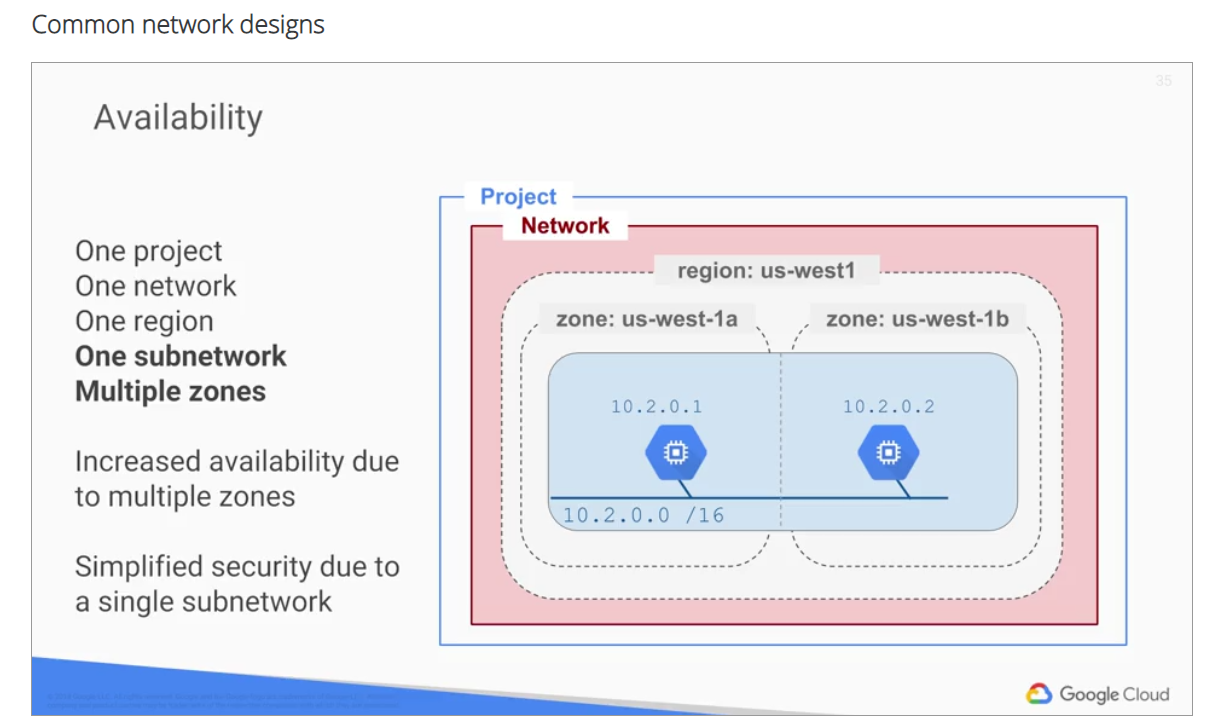
Common network designs

Specifically, we have gone over projects, networks, subnetworks, regions and zones. The question now is, how do all of these elements work together? In short, they provide a rich set of alternatives for managing groups of resources with varying availability and access control requirements. So, what does that mean? Well, basically you can now manage resources at a very granular level, depending on how specific you need to get. If you need to work things globally, you have that capacity to do so, but you can also restrict yourself to very finite resources

# Availability

You could place two virtual machines into multiple zones within the same subnetwork, as shown on the slide.

