|  |
| --- |
| **[ cLOUD COMPUTING ]** |
|  |

Practical: 9

**-:AIM:-**

**Configuration of Google App Engine**

Submitted By: Dharmay Sureja

Enrollment No:17012011056



**GANPAT UNIVERSITY**

**U. V. Patel College of Engineering**

**Computer Engineering Department**

**AIM:- Configuration of Google App Engine**

# 1.Abstract

**Google App Engine** was first released as a [beta](http://en.wikipedia.org/wiki/Beta_software) version in April 2008. It is a platform for developing and hosting [web applications](http://en.wikipedia.org/wiki/Web_application) in Google-managed data centers. Google’s

App Engine opens Google’s production to any person in the world at no charge. Much like Google gives us all free email with an amazing amount of long term storage, we now have the ability to run the software that we write in Google’s data centers.

Google App Engine is [cloud computing](http://en.wikipedia.org/wiki/Cloud_computing) technology. Google App Engine is software that facilitates the user to run his web applications on Google infrastructure. It is more reliable because failure of any server will not affect either the performance of the end user or the service of Google.

**2.Introduction**

Google App Engine lets you run your web applications on Google's infrastructure. App Engine applications are easy to build, easy to maintain, and easy to scale as your traffic and data storage needs grow. With App Engine, there are no servers to maintain: You just upload your application, and it's ready to serve your users.

You can serve your app from your own domain name (such as http://www.example.com/) using [Google Apps.](http://www.google.com/a/) Or, you can serve your app using a free name on the appspot.com domain. You can share your application with the world, or limit access to members of your organization.

Google App Engine supports apps written in several programming languages. With App Engine's Java runtime environment, you can build your app using standard Java technologies, including the JVM, Java servlets, and the Java programming language—or any other language using a JVM-based interpreter or compiler, such as JavaScript or Ruby. App Engine also features a dedicated Python runtime environment, which includes a fast Python interpreter and the Python standard library. The Java and Python runtime environments are built to ensure that your application runs quickly, securely, and without interference from other apps on the system.

**3.Cloud Endpoints Components**

* **Extensible Service Proxy**

The Extensible Service Proxy is a NGINX-based proxy that runs in front of the backend and injects Cloud Endpoints functionality such as authentication, monitoring and logging. The proxy retrieves a service configuration from Google Service Management and uses it to perform validation on incoming requests.

The proxy is designed to be deployed in a containerized environment and validate JWTs and Google ID tokens. It employs a variety of techniques such as heavy caching and asynchronous calls to remain lightweight and highly performant.

* **Google Service Control**

[Service Control](https://cloud.google.com/service-control/) applies API management rules at runtime such as key authentication, monitoring, and logging. It has two main features:

* Check - verifies authentication and API keys, and indicates whether a call should be permitted
* Report - notifies the systems of record for logging and monitoring
* **Google Service Management**

[Cloud Endpoints](https://cloud.google.com/endpoints/) uses the OpenAPI specification to describe an API. Deploying this specification to Service Management - typically using Google Cloud SDK - configured the API management rules. Other configuration related tasks also happen here, such as sharing your API with other developers, enabling/disabling the API in different projects, and generating API keys.

* **Google Cloud SDK**

[Cloud SDK](https://cloud.google.com/sdk/) provides the gcloud command-line application that can be used to make calls to various services of Google Cloud Platform. The gcloud command-line tool is also used to deploy your API Configuration to Service Management.

* **Google Cloud Console**

[Cloud Console](https://cloud.google.com/cloud-console/) is the graphical user interface for Google Cloud Platform. Cloud Endpoints uses Cloud Console to expose monitoring and logging data that are sent from the proxy and recorded by Service Control and share APIs with other developers, and for them to generate API keys to call the API.

**5.Advantages & Disadvantage**

* **Advantages**
* **Cost-effective:** The price - or lack thereof - is the biggest selling point. The [free tier](http://blog.cbtnuggets.com/2014/07/5-reasons-learn-google-app-engine/) [is incredibly generous](http://blog.cbtnuggets.com/2014/07/5-reasons-learn-google-app-engine/) and most small applications don’t need anything more.

However, if you do find that you need more features, even the paid apps are cost-friendly. You only pay for the resources you use, and they’re relatively cheap. Best of all, you only get charged for your application when it handles requests. Talk about maximizing capital!

* **No servers to maintain:** With App Engine, you don’t have any reason or need tomaintain a server. You simply upload your applications and they’re ready to go. Google takes care of the rest for you. This frees up your resources so they can be better utilized in other areas of your business.
* **Fully integrated:** One of the advantages that often flies under the radar is that AppEngine is connected to Google’s many other products and services. App Engine,

BigQuery, Compute Engine, and Cloud Storage all work together for your greater good.

