Accessing the GCP Console and Cloud Shell

1 hourFree

Rate Lab

**Overview**

In this lab, you become familiar with the GCP web-based interface. Two integrated environments are available:

* A GUI environment called the GCP Console
* A command-line interface called Cloud Shell, which has the commands from the Cloud SDK pre-installed

In this course, you use both environments.

You need to know a few things about the GCP Console:

* The GCP Console is under continuous development, so the graphical layout occasionally changes. Often, these changes are made to accommodate new GCP features or changes in the technology, resulting in a slightly different workflow.
* You can perform most common GCP actions in the GCP Console. Sometimes new features are implemented in the Cloud SDK before they are made available in the GCP Console.
* The GCP Console is extremely fast for some activities. The GCP Console can perform multiple actions on your behalf that might require many command-line actions.
* The commands in the Cloud SDK are valuable tools for automation.

**Objectives**

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

* Learn how to access the GCP Console and Cloud Shell
* Become familiar with the GCP Console
* Become familiar with Cloud Shell features, including the Cloud Shell code editor
* Use the GCP Console and Cloud Shell to create buckets and VMs and service accounts
* Perform other commands in Cloud Shell

**Task 0. Lab Setup**

Access Qwiklabs

**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 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 the Google Cloud Platform for the duration of the lab.

