

CLOUD COMPUTING

WORKING WITH GOOGLE CLOUD



Google Cloud Platform

Author

Carlos Sánchez Páez

<http://www.github.com/csp98>

BUDAPEST UNIVERSITY OF TECHNOLOGY AND ECONOMICS

ACADEMIC YEAR 2018-2019

Contents

1	What is Google Cloud Platform?	2
1.1	Main advantages of the platform	2
2	Companies that use it	4
3	Google Cloud Platform products	4
3.1	Computing	4
3.2	Networking	5
3.3	Big Data	5
3.4	Machine Learning	5
4	Example of use	5
	Bibliography	9

1 What is Google Cloud Platform?

Google Cloud Platform (GCP) is a set of Google computing resources. These resources are internally used by the company to develop their own products, such as YouTube, Search, Translate, etc. These services are available to the public via services (public cloud offer).

Google Cloud Platform is the third company in the public cloud market (behind Amazon Web Services and Microsoft Azure). It offers IaaS, SaaS, PaaS, etc. in more than 50 products.

1.1 Main advantages of the platform

- Free customization of the resources of the VM or the possibility of using a template depending on the load that it will have.
- Interactive tutorials in the platform to avoid people getting lost.
- Project Control Panel to manage the project and study statistics, metrics, error reporting, etc.
- Android & IOS application to access the control panel from everywhere.
- Online terminal in the website or downloadable SDK to work from the system's command-line interface.
- Free trial of one year and 300 USD to spend in any product.

