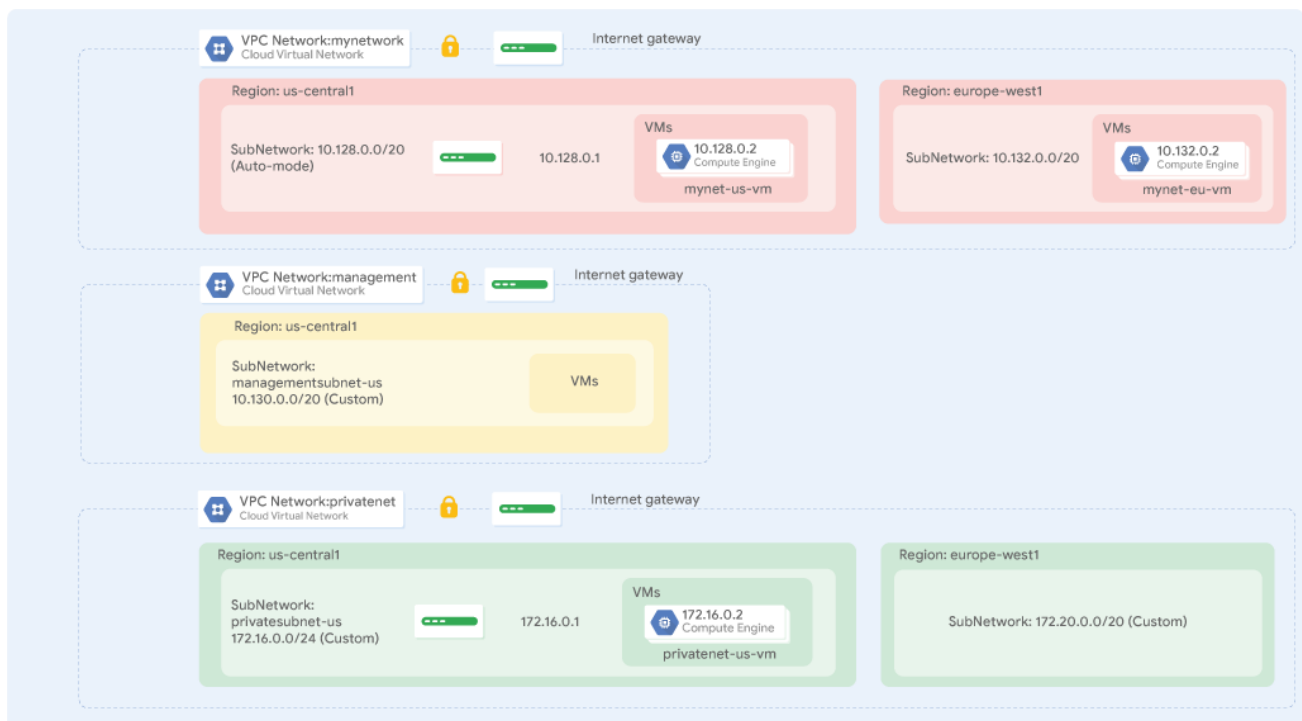


# VPC Networking

GCP Virtual Private Cloud (VPC) provides networking functionality to Compute Engine virtual machine (VM) instances, Kubernetes Engine containers, and the App Engine flexible environment. In other words, without a VPC network, you cannot create VM instances, containers, or App Engine applications. Therefore, each GCP project has a **default** network to get you started.

You can think of a VPC network as similar to a physical network, except that it is virtualized within GCP. A VPC network is a global resource that consists of a list of regional virtual subnetworks (subnets) in data centers, all connected by a global wide area network (WAN). VPC networks are logically isolated from each other in GCP.

In this lab, you create an auto mode VPC network with firewall rules and two VM instances. Then, you convert the auto mode network to a custom mode network and create other custom mode networks as shown in the network diagram below. You also test connectivity across networks.



# Objectives

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

- Explore the default VPC network
- Create an auto mode network with firewall rules
- Convert an auto mode network to a custom mode network
- Create custom mode VPC networks with firewall rules
- Create VM instances using Compute Engine
- Explore the connectivity for VM instances across VPC networks

## 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 the default network

Each GCP project has a **default** network with subnets, routes, and firewall rules.