* **Risk Free Trial Period:** we can use its services for free for one month with $300credit initially
* **Disadvantages**
* **You are putting your application in Google's hands. Think** about that for aminute. You are at the mercy of Google -- if disaster strikes and Google one day disappears, you are done too. Or, more realistically, if the Google App Engine goes down for an hour, you are also down for an hour -- and you will have no idea what happened. Even if you try and get an answer from someone at Google, you won't. Just like Google Apps, it will be impossible to explain things to your end users.
* **Once you are in, you are really in** Using Google's infrastructure is very tempting. Butany smart company should have some sort of plan for the future. What if you realized that you didn't want to host your application on Google App Engine anymore? Good luck, almost everything you are given access to is proprietary -- that means all your data is locked into BigTable in a format that isn't like a traditional relational database. It's also very tempting to use the API's Google provides to interface with things like Google accounts.
* **Privacy should not be taken lightly** Google has a very strong privacy policy --and personally I trust them. However, I'm trusting them with my personal information -- you will be trusting them with all of your company's data. These are two completely different things. If you have a low trust tolerance, you may not want to risk putting everything that belongs to your company behind Google's doors. That said, I personally would still feel comfortable putting company data on their infrastructure -- simply because I know it's proven to be secure, scalable and robust over the last several years with their own services.

# 6. Creating/Deploying Applications

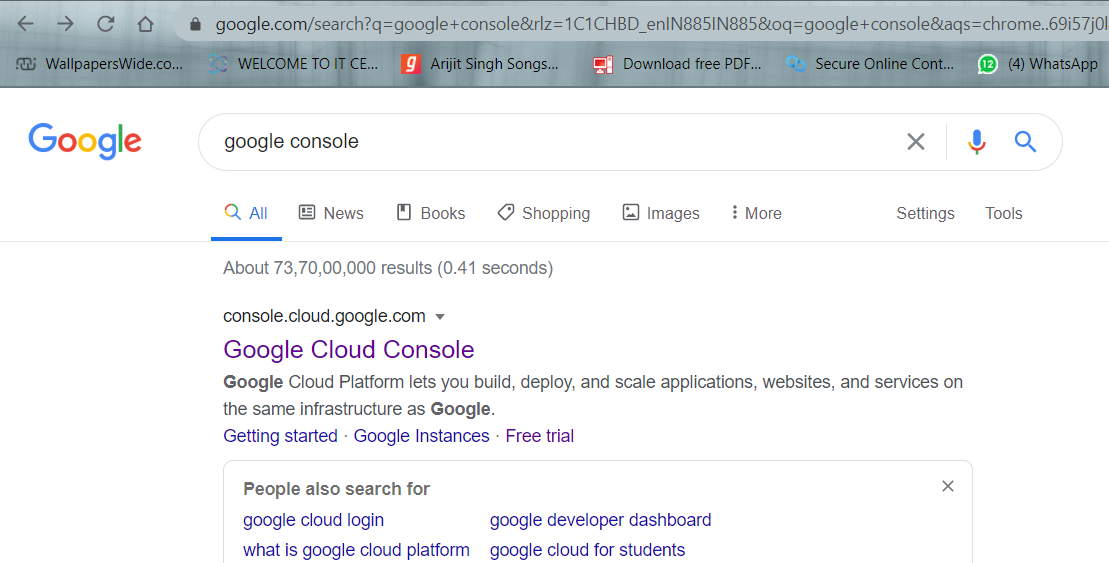
Creating an App Engine application is easy, and only takes a few minutes. And it's free to start: upload your app and share it with users right away, at no charge and with no commitment required.

Google App Engine applications can be written in either the Java or Python programming languages.

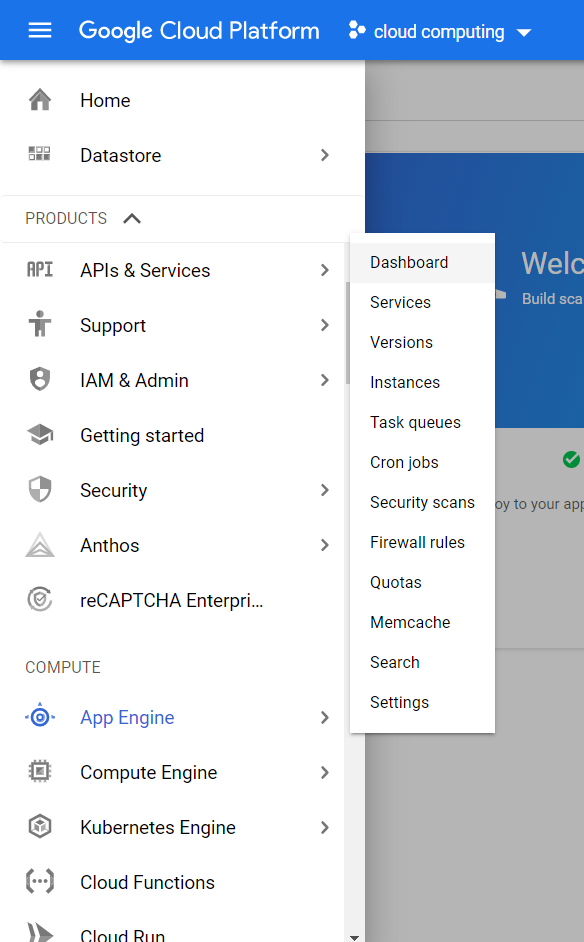
The Steps for how to create an application and deploy it on the app engine is shown below.

