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?

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 (bewind 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 stadistics, 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.

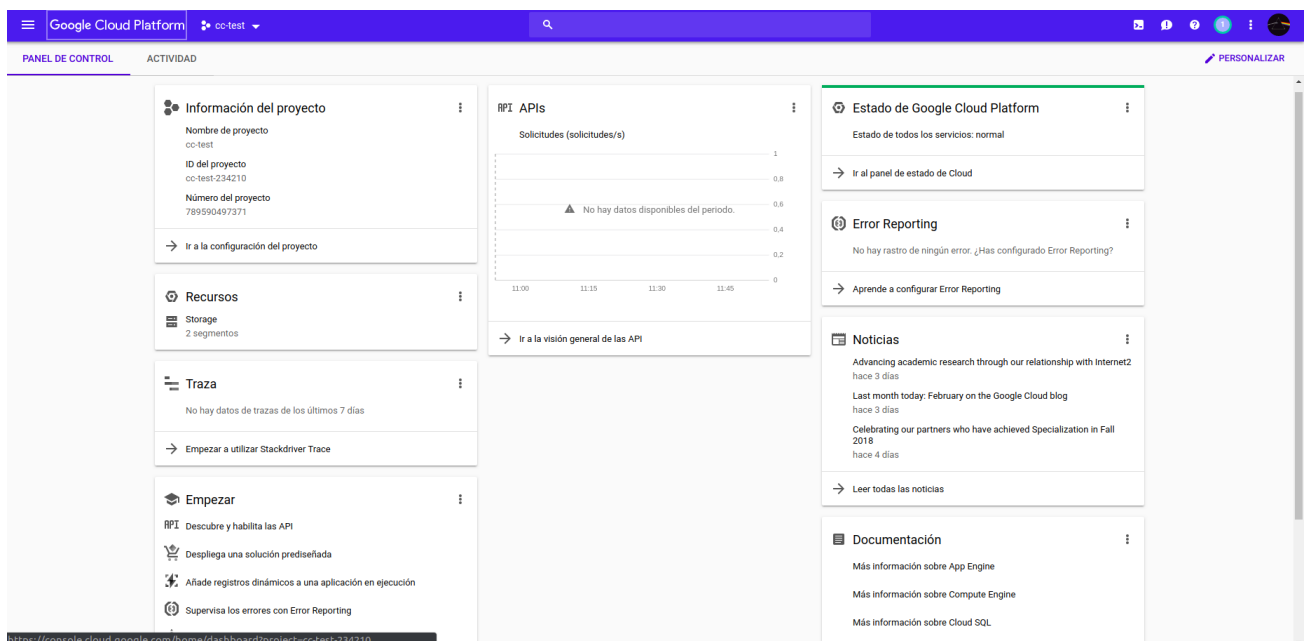- Free trial of one year and 300 USD to spend in any product.
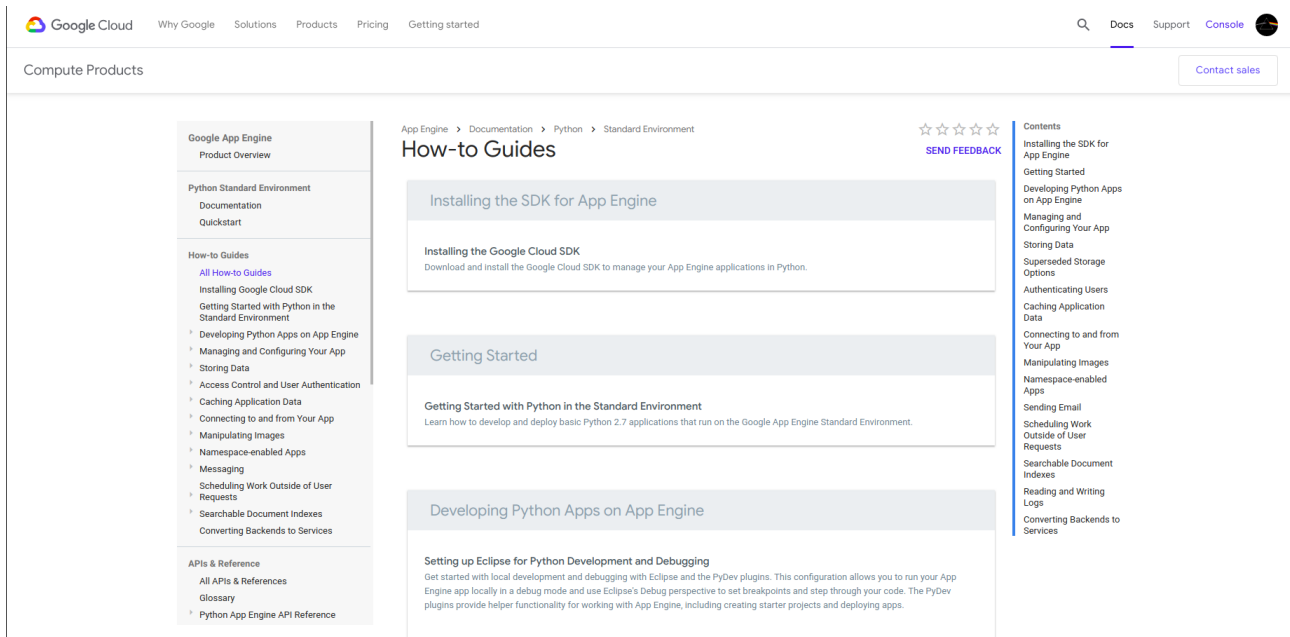

