

VPC Networks - Controlling Access

1 hour7 Credits

[Rate Lab](#)

GSP213



Google Cloud Self-Paced Labs

Overview

In this lab, you create two nginx web servers and control external HTTP access to the web servers using tagged firewall rules. Then, you explore IAM roles and service accounts.

Objectives

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

- Create an nginx web server
- Create tagged firewall rules
- Create a service account with IAM roles
- Explore permissions for the Network Admin and Security Admin roles

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 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 Google Cloud for the duration of the lab.

What you need

To complete this lab, you need:

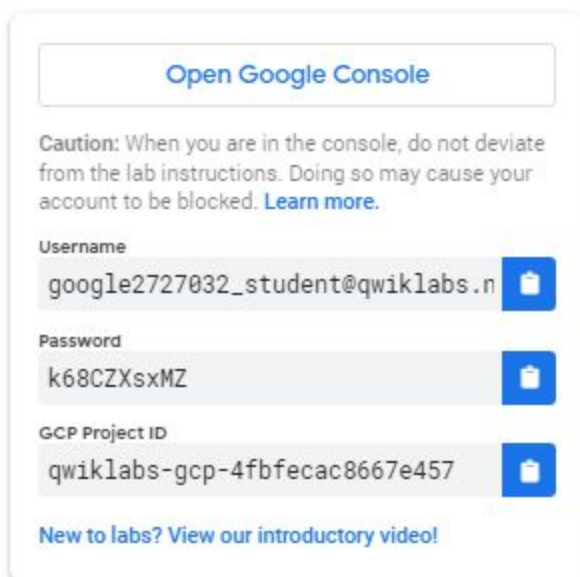
- Access to a standard internet browser (Chrome browser recommended).
- Time to complete the lab.

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

Note: If you are using a Pixelbook, open an Incognito window to run this lab.

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 a panel populated with the temporary credentials that you must use for this lab.

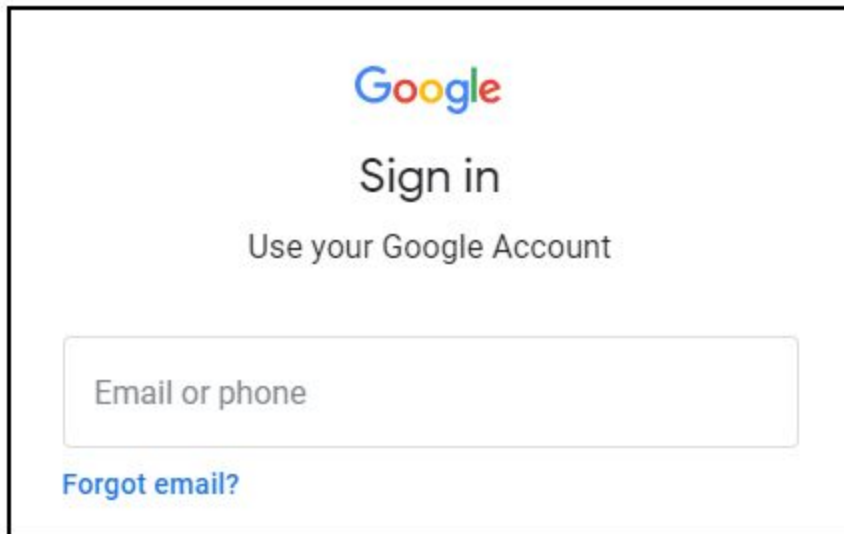


The screenshot shows a sign-in panel for the Google Cloud Console. At the top is a button labeled "Open Google Console". Below it is a caution message: "Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. [Learn more.](#)". The panel contains three input fields, each with a blue copy icon to its right:

- Username:** google2727032_student@qwiklabs.n
- Password:** k68CZsxMZ
- GCP Project ID:** qwiklabs-gcp-4fbfecac8667e457