## View the subnets

The **default** network has a subnet in [each GCP region](#).

- In the GCP Console, on the **Navigation menu** (≡), click **VPC network > VPC networks**. Notice the **default** network with its subnets. Each subnet is associated with a GCP region and a private RFC 1918 CIDR block for its internal **IP addresses range** and a **gateway**.

## View the routes

Routes tell VM instances and the VPC network how to send traffic from an instance to a destination, either inside the network or outside GCP. Each VPC network comes with some default routes to route traffic among its subnets and send traffic from eligible instances to the internet.

- In the left pane, click **Routes**. Notice that there is a route for each subnet and one for the **Default internet gateway** (0.0.0.0/0). These routes are managed for you, but you can create custom static routes to direct some packets to specific destinations. For example, you can create a route that sends all outbound traffic to an instance configured as a NAT gateway.

## View the firewall rules

Each VPC network implements a distributed virtual firewall that you can configure. Firewall rules allow you to control which packets are allowed to travel to which destinations. Every VPC network has two implied firewall rules that block all incoming connections and allow all outgoing connections.

- In the left pane, click **Firewall rules**. Notice that there are 4 **Ingress** firewall rules for the **default** network:
  - default-allow-icmp
  - default-allow-rdp
  - default-allow-ssh
  - default-allow-internal

These firewall rules allow **ICMP**, **RDP**, and **SSH** ingress traffic from anywhere (0.0.0.0/0) and all **TCP**, **UDP**, and **ICMP** traffic within the network (10.128.0.0/9). The **Targets**, **Filters**, **Protocols/ports**, and **Action** columns explain these rules.

## Delete the Firewall rules

1. In the left pane, click **Firewall rules**.
2. Select all default network firewall rules.
3. Click **Delete**.
4. Click **Delete** to confirm the deletion of the firewall rules.

## Delete the default network

1. In the left pane, click **VPC networks**.
2. Select the **default** network.
3. Click **Delete VPC network**.
4. Click **Delete** to confirm the deletion of the **default** network. Wait for the network to be deleted before continuing.
5. In the left pane, click **Routes**. Notice that there are no routes.
6. In the left pane, click **Firewall rules**. Notice that there are no firewall rules.

Without a VPC network, there are no routes!

Without a VPC network, you cannot create VM instances, containers, or App Engine applications.

True

False

## Try to create a VM instance

Verify that you cannot create a VM instance without a VPC network.

1. On the **Navigation menu** (≡), click **Compute Engine > VM instances**.
2. Click **Create**.
3. Accept the default values and click **Create**. Notice the error.
4. Click **Management, security, disks, networking, sole tenancy**.
5. Click **Networking**. Notice the **No local network available** error under **Network interfaces**.
6. Click **Cancel**.

As expected, you cannot create a VM instance without a VPC network!

## Task 2. Create an auto mode network

You have been tasked to create an auto mode network with two VM instances. Auto mode networks are easy to set up and use because they automatically create subnets in each region. However, you don't have complete control over the subnets created in your VPC network, including regions and IP address ranges used. Feel free to explore more [considerations for choosing an auto mode network](#), but for now, assume that you are using the auto mode network for prototyping purposes.

### Create an auto mode VPC network with firewall rules

1. On the **Navigation menu** (≡), click **VPC network > VPC networks**.
2. Click **Create VPC network**.
3. For **Name**, type **mynetwork**
4. For **Subnet creation mode**, click **Automatic**. Auto mode networks create subnets in each region automatically.
5. For **Firewall rules**, select all available rules.

These are the same standard firewall rules that the default network had. The **deny-all-ingress** and **allow-all-egress** rules are also displayed, but you cannot select or disable them because they are implied. These two rules have a lower **Priority** (higher integers indicate lower priorities) so that the allow ICMP, internal, RDP, and SSH rules are considered first.

6. Click **Create**. When the new network is ready, notice that a subnet was created for each region.
7. Record the IP address range for the subnets in **us-central1** and **europe-west1**. These will be referred to in the next steps.

Tip: If you ever delete the default network, you can quickly re-create it by creating an auto mode network as you just did.

## Create a VM instance in us-central1

Create a VM instance in the us-central1 region. Selecting a region and zone determines the subnet and assigns the internal IP address from the subnet's IP address range.