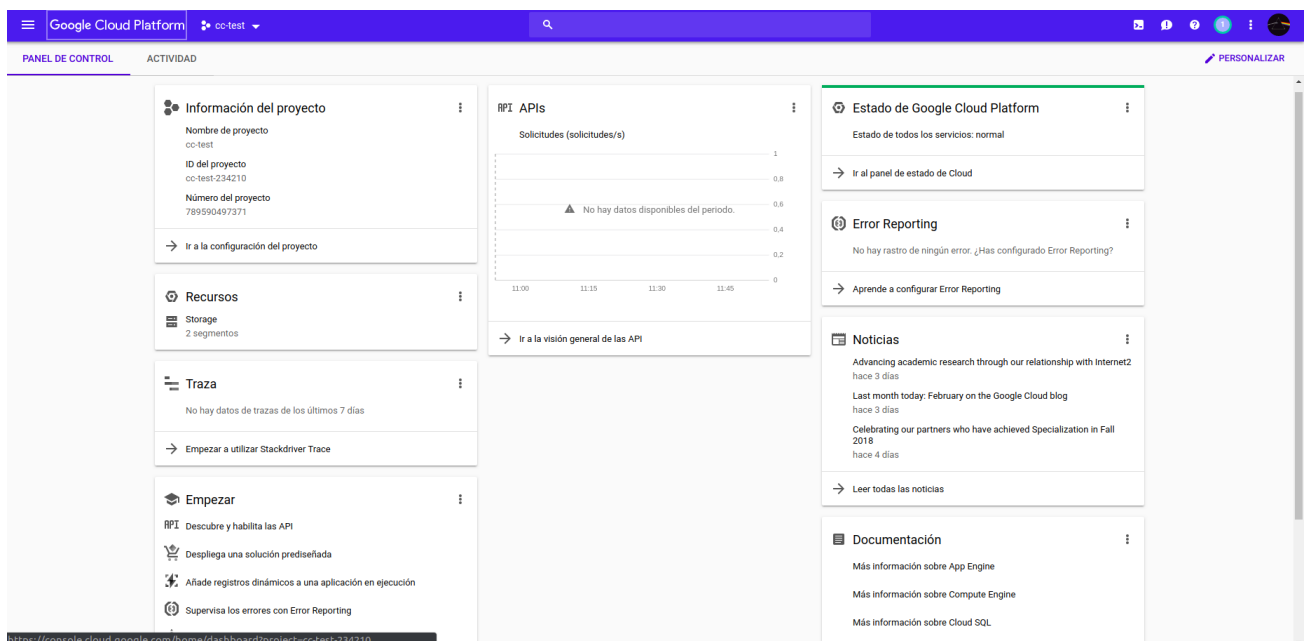


Figure 1: Project Control Panel

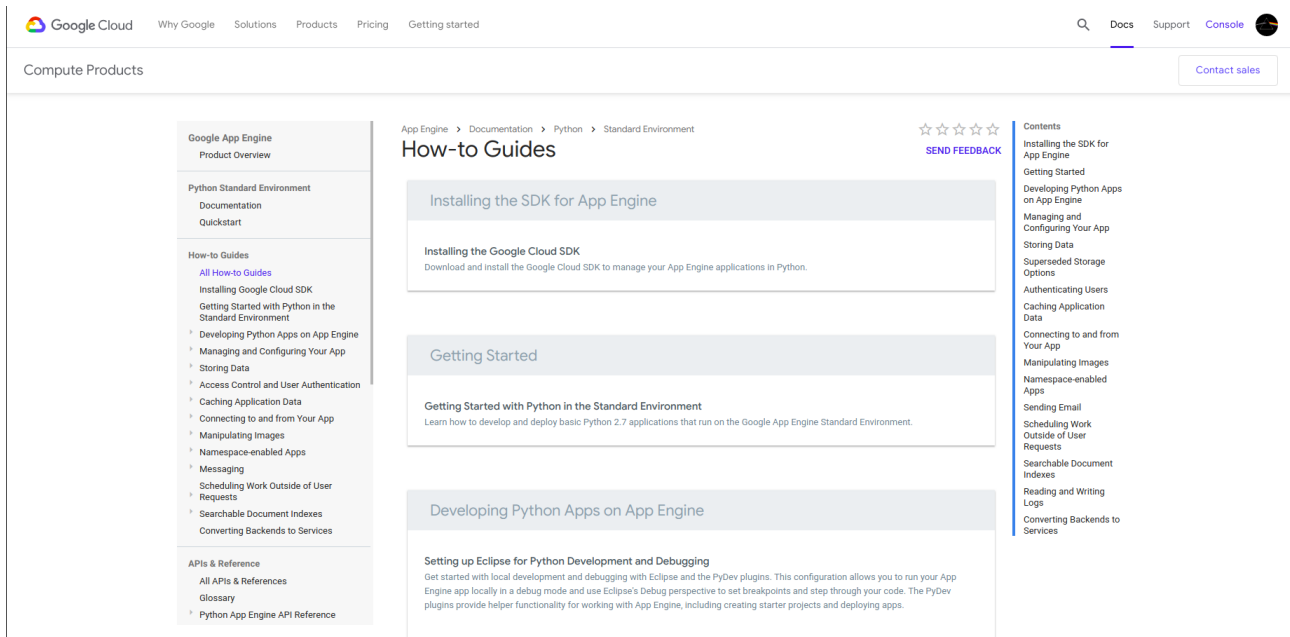


Figure 2: Tutorials in the platform

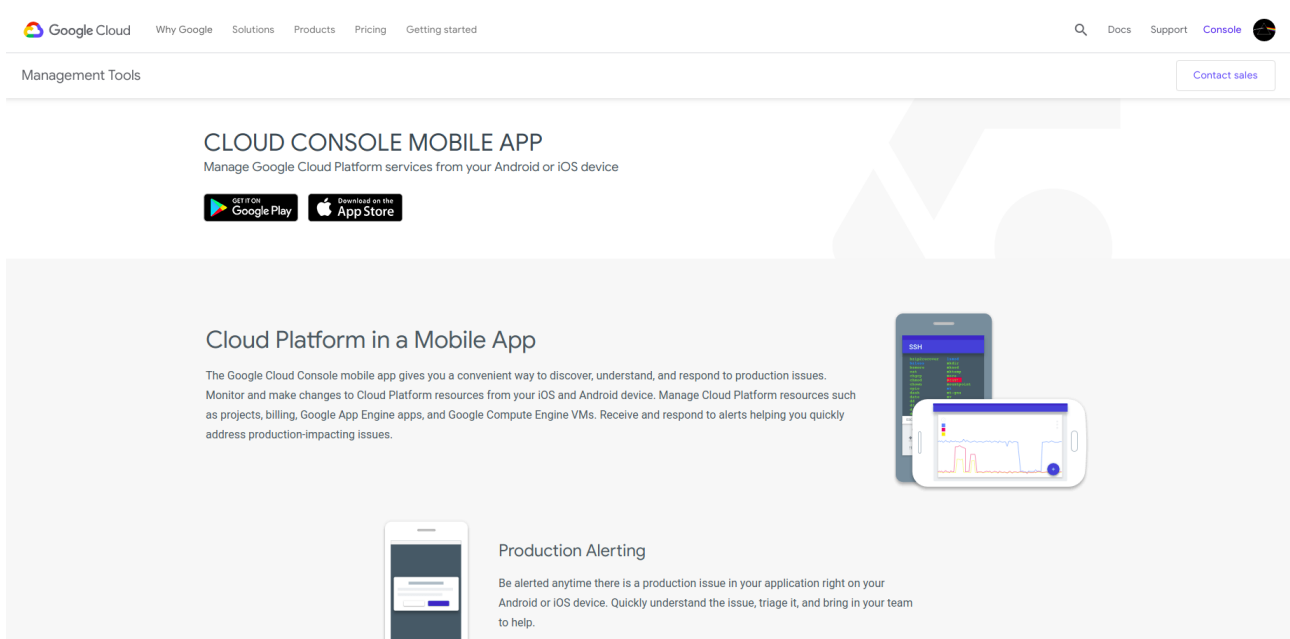


Figure 3: Mobile app

2 Companies that use it

A huge amount of companies use GCP. In February 2018, Google reported that it had more than four million paying customers. Some of the most important businesses are:

- New York Times.
- Twitter.
- Ebay.
- Telenor.
- Paypal.



Figure 4: Companies that use GCP

3 Google Cloud Platform products

3.1 Computing

- App Engine: offers scalable apps via Platform as a Service.
- Compute Engine allows us to connect to a provisioned machine via SSH (Infrastructure as a Service).
- Kubernetes makes possible to run containers, for example Docker ones. Containers pack applications and their dependences and libraries, so that they can be runt in every platform. This is a case of virtualisation where the hypervisor is Docker.

3.2 Networking

- Cloud Armor is used to prevent Denial of Service attacks.
- Network Service Tiers optimizes our product by taking the best tier according of what we want to improve: performance or cost.
- Load balancing allows applications to automatically scale depending on the demand that they receive.

3.3 Big Data

- Dataprep offers data cleaning and preprocessing.
