

Getting Started with Cloud Shell and gcloud | Google Cloud Skills Boost

Qwiklabs : 21-26 minutes

GSP002



Google Cloud Self-Paced Labs

Overview

Cloud Shell provides you with command-line access to computing resources hosted on Google Cloud. Cloud Shell is a Debian-based virtual machine with a persistent 5-GB home directory, which makes it easy for you to manage your Google Cloud projects and resources. The gcloud command-line tool and other utilities you need are pre-installed in Cloud Shell, which allows you to get up and running quickly.

In this hands-on lab, you learn how to connect to computing resources hosted on Google Cloud via Cloud Shell with the gcloud tool.

You are encouraged to type the commands themselves, which reinforces the core concepts. Many labs will include a code block that contains the required commands. You can easily copy and paste the commands from the code block into the appropriate places during the lab.

What you'll do

- Practice using gcloud commands.
- Connect to compute services hosted on Google Cloud.

Prerequisites

- Familiarity with standard Linux text editors such as vim, emacs, or nano.

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 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 Console**. 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**.

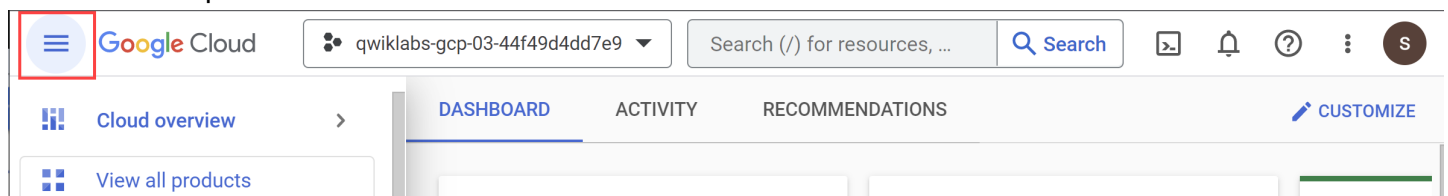
3. If necessary, copy the **Username** from the **Lab Details** panel and paste it into the **Sign in** dialog. Click **Next**.
4. Copy the **Password** from the **Lab Details** panel and paste it into the **Welcome** dialog. Click **Next**.

Important: You must use the credentials from the left panel. Do not use your Google Cloud Skills Boost credentials. **Note:** Using your own Google Cloud account for this lab may incur extra charges.

5. 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.

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**. The output contains a line that declares the **PROJECT_ID** for this session:

Your Cloud Platform project in this session is set to **YOUR_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**.

4. Your output should now look like this:

Output:

```
ACTIVE: * ACCOUNT: student-01-xxxxxxxxxxxx@qwiklabs.net To set the active account, run: $ gcloud
config set account `ACCOUNT`
```

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

After Cloud Shell is activated, you can use the command line to invoke the Cloud SDK **gcloud** tool or other tools available on the virtual machine instance. Later in the lab, you will use your **\$HOME** directory, which is used in persistent disk storage to store files across projects and between Cloud Shell sessions. Your **\$HOME** directory is private to you and cannot be accessed by other users.

Task 1. Configuring your environment

In this section, you'll learn about aspects of the development environment that you can adjust.

Understanding regions and zones

Certain [Google Compute Engine](#) resources live in regions or zones. A region is a specific geographical location where you can run your resources. Each region has one or more zones. For example, the `us-central1` region denotes a region in the Central United States that has zones `us-central1-a`, `us-central1-b`, `us-central1-c`, and `us-central1-f`. The following table shows zones in their respective regions:

Western US Central US Eastern US Western Europe Eastern Asia

<code>us-west1-a</code>	<code>us-central1-a</code>	<code>us-east1-b</code>	<code>eu-west1-b</code>	<code>asia-east1-a</code>
<code>us-west1-b</code>	<code>us-central1-b</code>	<code>us-east1-c</code>	<code>eu-west1-c</code>	<code>asia-east1-b</code>
-	<code>us-central1-c</code>	<code>us-east1-d</code>	<code>eu-west1-d</code>	<code>asia-east1-c</code>
-	<code>us-central1-f</code>	-	-	-

Resources that live in a zone are referred to as *zonal* resources. Virtual machine instances and persistent disks live in a zone. If you want to attach a persistent disk to a virtual machine instance, both resources must be in the same zone. Similarly, if you want to assign a static IP address to an instance, the instance must be in the same region as the static IP address.

Note: Learn more about regions and zones and see a complete list in Google Cloud Compute Engine's [Regions and Zones documentation](#).

