**LAB 1:**

**Course: Architecting with Google Kubernetes Engine - Foundations**

**Module**: Introduction to Google Cloud Platform

AK8S-01 Accessing the GCP Console and Cloud Shell

**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**

**Task 1. Explore the GCP Console**

Verify that your project is selected

#Verify\_project

$ gcloud config list project

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

STEPS 1-2. Get the project ID

#Default\_Project\_ID

gcloud config list --format 'value(core.project)' 2>/dev/null

STEPS 3-6: Bucket

#Creating\_a\_bucket

gsutil mb gs://qwiklabs-gcp-03-1589bd7eb44b

### Create a virtual machine (VM) instance

STEPS 1-11:

#Creating\_a\_VM\_instance

gcloud compute instances create first-vm --zone=central1-c --custom-cpu=4 --machine-type=n1-standard-2 --tags http-server

### Explore the VM details

gcloud compute instances describe first-vm --zone us-central1-c

gcloud compute disks describe my-disk --zone=us-east1-a

### Create an IAM service account

STEPS 1-11:

#creating\_service\_accounts

gcloud iam service-accounts create test-service-account \

--description="editor" \

--display-name="editor"\

--condition=[key=JSON]**T**

## **Task 2. Explore Cloud Shell**

### Open Cloud Shell and explore its features

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

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 a 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

STEPS 1-2:

#Creating\_environment\_variables

MY\_BUCKET\_NAME\_1=qwiklabs-gcp-03-1589bd7eb44b

MY\_BUCKET\_NAME\_2=my\_bucket\_2

MY\_REGION=us-central1

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

STEP 1:

#Downloading\_the\_file

cloudshell download JSON

Rename it to **credentials.json**

STEP 2:

#Uploading\_credentials.json\_to\_first-vm

gcloud compute scp Desktop/credentials.json first-vm:~

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

STEP 1:

#createing\_a\_bucket

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

STEP 2:

#verifying\_bucket\_creation

gcloud compute instance list

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

STEP 1:

#list\_zones

gcloud compute zones list | grep $MY\_REGION

STEP 2:

Select a zone from the list #us-central1-b

STEP 3:

MY\_ZONE=us-central1-b

STEP 4:

gcloud config set compute/zone $MY\_ZONE

STEP 5:

#Environment\_virable\_for\_my\_second\_vm

MY\_VMNAME=second-vm

STEP 6:

#Create\_a\_vm\_in\_a\_default\_zone

gcloud compute instances create $MY\_VMNAME \

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

--image-project "debian-cloud" \

--image-family "debian-9" \

--subnet "default"

STEP 7:

#Display\_instance\_list

gcloud compute instances list

STEP 8-10:

Copy the first-vm’s external IP

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

### STEP 1-2:

#Create\_service\_account

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

STEP 3:

#Bind\_roles

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

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

STEP 1:

#Copy\_a\_picture

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

STEP 2:

#Copy\_the\_file\_into\_the\_bucket

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

STEP 3:

#Copy\_the\_file\_from\_the\_first\_bucket\_to\_the\_second\_bucket

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

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

STEP 1:

#Get\_the\_default\_access\_list

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

cat acl.txt

STEP 2:

#Change\_the\_private\_access

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

STEP 3:

#Verify\_ACL

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

cat acl-2.txt

### Authenticate as a service account in Cloud Shell

STEP 1:

#View\_current\_configs

gcloud config list

STEP 2:

#Change\_the\_authenticated\_user

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

STEP 3:

#Verify\_current\_configs

gcloud config list

STEP 4:

#Verif\_lists\_of\_authorized\_users

gcloud auth list

STEP 5:

#Verify\_access\_roles

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

STEP 6:

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

STEP 7:

#Switch\_service\_account

gcloud config set account student-03-eb20a744a395@qwiklabs.net

STEP 8:

#verify\_that\_you\_can\_access\_the\_cat.jpg

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

STEP 9:

#Make\_the\_first\_storage\_bucket\_readable

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

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

STEP 1:

Open the cloud shell

STEP 2:

#clone\_a\_repository

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

STEP 3:

#Create\_a\_directory

mkdir test

STEP 4:

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

STEP 5:

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

STEP 6:

echo Finished cleanup!

STEP 7:

#Change\_directory\_and\_display\_cleanup.sh

cd orchestrate-with-kubernetes

cat cleanup.sh

STEP 8

Verify that the output of cat cleanup.sh

STEP 9:

#Return\_to\_home\_directory

cd

STEP 10:

In the Cloud Shell code editor, click to open the File menu and choose New File. Name the file index.html.

STEP 11:

#EDIT\_THE\_HTML\_FILE

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

<body>

<h1>Cat</h1>

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

</body></html>

STEP 12:

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

STEP 13-14:

ssh into the vm.

STEP 15:

#Install\_nginx

sudo apt-get update

sudo apt-get install nginx

STEP 16

#Copy\_the\_html\_file\_into\_the \_vm

cd orchestrate-with-kubernetes

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

STEP 17:

#Copy\_the\_html\_file\_into\_the\_document\_root

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

STEP 18:

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.

STEP 19:

End of LAB

**LAB 2:**

**Course: Architecting with Google Kubernetes Engine - Foundations**

**Module**: Kubernetes Architecture

AK8S-03 Creating a GKE Cluster via GCP Console -

## **Objectives**

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

* Use the GCP Console to build and manipulate GKE clusters
* Use the GCP Console to deploy a Pod
* Use the GCP Console to examine the cluster and Pods

## **Task 0. Lab Setup**

## **Task 1. Deploy GKE clusters**

STEP 1:

#SET\_ENVIRONMENT\_VARIABLES

export my\_zone=us-central1-a

export my\_cluster=standard-cluster-1

STEP 2:

#CREATE\_CLUSTERS

gcloud container clusters create $my\_cluster --num-nodes 3 --zone $my\_zone --enable-ip-alias

## 

## **Task 2. Modify GKE clusters**

STEP 1:

#MODIFY\_STANDARD-CLUSTER-1

gcloud container clusters resize $my\_cluster --zone $my\_zone --size=4

STEP 2:

When prompted with Do you want to continue (Y/n), press y to confirm.

## **Task 3. Deploy a sample workload**

STEPS 1-5:

#DEPLOYING\_WORKLOADS

kubectl create deployment nginx-1 --image=nginx:latest

## **Task 4. View details about workloads in the GCP Console**

STEPS 1-10:

#VIEWING\_PODS

kubectl get pods

#VIEWING\_WORKLOADS

kubectl get service nginx

**END OF LAB**