- Dataflow processes the batches.
- Genomics is used in bioinformatics.

3.4 Machine Learning

- Vision API recognises objects in an image. It is used in Google Lens.
- Speech API is used to transform voice into text. Used in Google Assistant.
- Natural Language API extracts information about something mentioned in a text (sentiment, importance in the text, etc.).
- Translate API is used in Google Translate.

4 Example of use

Let's develop and deploy a simple application using App Engine (PaaS). Our source code is very simple, it just shows the sentence "Hello World" in the homepage. Here we can see it:

```
import webapp2
class MainPage(webapp2.RequestHandler):
    def get(self):
        self.response.headers
        ['Content-Type'] = 'text/plain'
        self.response.write('Hello world!')

app = webapp2.WSGIApplication([
    ('/', MainPage),
], debug=True)
```

Figure 5: Source code of the app

The first step is creating the project. When doing that, we will be able to select the region for the deployment so that the nearest users have the lowest latency.

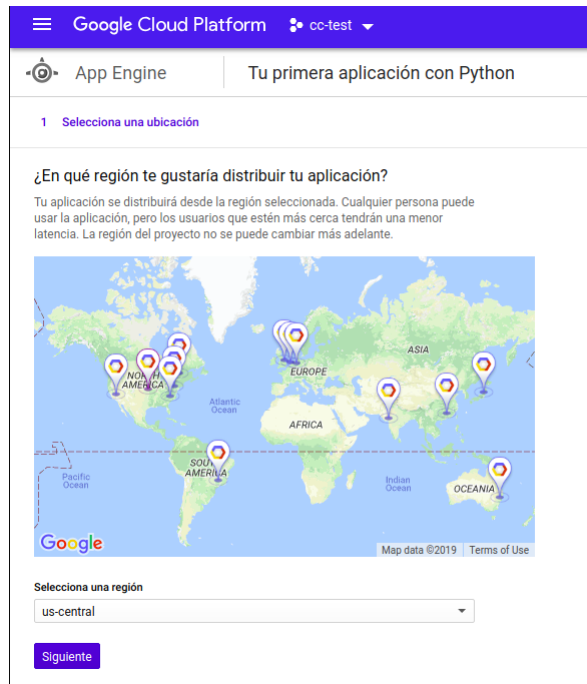


Figure 6: Selection of the deployment region

Then, we must set the project up with the command `gcloud config set project iID_i` where ID is the identifier of the project.

