Homework #5 Google Cloud Platform (GCP) with Python

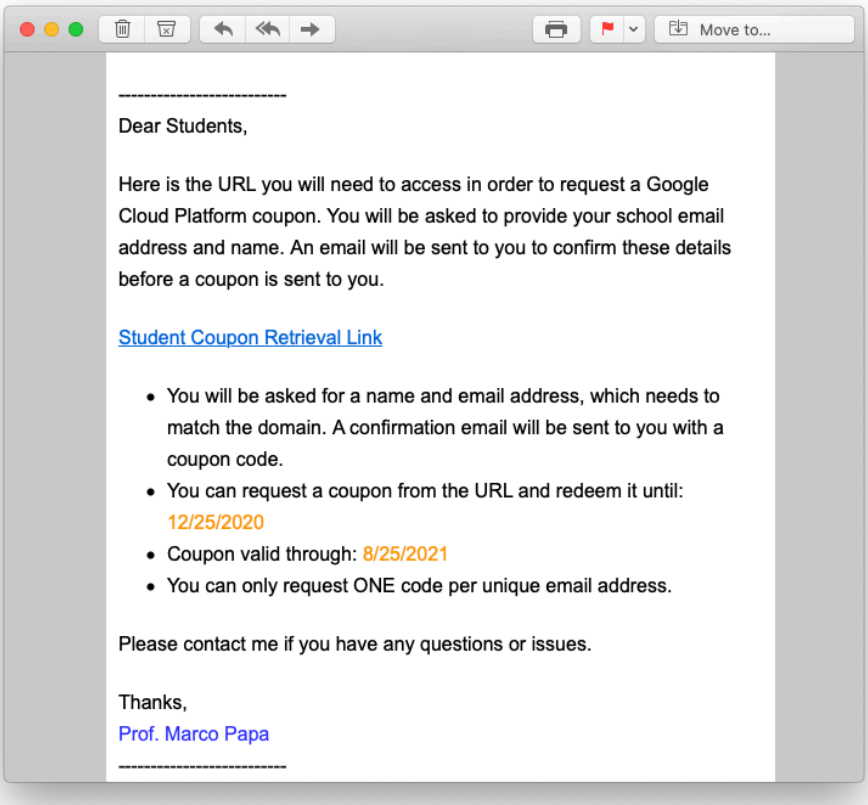
This semester we are allowing all students to explore cloud computing as offered by the Google Cloud Platform using Python. Using the instructions below one can establish a service using Google App Engine. Once established, you will be able to move your Python program developed for Assignment #6 to your Google App Engine instance and have it executed there.

1. Sign up for Google Cloud Platform

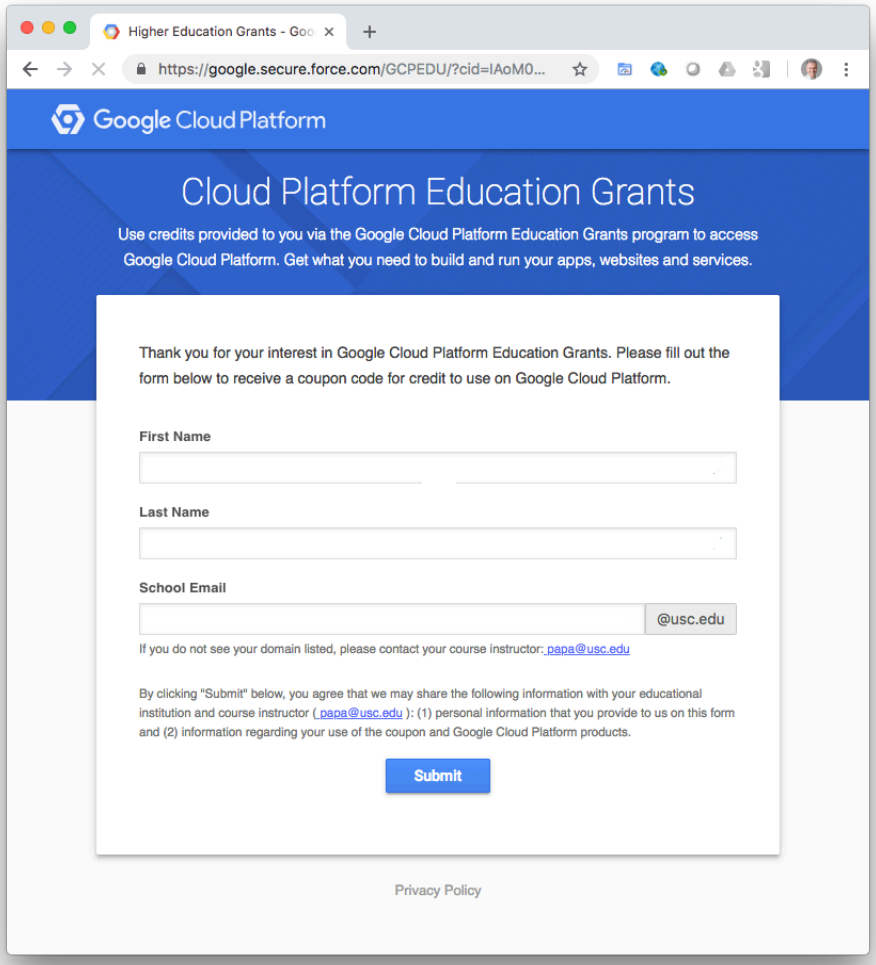
If you do not have a credit card, Google provides you with a coupon code via the Google Cloud Platform Education Grants program (see **section 1.1**). If you do have a credit card, you can sign up for the Google Cloud Platform "Free Trial" (see **section 1.2**).

1.1 Get Google Cloud Platform Education Grants credit

On Piazza and by e-mail, you will receive a communication like the one displayed below. The communication contains information on how to request a Google Cloud Platform coupon. **Click** on the text **Student Coupon Retrieval Link**, or the link provided in the Piazza post.



You will be redirected to a web form as shown below:



Enter your **First Name**, **Last Name** and your **USC e-mail address**. @usc.edu will be pre-filled. **Click** on **Submit**. If you entered a valid USC e-mail address, an email will be sent to that USC email address to verify that you own such address. A sample email is shown below:

Dear Laurie,

Thank you for your interest in downloading a Google Cloud Platform Coupon Code. Please click on this link to verify your email address and a code will be sent to your email account.

Notice that anyone with the URL from USC can request a coupon, so please be careful and do not share the Student Coupon Retrieval Link or the link to verify your email.