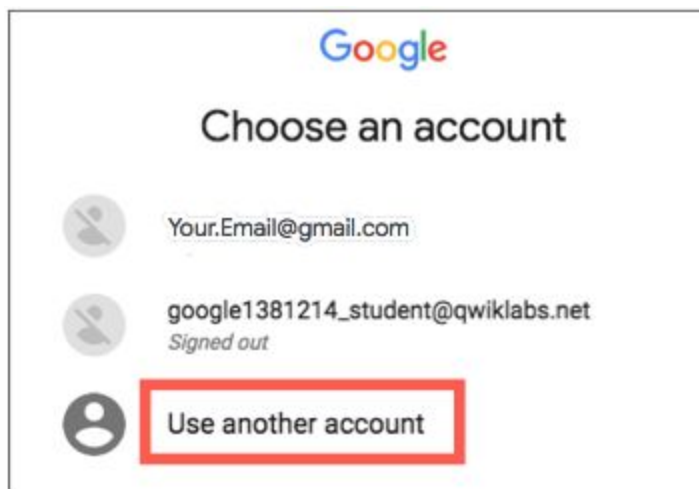
At the bottom of the panel is a link: "New to labs? View our introductory video!"

2. Copy the username, and then click **Open Google Console**. The lab spins up resources, and then opens another tab that shows the **Sign in** page.



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

If you see the **Choose an account** page, click **Use Another Account**.



3. In the **Sign in** page, paste the username that you copied from the Connection Details panel. Then copy and paste the password.

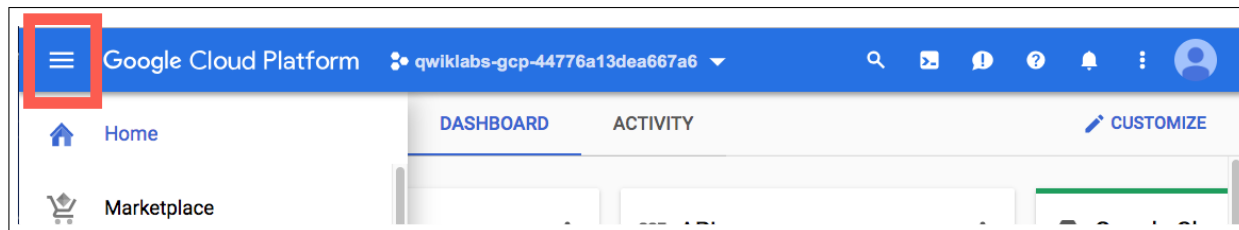
Important: You must use the credentials from the Connection Details panel. Do not use your Qwiklabs credentials. If you have your own Google Cloud account, do not use it for this lab (avoids incurring charges).

4. 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 Cloud Console opens in this tab.

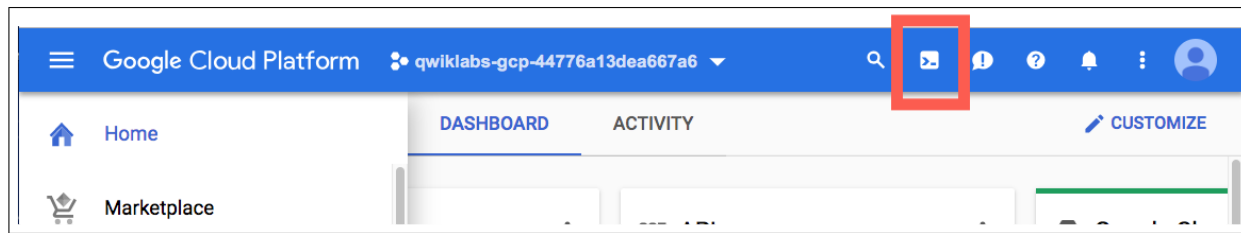
Note: You can view the menu with a list of Google Cloud Products and Services by clicking the **Navigation menu** at the top-left.