Figure 1: Project Control Panel

Google Cloud — Why Google Solutions Products Pricing Getting started — Docs Support Console

Compute Products — Contact sales

App Engine > Documentation > Python > Standard Environment

How-to Guides

SEND FEEDBACK

Google App Engine
Product Overview

Python Standard Environment
Documentation
Quickstart

How-to Guides
All How-to Guides
Installing Google Cloud SDK
Getting Started with Python in the Standard Environment
Developing Python Apps on App Engine
Managing and Configuring Your App
Storing Data
Access Control and User Authentication
Caching Application Data
Connecting to and from Your App
Manipulating Images
Namespace-enabled Apps
Messaging
Scheduling Work Outside of User Requests
Searchable Document Indexes
Converting Backends to Services

APIs & Reference
All APIs & References
Glossary
Python App Engine API Reference

Installing the SDK for App Engine

Installing the Google Cloud SDK
Download and install the Google Cloud SDK to manage your App Engine applications in Python.

Getting Started

Getting Started with Python in the Standard Environment
Learn how to develop and deploy basic Python 2.7 applications that run on the Google App Engine Standard Environment.

Developing Python Apps on App Engine

Setting up Eclipse for Python Development and Debugging
Get started with local development and debugging with Eclipse and the PyDev plugins. This configuration allows you to run your App Engine app locally in a debug mode and use Eclipse's Debug perspective to set breakpoints and step through your code. The PyDev plugins provide helper functionality for working with App Engine, including creating starter projects and deploying apps.

Contents
Installing the SDK for App Engine
Getting Started
Developing Python Apps on App Engine
Managing and Configuring Your App
Storing Data
Superseded Storage Options
Authenticating Users
Caching Application Data
Connecting to and from Your App
Manipulating Images
Namespace-enabled Apps
Sending Email
Scheduling Work Outside of User Requests
Searchable Document Indexes
Reading and Writing Logs
Converting Backends to Services

Figure 2: Tutorials in the platform

Google Cloud — Why Google Solutions Products Pricing Getting started — Docs Support Console

Management Tools — Contact sales

CLOUD CONSOLE MOBILE APP
Manage Google Cloud Platform services from your Android or iOS device

GET IT ON Google Play   Download on the App Store

Cloud Platform in a Mobile App

The Google Cloud Console mobile app gives you a convenient way to discover, understand, and respond to production issues. Monitor and make changes to Cloud Platform resources from your iOS and Android device. Manage Cloud Platform resources such as projects, billing, Google App Engine apps, and Google Compute Engine VMs. Receive and respond to alerts helping you quickly address production-impacting issues.

