Start Lab 01:07:30

Configuring Networks via gcloud

Lab 45 minutes No cost Introductory

GSP630

gcloud

Lab 45 minutes No cost Introductory

GSP630

gcloud

Lab 45 minutes No cost Introductory

GSP630

gcloud

Lab 45 minutes No cost Introductory

★★★★★

GSP630

gcloud

Lab 45 minutes No cost Introductory

★★★★★

GSP630

6. 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.

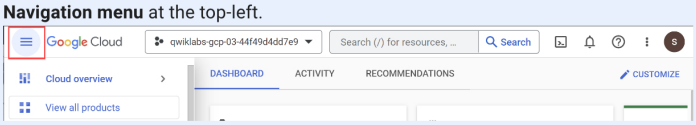
7. Click through the subsequent pages:

   - 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** 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**, `PROJECT_ID`. The output contains a line that declares the **Project_ID** for this session:

```
Your Cloud Platform project in this session is set to "PROJECT_ID"
```

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

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

```
gcloud auth list
```

3. Click **Authorize**.

Output:

```
ACTIVE: *
ACCOUNT: "ACCOUNT"

To set the active account, run:
    $ gcloud config set account `ACCOUNT`
```

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

```
gcloud config list project
```

Output:

```
[core]
project = "PROJECT_ID"
```

Note: For full documentation of `gcloud`, in Google Cloud, refer to the gcloud CLI overview guide.

Install `ping` to use later in this lab:

```
sudo apt install iputils-ping
```

Task 1. Create network

Output:

```
[core]
project = "PROJECT_ID"
```

Note: For full documentation of `gcloud`, in Google Cloud, refer to the gcloud CLI overview guide.

Install `ping` to use later in this lab:

```
sudo apt install iputils-ping
```

Task 1. Create network

Output:

```
[core]
project = "PROJECT_ID"
```

Note: For full documentation of `gcloud`, in Google Cloud, refer to the gcloud CLI overview guide.

Install `ping` to use later in this lab:

```
sudo apt install iputils-ping
```

Task 1. Create network

Output:

```
[core]
project = "PROJECT_ID"
```

Note: For full documentation of `gcloud`, in Google Cloud, refer to the gcloud CLI overview guide.

Install `ping` to use later in this lab:

```
sudo apt install iputils-ping
```

Task 1. Create network

Output:

```
[core]
project = "PROJECT_ID"
```

Note: For full documentation of `gcloud`, in Google Cloud, refer to the gcloud CLI overview guide.

Install `ping` to use later in this lab:

```
sudo apt install iputils-ping
```

Task 1. Create network

Output:

```
[core]
project = "PROJECT_ID"
```

Note: For full documentation of `gcloud`, in Google Cloud, refer to the gcloud CLI overview guide.

Install `ping` to use later in this lab:

```
sudo apt install iputils-ping
```

Task 1. Create network

Output:

```
[core]
project = "PROJECT_ID"
```

Note: For full documentation of `gcloud`, in Google Cloud, refer to the gcloud CLI overview guide.

Install `ping` to use later in this lab:

```
sudo apt install iputils-ping
```

Task 1. Create network

Output:

```
[core]
project = "PROJECT_ID"
```

Note: For full documentation of `gcloud`, in Google Cloud, refer to the gcloud CLI overview guide.

Install `ping` to use later in this lab:

```
sudo apt install iputils-ping
```

Task 1. Create network

Output:

```
[core]
project = "PROJECT_ID"
```

Note: For full documentation of `gcloud`, in Google Cloud, refer to the gcloud CLI overview guide.

Install `ping` to use later in this lab:

```
sudo apt install iputils-ping
```

Task 1. Create network

networks to test the ability to communicate with the networks.

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

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

2. Create the **private-sub** subnet:

```
gcloud compute networks subnets create private-sub \
    --network=privatenet \
```

```
    --region="REGION" \
    --range 10.1.0.0/28
```

Create the firewall rules for privatenet

1. Run the following command to create the **privatenet-deny** firewall rule:

```
gcloud compute firewall-rules create privatenet-deny \
    --network=privatenet \
    --action=DENY \
    --rules=icmp,tcp:22 \
    --source-ranges=0.0.0.0/0
```

This firewall rule denies all access from the internal protocol.

The output should look like this:

```
Creating firewall...done.
NAME: privatenet-deny
NETWORK: privatenet
DIRECTION: INGRESS
PRIORITY: 1000
ALLOW:
DENY: icmp,tcp:22
DISABLED: False
```

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

Create another VPC, subnet and required deny firewall rules

Check my progress

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

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

3. Look for the networks you created to quickly find your firewall rules.

Task 8. Create VM instances

Create two VM instances in the subnets:

- **pnet-vm** in **private-sub**
- **lnet-vm** in **labnet-sub**

Create the pnet-vm instance

- Run the following command to create the **pnet-vm** instance in the `private-sub` subnet:

```
gcloud compute instances create pnet-vm \
--zone="ZONE" \
```

```
--machine-type=n1-standard-1 \
--subnet=private-sub
```

The output should look like this:

```
NAME: pnet-vm
ZONE:"ZONE"
MACHINE_TYPE: n1-standard-1
PREEMPTIBLE:
INTERNAL_IP: 10.1.0.2
EXTERNAL_IP: 34.122.106.173
STATUS: RUNNING
```

Create the lnet-vm instance

1. Using the previous step as your guide, create a VM with the following values:

Property	Value
Name	lnet-vm
Zone	`ZONE`
Machine type	n1-standard-1
Subnet	labnet-sub

You should see a similar output when your subnet is created.

2. Now list all the VM instances (sorted by zone):

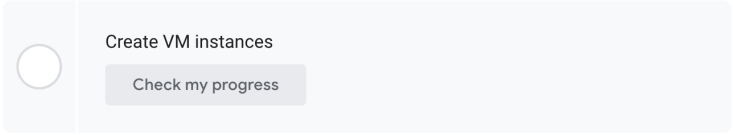
```
gcloud compute instances list --sort-by=ZONE
```

For this command you're using the `instance` subgroup, with it's specialized command `list`.

You should see the 2 VMs you just created:

```
NAME     ZONE           MACHINE_TYPE   ...  INTERNAL_IP  EXTERNAL_IP      S
lnet-vm  ZONE  n1-standard-1              10.0.0.2     35.202.156.230   RU
pnet-vm  ZONE  n1-standard-1              10.0.0.2     104.154.146.108  RU
```

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



Task 9. Explore the connectivity

When you created the networks, you applied firewall rules to each - so one network allows INGRESS traffic, and the other denies INGRESS traffic.

For this experiment, you should be able to communicate with the first network, but be unable to communicate with the second one.

Ping the external IP addresses

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

```
ping -c 3 <Enter lnet-vm's external IP here>
```

This should work - `lnet-vm`'s network has a firewall rule that allows traffic.

2. Repeat the command, but use `pnet-vm`'s external IP address.

This should not work - nothing should be happening. `pnet-vm`'s network has a firewall rule that denies traffic. Use **Ctrl+C** to end the process.

Congratulations!

In this lab you created two custom mode VPC networks, firewall rules, and VM instances using the Cloud Shell command line. Then you tested the ability of the VPC networks to receive traffic from the public internet.

Next steps / learn more

Learn more about VPC networking:

- Using VPC Networks.
- Read more on the rules for creating or editing a subnet.

Google Cloud training and certification

...helps you make the most of Google Cloud technologies. Our classes 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 help you validate and prove your skill and expertise in Google Cloud technologies.

Manual Last Updated May 24, 2024

Lab Last Tested May 24, 2024