DEPLOYING A SIMPLE “HELLO WORLD” APP

**1**.Open Browser & search Google console ,click on the first link.



2.Search and click App Engine

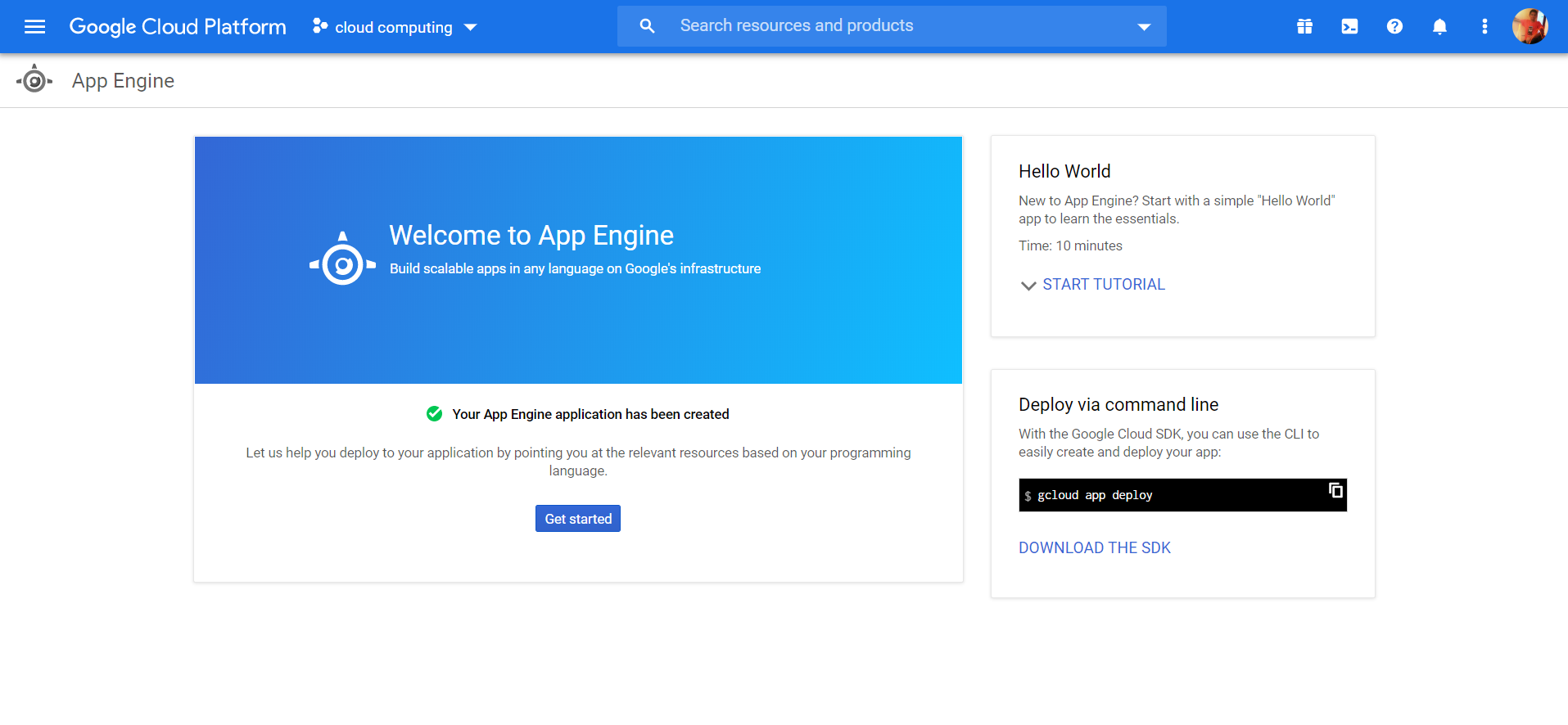


OR

