	Bansilal Ramnath Agarwal Charitable Trust's Vishwakarma Institute of Information Technology **Department of** **Artificial Intelligence and Data Science**	
Student Name: Pratham Raka		
Class: TY-B.Tech	Division: A	Roll No: 371046
Semester: V	Academic Year: 2021-22	
Subject Name & Code: Cloud Computing and Analytics		
Title of Assignment: Assignment-7 Deploy web applications on commercial cloud Technology: Google App Engine/ Windows Azure 2.		
Date of Performance: 22/11/22	Date of Submission: 08/12/22	

Aim: Deploy web applications on commercial cloud Technology: Google App Engine/ Windows Azure 2.

Problem Statement: Deploy web applications on commercial cloud Technology: Google App Engine/ Windows Azure 2.

Background Information:

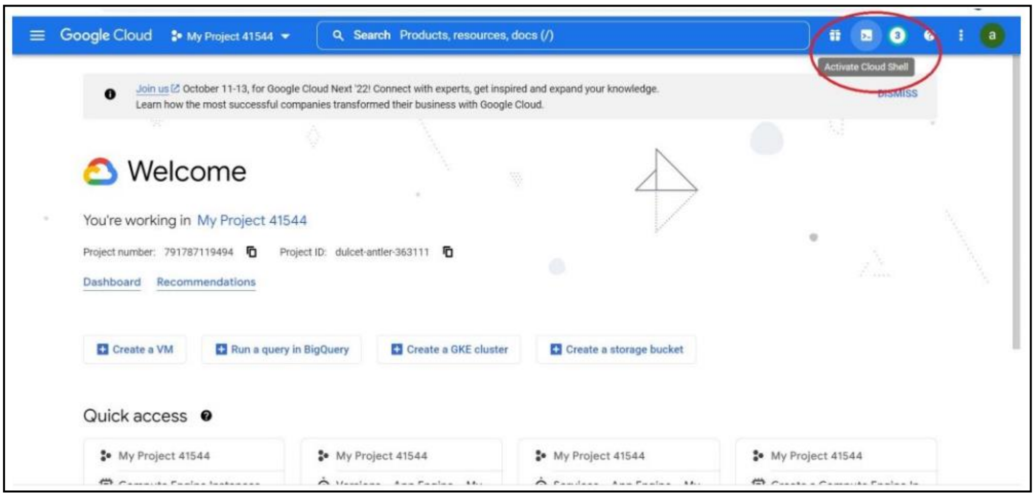
App Engine is a managed service provided by the google cloud platform, it provides you with the simplest way to deploy and scale your applications in GCP and end-to-end application management, it supports Go, Java, .Net, Node.js, PHP, Python, and Ruby, using pre-configured run times, you can also use a custom run-time and write code in any language, App engine also supports running containers, Some of the important features of app engine are Automatic load balancing & Auto scaling, Application health monitoring, Application versioning, Traffic splitting.

App Engine is a PAAS (Platform as a service) serverless/ fully managed service provided by GCP, you will have fewer responsibilities and lower flexibility compared to the app engine.

Cloud Resource Requirements: Google App Engine

Steps:

Step 1: You can launch a cloud shell from the web console as shown below.



Join us October 11-13, for Google Cloud Next '22! Connect with experts, get inspired and expand your knowledge. Learn how the most successful companies transformed their business with Google Cloud.

You're working in My Project 41544

Project number: 791787119494

Project ID: dulcet-antler-363111

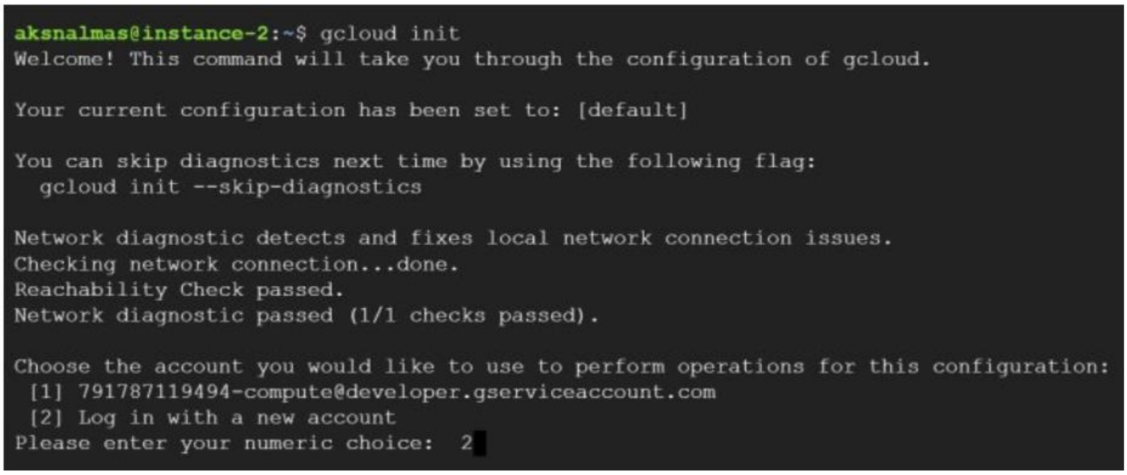
Step2: Installing Gcloud CLI

```
curl -O https://dl.google.com/dl/cloudsdk/channels/rapid/downloads/google-cloud-cli-403.0.0-linux-x86_64.tar.gz

tar -xf google-cloud-cli-403.0.0-linux-x86_64.tar.gz
```

