**Accessing the Google Cloud Console and Cloud Shell**

experimentLabschedule1 houruniversal\_currency\_alt5 Creditsshow\_chartIntroductory

**Overview**

In this lab, you become familiar with Google Cloud's web-based interface. Two integrated environments are available:

* A GUI environment called the Google Cloud 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 Google Cloud console:

* The Google Cloud console is under continuous development, so the graphical layout occasionally changes. These changes are often made to accommodate new Google Cloud features or changes in the technology, resulting in a slightly different workflow.
* You can perform most common Google Cloud actions in the Google Cloud console. Sometimes new features are implemented in the Cloud SDK before they are made available in the Google Cloud console.
* The Google Cloud console is extremely fast for some activities. The Google Cloud 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 Google Cloud console and Cloud Shell
* Become familiar with the Google Cloud console
* Become familiar with Cloud Shell features, including the Cloud Shell Editor
* Use the Google Cloud console and Cloud Shell to create buckets and VMs and service accounts
* Perform other commands in Cloud Shell

**Lab Setup**

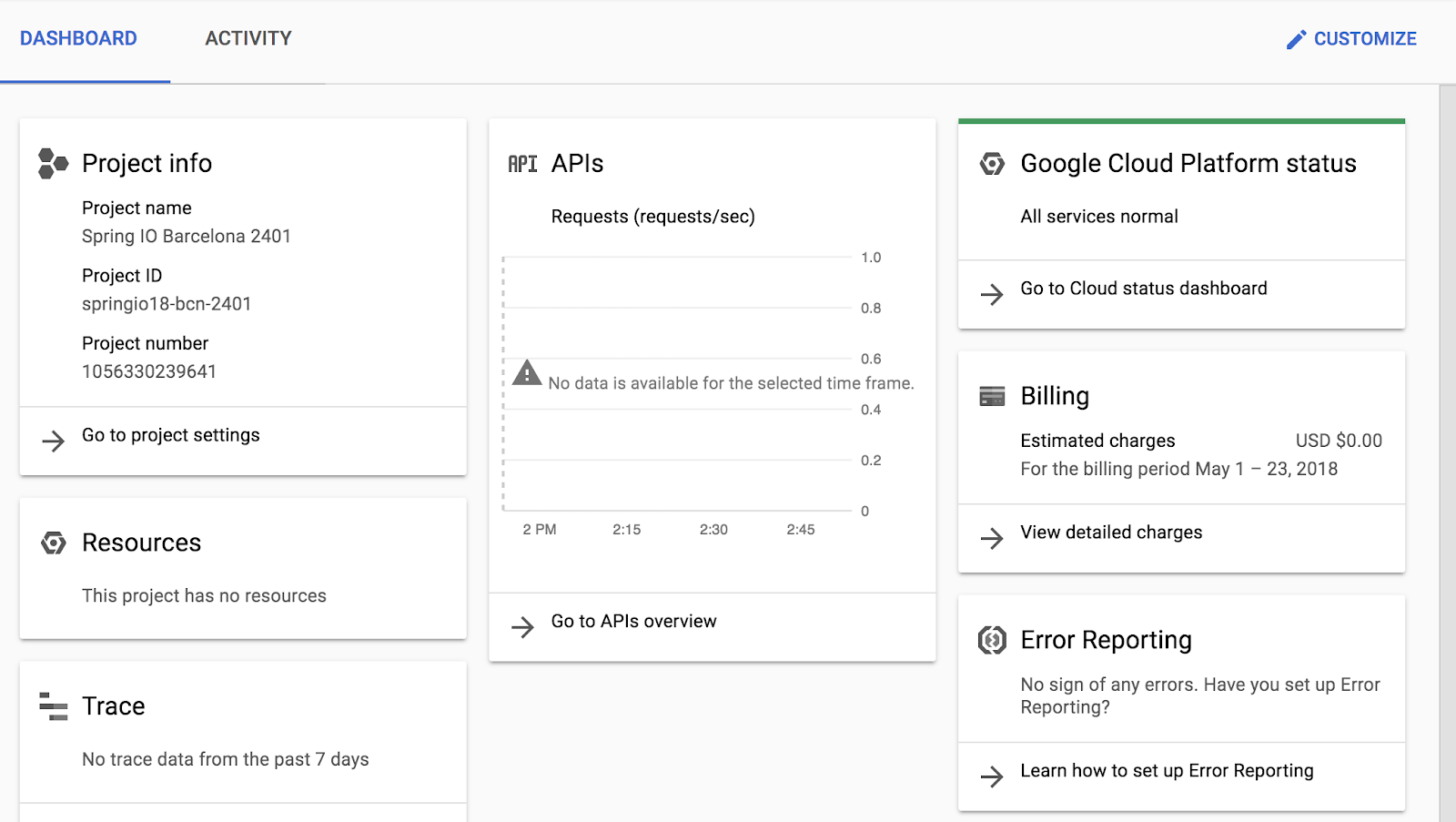
Access Qwiklabs

For each lab, you get a new Google Cloud project and set of resources for a fixed time at no cost.

1. Sign in to Qwiklabs using an **incognito window**.
2. Note the lab's access time (for example, 1:15:00), and make sure you can finish within that time.  
   There is no pause feature. You can restart if needed, but you have to start at the beginning.
3. When ready, click **Start lab**.
4. Note your lab credentials (**Username** and **Password**). You will use them to sign in to the Google Cloud Console.
5. Click **Open Google Console**.
6. Click **Use another account** and copy/paste credentials for **this** lab into the prompts.  
   If you use other credentials, you'll receive errors or **incur charges**.
7. Accept the terms and skip the recovery resource page.

**Note:** Do not click **End Lab** unless you have finished the lab or want to restart it. This clears your work and removes the project.

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



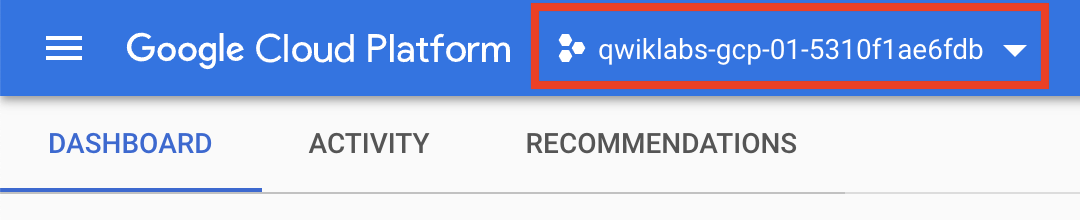
**Task 1. Explore the Google Cloud console**

In this task, you explore the Google Cloud console and create resources.