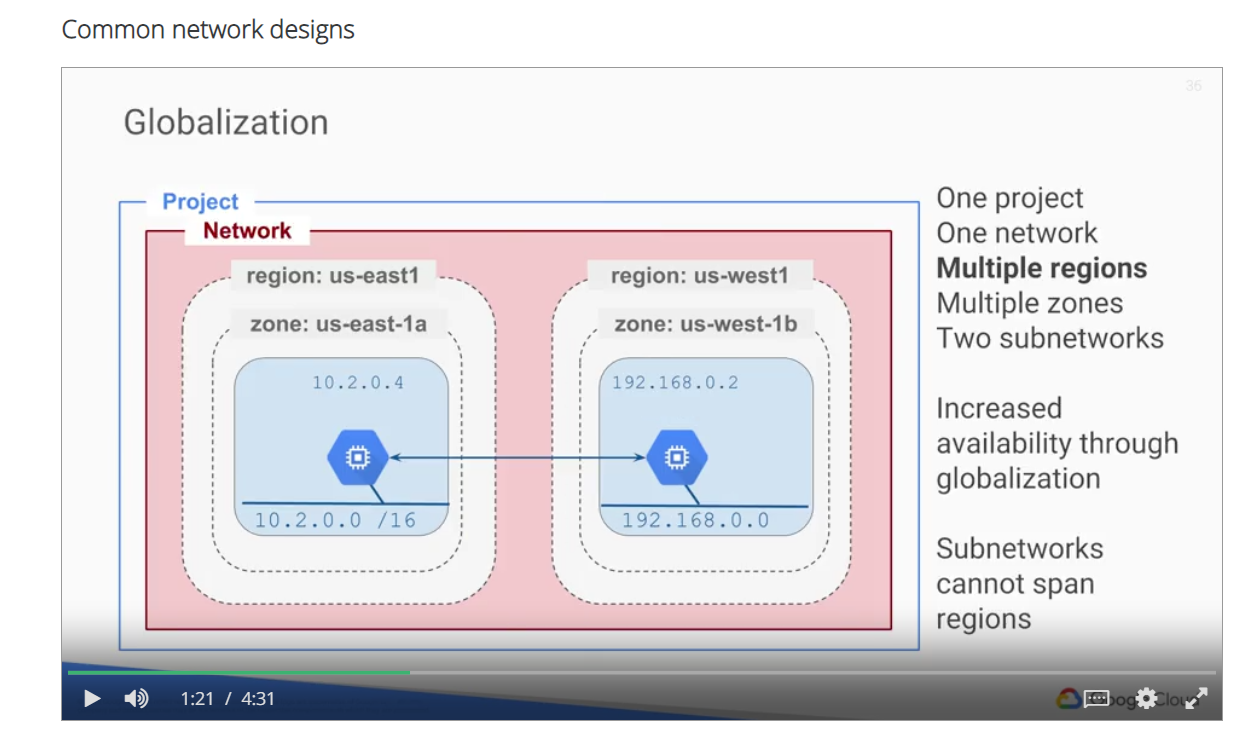


Using a single subnetwork allows you to create a firewall rule against the subnetwork 10.2.0.0. Therefore, by allocating VMs on a single subnet to separate zones, you get improved availability without

additional security complexity.

# Globalization

In the previous design, we placed resources in different zones in a region, which provides isolation from many types of infrastructure, hardware and software failures.



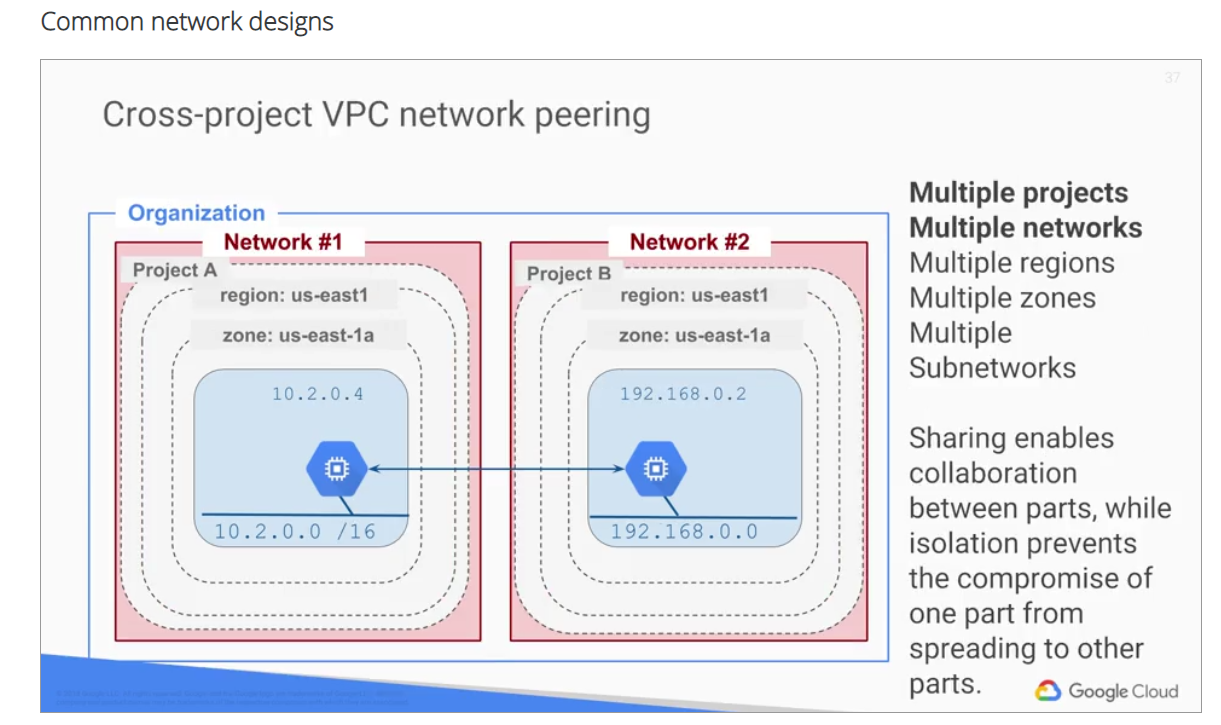
Putting resources in different regions, as shown on this slide, provides an even higher degree of fail independence. This allows you to design robust systems with resources spread across different failure domains. Load balancer Can route the traffic to the region that is closest to the user. Globalization does not get you the simplified security that we just saw with a single subnetwork; however,if the VMs are in a single network, as shown here, they can still communicate through GCP's internal global network. Let's take this separation one step further by placing resources in different regions within different