Now run below given command to install gcloud CLI , and enter yes and give the path when it prompts for

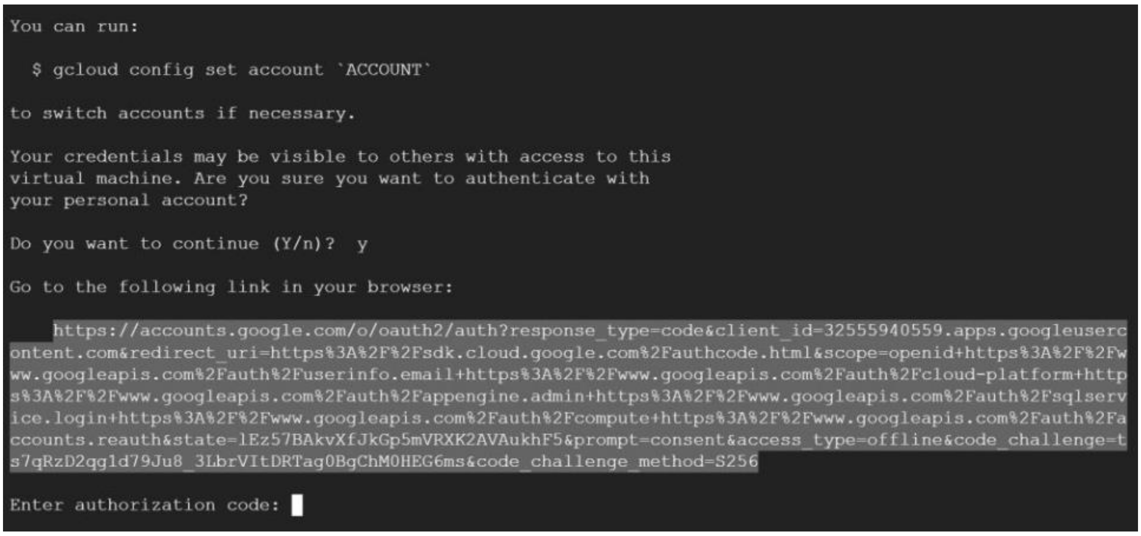
```
./google-cloud-sdk/install.sh
```

Step3: Initializing gcloud CLI: then to initialize gcloud CLI run **gcloud init** command on your terminal, Select the option **Log in with a new account** when it prompts "choose an account that you would like to initialize with, same is shown in the below-given snapshot"



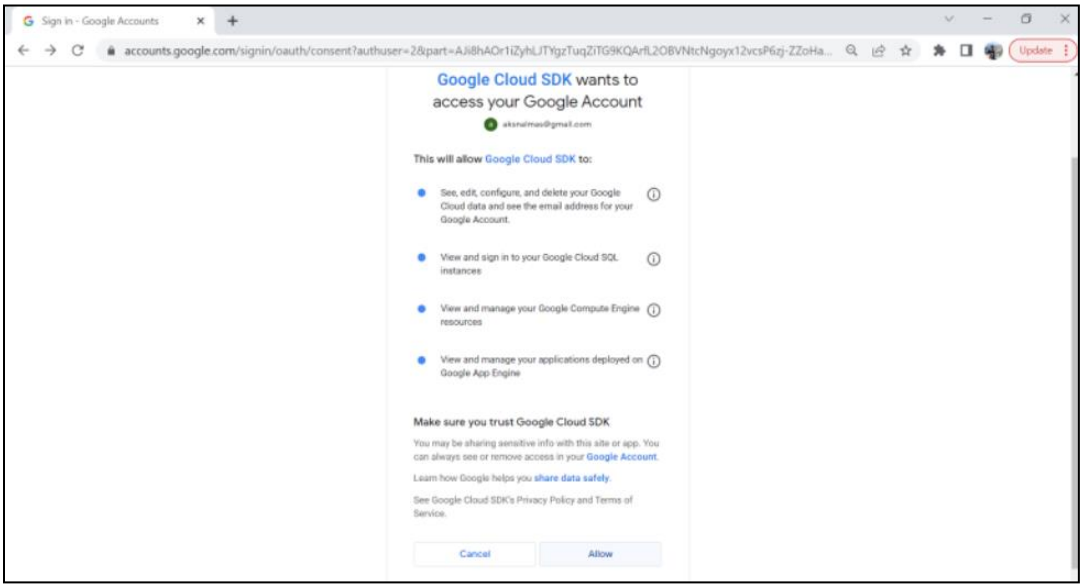
```
aksnalmas@instance-2:~$ gcloud init
Welcome! This command will take you through the configuration of gcloud.

Your current configuration has been set to: [default]

You can skip diagnostics next time by using the following flag:
  gcloud init --skip-diagnostics

Network diagnostic detects and fixes local network connection issues.
Checking network connection...done.
Reachability Check passed.
Network diagnostic passed (1/1 checks passed).

Choose the account you would like to use to perform operations for this configuration:
 [1] 791787119494-compute@developer.gserviceaccount.com
 [2] Log in with a new account
Please enter your numeric choice:  2
```