**Verify that your project is selected**

1. In the Google Cloud console title bar, click the **Select a project** drop-down list and select the project ID that Qwiklabs provided with your authentication credentials.
2. The project ID will take a form like [*qwiklabs-gcp-*] followed by a long hexadecimal number.
3. Click **Cancel** to close the dialog.

Your title bar should indicate the project ID as shown in the following 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 worldwide 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.

A Cloud Storage bucket must have a globally unique name. In your organization, you should follow Google Cloud's [Best practices for Cloud Storage Guide](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 Google Cloud project that Qwiklabs created for us, because Google Cloud project IDs are also globally unique.

1. In the Google Cloud console, on the **Navigation menu** (Navigation menu icon), click **Cloud overview > Dashboard** .
2. In the **Dashboard** tab of the resulting screen, the **Project info** section shows your Google Cloud project ID.
3. Select and copy the project ID. Because this project ID was created for you by Qwiklabs, it will take the form of [*qwiklabs-gcp-*] followed by a long hexadecimal number.
4. In the Google Cloud console, on the **Navigation menu** (Navigation menu icon), click **Cloud Storage** > **Buckets**.
5. Click **Create bucket**.
6. For **Name**, paste in the Google Cloud project ID string you copied in an earlier step. These lab instructions will later refer to the name that you typed as [BUCKET\_NAME].
7. Click **Continue**.
8. For **Location Type**, select **Region** and select europe-west4 from the dropdown.
9. Click **Continue**.
10. Click on **Choose how to control access to objects** and uncheck **Enforce public access prevention on this bucket**, then select **Fine-grained**.
11. Click **Continue**.
12. Leave all other values as their defaults.
13. Click **Create**.

**Note:** The Google Cloud 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 Google Kubernetes Engine.

1. On the **Navigation menu** (Navigation menu icon), click **Compute Engine** > **VM instances**.
2. Click **Create instance**.
3. For **Name**, type **first-vm** as the name of your instance.
4. For **Region**, select europe-west4.
5. For **Zone**, select europe-west4-a.
6. For **Machine type**, examine the options.

