	Bansilal Ramnath Agarwal Charitable Trust's Vishwakarma Institute of Information Technology **Department of Artificial Intelligence and Data Science**

Student Name: Vidhyapati Umesh Shelke		
Class: TY	Division: A	Roll No: 371069
Semester: 5th	Academic Year: 2022-23	
Subject Name & Code: Cloud Computing		
Title of Assignment: Assignment 7: Deploy web applications on commercial cloud Technology: Google app Engine/ Windows Azure 2.		
Date of Performance: 5/12/2022	Date of Submission: 6/12/2022	

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.

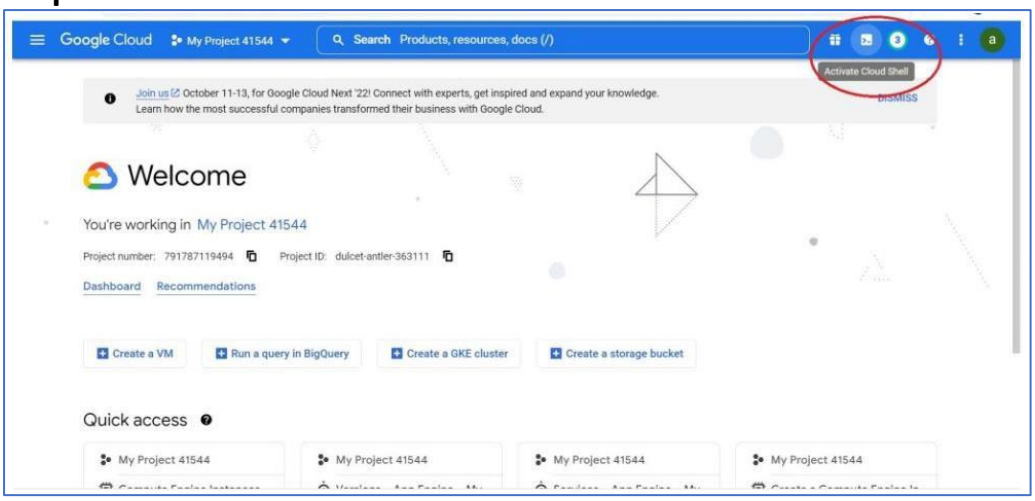
Cloud Resource Requirements: Google App Engine

Background Information:

- **Google App Engine** is a managed service provided by google cloud platform, it provides you 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, 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.

Steps:

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



Step 2: 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
```

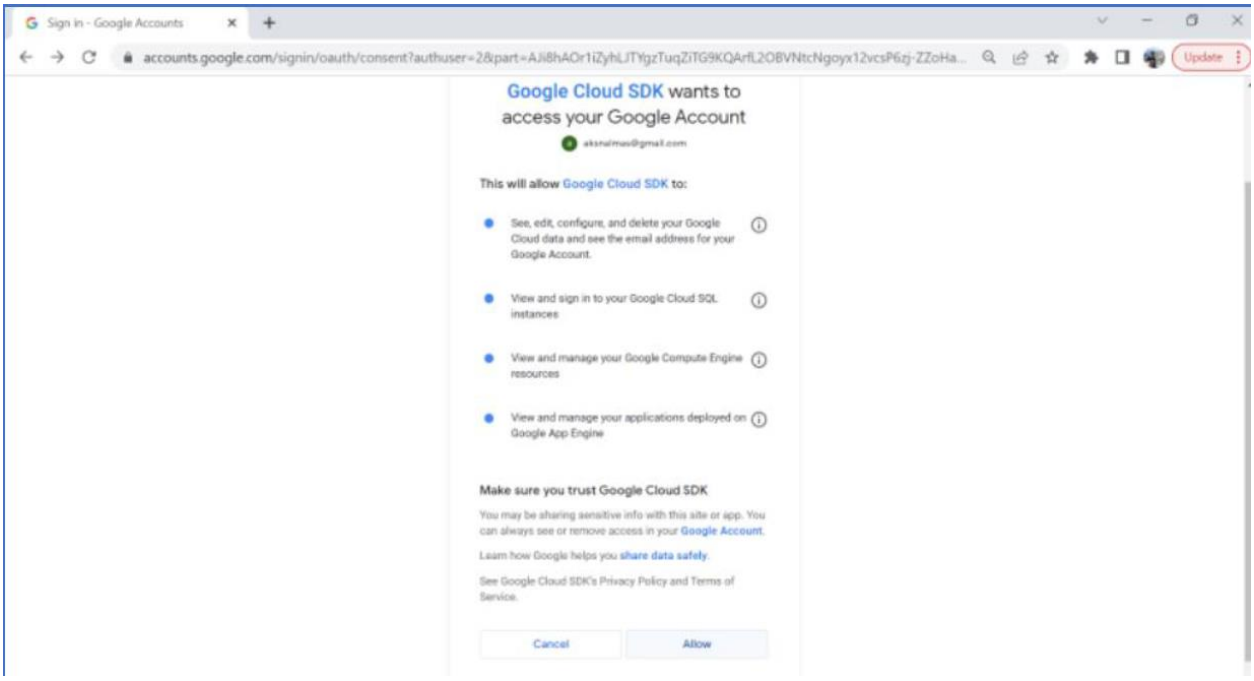
Step 3: Initializing gcloud CLI: then to initialize gcloud CLI run **gcloud init** command on your terminal, Select option **Log in with a new account** when it prompts “choose an account that you would like to initialize with”