networks and different projects. These resources are now isolated, which prevents compromise of one part from spreading to other parts.

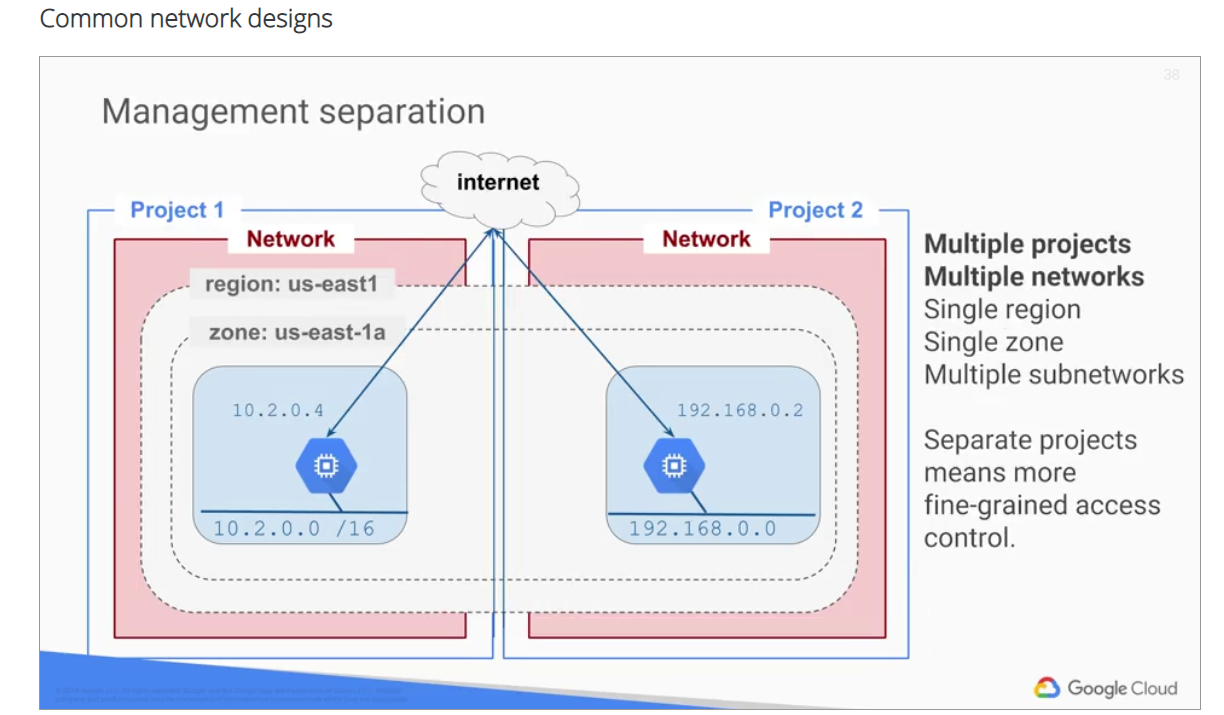
# VPC network peering

these resources can still communicate over a private address space. Finally, in this last case, the VMs

are isolated into separate projects, but within the same zone.