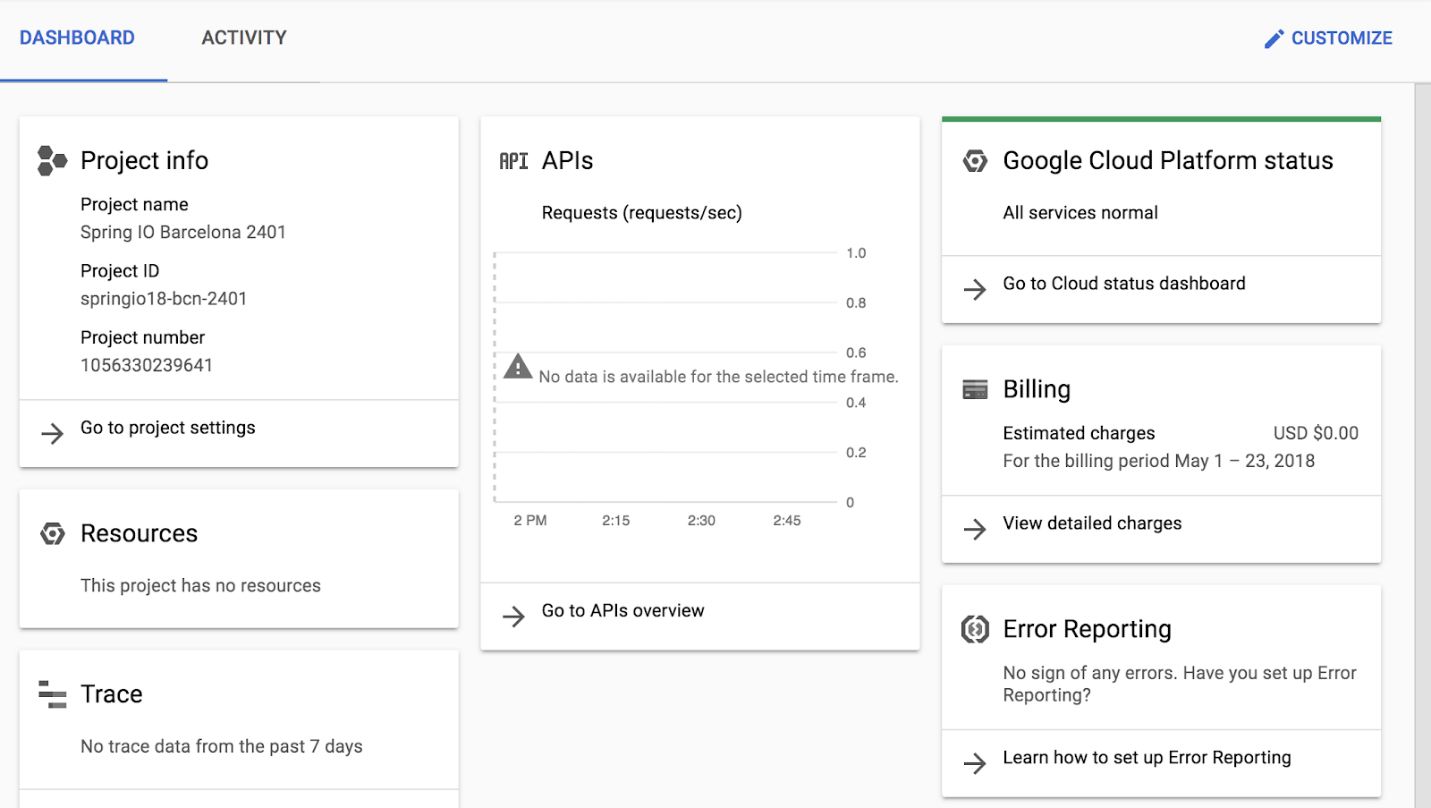
**What you need**

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 GCP account or project, do not use it for this lab.

After you complete the initial sign-in steps, the project dashboard appears.



**Task 1. Explore the GCP Console**

In this task, you explore the GCP Console and create resources.

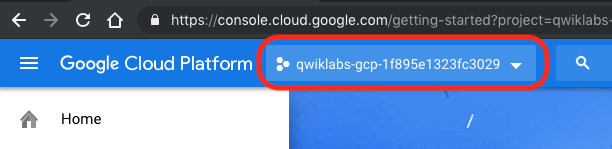
**Verify that your project is selected**

1. In the **Select a project** drop-down list in the title bar select the project ID that Qwiklabs provided with your authentication credentials.

The project ID will resemble **qwiklabs-gcp-** followed by a long hexadecimal number.

1. Click **Cancel** to close the dialog.

Your title bar should indicate the project ID as shown in the screenshot. Each lab in the Qwiklabs environment has a unique project ID, as well as unique authentication credentials.



**Navigate to Google Cloud Storage and create a bucket**

Cloud Storage allows world-wide storage and retrieval of any amount of data at any time. You can use Cloud Storage for a range of scenarios including serving website content, storing data for archival and disaster recovery, or distributing large data objects to users via direct download.

Cloud Storage buckets must have a globally unique name. In your organization, you should follow Google Cloud's [recommended best practices for naming buckets](https://cloud.google.com/storage/docs/best-practices). For this lab, we can easily get a unique name for our bucket by using the ID of the GCP project that Qwiklabs created for us, because GCP project IDs are also globally unique.

1. In the GCP Console, on the **Navigation menu** (Navigation menu), click **Home** .
2. In the **Dashboard** tab of the resulting screen, the **Project info** section shows your GCP project ID. Select and copy the project ID. Because this project ID was created for you by Qwiklabs, it will resemble **qwiklabs-gcp-** followed by a long hexadecimal number.
3. In the GCP Console, on the **Navigation menu** (Navigationmenu), click **Storage** > **Browser**.
4. Click **Create bucket**.
5. For **Name**, paste in the GCP project ID string you copied in an earlier step. Leave all other values as their defaults.

These lab instructions will later refer to the name that you typed as [BUCKET\_NAME].

1. Click **Create**.

The GCP Console has a **Notifications** (notifications icon) icon. Feedback from the underlying commands is sometimes provided there. You can click the icon to check the notifications for additional information and history.

Create a virtual machine (VM) instance

Google Compute Engine offers virtual machines running in Google's datacenters and on its network as a service. Google Kubernetes Engine makes use of Compute Engine as a component of its architecture. For this reason, it's helpful to learn a bit about Compute Engine before learning about Kubernetes Engine.

1. On the **Navigation menu** (Navigation menu ), click **Compute Engine** > **VM instances**.
2. Click **Create Instance**.
3. For **Name**, type first-vm as the name for your instance.
4. For **Region**, select **us-central1**.
5. For **Zone**, select **us-central1-c**.
6. For **Machine type**, examine the options.

The **Machine type:** menu lists the number of virtual CPUs, the amount of memory, and a symbolic name such as *n1-standard-1*. The symbolic name is the parameter you use to select the machine type when using the gcloud command to create a VM. To the right of the region, zone, and machine type is a per-month estimated cost.

1. To see the breakdown of estimated costs, click **Details** to the right of the **Machine type** list underneath the estimated costs.
2. For **Machine type**, click **2 vCPUs (n1-standard-2)**.

How did the cost change?

1. For **Machine type,** click **micro (1 shared vCPU)**.

The micro type is a shared-core VM that is inexpensive.