**Note:** The machine type menu lists the number of virtual CPUs, the amount of memory, and a symbolic name such as *e2-medium*. 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 (e2-standard-2)**.

How did the cost change?

1. For **Machine type,** click **e2-micro (2 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. In **Machine configuration**, notice the value of **CPU platform**, notice the value, and click **Edit**.
3. Click on the pencil icon on the top to edit the first-vm instance.

**Note:** You can't change the machine type, the CPU platform, or the zone of a running Google Cloud 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 to the **Management** section and examine **Availability policies**.

**Note:** Compute Engine offers preemptible VM instances, which cost less per hour but can be terminated by Google Cloud 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 and Continue**.
5. On the **Grant this service account access to project** page, specify the role as **Basic** > **Editor**.
6. Click **Continue**.
7. Click **Done**.
8. On the **Service accounts** page, move to the extreme right of the test-service-account and click on the three dots.
9. Click **Manage keys**.
10. Click **ADD KEY**
11. Select **Create new key**
12. Select **JSON** as the key type.
13. 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**.

Click **Check my progress** to verify the objective.

Assessment Completed!

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

Check my progress

*Assessment Completed!*

**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 Google Cloud services
* gcloud storage and 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 one 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 Google Cloud console title bar, click **Activate Cloud Shell** (Cloud Shell icon).
2. When prompted, click **Continue**.

Cloud Shell opens at the bottom of the Google Cloud 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 Google cloud Console without closing Cloud Shell.
* **Open in new window:** Having Cloud Shell at the bottom of the Google Cloud 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. Every time 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 of your choice.
3. In Cloud Shell, execute the following command to create environment variables:

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

Copied!

content\_copy

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

Copied!

content\_copy

MY\_REGION=europe-west4

Copied!

content\_copy

**Note:** When you are working in the Cloud Shell or writing scripts, creating environment variables is a good practice. You can easily and consistently reuse these environment variables, which makes your work less error-prone.**Note:** 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-dot menu icon (Three-dot menu icon) in the Cloud Shell toolbar to display further options.
3. Click **Upload** and choose credentials.json file from your local machine.
4. Click **Open**.
5. Click **Upload** to transfer credentials.json to Cloud Shell VM.
6. Click the **X** icon to close the file upload pop-up window.
7. 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 Google Cloud console**

The gcloud storage and gsutilcommands let you work with Cloud Storage from the command line. In this task, you use the gcloud storage command in Cloud Shell.

1. In Cloud Shell, use the gcloud storage command to create a bucket:

gcloud storage buckets create gs://$MY\_BUCKET\_NAME\_2 --location=europe-west4

Copied!

content\_copy

Click **Authorize** if prompted.

1. In the Google Cloud console, on the **Navigation menu** (Navigation menu icon), click **Cloud Storage** > **Buckets**, or click **Refresh** if you are already in the Cloud Storage page.

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

Copied!

content\_copy

1. Select a zone from the first column of the list. Notice that a Google Cloud zone's name consists of its 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]

Copied!

content\_copy

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

gcloud config set compute/zone $MY\_ZONE

Copied!

content\_copy

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

Copied!

content\_copy

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 "e2-standard-2" \

--image-project "debian-cloud" \

--image-family "debian-11" \

--subnet "default"

Copied!

content\_copy

1. List the virtual machine instances in your project:

gcloud compute instances list

Copied!

content\_copy

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

1. In the Google Cloud console, on the **Navigation menu** (Navigation menu icon), 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 Google Cloud console offers the link because you configured this VM's firewall to allow HTTP traffic.
3. Click the link you found in your first 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"

Copied!

content\_copy

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

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 Google Cloud console, on the **Navigation menu** (Navigation menu icon), click **IAM & admin** > **Service accounts**.

**Note:** Refresh the page until you see *test-service-account2*.

Click **Check my progress** to verify the objective.

Assessment Completed!

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

Check my progress

*Assessment Completed!*

1. In Cloud Shell, execute the following command to grant the second service account the 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

Copied!

content\_copy

**Note:** 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 Google Cloud console, on the **Navigation menu** (Navigation menu icon), click **IAM & admin** > **IAM**.
2. You will see the new service account called 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:

gcloud storage cp gs://cloud-training/ak8s/cat.jpg cat.jpg

Copied!

content\_copy

1. Copy the file into the first bucket that you created earlier:

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

Copied!

content\_copy

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

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

Copied!

content\_copy

1. In the Google Cloud console, on the **Navigation menu**(Navigation menu icon), click **Cloud Storage** > **Buckets**, select both 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 next two commands.
2. First, execute the following command in Cloud Shell:

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

cat acl.txt

Copied!

content\_copy

The output should look like the following example, but with different numbers. This output shows that anyone with an owner, editor, or viewer role for the project has access (OWNER access for owners and editors, and READER access for viewers).