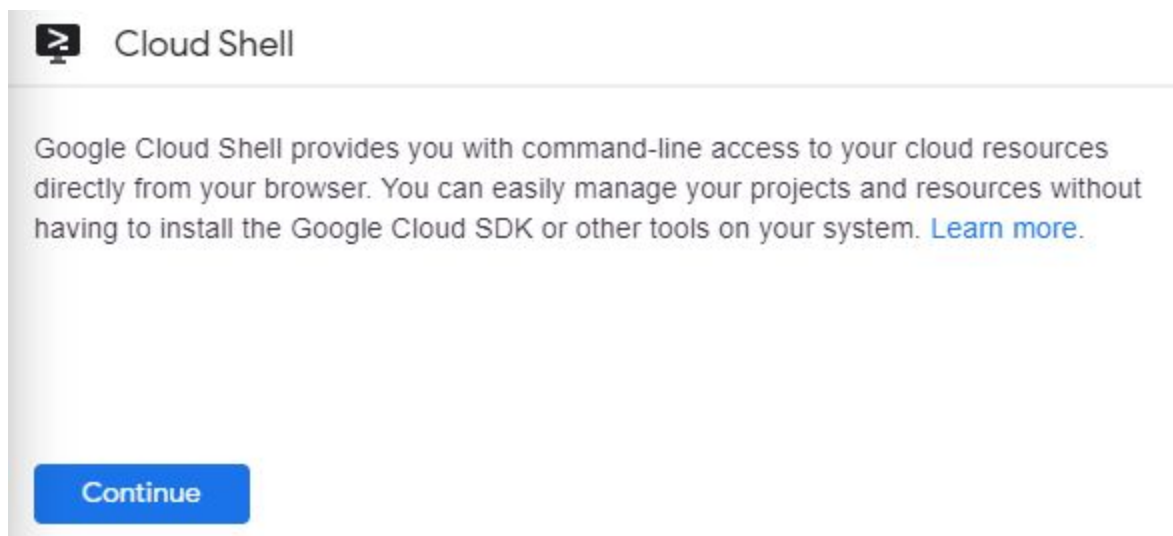
Activate Cloud Shell

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.

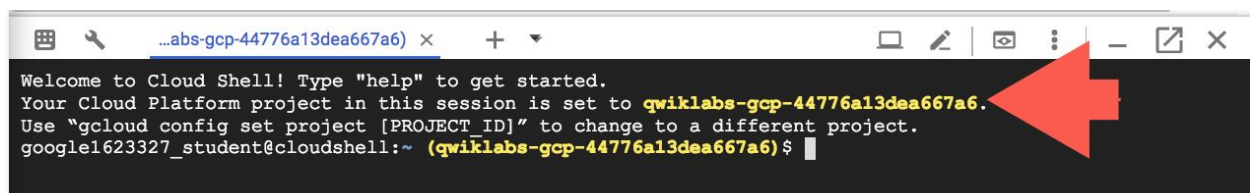
In the Cloud Console, in the top right toolbar, click the **Activate Cloud Shell** button.



Click **Continue**.



It takes a few moments to provision and connect to the environment. When you are connected, you are already authenticated, and the project is set to your *PROJECT_ID*. For example:



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

You can list the active account name with this command:

```
gcloud auth list
```

(Output)

```
Credentialed accounts:  
- <myaccount>@<mydomain>.com (active)
```

(Example output)

```
Credentialed accounts:  
- google1623327_student@qwiklabs.net
```

You can list the project ID with this command:

```
gcloud config list project
```

(Output)

```
[core]  
project = <project_ID>
```

(Example output)

```
[core]  
project = qwiklabs-gcp-44776a13dea667a6
```


For full documentation of gcloud see the [gcloud command-line tool overview](#).