1. For **Firewall**, click **Allow HTTP traffic**.
2. Leave the remaining settings as their defaults, and click **Create**.

Wait until the new VM is created.

**Explore the VM details**

1. On the **VM instances** page, click the name of your VM, first-vm.
2. Locate **CPU platform**, notice the value, and click **Edit**.

You can't change the machine type, the CPU platform, or the zone of a running GCP VM. You can add network tags and allow specific network traffic from the internet through firewalls.

Some properties of a VM are integral to the VM and are established when the VM is created. They cannot be changed. Other properties can be edited. For example, you can add disks, and you can determine whether the boot disk is deleted when the instance is deleted.

1. Scroll down and examine **Availability policies**.

Compute Engine offers preemptible VM instances, which cost less per hour but can be terminated by GCP at any time. These preemptible instances can save you a lot of money, but you must make sure that your workloads are suitable to be interrupted. You can't convert a non-preemptible instance into a preemptible one. This choice must be made at VM creation.

If a VM is stopped for any reason (for example, an outage or a hardware failure), the automatic restart feature starts it back up. Is this the behavior you want? Are your applications idempotent (written to handle a second startup properly)?

During host maintenance, the VM is set for live migration. However, you can have the VM terminated instead of migrated.

If you make changes, they can sometimes take several minutes to be implemented, especially if they involve networking changes, like adding firewalls or changing the external IP.

1. Click **Cancel**.

**Create an IAM service account**

An IAM service account is a special type of Google account that belongs to an application or a virtual machine, instead of to an individual end user.

1. On the **Navigation menu**, click **IAM & admin** > **Service accounts**.
2. Click **+ Create service account**.
3. On the **Service account details** page, specify the **Service account name** as test-service-account.
4. Click **Create**.
5. On the **Service account permissions** page, specify the role as **Project** > **Editor**.
6. Click **Continue**.
7. Click **+ Create Key**.
8. Select **JSON** as the key type.
9. Click **Create**.

A JSON key file is downloaded. In a later step, you find this key file and upload it to the VM.

1. Click **Close**.
2. Click **Done**.

Click *Check my progress* to verify the objective.

Create a bucket, VM instance with necessary firewall rule and an IAM service account.

Check my progress

**Task 2. Explore Cloud Shell**

Cloud Shell provides you with command-line access to your cloud resources directly from your browser. With Cloud Shell, Cloud SDK command-line tools such as gcloud are always available, up to date, and fully authenticated.

Cloud Shell provides the following features and capabilities:

* Temporary Compute Engine VM
* Command-line access to the instance through a browser
* 5 GB of persistent disk storage ($HOME dir)
* Preinstalled Cloud SDK and other tools
* gcloud: for working with Compute Engine, Google Kubernetes Engine (GKE) and many GCP services
* gsutil: for working with Cloud Storage
* kubectl: for working with GKE and Kubernetes
* bq: for working with BigQuery
* Language support for Java, Go, Python, Node.js, PHP, and Ruby
* Web preview functionality
* Built-in authorization for access to resources and instances

After 1 hour of inactivity, the Cloud Shell instance is recycled. Only the /home directory persists. Any changes made to the system configuration, including environment variables, are lost between sessions.

In this task, you use Cloud Shell to create and examine some resources.

**Open Cloud Shell and explore its features**

1. On the GCP Console title bar, click **Activate Cloud Shell** (e92fcd01cbb5e0ff.png).
2. When prompted, click **Start Cloud Shell**.

Cloud Shell opens at the bottom of the GCP Console window.

The following icons are on the far right of Cloud Shell toolbar:

* **Hide/Restore:** This icon hides and restores the window, giving you full access to the GCP Console without closing Cloud Shell.
* **Open in new window:** Having Cloud Shell at the bottom of the GCP Console is useful when you are issuing individual commands. But when you edit files or want to see the full output of a command, clicking this icon displays Cloud Shell in a full-sized terminal window.
* **Close all tabs:** This icon closes Cloud Shell. Everytime you close Cloud Shell, the virtual machine is recycled and all machine context is lost. However, data that you stored in your home directory is still available to you the next time you start Cloud Shell.

**Use Cloud Shell to set up the environment variables for this task**

In Cloud Shell, use the following commands to define the environment variables used in this task.

1. Replace [BUCKET\_NAME] with the name of the first bucket from task 1.
2. Replace [BUCKET\_NAME\_2] with a globally unique name.

