

Homework - 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, you can establish a service using Google App Engine. Once established, you will be able to move your Python program developed for Assignment #2 to your Google App Engine instance and have it executed there.

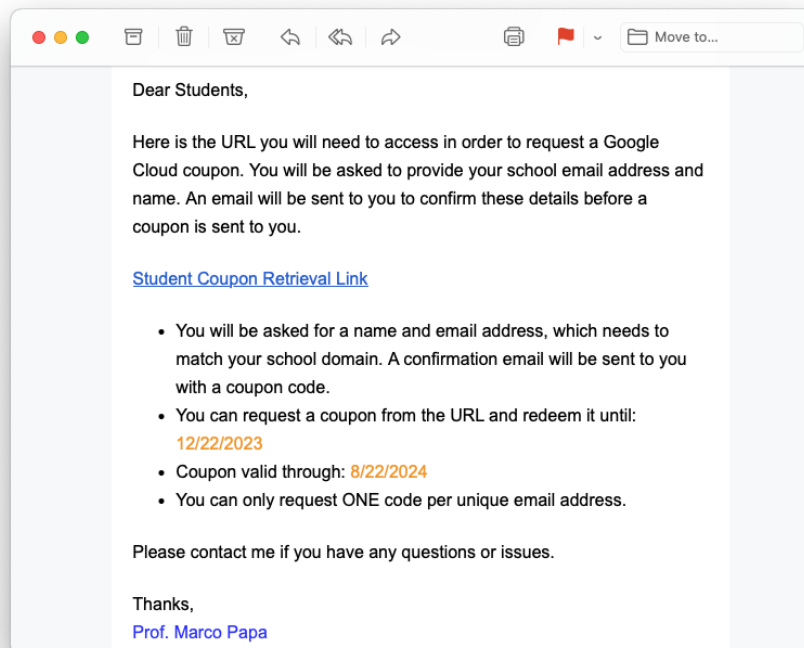
1. Sign up for Google Cloud Platform

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:



The image shows a web browser window with the URL <https://google.secure.force.com/GCPEDU/?cid=IAoM0...>. The page header features the Google Cloud Platform logo and the text "Cloud Platform Education Grants". Below this, a blue banner contains the text: "Use credits provided to you via the Google Cloud Platform Education Grants program to access Google Cloud Platform. Get what you need to build and run your apps, websites and services." The main content area is a white box with 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 includes three input fields: "First Name", "Last Name", and "School Email". The "School Email" field has a dropdown menu showing "@usc.edu". Below the form, there is a "Submit" button. A small text block below the button states: "By clicking 'Submit' below, you agree that we may share the following information with your educational institution and course instructor (papa@usc.edu): (1) personal information that you provide to us on this form and (2) information regarding your use of the coupon and Google Cloud Platform products." At the bottom of the page, there is a link to the "Privacy Policy".

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!

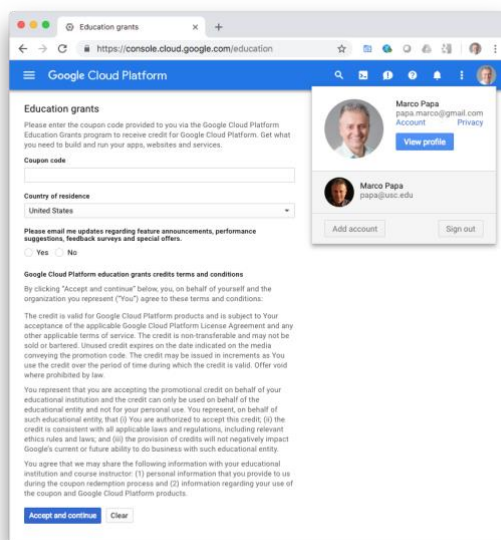
Important step: Before clicking on the link labeled **[here]**, you should open your default browser, and **login** to a **Gmail** account. USC no longer provides @gmail.com accounts to students. If you already have a @gmail.com account, you can use it. If not, you can create one by going here:

<https://support.google.com/mail/answer/56256?hl=en>

Once logged into Gmail, you can click on **[here]** in the e-mail, or you can go to this page:

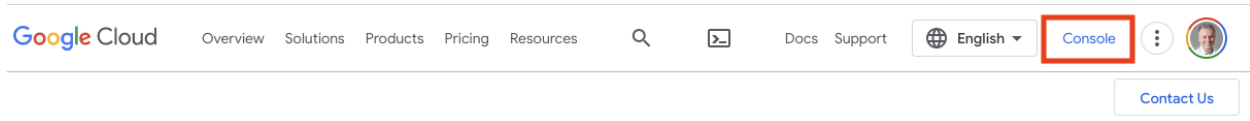
<https://console.cloud.google.com/education>

and paste the Coupon Code, to redeem your coupon. The web form below will be displayed.

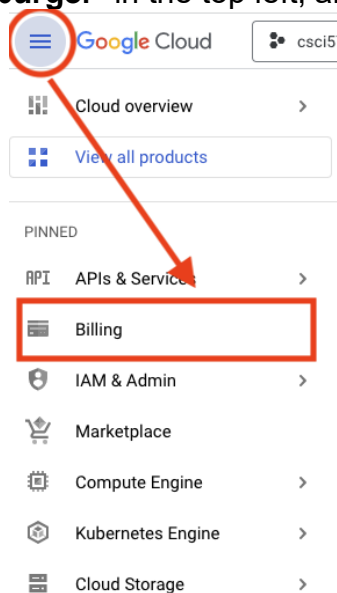
A screenshot of a web browser displaying the Google Cloud Platform Education grants redemption page. The page has a blue header with the Google Cloud Platform logo. The main content area is white and contains a form for redeeming a coupon code. The form includes a text input field for the 'Coupon code', a dropdown menu for 'Country of residence' (currently set to 'United States'), and a checkbox for 'Please email me updates regarding feature announcements, performance suggestions, feedback surveys and special offers.' (currently checked). Below the form is a section titled 'Google Cloud Platform education grants credits terms and conditions' with a scrollable text area. At the bottom of the form are two buttons: 'Accept and continue' and 'Clear'. On the right side of the page, there is a user profile section for 'Marco Papa' with a profile picture, email address 'papa.marco@gmail.com', and a 'View profile' button. Below this is a 'Sign out' button.

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.