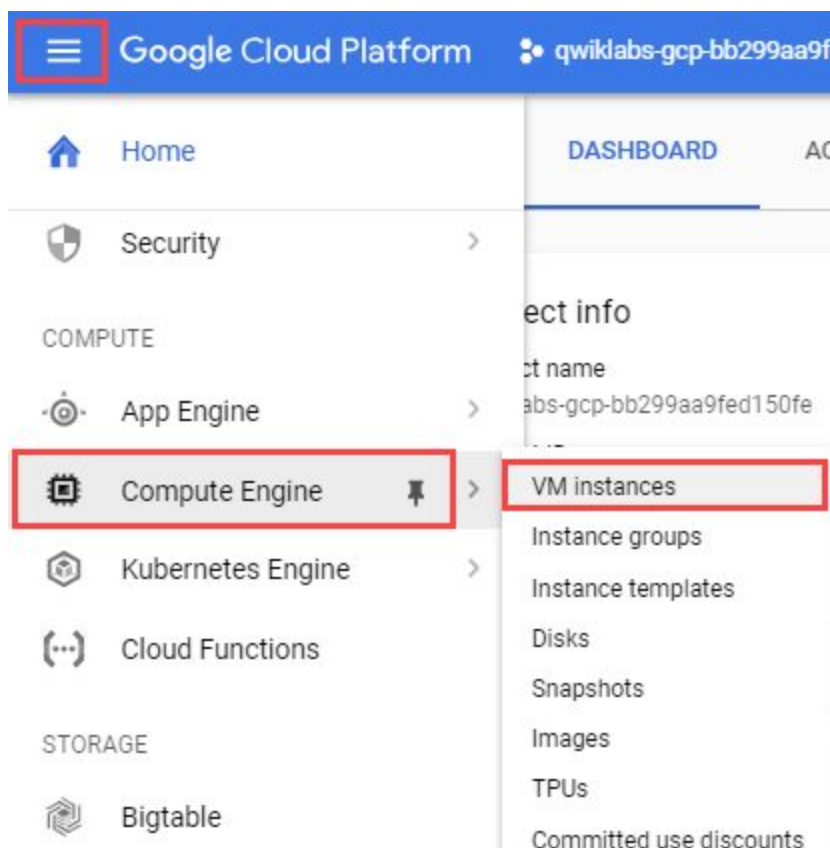
Create the web servers

Create two web servers (**blue** and **green**) in the **default** VPC network. Then, install **nginx** on the web servers and modify the welcome page to distinguish the servers.

Create the blue server

Create the **blue** server with a network tag.

1. In the Console, navigate to **Navigation menu** () > **Compute Engine** > **VM instances**.



2. Click **Create**.

3. Set the following values, leave all other values at their defaults:

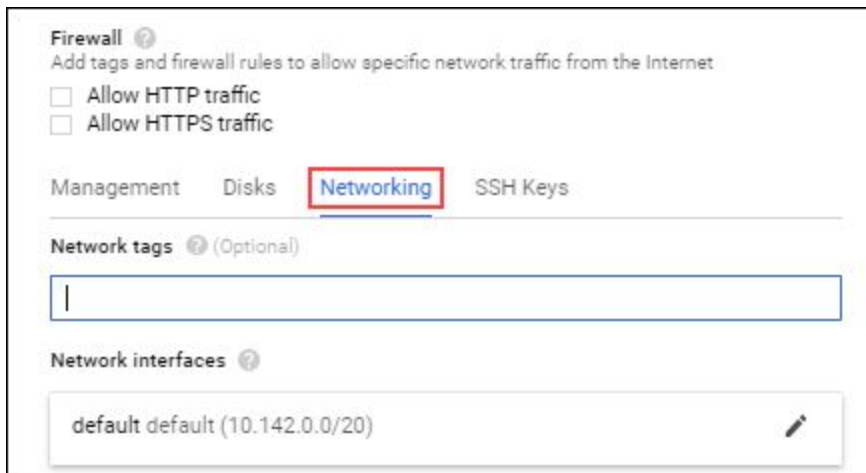
Property	Value (type value or select option as specified)
Name	blue
Region	us-central1 (Iowa)
Zone	us-central1-a

4.

For more information on available regions and zones, refer [here](#).

5. Click **Management, disks, networking, sole tenancy**.

6. Click **Networking**.




Firewall ⓘ
Add tags and firewall rules to allow specific network traffic from the Internet

☐ Allow HTTP traffic
☐ Allow HTTPS traffic

Management Disks **Networking** SSH Keys

Network tags ⓘ (Optional)

Network interfaces ⓘ
default default (10.142.0.0/20) 

7. For **Network tags**, type **web-server**.

Note: Networks use network tags to identify which VM instances are subject to certain firewall rules and network routes. Later in this lab, you create a firewall rule to allow HTTP access for VM instances with the **web-server** tag.