then a link will be provided which you should copy and open in your browser and log in to your GCP account in order to get the authorization code.



```
You can run:

  $ gcloud config set account `ACCOUNT`

to switch accounts if necessary.

Your credentials may be visible to others with access to this
virtual machine. Are you sure you want to authenticate with
your personal account?

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

Go to the following link in your browser:

    https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=32555940559.apps.googleuserc
ontent.com&redirect_uri=https%3A%2F%2Fsdk.cloud.google.com%2Fauthcode.html&scope=openid+https%3A%2F%2Fw
ww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+http
s%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fsqlserv
ice.login+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fa
ccounts.reauth&state=1Ez57BAkvXfJkGp5mVRXK2AVAukhF5&prompt=consent&access_type=offline&code_challenge=t
s7qRzD2qg1d79Ju8_3LbrVItDRTag0BgChM0HEG6ms&code_challenge_method=S256

Enter authorization code: █
```

After opening the link in your browser, select the account that you want to authorize and click on allow,


Sign in - Google Accounts
accounts.google.com/signin/oauth/consent?authuser=2&part=AJi8hAOr1iZyhLJTYgzTuqZiTG9KQArfL2OBVNtcNgoyx12vcsP6zj-ZZoHa...
Update

Google Cloud SDK wants to access your Google Account

aksnalmas@gmail.com

This will allow Google Cloud SDK to:

- See, edit, configure, and delete your Google Cloud data and see the email address for your Google Account.
- View and sign in to your Google Cloud SQL instances
- View and manage your Google Compute Engine resources
- View and manage your applications deployed on Google App Engine

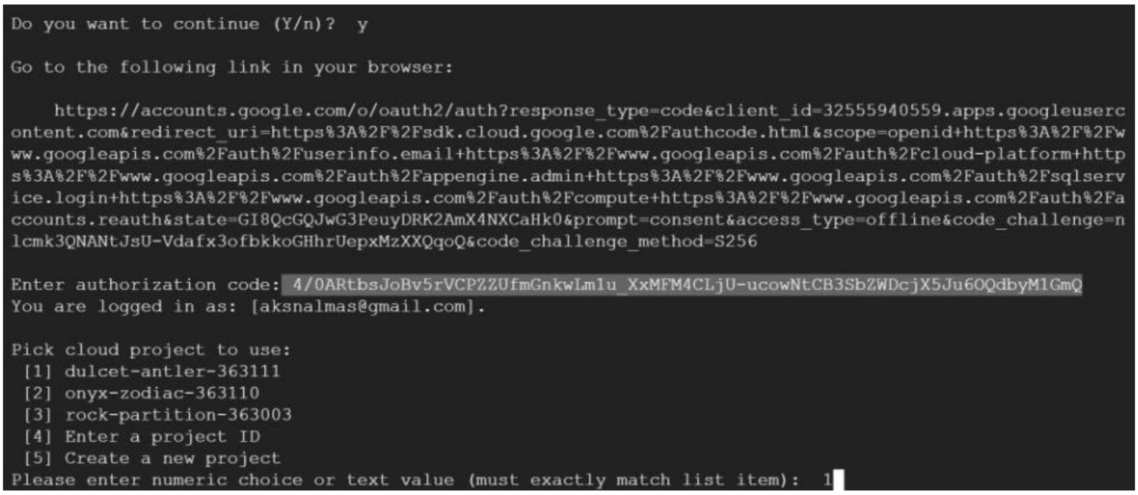
Make sure you trust Google Cloud SDK

You may be sharing sensitive info with this site or app. You can always see or remove access in your Google Account.

Learn how Google helps you share data safely.

