**Assignment No-2**

**Aim:** Use GAE launcher to launch the web applications.

# Objective:

# Deploying Python web application on Google App Engine

* Understand the process for building and launching web applications in cloud.

# Theory:

The growth in the number of web-based applications and frameworks in the recent times is astounding. As companies such as Google, Amazon, and Microsoft provide more and more easy-to-use tools to build and deploy applications, it makes more sense to use the services and tools provided by them instead of building things in-house and hosting it on-premise.

Google App Engine (GAE) is a great way to get started with learning web development. It provides a bunch of useful features such as sharding, automatic database replication, automatic scaling, memcache, and so on. Google allows you to add your web application code to the platform while managing the infrastructure for you. The engine ensures that your web apps are secure and running and saves them from malware and threats by enabling the firewall.

**Task1: To install and run with Google App Engine**

You need to first install the Google App Engine Software Development Kit (SDK)  on Windows. The App Engine SDK allows you to run Google App Engine Applications on your local computer. It simulates the run-­‐time environment of the Google App Engine infrastructure.

# Pre-­Requisites: Python 2.5.4

If you don't already have Python 3.8.0 installed in your computer, download and Install Python 2.5.4 from:

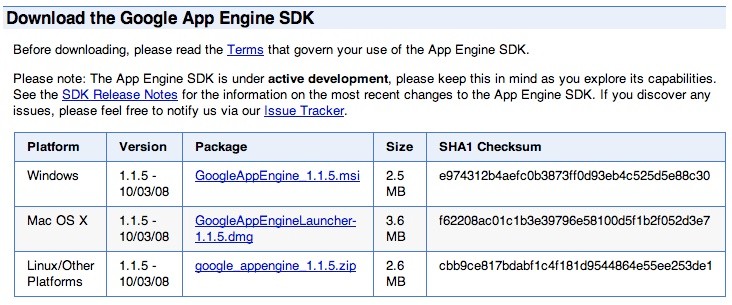
<http://www.python.org/download/releases/3.8.0/>

# Download and Install

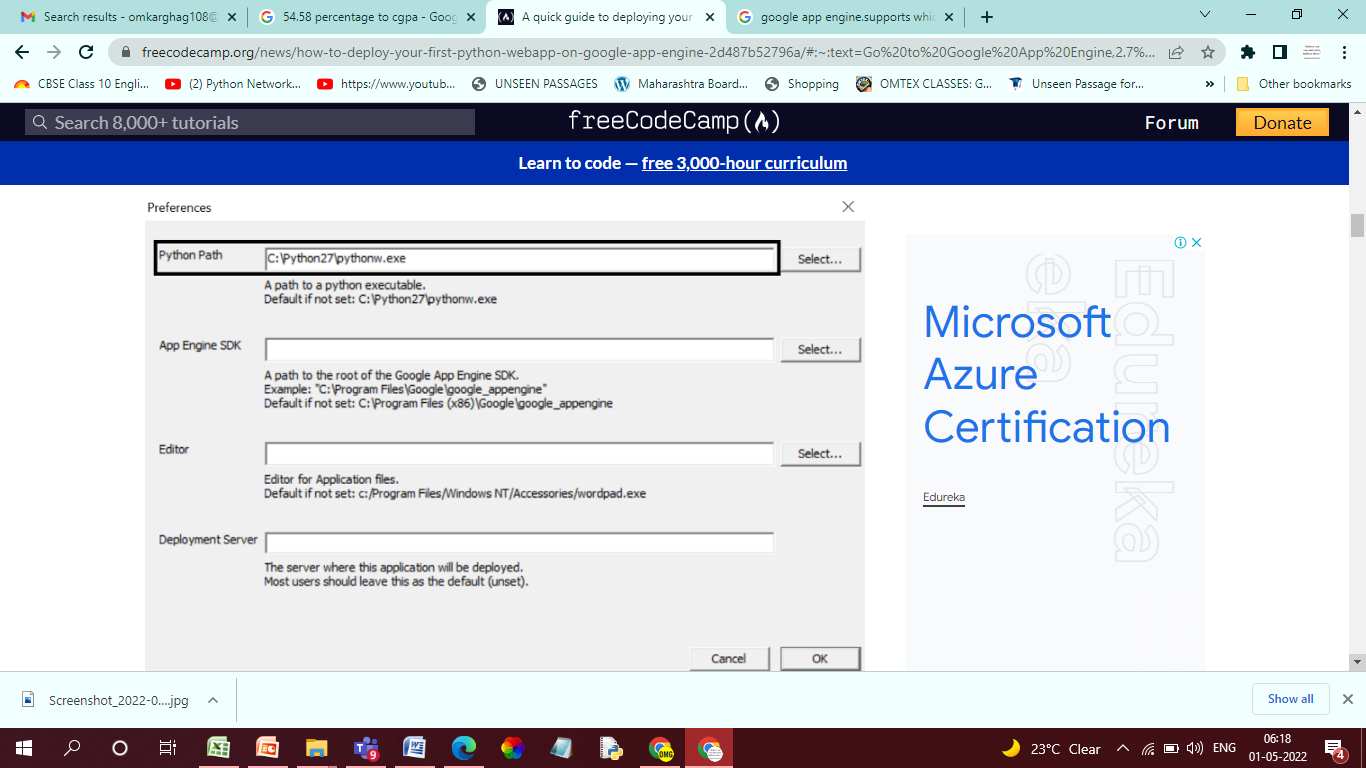
You can download the Google App Engine SDK by going to:

<http://code.google.com/appengine/downloads.html>

and download the appropriate install package.



1. Download the Windows installer – the simplest thing is to download it to your Desktop or another folder that you remember. Set the Python path in the Google App Engine launcher  
   After downloading the SDK, launch the App Engine launcher, go to Edit -> Preferences and make sure you set the path for where you installed Python in step 1 above.



Set the Python path in Google App Engine launcher

### TASK 2. App Engine sign-up

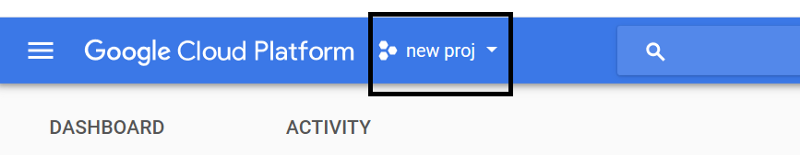
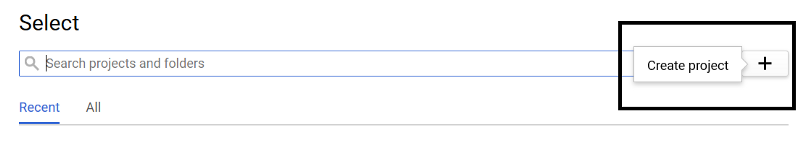
Steps for sign-up are as follows:

1. Go to the Google Cloud landing page
2. Follow the sign-up process and go to your App Engine dashboard

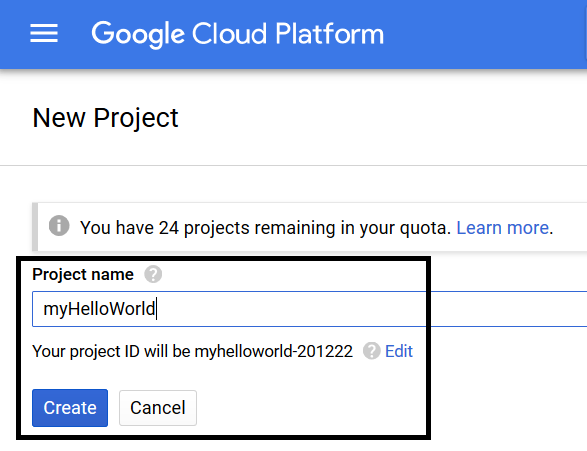
### TASK 3. Create a new project

The next step is to create a new Python project that you can work on. Follow the screenshots below to create a new project.

Launch the new project wizard.

Image courtesy. <https://console.cloud.google.com/home>

Give your app a name and make a note of your project ID.

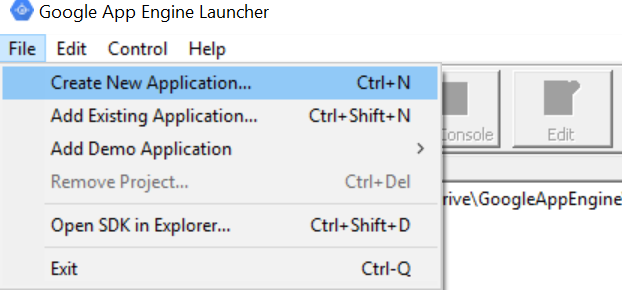


Hit the create button and Google should take a few minutes to set up all that is necessary for your newly created app.