Alternatively, you could check the **Allow HTTP traffic** checkbox, which would tag this instance as **http-server** and create the tagged firewall rule for tcp:80 for you.

7. Click **Create**.

Test Completed Task

Click **Check my progress** to verify your performed task. If you have completed the task successfully you will be granted with an assessment score.

Create the blue server.

Check my progress

Create the green server

Create the **green** server without a network tag.

1. Still in the Console, in the **VM instances** dialog, 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	green
Region	us-central1 (Iowa)
Zone	us-central1-a

3.

Click **Create**.

Test Completed Task

Click **Check my progress** to verify your performed task. If you have completed the task successfully you will be granted with an assessment score.

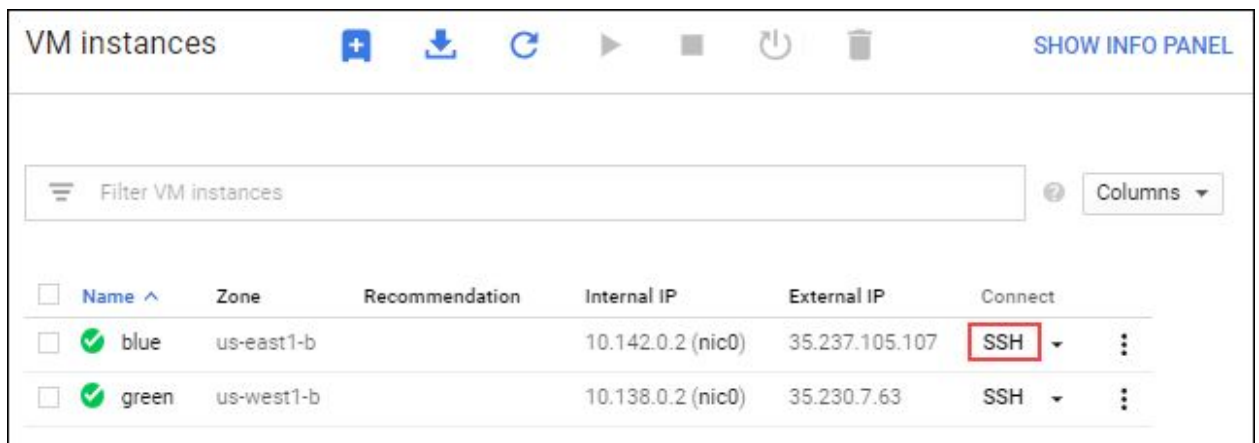
Create the green server.

Check my progress

Install nginx and customize the welcome page

Install nginx on both VM instances and modify the welcome page to distinguish the servers.

1. Still in the **VM instances** dialog, for **blue**, click **SSH** to launch a terminal and connect.



The screenshot shows the 'VM instances' page in the Google Cloud console. At the top, there are icons for adding, downloading, refreshing, playing, stopping, restarting, and deleting instances, along with a 'SHOW INFO PANEL' link. Below this is a search bar labeled 'Filter VM instances' and a 'Columns' dropdown menu. The main table lists two VM instances: 'blue' and 'green'. The 'blue' instance is in the 'us-east1-b' zone with an internal IP of 10.142.0.2 and an external IP of 35.237.105.107. The 'green' instance is in the 'us-west1-b' zone with an internal IP of 10.138.0.2 and an external IP of 35.230.7.63. Both instances have a status of 'Running' (indicated by a green checkmark). The 'Connect' column for each instance shows an 'SSH' button, which is highlighted with a red box for the 'blue' instance.