[

{

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

"projectTeam": {

"projectNumber": "560255523887",

"team": "owners"

},

"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. Then, to change the object to have private access, execute the following command:

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

Copied!

content\_copy

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

Copied!

content\_copy

The output should look similar to the following example.

[

{

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

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

"role": "OWNER"

}

]

Now only the original creator of the object (your lab account) has OWNER access.

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

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

gcloud config list

Copied!

content\_copy

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.

[component\_manager]

disable\_update\_check = True

[compute]

gce\_metadata\_read\_timeout\_sec = 30

zone = europe-west4-a

[core]

account = google12345678\_student@qwiklabs.net

disable\_usage\_reporting = False

project = qwiklabs-Google Cloud-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

Copied!

content\_copy

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

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

gcloud config list

Copied!

content\_copy

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

[component\_manager]

disable\_update\_check = True

[compute]

gce\_metadata\_read\_timeout\_sec = 30

zone = europe-west4-a

[core]

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

disable\_usage\_reporting = False

project = qwiklabs-Google Cloud-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

Copied!

content\_copy

You should see output that looks like the following example.

Credentialed Accounts

ACTIVE:

ACCOUNT: student-03-5165fd82c14b@qwiklabs.net

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:

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

Copied!

content\_copy

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**

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:

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

Copied!

content\_copy

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

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, replacing *[USERNAME]* with the username provided in the **Qwiklabs Connection Details** pane on the left of the lab instructions page:

gcloud config set account [USERNAME]

Copied!

content\_copy

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:

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

Copied!

content\_copy

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.

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

Copied!

content\_copy

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

1. In the Cloud Console, on the **Navigation menu** (Navigation menu icon), click **Cloud Storage** > **Buckets**.
2. Select the first storage bucket with the project ID that you created. Notice that the cat.jpg file has Public access.
3. Click **Copy URL** to copy the link.
4. Open a new 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.

Assessment Completed!

Work with Cloud Storage in Cloud Shell

Check my progress

*Assessment Completed!*

**Task 4. Explore the Cloud Shell Editor**

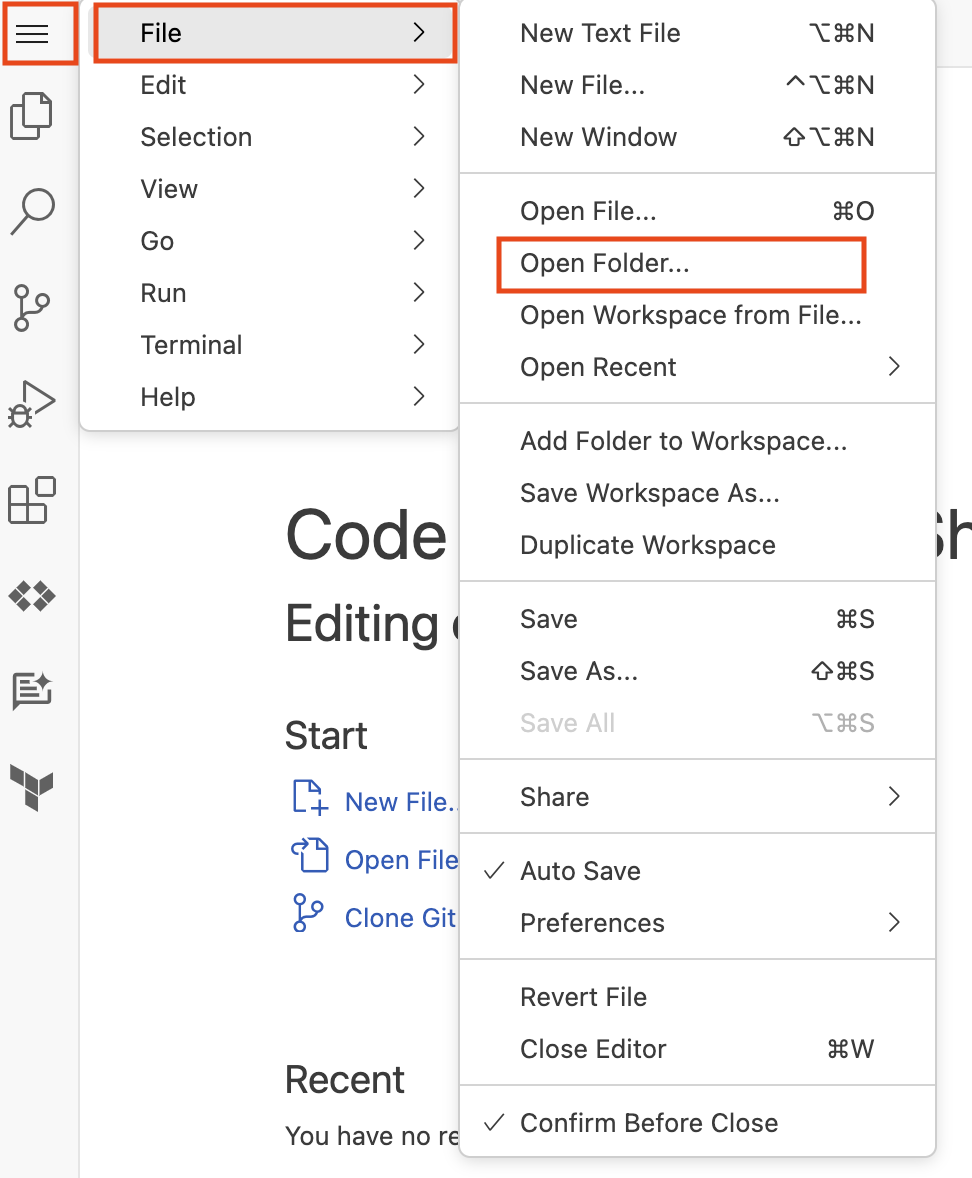
In this task, you explore using the Cloud Shell code editor.