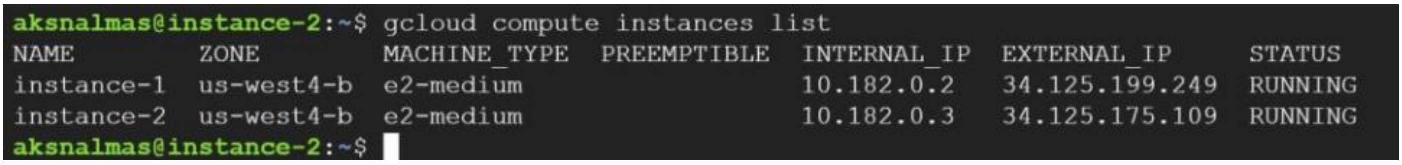
See Google Cloud SDK's Privacy Policy and Terms of Service.

then copy the authorization code and give it in your terminal:



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

Go to the following link in your browser:

    https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=32555940559.apps.googleuserc
ontent.com&redirect_uri=https%3A%2F%2Fsdk.cloud.google.com%2Fauthcode.html&scope=openid+https%3A%2F%2Fw
ww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+http
s%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fsqlserv
ice.login+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fa
ccounts.reauth&state=GI8QcGQJwG3PeuyDRK2AmX4NXCaHk0&prompt=consent&access_type=offline&code_challenge=n
lcmk3QNANtJsU-Vdafx3ofbkkoGHhrUepxMzXXQqoQ&code_challenge_method=S256

Enter authorization code: 4/0ARtbsJoBv5rVCPZZUfmGnkwLm1u_XxMFM4CLjU-ucowNtCB3SbZWDcjX5Ju6OQdbyM1GmQ
You are logged in as: [aksnalmas@gmail.com].

Pick cloud project to use:
 [1] dulcet-antler-363111
 [2] onyx-zodiac-363110
 [3] rock-partition-363003
 [4] Enter a project ID
 [5] Create a new project
Please enter numeric choice or text value (must exactly match list item):  1
```

Check the initialization by using any gcloud command



```
aksnalmas@instance-2:~$ gcloud compute instances list
NAME        ZONE        MACHINE_TYPE  PREEMPTIBLE  INTERNAL_IP  EXTERNAL_IP     STATUS
instance-1  us-west4-b  e2-medium                  10.182.0.2   34.125.199.249  RUNNING
instance-2  us-west4-b  e2-medium                  10.182.0.3   34.125.175.109  RUNNING
aksnalmas@instance-2:~$
```

Step 4: Writing the Web App: create a folder with any name (helloworld is the name used in this blog), and create files named app.yaml, main.py, requirements.txt. To deploy your web app to App Engine, you need a configuration file named **app.yaml** , it defines your web application’s settings for App Engine, **main.py** is where the python code for this specific application is present, **requirements.txt** is where the dependencies of this application are mentioned. Copy the content into the files as given below **main.py**

```
import flask

# If `entrypoint` is not defined in app.yaml, App Engine will look for an app
called `app` in `main.py`.

app = flask.Flask(__name__)

@app.get("/")
```

```
def hello():
    """Return a friendly HTTP greeting."""
    return "Hello World!\n"
if __name__ == "__main__":
    app.run(host="localhost", port=8080, debug=True)
```

above given code is written using flask frame work , this just gives **Hello World!** on the website.

requirements.txt:

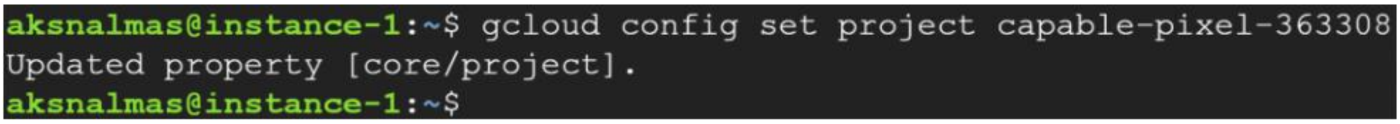
```
Flask==2.0.3
```

app.yaml:

```
runtime: python39
```

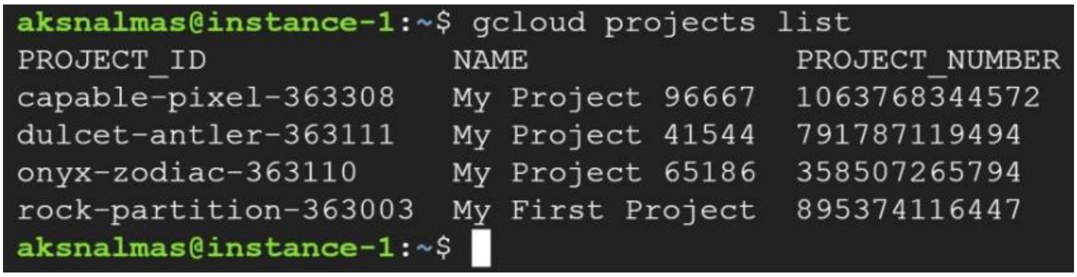
Now let us deploy this application on google App Engine, go on to your terminal, and set the project that you want to deploy on, use the below-given command.

