

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.

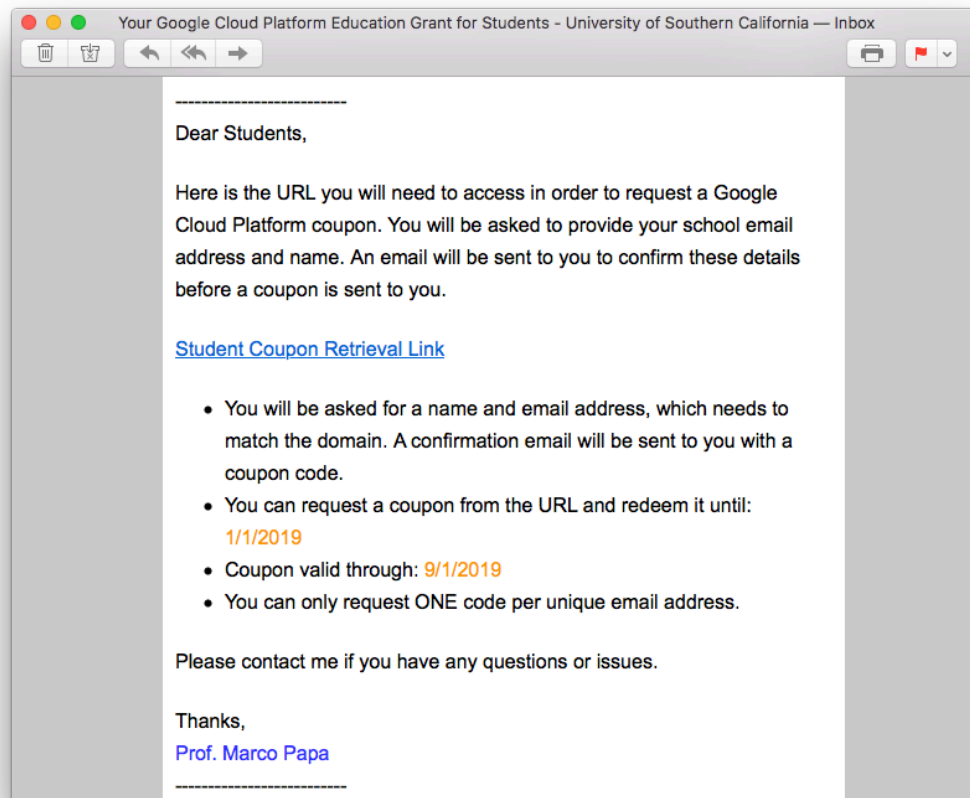
1. Sign up for Google Cloud Platform

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**).

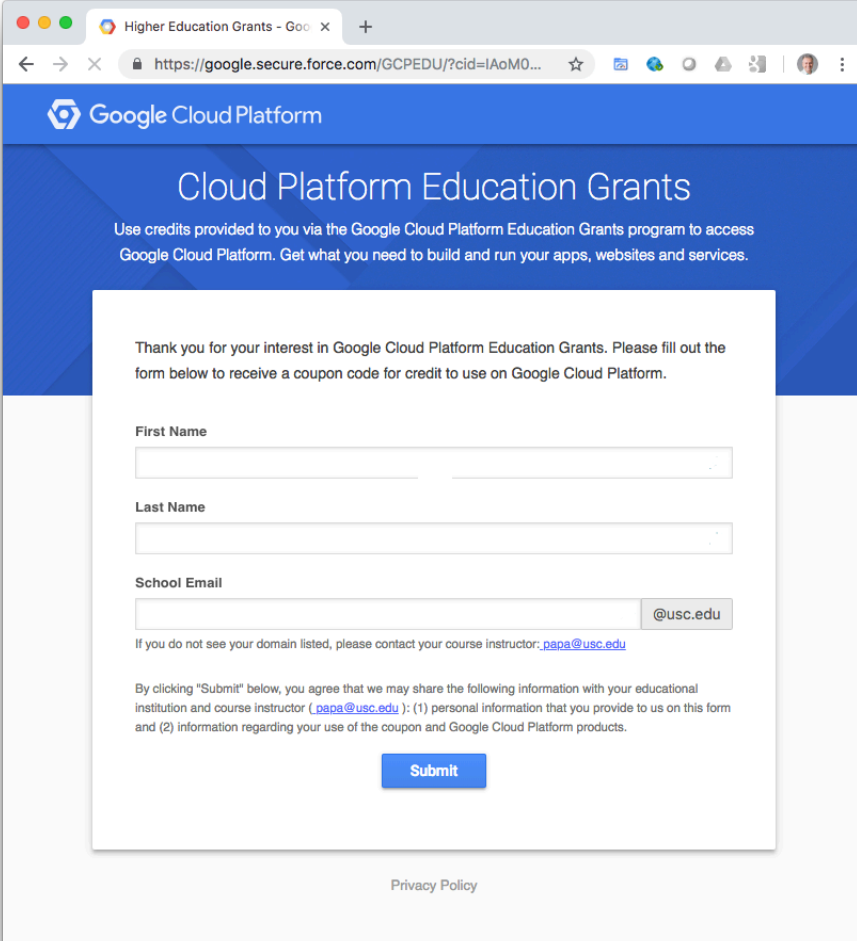
1.1 Get Google Cloud Platform Education Grants credit

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**.



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



The screenshot shows a web browser window with the address bar displaying "https://google.secure.force.com/GCPEDU/?cid=IAoM0...". The page has a blue header with the Google Cloud Platform logo and the text "Cloud Platform Education Grants". Below the header, there is a white box containing the following text: "Thank you for your interest in Google Cloud Platform Education Grants. Please fill out the form below to receive a coupon code for credit to use on Google Cloud Platform." The form has three input fields: "First Name", "Last Name", and "School Email". The "School Email" field has a dropdown menu with "@usc.edu" selected. Below the form, there is a "Submit" button. At the bottom of the page, there is a "Privacy Policy" link.