<input type="checkbox"/>	Name ^	Zone	Recommendation	Internal IP	External IP	Connect
<input type="checkbox"/>	✓ blue	us-east1-b		10.142.0.2 (nic0)	35.237.105.107	SSH
<input type="checkbox"/>	✓ green	us-west1-b		10.138.0.2 (nic0)	35.230.7.63	SSH

2. In the SSH terminal to blue, run the following command to install nginx:

```
sudo apt-get install nginx-light -y
```

3. Open the welcome page in the nano editor:

```
sudo nano /var/www/html/index.nginx-debian.html
```

4. Replace the `<h1>Welcome to nginx!</h1>` line with `<h1>Welcome to the blue server!</h1>`.
5. Press **CTRL+o**, **ENTER**, **CTRL+x**.
6. Verify the change:

```
cat /var/www/html/index.nginx-debian.html
```

The output should contain the following (**do not copy; this is example output**):

```
...  
<h1>Welcome to the blue server!</h1>  
<p>If you see this page, the nginx web server is successfully installed and  
working. Further configuration is required.</p>  
...
```

7. Close the SSH terminal to **blue**:

```
exit
```

Repeat the same steps for the **green** server:

8. For **green**, click **SSH** to launch a terminal and connect.
9. Install nginx:

```
sudo apt-get install nginx-light -y
```

10. Open the welcome page in the nano editor:

```
sudo nano /var/www/html/index.nginx-debian.html
```

11. Replace the `<h1>Welcome to nginx!</h1>` line with `<h1>Welcome to the green server!</h1>`.
12. Press **CTRL+o**, **ENTER**, **CTRL+x**.
13. Verify the change:

```
cat /var/www/html/index.nginx-debian.html
```

The output should contain the following (**do not copy; this is example output**):

```
...  
<h1>Welcome to the green server!</h1>  
<p>If you see this page, the nginx web server is successfully installed and  
working. Further configuration is required.</p>  
...
```

14. Close the SSH terminal to **green**:

```
exit
```

Test Completed Task

Click **Check my progress** to verify your performed task. If you have completed the task successfully you will be granted with an assessment score.

Install Nginx and customize the welcome page.


Check my progress

Create the firewall rule

Create the tagged firewall rule and test HTTP connectivity.

Create the tagged firewall rule

Create a firewall rule that applies to VM instances with the **web-server** network tag.

1. In the Console, navigate to **Navigation menu** () > **VPC network** > **Firewall rules**.
2. Notice the **default-allow-internal** firewall rule.

The **default-allow-internal** firewall rule allows traffic on all protocols/ports within the **default** network. You want to create a firewall rule to allow traffic from outside this network to only the **blue** server, by using the network tag **web-server**.

3. Click **Create Firewall Rule**.
4. Set the following values, leave all other values at their defaults and click **Create**:

Property	Value (type value or select option as specified)
Name	allow-http-web-server
Network	default
Targets	Specified target tags
Target tags	web-server
Source filter	IP Ranges
Source IP ranges	0.0.0.0/0
Protocols and ports	Specified protocols and ports, and then <i>check</i> tcp, <i>type</i> : 80; and <i>check</i> Other protocols, <i>type</i> : icmp.

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

5. Click **Create**.

Test Completed Task

Click **Check my progress** to verify your performed task. If you have completed the task successfully you will be granted with an assessment score.

Create the tagged firewall rule.

Check my progress

Create a test-vm

Create a **test-vm** instance using the Cloud Shell command line.

Create a **test-vm** instance, in the us-central1-a zone:

```
gcloud compute instances create test-vm --machine-type=f1-micro  
--subnet=default --zone=us-central1-a
```

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

NAME	ZONE	MACHINE_TYPE	PREEMPTIBLE	INTERNAL_IP	EXTERNAL_IP
test-vm	us-central1-a	f1-micro		10.142.0.4	35.237.134.68

You can easily create VM instances from the Console or the gcloud command line.


Test Completed Task

Click **Check my progress** to verify your performed task. If you have completed the task successfully you will be granted with an assessment score.

Create a test-vm.

Test HTTP connectivity

From **test-vm** curl the internal and external IP addresses of **blue** and **green**.

