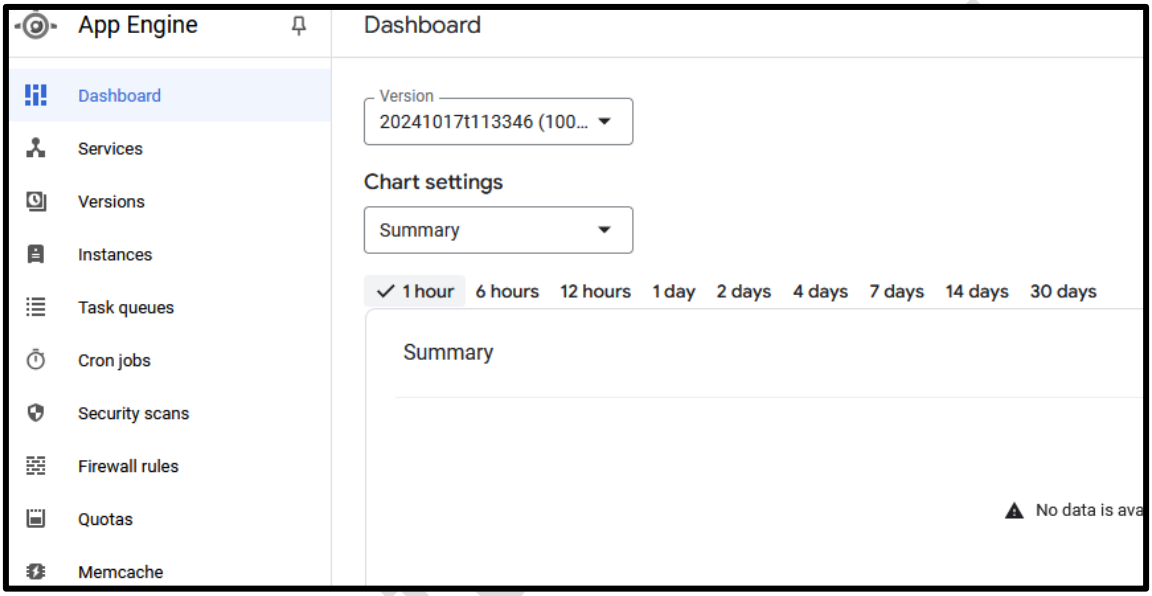
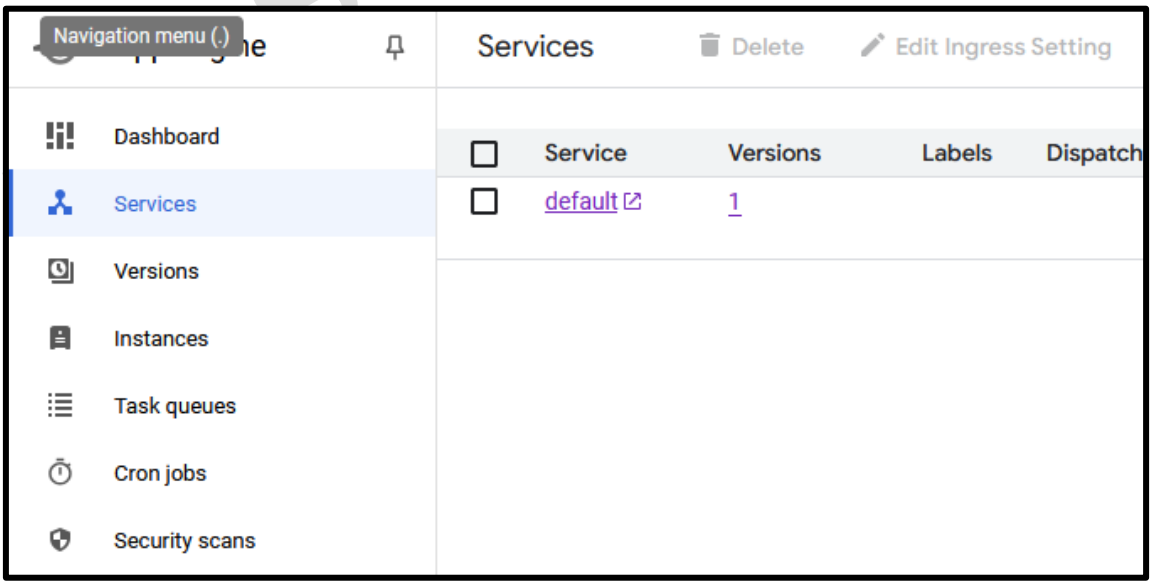
Google Cloud Platform Practices

1. App Engine

- Let's create an example with App Engine.
- First of all, we access App Engine from the console. Since we've created a project in the videos above, something like the following should appear.

