Lab 1: GCP Fundamentals: Getting Started with Compute Engine

**Objectives :**

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

* Create a Compute Engine virtual machine using the Google Cloud Platform (GCP) Console.
* Create a Compute Engine virtual machine using the gcloud command-line interface.
* Connect between the two instances.

**Steps :**

* 1. **Create a virtual machine using the GCP console**

gcloud compute instances create “my-vm-1” --machine-type “n1-standard-1” --image-project “debian-cloud” --image “debian-9-stretch-v2019213” --subnet “default” --tags http

gcloud compute firewall-rules create “allow-http” --action=ALLOW --direction=INGRESS --target-tags=http

* 1. **Create a virtual machine using the gcloud command line interface**

gcloud config set compute/zone us-central1-b

gcloud compute instances create "my-vm-2" --machine-type "n1-standard-1" --image-project "debian-cloud" --image "debian-9-stretch-v20190213" --subnet "default"

* 1. **Connect between VM instances**

**To open a command prompt on the my-vm-2 instance**

gcloud compute ssh my-vm-2

**Use the ping command to confirm that my-vm-2 can reach my-vm-1 over the network:**

ping –c 5 my-vm-1

**Use the ssh command to open a command prompt on my-vm-1:**

ssh my-vm-1

**At the command prompt on my-vm-1, install the Nginx web server:**

sudo apt-get install nginx-light –y

**Use the nano text editor to add a custom message to the home page of the web server:**

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

**Use the arrow keys to move the cursor to the line just below the h1 header. Add text like this, and replace YOUR\_NAME with your name:**

Hi from **DALUBUHLE** DLODLO

**Confirm that the web server is serving your new page. At the command prompt on my-vm-1, execute this command:**

curl <http://localhost/>

*The response will be the HTML source of the web server's home page, including your line of custom text.*

**To exit the command prompt on my-vm-1, execute this command:**

exit

*You will return to the command prompt on****my-vm-2***

**To confirm that my-vm-2 can reach the web server on my-vm-1, at the command prompt on my-vm-2, execute this command:**

curl <http://my-vm-1/>

*The response will again be the HTML source of the web server's home page, including your line of custom text.*

**Copy the External IP address for my-vm-1 and paste it into the address bar of a new browser tab. You will see your web server's home page, including your custom text.**

**Lab 2: Google Cloud Fundamentals: Getting Started with GKE**

**Objectives :**

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

* Provision a [Kubernetes](http://kubernetes.io/) cluster using [Kubernetes Engine.](https://cloud.google.com/container-engine)
* Deploy and manage Docker containers using kubectl.

**Steps :**

1. **Start a Kubernetes Engine cluster**

*For convenience, place the zone that Qwiklabs assigned you to into an environment variable called MY\_ZONE :*

export MY\_ZONE=us-central1-a

*Start a Kubernetes cluster managed by Kubernetes Engine. Name the cluster webfrontend and configure it to run 2 nodes:*

gcloud container clusters create webfrontend --zone $MY\_ZONE --num-nodes 2

*After the cluster is created, check your installed version of Kubernetes* :

kubectl version

*View your running nodes in the GCP Console :*

gcloud compute instances list

1. **Run and deploy a container**

*From your Cloud Shell prompt, launch a single instance of the nginx container.*

kubectl create deploy nginx --image=nginx:1.17.10

*View the pod running the nginx container*:

kubectl get pods

*Expose the nginx container to the Internet:*

kubectl expose deployment nginx --port 80 --type LoadBalancer

*View the new service:*

kubectl get services

You can use the displayed external IP address to test and contact the nginx container remotely.

*Scale up the number of pods running on your service:*

kubectl scale deployment nginx --replicas 3

*Confirm that Kubernetes has updated the number of pods:*

kubectl get pods

*Confirm that your external IP address has not changed:*

kubectl get services

# Lab 3: Google Cloud Fundamentals: Getting Started with App Engine

**Objectives :**

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

* Initialize App Engine.
* Preview an App Engine application running locally in Cloud Shell.
* Deploy an App Engine application, so that others can reach it.

**Steps :**

1. **Initialize App Engine**

*Initialize your App Engine app with your project and choose its region:*

gcloud app create --project=$DEVSHELL\_PROJECT\_ID

When prompted, select the [region](https://cloud.google.com/appengine/docs/locations)where you want your App Engine application located.

*Clone the source code repository for a sample application in the****hello\_world****directory:*

git clone <https://github.com/GoogleCloudPlatform/python-docs-samples>

*Navigate to the source directory*:

cd python-docs-samples/appengine/standard\_python3/hello\_world

1. **Run Hello World Application locally**

*Execute the following command to download and update the packages list*.

sudo apt-get update

*Set up a virtual environment in which you will run your application. Python virtual environments are used to isolate package installations from the system.*

sudo apt-get install virtualenv

If prompted [Y/n], press Y and then Enter.

virtualenv -p python3 venv

*Activate the virtual environment.*

source venv/bin/activate

*Navigate to your project directory and install dependencies.*

pip install -r requirements.txt

*Run the application:*

python main.py

Please ignore the warning if any.

*In****Cloud Shell****, click****Web preview****(Web Preview) >****Preview on port 8080****to preview the application.*

Result : “Hello World!” on the browser

To end the test, return to Cloud Shell and press **Ctrl+C** to abort the deployed service.

1. **Deploy and run Hello World on App Engine**

*Navigate to the source directory:*

cd ~/python-docs-samples/appengine/standard\_python3/hello\_world

*Deploy your Hello World application.*

gcloud app deploy

If prompted "Do you want to continue (Y/n)?", press Y and then Enter.

This **app deploy** command uses the app.yaml file to identify project configuration.

*Launch your browser to view the app at* [*http://YOUR\_PROJECT\_ID.appspot.com*](http://YOUR_PROJECT_ID.appspot.com)

gcloud app browse

Copy and paste the URL into a new browser window.

Result :

“Hello World!” on the browser