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.

Dear Laurie,

Here is your Google Cloud Platform Coupon Code: 4G8B-E0XC-6J2H-65TR

Click [\[here\]](#) to redeem.

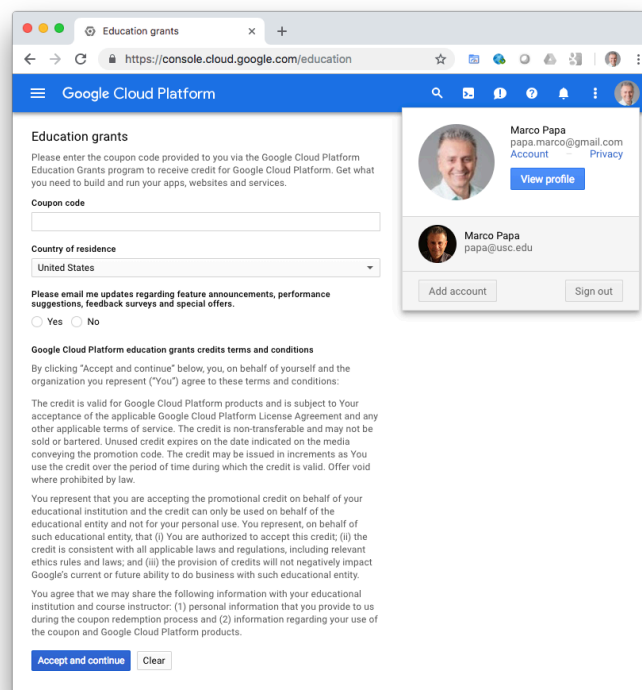
Danger!

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:

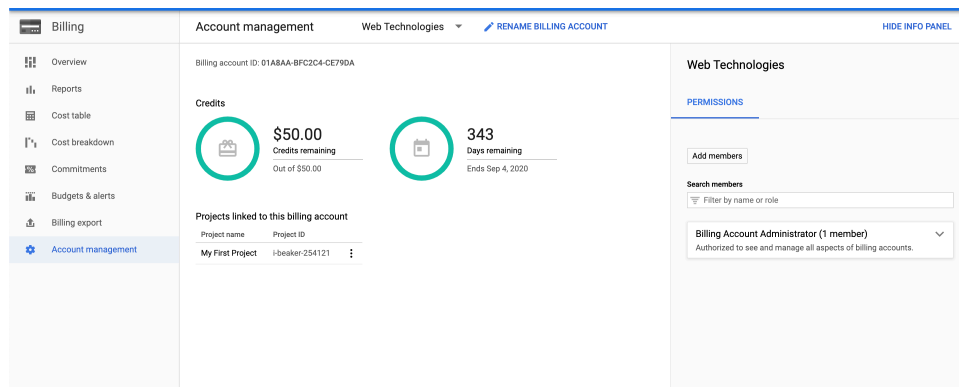
<https://console.cloud.google.com/education>

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



The screenshot shows a web browser window with the URL <https://console.cloud.google.com/education>. The page is titled "Education grants" and contains a form for redeeming a coupon code. The form includes fields for "Coupon code", "Country of residence" (set to "United States"), and a checkbox for "Please email me updates regarding feature announcements, performance suggestions, feedback surveys and special offers." (set to "No"). Below the form is a section for "Google Cloud Platform education grants credits terms and conditions" with an "Accept and continue" button. On the right side of the page, there is a user profile for "Marco Papa" with email "papa.marco@gmail.com" and a "View profile" button. Below the profile, there are buttons for "Add account" and "Sign out".

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

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**.

Google Cloud Platform

Try Cloud Platform for free

Country

United States

Acceptances

Please email me updates regarding feature announcements, performance suggestions, feedback surveys and special offers.

☐ Yes ☒ No

I have read and agree to the [Google Cloud Platform Free Trial Terms of Service](#).

Required to continue

☒ Yes ☐ No

Agree and continue

Access to all Cloud Platform Products

Get everything you need to build and run your apps, websites and services, including Firebase and the Google Maps API.

\$300 credit for free

Sign up and get \$300 to spend on Google Cloud Platform over the next 12 months.

No autocharge after free trial ends

We ask you for your credit card to make sure you are not a robot. You won't be charged unless you manually upgrade to a paid account.

Privacy policy | FAQs

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.

Google Cloud Platform

Try Cloud Platform for free

Customer info

Account type ☒ Individual

Name and address

Name

Marco Papa

Address line 1

Address line 2

City

State

California

ZIP code

Phone number

How you pay

☒ Automatic payments

You pay for this service only after you accrue costs, via an automatic charge when you reach your billing threshold or 30 days after your last automatic payment, whichever comes first.

Payment method

☒ Add credit or debit card

Visa

☒ Credit or debit card address is same as above

Start my free trial

Access to all Cloud Platform Products

Get everything you need to build and run your apps, websites and services, including Firebase and the Google Maps API.