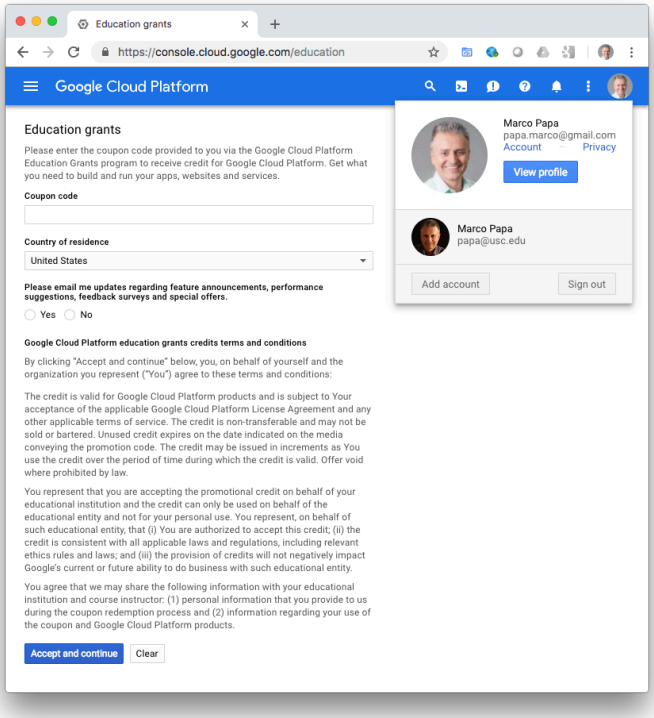
Once your USC email address is “verified”, you will receive a second email with a Google Cloud Platform Coupon Code, as shown below.

Important step: Before clicking on the link labeled **[here]**, you should open your default browser, and **login** to a **Gmail** account. Every USC student has been provided with a Gmail account.

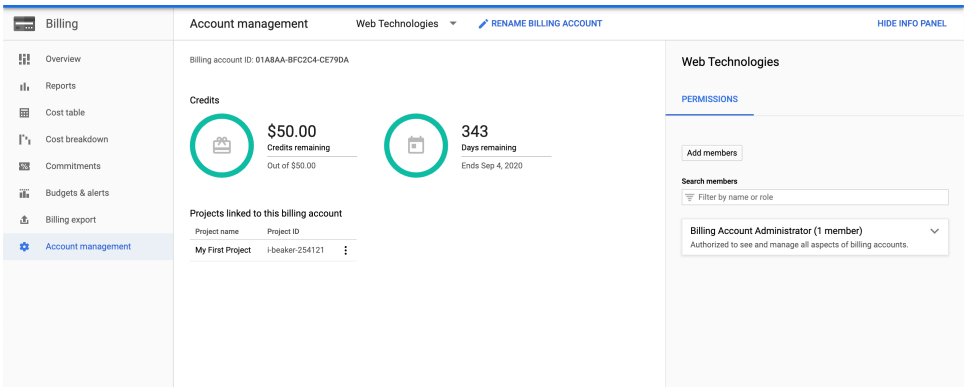
Once logged into Gmail, you can click on **[here]**, or you can go to this page:

https://console.cloud.google.com/education

to redeem your coupon. The web form below will be displayed.



You need to paste your coupon into the field labeled **Coupon code**. Select **Yes** or **No** to receive announcements. Make sure that the active profile in the top right is the one associated with your Gmail account. **Click** on **Accept and continue**. You will now be taken to the Google Cloud Platform's **Home** section. You can navigate to the **Billing** section and navigate to **Account Management** to see the amount of your credit, as shown below.



Important Note: if you have redeemed your coupon with your USC e-mail account, instead of your Gmail account, your coupon will not be usable, as the USC G Suite account does not allow the user to create GCP Projects. If you accidentally did this, you can apply the coupon to the correct billing account, by following the steps in this document:

http://csci571.com/hw/hw5/GCP_G_Suite_Workaround.pdf

1.2 Sign up for Google Cloud Platform Free Trial

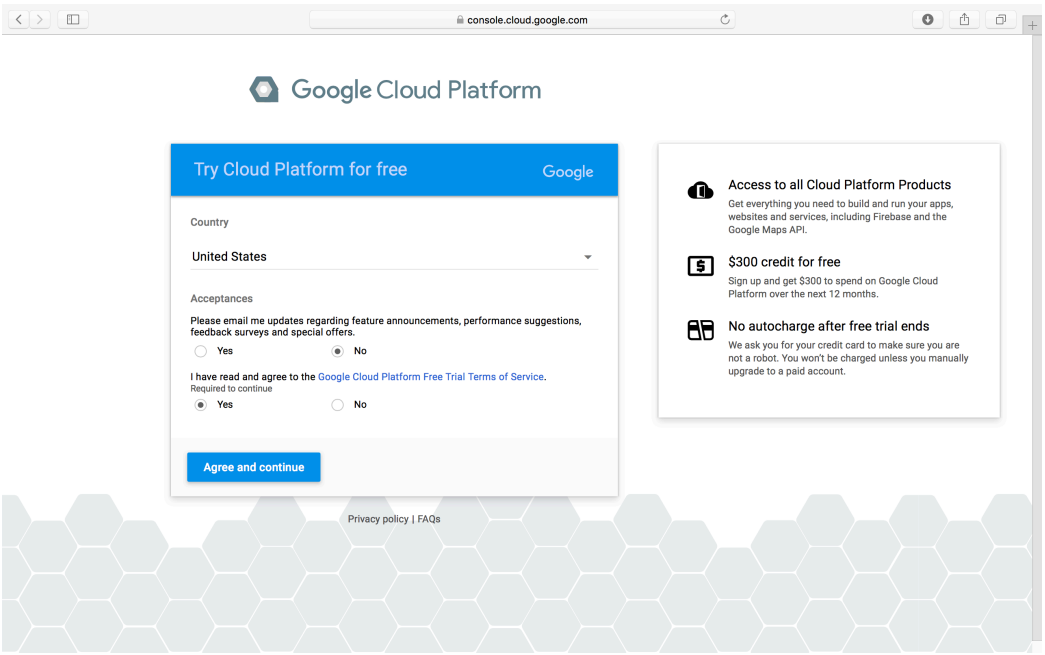
IMPORTANT: you should follow the steps in this section, only of you were unable to obtain the $50 coupon.

To sign up for the Free Trial, with an additional $300 credit, you need a credit card. Unfortunately, an American Express or other pre-paid Gift card will not work with Google Cloud.