1. On the **Navigation menu** (≡), click **Compute Engine > VM instances**.
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	mynet-us-vm
Region	us-central1
Zone	us-central1-c
Machine type	f1-micro

4. Click **Create**.
5. Verify that the **Internal IP** for the new instance was assigned from the IP address range for the subnet in **us-central1** (10.128.0.0/20).  
The **Internal IP** should be 10.128.0.2 because 10.128.0.1 is reserved for the gateway, and you have not configured any other instances in that subnet.

## Create a VM instance in europe-west1

Create a VM instance in the europe-west1 region.

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


Property	Value (type value or select option as specified)
Name	mynet-eu-vm
Region	europe-west1
Zone	europe-west1-c
Machine type	f1-micro

3. Click **Create**.
4. Verify that the **Internal IP** for the new instance was assigned from the IP address range for the subnet in **europe-west1** (10.132.0.0/20).  
The **Internal IP** should be 10.132.0.2 because 10.132.0.1 is reserved for the gateway, and you have not configured any other instances in that subnet.

The **External IP addresses** for both VM instances are ephemeral. If an instance is stopped, any ephemeral external IP addresses assigned to the instance are released back into the general Compute Engine pool and become available for use by other projects. When a stopped instance is started again, a new ephemeral external IP address is assigned to the instance. Alternatively, you can reserve a static external IP address, which assigns the address to your project indefinitely until you explicitly release it.

## Verify connectivity for the VM instances

The firewall rules that you created with **mynetwork** allow ingress SSH and ICMP traffic from within **mynetwork** (internal IP) and outside that network (external IP).

1. On the **Navigation menu** () , click **Compute Engine > VM instances**.  
Note the external and internal IP addresses for **mynet-eu-vm**.

2. For **mynet-us-vm**, click **SSH** to launch a terminal and connect.

You can SSH because of the **allow-ssh** firewall rule, which allows incoming traffic from anywhere (0.0.0.0/0) for **tcp:22**. The SSH connection works seamlessly because Compute Engine generates an SSH key for you and stores it in one of the following locations:

- By default, Compute Engine adds the generated key to project or instance metadata.
- If your account is configured to use OS Login, Compute Engine stores the generated key with your user account.

Alternatively, you can control access to Linux instances by creating SSH keys and editing public SSH key metadata.

3. 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>
```

You can ping **mynet-eu-vm**'s internal IP because of the **allow-internal** firewall rule.

4. Repeat the same test by running the following:

```
ping -c 3 mynet-eu-vm
```

You can ping **mynet-eu-vm** by its name because VPC networks have an internal DNS service that allows you to address instances by their DNS names instead of their internal IP addresses. This is very useful because the internal IP address can change when you delete and re-create an instance.

5. 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's external IP here>
```

Which firewall rule allows the ping to mynet-eu-vm's external IP address?



mynetwork-allow-icmp



mynetwork-allow-ssh



mynetwork-allow-internal



mynetwork-allow-rdp

Submit

You can SSH to **mynet-us-vm** and ping **mynet-eu-vm**'s internal and external IP addresses as expected. Alternatively, you can SSH to **mynet-eu-vm** and ping **mynet-us-vm**'s internal and external IP addresses, which also works.

## Convert the network to a custom mode network

The auto mode network worked great so far, but you have been asked to convert it to a custom mode network so that new subnets aren't automatically created as new regions become available. This could result in overlap with IP addresses used by manually created subnets or static routes, or could interfere with your overall network planning.

1. On the **Navigation menu** (**≡**), click **VPC network > VPC networks**.
2. Click **mynetwork** to open the network details.
3. Click **Edit**.
4. Select **Custom** for the **Subnet creation mode**.
5. Click **Save**.
6. Return to the **VPC networks** page. Wait for the **Mode** of **mynetwork** to change to **Custom**. You can click **Refresh** while you wait.

Click *Check my progress* to verify the objective.