1. Set the region to

```
gcloud config set compute/region {{{project_0.startup_script.project_region}}}
```

2. To view the project region setting, run the following command:

```
gcloud config get-value compute/region
```

3. Set the zone to :

```
gcloud config set compute/zone {{{project_0.startup_script.project_zone}}}
```

4. To view the project zone setting, run the following command:

```
gcloud config get-value compute/zone
```

Finding project information

1. Copy your project ID to your clipboard or text editor. The project ID is listed in 2 places:

- In the Cloud Console, on the Dashboard, under **Project info**. (Click **Navigation menu** (☰), and then click **Cloud overview > Dashboard**.)
- On the lab tab near your username and password.

2. In **Cloud Shell**, run the following `gcloud` command, to view the project id for your project:

```
gcloud config get-value project
```

3. In **Cloud Shell**, run the following `gcloud` command to view details about the project:

```
gcloud compute project-info describe --project $(gcloud config get-value project)
```

Find the zone and region metadata values in the output. You'll use the zone (google-compute-default-zone) from the output later in this lab.

Note: When the google-compute-default-region and google-compute-default-zone keys and values are missing from the output, no default zone or region is set. The output includes other useful information regarding your project. Take some time to explore this in more detail.

Setting environment variables

Environment variables define your environment and help save time when you write scripts that contain APIs or executables.

1. Create an environment variable to store your Project ID:

```
export PROJECT_ID=$(gcloud config get-value project)
```

2. Create an environment variable to store your Zone:

```
export ZONE=$(gcloud config get-value compute/zone)
```

3. To verify that your variables were set properly, run the following commands:

```
echo -e "PROJECT ID: $PROJECT_ID\nZONE: $ZONE"
```

If the variables were set correctly, the echo commands will output your Project ID and Zone.

Creating a virtual machine with the gcloud tool

Use the gcloud tool to create a new virtual machine (VM) instance.

1. To create your VM, run the following command:

```
gcloud compute instances create gcelab2 --machine-type e2-medium --zone $ZONE
```

Output:

```
Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-04-326fae68bc3d/zones/us-east1-c/instances/gcelab2]. NAME ZONE MACHINE_TYPE PREEMPTIBLE INTERNAL_IP  
EXTERNAL_IP STATUS gcelab2 {{{project_0.startup_script.project_zone}}} e2-medium 10.128.0.2  
34.67.152.90 RUNNING
```

Command details

- gcloud compute allows you to manage your Compute Engine resources in a format that's simpler than the Compute Engine API.
- instances create creates a new instance.
- gcelab2 is the name of the VM.
- The --machine-type flag specifies the machine type as e2-medium.
- The --zone flag specifies where the VM is created.

- If you omit the `--zone` flag, the `gcloud` tool can infer your desired zone based on your default properties. Other required instance settings, such as machine type and image, are set to default values if not specified in the create command.

Test completed task

Click **Check my progress** to verify your performed task. If you have successfully created a virtual machine with the `gcloud` tool, an assessment score is displayed.

Create a virtual machine with `gcloud`

- To open help for the create command, run the following command:

`gcloud compute instances create --help` **Note:** Press **Enter** or the spacebar to scroll through the help content. To exit the content, type **Q**.

Exploring `gcloud` commands

The `gcloud` tool offers simple usage guidelines that are available by adding the `-h` flag (for help) onto the end of any `gcloud` command.

1. Run the following command:

```
gcloud -h
```

You can access more verbose help by appending the `--help` flag onto a command or running the `gcloud help` command.

2. Run the following command:

```
gcloud config --help
```

To exit type **Q** and hit **Enter**.

3. Run the following command:

```
gcloud help config
```

The results of the `gcloud config --help` and `gcloud help config` commands are equivalent. Both return long, detailed help.

There are **global flags** in `gcloud` that govern the behavior of commands on a per-invocation level. Flags override any values set in SDK properties.

4. View the list of configurations in your environment:

```
gcloud config list
```

5. To see all properties and their settings:

```
gcloud config list --all
```

6. List your components:

gcloud components list

This command displays the gcloud components that are ready for you to use in this lab.

Task 2. Filtering command-line output

The gcloud command-line interface (CLI) is a powerful tool for working at the command line. You may want specific information to be displayed.

1. List the compute instance available in the project:

gcloud compute instances list **Note:** Having multiple resources deployed in a project is very common. Fortunately gcloud has some clever formatting that can help identify specific resources.

Example Output:

NAME: gcelab2 ZONE: {{{project_0.startup_script.project_zone}}} MACHINE_TYPE: e2-medium
PREEMPTIBLE: INTERNAL_IP: 10.142.0.2 EXTERNAL_IP: 35.237.43.111 STATUS: RUNNING

2. List the gcelab2 virtual machine:

```
gcloud compute instances list --filter="name=('gcelab2')"
```

Example Output:

NAME: gcelab2 ZONE: {{{project_0.startup_script.project_zone}}} MACHINE_TYPE: e2-medium
PREEMPTIBLE: INTERNAL_IP: 10.142.0.2 EXTERNAL_IP: 35.237.43.111 STATUS: RUNNING

In the above command, we have asked gcloud to only show the information matching the criteria i.e. a virtual instance name matching the criteria.

