: privatenet-allow-icmp-ssh-rdp

NETWORK: privatenet

DIRECTION: INGRESS

PRIORITY: 1000

ALLOW: icmp,tcp:22,tcp:3389

DENY:

DISABLED: False

**Test Completed Task**

Click **Check my progress** to verify your performed task. If you have successfully created firewall rules for privatenet network, you will see an assessment score.

Assessment Completed!

Create the firewall rules for privatenet

Check my progress

*Assessment Completed!*

1. Run the following command to list all the firewall rules (sorted by VPC network):

gcloud compute firewall-rules list --sort-by=NETWORK

Copied!

content\_copy

The output should look like this:

NAME: default-allow-icmp

NETWORK: default

DIRECTION: INGRESS

PRIORITY: 65534

ALLOW: icmp

DENY:

DISABLED: False

NAME: default-allow-internal

NETWORK: default

DIRECTION: INGRESS

PRIORITY: 65534

ALLOW: tcp:0-65535,udp:0-65535,icmp

DENY:

DISABLED: False

...

The firewall rules for **mynetwork** network have been created for you. You can define multiple protocols and ports in one firewall rule (**privatenet** and **managementnet**), or spread them across multiple rules (**default** and **mynetwork**).

1. In the Cloud console, navigate to **Navigation menu** > **VPC network** > **Firewall**.
2. You see that the same firewall rules are listed in the Cloud console.

**Task 2. Create VM instances**

Create two VM instances:

* **managementnet-us-vm** in **managementsubnet-us**
* **privatenet-us-vm** in **privatesubnet-us**

Create the managementnet-us-vm instance

Create the **managementnet-us-vm** instance using the Cloud console.

1. In the Cloud console, navigate to **Navigation menu** > **Compute Engine** > **VM instances**.

The **mynet-eu-vm** and **mynet-us-vm** has been created for you, as part of your network diagram.

1. Click **Create instance**.
2. Set the following values, leave all other values at their defaults:

|  |  |
| --- | --- |
| **Property** | **Value (type value or select option as specified)** |
| Name | managementnet-us-vm |
| Region | us-west1 |
| Zone | us-west1-c |
| Series | E2 |
| Machine type | e2-micro |

1. From **Advanced options**, click **Networking, Disks, Security, Management, Sole-tenancy** dropdown.
2. Click **Networking**.
3. For **Network interfaces**, click the dropdown to edit.
4. Set the following values, leave all other values at their defaults:

|  |  |
| --- | --- |
| **Property** | **Value (type value or select option as specified)** |
| Network | managementnet |
| Subnetwork | managementsubnet-us |

1. Click **Done**.
2. Click **EQUIVALENT CODE**.

This illustrate that VM instances can also be created using the Cloud Shell command line. You will create the **privatenet-us-vm** instance using these commands with similar parameters.

1. Click **Create**.

**Test Completed Task**

Click **Check my progress** to verify your performed task. If you have successfully created VM instance in managementnet network, you will see an assessment score.

Assessment Completed!

Create the managementnet-us-vm instance

Check my progress

*Assessment Completed!*

Create the privatenet-us-vm instance

Create the **privatenet-us-vm** instance using the Cloud Shell command line.

1. In Cloud Shell, run the following command to create the **privatenet-us-vm** instance:

gcloud compute instances create privatenet-us-vm --zone=us-west1-c --machine-type=e2-micro --subnet=privatesubnet-us

Copied!

content\_copy

The output should look like this:

Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-04-972c7275ce91/zones/"us-west1-c"/instances/privatenet-us-vm].

NAME: privatenet-us-vm

ZONE: us-west1-c

MACHINE\_TYPE: e2-micro

PREEMPTIBLE:

INTERNAL\_IP: 172.16.0.2

EXTERNAL\_IP: 34.135.195.199

STATUS: RUNNING

**Test Completed Task**

Click **Check my progress** to verify your performed task. If you have successfully created VM instance in privatenet network, you will see an assessment score.

Assessment Completed!