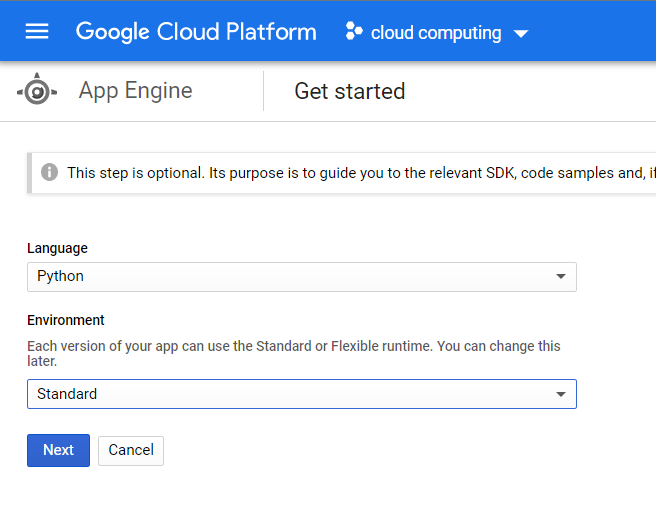
To Open Google app engine write below link into your browser,

<https://cloud.google.com/appengine/>

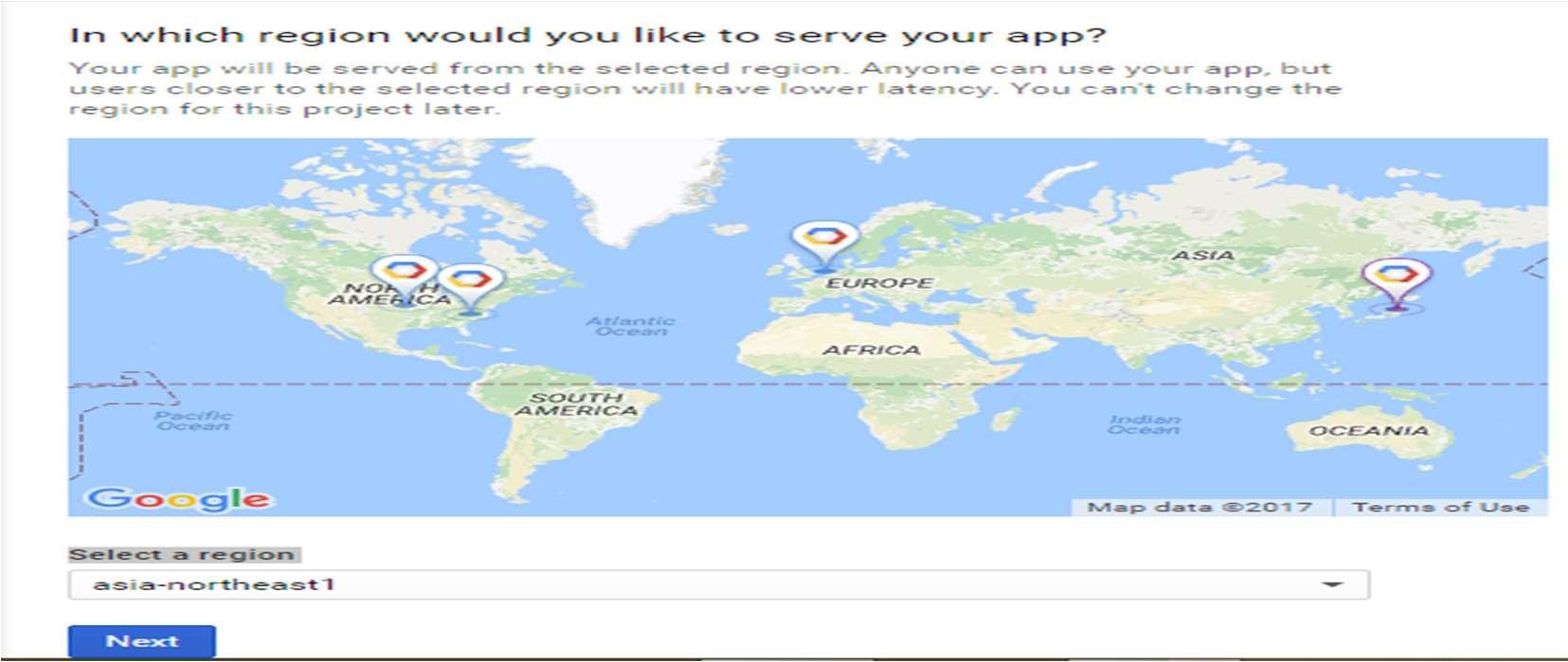
3.Select Get started.



