GCP Fundamentals: Getting Started with App Engine

**Overview**

In this lab, you create and deploy a simple App Engine application using a virtual environment in the Google Cloud Shell.

**Objectives**

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

* Install the Cloud SDK for App Engine
* Preview an App Engine application running locally in Cloud Shell.
* Deploy an App Engine application, so that others can reach it.
* Disable an App Engine application, when you no longer want it to be visible.

**Set up your lab environment**

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

1. Make sure you signed into Qwiklabs using an **incognito window**.
2. Note the lab's access time (for example, img/time.png and make sure you can finish in that time block.

There is no pause feature. You can restart if needed, but you have to start at the beginning.

1. When ready, click img/start_lab.png.
2. Note your lab credentials. You will use them to sign in to Cloud Platform Console. 
3. Click **Open Google Console**.
4. Click **Use another account** and copy/paste credentials for **this** lab into the prompts.

If you use other credentials, you'll get errors or **incur charges**.

1. Accept the terms and skip the recovery resource page.

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

Activate Google Cloud Shell

Google Cloud Shell is a virtual machine that is loaded with development tools. It offers a persistent 5GB home directory and runs on the Google Cloud. Google Cloud Shell provides command-line access to your GCP resources.

1. In GCP console, on the top right toolbar, click the Open Cloud Shell button.



1. Click **Continue**. 

It takes a few moments to provision and connect to the environment. When you are connected, you are already authenticated, and the project is set to your *PROJECT\_ID*. For example:



**gcloud** is the command-line tool for Google Cloud Platform. It comes pre-installed on Cloud Shell and supports tab-completion.

You can list the active account name with this command:

gcloud auth list

Output:

Credentialed accounts:

- <myaccount>@<mydomain>.com (active)

Example output:

Credentialed accounts:

- google1623327\_student@qwiklabs.net

You can list the project ID with this command:

gcloud config list project

Output:

[core]

project = <project\_ID>

Example output:

[core]

project = qwiklabs-gcp-44776a13dea667a6

Full documentation of **gcloud** is available on [Google Cloud gcloud Overview](https://cloud.google.com/sdk/gcloud).

**Task 1: Install the Cloud SDK for App Engine**

1. Run the following command to install the gcloud component that includes the App Engine extension for Python 3.7:
2. gcloud components install app-engine-python
3. Initialize your App Engine app with your project and choose its region:
4. 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.

1. Clone the source code repository for a sample application in the **hello\_world** directory:
2. git clone https://github.com/GoogleCloudPlatform/python-docs-samples
3. Navigate to the source directory:
4. cd python-docs-samples/appengine/standard\_python37/hello\_world

**Task 2: Run Hello World application locally**

In this task, you run the Hello World application in a local, virtual environment in Cloud Shell.

Ensure that you are at the Cloud Shell command prompt.

1. Execute the following command to download and update the packages list.
2. sudo apt-get update
3. 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

virtualenv -p python3 venv

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

1. Activate the virtual environment.
2. source venv/bin/activate
3. Navigate to your project directory and install dependencies.
4. pip install -r requirements.txt
5. Run the application:
6. python main.py
7. In **Cloud Shell**, click **Web preview** (Web Preview) > **Preview on port 8080** to preview the application.

To access the **Web preview** icon, you may need to collapse the **Navigation menu**.

Result:



1. To end the test, return to Cloud Shell and press **Ctrl+C** to abort the deployed service.
2. Using the Cloud Console, verify that the app is not deployed. In the Cloud Console, on the **Navigation menu** (), click **App Engine** > **Dashboard**.

Notice that no resources are deployed.