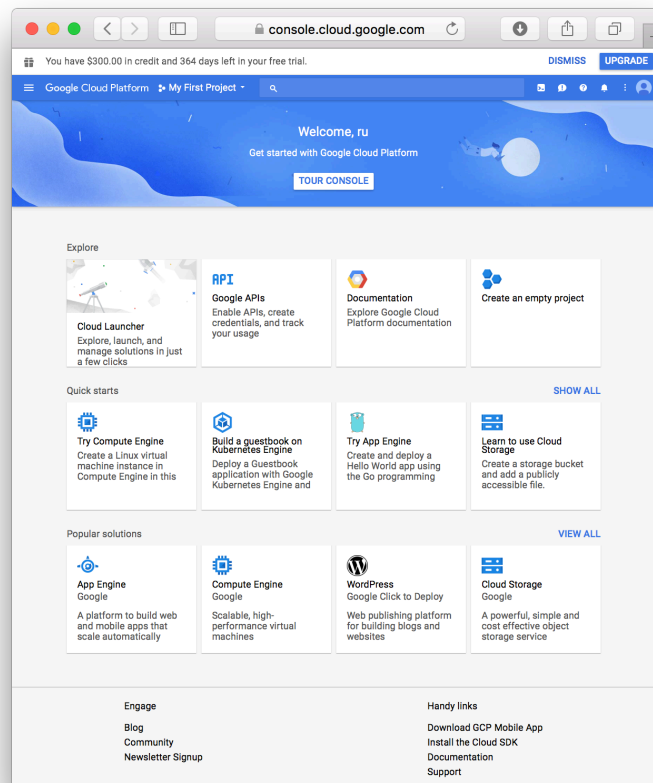
\$300 credit for free

Sign up and get \$300 to spend on Google Cloud Platform over the next 12 months.

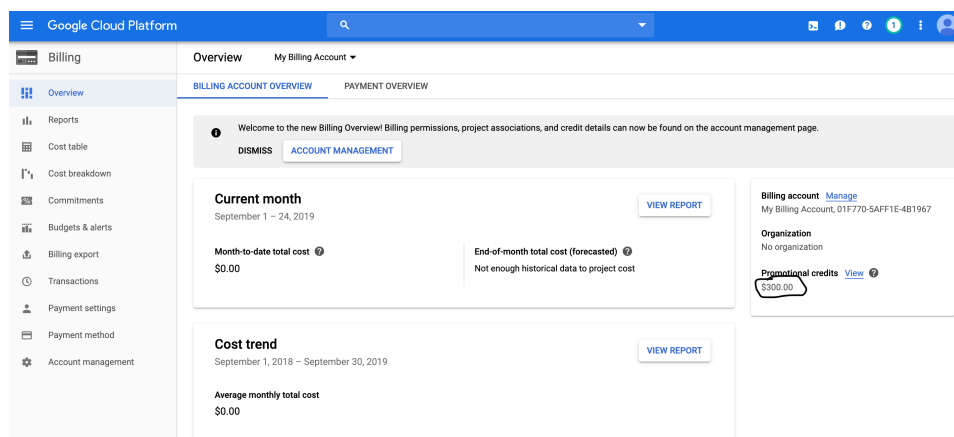
No autocharge after free trial ends

We ask you for your credit card to make sure you are not a robot. You won't be charged unless you manually upgrade to a paid account.

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 later 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
```

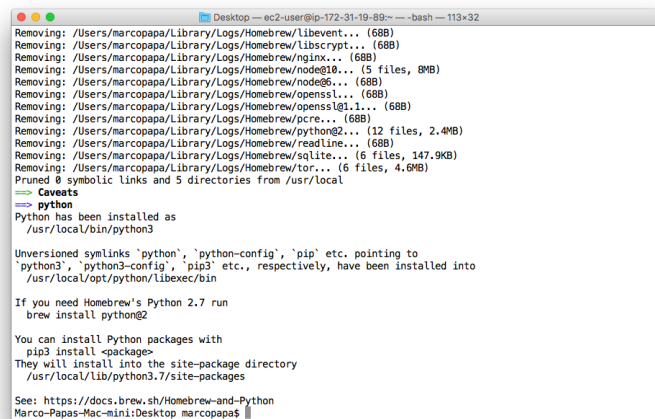
To avoid interfering with macOS, we recommend creating a separate development environment and installing the latest version of Python (version 3.7). To install Python, use **Homebrew**, available at:

<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.

A terminal window screenshot showing the output of the 'brew install python' command. The output lists various dependencies being removed, such as Homebrew's own components and other packages like node, openssl, and readline. It then shows the installation of Python 3.7.6, including the creation of symbolic links for 'python' and 'pip' to the new installation path. The terminal also displays instructions on how to use Homebrew's Python 2.7 and how to install Python packages with pip3.

```
Desktop — ec2-user@ip-172-31-19-89:~ — bash — 113x32
Removing: /Users/marcopapa/Library/Logs/Homebrew/Libevent... (68B)
Removing: /Users/marcopapa/Library/Logs/Homebrew/Libcrypt... (68B)
Removing: /Users/marcopapa/Library/Logs/Homebrew/nginx... (68B)
Removing: /Users/marcopapa/Library/Logs/Homebrew/node@10... (5 files, 80B)
Removing: /Users/marcopapa/Library/Logs/Homebrew/node@6... (68B)
Removing: /Users/marcopapa/Library/Logs/Homebrew/openssl... (68B)
Removing: /Users/marcopapa/Library/Logs/Homebrew/openssl@1... (68B)
Removing: /Users/marcopapa/Library/Logs/Homebrew/pcre... (68B)
Removing: /Users/marcopapa/Library/Logs/Homebrew/python@2... (12 files, 2.4MB)
Removing: /Users/marcopapa/Library/Logs/Homebrew/readline... (68B)
Removing: /Users/marcopapa/Library/Logs/Homebrew/sqlite... (6 files, 147.9KB)
Removing: /Users/marcopapa/Library/Logs/Homebrew/tor... (6 files, 4.6MB)
Pruned 0 symbolic links and 5 directories from /usr/local
==> Caveats
==> python
Python has been installed as
  /usr/local/bin/python3

Unversioned symlinks 'python', 'python-config', 'pip' etc. pointing to
python3, 'python3-config', 'pip3' etc., respectively, have been installed into
  /usr/local/opt/python/libexec/bin

If you need Homebrew's Python 2.7 run
  brew install python@2

You can install Python packages with
  pip3 install <package>
They will install into the site-package directory
  /usr/local/lib/python3.7/site-packages