```
gcloud config set project <project-ID>
```



```
aksnalmas@instance-1:~$ gcloud config set project capable-pixel-363308
Updated property [core/project].
aksnalmas@instance-1:~$
```

you can get project-ID by using the command **gcloud projects list**

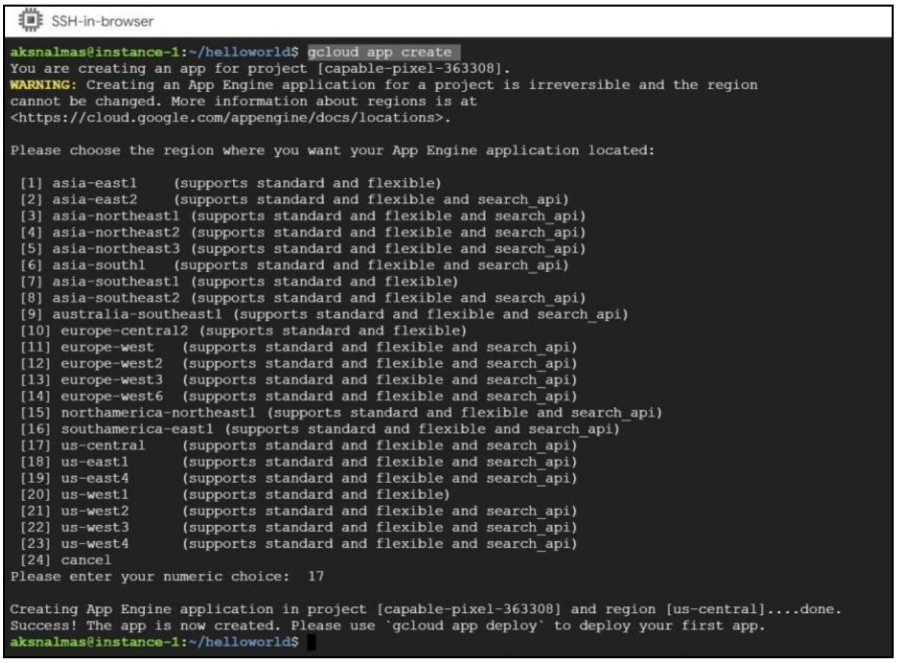


```
aksnalmas@instance-1:~$ gcloud projects list
PROJECT_ID               NAME              PROJECT_NUMBER
capable-pixel-363308     My Project 96667  1063768344572
dulcet-antler-363111     My Project 41544  791787119494
onyx-zodiac-363110       My Project 65186  358507265794
rock-partition-363003    My First Project  895374116447
aksnalmas@instance-1:~$
```

Now change to the directory where your application files are stored

```
cd helloworld
```

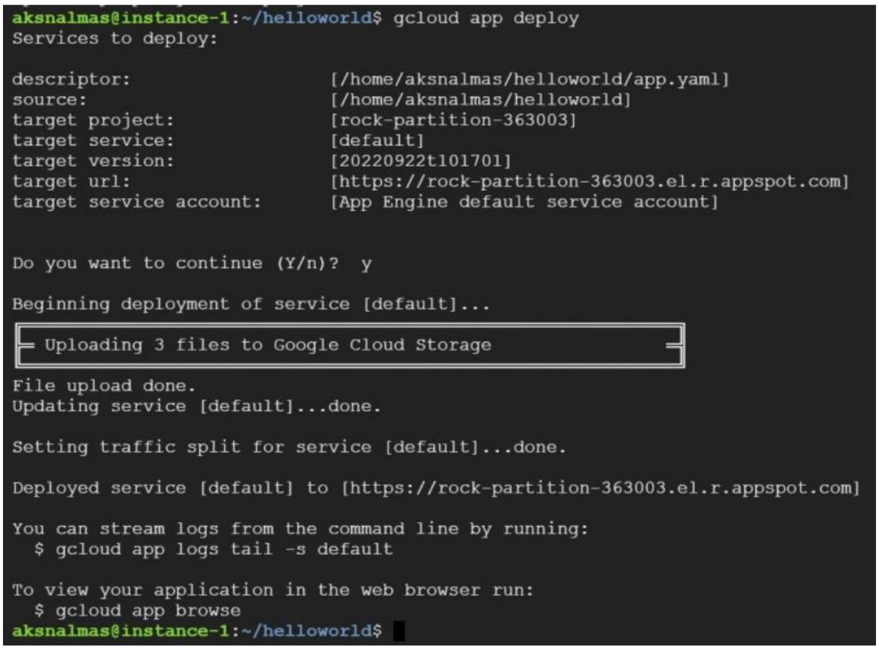
Create app using command **gcloud app create** and select the region that you want to deploy your app on, all the app engine commands start with gcloud <app> , and it starts with gcloud <compute> for compute engine commands.