You can append a 2 to the globally unique bucket name that you used previously.

MY\_BUCKET\_NAME\_1=[BUCKET\_NAME]

MY\_BUCKET\_NAME\_2=[BUCKET\_NAME\_2]

MY\_REGION=us-central1

When you are working in the Cloud Shell or writing scripts, creating environment variables is a good practice. You can easily and consistently re-use these environment variables, which makes your work less error-prone.

Make sure you replace the full placeholder string, such as [BUCKET\_NAME]with the unique name that you choose, for example MY\_BUCKET\_NAME\_1=unique\_bucket\_name.

**Move the credentials file you created earlier into Cloud Shell**

You downloaded a JSON-encoded credentials file in an earlier task when you created your first Cloud IAM service account.

1. On your local workstation, locate the JSON key that you just downloaded and rename the file to credentials.json.
2. In Cloud Shell, click the three dots ( Three-dot menu icon) icon in the Cloud Shell toolbar to display further options.
3. Click **Upload file** and upload the credentials.json file from your local machine to the Cloud Shell VM.
4. Click the **X** icon to close the file upload pop-up window.
5. In Cloud Shell, type **ls** and press ENTER to confirm that the file was uploaded.

**Create a second Cloud Storage bucket and verify it in the GCP Console**

The gsutil command, which is supplied by the Cloud SDK, lets you work with Cloud Storage from the command line. In this task, you use the gsutil command in Cloud Shell.

1. In Cloud Shell, use the gsutil command to create a bucket.

gsutil mb gs://$MY\_BUCKET\_NAME\_2

1. In the GCP Console, on the **Navigation menu** (Navigation menu), click **Storage** > **Browser**, or click **Refresh** if you are already in the Storage Browser.

The second bucket should appear in the **Buckets** list.

**Use the gcloud command line to create a second virtual machine**

1. In Cloud Shell, execute the following command to list all the zones in a given region:

gcloud compute zones list | grep $MY\_REGION

1. Select a zone from the first column of the list. Notice that GCP zones' names consist of their region name, followed by a hyphen and a letter.

You may choose a zone that is the same as or different from the zone that you used for the first VM in task 1.

1. Execute the following command to store your chosen zone in an environment variable.

You replace [ZONE] with your selected zone.

MY\_ZONE=[ZONE]

1. Set this zone to be your default zone by executing the following command.

gcloud config set compute/zone $MY\_ZONE

1. Execute the following command to store a name in an environment variable you will use to create a VM. You will call your second VM second-vm.

MY\_VMNAME=second-vm

1. Create a VM in the default zone that you set earlier in this task using the new environment variable to assign the VM name.

gcloud compute instances create $MY\_VMNAME \

--machine-type "n1-standard-1" \

--image-project "debian-cloud" \

--image-family "debian-9" \

--subnet "default"

1. List the virtual machine instances in your project.

gcloud compute instances list

You will see both your newly created and your first virtual machine in the list.

1. In the GCP Console, on the **Navigation menu** ( Navigation menu), click **Compute Engine** > **VM Instances**. Just as in the output of gcloud compute instances list, you will see both of the virtual machines you created.
2. Look at the External IP column. Notice that the external IP address of the first VM you created is shown as a link. (If necessary, click the HIDE INFO PANEL button to reveal the External IP column.) The GCP Console offers the link because you configured this VM's firewall to allow HTTP traffic.
3. Click the link you found in your VM's External IP column. Your browser will present a Connection refused message in a new browser tab. This message occurs because, although there is a firewall port open for HTTP traffic to your VM, no Web server is running there. Close the browser tab you just created.

**Use the gcloud command line to create a second service account**

1. In Cloud Shell, execute the following command to create a new service account:

gcloud iam service-accounts create test-service-account2 --display-name "test-service-account2"

If you see the following output, type **y** and press **ENTER**:

**Output (do not copy)**

API [iam.googleapis.com] not enabled on project [560255523887]. Would

you like to enable and retry (this will take a few minutes)? (y/N)?

1. In the GCP Console, on the **Navigation menu** (Navigation menu), click **IAM & admin** > **Service accounts**.

Refresh the page till you see **test-service-account2**.

Click *Check my progress* to verify the objective.

Create a second bucket, VM instance and an IAM service account.

