GCP Fundamentals: Getting Started with Deployment Manager and Stackdriver

In this lab, you create a deployment using Deployment Manager and use it to maintain a consistent state of your deployment. You will also view resource usage in a VM instance using Stackdriver.

Objectives

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

- Create a Deployment Manager deployment.
- Update a Deployment Manager deployment.
- View the load on a VM instance using Stackdriver.

Task 1: Sign in to the Google Cloud Platform (GCP) Console

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.

Task 2: Confirm that needed APIs are enabled

1. Make a note of the name of your GCP project. This value is shown in the top bar of the Google Cloud Platform Console. It will be of the form `qwiklabs-gcp-` followed by hexadecimal numbers.
2. In the GCP Console, on the **Navigation menu** (☰), click **APIs & services**.
3. Scroll down in the list of enabled APIs, and confirm that these APIs are enabled:

- Cloud Deployment Manager v2 API
- Cloud Runtime Configuration API
- Stackdriver Monitoring API

4. If one or more APIs is missing, click the **Enable APIs and Services** button at top. Search for the above APIs by name and enable each for your current project. (You noted the name of your GCP project above.)

Task 3: Create a Deployment Manager deployment

1. On the **Google Cloud Platform** menu, click **Activate Cloud Shell** (>_). If a dialog box appears, click **Start Cloud Shell**.
2. For your convenience, place the zone that Qwiklabs assigned you to into an environment variable called MY_ZONE. At the Cloud Shell prompt, type this partial command:

```
export MY_ZONE=
```

followed by the zone that Qwiklabs assigned you to. Your complete command will look like this:

```
export MY_ZONE=us-central1-f
```

3. At the Cloud Shell prompt, download an editable Deployment Manager template:

```
gsutil cp gs://cloud-training/gcpfcoreinfra/mydeploy.yaml mydeploy.yaml
```

4. Insert your Google Cloud Platform project ID into the file in place of the string `PROJECT_ID` using this command:

```
sed -i -e 's/PROJECT_ID/'$DEVSHELL_PROJECT_ID/ mydeploy.yaml
```

5. Insert your assigned Google Cloud Platform zone into the file in place of the string `ZONE` using this command:

```
sed -i -e 's/ZONE/'$MY_ZONE/ mydeploy.yaml
```

6. View the `mydeploy.yaml` file, with your modifications, with this command:

```
cat mydeploy.yaml
```

The file will look something like this:

```
 resources:
 - name: my-vm
  type: compute.v1.instance
  properties:
   zone: us-central1-a
   machineType: zones/us-central1-a/machineTypes/n1-standard-1
   metadata:
    items:
    - key: startup-script
     value: "apt-get update"
   disks:
   - deviceName: boot
    type: PERSISTENT
    boot: true
    autoDelete: true
    initializeParams:
     sourceImage:  https://www.googleapis.com/compute/v1/projects/debian-
cloud/global/images/debian-9-stretch-v20180806
   networkInterfaces:
   -  network:  https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-
dcdf854d278b50cd/global/networks/default
    accessConfigs:
    - name: External NAT
     type: ONE_TO_ONE_NAT
```

Do not use the above text literally in your own **mydeploy.yaml** file. Be sure that the zone that is named on the **zone:** and **machineType:** lines in your file matches the zone to which Qwiklabs assigned you. Be sure that the GCP project ID on the **network:** line in your file matches the project ID to which Qwiklabs assigned you, not the one in this example.

7. Build a deployment from the template:

```
gcloud deployment-manager deployments create my-first-depl --config
mydeploy.yaml
```

When the deployment operation is complete, the **gcloud** command displays a list of the resources named in the template and their current state.

8. Confirm that the deployment was successful. In the GCP Console, on the **Navigation menu** (☰), click **Compute Engine > VM instances**. You will see that a VM instance called **my-vm** has been created, as specified by the template.

9. Click on the VM instance's name to open its VM instance details screen.

10. Scroll down to the **Custom metadata** section. Confirm that the startup script you specified in your Deployment Manager template has been installed.

Click *Check my progress* to verify the objective.

Create a Deployment Manager deployment

Check my progress

Task 4: Update a Deployment Manager deployment

1. Return to your Cloud Shell prompt. Launch the `nano` text editor to edit the **mydeploy.yaml** file:

```
nano mydeploy.yaml
```

2. Find the line that sets the value of the startup script, `value: "apt-get update"`, and edit it so that it looks like this:

```
      value: "apt-get update; apt-get install nginx-light -y"
```

Do not disturb the spaces at the beginning of the line. The YAML templating language relies on indented lines as part of its syntax. As you edit the file, be sure that the `v` in the word `value` in this new line is immediately below the `k` in the word `key` on the line above it.

3. Press **Ctrl+O** and then press **Enter** to save your edited file.
4. Press **Ctrl+X** to exit the **nano** text editor.
5. Return to your Cloud Shell prompt. Enter this command to cause Deployment Manager to update your deployment to install the new startup script:

```
gcloud deployment-manager deployments update my-first-depl --config
mydeploy.yaml
```

Wait for the **gcloud** command to display a message confirming that the update operation was completed successfully.

6. In the GCP console, on the **Navigation menu** (☰), click **Compute Engine > VM instances**.
7. Click on the **my-vm** VM instance's name to open its **VM instance details** pane.
8. Scroll down to the **Custom metadata** section. Confirm that the startup script has been updated to the value you declared in your Deployment Manager template.

Click *Check my progress* to verify the objective.

Update the Deployment Manager deployment

Check my progress

Task 5: View the Load on a VM using Stackdriver

1. In the GCP Console, on the **Navigation menu** (☰), click **Compute Engine** > **VM instances**.
2. To open a command prompt on the **my-vm** instance, click **SSH** in its row in the **VM instances** list.
3. In the ssh session on **my-vm**, execute this command to create a CPU load:

```
dd if=/dev/urandom | gzip -9 >> /dev/null &
```

This Linux pipeline forces the CPU to work on compressing a continuous stream of random data.

Leave the window containing your SSH session open while you proceed with the lab.

4. In the GCP Console, on the **Navigation menu** (☰), under **Stackdriver**, click **Monitoring**.
5. If prompted to log in again, log in with the student account and password you were provided for this lab. Stackdriver will then automatically provision a workspace for your project. Note: this may take a few minutes.

6. Once your project workspace has been created, click the **Add GCP projects** link on the main page. On the next screen, confirm that the GCP project which Qwiklabs created for you is shown under the **Monitored Accounts** tab. Leave the rest of the options as they are, by default.
7. Under the **Settings** tab menu, click **Agent**. Using your VM's open SSH window and the code shown on the Agents page, install both the Monitoring and Logging agents on your project's VM.
8. Once both of the agents have been installed on your project's VM, click **Resources > Metrics Explorer** under the main Stackdriver menu on the far left.
9. In the **Metrics Explorer**'s **Metric** pane, select the resource type **GCE VM instance** and the metric **CPU usage**.

In the resulting graph, notice that CPU usage increased sharply a few minutes ago.

10. Terminate your workload generator. Return to your ssh session on **my-vm** and enter this command:

```
kill %1
```

Congratulations!