3.Then select language and Environment for your application



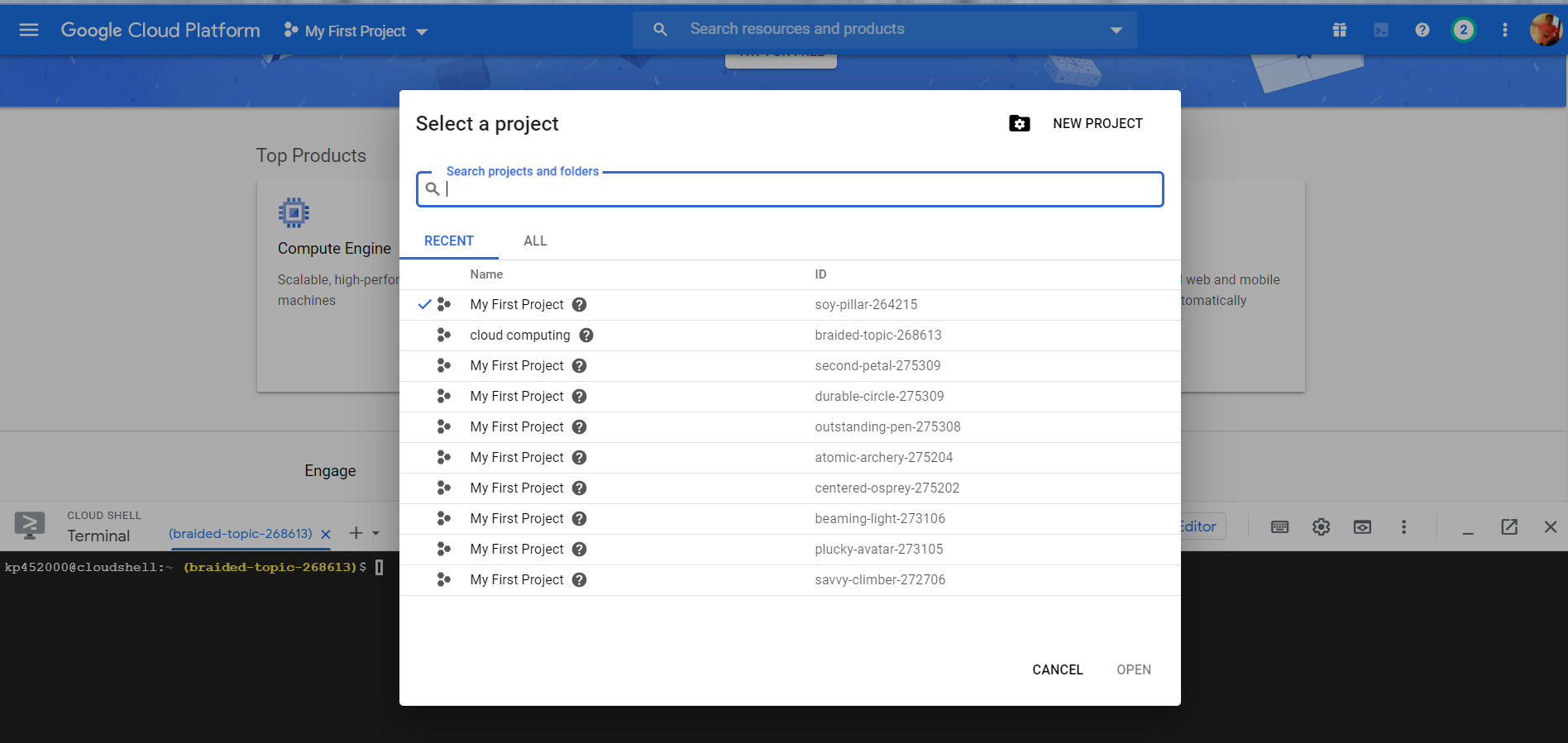
4.Select region where u want to host



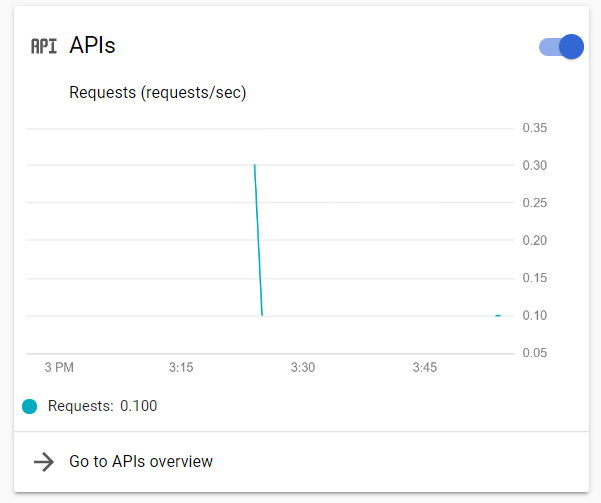
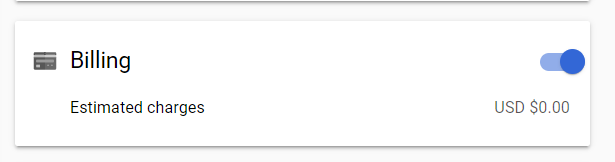
OR