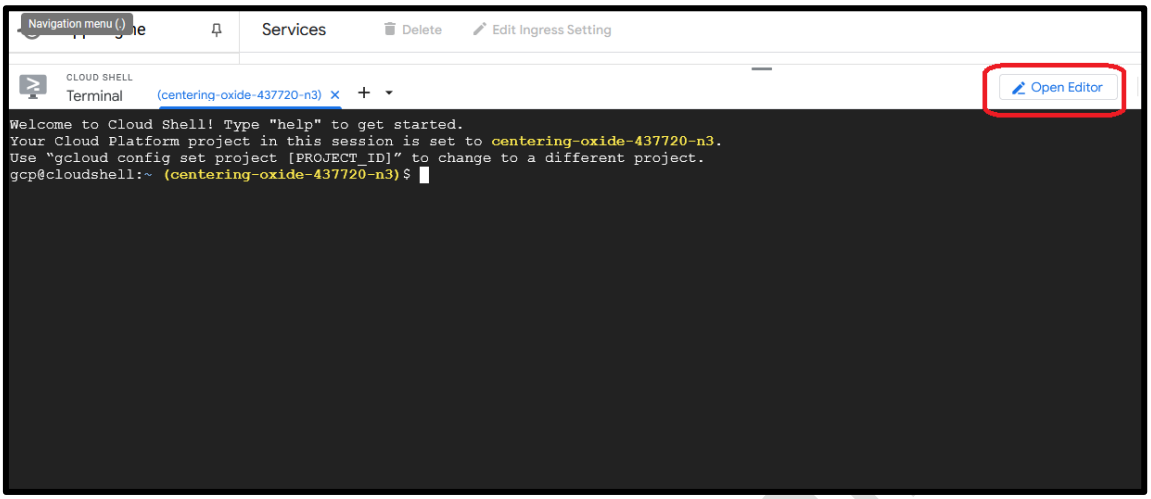


- If we go to "services" something like this should appear. It is the service created in the previous videos.



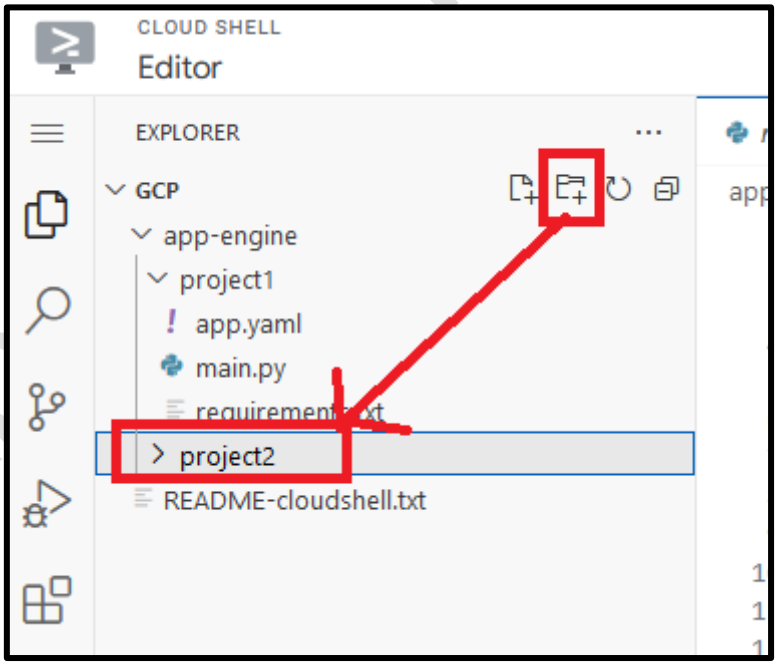
- Let's open a Cloud Shell session to create a new App Engine sample

- Once inside Cloud Shell we are going to open the editor as it will be much easier to work with this environment.



- The first thing we're going to do within the editor is create a folder where we can put our new App engine project.

- Let's call it "project2" for example. We need to make sure that we are within the App Engine directory that we have created throughout the previous videos. Although in reality we can create it wherever we want.