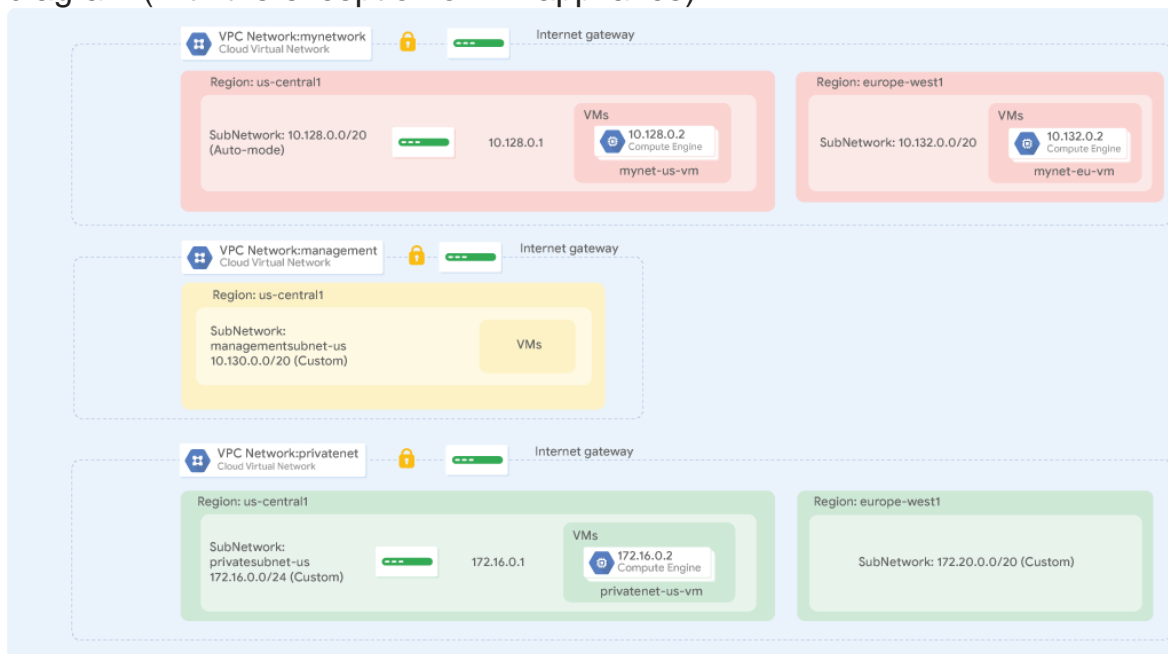
Create a VPC network and VM instances

Check my progress

Converting an auto mode network to a custom mode network is an easy task, and it provides you with more flexibility. We recommend that you use custom mode networks in production.

## Task 3. Create custom mode networks

You have been tasked to create two additional custom networks, **managementnet** and **privatenet**, along with firewall rules to allow **SSH**, **ICMP**, and **RDP** ingress traffic and VM instances as shown in this diagram (with the exception of vm-appliance):



Note that the IP CIDR ranges of these networks do not overlap. This allows you to set up mechanisms such as VPC peering between the networks. If you specify IP CIDR ranges that are different from your on-premises network, you could even configure hybrid connectivity using VPN or Cloud Interconnect.



## Create the managementnet network

Create the **managementnet** network using the GCP Console.

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

Property	Value (type value or select option as specified)
Name	managementsubnet-us
Region	us-central1
IP address range	10.130.0.0/20

6. Click **Done**.
7. Click **command line**.

These commands illustrate that networks and subnets can be created using the `gcloud` command line. You will create the **privatenet** network using these commands with similar parameters.
8. Click **Close**.
9. Click **Create**.

## Create the privatenet network

Create the **privatenet** network using the `gcloud` command line.

1. In the GCP Console, click **Activate Cloud Shell** (▶).
2. If prompted, click **Continue**.
3. To create the **privatenet** network, run the following command:

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

4. To create the **privatesubnet-us** subnet, run the following command:

```
gcloud compute networks subnets create privatesubnet-us --network=privatenet --region=us-central1 --range=172.16.0.0/24
```

5. To create the **privatesubnet-eu** subnet, run the following command:

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

6. To list the available VPC networks, run the following command:

```
gcloud compute networks list
```

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

NAME	SUBNET_MODE	BGP_ROUTING_MODE	IPV4_RANGE	GATEWAY_IPV4
managementnet	CUSTOM	REGIONAL		
mynetwork	CUSTOM	REGIONAL		
privatenet	CUSTOM	REGIONAL		

