

Getting Started with Cloud Shell & gcloud

45 minutes 1 Credit

[Rate Lab](#)

GSP002



Google Cloud Self-Paced Labs

Overview

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

In this hands-on lab you will learn how to connect to computing resources hosted on the Google Cloud via Cloud Shell with the `gcloud` command-line.

Students 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 the Google Cloud.

Prerequisites

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

Setup

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.

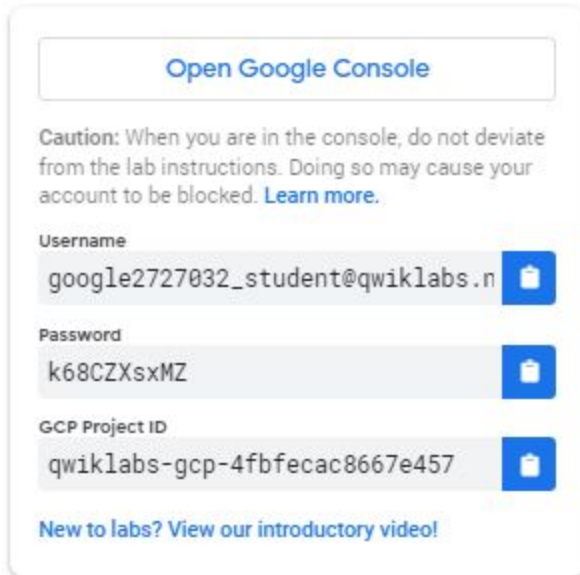
Note: If you already have your own personal Google Cloud account or project, do not use it for this lab.

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.



Open Google Console

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.](#)

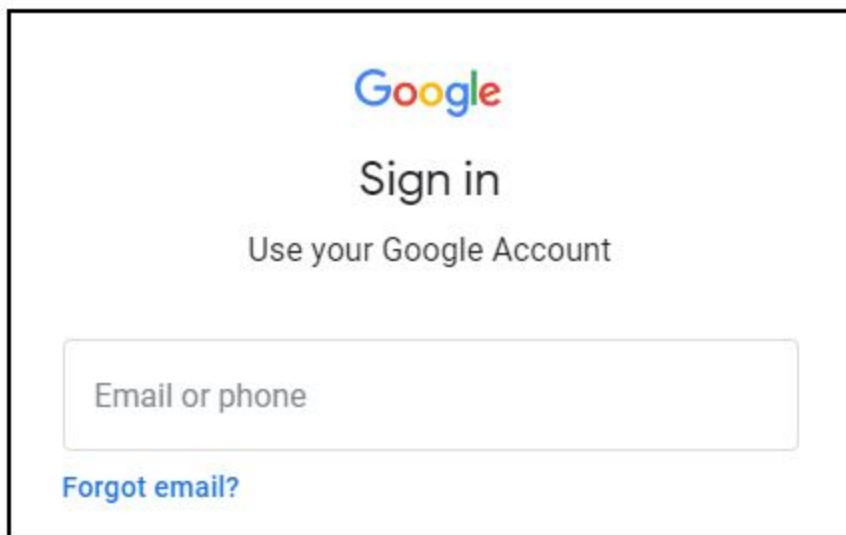
Username
google2727032_student@qwiklabs.n

Password
k68CZSxMZ

GCP Project ID
qwiklabs-gcp-4fbfecac8667e457

[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.



Google

Sign in

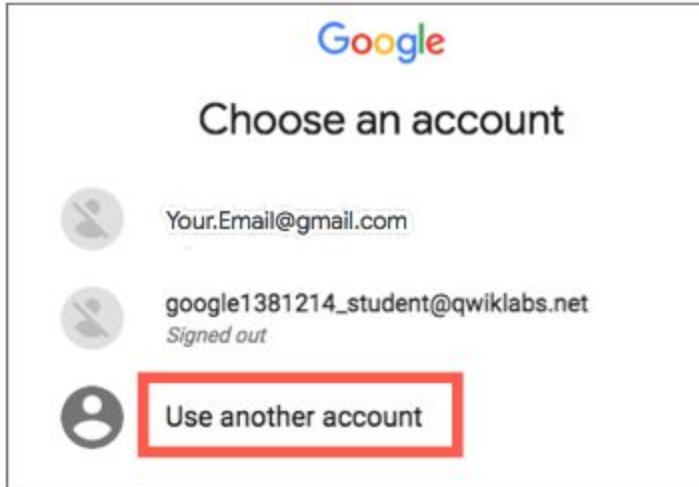
Use your Google Account

Email or phone

[Forgot email?](#)