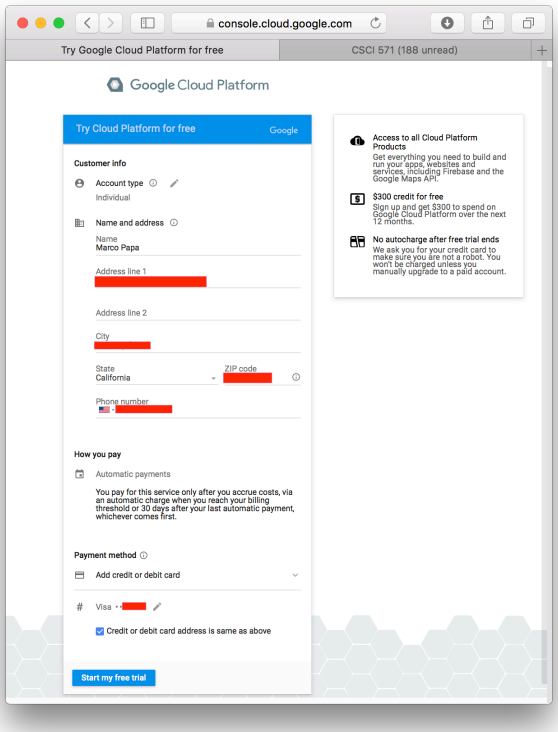
To sign up go to:

https://console.cloud.google.com/freetrial?pli=1&page=0

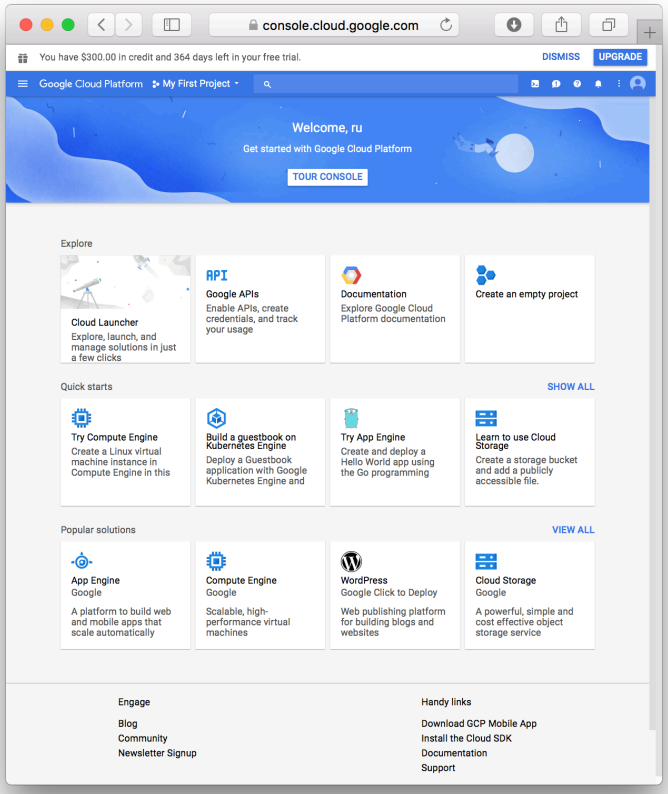
In the Try Cloud Platform for free page, select **Yes** under "I have read and agree to the Google Cloud Platform Free Trial terms of Service" and click on **Agree and continue**.



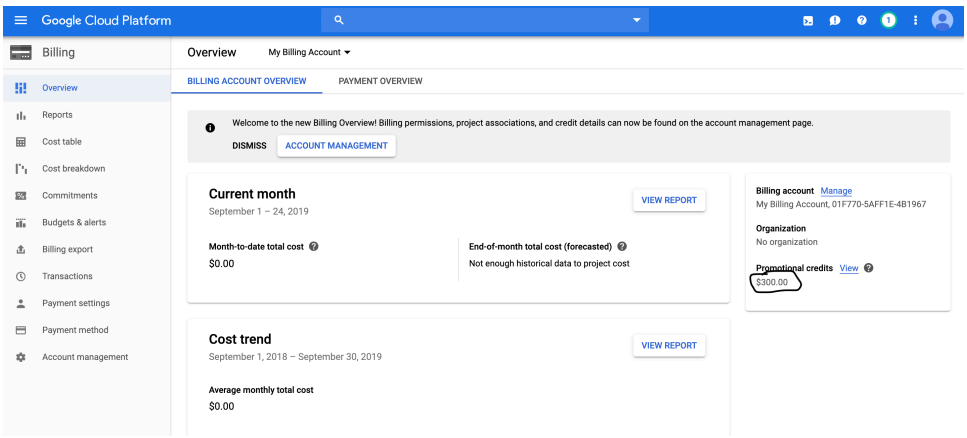
Select **Account type Individual**. Follow the instructions to enter your account data. You should **not** be using your @usc.edu e-mail account for your primary contact e-mail address, but instead use your @gmail.com address and finish by clicking **Start my free trial**. You will have to provide a credit or debit card.



After you are signed up, you will see the message “Creating project. This may take a few moments”.” You will then be redirected to the **Dashboard** of the **Google Developer Console**.



To confirm your credits, navigate to **Billing > Account Management** from the left navigation bar to see a credit value of $300 valid for 365 days or you can verify it as below.



If you previously developed any projects using Google APIs, you will find them listed.

1.3. How to get additional student Coupons

If you follow our instructions to install Python (and late on Node.js) you will likely never incur charges that exceed the value of your coupons. But there are always students that want to play around and run services all over the place.

When a student exceeds 60% of the value of a coupon, Google sends a notification, by e-mail, to the instructor. The instructor can get additional coupons for the student by filling out the same form listed on page 2, using the instructor's e-mail address that was used to obtain the grant. The instructor will receive the coupon and deliver it to the student by e-mail.

Google limits the additional coupons to 2 for each student account used in a given course.

2. Setting up a Python development environment

To set up a Python development environment for GCP to develop Python apps that run on Google Cloud, you should follow the steps from this tutorial:

https://cloud.google.com/python/setup

The tutorial covers all the following:

- Install the latest version of Python.
- Use `venv` to isolate dependencies.
- Install an editor (optional).
- Install the Cloud SDK (optional).
- Install the Cloud Client Libraries for Python (optional).
- Install other useful tools.

2.1 Installing Python

The tutorial provides steps to install the latest version of Python 3 on macOS, Windows and Linux.