1. In the Console, navigate to **Navigation menu** () > **Compute Engine** > **VM instances**.
2. Note the internal and external IP addresses of **blue** and **green**.
3. For **test-vm**, click **SSH** to launch a terminal and connect.
4. To test HTTP connectivity to **blue**'s internal IP, run the following command, replacing **blue**'s internal IP:

```
curl <Enter blue's internal IP here>
```

- 5.
6. You should see the Welcome to the blue server! header.
7. To test HTTP connectivity to **green**'s internal IP, run the following command, replacing **green**'s internal IP:

```
curl -c 3 <Enter green's internal IP here>
```

You should see the Welcome to the green server! header.

You are able to HTTP access both servers using their internal IP addresses. The connection on tcp:80 is allowed by the **default-allow-internal** firewall rule, as **test-vm** is on the same VPC network as the web servers **default** network).

6. To test HTTP connectivity to **blue**'s external IP, run the following command, replacing **blue**'s external IP:


```
curl <Enter blue's external IP here>
```

- 7.
8. You should see the Welcome to the blue server! header.
9. To test HTTP connectivity to **green's** external IP, run the following command, replacing **green's** external IP:

```
curl -c 3 <Enter green's external IP here>
```

This should not work! The request hangs.

8. Press **CTRL+c** to stop the HTTP request.

As expected, you are only able to HTTP access the external IP address of the **blue** server as the **allow-http-web-server** only applies to VM instances with the **web-server** tag.

You can verify the same behavior from your browser by opening a new tab and navigating to `http://[External IP of server]`.

Explore the Network and Security Admin roles

Cloud IAM lets you authorize who can take action on specific resources, giving you full control and visibility to manage cloud resources centrally. The following

roles are used in conjunction with single-project networking to independently control administrative access to each VPC Network:

- **Network Admin:** Permissions to create, modify, and delete networking resources, except for firewall rules and SSL certificates.
- **Security Admin:** Permissions to create, modify, and delete firewall rules and SSL certificates.

Explore these roles by applying them to a service account, which is a special Google account that belongs to your VM instance, instead of to an individual end user. Rather than creating a new user, you will authorize **test-vm** to use the service account to demonstrate the permissions of the **Network Admin** and **Security Admin** roles.

Verify current permissions

Currently, **test-vm** uses the [Compute Engine default service account](#), which is enabled on all instances created by Cloud Shell command-line and the Cloud Console.

Try to list or delete the available firewall rules from **test-vm**.

1. Return to the **SSH** terminal of the **test-vm** instance.
2. Try to list the available firewall rules:

```
gcloud compute firewall-rules list
```

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

```
ERROR: (gcloud.compute.firewall-rules.list) Some requests did not succeed:  
- Insufficient Permission
```

This should not work!

3. Try to delete the **allow-http-web-server** firewall rule:

```
gcloud compute firewall-rules delete allow-http-web-server
```

4. Enter **Y**, if asked to continue.

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


```
ERROR: (gcloud.compute.firewall-rules.delete) Could not fetch resource:  
- Insufficient Permission
```

This should not work!

The **Compute Engine default service account** does not have the right permissions to allow you to list or delete firewall rules. The same applies to other users who do not have the right roles.

Create a service account

Create a service account and apply the **Network Admin** role.

1. In the Console, navigate to **Navigation menu** () > **IAM & admin** > **Service accounts**.

2. Notice the **Compute Engine default service account**.
3. Click **Create service account**.
4. Set the **Service account** name to **Network-admin** and click **CREATE**.
5. For **Select a role**, select **Compute Engine > Compute Network Admin** and click **CONTINUE**.
6. Click **CREATE KEY**.
7. For **Key type**, select **JSON** and click **CREATE**.
8. Click **Save**.

A JSON key file download to your local computer. Find this key file, you will upload it in to the VM in a later step.
9. Rename the JSON key file on your local machine to **credentials.json**
10. Click **Close**.
11. Click **DONE**.

Test Completed Task

Click **Check my progress** to verify your performed task. If you have completed the task successfully you will be granted with an assessment score.

Create a Network-admin service account.

Check my progress

Authorize test-vm and verify permissions

Authorize **test-vm** to use the **Network-admin** service account.