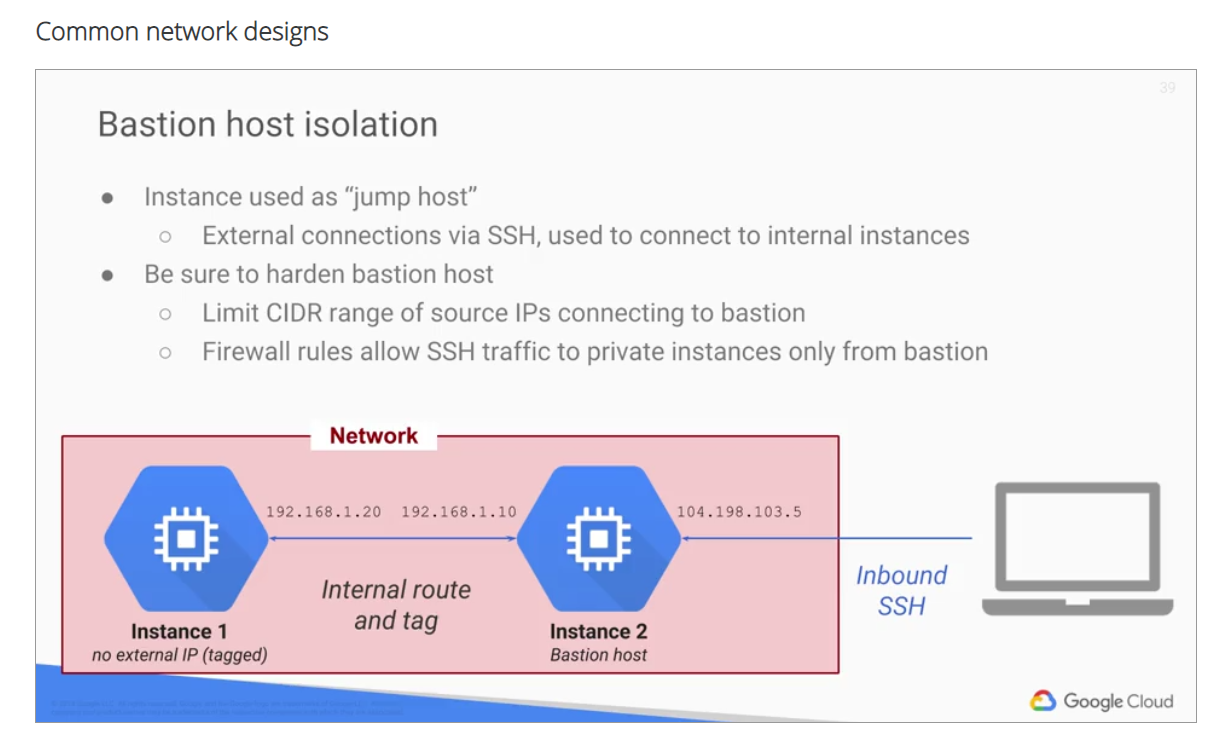
This can be useful for identity and access management, which is covered in detail in a later course.



For example, if software development is Project 1, and test engineering is Project 2, you can assign different people to different roles in the projects for management separation. Consider dividing a system up into multiple projects for better access control, but remember that a network cannot span projects. So using separate projects implies that the VMs must communicate via the Internet.