See: https://docs.brew.sh/Homebrew-and-Python
Marco-Papas-Mac-mini:Desktop marcopapa$
```

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:

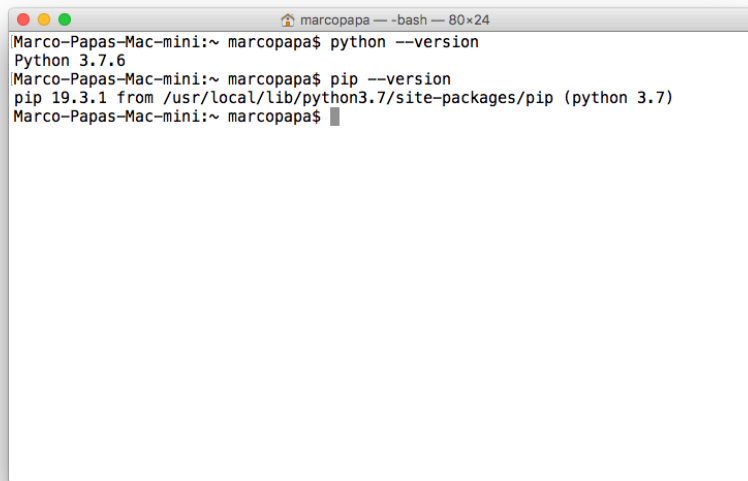
```
echo "alias python=/usr/local/bin/python3.7" >> ~/.bash_profile
echo "alias pip=/usr/local/bin/pip3" >> ~/.bash_profile.
```

You will have to re-start Terminal, so the aliases will take effect. Start a 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
```

A terminal window titled 'marcopapa - bash - 80x24' showing the output of 'python --version' and 'pip --version' commands. The output for 'python --version' is 'Python 3.7.6' and for 'pip --version' is 'pip 19.3.1 from /usr/local/lib/python3.7/site-packages/pip (python 3.7)'.

```
Marco-Papas-Mac-mini:~ marcopapa$ python --version
Python 3.7.6
Marco-Papas-Mac-mini:~ marcopapa$ pip --version
pip 19.3.1 from /usr/local/lib/python3.7/site-packages/pip (python 3.7)
Marco-Papas-Mac-mini:~ marcopapa$
```

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 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 “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/>

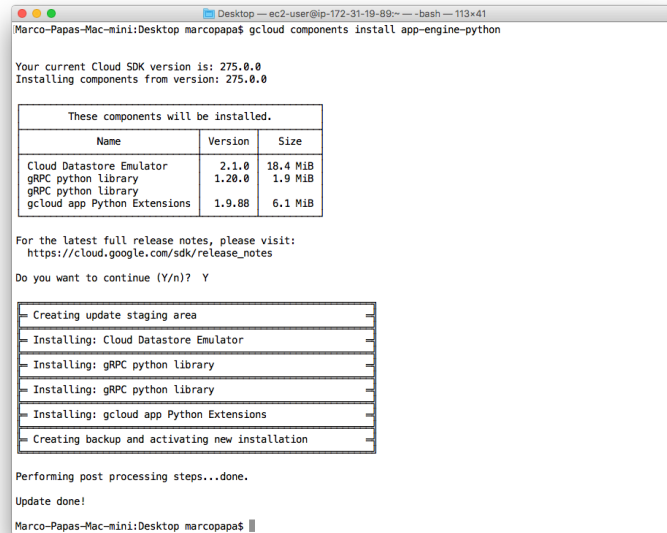
While in most of the GCP docs it is documented that `gcloud` needs to run on Python 2, according to the Note in the above site “As of Cloud SDK version 274.0.0, the `gcloud` CLI has GA support for running using a **Python 3.5 and up** interpreter (run `gcloud topic startup` for exclusions and more information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

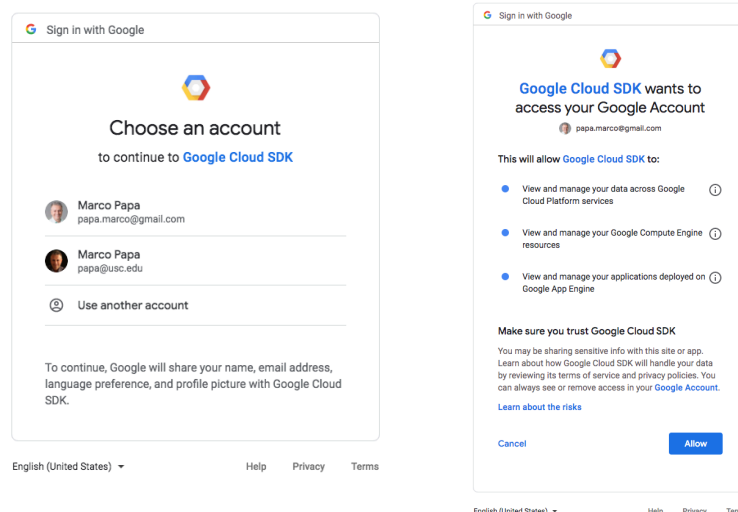
```
gcloud components install app-engine-python
```