Installing on macOS

macOS includes a version of Python by default and uses it for its own purposes (normally version 2.7.X). Verify your Mac's Python installation using the following command:


```
/usr/bin/env python -V
```

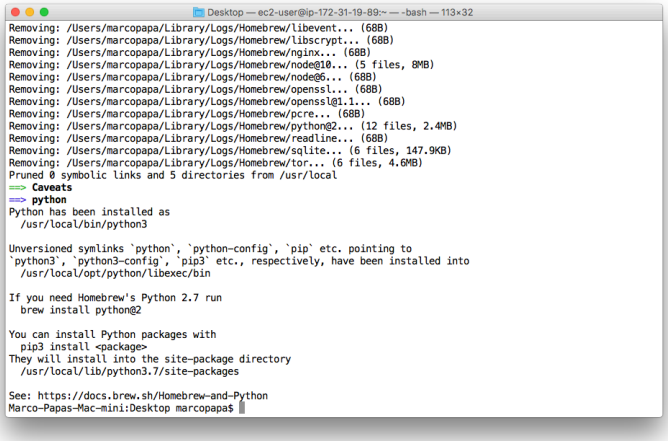
To avoid interfering with macOS, we recommend creating a separate development environment and installing the latest version of Python (version 3.7). To install Python, use **Homebrew**, available at:

https://brew.sh/

After installing Homebrew, you can install the latest Python with:

```
brew install python
```

As of this writing, Homebrew will install **Python 3.7.6**. If all is well, the installation will complete, as shown below.



Normally Python 3 will be installed in `/usr/local/bin/python3`. If you have kept the default Python 2.7, you will have to add aliases to your startup files (for Bash and Zsh) for Python 3 and Pip 3 locations, run the following commands:

```
echo "alias python=/usr/local/bin/python3.7" >> ~/.zshrc
echo "alias python=/usr/local/bin/python3.7" >> ~/.bashrc
echo "alias pip=/usr/local/bin/pip3" >> ~/.zshrc
echo "alias pip=/usr/local/bin/pip3" >> ~/.bashrc
```

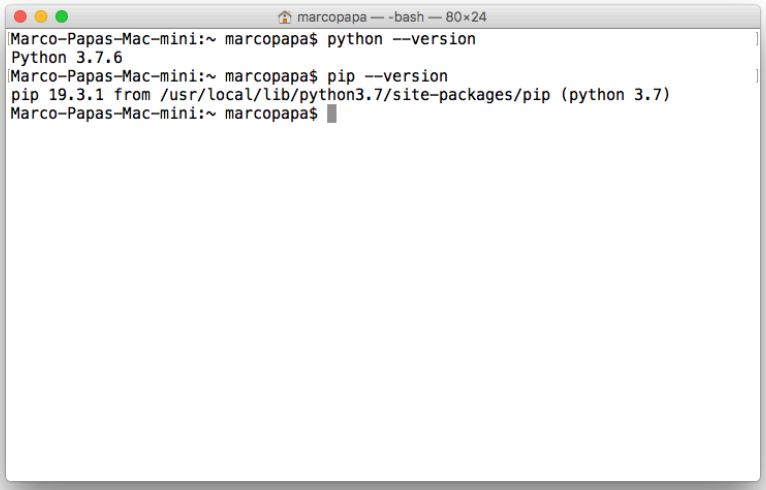
If you are using bash and ~/.bash_profile, you may have to do this instead:

```
echo "alias python=/usr/local/bin/python3.7" >> ~/.bash_profile
echo "alias pip=/usr/local/bin/pip3" >> ~/.bash_profile.
```

You will have to re-start Terminal, so the aliases will take effect. Start and new terminal session and verify Python 3 is available as python and python3, and that pip is also installed, by running the following commands:

```
python -version
```

```
pip --version
```



Installing on Windows

Since Windows does not come with Python, download the installers for the latest versions of Python from the Python website at:

https://www.python.org/downloads/windows/

as of this writing, we recommend you download **Python 3.7.6**, the same version that we recommend for macOS. Complete the installation by adding the proper PATH and verifying the version of Python 3 and pip installed, as outlined in the tutorial.

2.2 Use venv to isolate dependencies

venv is a tool that creates isolated Python environments. Use the **venv** command to create a virtual copy of the entire Python installation.

Follow the tutorial to do the following:

- Create a *virtual copy* in a folder named venv
- Set your shell to use the venv paths for Python by *activating* the virtual environment
- *Install packages* without affecting other projects or your global Python installation
- If you want to stop using the virtual environment and go back to your global Python, you can *deactivate* it

2.3 Installing a Python editor

There are several, popular editors for Python. In particular **Sublime Text**, **Atom** and **PyCharm**. We recommend that you use **PyCharm**, as it is free for students from JetBrains, and available at:

https://www.jetbrains.com/pycharm/

The free "educational" version of **PyCharm** can be downloaded here:

https://www.jetbrains.com/education/download/#section=pycharm-edu

PyCharm is available for macOS, Windows and Linux.

3. Creating a Project and Application using CLI

The Cloud SDK is a set of command-line tools for Google Cloud. It contains **gcloud**, and **gsutil**, which you can use to access App Engine, Compute Engine, Cloud Storage, and other products and services from the command line. The Cloud SDK is available at:

https://cloud.google.com/sdk/

The Cloud SDK is available for **Linux**, Ubuntu, CentOS, **macOS** and **Windows**. Quickstarts for each platform are available here:

https://cloud.google.com/sdk/docs/quickstarts

1. The "*Quickstart for Python 3 in the App Engine Standard Environment*" page is available at:

https://cloud.google.com/appengine/docs/standard/python3/quickstart

2. The QuickStart tutorial provides all the steps needed to do all the following:

    - Downloading and installing the Cloud SDK
    - Creating a new project
    - Initialize App Engine app
    - Enable billing for the project
    - Downloading and installing Git
    - Install the App Engine extension for Python 3
    - Download the Hello World app written with **Flask**
    - Run Hello World on your local machine
    - Deploy and run Hello World on App Engine

- Clean-up to stop billing

3. Download and install the **Google Cloud SDK** version for your platform (Mac OS, Windows) from:

   https://cloud.google.com/sdk/docs/

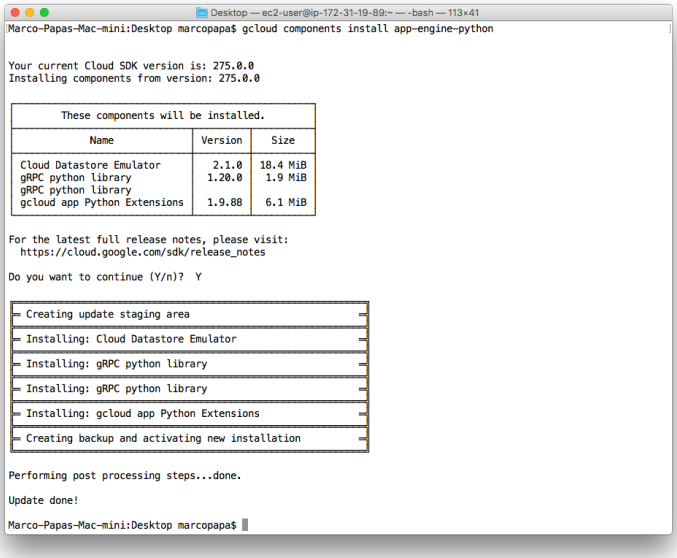
   While in most of the GCP docs it is documented that `gcloud` needs to run on Python 2, according to the Note in the above site "As of Cloud SDK version 274.0.0, the `gcloud` CLI has GA support for running using a **Python 3.5 and up** interpreter (run `gcloud topic startup` for exclusions and more information on configuring your Python interpreter)."