# Bastion host isolation

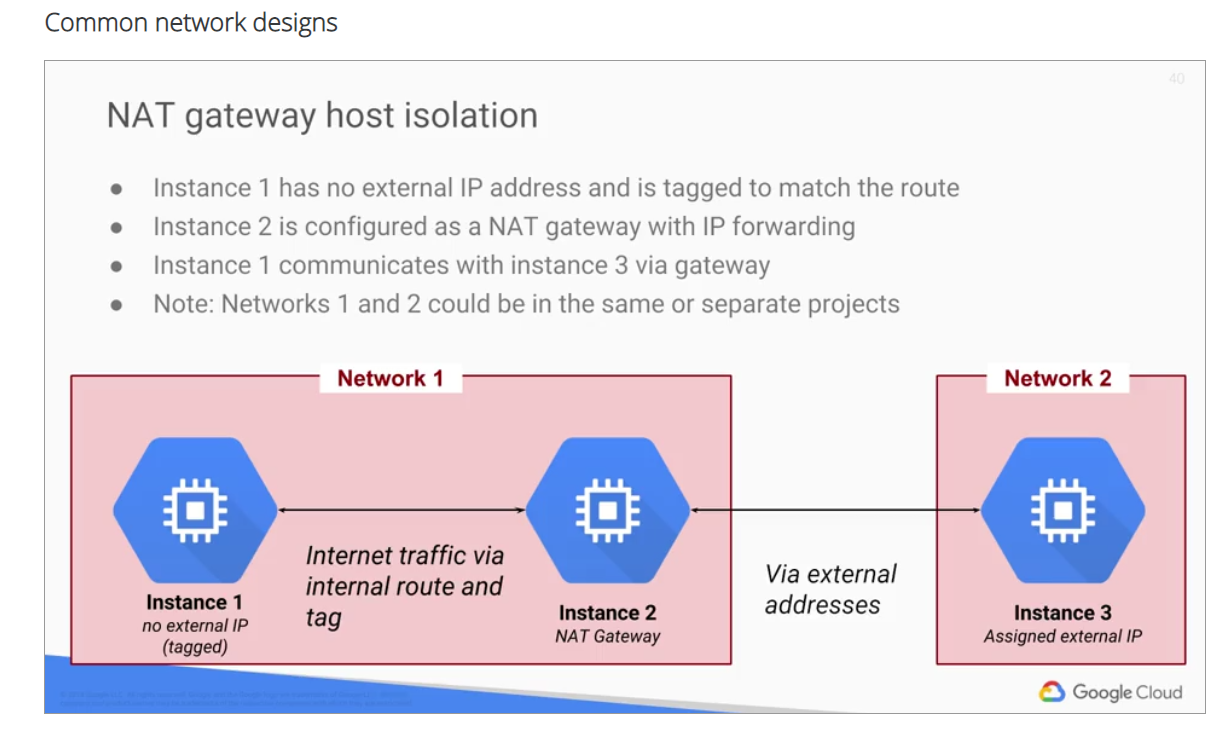
Bastion hosts provide an external face and point of entry into a network containing private network instances.



This host can provide a single point of fortification or audit, and can be started and stopped to enable or disable inbound SSH communication from the Internet. For example, on this slide, Instance 1 represents a service provided to an internal corporate audience. Therefore, this instance does not have an external IP address. In order to gain access to this instance, you can create a maintenance host known as a bastion host. You will configure such a network design in the upcoming lab and verify connectivity.

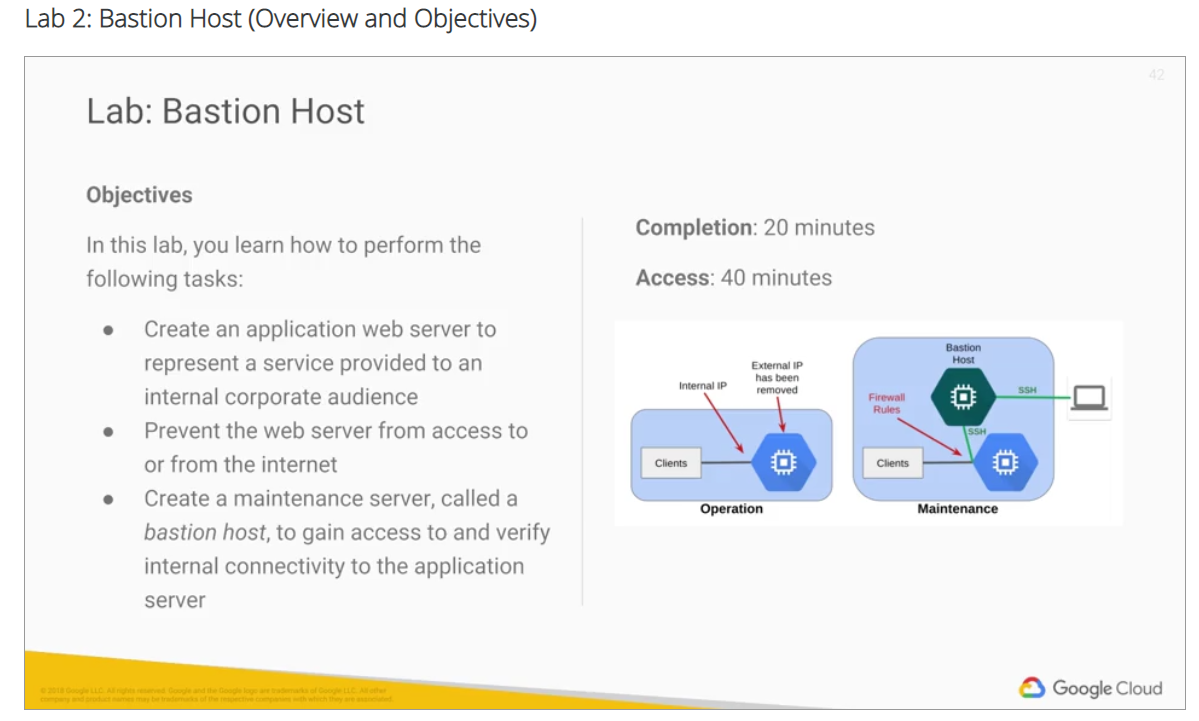
NAT gateway isolation.