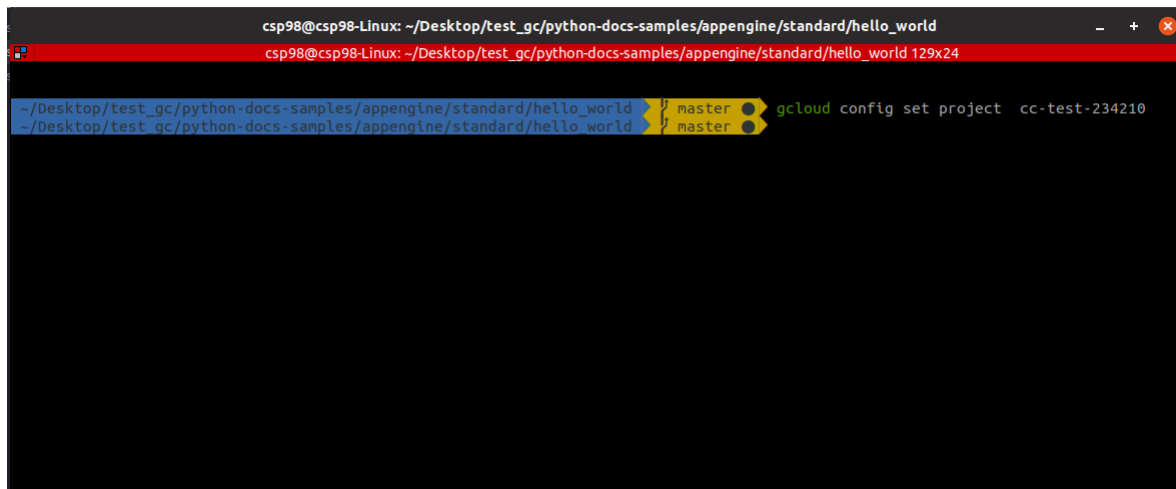
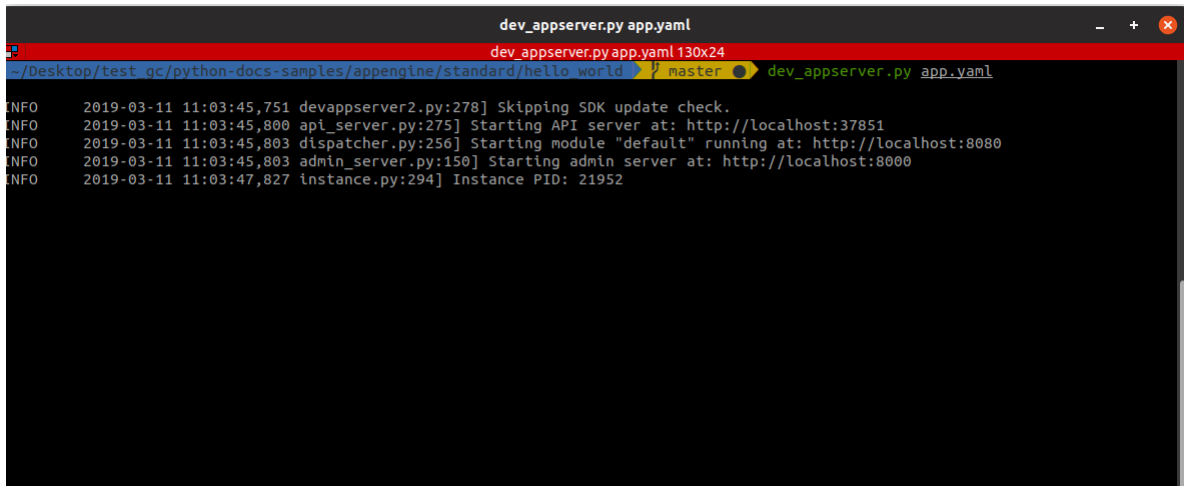