Create the privatenet-us-vm instance

Check my progress

*Assessment Completed!*

1. Run the following command to list all the VM instances (sorted by zone):

gcloud compute instances list --sort-by=ZONE

Copied!

content\_copy

The output should look like this:

NAME: mynet-eu-vm

ZONE: europe-west1-c

MACHINE\_TYPE: e2-micro

PREEMPTIBLE:

INTERNAL\_IP: 10.164.0.2

EXTERNAL\_IP: 34.147.23.235

STATUS: RUNNING

NAME: mynet-us-vm

ZONE: us-west1-c

MACHINE\_TYPE: e2-micro

PREEMPTIBLE:

INTERNAL\_IP: 10.128.0.2

EXTERNAL\_IP: 35.232.221.58

STATUS: RUNNING

...

1. In the Cloud console, navigate to **Navigation menu** (Navigation menu icon) > **Compute Engine** > **VM instances**.
2. You see that the VM instances are listed in the Cloud console.
3. Click on **Column display options**, then select **Network**. Click **Ok**.

There are three instances in **us-west1** and one instance in **europe-west1**. However, these instances are spread across three VPC networks (**managementnet**, **mynetwork** and **privatenet**), with no instance in the same zone and network as another. In the next section, you explore the effect this has on internal connectivity.

**Task 3. Explore the connectivity between VM instances**

Explore the connectivity between the VM instances. Specifically, determine the effect of having VM instances in the same zone versus having instances in the same VPC network.

Ping the external IP addresses

Ping the external IP addresses of the VM instances to determine if you can reach the instances from the public internet.

1. In the Cloud console, navigate to **Navigation menu** > **Compute Engine** > **VM instances**.
2. Note the external IP addresses for **mynet-eu-vm**, **managementnet-us-vm**, and **privatenet-us-vm**.
3. For **mynet-us-vm**, click **SSH** to launch a terminal and connect.
4. To test connectivity to **mynet-eu-vm**'s external IP, run the following command, replacing **mynet-eu-vm**'s external IP:

ping -c 3 'Enter mynet-eu-vm external IP here'

Copied!

content\_copy

This should work!

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

ping -c 3 'Enter managementnet-us-vm external IP here'

Copied!

content\_copy

This should work!

1. 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 external IP here'

Copied!

content\_copy

This should work!

**Note:**You are able to ping the external IP address of all VM instances, even though they are either in a different zone or VPC network. This confirms public access to those instances is only controlled by the **ICMP** firewall rules that you established earlier.

Ping the internal IP addresses

Ping the internal IP addresses of the VM instances to determine if you can reach the instances from within a VPC network.

1. In the Cloud console, navigate to **Navigation menu** > **Compute Engine** > **VM instances**.
2. Note the internal IP addresses for **mynet-eu-vm**, **managementnet-us-vm**, and **privatenet-us-vm**.
3. Return to the **SSH** terminal for **mynet-us-vm**.
4. To test connectivity to **mynet-eu-vm**'s internal IP, run the following command, replacing **mynet-eu-vm**'s internal IP:

ping -c 3 'Enter mynet-eu-vm internal IP here'

Copied!

content\_copy

**Note:**You are able to ping the internal IP address of **mynet-eu-vm** because it is on the same VPC network as the source of the ping (**mynet-us-vm**), even though both VM instances are in separate zones, regions and continents!

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

ping -c 3 'Enter managementnet-us-vm internal IP here'

Copied!

content\_copy

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

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 internal IP here'

Copied!

content\_copy

**Note:**This should not work either as indicated by a 100% packet loss! You are unable to ping the internal IP address of **managementnet-us-vm** and **privatenet-us-vm** because they are in separate VPC networks from the source of the ping (**mynet-us-vm**), even though they are all in the same region **us-west1**.

VPC networks are by default isolated private networking domains. However, no internal IP address communication is allowed between networks, unless you set up mechanisms such as VPC peering or VPN.

**Note:**For the below task consider **region\_1 = us-west1** and **region\_2 = europe-west1**`

Which instance(s) should you be able to ping from mynet-region\_1-vm using internal IP addresses?

checkmynet-region\_2-vm

privatenet-region\_1-vm

managementnet-region\_1-vm

Submit

**Task 4. Create a VM instance with multiple network interfaces**

Every instance in a VPC network has a default network interface. You can create additional network interfaces attached to your VMs. Multiple network interfaces enable you to create configurations in which an instance connects directly to several VPC networks (up to 8 interfaces, depending on the instance's type).