Similar to the previous design, we have an instance that does not have an external IP address.



In this case, you can configure another instance in the same network as a NAT gateway with IP forwarding. This allows Instance 1 to communicate with another instance on a separate network via the gateway. The two networks do not have to be in the same project for this design to work.

# Lab



In this lab, you created a Web Server VM and restricted access to it by removing the external IP address. Then, you created a bastion host to gain access to the web server over its internal IP address. Normally, you would harden the bastion host by restricting the source IPs that can access the bastion host, bed and firewall rules. Also, when you're not using the bastion host you would shut it down. There are other security alternatives to provide routine administration access to web server like using Cloud VPN, which is covered in a later course of this specialization.

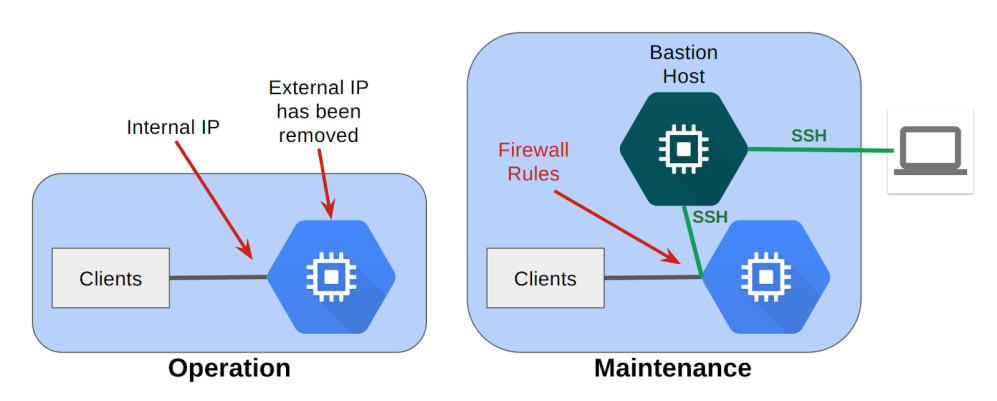
# Bastion Host v1.5

40 minutesFree

Rate Lab

## Overview

A best practice for infrastructure administration is to limit access to the resources. In this lab, you learn one method of hardening an infrastructure called a Bastion Host.



During operations, you harden the server by removing its external IP address, which prevents connections from the internet. During maintenance, you start up a bastion host that has an external IP address. You then connect via SSH to the bastion host, and from there to the server over the internal IP address. You can further restrict access with firewall rules.

## Objectives

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

* Create an application web server to represent a service provided to an internal corporate audience
* Prevent the web server from access to or from the internet
* Create a maintenance server, called a Bastion Host, to gain access to and verify internal connectivity to the application server

### What you'll need

To complete this lab, you’ll need:

* Access to a standard internet browser (Chrome browser recommended).
* Time. Note the lab’s **Completion** time in Qwiklabs. This is an estimate of the time it should take to complete all steps. Plan your schedule so you have time to complete the lab. Once you start the lab, you will not be able to pause and return later (you begin at step 1 every time you start a lab).
* The lab's **Access** time is how long your lab resources will be available. If you finish your lab with access time still available, you will be able to explore the Google Cloud Platform or work on any section of the lab that was marked "if you have time". Once the Access time runs out, your lab will end and all resources will terminate.
* You **DO NOT** need a Google Cloud Platform account or project. An account, project and associated resources are provided to you as part of this lab.
* If you already have your own GCP account, make sure you do not use it for this lab.
* If your lab prompts you to log into the console, **use only the student account provided to you by the lab**. This prevents you from incurring charges for lab activities in your personal GCP account.