6. Login to GCP with the CLI:

```
gcloud auth login
```

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

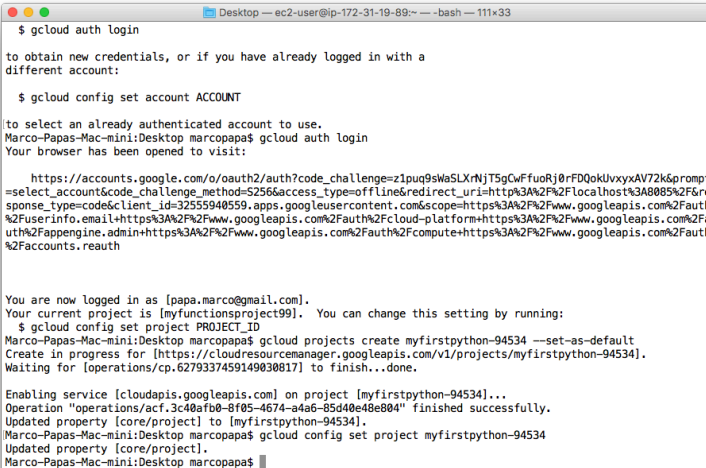


7. Create a new project using this command syntax:

```
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
```

A terminal window titled "Desktop — ec2-user@ip-172-31-19-89:~ — bash — 111x33" showing the execution of gcloud commands. The user runs "gcloud auth login", which opens a browser for authentication. After logging in, the user runs "gcloud config set account ACCOUNT". Then, they run "gcloud projects create myfirstpython-94534 --set-as-default". The terminal output shows the project creation progress, including enabling the service "cloudapis.googleapis.com" on the project "myfirstpython-94534". The final output is "Marco-Papas-Mac-mini:Desktop marcopapa\$".

```
$ gcloud auth login
to obtain new credentials, or if you have already logged in with a
different account:

$ gcloud config set account ACCOUNT

to select an already authenticated account to use.
Marco-Papas-Mac-mini:Desktop marcopapa$ gcloud auth login
Your browser has been opened to visit:

https://accounts.google.com/o/oauth2/auth?code_challenge=z1puq9sWaSLXrNjT5gCwFfuorJ0rFDQokUvxyxAV72k6&prompt
=select_account&code_challenge_method=S256&access_type=offline&redirect_uri=http%3A%2F%2Flocalhost%3A8085%2F&re
sponse_type=code&client_id=32555940559.apps.googleusercontent.com&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth
%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fp
auth%2Fappengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth
%2Faccounts.reauth

You are now logged in as [papa.marco@gmail.com].
Your current project is [myfunctionsproject99]. You can change this setting by running:
$ gcloud config set project PROJECT_ID
Marco-Papas-Mac-mini:Desktop marcopapa$ gcloud projects create myfirstpython-94534 --set-as-default
Create in progress for [https://cloudresourcemanager.googleapis.com/v1/projects/myfirstpython-94534].
Waiting for [operations/cp.6279337459149830817] to finish...done.

Enabling service [cloudapis.googleapis.com] on project [myfirstpython-94534]...
Operation "operations/acf.3c40afb0-8f05-4674-44a6-85d40e48e804" finished successfully.
Updated property [core/project] to [myfirstpython-94534].
Marco-Papas-Mac-mini:Desktop marcopapa$ gcloud config set project myfirstpython-94534
Updated property [core/project].
Marco-Papas-Mac-mini:Desktop marcopapa$
```

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
```

```
Desktop — ec2-user@ip-172-31-19-89:~ — bash — 110x30
Marco-Papas-Mac-mini:Desktop marcopapa$ gcloud app create --project=myfirstpython-94534
You are creating an app for project [myfirstpython-94534].
WARNING: Creating an App Engine application for a project is irreversible and the region
cannot be changed. More information about regions is at
<https://cloud.google.com/appengine/docs/locations>.

Please choose the region where you want your App Engine application
located:

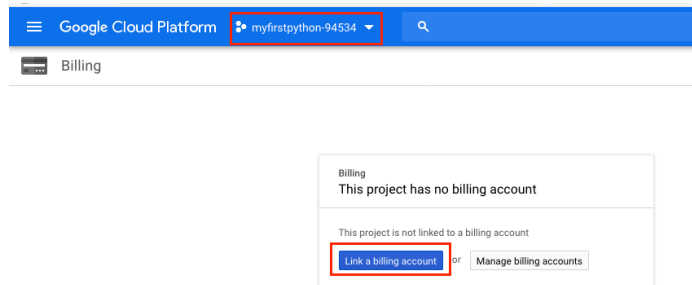
[1] asia-east2 (supports standard and flexible)
[2] asia-northeast1 (supports standard and flexible)
[3] asia-northeast2 (supports standard and flexible)
[4] asia-south1 (supports standard and flexible)
[5] australia-southeast1 (supports standard and flexible)
[6] europe-west (supports standard and flexible)
[7] europe-west2 (supports standard and flexible)
[8] europe-west3 (supports standard and flexible)
[9] europe-west6 (supports standard and flexible)
[10] northamerica-northeast1 (supports standard and flexible)
[11] southamerica-east1 (supports standard and flexible)
[12] us-central (supports standard and flexible)
[13] us-east1 (supports standard and flexible)
[14] us-east4 (supports standard and flexible)
[15] us-west2 (supports standard and flexible)
[16] cancel
Please enter your numeric choice: 15