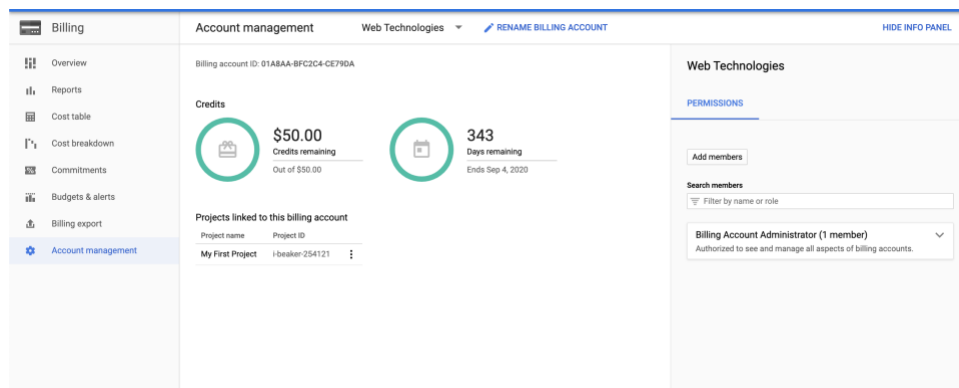
If you see a “**Console**” link next to your picture / avatar, click on it.



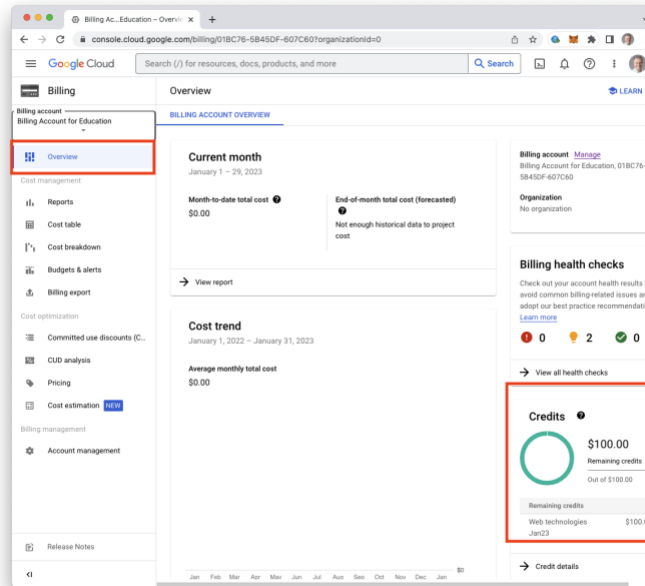
On the next page, click on the “**burger**” in the top left, and then select “**Billing**”.



You can navigate to the **Billing** section and navigate to **Account Management** to see the amount of your credit, as shown below.



If the **Account Management** does not show your credits, select Overview. Your **\$25 credits** should show on the Credits section.



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

If you were successful in signing up and obtaining the \$25 CGP Education Credit, skip to section **1.3. How to get additional student Coupons**.

1.2 Sign up for Google Cloud Platform Free Trial

IMPORTANT: you should follow the steps in this section, only if 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**.

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.

☒ 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.

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 Google 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

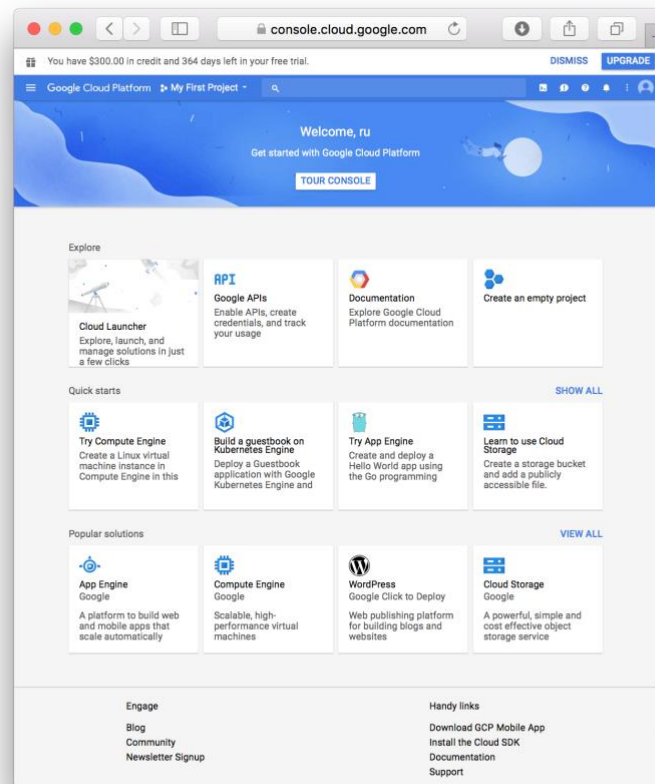
Visa

Credit or debit card address is same as above

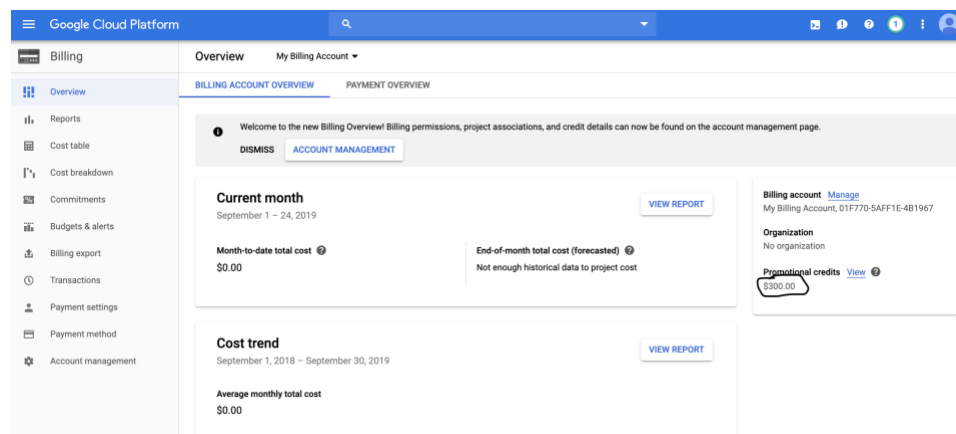
Start my free trial

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 each 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/docs/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.

Note: As of 6/9/2023, App Engine on Google Cloud is compatible with Python 3.7, 3.8, 3.9, 3.10 and 3.11. Quoting from:

<https://cloud.google.com/appengine/docs/standard/python3/runtime>

“The Python 3 runtime supports Python 3.7, Python 3.8, Python 3.9, Python 3.10, and Python 3.11 and uses the latest stable release of the version that is specified in your `app.yaml` file.”

Note: Support for Python 3.5-3.7 will be deprecated on August 8th, 2023.