Check my progress

1. In Cloud Shell, execute the following command to grant the second service account the Project viewer role:

gcloud projects add-iam-policy-binding $GOOGLE\_CLOUD\_PROJECT --member serviceAccount:test-service-account2@${GOOGLE\_CLOUD\_PROJECT}.iam.gserviceaccount.com --role roles/viewer

GOOGLE\_CLOUD\_PROJECT is an environment variable that is automatically populated in Cloud Shell and is set to the project ID of the current context.

1. In the GCP Console, on the **Navigation menu** (Navigation menu), click **IAM & admin** > **IAM**. Select the new service account called **test-service-account2**.
2. On the right hand side of the page, click on pencil icon and expand the Viewer role.

You will see **test-service-account2** listed as a member of the Viewer role.

**Task 3. Work with Cloud Storage in Cloud Shell**

**Download a file to Cloud Shell and copy it to Cloud Storage**

1. Copy a picture of a cat from a Google-provided Cloud Storage bucket to your Cloud Shell.

gsutil cp gs://cloud-training/ak8s/cat.jpg cat.jpg

1. Copy the file into one of the buckets that you created earlier.

gsutil cp cat.jpg gs://$MY\_BUCKET\_NAME\_1

1. Copy the file from the first bucket into the second bucket:

gsutil cp gs://$MY\_BUCKET\_NAME\_1/cat.jpg gs://$MY\_BUCKET\_NAME\_2/cat.jpg

1. In the GCP Console, on the **Navigation menu** (Navigation menu), click **Storage** > **Browser**, select the buckets that you created, and verify that both contain the cat.jpg file.

**Set the access control list for a Cloud Storage object**

1. To get the default access list that's been assigned to cat.jpg (when you uploaded it to your Cloud Storage bucket), execute the following two commands:

gsutil acl get gs://$MY\_BUCKET\_NAME\_1/cat.jpg > acl.txt

cat acl.txt

The output should look like the following example, but with different numbers. This output shows that anyone with a Project Owner, Editor, or Viewer role for the project has access (Owner access for Owners/Editors and Reader access for Viewers).

**Output (do not copy)**

[

{

"entity": "project-owners-560255523887",

"projectTeam": {

"projectNumber": "560255523887",

"team": "owners"

},

gsutil cp gs://cloud-training/ak8s/cat.jpg cat.jpg "role": "OWNER"

},

{

"entity": "project-editors-560255523887",

"projectTeam": {

"projectNumber": "560255523887",

"team": "editors"

},

"role": "OWNER"

},

{

"entity": "project-viewers-560255523887",

"projectTeam": {

"projectNumber": "560255523887",

"team": "viewers"

},

"role": "READER"

},

{

"email": "google12345678\_student@qwiklabs.net",

"entity": "user-google12345678\_student@qwiklabs.net",

"role": "OWNER"

}

]

1. To change the object to have private access, execute the following command:

gsutil acl set private gs://$MY\_BUCKET\_NAME\_1/cat.jpg

1. To verify the new ACL that's been assigned to cat.jpg, execute the following two commands:

gsutil acl get gs://$MY\_BUCKET\_NAME\_1/cat.jpg > acl-2.txt

cat acl-2.txt

The output should look similar to the following example. Now only the original creator of the object (your lab account) has Owner access.

**Output (do not copy)**

[

{

"email": "google12345678\_student@qwiklabs.net",

"entity": "user-google12345678\_student@qwiklabs.net",

"role": "OWNER"

}

]

**Authenticate as a service account in Cloud Shell**

1. In Cloud Shell, execute the following command to view the current configuration:

gcloud config list

You should see output that looks like the following example. In your output, the zone should be equal to the zone that you set when you created your second VM in task 2. The account and project should match your Qwiklabs lab credentials.

**Output (do not copy)**

[component\_manager]

disable\_update\_check = True

[compute]

gce\_metadata\_read\_timeout\_sec = 5

zone = us-central1-a

[core]

account = google12345678\_student@qwiklabs.net

disable\_usage\_reporting = False

project = qwiklabs-gcp-1aeffbc5d0acb416

[metrics]

environment = devshell

Your active configuration is: [cloudshell-16441]

1. In Cloud Shell, execute the following command to change the authenticated user to the first service account (which you created in an earlier task) through the credentials that you downloaded to your local machine and then uploaded into Cloud Shell (credentials.json).

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

Cloud Shell is now authenticated as test-service-account.

1. To verify the active account, execute the following command:

gcloud config list

You should see output that looks like the following example. The account is now set to the test-service-account service account.