Creating App Engine application in project [myfirstpython-94534] and region [us-west2]....done.
Success! The app is now created. Please use `gcloud app deploy` to deploy your first app.
Marco-Papas-Mac-mini:Desktop marcopapa$
```

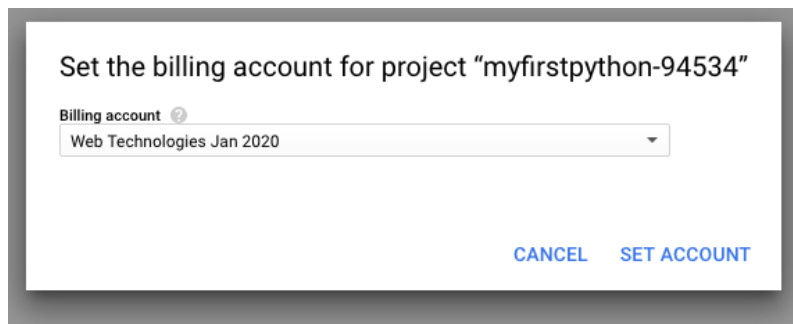
10. **Enable billing** for the project. You will do this in the Cloud console at:

<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.



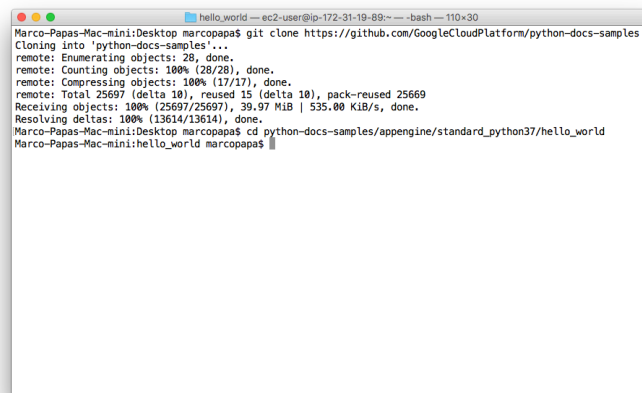
11. Install Git

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:

- Windows ONLY:** download and install **PowerShell** as indicated in the tutorial.
- Create an isolated Python environment

```
python3 -m venv env
```

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

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

- Install dependencies (this step will install Flask):

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

- Run the application:

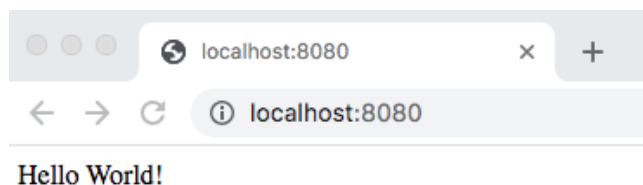
```
python main.py
```

```
hello_world — Python · Python main.py — 84x47
Marco-Papas-Mac-mini:hello_world marcopapa$ ls
app.yaml                                main_test.py
main.py                                requirements.txt
Marco-Papas-Mac-mini:hello_world marcopapa$ python3 -m venv env

Marco-Papas-Mac-mini:hello_world marcopapa$
Marco-Papas-Mac-mini:hello_world marcopapa$ source env/bin/activate
(env) Marco-Papas-Mac-mini:hello_world marcopapa$
(env) Marco-Papas-Mac-mini:hello_world marcopapa$ pip install -r requirements.txt
Collecting Flask==1.1.1
  Downloading https://files.pythonhosted.org/packages/9b/93/628509b8d5dc749656a9641f4caf13540e2cdec85276964ff8f43bbb1d3b/Flask-1.1.1-py2.py3-none-any.whl (94kB)
    |████████████████████████████████████████████████████████████████████████████████| 102kB 6.1MB/s
Collecting Werkzeug>=0.15
  Downloading https://files.pythonhosted.org/packages/ce/42/3aeda98f96e85fd26180534d36570e4d18108d62ae36f87694b476b83d6f/Werkzeug-0.16.0-py2.py3-none-any.whl (327kB)
    |████████████████████████████████████████████████████████████████████████████████| 327kB 4.3MB/s
Collecting itsdangerous>=0.24
  Downloading https://files.pythonhosted.org/packages/76/ae/44b03b253d6fade317f32c24d100b3b35c2239807046a4c953c7b89fa49e/itsdangerous-1.1.0-py2.py3-none-any.whl
Collecting Jinja2>=2.10.1
  Downloading https://files.pythonhosted.org/packages/65/e0/eb35e762802015cab1ccee04e8a277b03f1d8e53da3ec3106882ec42558b/Jinja2-2.10.3-py2.py3-none-any.whl (125kB)
    |████████████████████████████████████████████████████████████████████████████████| 133kB 9.1MB/s
Collecting click>=5.1
  Downloading https://files.pythonhosted.org/packages/fa/37/45185cb5abbc30d7257104c434fe0b07e5a195a6847506c074527aa599ec/Click-7.0-py2.py3-none-any.whl (81kB)
    |████████████████████████████████████████████████████████████████████████████████| 81kB 2.7MB/s
Collecting MarkupSafe>=0.23
  Downloading https://files.pythonhosted.org/packages/ce/c6/f000f1af136ef74e4a95e33785921c73595c5390403f102e9b231b065b7a/MarkupSafe-1.1.1-cp37-cp37m-macosx_10_6_intel.w
hl
Installing collected packages: Werkzeug, itsdangerous, MarkupSafe, Jinja2, click, Fl
ask
Successfully installed Flask-1.1.1 Jinja2-2.10.3 MarkupSafe-1.1.1 Werkzeug-0.16.0 cl
ick-7.0 itsdangerous-1.1.0
(env) Marco-Papas-Mac-mini:hello_world marcopapa$ python main.py
* Serving Flask app "main" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:8080/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 156-225-613
```

e. 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
```

```
hello_world — ec2-user@ip-172-31-19-89 — — bash — 124x55
Marco-Papas-Mac-mini:hello_world marcopapa$ gcloud app deploy
Services to deploy:

descriptor: [/Users/marcopapa/Desktop/python-docs-samples/appengine/standard_python37/hello_world/app.yaml]
source: [/Users/marcopapa/Desktop/python-docs-samples/appengine/standard_python37/hello_world]
target project: [myfirstpython-94534]
target service: [default]
target version: [20200108t113653]
target url: [https://myfirstpython-94534.appspot.com]

Do you want to continue (Y/n)? Y
Beginning deployment of service [default]...
Created .gcloudignore file. See 'gcloud topic gcloudignore' for details.
Uploading 429 files to Google Cloud Storage
File upload done.
Updating service [default]...failed.
ERROR: (gcloud.app.deploy) Error Response: [9] Cloud build 513691ca-bdc2-4ad9-9de8-0b0e6748344b status: FAILURE.
Build error details: Failed to download at least one file. Cannot continue.
Check the build log for errors: https://console.cloud.google.com/gcr/builds/513691ca-bdc2-4ad9-9de8-0b0e6748344b?project=675437181434
Marco-Papas-Mac-mini:hello_world marcopapa$ gcloud app deploy
Services to deploy:

descriptor: [/Users/marcopapa/Desktop/python-docs-samples/appengine/standard_python37/hello_world/app.yaml]
source: [/Users/marcopapa/Desktop/python-docs-samples/appengine/standard_python37/hello_world]
target project: [myfirstpython-94534]
target service: [default]
target version: [20200108t114129]
target url: [https://myfirstpython-94534.appspot.com]

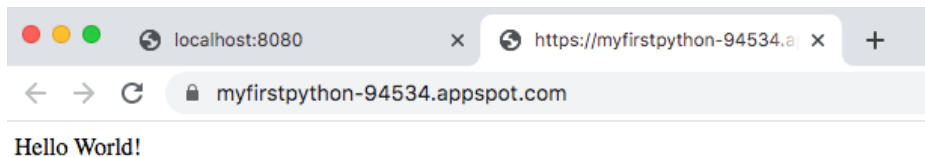
Do you want to continue (Y/n)? Y
Beginning deployment of service [default]...
Uploading 0 files to Google Cloud Storage
File upload done.
Updating service [default]...done.
Setting traffic split for service [default]...done.
Deployed service [default] to [https://myfirstpython-94534.appspot.com]

You can stream logs from the command line by running:
$ gcloud app logs tail -s default

To view your application in the web browser run:
$ gcloud app browse
Marco-Papas-Mac-mini:hello_world marcopapa$ gcloud app browse
Opening [https://myfirstpython-94534.appspot.com] in a new tab in your default browser.
Marco-Papas-Mac-mini:hello_world marcopapa$
```