```
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 it in your browser and login to your GCP account in order to get 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.googleusercontent.com&redirect_uri=https%3A%2F%2Fsdk.cloud.google.com%2Fauthcode.html&scope=openid+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fsqlservice.login+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.reauth&state=lEz57BAkvXfJkGp5mVRXK2AVAukhF5&prompt=consent&access_type=offline&code_challenge=ts7qRzD2qg1d79Ju8_3LbrVItDRTaq0BgChM0HEG6ms&code_challenge_method=S256

Enter authorization code:
```

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



Then copy the authorization code and give it in your terminal and 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 an 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**

```
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:

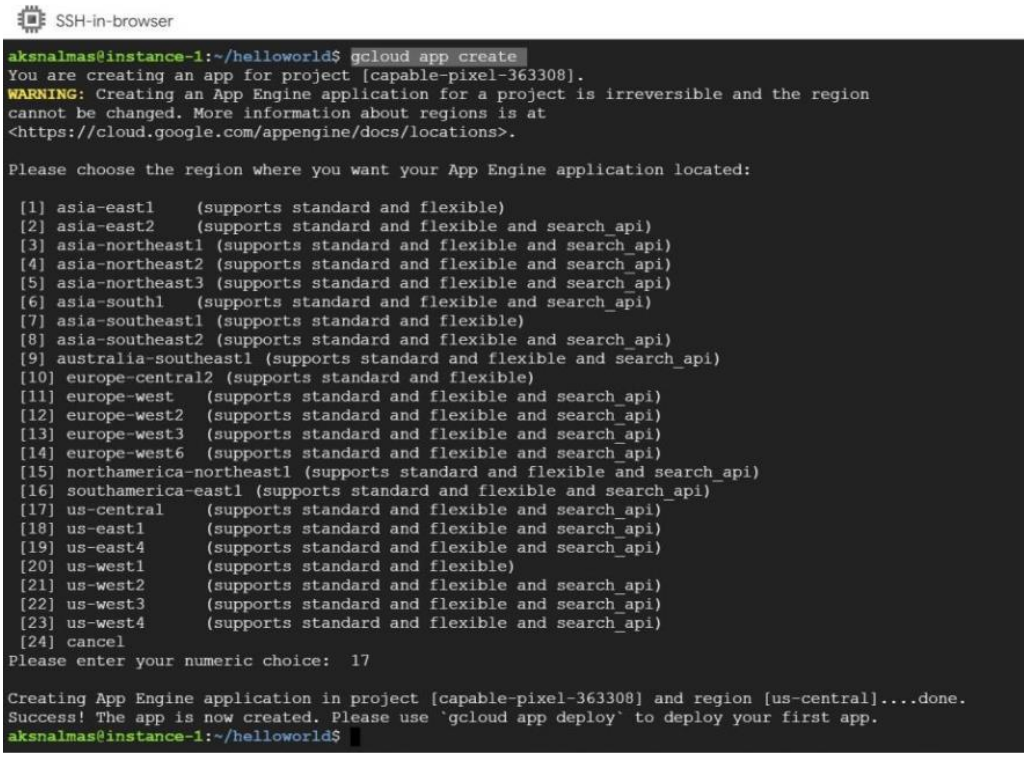
```
runtime: python39
```

Now lets deploy this application on google App Engine, go on to your terminal and set the project that you want to deploy on, use 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:~$
```

Now change to the directory where your application files are stored and 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.



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 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$
```

As we successfully deployed the application, let's see how versioning and traffic splitting works in 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 same deploying same application with a different version tag.

```
gcloud app deploy –version=v2
```

GitHub Repo Link:
https://github.com/Vidhyapati/CCA-Assignments

Conclusion:

App engine provides two different kinds of environments Standard and Flexible, use Standard when you want to run application in pre-configured language specific sandboxes where you will only take of your application. If you want to deploy containers use 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.