7. To list the available VPC subnets (sorted by VPC network), run the following command:

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

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

NAME	REGION	NETWORK	RANGE
managementsubnet-us	us-central1	managementnet	10.130.0.0/20
mynetwork	asia-northeast1	mynetwork	10.146.0.0/20
mynetwork	us-west1	mynetwork	10.138.0.0/20
mynetwork	southamerica-east1	mynetwork	10.158.0.0/20
mynetwork	europa-west4	mynetwork	10.164.0.0/20
mynetwork	asia-east1	mynetwork	10.140.0.0/20
mynetwork	europa-north1	mynetwork	10.166.0.0/20
mynetwork	asia-southeast1	mynetwork	10.148.0.0/20
mynetwork	us-east4	mynetwork	10.150.0.0/20
mynetwork	europa-west1	mynetwork	10.132.0.0/20
mynetwork	europa-west2	mynetwork	10.154.0.0/20
mynetwork	europa-west3	mynetwork	10.156.0.0/20
mynetwork	australia-southeast1	mynetwork	10.152.0.0/20
mynetwork	asia-south1	mynetwork	10.160.0.0/20
mynetwork	us-east1	mynetwork	10.142.0.0/20
mynetwork	us-central1	mynetwork	10.128.0.0/20
mynetwork	northamerica-northeast1	mynetwork	10.162.0.0/20
privatesubnet-eu	europa-west1	privatenet	172.20.0.0/20
privatesubnet-us	us-central1	privatenet	172.16.0.0/24

The **managementnet** and **privatenet** networks only have the subnets that you created because they are custom mode networks. **mynetwork** is also a custom mode network, but it started out as an auto mode network, resulting in subnets in each region.

8. In the GCP Console, on the **Navigation menu** (≡), click **VPC network > VPC networks**. Verify that the same networks and subnets are listed in the GCP 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 GCP Console, on the **Navigation menu** (≡), click **VPC network > Firewall rules**.
2. Click **Create Firewall Rule**.
3. Specify the following, and leave the remaining settings as 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	IP Ranges
Source IP ranges	0.0.0.0/0
Protocols and ports	Specified protocols and ports

Make sure to include the **/0** in the **Source IP ranges** to specify all networks.

4. For **tcp**, specify ports **22** and **3389**.
5. Specify the **icmp** protocol.
6. Click **command line**.

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

7. Click **Close**.
8. Click **Create**.

## Create the firewall rules for privatenet

Create the firewall rules for **privatenet** network using the `gcloud` command line.

1. Return to **Cloud Shell**. If necessary, click **Activate Cloud Shell** (▶).
2. To create the **privatenet-allow-icmp-ssh-rdp** firewall rule, run the following command:

```
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
```

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

NAME	NETWORK	DIRECTION	PRIORITY	ALLOW
DENY				
privatenet-allow-icmp-ssh-rdp	privatenet	INGRESS		1000
icmp,tcp:22,tcp:3389				

3. To list all the firewall rules (sorted by VPC network), run the following command:

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

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

NAME	NETWORK	DIRECTION	PRIORITY	ALLOW
managementnet-allow-icmp-ssh-rdp	managementnet	INGRESS		1000
icmp,tcp:22,tcp:3389				
mynetwork-allow-icmp	mynetwork	INGRESS	1000	icmp
mynetwork-allow-internal	mynetwork	INGRESS	65534	all
mynetwork-allow-rdp	mynetwork	INGRESS	1000	tcp:3389
mynetwork-allow-ssh	mynetwork	INGRESS	1000	tcp:22
privatenet-allow-icmp-ssh-rdp	privatenet	INGRESS		1000
icmp,tcp:22,tcp:3389				

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**).

4. In the GCP Console, on the **Navigation menu** (≡), click **VPC network > Firewall rules**. Verify that the same firewall rules are listed in the GCP Console.

Click *Check my progress* to verify the objective.

Create custom mode VPC networks with firewall rules

Check my progress

Next, 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 GCP Console.