Create the VM instance with multiple network interfaces

Create the **vm-appliance** instance with network interfaces in **privatesubnet-us**, **managementsubnet-us** and **mynetwork**. The CIDR ranges of these subnets do not overlap, which is a requirement for creating a VM with multiple network interface controllers (NICs).

1. In the Cloud console, navigate to **Navigation menu** > **Compute Engine** > **VM instances**.
2. Click **Create instance**.
3. Set the following values, leave all other values at their defaults:

|  |  |
| --- | --- |
| **Property** | **Value (type value or select option as specified)** |
| Name | vm-appliance |
| Region | us-west1 |
| Zone | us-west1-c |
| Series | E2 |
| Machine type | e2-standard-4 |

**Note:**The number of interfaces allowed in an instance is dependent on the instance's machine type and the number of vCPUs. The e2-standard-4 allows up to 4 network interfaces. Refer to [the Maximum number of network interfaces section of the Google Cloud Guide](https://cloud.google.com/vpc/docs/create-use-multiple-interfaces#max-interfaces) for more information.

1. From **Advanced options**, click **Networking, Disks, Security, Management, Sole-tenancy** dropdown.
2. Click **Networking**.
3. For **Network interfaces**, click the dropdown to edit.
4. Set the following values, leave all other values at their defaults:

|  |  |
| --- | --- |
| **Property** | **Value (type value or select option as specified)** |
| Network | privatenet |
| Subnetwork | privatesubnet-us |

1. Click **Done**.
2. Click **Add a network interface**.
3. Set the following values, leave all other values at their defaults:

|  |  |
| --- | --- |
| **Property** | **Value (type value or select option as specified)** |
| Network | managementnet |
| Subnetwork | managementsubnet-us |

1. Click **Done**.
2. Click **Add a network interface**.
3. Set the following values, leave all other values at their defaults:

|  |  |
| --- | --- |
| **Property** | **Value (type value or select option as specified)** |
| Network | mynetwork |
| Subnetwork | mynetwork |

1. Click **Done**.
2. Click **Create**.

**Test Completed Task**

Click **Check my progress** to verify your performed task. If you have successfully created VM instance with multiple network interfaces, you will see an assessment score.

Assessment Completed!

Create a VM instance with multiple network interfaces

Check my progress

*Assessment Completed!*

Explore the network interface details

Explore the network interface details of **vm-appliance** within the Cloud console and within the VM's terminal.

1. In the Cloud console, navigate to **Navigation menu** (Navigation menu icon) > **Compute Engine** > **VM instances**.
2. Click **nic0** within the **Internal IP** address of **vm-appliance** to open the **Network interface details** page.
3. Verify that **nic0** is attached to **privatesubnet-us**, is assigned an internal IP address within that subnet (172.16.0.0/24), and has applicable firewall rules.
4. Click **nic0** and select **nic1**.
5. Verify that **nic1** is attached to **managementsubnet-us**, is assigned an internal IP address within that subnet (10.130.0.0/20), and has applicable firewall rules.
6. Click **nic1** and select **nic2**.
7. Verify that **nic2** is attached to **mynetwork**, is assigned an internal IP address within that subnet (10.128.0.0/20), and has applicable firewall rules.

**Note:**Each network interface has its own internal IP address so that the VM instance can communicate with those networks.

1. In the Cloud console, navigate to **Navigation menu** > **Compute Engine** > **VM instances**.
2. For **vm-appliance**, click **SSH** to launch a terminal and connect.
3. Run the following, to list the network interfaces within the VM instance:

sudo ifconfig

Copied!

content\_copy

The output should look like this:

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1460

inet 172.16.0.3 netmask 255.255.255.255 broadcast 172.16.0.3

inet6 fe80::4001:acff:fe10:3 prefixlen 64 scopeid 0x20<link>

ether 42:01:ac:10:00:03 txqueuelen 1000 (Ethernet)

RX packets 626 bytes 171556 (167.5 KiB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 568 bytes 62294 (60.8 KiB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1460

inet 10.130.0.3 netmask 255.255.255.255 broadcast 10.130.0.3

inet6 fe80::4001:aff:fe82:3 prefixlen 64 scopeid 0x20<link>

ether 42:01:0a:82:00:03 txqueuelen 1000 (Ethernet)

RX packets 7 bytes 1222 (1.1 KiB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 17 bytes 1842 (1.7 KiB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1460

inet 10.128.0.3 netmask 255.255.255.255 broadcast 10.128.0.3

inet6 fe80::4001:aff:fe80:3 prefixlen 64 scopeid 0x20<link>

ether 42:01:0a:80:00:03 txqueuelen 1000 (Ethernet)

RX packets 17 bytes 2014 (1.9 KiB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 17 bytes 1862 (1.8 KiB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

**Note:**The **sudo ifconfig** command lists a Linux VM's network interfaces along with the internal IP addresses for each interface.