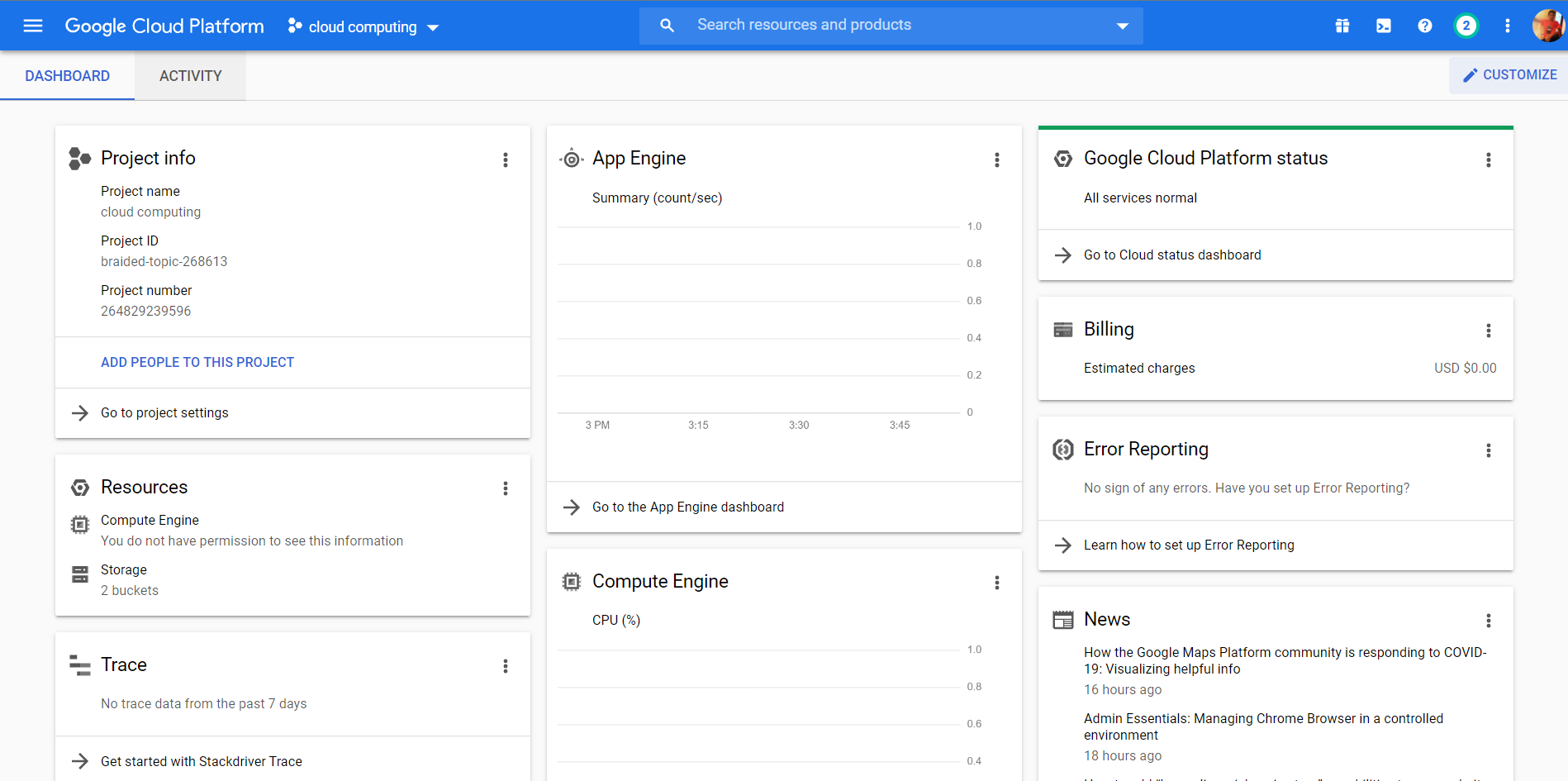
We can select hosts using the command shell.