16. View your application in the cloud. Launch your browser with the app at [http://\[YOUR_PROJECT_ID\].appspot.com](http://[YOUR_PROJECT_ID].appspot.com), running the command:

`gcloud app browse`



Or type the URL in the browser:

`https://myfirstpython-94534.appspot.com/`

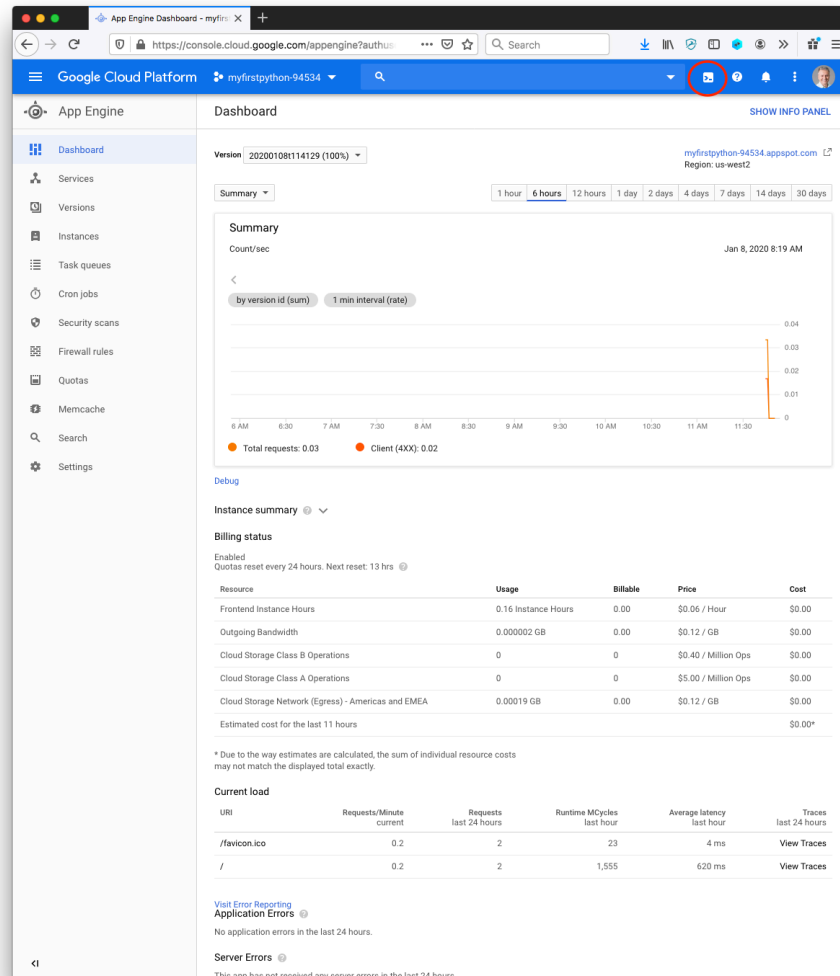
17. Clean up. First stop using the virtual environment. Type this to the (env) prompt:

`deactivate`

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

4. Check App Engine Dashboard

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

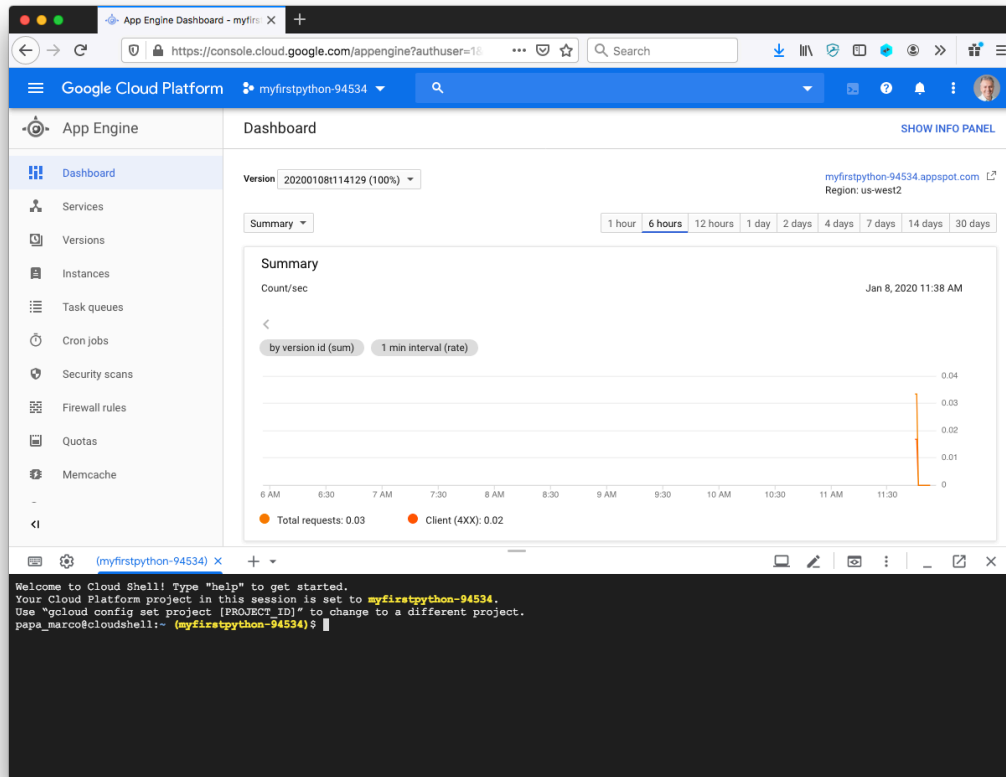


5. Set up Exploring Your instance (Optional)

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.

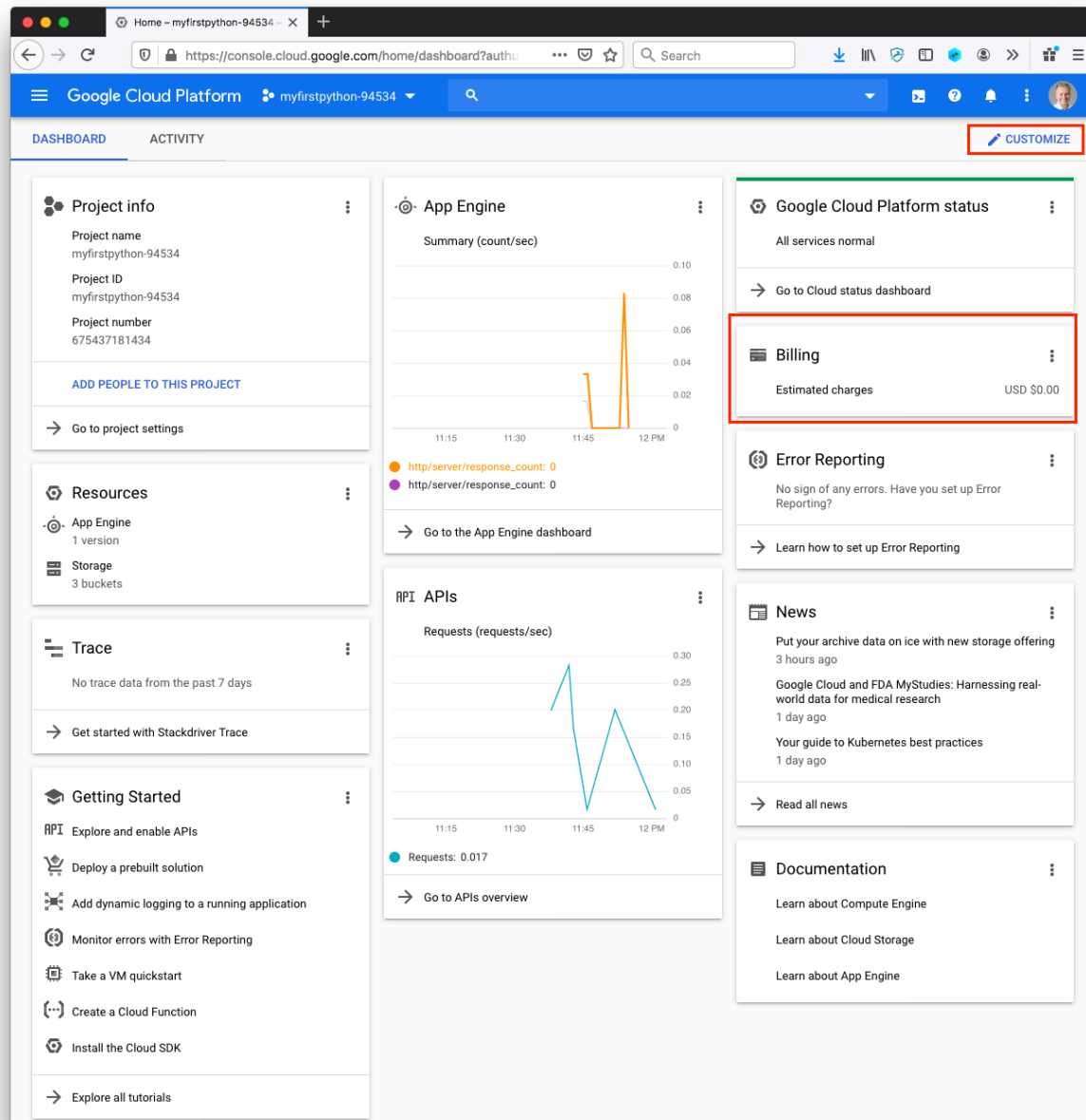


You can read more about the **Google Cloud Shell** here:

<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. You will likely see \$0.00 estimated charges.



Have fun exploring Google Cloud Platform!!