Installing on macOS

Older versions of 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
```

Newer version of macOS, no longer include Python 2.7. In April 2022, Apple removed support for Python 2.7 on macOS devices running **Monterey 12.3** and above.

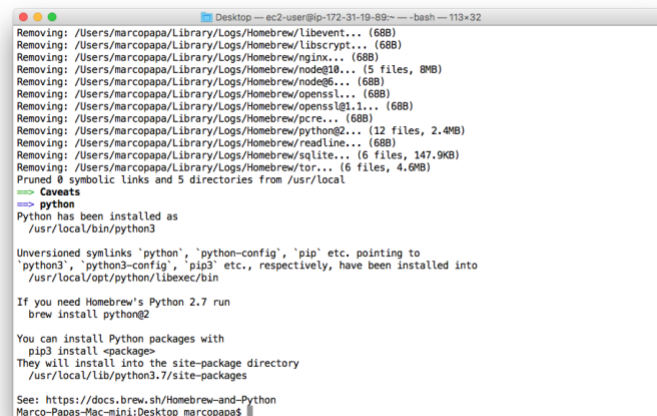
To avoid interfering with macOS, we recommend creating a separate development environment and installing the latest version of Python (version 3.8 or later). To install Python, use **Homebrew**, available at:

<https://brew.sh/>

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

```
brew install python
```

As of this writing, Homebrew will install **Python 3.10**. 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 libevent, libcrypto, nginx, nodejs, openssl, perl, postgresql, readline, sqlite, and tor. It then shows the installation of Python 3.10, creating symbolic links for 'python', 'python-config', and 'pip' to the new installation path. The terminal also provides instructions for installing Python packages with pip3 and a link to the Homebrew documentation for Python.

```
Removing: /Users/marcopapa/Library/Logs/Homebrew/libevent... (68B)
Removing: /Users/marcopapa/Library/Logs/Homebrew/libcrypto... (68B)
Removing: /Users/marcopapa/Library/Logs/Homebrew/nginx... (68B)
Removing: /Users/marcopapa/Library/Logs/Homebrew/nodejs... (5 files, 8MB)
Removing: /Users/marcopapa/Library/Logs/Homebrew/nodejs... (68B)
Removing: /Users/marcopapa/Library/Logs/Homebrew/openssl... (68B)
Removing: /Users/marcopapa/Library/Logs/Homebrew/openssl1.1... (68B)
Removing: /Users/marcopapa/Library/Logs/Homebrew/postgres... (68B)
Removing: /Users/marcopapa/Library/Logs/Homebrew/postgresql... (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 8 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 marcopapas
```

Normally Python 3 will be installed in `/usr/local/bin/python3` (a symbolic link to another folder). 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/python3.9" >> ~/.zshrc
```

```
echo "alias python=/usr/local/bin/python3.9" >> ~/.bashrc
echo "alias pip=/usr/local/bin/pip3" >> ~/.zshrc
echo "alias pip=/usr/local/bin/pip3" >> ~/.bashrc
```

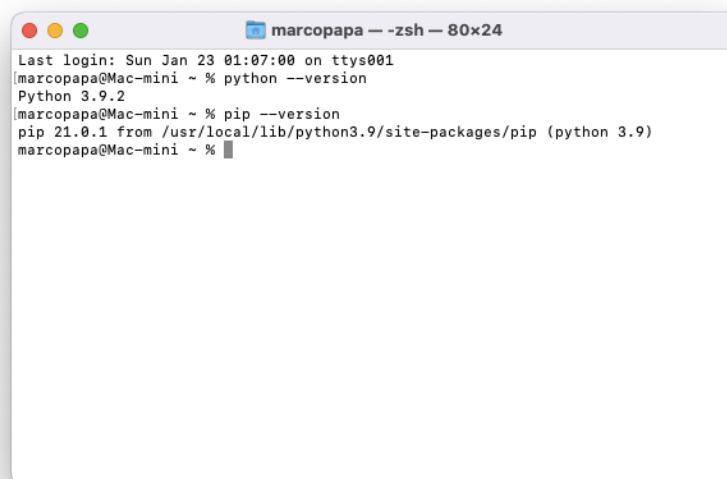
If you are using bash and ~/.bash_profile, you may have to do this instead:

```
echo "alias python=/usr/local/bin/python3.9" >> ~/.bash_profile
echo "alias pip=/usr/local/bin/pip3" >> ~/.bash_profile.
```

Replace python3.9 in the script above with your installed version, such as python3.11.

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 screenshot of a macOS Terminal window titled "marcopapa -- zsh - 80x24". The window shows the following text: "Last login: Sun Jan 23 01:07:00 on ttys001", "marcopapa@Mac-mini ~ % python --version", "Python 3.9.2", "marcopapa@Mac-mini ~ % pip --version", "pip 21.0.1 from /usr/local/lib/python3.9/site-packages/pip (python 3.9)", and "marcopapa@Mac-mini ~ %". The cursor is at the end of the last line.

```
Last login: Sun Jan 23 01:07:00 on ttys001
marcopapa@Mac-mini ~ % python --version
Python 3.9.2
marcopapa@Mac-mini ~ % pip --version
pip 21.0.1 from /usr/local/lib/python3.9/site-packages/pip (python 3.9)
marcopapa@Mac-mini ~ %
```

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**, **Visual Studio Code**, 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.

Another good editor is **Visual Studio Code** by Microsoft. It can be downloaded here:

<https://code.visualstudio.com/>

Visual Studio Code 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** “latest” version for your platform (Mac OS, Windows) from:

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

4. Under “Guides” click on **Installing the gcloud CLI**.

5. Select the package for your OS platform and extract the archive file on your local file system.
6. (Optional) Add the Cloud SDK tools to your PATH. Run the script (from the root of the folder you extracted in the last step) using this command:

```
./google-cloud-sdk/install.sh
```

```
marcopapa -- zsh — 121x98
(env) marcopapa@Mac-mini-2 ~ % ./google-cloud-sdk/install.sh
Welcome to the Google Cloud CLI!

To help improve the quality of this product, we collect anonymized usage data
and anonymized stacktraces when crashes are encountered; additional information
is available at <https://cloud.google.com/sdk/usage-statistics>. This data is
handled in accordance with our privacy policy
<https://cloud.google.com/terms/cloud-privacy-notice>. You may choose to opt in this
collection now (by choosing 'Y' at the below prompt), or at any time in the
future by running the following command:

    gcloud config set disable_usage_reporting false

Do you want to help improve the Google Cloud CLI (y/N)? N

Your current Google Cloud CLI version is: 434.0.0
The latest available version is: 434.0.0
```

