

# Getting Started with Deployment Manager and Stackdriver

## Overview

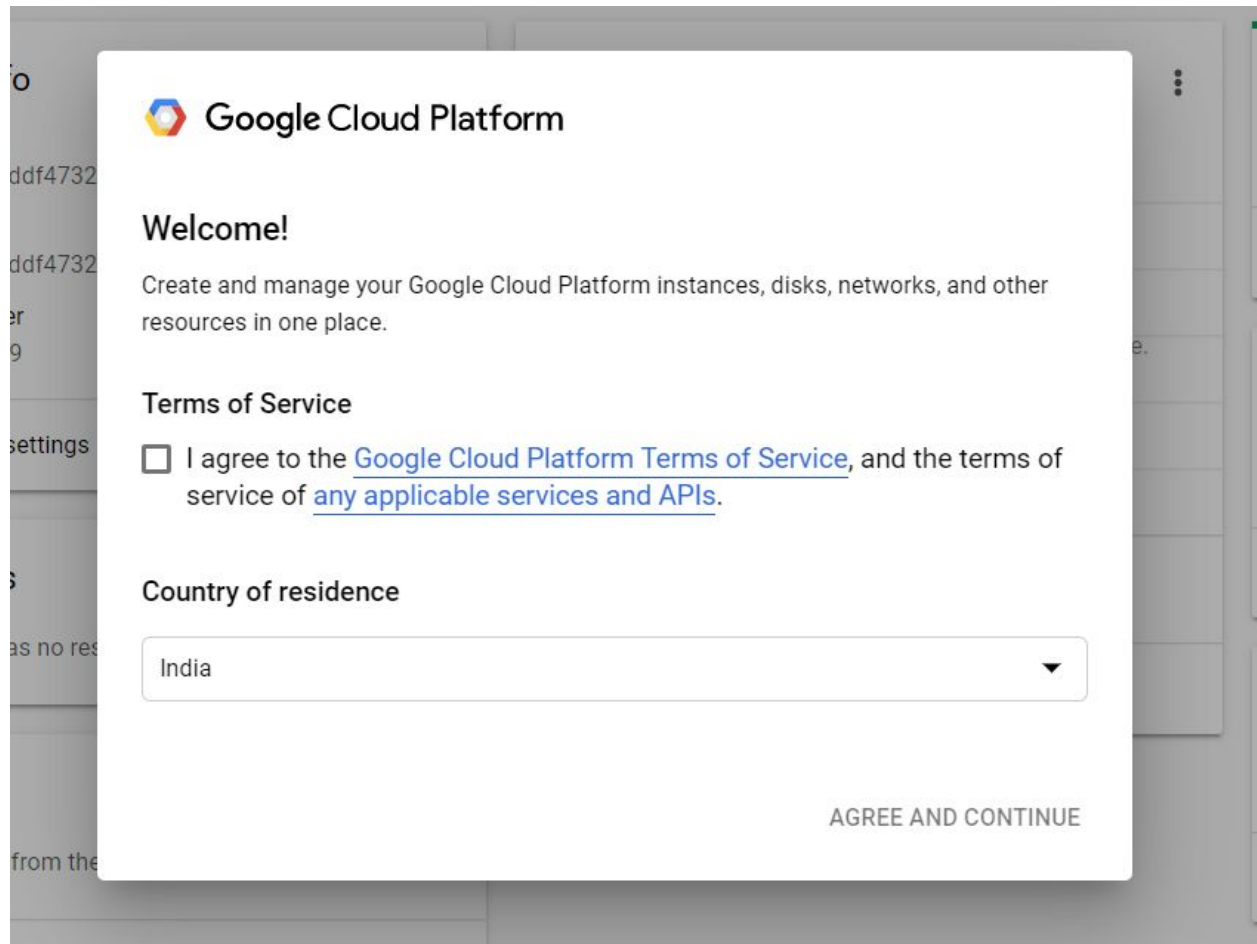
In this lab, you create a deployment using Deployment Manager and use it to maintain a consistent state of your deployment. You will also view resource usage in a VM instance using Stackdriver.


## 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**

A screenshot of a Google Cloud Platform welcome dialog box. The dialog is white with a grey border and is centered on a blurred background of a web interface. It contains the Google Cloud Platform logo and name at the top, followed by a 'Welcome!' heading and a brief description of the platform. Below this is a 'Terms of Service' section with an unchecked checkbox and a link to the terms of service. The 'Country of residence' section features a dropdown menu with 'India' selected. At the bottom right is an 'AGREE AND CONTINUE' button.