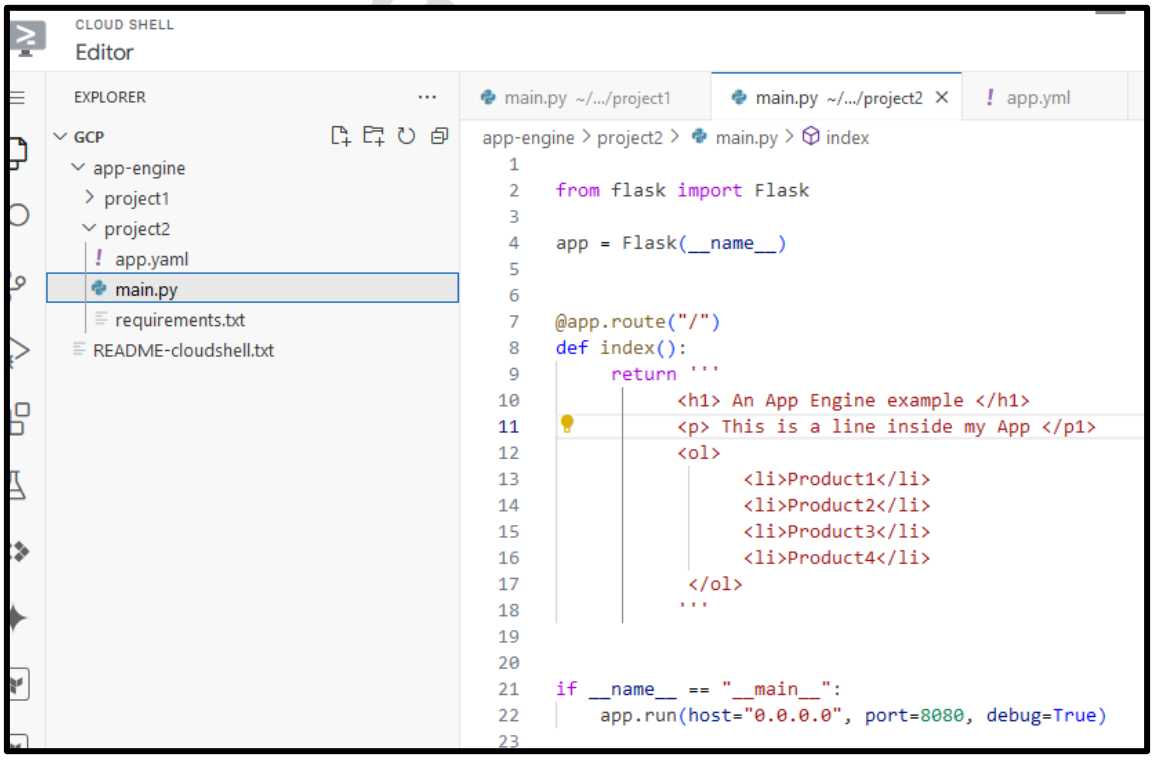


- We are going to create a file called "main.py" with the following content. It's a very simple example than just viewing a Web page.

- We put the following content.

```
from flask import Flask

app = Flask(__name__)


@app.route("/")
def index():
    return '''
        <h1> An App Engine example </h1>
        <p> This is a line inside my App </p1>
        <ol>
            <which>Product1</li>
            <which>Product2</li>
            <which>Product3</li>
            <which>Product4</li>
         </ol>
        '''


if __name__ == "__main__":
    app.run(host="0.0.0.0", port=8080, debug=True)
```

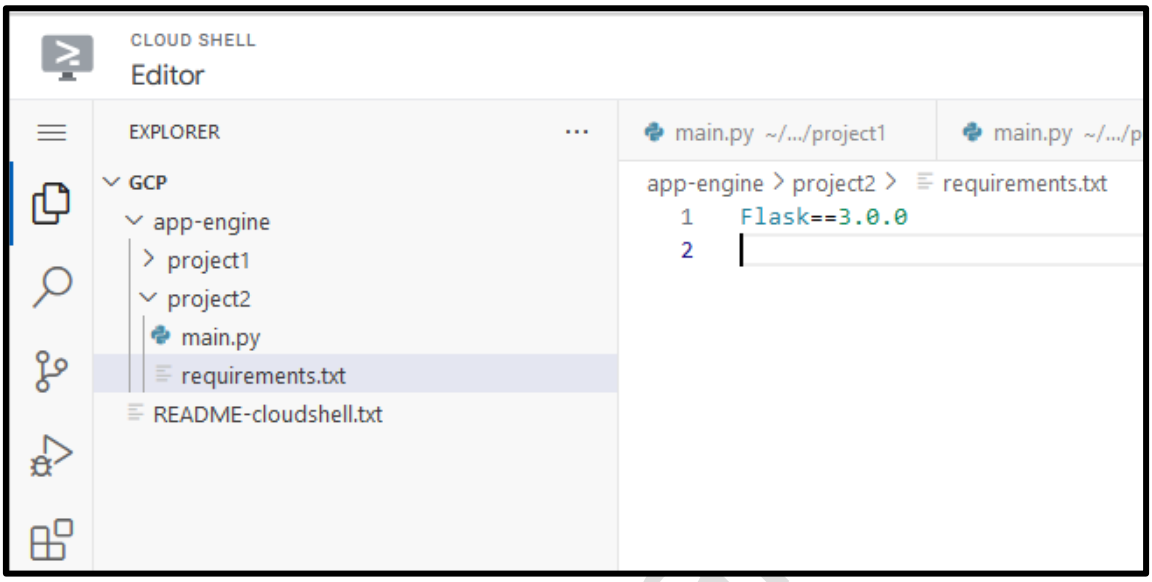
- The environment should be as follows:



- We create a requirements file called "requierements.txt" and put the following content to enable Flask and therefore have a Web service.

```
Flask==3.0.0
```

- It should look something like:



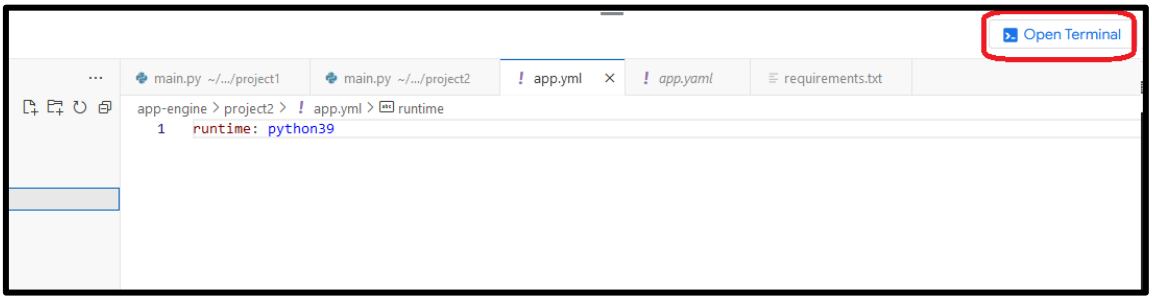
- And finally we create a file called "app.yaml" with which we are going to indicate the language we are going to use and the name of the service to use. In the previous videos we didn't use any and that's why it was labeled as "default".

- We write:

```
service: service2
runtime: python39
```

- It should look like the following

- Now we return to terminal mode



- Access the "project2" directory that we have previously created

```
Welcome to Cloud Shell! Type "help" to get started.
To set your Cloud Platform project in this session use "gc
gcp@cloudshell:~$ cd app-engine/
gcp@cloudshell:~/app-engine$ cd project2
gcp@cloudshell:~/app-engine/project2$ ls -l
total 12
-rw-r--r-- 1 gcp gcp  17 Oct 19 14:32 app.yml
-rw-r--r-- 1 gcp gcp 467 Oct 19 12:52 main.py
-rw-r--r-- 1 gcp gcp  13 Oct 19 12:49 requirements.txt
gcp@cloudshell:~/app-engine/project2$
```

- We deploy the project with the command "gcloud app deploy".
- We must make sure that service 2 is displayed.