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

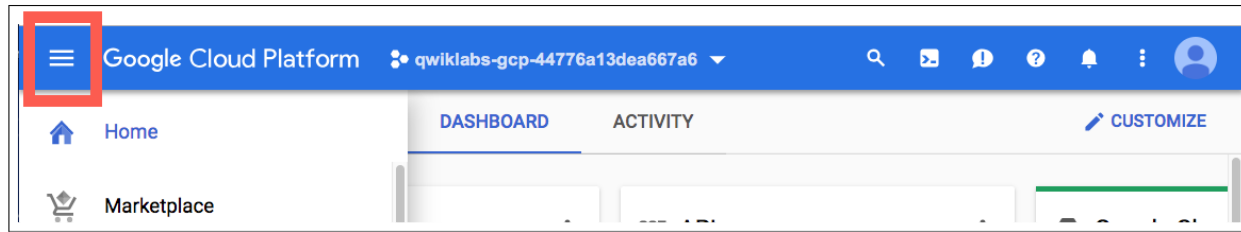
If you see the **Choose an account** page, click **Use Another Account**.



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.

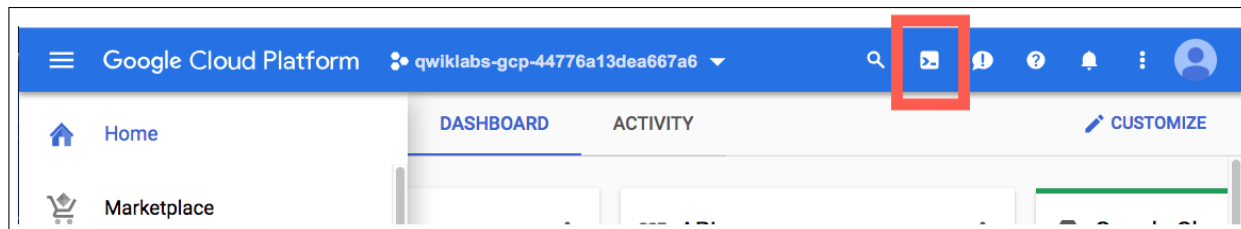
After a few moments, the Cloud Console opens in this tab.