Production Alerting
Be alerted anytime there is a production issue in your application right on your Android or iOS device. Quickly understand the issue, triage it, and bring in your team to help.
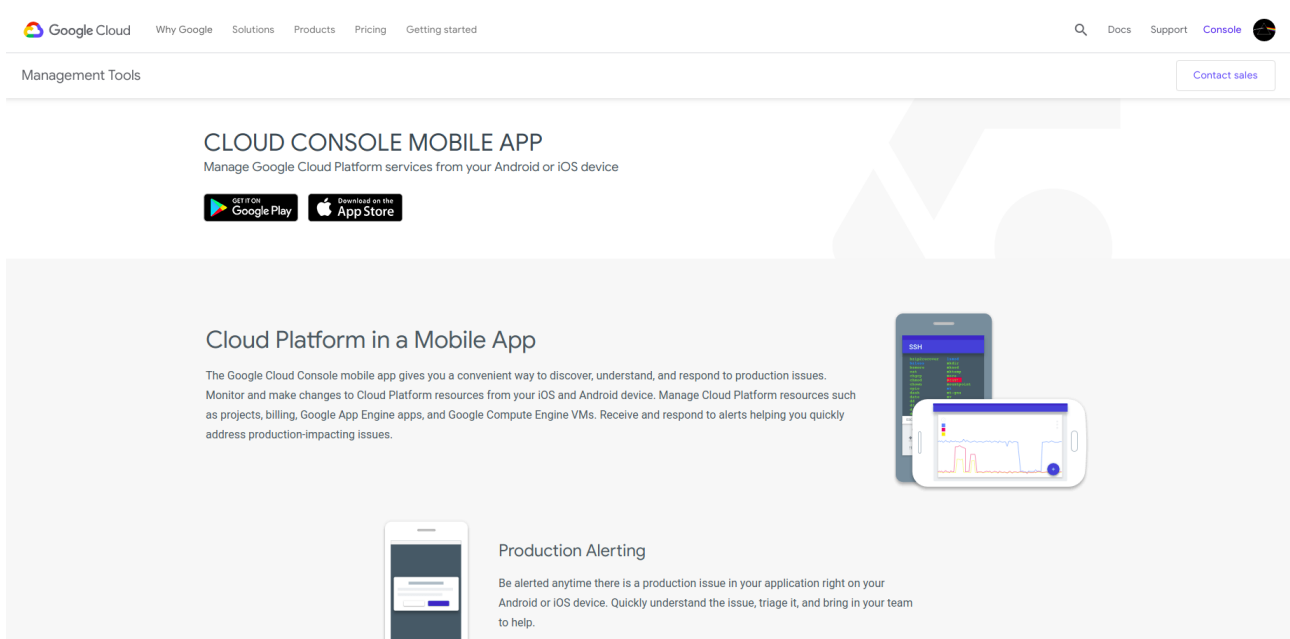
Figure 3: Mobile app