4. Extract the file on your local file system. Initialize the `gcloud` tool to initialize the SDK:

```
gcloud init
```

5. Run a command to install the cloud component that includes the App Engine extension for Python:

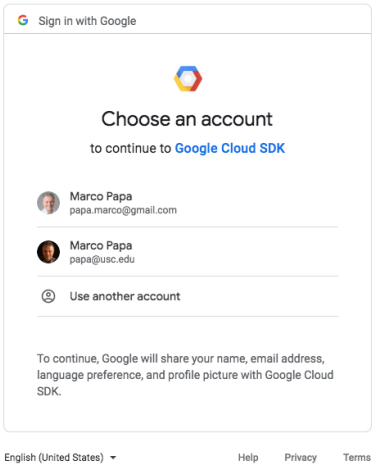
```
gcloud components install app-engine-python
```

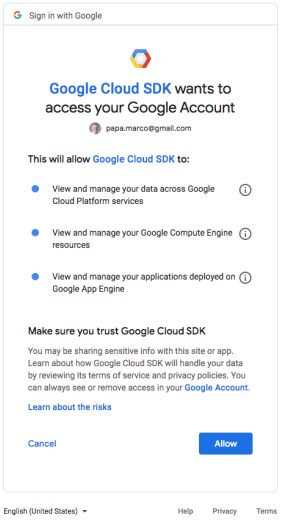


6. Login to GCP with the CLI:

```
gcloud auth login
```

You will be asked to **Choose an account** and **Allow** access as shown below.



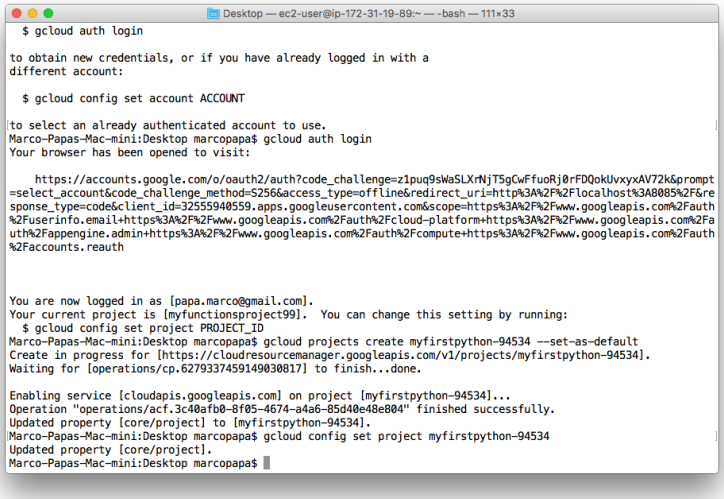


7. Create a new project using this command syntax:

```
gcloud projects create [YOUR_PROJECT_ID] --set-as-default
```

Project IDs must start with a lowercase letter and can have lowercase ASCII letters, digits or hyphens. Project IDs must be between 6 and 30 characters. For example:

```
gcloud projects create myfirstpython-94534 --set-as-default
```



8. Verify the project was created and see its details:

```
gcloud projects describe myfirstpython-94534
```

For example, you'll see something like this:

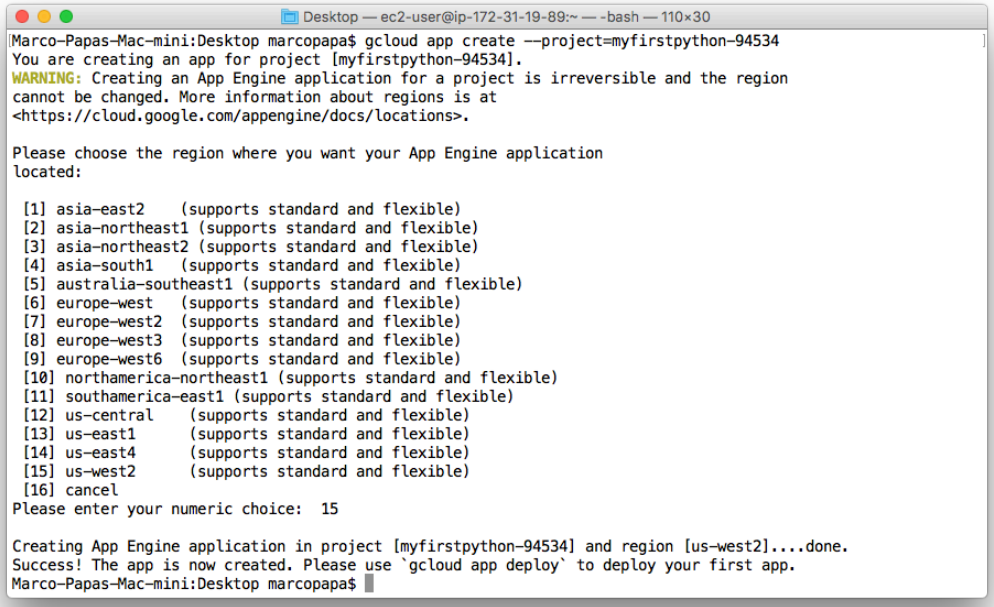

```
createTime: '2020-01-08T18:34:36.846Z'
lifecycleState: ACTIVE
name: myfirstpython-94534
projectId: myfirstpython-94534
projectNumber: '675437181434'
```

9. Initialize the App Engine app with your newly created project and choose its region (or example us-west2):

```
gcloud app create --project=[YOUR_PROJECT_ID]
```

for example:

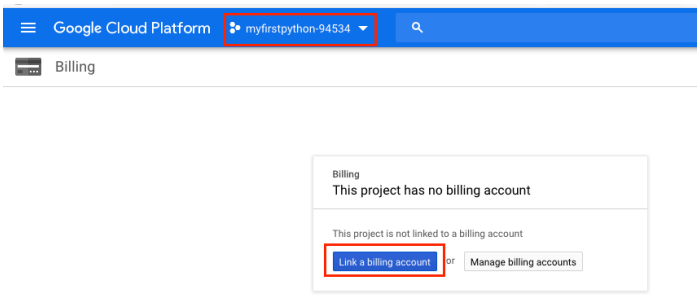
```
gcloud app create --project= myfirstpython-94534
```



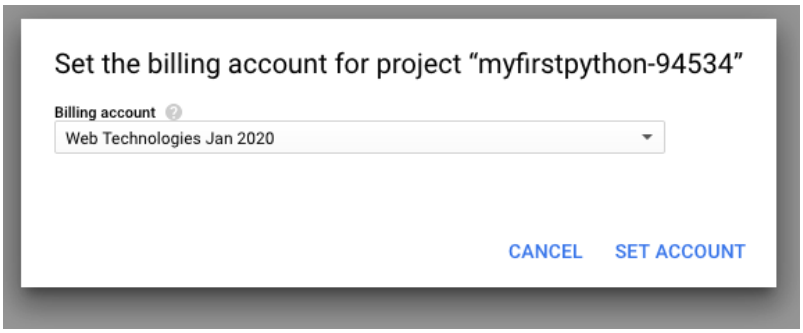
10. **Enable billing** for the project. You will do this in the Cloud console at:

https://console.cloud.google.com/projectselector/billing?lang=python3

You will have to select the project and click **Link a billing account**.



Select the billing account you created with your Google credits.



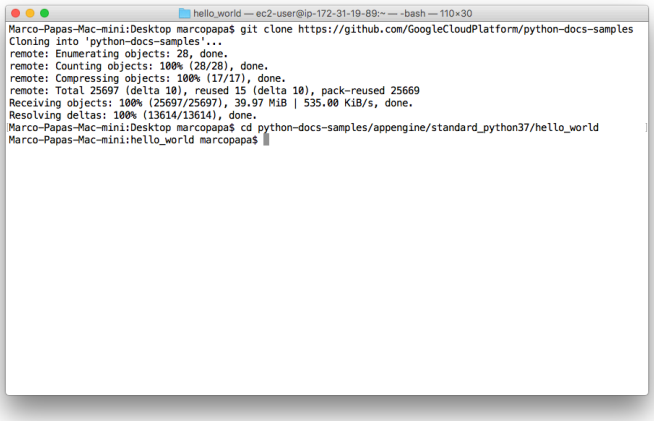
11. Install **Git**

12. Download the Hello World app from Github:

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

13. Change to the directory that contains the sample code:

```
cd python-docs-samples/appengine/standard_python37/hello_world
```



14. Test the app on your local machine:

a. **Windows ONLY**: download and install **PowerShell** as indicated in the tutorial.

b. Create an isolated Python environment

```
python3 -m venv env
```

```
source env/bin/activate
```

(Note: run `env/Scripts/activate` in Windows)

c. If the file requirements.txt does not exists, create it with:

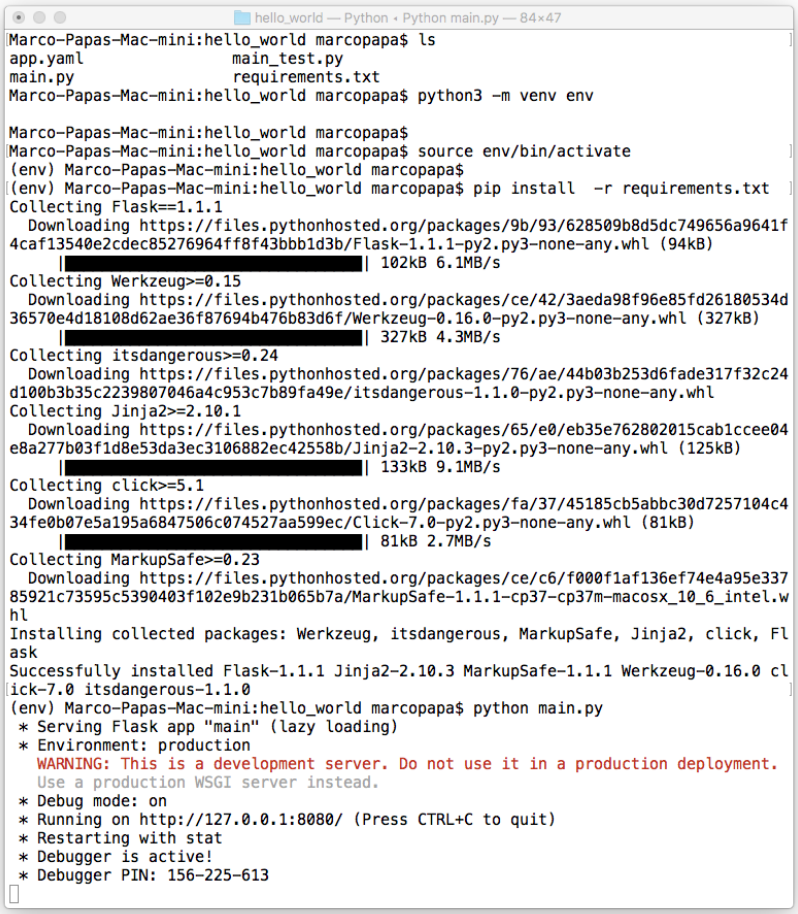
```
pip freeze > requirements.txt
```

d. Install dependencies (this step will install Flask):

```
pip install -r requirements.txt
```