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

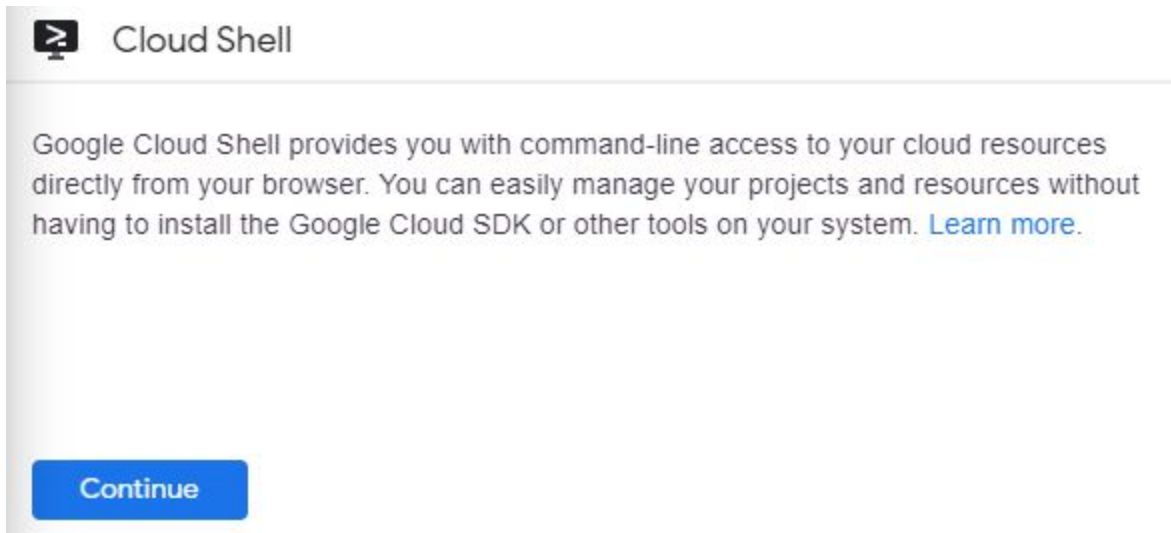


Start Cloud Shell

Open a Cloud Shell session by clicking on the icon in the top right corner of the Cloud Console:



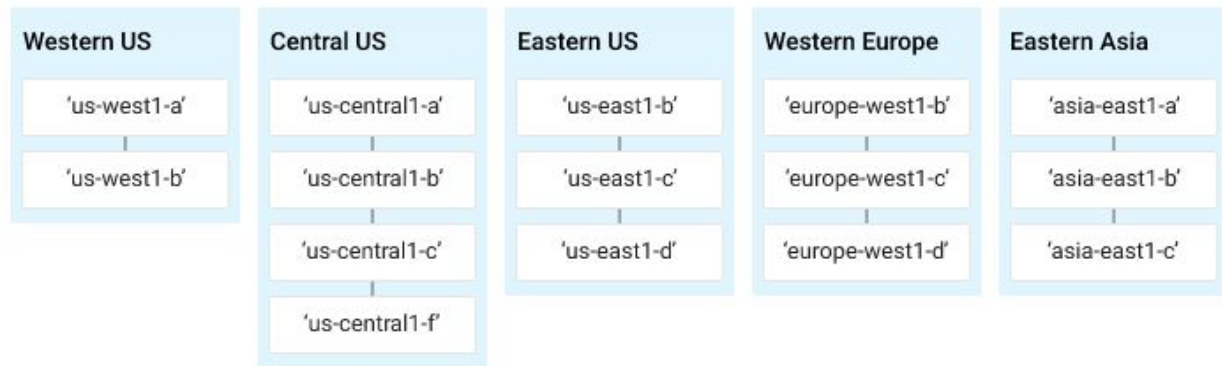
Then Click Continue:



After Cloud Shell is activated you can use the command line to invoke the Cloud SDK `gcloud` command 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.

Understanding Regions and Zones

Certain 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`.



Resources that live in a zone are referred to as zonal resources. Virtual machine Instances and persistent disks live in a zone. 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.

Learn more about regions and zones and see a complete list in [Regions & Zones documentation](#).

Default regions and zones are set by using the following values:

```
google-compute-default-zone google-compute-default-region
```

To see what your default region and zone settings are, run the following `gcloud` command, replacing `<your_project_id>` which you can see on the Home page in the Console or look in the Qwiklabs tab where you started this lab, with your Project ID:

```
gcloud compute project-info describe --project <your_project_ID>
```

You'll use the zone (`google-compute-default-zone`) from the output later in this lab.

Look for the default zone and region metadata values in the response. If the `google-compute-default-region` and `google-compute-default-zone` keys and values are missing from the response, that means no default zone or region is set.

Initializing Cloud SDK

The `gcloud` CLI is a part of the Cloud SDK. You need to download and install the SDK on your own system and initialize it (by running `gcloud init`) before you can use the `gcloud` command-line tool.

The `gcloud` CLI is automatically available in Cloud Shell. Since you're using Cloud Shell for this lab, you don't need to install `gcloud` manually.

Setting environment variables

Environment variables are variables that define your environment. Define your own variables and save yourself time when writing scripts that contain APIs or executables.

Make a couple of environment variables:

```
export PROJECT_ID=<your_project_ID>
```

Set your ZONE environment variable (use the value for zone from the earlier command):

```
export ZONE=<your_zone>
```

Verify that your variables were set properly:

```
echo $PROJECT_ID  
echo $ZONE
```

Create a virtual machine with gcloud

Create a new virtual machine instance using `gcloud`. In the following command you'll use:

- `gcloud compute` which enables you to easily manage your Compute Engine resources in a friendlier format than using the Compute Engine API.
- `instances create` creates a new instance.