4.Create Projects or created project select

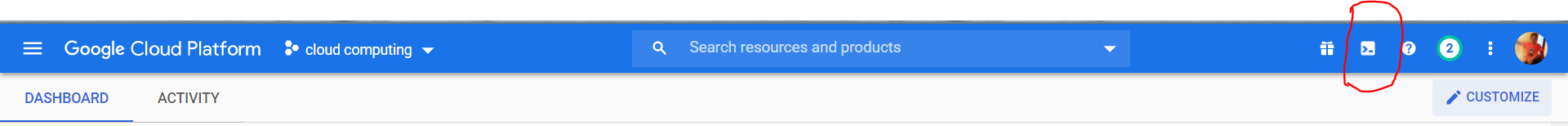


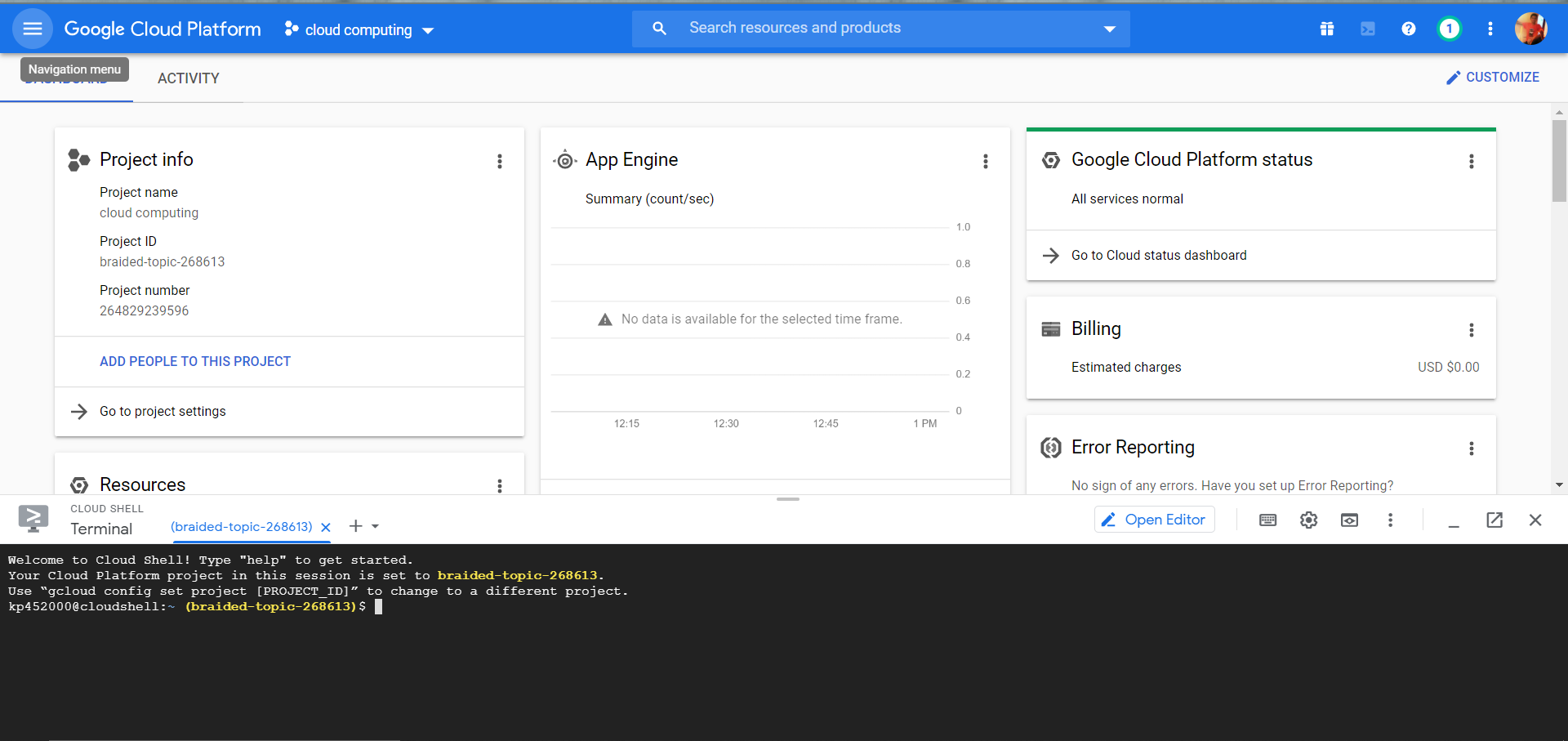
6.Then Finally enabled billing account & APIs





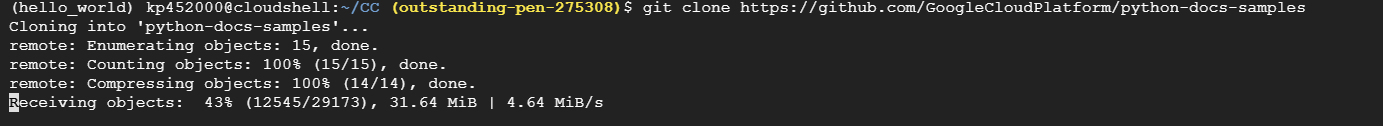
**7**.Open cloud shell by click on top right 2nd icon





1.Download the sample application repository using [Git](https://git-scm.com/):