Figure 7: Setting up the project in the SDK

We can use the `dev_appserver.py` utility included in the GCP SDK to test the application in our local computer. This utility is modification-aware, so if we change the source code while executing it, the app will automatically re-deploy to show the changes.



```
dev_appserver.py app.yaml
dev_appserver.py app.yaml 130x24
~/Desktop/test_gc/python-docs-samples/appengine/standard/hello_world master dev_appserver.py app.yaml
INFO 2019-03-11 11:03:45,751 devappserver2.py:278] Skipping SDK update check.
INFO 2019-03-11 11:03:45,800 api_server.py:275] Starting API server at: http://localhost:37851
INFO 2019-03-11 11:03:45,803 dispatcher.py:256] Starting module "default" running at: http://localhost:8080
INFO 2019-03-11 11:03:45,803 admin_server.py:150] Starting admin server at: http://localhost:8000
INFO 2019-03-11 11:03:47,827 instance.py:294] Instance PID: 21952
```

Figure 8: Local deployment using *dev_appserver.py*

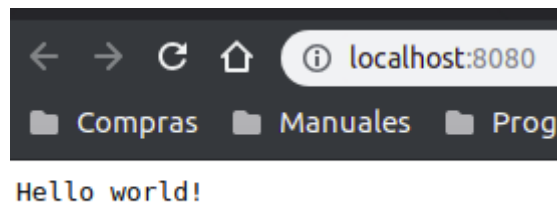
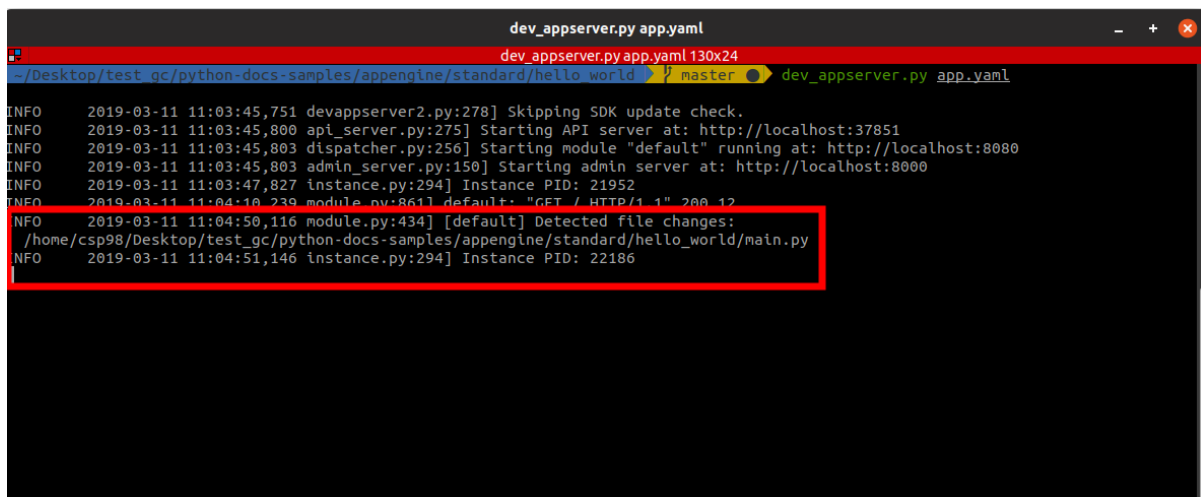


Figure 9: Checking the local deployment



```
dev_appserver.py app.yaml
dev_appserver.py app.yaml 130x24
~/Desktop/test_gc/python-docs-samples/appengine/standard/hello_world master dev_appserver.py app.yaml
INFO 2019-03-11 11:03:45,751 devappserver2.py:278] Skipping SDK update check.
INFO 2019-03-11 11:03:45,800 api_server.py:275] Starting API server at: http://localhost:37851
INFO 2019-03-11 11:03:45,803 dispatcher.py:256] Starting module "default" running at: http://localhost:8080
INFO 2019-03-11 11:03:45,803 admin_server.py:150] Starting admin server at: http://localhost:8000
INFO 2019-03-11 11:03:47,827 instance.py:294] Instance PID: 21952
INFO 2019-03-11 11:04:10,239 module.py:861] default: "GET / HTTP/1.1" 200 12
INFO 2019-03-11 11:04:50,116 module.py:434] [default] Detected file changes:
/home/csp98/Desktop/test_gc/python-docs-samples/appengine/standard/hello_world/main.py
INFO 2019-03-11 11:04:51,146 instance.py:294] Instance PID: 22186
```