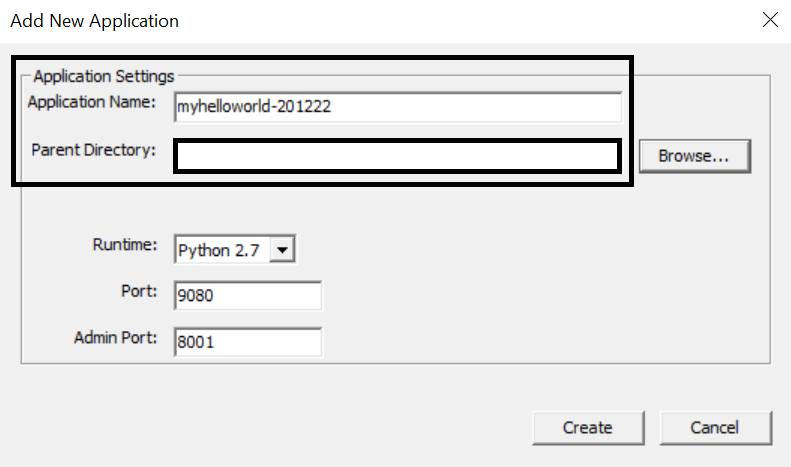
### TASK 4. Fork the app to develop it locally

The next step in the process is to fork the app on your local machine. This will allow you to make changes to the app locally and deploy it whenever you wish to.

Go to Google App Engine launcher and create a new application.



Enter the project ID of your newly created app. Also, provide the folder (local destination) where you wish to store the app locally. Make sure you select the Python 2.7 as your runtime engine.

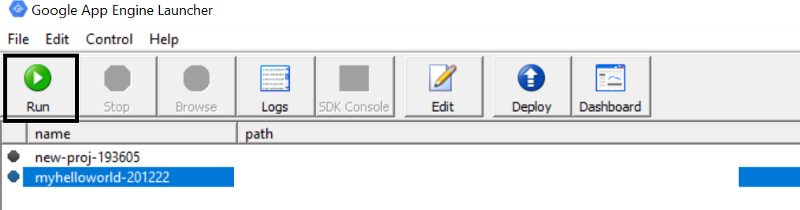


Hit the create button, and you should see your app listed on the window that follows. You should also check that you now see some files in your local storage (the directory you chose in the screenshot above) after this step.

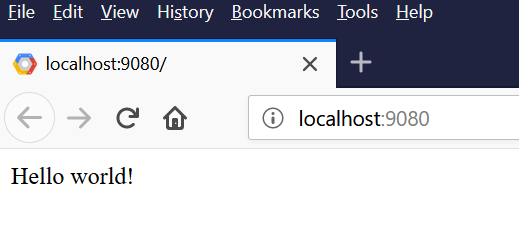
### TASK 5. Run the app locally

Before you go ahead and make some changes to the app, it is important to check whether or not you have executed all the above steps correctly. This can be done by simply running the app locally.

Select the app and hit the run button on the window.



Wait for a few seconds until you can hit the **Browse** button. Once the **Browse**button becomes clickable, click it. This should take you to the browser, and you should see the hello world text appear in your browser window. Alternatively, you can manually go to the browser and use the port specified to access the app.



As long as you see the above screen, you are all set.

### TASK 6. Understand the app structure

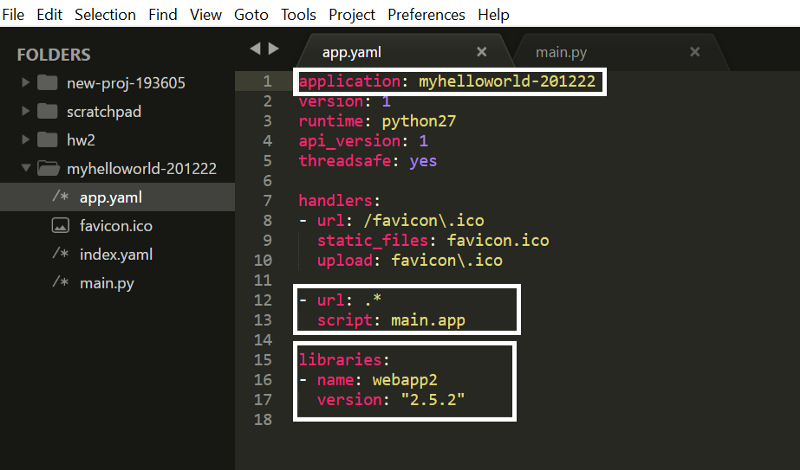
It is finally time to look at the lines of code which are running this webapp. Open your app folder in the text editor of your choice. I recommend [Sublime text](https://www.sublimetext.com/) or [VS Code](https://code.visualstudio.com/). However, feel free to choose the one you prefer.

Here is a description of the various files.

**app.yaml**

This file is a basic markup file that stores information (some metadata) about the app. It is important to note the following crucial parts of the file.