 **Google Cloud Platform**

**Welcome!**

Create and manage your Google Cloud Platform instances, disks, networks, and other resources in one place.

**Terms of Service**

☐ I agree to the [Google Cloud Platform Terms of Service](#), and the terms of service of [any applicable services and APIs](#).

**Country of residence**

India ▼

AGREE AND CONTINUE

## 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.

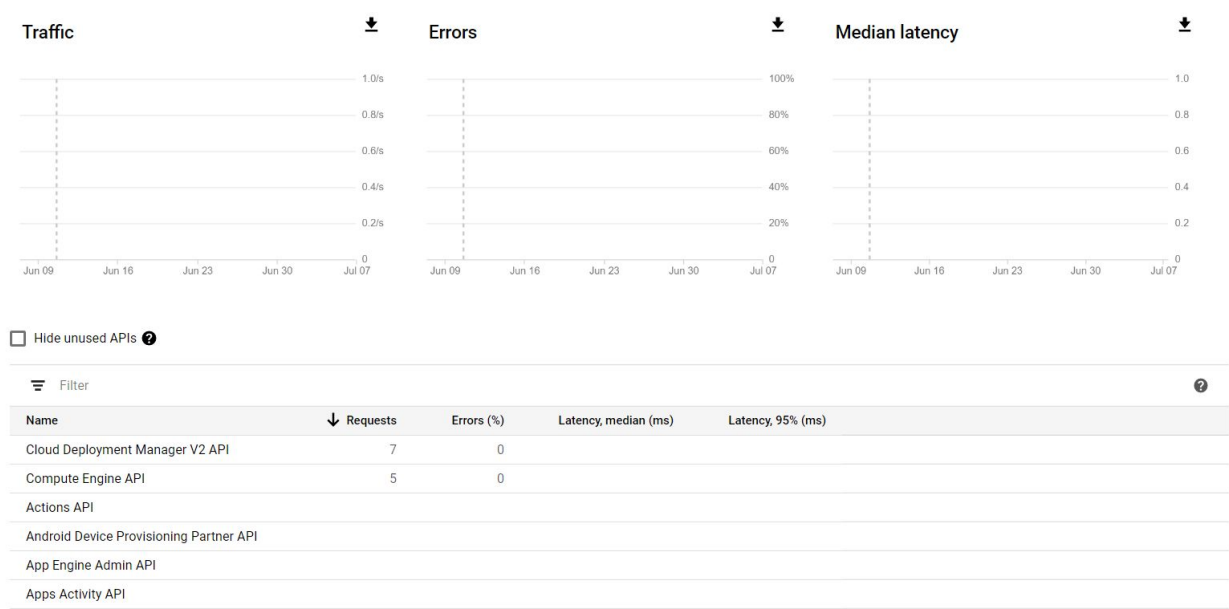


2. In the GCP Console, on the **Navigation menu** ( ), click **APIs & services**.

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
```

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

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

1. 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/'$DEVSHHELL_PROJECT_ID'/ mydeploy.yaml'
```

1. 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'
```

1. 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.

1. 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.

1. 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.

2. Click on the VM instance's name to open its VM instance details screen.
3. Scroll down to the **Custom metadata** section. Confirm that the startup script you specified in your Deployment Manager template has been installed.

```
Waiting for create [operation-1562469535754-58d0ec9a6a857-e12ee41c-8c9fc096]...done.
WARNING: Create operation operation-1562469535754-58d0ec9a6a857-e12ee41c-8c9fc096 completed with warnings:
...
code: EXTERNAL_API_WARNING
data:
- key: resource_name
  value: projects/debian-cloud/global/images/debian-9-stretch-v20180806
- key: replacement_suggestion
  value: A suggested replacement is 'projects/debian-cloud/global/images/debian-9-stretch-v20180814'.
message: The resource 'projects/debian-cloud/global/images/debian-9-stretch-v20180806'
is deprecated. A suggested replacement is 'projects/debian-cloud/global/images/debian-9-stretch-v20180814'.