```
aksnalmas@instance-1:~/helloworld$ gcloud app create
You are creating an app for project [capable-pixel-363308].
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-east1    (supports standard and flexible)
 [2] asia-east2    (supports standard and flexible and search_api)
 [3] asia-northeast1 (supports standard and flexible and search_api)
 [4] asia-northeast2 (supports standard and flexible and search_api)
 [5] asia-northeast3 (supports standard and flexible and search_api)
 [6] asia-south1   (supports standard and flexible and search_api)
 [7] asia-southeast1 (supports standard and flexible)
 [8] asia-southeast2 (supports standard and flexible and search_api)
 [9] australia-southeast1 (supports standard and flexible and search_api)
 [10] europe-central2 (supports standard and flexible)
 [11] europe-west   (supports standard and flexible and search_api)
 [12] europe-west2  (supports standard and flexible and search_api)
 [13] europe-west3  (supports standard and flexible and search_api)
 [14] europe-west6  (supports standard and flexible and search_api)
 [15] northamerica-northeast1 (supports standard and flexible and search_api)
 [16] southamerica-east1 (supports standard and flexible and search_api)
 [17] us-central    (supports standard and flexible and search_api)
 [18] us-east1      (supports standard and flexible and search_api)
 [19] us-east4      (supports standard and flexible and search_api)
 [20] us-west1      (supports standard and flexible)
 [21] us-west2      (supports standard and flexible and search_api)
 [22] us-west3      (supports standard and flexible and search_api)
 [23] us-west4      (supports standard and flexible and search_api)
 [24] cancel
Please enter your numeric choice:  17

Creating App Engine application in project [capable-pixel-363308] and region [us-central]....done.
Success! The app is now created. Please use `gcloud app deploy` to deploy your first app.
aksnalmas@instance-1:~/helloworld$
```

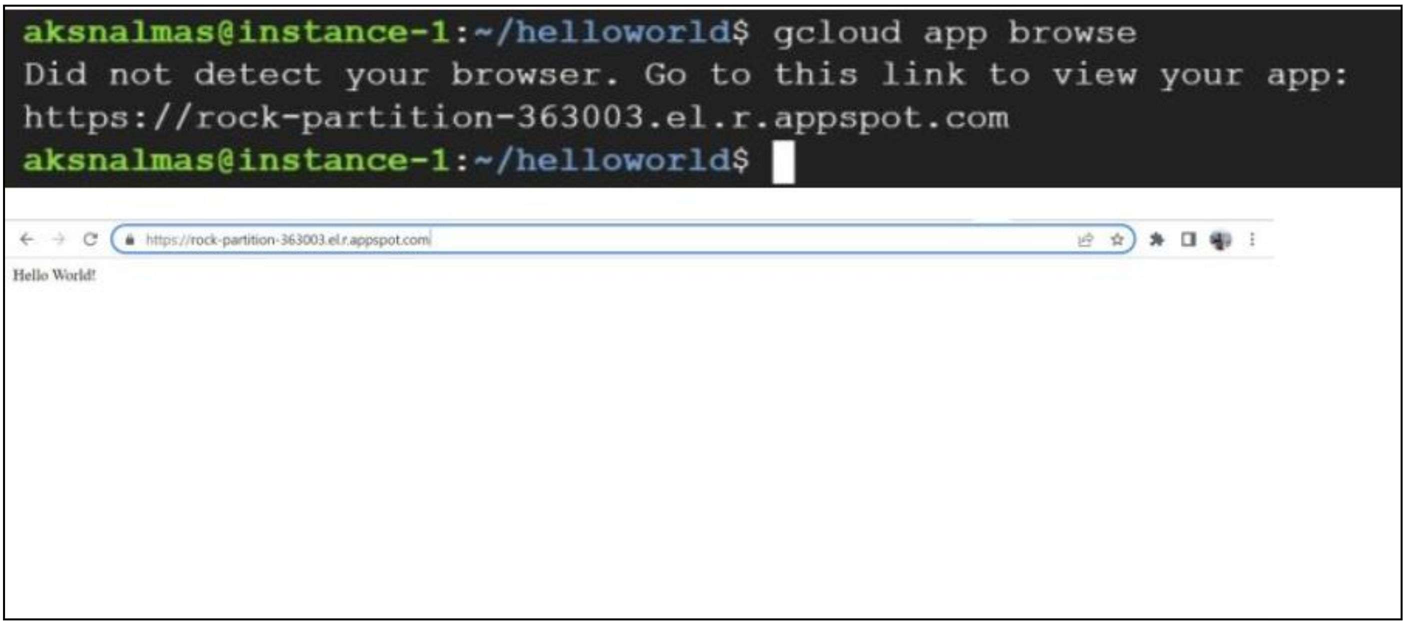
Now run the following command to deploy the application.