**Output (do not copy)**

[component\_manager]

disable\_update\_check = True

[compute]

gce\_metadata\_read\_timeout\_sec = 5

zone = us-central1-a

[core]

account = test-service-account@qwiklabs-gcp-1aeffbc5d0acb416.iam.gserviceaccount.com

disable\_usage\_reporting = False

project = qwiklabs-gcp-1aeffbc5d0acb416

[metrics]

environment = devshell

Your active configuration is: [cloudshell-16441]

1. To verify the list of authorized accounts in Cloud Shell, execute the following command:

gcloud auth list

You should see output that looks like the following example.

**Output (do not copy)**

Credentialed Accounts

ACTIVE ACCOUNT

google12345678\_student@qwiklabs.net

\* test-service-account@qwiklabs-gcp-1aeffbc5d0acb416.iam.gserviceaccount.com

To set the active account, run:

$ gcloud config set account `ACCOUNT`

1. To verify that the current account (test-service-account) cannot access the cat.jpg file in the first bucket that you created, execute the following command:

gsutil cp gs://$MY\_BUCKET\_NAME\_1/cat.jpg ./cat-copy.jpg

Because you restricted access to this file to the owner earlier in this task you should see output that looks like the following example.

**Output (do not copy)**

Copying gs://test-bucket-123/cat.jpg...

AccessDeniedException: 403 KiB]

1. Verify that the current account (test-service-account) can access the cat.jpg file in the second bucket that you created:

gsutil cp gs://$MY\_BUCKET\_NAME\_2/cat.jpg ./cat-copy.jpg

Because access has not been restricted to this file you should see output that looks like the following example.

**Output (do not copy)**

Copying gs://test-bucket-123/cat.jpg...

- [1 files][ 81.7 KiB/ 81.7 KiB]

Operation completed over 1 objects/81.7 KiB.

1. To switch to the lab account, execute the following command.

You replace [USERNAME] with the username provided in the Qwiklabs Connection Details pane on the left of the lab instructions page. .

gcloud config set account [USERNAME]

1. To verify that you can access the cat.jpg file in the [BUCKET\_NAME] bucket (the first bucket that you created), execute the following command.

gsutil cp gs://$MY\_BUCKET\_NAME\_1/cat.jpg ./copy2-of-cat.jpg

You should see output that looks like the following example. The lab account created the bucket and object and remained an Owner when the object access control list (ACL) was converted to private, so the lab account can still access the object.

**Output (do not copy)**

Copying gs://test-bucket-123/cat.jpg...

- [1 files][ 81.7 KiB/ 81.7 KiB]

Operation completed over 1 objects/81.7 KiB.

1. Make the first Cloud Storage bucket readable by everyone, including unauthenticated users.

gsutil iam ch allUsers:objectViewer gs://$MY\_BUCKET\_NAME\_1

This is an appropriate setting for hosting public website content in Cloud Storage.

1. In the GCP Console, on the **Navigation menu** (Navigation menu), click **Storage** > **Browser**, select the first storage bucket that you created. Notice that the cat.jpg file has a Public link. Copy this link.
2. Open an incognito browser tab and paste the link into its address bar. You will see a picture of a cat. Leave this browser tab open.

Click *Check my progress* to verify the objective.

Work with the Cloud Storage in Cloud Shell.

Check my progress

**Task 4. Explore the Cloud Shell code editor**

In this task, you explore using the Cloud Shell code editor.

**Open the Cloud Shell code editor**

1. In Cloud Shell, click the pencil icon on the top right.

Cloud Shell Code Editor icon

A new tab opens with the Cloud Shell Code editor and the Cloud Shell. The GCP console remains on the original tab. You can switch between the GCP Console and Cloud Shell by clicking the tab.

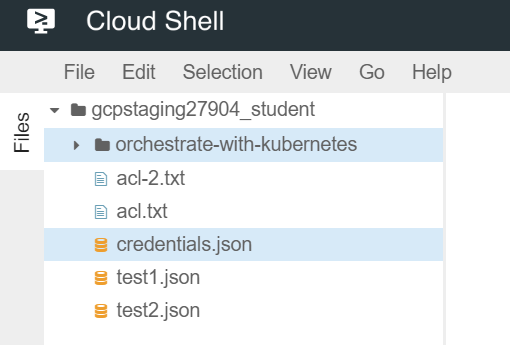
1. In Cloud Shell, execute the following command to clone a git repository:

git clone https://github.com/googlecodelabs/orchestrate-with-kubernetes.git

The orchestrate-with-kubernetes folder appears in the left pane of the Cloud Shell code editor window.