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 test, 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:

```python
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.
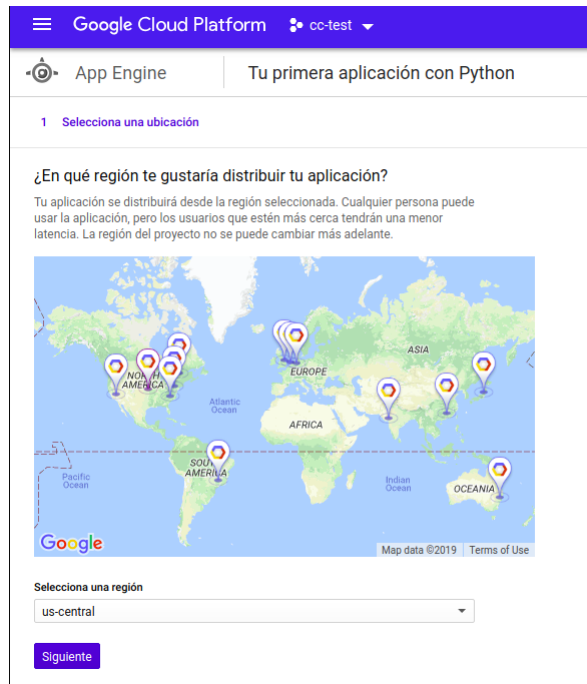
Figure 6: Selection of the deployment region

Then, we must set the project up with the command *gcloud config set project ¡ID¿* where ID is the identificator of the project.
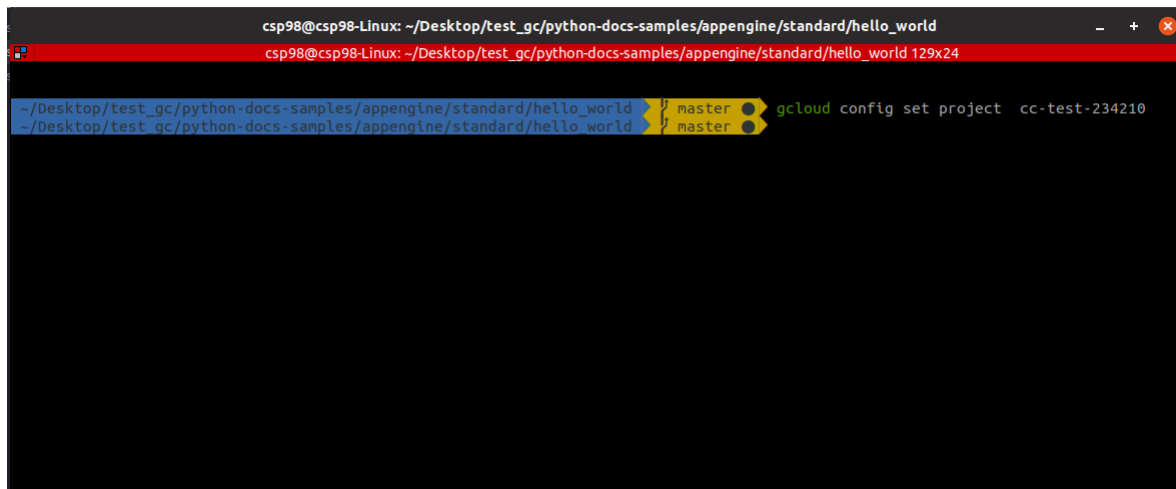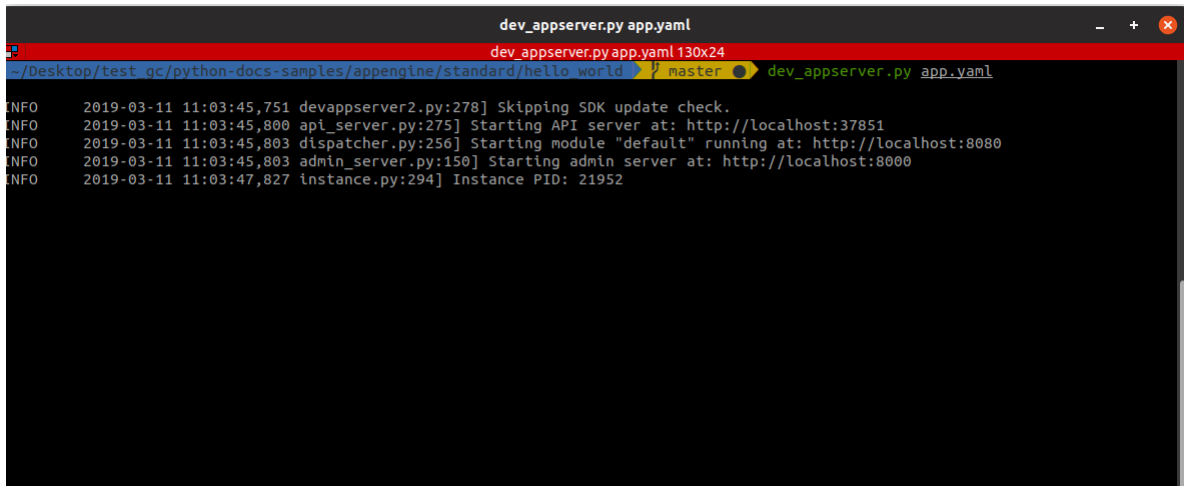
Figure 7: Setting up the project in the SDK