Explore the network interface connectivity

Demonstrate that the **vm-appliance** instance is connected to **privatesubnet-us**, **managementsubnet-us** and **mynetwork** by pinging VM instances on those subnets.

1. In the Cloud console, navigate to **Navigation menu** > **Compute Engine** > **VM instances**.
2. Note the internal IP addresses for **privatenet-us-vm**, **managementnet-us-vm**, **mynet-us-vm**, and **mynet-eu-vm**.
3. Return to the **SSH** terminal for **vm-appliance**.
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'

Copied!

content\_copy

This works!

1. Repeat the same test by running the following:

ping -c 3 privatenet-us-vm

Copied!

content\_copy

**Note:**You are able to ping **privatenet-us-vm** by its name because VPC networks have an internal DNS service that allows you to address instances by their DNS names rather than their internal IP addresses. When an internal DNS query is made with the instance hostname, it resolves to the primary interface (nic0) of the instance. Therefore, this only works for **privatenet-us-vm** in this case.

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

ping -c 3 'Enter managementnet-us-vm's internal IP here'

Copied!

content\_copy

This works!

1. 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'

Copied!

content\_copy

This works!

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

ping -c 3 'Enter mynet-eu-vm's internal IP here'

Copied!

content\_copy

**Note:**This does not work! In a multiple interface instance, every interface gets a route for the subnet that it is in. In addition, the instance gets a single default route that is associated with the primary interface eth0. Unless manually configured otherwise, any traffic leaving an instance for any destination other than a directly connected subnet will leave the instance via the default route on eth0.

1. To list the routes for **vm-appliance** instance, run the following command:

ip route

Copied!

content\_copy

The output should look like this:

default via 172.16.0.1 dev eth0

10.128.0.0/20 via 10.128.0.1 dev eth2

10.128.0.1 dev eth2 scope link

10.130.0.0/20 via 10.130.0.1 dev eth1

10.130.0.1 dev eth1 scope link

172.16.0.0/24 via 172.16.0.1 dev eth0

172.16.0.1 dev eth0 scope link

**Note:**The primary interface eth0 gets the default route (default via 172.16.0.1 dev eth0), and all three interfaces eth0, eth1 and eth2 get routes for their respective subnets. Since, the subnet of **mynet-eu-vm** (**10.132.0.0/20**) is not included in this routing table, the ping to that instance leaves **vm-appliance** on eth0 (which is on a different VPC network). You could change this behavior by configuring policy routing as documented in the [Configuring policy routing section of the Google Cloud Guide](https://cloud.google.com/vpc/docs/create-use-multiple-interfaces#configuring_policy_routing).

**Congratulations!**

In this lab you created a VM instance with three network interfaces and verified internal connectivity for VM instances that are on the subnets that are attached to the multiple interface VM.

You also explored the default network along with its subnets, routes, and firewall rules. You then created tested connectivity for a new auto mode VPC network.

Next steps / Learn more

To learn more about VPC networking, see [Using VPC Networks](https://cloud.google.com/vpc/docs/using-vpc).

Google Cloud training and certification

...helps you make the most of Google Cloud technologies. [Our classes](https://cloud.google.com/training/courses) include technical skills and best practices to help you get up to speed quickly and continue your learning journey. We offer fundamental to advanced level training, with on-demand, live, and virtual options to suit your busy schedule. [Certifications](https://cloud.google.com/certification/) help you validate and prove your skill and expertise in Google Cloud technologies.

**Manual last updated April 03, 2024**

**Lab last tested April 02, 2024**

Copyright 2024 Google LLC All rights reserved. Google and the Google logo are trademarks of Google LLC. All other company and product names may be trademarks of the respective companies with which they are associated.