The Network Admin role provides permissions to:

Create a firewall rules

Modify the available firewall rules

Neither list, create, modify, or delete the available firewall rules

Delete the available firewall rules

List the available firewall rules

Submit

1. Return to the **SSH** terminal of the **test-vm** instance.
2. To upload **credentials.json** through the SSH VM terminal, click on the gear icon in the upper-right corner, and then click **Upload file**.
3. Select **credentials.json** and upload it.
4. Click **Close** in the File Transfer window.
5. Authorize the VM with the credentials you just uploaded:

```
gcloud auth activate-service-account --key-file credentials.json
```

The image you are using has the Cloud SDK pre-installed; therefore, you don't need to initialize the Cloud SDK. If you are attempting this lab in a different environment, make sure you have followed the [procedures regarding installing the Cloud SDK](#).

6. Try to list the available firewall rules:

```
gcloud compute firewall-rules list
```

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

NAME	NETWORK	DIRECTION	PRIORITY	ALLOW	DENY
allow-http-web-server	default	INGRESS	1000	tcp:80	
default-allow-icmp	default	INGRESS	65534	icmp	
default-allow-internal	default	INGRESS	65534	all	
default-allow-rdp	default	INGRESS	65534	tcp:3389	

```
default-allow-ssh    default    INGRESS    65534    tcp:22
```

This should work!

7. Try to delete the **allow-http-web-server** firewall rule:

```
gcloud compute firewall-rules delete allow-http-web-server
```

8. Enter **Y**, if asked to continue.

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

```
ERROR: (gcloud.compute.firewall-rules.delete) Could not fetch resource:  
- Required 'compute.firewalls.delete' permission for  
'projects/[PROJECT_ID]/global/firewalls/allow-http-web-server'
```

This should not work!

As expected, the **Network Admin** role has permissions to list but not modify/delete firewall rules.

Update service account and verify permissions

Update the **Network-admin** service account by providing it the **Security Admin** role.

The Security Admin role, provides permissions to:

List the available firewall rules


Neither list, create, modify, or delete the available firewall rules

Create a firewall rules

Modify the available firewall rules

Delete the available firewall rules

Submit

1. In the Console, navigate to **Navigation menu** () > **IAM & admin** > **IAM**.
2. Find the **Network-admin** account. Focus on the **Name** column to identify this account.
3. Click on the pencil icon for the **Network-admin** account.
4. Change **Role** to **Compute Engine > Compute Security Admin**.
5. Click **Save**.
6. Return to the **SSH** terminal of the **test-vm** instance.
7. Try to list the available firewall rules:

```
gcloud compute firewall-rules list
```

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

NAME	NETWORK	DIRECTION	PRIORITY	ALLOW	DENY
allow-http-web-server	default	INGRESS	1000	tcp:80	
default-allow-icmp	default	INGRESS	65534	icmp	
default-allow-internal	default	INGRESS	65534	all	
default-allow-rdp	default	INGRESS	65534	tcp:3389	
default-allow-ssh	default	INGRESS	65534	tcp:22	

This should work!

8. Try to delete the **allow-http-web-server** firewall rule:

```
gcloud compute firewall-rules delete allow-http-web-server
```

9. Enter **Y**, if asked to continue.

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

```
Deleted
```

```
[https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-00e186e4b1cec086/global/firewalls/allow-http-web-server].
```

This should work!

As expected, the **Security Admin** role has permissions to list and delete firewall rules.

Verify the deletion of the firewall rule

Verify that you can no longer HTTP access the external IP of the **blue** server, because you deleted the **allow-http-web-server** firewall rule.

1. Return to the **SSH** terminal of the **test-vm** instance.
2. To test HTTP connectivity to **blue**'s external IP, run the following command, replacing **blue**'s external IP:

```
curl -c 3 <Enter blue's external IP here>
```

This should not work!

3. Press **CTRL+c** to stop the HTTP request.

Provide the **Security Admin** role to the right user or service account to avoid any unwanted changes to your firewall rules!