```
gcloud app deploy
```



```
aksnalmas@instance-1:~/helloworld$ gcloud app deploy
Services to deploy:

descriptor:                  [/home/aksnalmas/helloworld/app.yaml]
source:                      [/home/aksnalmas/helloworld]
target project:              [rock-partition-363003]
target service:              [default]
target version:              [20220922t101701]
target url:                  [https://rock-partition-363003.el.r.appspot.com]
target service account:      [App Engine default service account]


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

Beginning deployment of service [default]...
╔════════════════════════════════════════════════════════════╗
╠═ Uploading 3 files to Google Cloud Storage                 ═╣
╚════════════════════════════════════════════════════════════╝
File upload done.
Updating service [default]...done.

Setting traffic split for service [default]...done.

Deployed service [default] to [https://rock-partition-363003.el.r.appspot.com]

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

To view your application in the web browser run:
  $ gcloud app browse
aksnalmas@instance-1:~/helloworld$
```

Then use the command **gcloud app browse**, it will give you a link to the application or it will directly open your application on the browser.

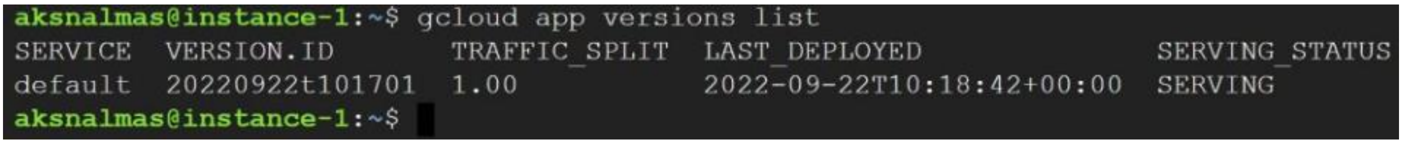


```
aksnalmas@instance-1:~/helloworld$ gcloud app browse
Did not detect your browser. Go to this link to view your app:
https://rock-partition-363003.el.r.appspot.com
aksnalmas@instance-1:~/helloworld$
```

https://rock-partition-363003.el.r.appspot.com

Hello World!

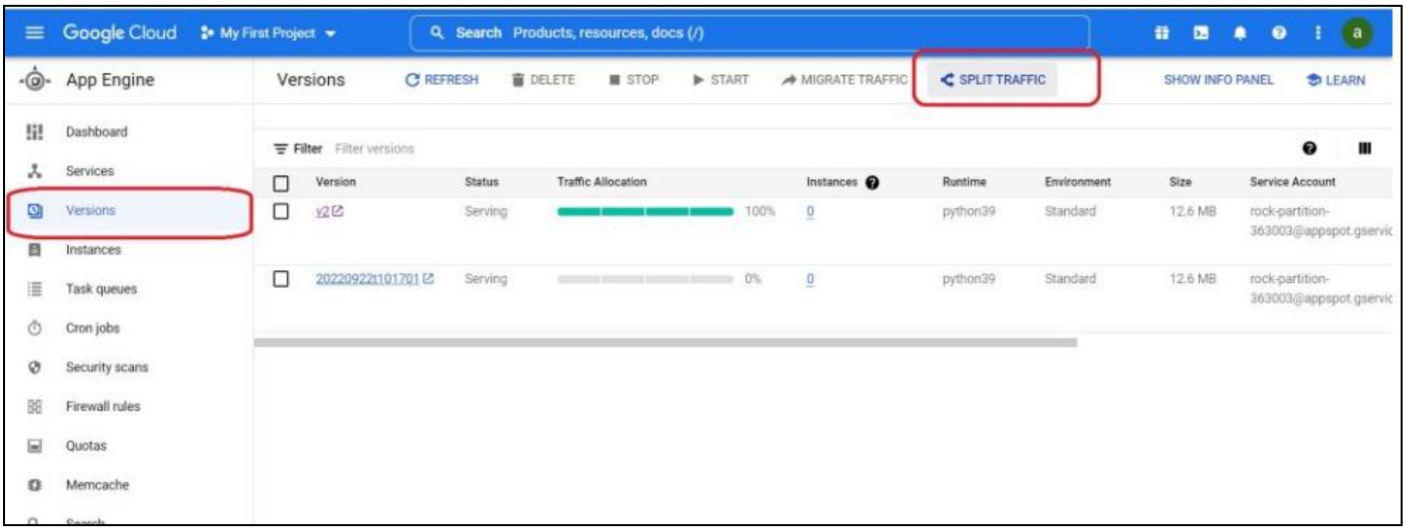
As we successfully deployed the application, let’s see how versioning and traffic splitting works in the app engine, you can list versions by using **gcloud app versions list**