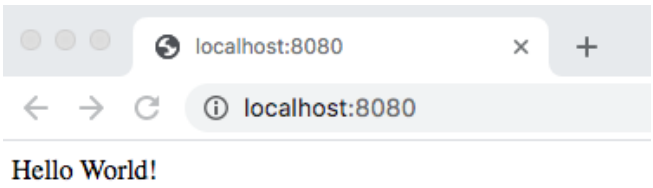
e. Run the application:

```
python main.py
```

f. Open the app in your browser

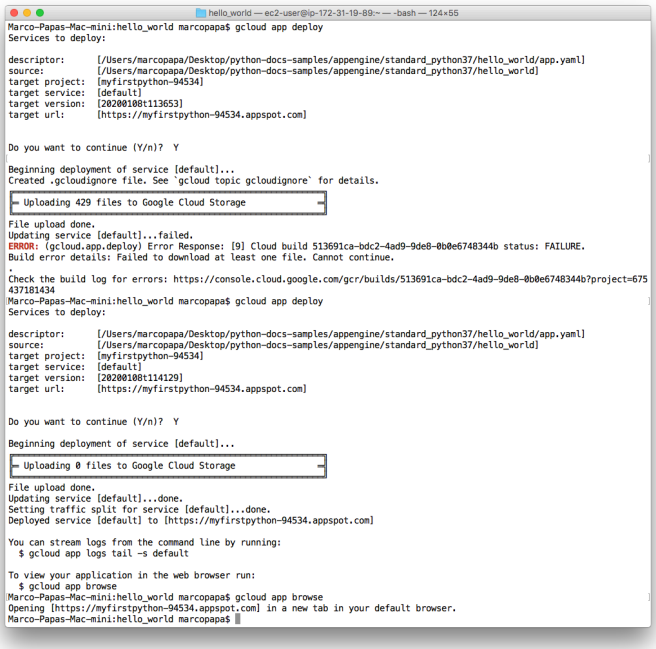
http://localhost:8080



Type CTRL-C to quit serving locally the Flask app.

15. Deploy and run Hello World on App Engine:

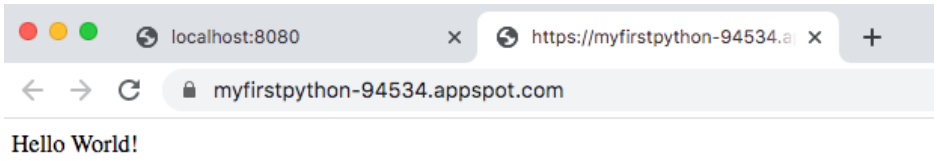
```
gcloud app deploy
```

16. View your application in the cloud. Launch your browser with the app at http://[YOUR_PROJECT_ID].[**REGION_ID**].r.appspot.com,

running the command:

```
gcloud app browse
```



Or type the URL in the browser:

```
https://myfirstpython-94534.us-west2.r.appspot.com/
```

17. Clean up. First stop using the virtual environment. Type this to the (env) prompt:

```
deactivate
```

18. To avoid incurring charges, **delete your Cloud Platform project** to stop billing on all resources.

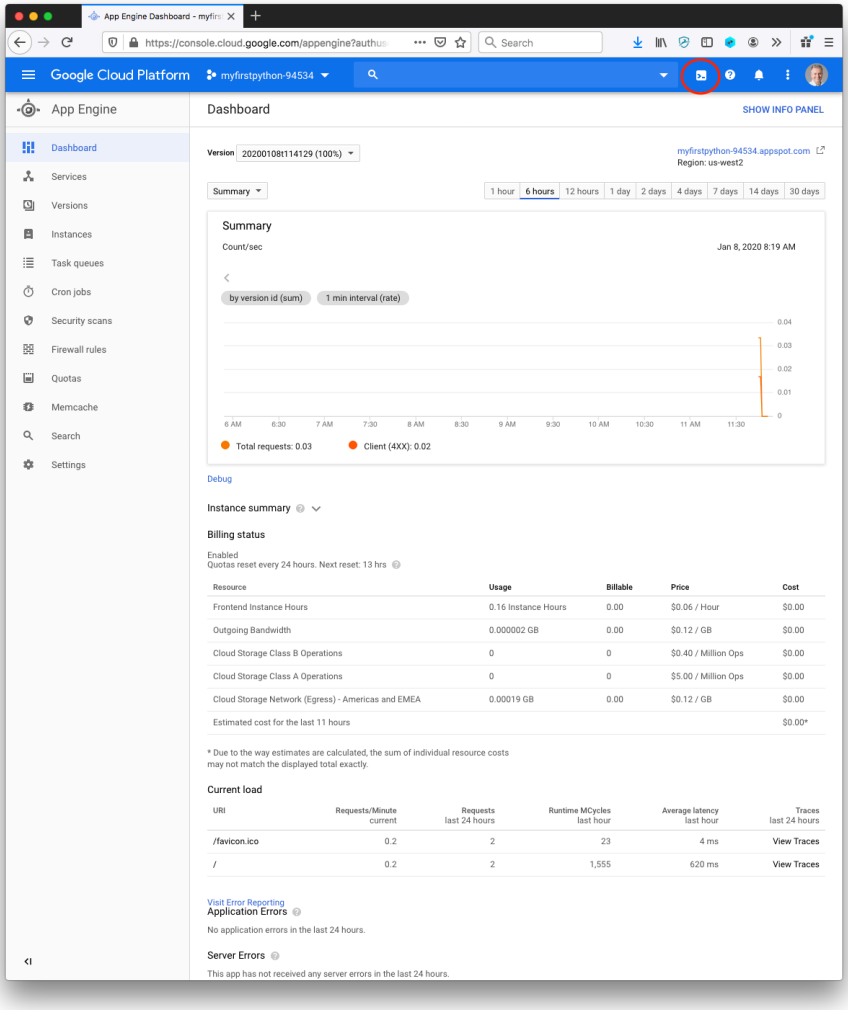
19. The “Hello World” program comes with a file named `requirements.txt`. This file needs to be deployed to GCP. Once you add your code and add some Python libraries, this file needs to be updated. Use pip to install your libraries locally. Then run the following pip command:

    ```
    pip freeze > requirements.txt
    ```

    This pip command will update the `requirements.txt` file with all the needed libraries. The local Python libraries should not be uploaded and deployed to CGP. Instead the libraries included in the deployed `requirements.txt` file will be automatically downloaded and installed by GCP. Every time you add a new library to your local copy, you need to run “pip freeze” before deploying to GCP.

4. Check App Engine Dashboard

Click on “triple bar” on top left of the GCP console. Select App Engine. Select your Project ID.