**Open the Cloud Shell Editor**

1. In Cloud Shell, click the **Open Editor** icon (Cloud Shell editor icon) and then click on the **Open in a new window** link.

A new tab opens with the Cloud Shell Editor. The Google Cloud console and Cloud Shell remain on the original tab. You can switch between Cloud Shell and the code editor by clicking the tab.

1. In Cloud Shell Editor, click the menu and select **File** > **Open Folder**.



1. Click **Ok**.

The folder opens and displays a list of files and directories in the left pane.

1. On the Google Cloud console tab, click **Open Terminal**, and in Cloud Shell, execute the following command to clone a git repository:

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

Copied!

content\_copy

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

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

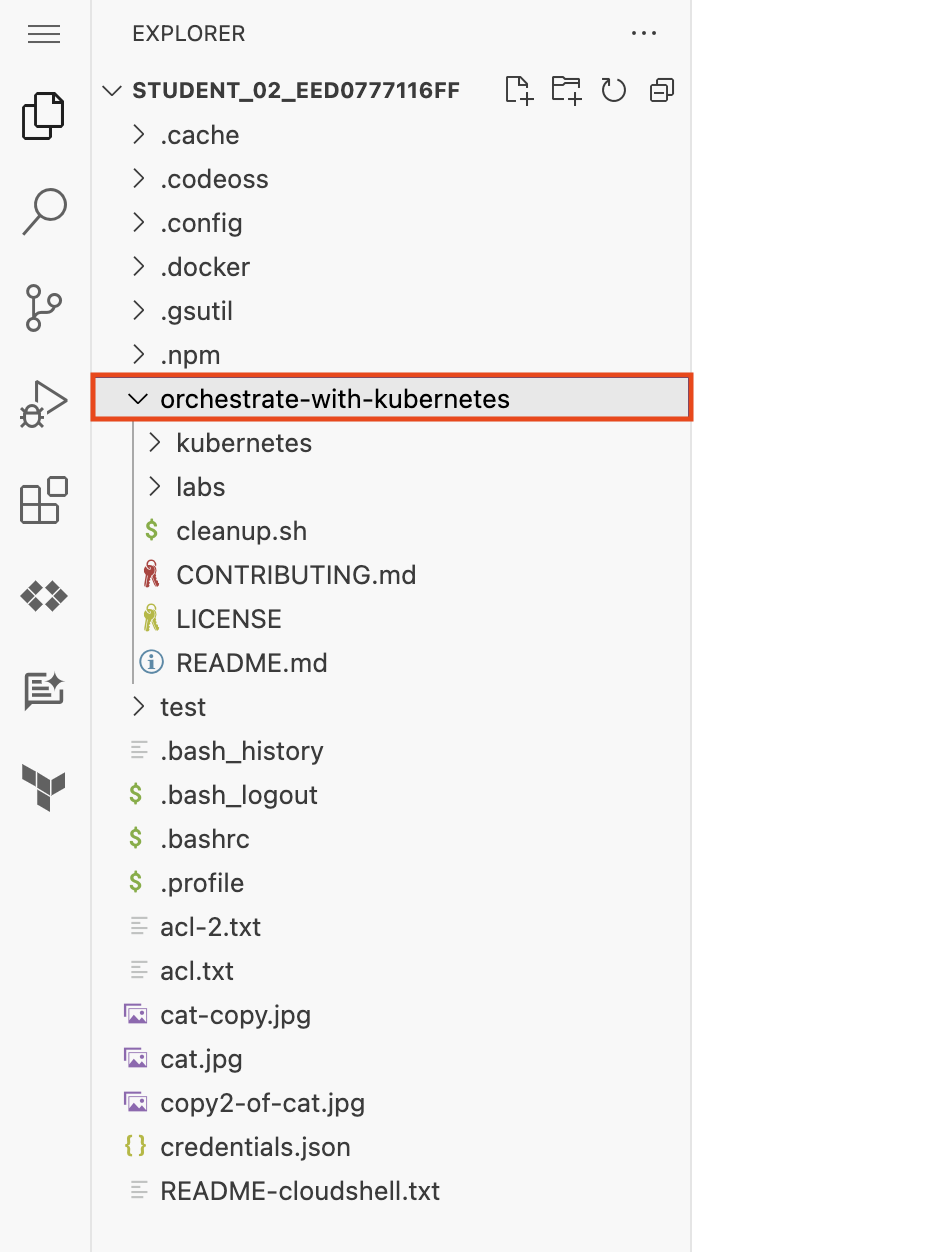
mkdir test

Copied!

content\_copy

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

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



1. In the left pane, click the cleanup.sh file to open it in the right pane of the Cloud Shell Editor window.
2. Add the following text as the last line of the cleanup.sh file:

echo Finished cleanup!

Copied!

content\_copy

**Note:** 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

Copied!

content\_copy

1. Verify that the output of cat cleanup.sh includes the line of text that you added.
2. In the Cloud Shell code editor, right click the orchestrate-with-kubernetes folder, and select **New File**.
3. Name the file **index.html**.
4. Click **Save**.
5. 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>

Copied!

content\_copy

**Note:** 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:

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

1. On the **Navigation menu** (Navigation menu icon), 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 remove -y --purge man-db