Components			
Status	Name	ID	Size
Not Installed	App Engine Go Extensions	app-engine-go	4.4 MiB
Not Installed	Appctl	appctl	18.5 MiB
Not Installed	Artifact Registry Go Module Package Helper	package-go-module	< 1 MiB
Not Installed	Cloud Bigtable Command Line Tool	cbt	11.2 MiB
Not Installed	Cloud Bigtable Emulator	bigtable	6.9 MiB
Not Installed	Cloud Datastore Emulator	cloud-datastore-emulator	35.1 MiB
Not Installed	Cloud Firestore Emulator	cloud-firestore-emulator	42.4 MiB
Not Installed	Cloud Pub/Sub Emulator	pubsub-emulator	62.6 MiB
Not Installed	Cloud Run Proxy	cloud-run-proxy	11.8 MiB
Not Installed	Cloud SQL Proxy	cloud_sql_proxy	7.6 MiB
Not Installed	Google Container Registry's Docker credential helper	docker-credential-gcr	2.2 MiB
Not Installed	Kustomize	kustomize	7.6 MiB
Not Installed	Log Streaming	log-streaming	12.3 MiB
Not Installed	Minikube	minikube	33.2 MiB
Not Installed	Nomos CLI	nomos	25.8 MiB
Not Installed	On-Demand Scanning API extraction helper	local-extract	14.0 MiB
Not Installed	Scaffold	scaffold	24.1 MiB
Not Installed	Terraform Tools	terraform-tools	62.1 MiB
Not Installed	anthos-auth	anthos-auth	28.2 MiB
Not Installed	config-connector	config-connector	57.1 MiB
Not Installed	enterprise-certificate-proxy	enterprise-certificate-proxy	6.7 MiB
Not Installed	gcloud Alpha Commands	alpha	< 1 MiB
Not Installed	gcloud Beta Commands	beta	< 1 MiB
Not Installed	gcloud app Java Extensions	app-engine-java	64.6 MiB
Not Installed	gcloud app PHP Extensions	app-engine-php	21.9 MiB
Not Installed	gcloud app Python Extensions	app-engine-python	8.5 MiB
Not Installed	gcloud app Python Extensions (Extra Libraries)	app-engine-python-extras	27.3 MiB
Not Installed	gke-gcloud-auth-plugin	gke-gcloud-auth-plugin	7.5 MiB
Not Installed	kpt	kpt	14.6 MiB
Not Installed	kubect1	kubect1	< 1 MiB
Not Installed	kubect1-oidc	kubect1-oidc	20.2 MiB
Not Installed	pkg	pkg	
Installed	BigQuery Command Line Tool	bq	1.6 MiB
Installed	Cloud Storage Command Line Tool	gsutil	11.3 MiB
Installed	Google Cloud CLI Core Libraries	core	20.7 MiB
Installed	Google Cloud CRC32C Hash Tool	gcloud-crc32c	1.2 MiB

```
To install or remove components at your current SDK version [434.0.0], run:
$ gcloud components install COMPONENT_ID
$ gcloud components remove COMPONENT_ID

To update your SDK installation to the latest version [434.0.0], run:
$ gcloud components update

To take a quick anonymous survey, run:
$ gcloud survey

Modify profile to update your $PATH and enable shell command completion?

Do you want to continue (Y/n)? Y

The Google Cloud SDK installer will now prompt you to update an rc file to bring the Google Cloud CLIs into your
environment.

Enter a path to an rc file to update, or leave blank to use [/Users/marcopapa/.zshrc]:
Backing up [/Users/marcopapa/.zshrc] to [/Users/marcopapa/.zshrc.backup].
[/Users/marcopapa/.zshrc] has been updated.

==> Start a new shell for the changes to take effect.

Google Cloud CLI works best with Python 3.7 and certain modules.

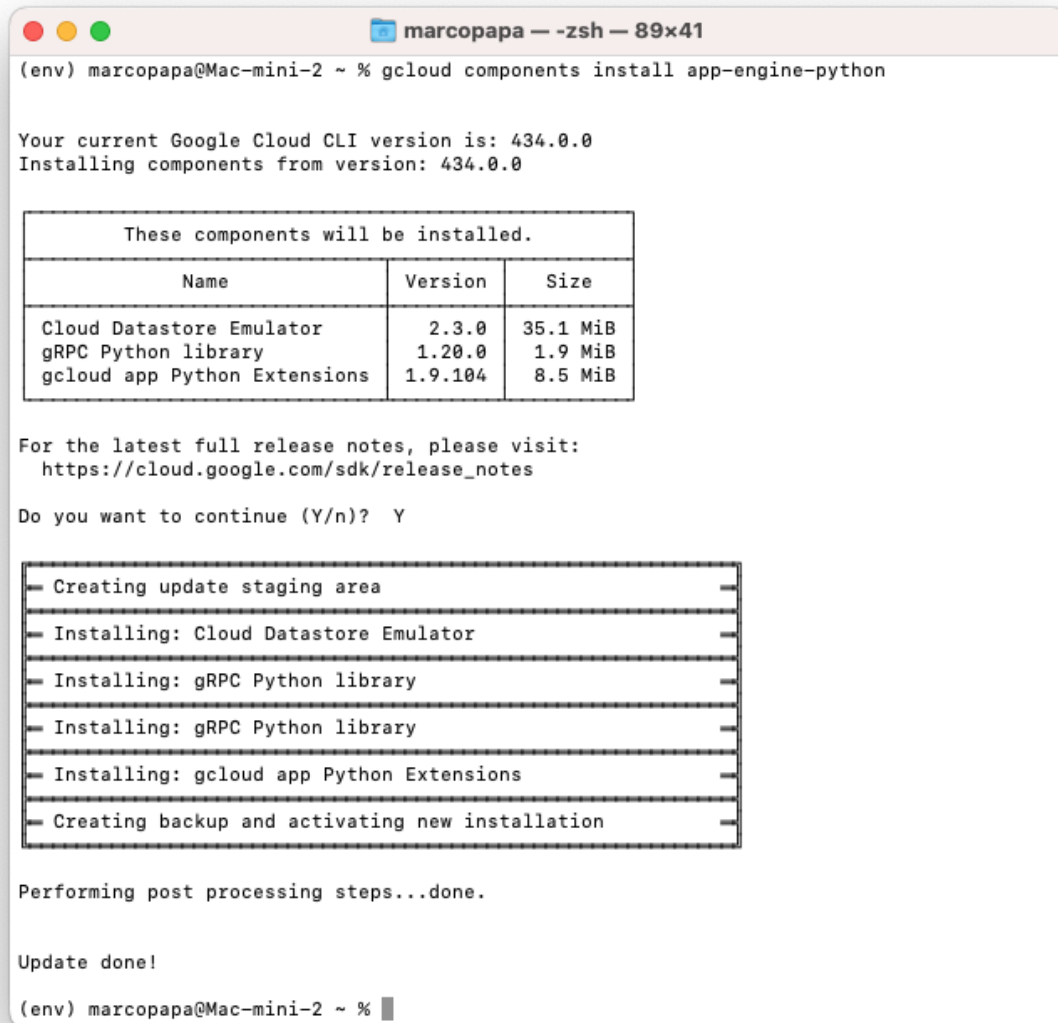
Download and run Python 3.7 installer? (Y/n)? n

For more information on how to get started, please visit:
https://cloud.google.com/sdk/docs/quickstarts

(env) marcopapa@Mac-mini-2 ~ %
```