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.

Figure 8: Local deployment using *dev_appserver.py*
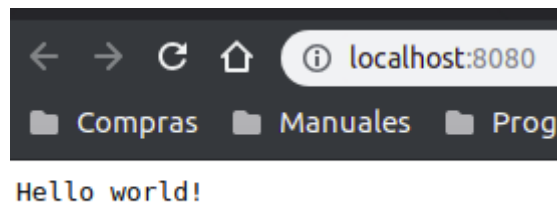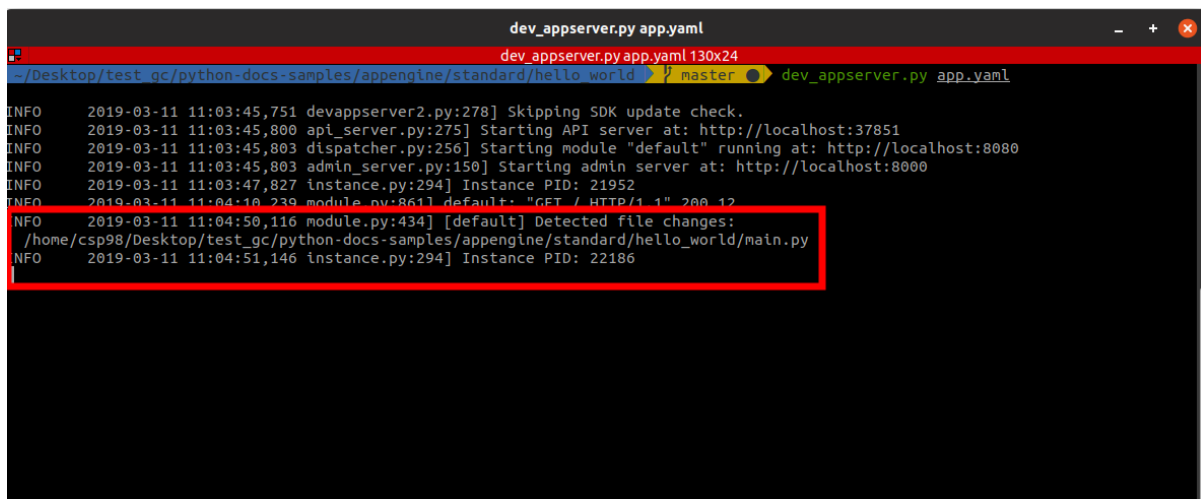
Figure 9: Checking the local deployment

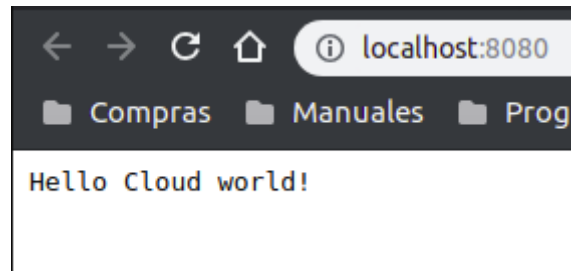Figure 10: Detection of the modifications

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.
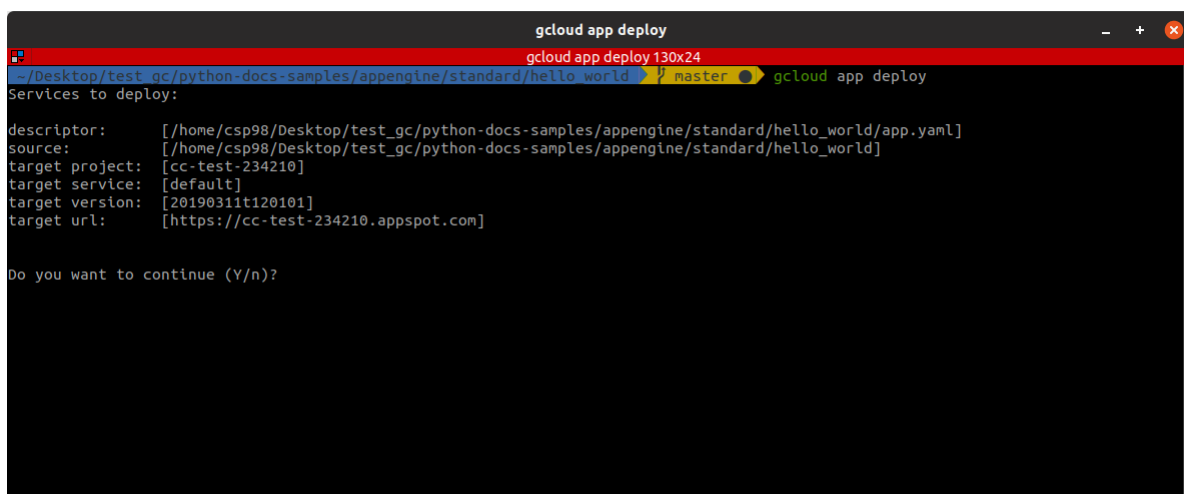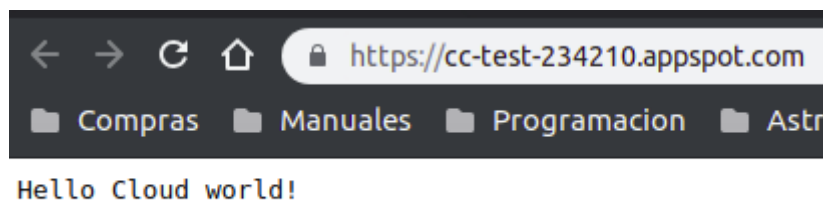

Figure 12: Summary of the project

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`