3. List the firewall rules in the project:

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

Output:

NAME	NETWORK	DIRECTION	PRIORITY	ALLOW	DENY	DISABLED
default-allow-icmp	default	INGRESS	65534	icmp	False	default-allow-internal
default-allow-internal	default	INGRESS	65534	tcp:0-65535,udp:0-65535,icmp	False	default-allow-rdp
default-allow-rdp	default	INGRESS	65534	tcp:3389	False	default-allow-ssh
default-allow-ssh	default	INGRESS	65534	tcp:22	False	dev-net-allow-ssh
dev-net-allow-ssh	dev-network	INGRESS	1000	tcp:22	False	serverless-to-vpc-connector
serverless-to-vpc-connector	dev-network	INGRESS	1000	icmp,udp:665-666,tcp:667	False	vpc-connector-egress
vpc-connector-egress	dev-network	INGRESS	1000	icmp,udp,tcp	False	vpc-connector-health-check
vpc-connector-health-check	dev-network	INGRESS	1000	tcp:667	False	vpc-connector-to-serverless
vpc-connector-to-serverless	dev-network	EGRESS	1000	icmp,udp:665-666,tcp:667	False	

4. List the firewall rules for the default network:

```
gcloud compute firewall-rules list --filter="network='default'"
```

Output:


```
NAME NETWORK DIRECTION PRIORITY ALLOW DENY DISABLED default-allow-icmp default
INGRESS 65534 icmp False default-allow-internal default INGRESS 65534 tcp:0-65535,udp:0-
65535,icmp False default-allow-rdp default INGRESS 65534 tcp:3389 False default-allow-ssh default
INGRESS 65534 tcp:22 False
```

5. List the firewall rules for the default network where the allow rule matches an ICMP rule:

```
gcloud compute firewall-rules list --filter="NETWORK:'default' AND ALLOW:'icmp'"
```

Output:

```
NAME NETWORK DIRECTION PRIORITY ALLOW DENY DISABLED default-allow-icmp default
INGRESS 65534 icmp False default-allow-internal default INGRESS 65534 tcp:0-65535,udp:0-
65535,icmp False
```

Task 3. Connecting to your VM instance

`gcloud compute` makes connecting to your instances easy. The `gcloud compute ssh` command provides a wrapper around SSH, which takes care of authentication and the mapping of instance names to IP addresses.

1. To connect to your VM with SSH, run the following command:

```
gcloud compute ssh gcelab2 --zone $ZONE
```

Output:

```
WARNING: The public SSH key file for gcloud does not exist. WARNING: The private SSH key file for
gcloud does not exist. WARNING: You do not have an SSH key for gcloud. WARNING: [/usr/bin/ssh-
keygen] will be executed to generate a key. This tool needs to create the directory
[/home/gcpstaging306_student/.ssh] before being able to generate SSH Keys. Do you want to
continue? (Y/n)
```

2. To continue, type **Y**.

```
Generating public/private rsa key pair. Enter passphrase (empty for no passphrase)
```

3. To leave the passphrase empty, press **Enter** twice.

Note: You have connected to the virtual machine created earlier in the lab. Did you notice how the command prompt changed? The prompt now says something similar to **sa_107021519685252337470@gcelab2**.

- The reference before the **@** indicates the account being used.
- After the **@** sign indicates the host machine being accessed.

4. Install **nginx** web server on to virtual machine:

```
sudo apt install -y nginx
```

5. You don't need to do anything here; so to disconnect from SSH and exit the remote shell, run the following command:

exit

You should be back at your project's command prompt.

Task 4. Updating the firewall

When using compute resources such as virtual machines, it's important to understand the associated firewall rules.

1. List the firewall rules for the project:

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

Output:

```
NAME NETWORK DIRECTION PRIORITY ALLOW DENY DISABLED default-allow-icmp default
INGRESS 65534 icmp False default-allow-internal default INGRESS 65534 tcp:0-65535,udp:0-
65535,icmp False default-allow-rdp default INGRESS 65534 tcp:3389 False default-allow-ssh default
INGRESS 65534 tcp:22 False dev-net-allow-ssh dev-network INGRESS 1000 tcp:22 False
serverless-to-vpc-connector dev-network INGRESS 1000 icmp,udp:665-666,tcp:667 False vpc-
connector-egress dev-network INGRESS 1000 icmp,udp,tcp False vpc-connector-health-check dev-
network INGRESS 1000 tcp:667 False vpc-connector-to-serverless dev-network EGRESS 1000
icmp,udp:665-666,tcp:667 False
```

From the above we can see we have two networks available. The default network is where our virtual machine gcelab2 is located.