**Task 3: Deploy and run Hello World on App Engine**

To deploy your application to the App Engine Standard environment:

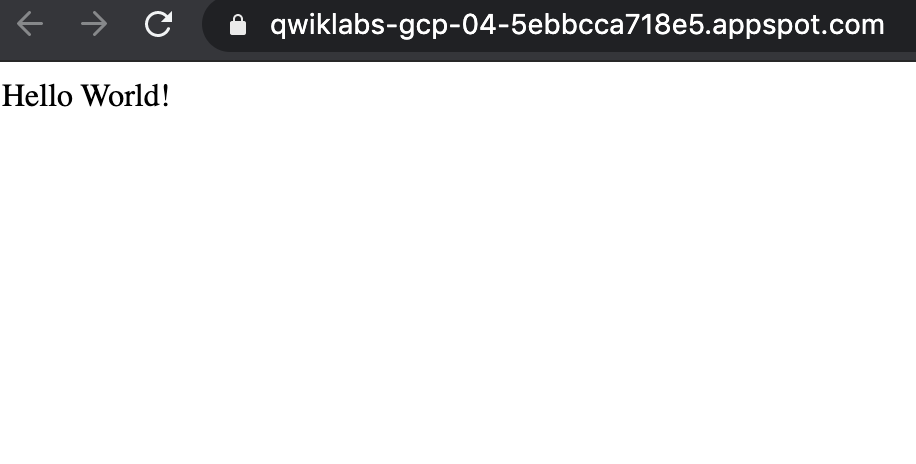
1. Navigate to the source directory:
2. cd ~/python-docs-samples/appengine/standard\_python37/hello\_world
3. Deploy your Hello World application.
4. gcloud app deploy

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

1. Launch your browser to view the app at http://YOUR\_PROJECT\_ID.appspot.com
2. gcloud app browse

Copy and paste the URL into a new browser window.

Result:



Congratulations! You created your first application using App Engine.

Click *Check my progress* to verify the objective.

Deploy the Hello World application to App Engine

Check my progress

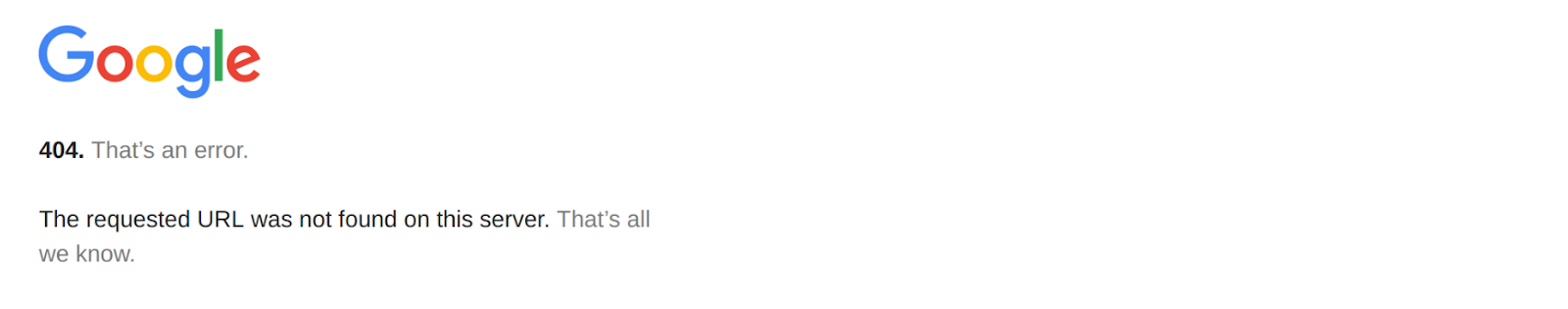
**Task 4: Disable the application**

App Engine offers no option to **Undeploy** an application. After an application is deployed, it remains deployed, although you could instead replace the application with a simple page that says something like "not in service."

However, you can disable the application, which causes it to no longer be accessible to users.

1. In the Cloud Console, on the **Navigation menu** (), click **App Engine** > **Settings**.
2. Click **Disable application**.
3. Read the dialog message. Enter the **App ID** and click **DISABLE**.

If you refresh the browser window you used to view to the application site, you'll get a 404 error.



**Congratulations!**

You created your first application using App Engine!