Figure 10: Detection of the modifications

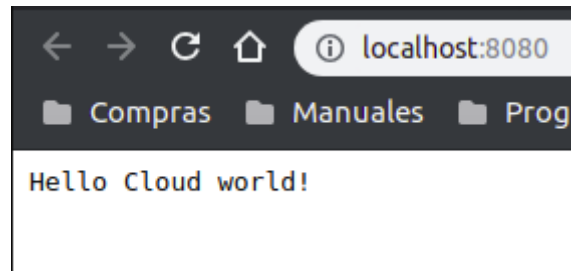


Figure 11: Refreshing the local deployment to see the changes

When the app is finished, it's time to deploy it. We just have to type `gcloud app deploy`. We will see a summary of the properties of the project. After the process finishes, the app is ready on the specified website.

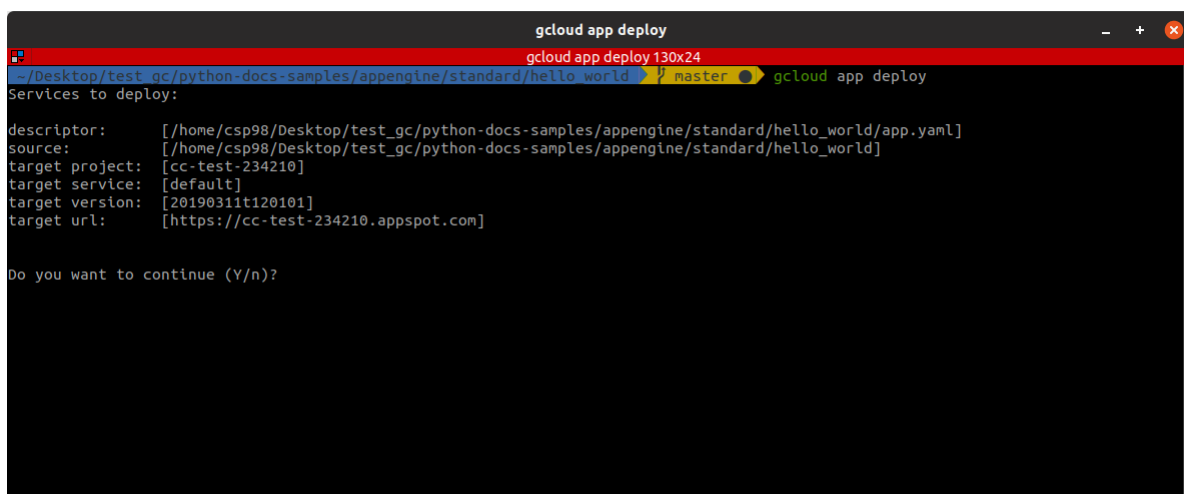


Figure 12: Summary of the project

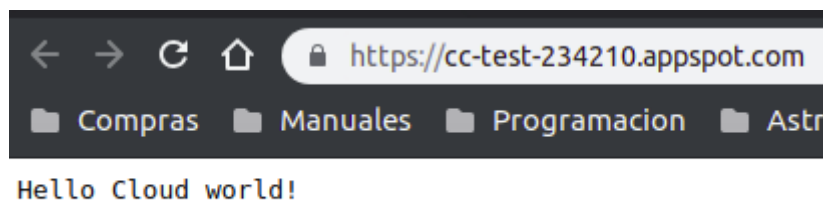


Figure 13: Browsing the project URL

Bibliography

- [1] Course Webpage
<https://www.iit.bme.hu/>
- [2] <https://cloud.google.com/appengine/docs/standard/python/how-to>
- [3] <https://medium.com/@retomeier/what-is-googles-cloud-platform-d92a9c9e5e89>
- [4] <https://cloud.google.com/appengine/docs/standard/python/quickstart>
- [5] <https://cloud.google.com/customers/>
- [6] <https://searchitoperations.techtarget.com/tip/What-are-containers-and-how-do-they-work>