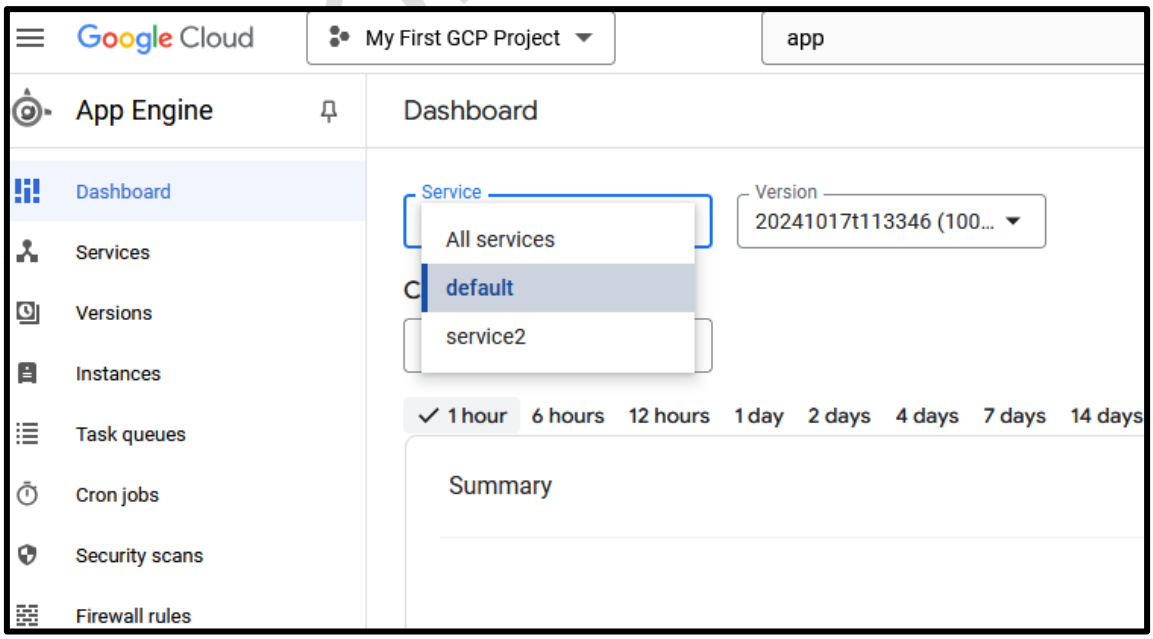
```
gcp@cloudshell:~/app-engine/project2 (centering-oxide-437720-n3)$ gcloud app deploy
Services to deploy:

descriptor:                  [/home/gcp/app-engine/project2/app.yaml]
source:                      [/home/gcp/app-engine/project2]
target project:              [centering-oxide-437720-n3]
target service:              [service2]
target version:              [20241019t144003]
target url:                  [https://service2-dot-centering-oxide-437720-n3.uc.r.appspot.com]
target service account:      [centering-oxide-437720-n3@appspot.gserviceaccount.com]


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

- We write Y to continue and it should deploy smoothly.

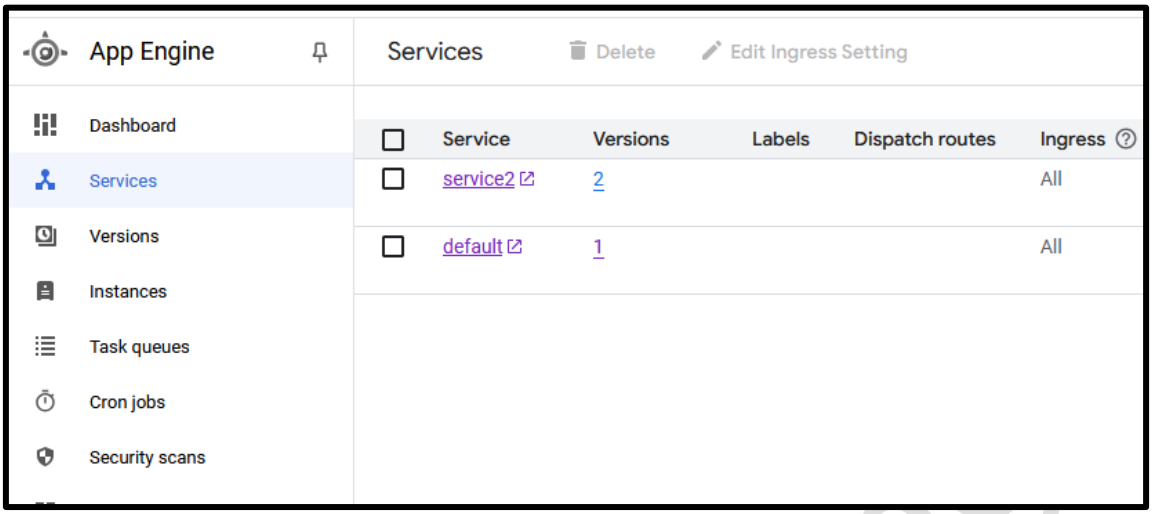
```
Beginning deployment of service [service2]...
Created .gcloudignore file. See `gcloud topic gcloudignore` for details.
Uploading 2 files to Google Cloud Storage
50%
100%
100%
File upload done.
Updating service [service2]...done.
Setting traffic split for service [service2]...done.
Deployed service [service2] to [https://service2-dot-centering-oxide-437720-n3.uc.r.appspot.com]

You can stream logs from the command line by running:
  $ gcloud app logs tail -s service2

To view your application in the web browser run:
  $ gcloud app browse -s service2
```

- We copy the URL it provides and open it in a browser.

- If we leave the Cloud Shell and return to the Google Cloud console
- The new service should appear in the Dashboards tab



- In the services tab we must have the new service



-