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

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



A terminal window titled "marcopapa — zsh — 89x41" showing the execution of the command `gcloud components install app-engine-python`. The output indicates the current Google Cloud CLI version is 434.0.0 and that components are being installed from this version. A table lists the components to be installed: Cloud Datastore Emulator (2.3.0, 35.1 MiB), gRPC Python library (1.20.0, 1.9 MiB), and gcloud app Python Extensions (1.9.104, 8.5 MiB). Below the table, a link to the release notes is provided, and the user is prompted to continue (Y/n), answering 'Y'. A progress bar shows the installation steps: creating an update staging area, installing the Cloud Datastore Emulator, gRPC Python library, and gcloud app Python Extensions, and finally creating a backup and activating the new installation. The process concludes with "Performing post processing steps...done." and "Update done!". The prompt returns to the user's shell.

```
(env) marcopapa@Mac-mini-2 ~ % gcloud components install app-engine-python

Your current Google Cloud CLI version is: 434.0.0
Installing components from version: 434.0.0

These components will be installed.


| Name                         | Version | Size     |
|------------------------------|---------|----------|
| Cloud Datastore Emulator     | 2.3.0   | 35.1 MiB |
| gRPC Python library          | 1.20.0  | 1.9 MiB  |
| gcloud app Python Extensions | 1.9.104 | 8.5 MiB  |



For the latest full release notes, please visit:
https://cloud.google.com/sdk/release\_notes

Do you want to continue (Y/n)? Y

- Creating update staging area -
- Installing: Cloud Datastore Emulator -
- Installing: gRPC Python library -
- Installing: gRPC Python library -
- Installing: gcloud app Python Extensions -
- Creating backup and activating new installation -

Performing post processing steps...done.

Update done!

(env) marcopapa@Mac-mini-2 ~ %
```

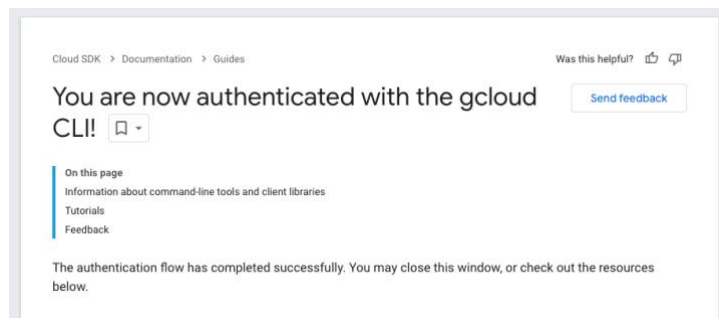
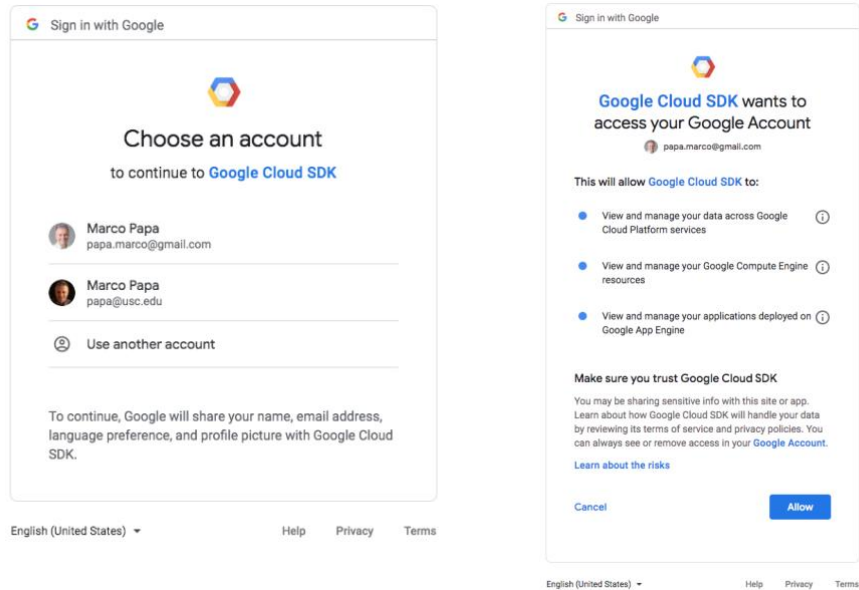
8. Initialize the `gcloud` tool to initialize the SDK:

```
gcloud init
```

You will be asked to “log in (Y/n)?” Answer Y.

You will be asked to “Pick a current project” or “**Create a new project**”. Select “create” and when asked, enter a name as appropriate.

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

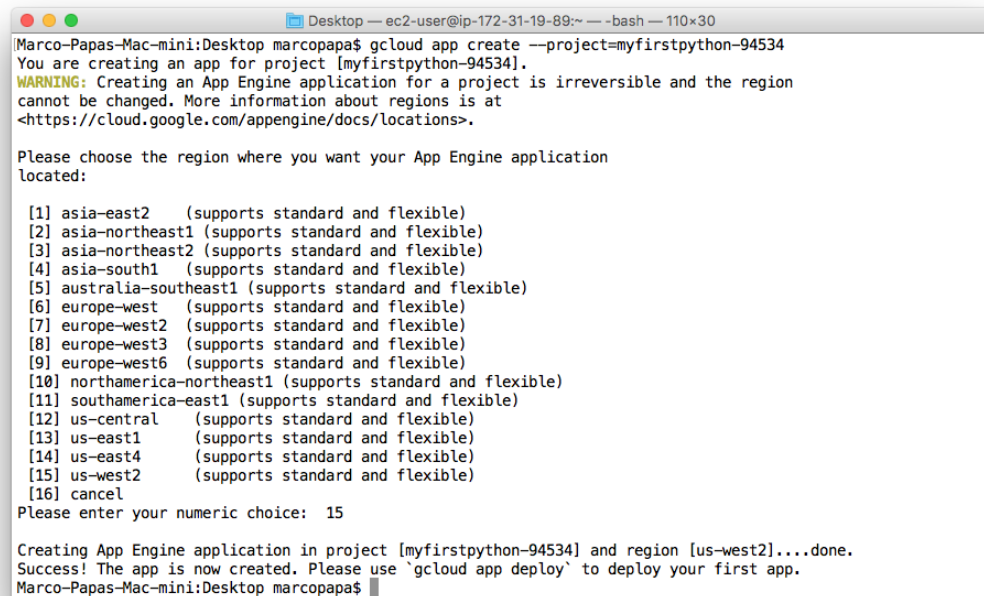


10. Initialize the App Engine app with your newly created project and choose its region (for example **us-west2**):

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

for example:

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



```
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

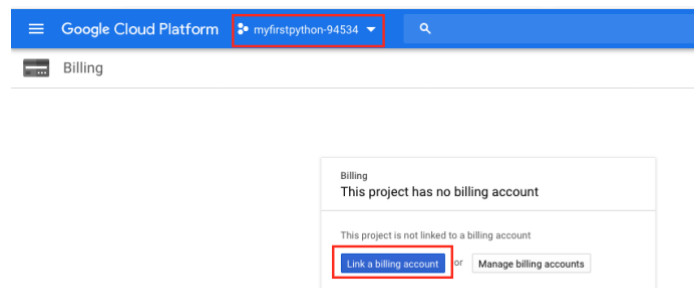
Please enter your numeric choice: 15

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$
```

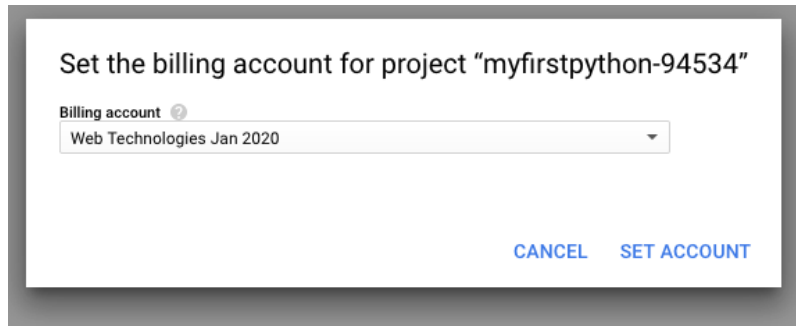
11. **Enable billing** for the project. You will do this in the Cloud console at:

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

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



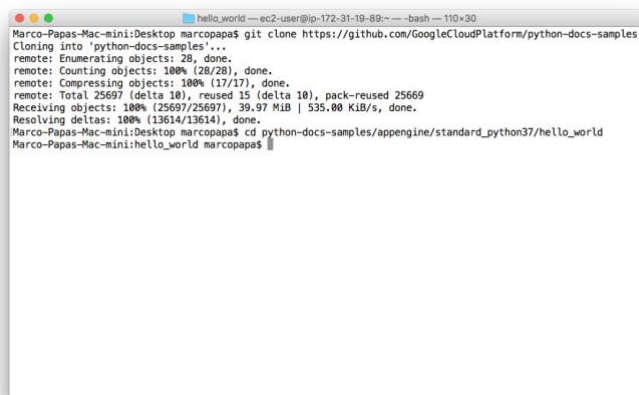
Select the billing account you created with your Google credits.



12. Install Git

13. Clone the **Google Cloud Platform Python Samples** repository from Github:

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



14. Test the **Hello World** app on your local machine:

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

macOS/Linux:

```
pip install virtualenv
virtualenv <your-env>
source <your-env>/bin/activate
```

Windows:

```
pip install virtualenv
virtualenv <your-env>
<your-env>\Scripts\activate
```

- c. Change to the directory that contains the **Hello World** sample code:

```
cd python-docs-  
samples/appengine/standard_python3/hello_world
```

- d. If the file requirements.txt does not exist, create it with:

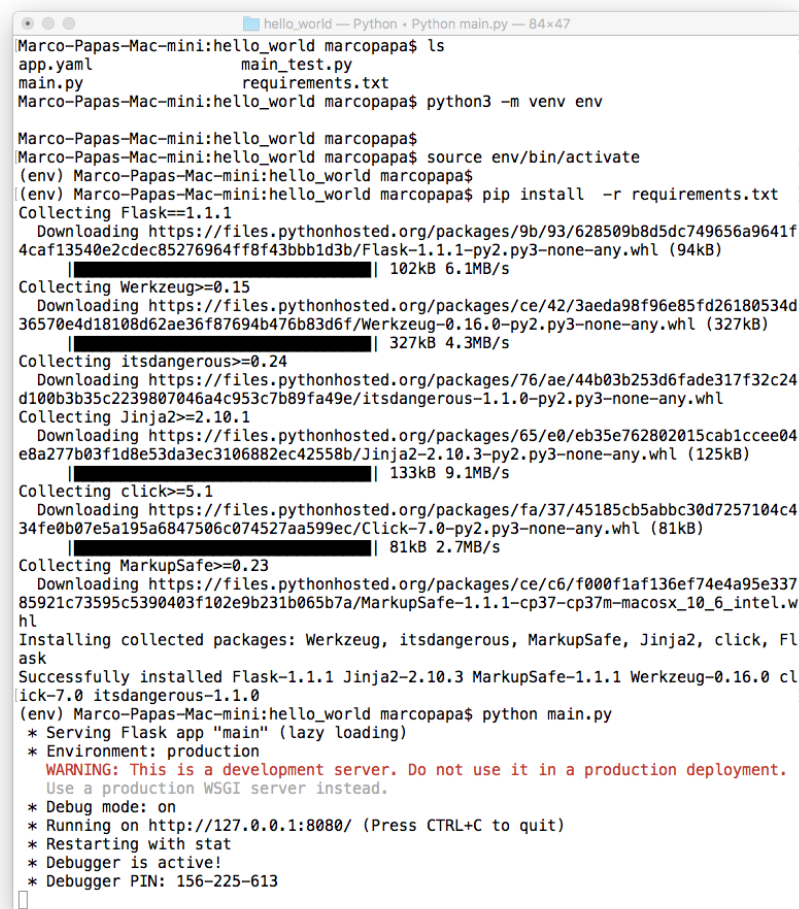
```
pip freeze > requirements.txt
```

- e. Install dependencies (this step will install Flask):

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

- f. Run the application:

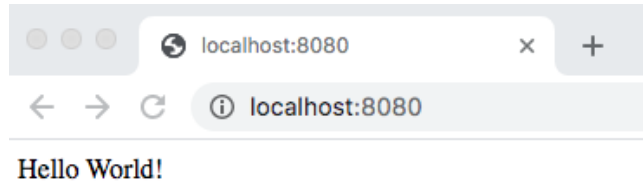
```
python main.py
```



```
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/37/45185cb5abb30d7257104c434fe0b07e5a195a6847506c074527aa599ec/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-cp37m-macosx_10_6_intel.whl  
Installing collected packages: Werkzeug, itsdangerous, MarkupSafe, Jinja2, click, Flask  
Successfully installed Flask-1.1.1 Jinja2-2.10.3 MarkupSafe-1.1.1 Werkzeug-0.16.0 click-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
```