NAME      TYPE      STATE      ERRORS  INTENT
my-vm     compute.v1.instance  COMPLETED  []
google4083741_student@cloudshell:~ (qwiklabs-gcp-ddf47320b3e727f4) $
```

Google Cloud Platform

16178981-9e727f4

Home

Marketplace

Billing

APIs & Services

Support

IAM & admin

Getting started

Security

COMPUTE

App Engine

Compute Engine

VM instances

Instance groups

Instance templates

Sole tenant nodes

Disks

Snapshots

Images

TPUs

Committed use discounts

Metadata

Health checks

Zones

Network endpoint groups

Operations

Security scans

Settings

ENABLE APIS AND SERVICES

Errors

1.0/s

0.8/s

0.6/s

0.4/s

0.2/s

0

Jun 23

Jun 30

Jul 07

Jun 09

Jun 16

e12ee41c-8c9fc096]...done.

6a857-e12ee41c-8c9fc096 completed with warni

Filter VM instances

Columns

<input type="checkbox"/>	Name ^	Zone	Recommendation	In use by	Internal IP	External IP	Connect
<input type="checkbox"/>	my-vm	us-central1-a			10.128.0.2 (nic0)	34.68.23.206	SSH

## 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
```

1. 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.


1. Press **Ctrl+O** and then press **Enter** to save your edited file.
2. Press **Ctrl+X** to exit the **nano** text editor.

3. 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.



1. In the GCP console, on the **Navigation menu** (  ), click **Compute Engine > VM instances**.
2. Click on the **my-vm** VM instance's name to open its **VM instance details** pane.
3. 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.


## 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:

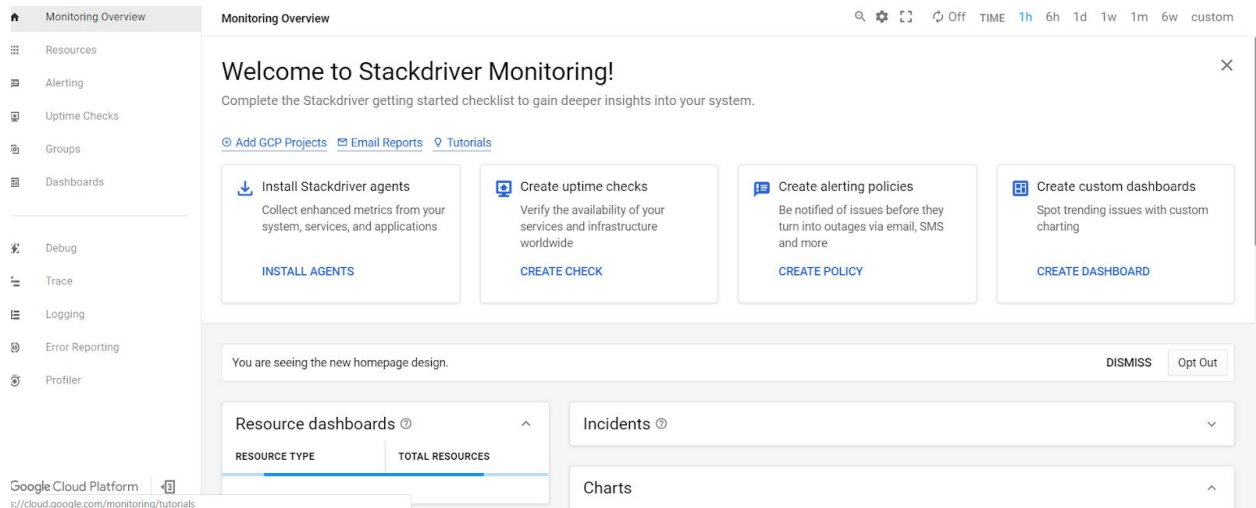
```
dd if=/dev/urandom | gzip -9 >> /dev/null &
```

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.

1. In the GCP Console, on the **Navigation menu** (  ), under **Stackdriver**, click **Monitoring**.

- 
2. 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.