2. Try to access the nginx service running on the gcelab2 virtual machine.

Note: Communication with the virtual machine will fail as it does not have an appropriate firewall rule. Our nginx web server is expecting to communicate on tcp:80. To get communication working we need to:

- Add a tag to the gcelab2 virtual machine
- Add a firewall rule for http traffic

3. Add a tag to the virtual machine:

```
gcloud compute instances add-tags gcelab2 --tags http-server,https-server
```

4. Update the firewall rule to allow:

```
gcloud compute firewall-rules create default-allow-http --direction=INGRESS --priority=1000 --
network=default --action=ALLOW --rules=tcp:80 --source-ranges=0.0.0.0/0 --target-tags=http-server
```

5. List the firewall rules for the project:

```
gcloud compute firewall-rules list --filter=ALLOW:'80'
```

Output:

NAME NETWORK DIRECTION PRIORITY ALLOW DENY DISABLED default-allow-http default
INGRESS 1000 tcp:80 False

6. Verify communication is possible for http to the virtual machine:

```
curl http://$(gcloud compute instances list --filter=name:gcelab2 --format='value(EXTERNAL_IP)')
```

You will see the default nginx output.

Task 5. Viewing the system logs

Viewing logs is essential to understanding the working of your project. Use gcloud to access the different logs available on Google Cloud.

1. View the available logs on the system:

```
gcloud logging logs list
```

Output:

```
NAME: projects/qwiklabs-gcp-01-4b75909db302/logs/GCEGuestAgent NAME: projects/qwiklabs-gcp-01-4b75909db302/logs/OSConfigAgent NAME: projects/qwiklabs-gcp-01-4b75909db302/logs/autoscaler.googleapis.com%2Fstatus_change NAME: projects/qwiklabs-gcp-01-4b75909db302/logs/cloudaudit.googleapis.com%2Factivity NAME: projects/qwiklabs-gcp-01-4b75909db302/logs/cloudaudit.googleapis.com%2Fdata_access NAME: projects/qwiklabs-gcp-01-4b75909db302/logs/cloudaudit.googleapis.com%2Fsystem_event NAME: projects/qwiklabs-gcp-01-4b75909db302/logs/compute.googleapis.com%2Fautoscaler NAME: projects/qwiklabs-gcp-01-4b75909db302/logs/compute.googleapis.com%2Finstance_group_manager_events NAME: projects/qwiklabs-gcp-01-4b75909db302/logs/compute.googleapis.com%2Fshielded_vm_integrity NAME: projects/qwiklabs-gcp-01-4b75909db302/logs/run.googleapis.com%2Fstderr NAME: projects/qwiklabs-gcp-01-4b75909db302/logs/run.googleapis.com%2Fstdout
```

2. View the logs that relate to compute resources:

```
gcloud logging logs list --filter="compute"
```

Output:

```
NAME: projects/qwiklabs-gcp-01-4b75909db302/logs/compute.googleapis.com%2Fautoscaler NAME: projects/qwiklabs-gcp-01-4b75909db302/logs/compute.googleapis.com%2Finstance_group_manager_events NAME: projects/qwiklabs-gcp-01-4b75909db302/logs/compute.googleapis.com%2Fshielded_vm_integrity
```

3. Read the logs related to the resource type of gce_instance:

```
gcloud logging read "resource.type=gce_instance" --limit 5
```

4. Read the logs for a specific virtual machine:

```
gcloud logging read "resource.type=gce_instance AND labels.instance_name='gcelab2'" --limit 5
```

Task 6. Testing your understanding

The following multiple-choice question should reinforce your understanding of this lab's concepts.

Congratulations!

You learned how to launch Cloud Shell and run some sample gcloud commands.

Finish your quest

This self-paced lab is part of the [Google Cloud Essentials](#) and [Using the Cloud SDK Command Line](#) quests. A quest is a series of related labs that form a learning path. Completing this quest earns you the badge above to recognize your achievement. You can make your badge (or badges) public and link to them in your online resume or social media account. Enroll in a quest and get immediate completion credit for taking this lab. Refer to the [Google Cloud Skills Boost catalog](#) for all available quests.

Take your next lab

Continue your quest with [Provision Services with Google Cloud Marketplace](#), or check out these other Google Cloud Skills Boost labs:

- [Creating a Persistent Disk](#)
- [Configuring Networks via gcloud](#)

Next steps / Learn more

- To learn more about Cloud Shell, refer to the [Cloud Shell documentation](#) or the YouTube video, [Using Google Cloud Shell](#).
- To learn more about gcloud, refer to the [gcloud documentation](#) or the YouTube video, [Getting Help with gcloud](#).

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 January 16, 2023

Lab Last Tested August 17, 2022

Copyright 2023 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.