- g. 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
```

A screenshot of a terminal window titled 'hello_world - zsh - 140x30'. The terminal shows the execution of 'gcloud app deploy'. It lists deployment details: descriptor, source, target project (my-python-project-67167), target service (default), target version (20230609t213445), target url (https://my-python-project-67167.wl.r.appspot.com), and target service account. It asks for confirmation to continue (Y/n) and receives 'Y'. It then shows the deployment process: 'Beginning deployment of service [default]...', 'Created .gcloudignore file.', 'Uploading 6 files to Google Cloud Storage', 'File upload done.', 'Updating service [default]...done.', 'Setting traffic split for service [default]...done.', and 'Deployed service [default] to [https://my-python-project-67167.wl.r.appspot.com]'. It provides instructions on how to stream logs and view the application in a web browser, ending with the prompt '(env) marcopapa@Mac-mini-2 hello_world %'.

16. View your application in the cloud. Launch your browser with the app at

[http://\[YOUR_PROJECT_ID\].\[REGION_ID\].r.appspot.com](http://[YOUR_PROJECT_ID].[REGION_ID].r.appspot.com),

running the command:

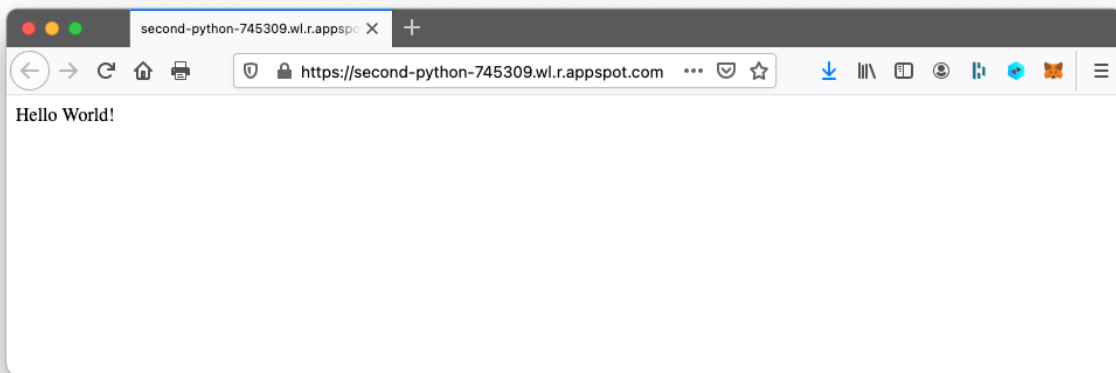
```
gcloud app browse
```

Or type the URL in the browser:

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

or

<https://second-python-745309.wl.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.

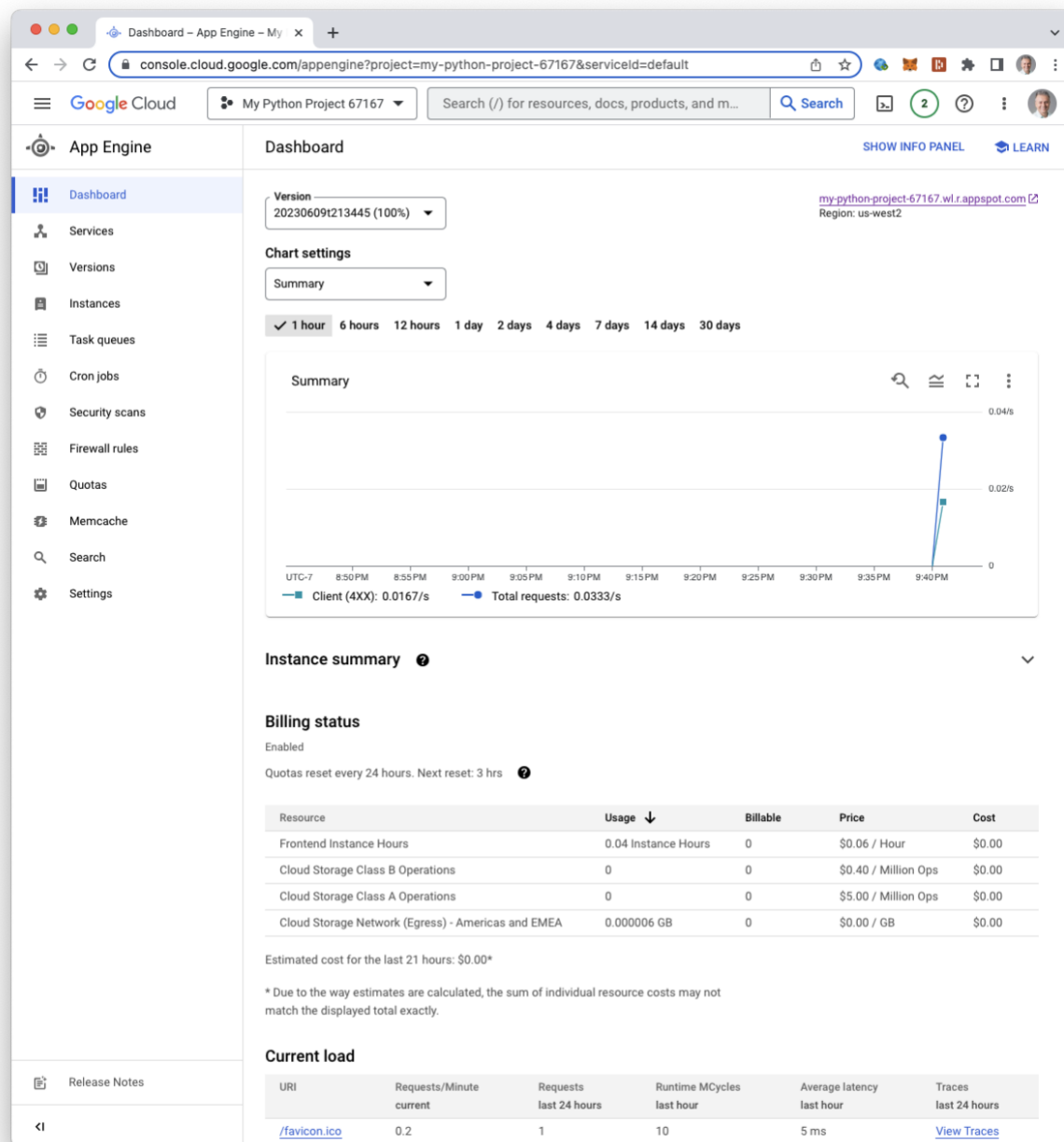
19. The “Hello World” program comes with a file named `requirements.txt`. This file needs to be deployed to GCP. Once you add your code and add some Python libraries, this file needs to be updated. Use pip to install your libraries locally. Then run the following pip command:

```
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.