```
aksnalmas@instance-1:~$ gcloud app versions list
SERVICE  VERSION.ID       TRAFFIC_SPLIT  LAST_DEPLOYED              SERVING_STATUS
default  20220922t101701  1.00           2022-09-22T10:18:42+00:00  SERVING
aksnalmas@instance-1:~$ 
```

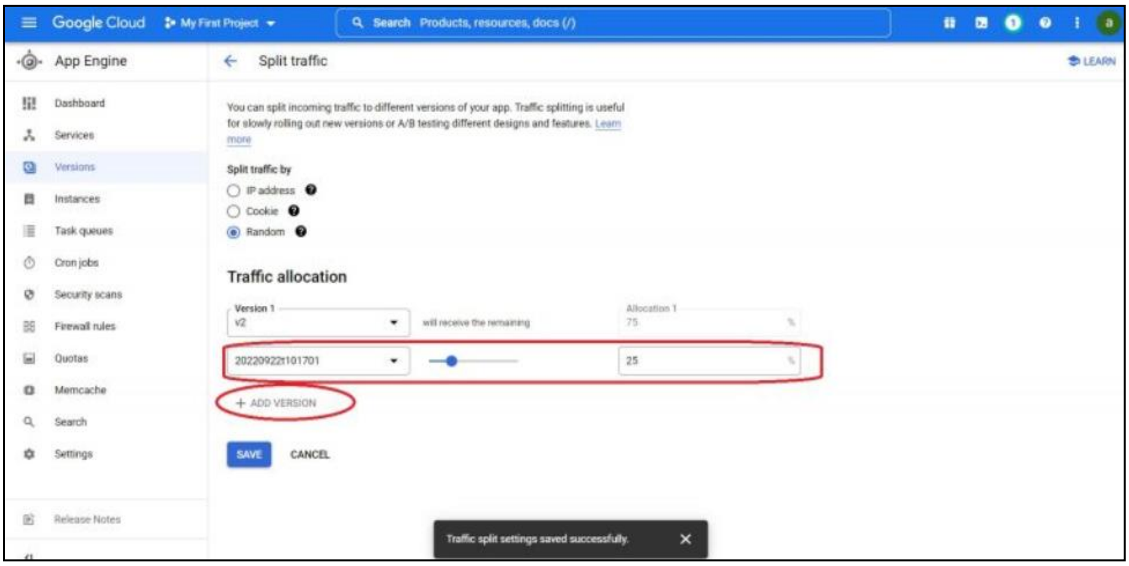
Now deploy the application again but with a version tag this time, in production you may have many changes in each version of the application to be deployed, but here I am deploying the same application with a different version tag.

```
gcloud app deploy --version=v2
```



□	Version	Status	Traffic Allocation	Instances	Runtime	Environment	Size	Service Account
□	v2	Serving	100%	0	python39	Standard	12.6 MB	rock-partition-363003@appspot.gservic
□	20220922t101701	Serving	0%	0	python39	Standard	12.6 MB	rock-partition-363003@appspot.gservic

Click on Add version and select version and traffic weight that you want to divert and save it.



Split traffic

You can split incoming traffic to different versions of your app. Traffic splitting is useful for slowly rolling out new versions or A/B testing different designs and features. Learn more

Split traffic by

○ IP address
○ Cookie
● Random

Traffic allocation

Version 1: v2 — will receive the remaining — Allocation 1: 75 %

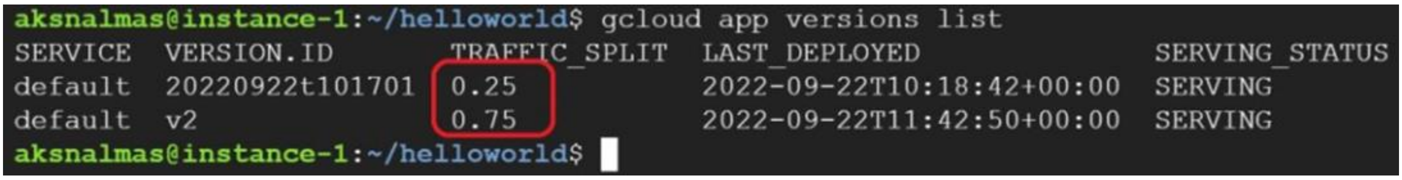
20220922t101701 — 25 %

+ ADD VERSION

SAVE CANCEL

Traffic split settings saved successfully.

On your terminal or web console check for the traffic splitting now,



```
aksnalmas@instance-1:~/helloworld$ gcloud app versions list
SERVICE  VERSION.ID       TRAFFIC_SPLIT  LAST_DEPLOYED              SERVING_STATUS
default  20220922t101701  0.25           2022-09-22T10:18:42+00:00  SERVING
default  v2               0.75           2022-09-22T11:42:50+00:00  SERVING
aksnalmas@instance-1:~/helloworld$
```

GitHub Repo Link: PrathamRaka/Cloud-Computing-Analysis (github.com)

Conclusion:

App engine provides two different kinds of environments Standard and Flexible, use Standard when you want to run the application in pre-configured language-specified boxes where you will only take of your application, it provides rvices only for applications using Java, Python, PHP, Node.js, and Ruby, o. If you want to deploy containers use a Flexible environment, it makes use of compute engine virtual machines, you can use any runtime as long as you can build a docker image for it, it also provides access to background processes in virtual machines.