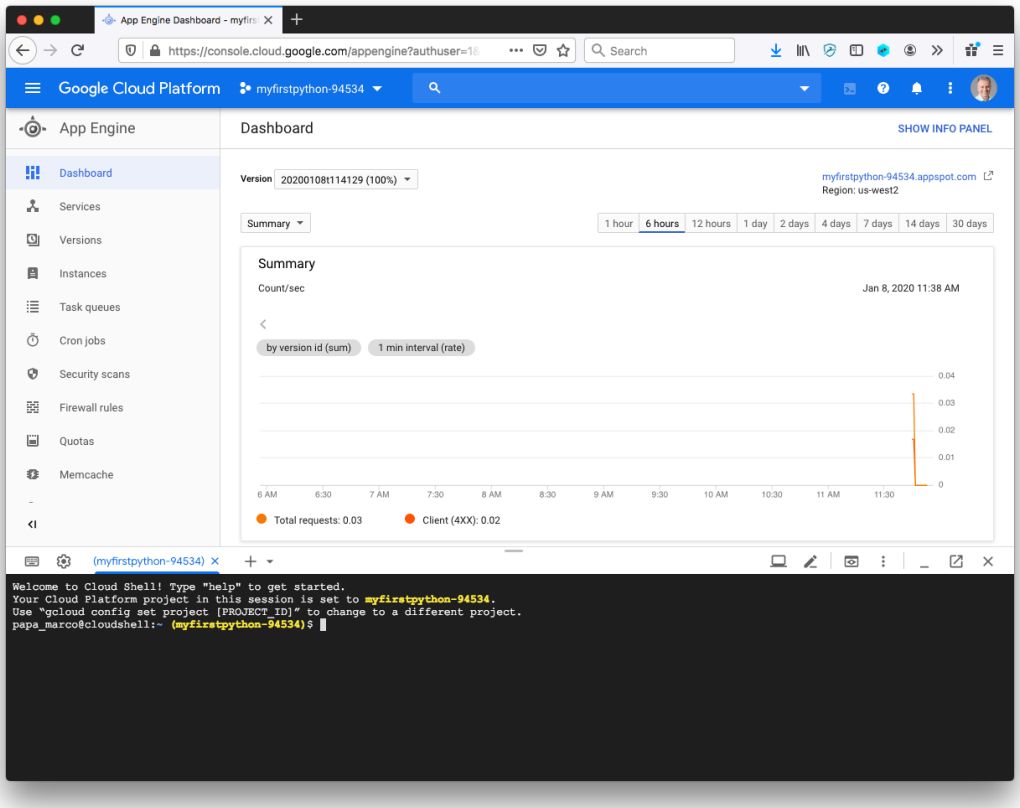


5. Set up Exploring Your instance (Optional)

If you want to explore your server instance you can activate the **Google Cloud Shell**.

Go to the App Engine Dashboard. Select the Hello World project from the dropdown. Now click on the **Activate Google Cloud Shell** icon in the top toolbar (see picture above).

After waiting a few minutes for Google to establish the connection, you will see the shell appear at the bottom of the browser window. You can now use Linux commands to manage your Cloud Platform Console projects and resources.

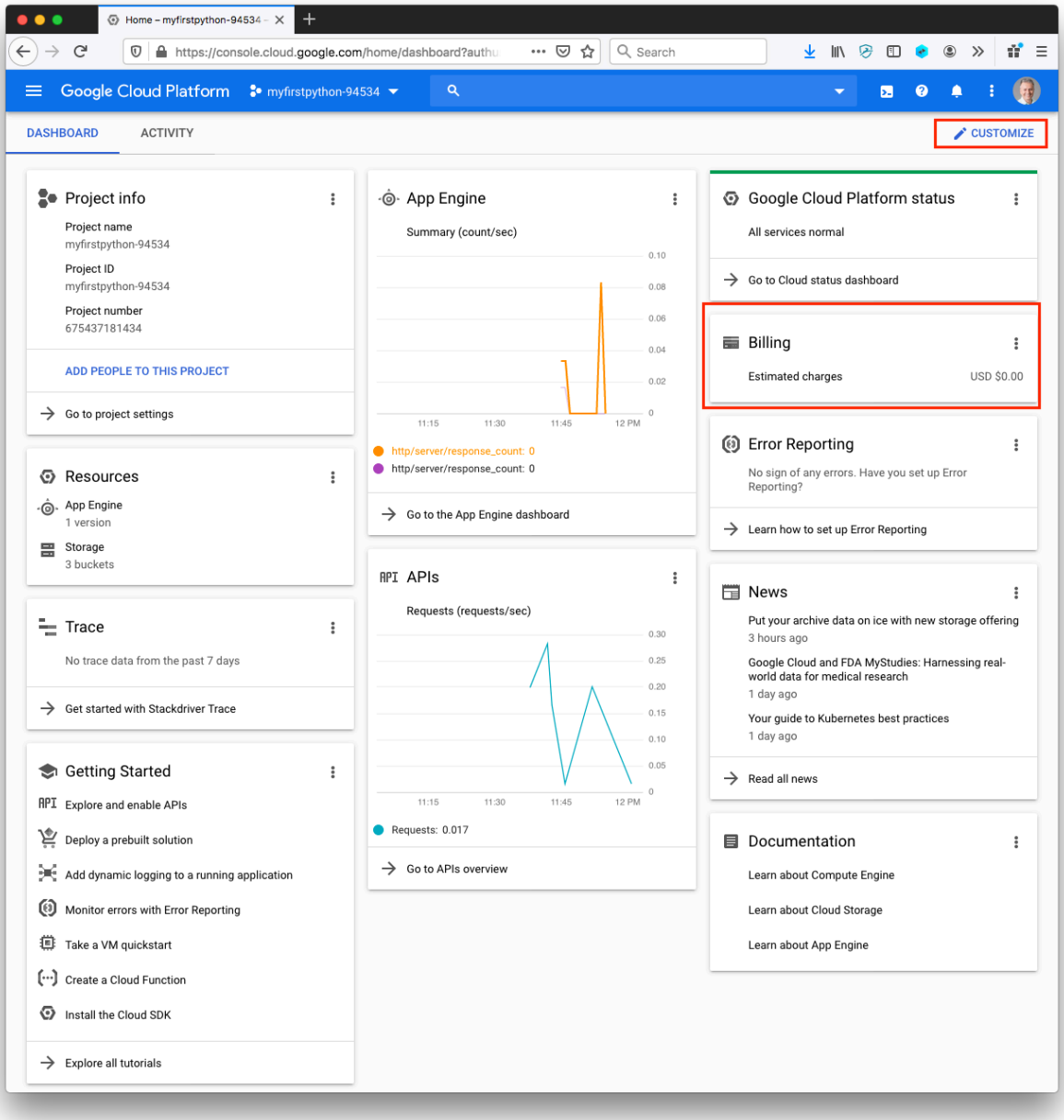


You can read more about the **Google Cloud Shell** here:

https://cloud.google.com/cloud-shell/docs/

6. Monitoring your instance and you bill

Select Google Cloud Platform and go to the Dashboard. If you do not see a **Billing** “tile”, click **CUSTOMIZE** in the upper left toolbar. Turn on the billing tile “switch” and click **DONE**. Under **Billing** you will see if you are incurring any charges. Hou will likely see $0.00 estimated charges.



Have fun exploring Google Cloud Platform!!