sudo touch /var/lib/man-db/auto-update

sudo apt-get update

sudo apt-get install nginx

Copied!

content\_copy

**Note:** It may take few minutes to complete the process. If prompted, click Y to continue.

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

gcloud compute scp index.html first-vm:index.nginx-debian.html --zone=europe-west4-a

Copied!

content\_copy

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

Copied!

content\_copy

Click **Check my progress** to verify the objective.

Please add the required image URL in the HTML file and then copy the file to the document root of the nginx Web server.

Install the nginx web server and customize the welcome page

Check my progress

*Please add the required image URL in the HTML file and then copy the file to the document root of the nginx Web server.*

1. On the **Navigation menu** (Navigation menu icon), click **Compute Engine** > **VM instances**.
2. Click the link in the **External IP** column for your **first-vm**. A new browser tab opens with a webpage that contains the cat image.

**End your lab**

When you have completed your lab, click **End Lab**. Google Cloud Skills Boost removes the resources you’ve used and cleans the account for you.

You will be given an opportunity to rate the lab experience. Select the applicable number of stars, type a comment, and then click **Submit**.

The number of stars indicates the following:

* 1 star = Very dissatisfied
* 2 stars = Dissatisfied
* 3 stars = Neutral
* 4 stars = Satisfied
* 5 stars = Very satisfied

You can close the dialog box if you don't want to provide feedback.

For feedback, suggestions, or corrections, please use the **Support** tab.

Copyright 2022 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.

[navigate\_beforePrevious](https://www.cloudskillsboost.google/course_sessions/14931163/video/490753)[Nextnavigate\_next](https://www.cloudskillsboost.google/course_sessions/14931163/quizzes/490755)