### Start your lab

When you are ready, click **Start Lab**. You can track your lab’s progress with the status bar at the top of your screen.

**Important** What is happening during this time? Your lab is spinning up GCP resources for you behind the scenes, including an account, a project, resources within the project, and permission for you to control the resources needed to run the lab. This means that instead of spending time manually setting up a project and building resources from scratch as part of your lab, you can begin learning more quickly.

### Find Your Lab’s GCP Username and Password

To access the resources and console for this lab, locate the Connection Details panel in Qwiklabs. Here you will find the account ID and password for the account you will use to log in to the Google Cloud Platform:



If your lab provides other resource identifiers or connection-related information, it will appear on this panel as well.

## Task 1: Launch an instance and verify access

### Launch an instance

1. In the Console, on the **Navigation menu** ( 7a91d354499ac9f1.png), 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** | **webserver** |
| **Region** | **us-central1** |
| **Zone** | **us-central1-c** |
| **Firewall** | **Allow HTTP traffic** |

1. Click **Create**.

Click Check my progress to verify the objective.

Launch an instance

Check my progress

### Verify IP access

1. For **webserver**, click **SSH** to launch a terminal and connect.

**Tip:**Setting the Source IP at creation time is a best practice for this lab because it allows the initial SSH credentials to be set for you behind the scenes.

1. Enter a few commands to test connectivity:

ls

pwd

1. Enter the following command to close the terminal:

exit

## Task 2: Restrict firewall rule settings for SSH

The default setting for a default or auto-type network is to allow SSH access from any source IP address. Restrict access to just your source IP address to see what happens when you try to connect from the GCP Console.

### Find your IP address

Find the IP address of the computer you are using. One easy way to do this is to go to a website that provides this address.

1. Open a browser in a new tab.
2. Go to [www.google.com](http://www.google.com/) and search for "what's my IP." It will either directly reply with your IP or give you a list of sites that perform this service.
3. Ensure that the IP address only contains numerals (IPv4) and is not represented in hexadecimals (IPv6).
4. Copy your IP address. It will be referred to as YOUR\_IP\_ADDRESS. You will be using it to modify the default firewall rule.

### Edit the default SSH rule

1. In the GCP Console, on the **Navigation menu** ( 7a91d354499ac9f1.png), click **VPC network** > **Firewall rules**.
2. Click the **default-allow-ssh** rule, and then click **Edit**.
3. Specify the following, and leave the remaining settings as their defaults:

|  |  |
| --- | --- |
| **Property** | **Value**  (type value or select option as specified) |
| **Description** | **Allow SSH from my IP only** |
| **Source IP ranges** | Remove **0.0.0.0/0**  Add **[YOUR\_IP\_ADDRESS]** |

1. Click **Save**. Wait until the firewall rule is updated (the status in the bottom pane is **Updating firewall rule**; when it closes, you can continue).

### Test connectivity

1. On the **Navigation menu** ( 7a91d354499ac9f1.png), click **Compute Engine** > **VM instances**.
2. For **webserver**, click **SSH** to launch a terminal and connect.

What happened?

When you connect via SSH to an instance from your browser, you need to allow SSH from Cloud Platform resources, so you must allow connections from either any IP address or from Google's IP address range, which you can get from Public SPF records. If you want to restrict SSH access to just your IP address, you need to SSH from a terminal session.

For this lab, leaving SSH open to any connections is sufficient.

### Reset the IP address range in the firewall rule

1. In the GCP Console, on the **Navigation menu** ( 7a91d354499ac9f1.png), click **VPC network** > **Firewall rules**.
2. Click the **default-allow-ssh** rule, and then click **Edit**.
3. Specify the following, and leave the remaining settings as their defaults:

|  |  |
| --- | --- |
| **Property** | **Value**  (type value or select option as specified) |
| **Description** | **Allow SSH from all IPs** |
| **Source IP ranges** | Add **0.0.0.0/0** |

1. Click **Save**. Wait until the firewall rule is updated (the status in the bottom pane is **Updating firewall rule**; when it closes, you can continue).

### Verify the change

1. On the **Navigation menu** ( 7a91d354499ac9f1.png), click **Compute Engine** > **VM instances**.
2. For **webserver**, click **SSH** to launch a terminal and connect. Leave the terminal open for the next task.

## Task 3: Install a simple web application

Install a simple web application on your instance to represent an internal application. You then secure it by preventing access from the internet.