**Note:**

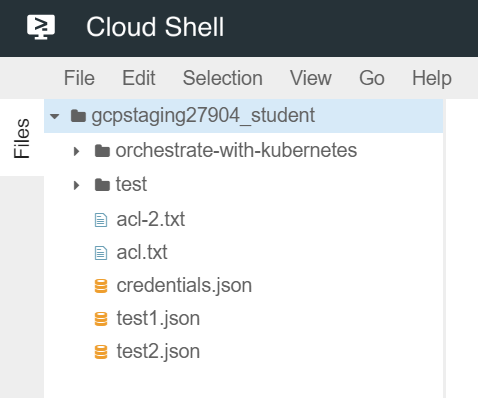
If you are using the older Cloud Shell code editor you will need to click **File**> **Refresh**in the Cloud Shell code editor menu to see the new folder.



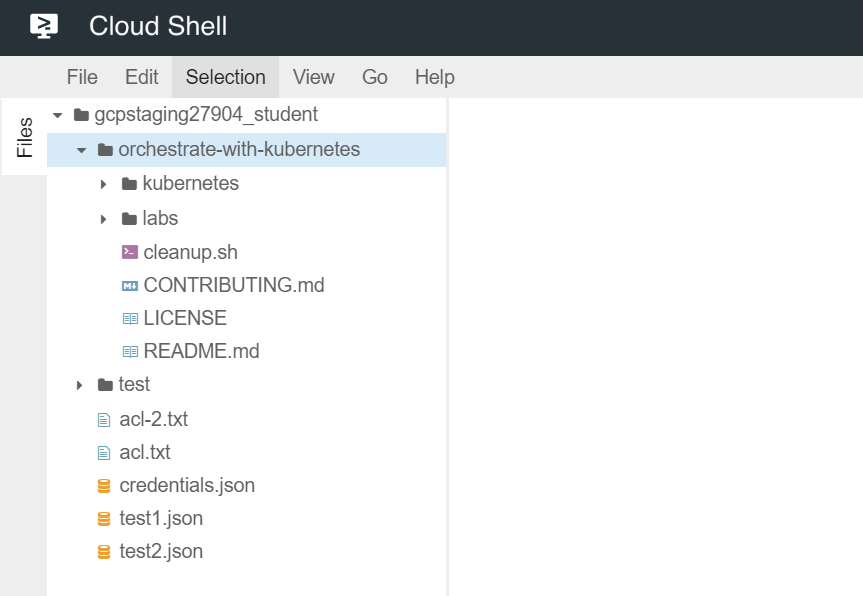
1. In Cloud Shell, execute the following command to create a test directory:

mkdir test

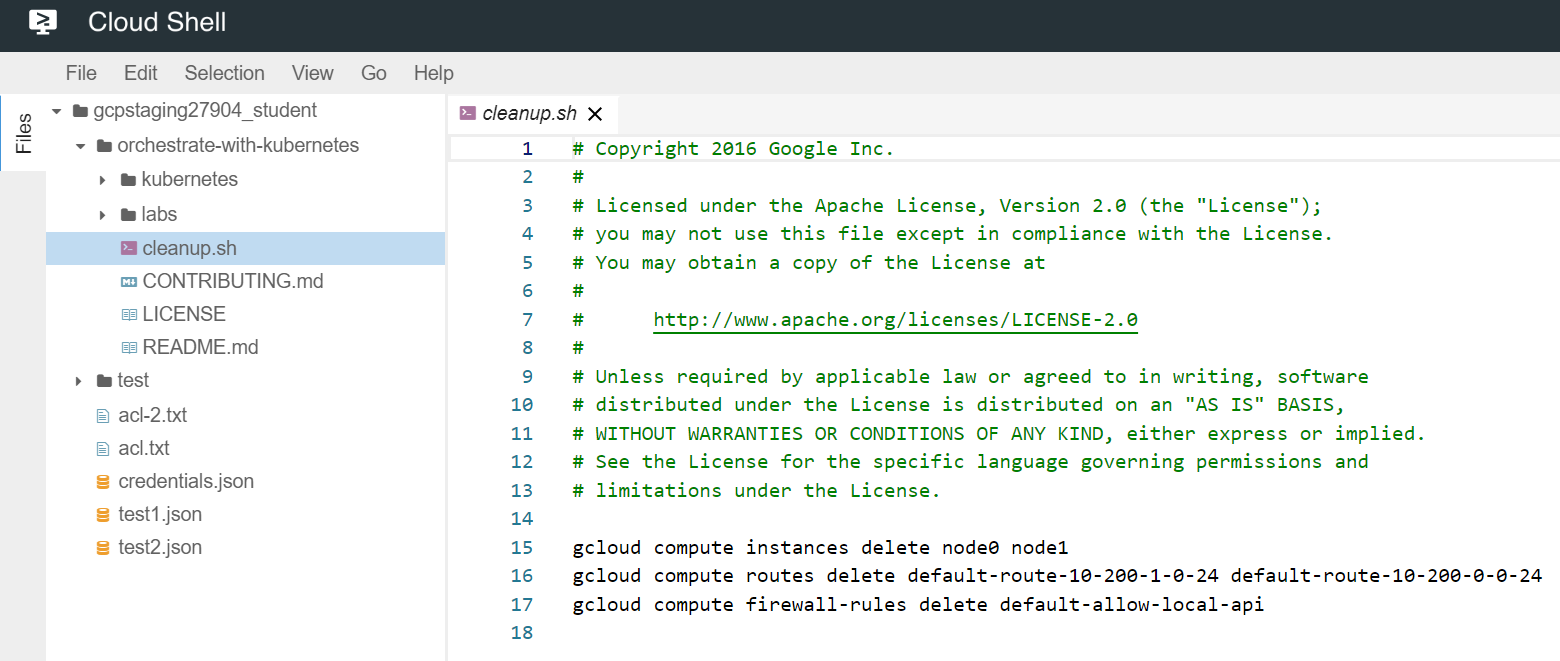
The test folder now appears in the left pane of the Cloud Shell code editor window.



1. In the Cloud Shell code editor, click the arrow to the left of orchestrate-with-kubernetes to expand the folder.



1. Click the cleanup.sh file to open it in the right pane of the Cloud Shell code editor window.



1. Add the following text as the last line of the cleanup.sh file:

echo Finished cleanup!

No action is necessary to save your work.

1. In Cloud Shell, execute the following commands to change directory and display the contents of cleanup.sh:

cd orchestrate-with-kubernetes

cat cleanup.sh

1. Verify that the output of cat cleanup.sh includes the line of text that you added.
2. In the Cloud Shell code editor, click to open the File menu and choose New File. Name the file index.html.
3. In the right hand pane, paste in this HTML text:

<html><head><title>Cat</title></head>

<body>

<h1>Cat</h1>

<img src="REPLACE\_WITH\_CAT\_URL">

</body></html>

Use your local computer's keyboard shortcut to paste: `Cmd-V` for a Mac, `Ctrl-V` for a Windows or Linux machine.

1. Replace the string REPLACE\_WITH\_CAT\_URL with the URL of the cat image from an earlier task. The URL will look like this:

**Example (do not copy)**

https://storage.googleapis.com/qwiklabs-gcp-1aeffbc5d0acb416/cat.jpg

1. On the **Navigation menu** (Navigation menu ), click **Compute Engine** > **VM instances**.
2. In the row for your first VM, click the SSH button.
3. In the SSH login window that opens on your VM, install the nginx Web server:

sudo apt-get update

sudo apt-get install nginx

1. In your Cloud Shell window, copy the HTML file you created using the Code Editor to your virtual machine:

gcloud compute scp index.html first-vm:index.nginx-debian.html --zone=us-central1-c

If you are prompted whether to add a host key to your list of known hosts, answer **y**.

If you are prompted to enter a passphrase, press the **Enter** key to respond with an empty passphrase. Press the **Enter** key again when prompted to confirm the empty passphrase.

1. In the SSH login window for your VM, copy the HTML file from your home directory to the document root of the nginx Web server:

sudo cp index.nginx-debian.html /var/www/html

Click *Check my progress* to verify the objective.

Install the nginx Web server and customize the welcome page.

Check my progress

1. On the **Navigation menu** (Navigation menu ), click **Compute Engine** > **VM instances**. Click the link in the External IP column for your first VM. A new browser tab opens, containing a Web page that contains the cat image.