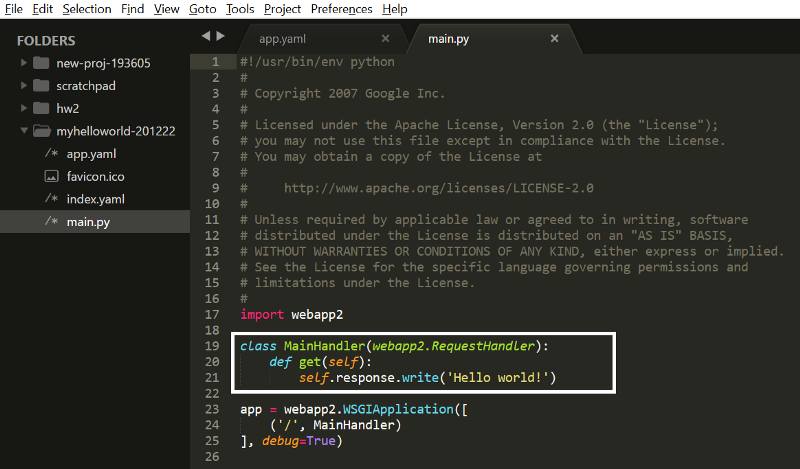
1. **application**  
   This is the project ID which you should never change. This is the unique identifier for the app
2. **url -> scr**ipt  
   This is the homepage for the app. In other words, this file will be rendered in your browser when you launch the app
3. **libraries**  
   This is where you can include external libraries to use within the webapp



app.yaml file in the webapp folder

**main.py**

This is the homepage of the app (as discussed above). Note that the hello world text in the browser window (step 5) is due to the code you see highlighted below.

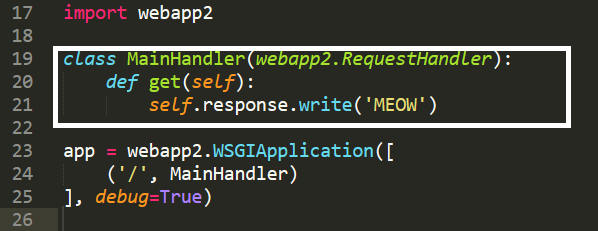


main.py file in the webapp folder

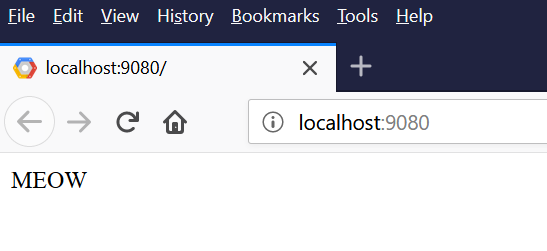
### TASK 7. Make your changes and deploy the new app

No hello world app is ever complete without the developer changing the hello world text to something else just to make sure that everything happening behind the scenes is working as it should.

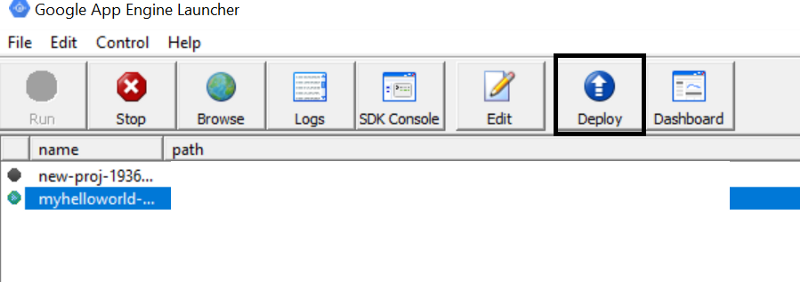
Go ahead and change the text in the above screenshot to something else.

main.py file in the webapp folder

Save the changes, go to the browser and refresh the page. You should see the page with the text “MEOW” displayed.



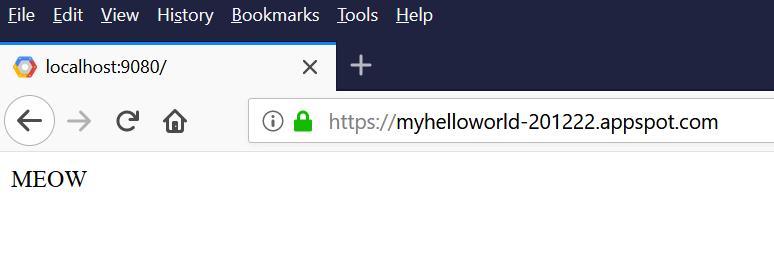
Finally, it is time to deploy your changes to the cloud to make them globally accessible via a URL. Go to the App Engine launcher, select the app, and hit the **Deploy** button.



This will ensure your app gets deployed onto Google Cloud. To check whether or not everything worked just fine, go to the URL below:

**https://<yourProjectID>.appspo**t.com/

You should see the exact same window as above, expect now, it is a URL that is globally accessible.



**Conclusion:** Thus, we have studied about building and launching web applications in cloud.