### Install and configure a web server

1. In the webserver SSH terminal, update the package index:

sudo apt-get update

1. Install the apache2 package:

sudo apt-get install apache2 -y

1. To create a new default web page by overwriting the default, run the following:

echo '<!doctype html><html><body><h1>Hello World!</h1></body></html>' | sudo tee /var/www/html/index.html

### Verify that the web server is working

Test that your instance is serving traffic on its external IP.

1. In the GCP Console, on the **Navigation menu** ( 7a91d354499ac9f1.png), click **Compute Engine** > **VM instances**.
2. For **webserver**, click the **external IP** to open in a new tab. You should see the "Hello World!" page you updated earlier.

## Task 4: Restrict firewall rule settings for HTTP

Restrict access to the web interface by changing the source IP address in the **default-allow-http** rule to your IP address.

### Restrict HTTP access

1. In the GCP Console, on the **Navigation menu** ( 7a91d354499ac9f1.png), click **VPC network** > **Firewall rules**.
2. Click the **default-allow-http** rule, and then click **Edit**.
3. Specify the following, and leave the remaining settings as their defaults:

|  |  |
| --- | --- |
| **Property** | **Value**  (type value or select option as specified) |
| **Description** | **Allow HTTP from my IP only** |
| **Source IP ranges** | Remove **0.0.0.0/0**  Add **[YOUR\_IP\_ADDRESS]** |

1. Click **Save**. Wait until the firewall rule is updated (the status in the bottom pane is **Updating firewall rule**; when it closes, you can continue).

### Verify that you still have access to the web server

1. On the **Navigation menu** ( 7a91d354499ac9f1.png), click **Compute Engine** > **VM instances**.
2. For **webserver**, click the **external IP** to open in a new tab. You should still see the "Hello World!" page.

## Task 5: Restrict access to the VM from the internet

### Edit the VM Properties

1. Return to the **VM instances** page of the GCP Console.
2. Click **webserver** to access the instance details.
3. Click **Edit**.
4. For **Network interfaces**, click the default network and change **External IP** from **Ephemeral** to **None**.
5. Click **Done**.
6. Click **Save**.

### Try to access the VM

1. First try HTTP: In the left pane, click **VM instances**. Notice that **webserver** doesn't have a value under **External IP**.
2. Try SSH: for **webserver**, try to use the **SSH** link to launch a terminal and connect.

What happened?

The VM is no longer associated with an External IP. It is no longer reachable from the internet.

Click Check my progress to verify the objective.

Restrict access to the VM from the internet

Check my progress

## Task 6: Create a Bastion Host

### Launch another instance

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** | **bastion** |
| **Region** | **us-central1** |
| **Zone** | **us-central1-c** |

1. Click **Create**.

Click Check my progress to verify the objective.

Create a Bastion Host

Check my progress

### Connect to the Bastion Host via SSH and verify access to webserver

1. For **bastion**, click **SSH** to launch a terminal and connect.
2. Verify that the home page on **webserver** is reachable from **bastion** by running the following command:

curl webserver

Even though **webserver** is no longer associated with an external IP address, clients inside your network can still view and use the web service on this VM over the internal IP address.

1. From the **bastion** SSH terminal, connect to **webserver** by running the following command:

ssh -a webserver

1. When prompted, type **yes** to continue.

When instances do not have external IP addresses, they can only be reached by other instances on the network or via a managed VPN gateway.

In this case, the bastion VM serves as a management and maintenance interface to the webserver VM.

## Task 7: Review

You restricted access to the **webserver** VM by removing the external IP address.

You created a bastion host named **bastion** to gain access to the webserver VM over its internal IP. Normally, you would harden the bastion host by restricting the source IPs that can access the bastion host, by editing the firewall rules just as you did earlier in this lab. When you're not using the bastion host, you can shut it down.

# Quiz

# Review

In this module, I gave you an overview of GCP's virtual private cloud. We looked at the different objects within VPC like projects, networks, IP addresses, routes and firewall rules. I also provided a brief overview of how your network design choices can affect billing. Then you applied the different concepts that we covered in a thorough lab. Next, we looked at common network designs, such as a bastion host isolation, which you got to implement in the lab. Now, that you have a solid understanding of how GCP has implemented networking, let's move on to learn more about other services. Next up is Compute Engine, which offer scalable, high-performance virtual machines.