Run the following to create your vm:

```
gcloud compute instances create gcelab2 --machine-type n1-standard-2 --zone $ZONE
```

- The name of the vm is "gcelab2",
- You're using the `--machine-type` flag to specify the machine type as "n1-standard-2"
- You're using the `--zone` flag to specify that it gets created in the zone you defined with your environment variable.

(Output)

NAME	ZONE	MACHINE_TYPE	PREEMPTIBLE	INTERNAL_IP	EXTERNAL_IP	STATUS
gcelab2	us-central1-a	n1-standard-2		10.128.0.2	35.184.139.176	RUNNING

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

Test Completed Task

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

Create a virtual machine with gcloud

Check my progress

You can see the default values by displaying help for the create command:

```
gcloud compute instances create --help
```

Using gcloud commands

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

Run the following command in Cloud Shell:

```
gcloud -h
```

More verbose help can be obtained by appending `--help` flag, or executing `gcloud help` command. Run the following in Cloud Shell:

```
gcloud config --help
```

Use the **Enter** key or the **Spacebar** to scroll through the help content.

Type `q` to exit the content.

Now run the following command:

```
gcloud help config
```

You can see that the `gcloud config --help` and `gcloud help config` commands are equivalent. Both give long, detailed help.

[gcloud Global Flags](#) govern the behavior of commands on a per-invocation level. Flags override any values set in SDK properties.

View the list of configurations in your environment:

```
gcloud config list
```

To check how other properties are set, see all properties by calling:

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

List your components:

```
gcloud components list
```

Here you will see what components are ready for you to use in this lab. Next you'll install a new component.

Auto-completion

`gcloud interactive` has auto prompting for commands and flags, and displays inline help snippets in the lower section as the command is typed.

Static information, like command and sub-command names, and flag names and enumerated flag values, are auto-completed using dropdown menus.

Install the beta components:

```
sudo apt-get install google-cloud-sdk
```

Enter the `gcloud` interactive mode:

```
gcloud beta interactive
```

When using the interactive mode, click on the **Tab** key to complete file path and resource arguments. If a dropdown menu appears, use the **Tab** key to move through the list, and the **Space bar** to select your choice.

Try it out! Start typing the following command, using auto-complete to finish the command:

```
gcloud compute instances describe <your_vm>
```

Across the bottom of Cloud Shell you can see the shortcut to toggle this feature.

Try out the F2 toggle:

F2:help:STATE Toggles the active help section, ON when enabled, OFF when disabled.

SSH into 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 name to IP address.

Use gcloud compute ssh to SSH into your vm:

```
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.
```

Type "Y" to continue:

```
Do you want to continue? (Y/n)
```

Press the **Enter** key through the passphrase section to leave the passphrase empty.

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

You don't need to do anything here, so disconnect from SSH by exiting from the remote shell by typing "exit":

```
exit
```

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

Use the Home directory

Now try out your Home directory. The contents of your Cloud Shell Home directory persists across projects between all Cloud Shell sessions, even after the virtual machine terminates and is restarted.

Change your current working directory:

```
cd $HOME
```

Open your `.bashrc` configuration file using `vi` text editor:

```
vi ~/.bashrc
```

The editor opens and displays the contents of the file. Press the ESC key and then `:wq` to exit the editor.

Test your Understanding

Below is a multiple choice question to reinforce your understanding of this lab's concepts. Answer to the best of your abilities.

Three basic ways to interact with the Google Cloud services and resources:

GStreamer

GLib

Command-line interface

Cloud Console

Client libraries

Submit

Congratulations!

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

Finish Your Quest



This self-paced lab is part of the Qwiklabs [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 if you've taken this lab. [See other available Qwiklabs Quests](#).

Take Your Next Lab

Continue your Quest with [Provision Services with Google Cloud Marketplace](#), or check out these suggestions:

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

Next steps / Learn More

- Cloud Shell [Documentation](#) and [tutorial video](#).
- gcloud [Documentation](#) and [tutorial video](#).

Google Cloud Training & 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 June 4, 2020

Lab Last Tested June 4, 2020

Copyright 2020 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.