- 
- 
3. 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.
4. 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.

```
google4083741_student@my-vm: ~ - Google Chrome
https://ssh.cloud.google.com/projects/qwiklabs-gcp-ddf47320b3e727f4/zones/us-central1-a/instances/my-vm?authuser=1&hl=en_US&...

Setting up libpython2.7-minimal:amd64 (2.7.13-2+deb9u3) ...
Setting up libpython2.7-stdlib:amd64 (2.7.13-2+deb9u3) ...
Setting up python2.7-minimal (2.7.13-2+deb9u3) ...
Setting up python2.7 (2.7.13-2+deb9u3) ...
Setting up libpython2.7:amd64 (2.7.13-2+deb9u3) ...
Setting up stackdriver-agent (5.5.2-384.stretch) ...
The instance has neither the application default credentials file nor the correct monitoring scopes; The agent will
likely fail.
Processing triggers for libc-bin (2.24-11+deb9u3) ...
Processing triggers for systemd (232-25+deb9u4) ...

=====
Installation of stackdriver-agent-5.5.2-384 completed successfully.

Please consult the documentation for troubleshooting advice:
  https://cloud.google.com/monitoring/agent

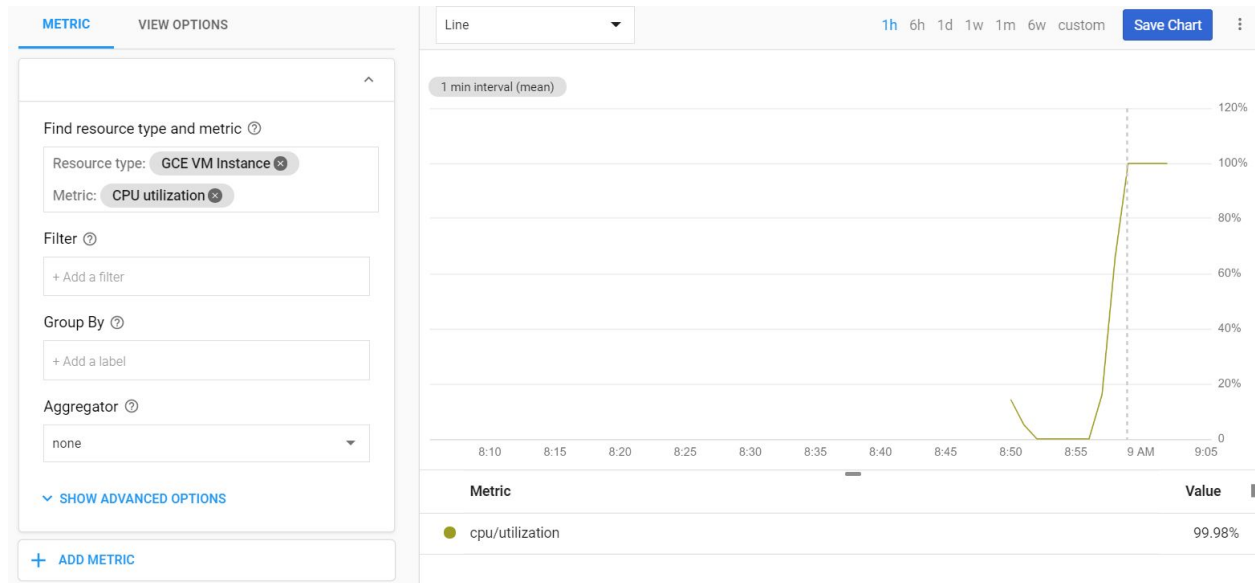
You can monitor the monitoring agent's logfile at:
  /var/log/syslog

=====
google4083741_student@my-vm:~$ curl -sSO https://dl.google.com/cloudagents/install-logging-agent.sh
google4083741_student@my-vm:~$ sudo bash install-logging-agent.sh

=====
Starting installation of google-fluentd
=====

Installing agents for Debian or Ubuntu.
OK
Selecting previously unselected package google-fluentd.
(Reading database ... 33705 files and directories currently installed.)
Preparing to unpack .../google-fluentd_1.6.13-1_amd64.deb ...
Unpacking google-fluentd (1.6.13-1) ...
Selecting previously unselected package google-fluentd-catch-all-config.
Preparing to unpack .../google-fluentd-catch-all-config_0.7_all.deb ...
Unpacking google-fluentd-catch-all-config (0.7) ...
Setting up google-fluentd (1.6.13-1) ...
Adding system user `google-fluentd' (UID 108) ...
Adding new group `google-fluentd' (GID 112) ...
Adding new user `google-fluentd' (UID 108) with group `google-fluentd'
```

5. 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.
6. 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.

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

```
kill %1
```

# Congratulations!



In this lab, you used Deployment Manager to create a deployment using a template, and you demonstrated Deployment Manager's ability to bring a deployment into compliance with a template. You also used Stackdriver to view resource consumption on a VM instance.