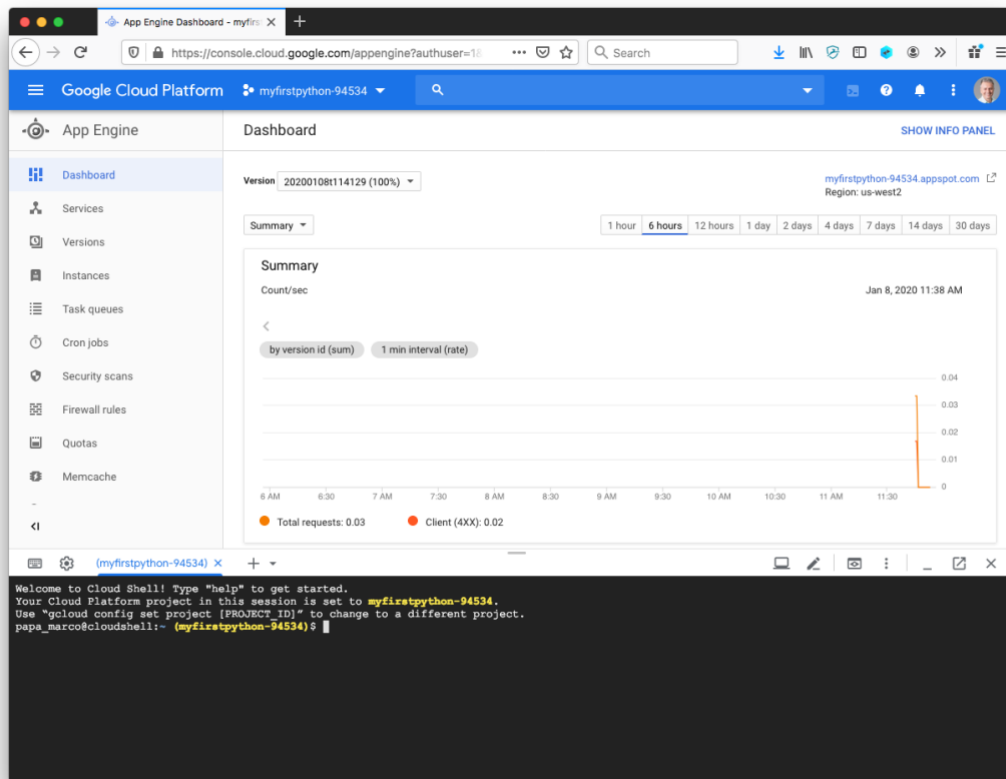
5. Set up Exploring Your instance (Optional)

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 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.

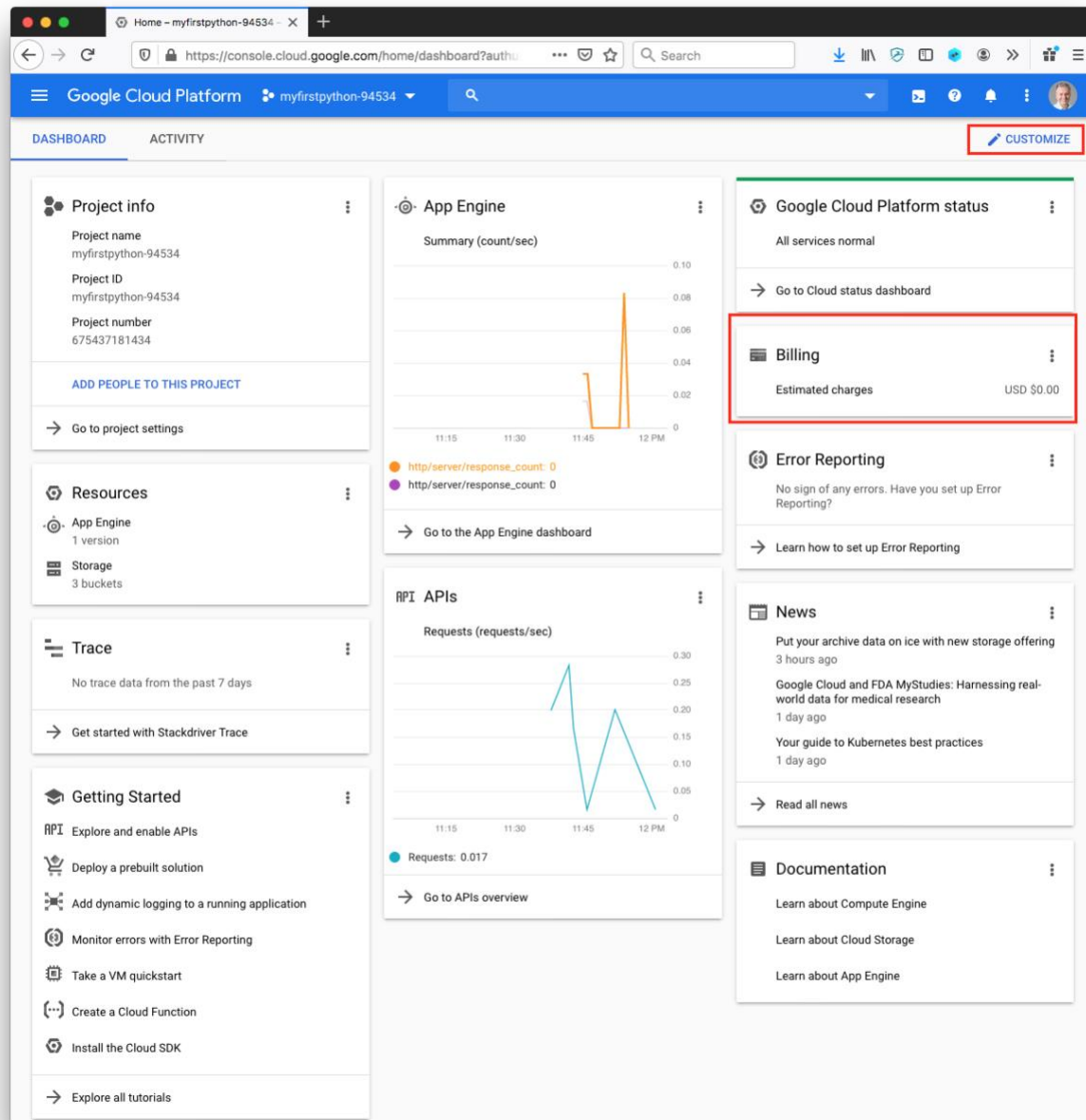


You can read more about the **Google Cloud Shell** here:

<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!!