1. In the GCP Console, on the **Navigation menu** (≡), 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	managementnet-us-vm
Region	us-central1
Zone	us-central1-c
Machine type	f1-micro

4. Click **Management, security, disks, networking, sole tenancy**.
5. Click **Networking**.
6. For **Network interfaces**, click the pencil icon to edit.
7. Specify the following, and leave the remaining settings as their defaults:

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

The subnets available for selection are restricted to those in the selected region (us-central1).

8. Click **Done**.
9. Click **command line**.

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

10. Click **Close**.
11. Click **Create**.

## Create the privatenet-us-vm instance

Create the **privatenet-us-vm** instance using the `gcloud` command line.

1. Return to **Cloud Shell**. If necessary, click **Activate Cloud Shell** (🔌).
2. To create the **privatenet-us-vm** instance, run the following command:

```
gcloud compute instances create privatenet-us-vm --zone=us-central1-c --  
machine-type=f1-micro --subnet=privatesubnet-us
```

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

NAME	ZONE	MACHINE_TYPE	PREEMPTIBLE	INTERNAL_IP
EXTERNAL_IP	STATUS			
privatenet-us-vm	us-central1-c	f1-micro		172.16.0.2
34.66.197.202	RUNNING			

3. To list all the VM instances (sorted by zone), run the following command:

```
gcloud compute instances list --sort-by=ZONE
```

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

NAME	ZONE	MACHINE_TYPE	PREEMPTIBLE	INTERNAL_IP
EXTERNAL_IP	STATUS			
mynet-eu-vm	europa-west1-c	f1-micro		10.132.0.2
34.76.115.41	RUNNING			
managementnet-us-vm	us-central1-c	f1-micro		10.130.0.2
35.239.68.123	RUNNING			
mynet-us-vm	us-central1-c	f1-micro		10.128.0.2
35.202.101.52	RUNNING			
privatenet-us-vm	us-central1-c	f1-micro		172.16.0.2
34.66.197.202	RUNNING			

4. In the GCP Console, on the **Navigation menu** (☰), click **Compute Engine > VM instances**. Verify that the VM instances are listed in the GCP Console.
5. For **Columns**, select **Network**.

There are three instances in **us-central1-c** and one instance in **europa-west1-c**. 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 task, you explore the effect this has on internal connectivity.

Click *Check my progress* to verify the objective.

Create VM instances

Check my progress

You can explore more networking information on each VM instance by clicking the **nic0** link in the **Internal IP** column. The resulting network interface details page shows the subnet along with the IP CIDR range, the firewall rules and routes that apply to the instance, and other network analysis.

## Task 4. Explore the connectivity across networks

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 whether you can reach the instances from the public internet.

1. In the GCP Console, on the **Navigation menu**, click **Compute Engine > VM instances**. Note the external IP addresses for **mynet-eu-vm**, **managementnet-us-vm**, and **privatenet-us-vm**.
2. For **mynet-us-vm**, click **SSH** to launch a terminal and connect.
3. 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's external IP here>
```

This should work!

4. 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's external IP here>
```

This should work!

5. 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!

You can ping the external IP address of all VM instances, even though they are in either a different zone or VPC network. This confirms that 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 whether you can reach the instances from within a VPC network.

☐ Which instances should you be able to ping from mynet-us-vm using internal IP addresses?

☐ managementnet-us-vm

```
et-eu-vm
privatenet-us-vm
Submit
```

1. In the GCP Console, on the **Navigation menu**, click **Compute Engine > VM instances**. Note the internal IP addresses for **mynet-eu-vm**, **managementnet-us-vm**, and **privatenet-us-vm**.
2. Return to the **SSH** terminal for **mynet-us-vm**.
3. 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>
```

You can 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!

4. 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>
```

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

5. 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 either, as indicated by a 100% packet loss! You cannot 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 zone, **us-central1-c**.

## Task 5. Review

In this lab, you explored the default network and determined that you cannot create VM instances without a VPC network. Thus, you created a new auto mode VPC network with subnets, routes, firewall rules, and two VM instances and tested the connectivity for the VM instances. Because auto mode networks aren't recommended for production, you converted the auto mode network to a custom mode network. Next, you created two more custom mode VPC networks with firewall rules and VM instances using the GCP Console and the `gcloud` command

line. Then you tested the connectivity across VPC networks, which worked when pinging external IP addresses but not when pinging internal IP addresses.

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

## **End your lab**