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



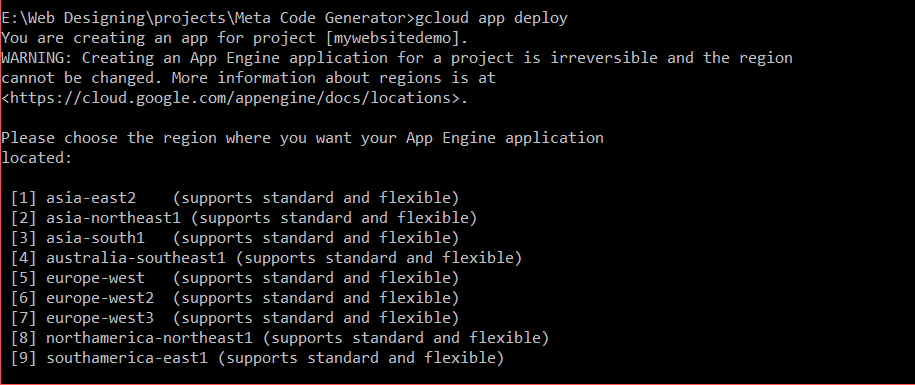
2.Navigate to the directory that contains a copy of the files from the previous step:

cd python-docs-samples/appengine/standard\_python37/building-an-app/building-an-app-1

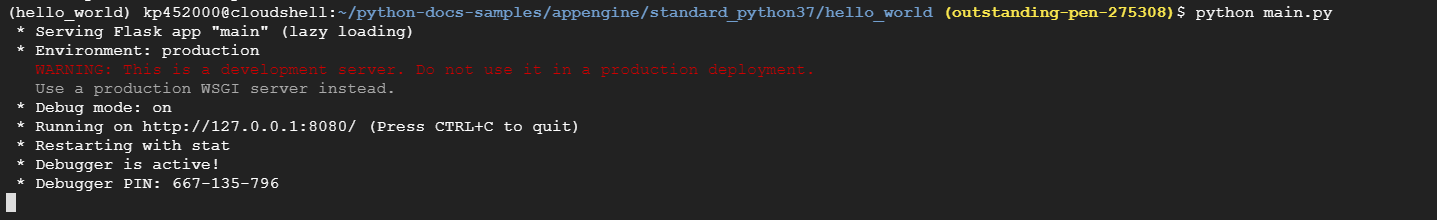


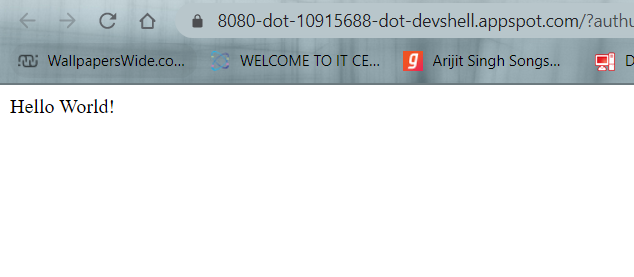
3.Select region

write command : gcloud app deploy

****

Now, See the web preview





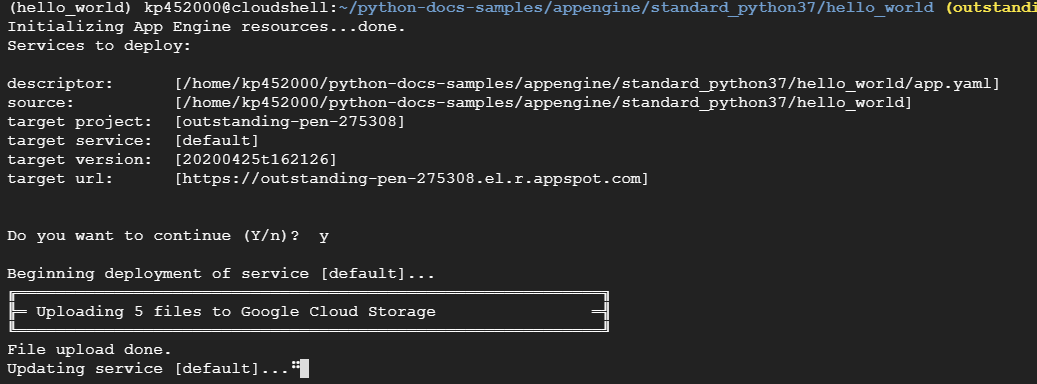
* Deploying with Cloud Shell

You can use Cloud Shell to deploy your app. To deploy your app enter the following:

gcloud app deploy app.yaml **or**  --project outstanding-pen-275308



* Next,type Y to continue to upload files .&



* Visit your app (Finally nameofproject.appspot.com)

Congratulations! Your app has been deployed. The default URL of your app is a subdomain on appspot.com that starts with your project's ID: [outstanding-pen-275308.appspot.com](http://outstanding-pen-275308.appspot.com/).

Try visiting your deployed application.

