Assignment 2, Cloud Application Development

Put all deliverables into github repository in your profile. Defend by explaining deliverables and answering questions.
Deliverables: report in pdf
Google form: https://docs.google.com/forms/d/e/1FAIpQLSe0GyNdOYlvM1tX_I_CtlPod5jBf-ACLGdHYZq1gVZbUeBzIg/viewform?usp=sf_link

Exercise 1: Google App Engine

Objective: Deploy a simple web application on Google App Engine.

Instructions:

1. **Setup**:
    - Ensure you have a Google Cloud account.
    - Install the Google Cloud SDK on your local machine.
2. **Create a Project**:
    - Create a new project in the Google Cloud Console.
3. **Prepare the Application**:
    - Write a simple "Hello, World!" web application using Python (Flask).

Example app.py:

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, World!'

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8080, debug=True)
```

○

4. **Create the App Engine Configuration**:

Create a app.yaml file with the following content:

```
runtime: python39
handlers:
  - url: /.*
    script: auto
```

○

5. **Deploy the Application**:

Use the following command to deploy the application to Google App Engine:

gcloud app deploy

6. **Access the Application**:
   - Once deployed, access your application using the URL provided by Google App Engine.

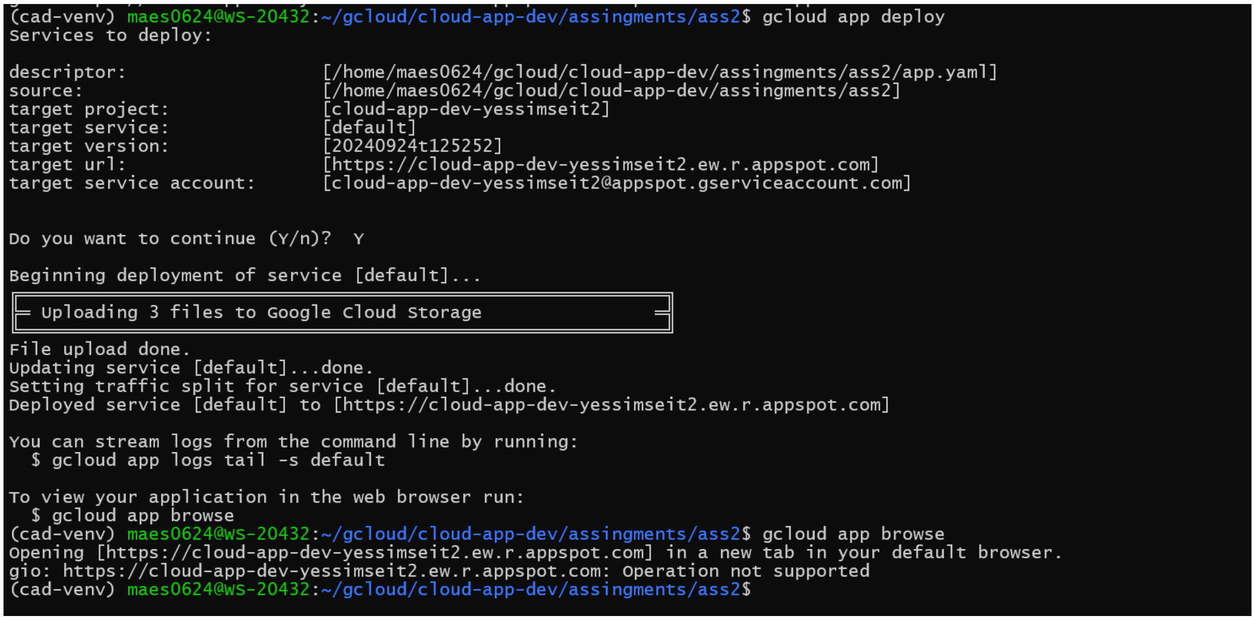
Deliverables:

- A deployed web application on Google App Engine.
- A screenshot of the running application.

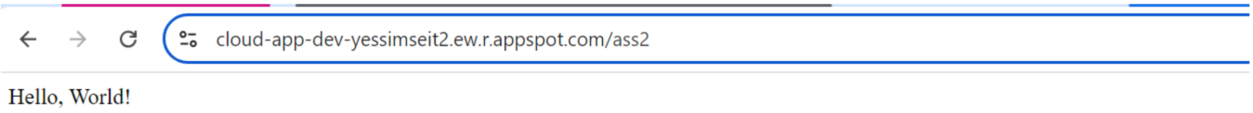
Firstly we create all files by instructions.

Then use command gcloud app deploy.

After command I have url and go to google .

- 

```
(cad-venv) maes0624@WS-20432:~/gcloud/cloud-app-dev/assingments/ass2$ gcloud app deploy
Services to deploy:

descriptor:                  [/home/maes0624/gcloud/cloud-app-dev/assingments/ass2/app.yaml]
source:                      [/home/maes0624/gcloud/cloud-app-dev/assingments/ass2]
target project:              [cloud-app-dev-yessimseit2]
target service:              [default]
target version:              [20240924t125252]
target url:                  [https://cloud-app-dev-yessimseit2.ew.r.appspot.com]
target service account:      [cloud-app-dev-yessimseit2@appspot.gserviceaccount.com]


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://cloud-app-dev-yessimseit2.ew.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
(cad-venv) maes0624@WS-20432:~/gcloud/cloud-app-dev/assingments/ass2$ gcloud app browse
Opening [https://cloud-app-dev-yessimseit2.ew.r.appspot.com] in a new tab in your default browser.
gio: https://cloud-app-dev-yessimseit2.ew.r.appspot.com: Operation not supported
(cad-venv) maes0624@WS-20432:~/gcloud/cloud-app-dev/assingments/ass2$
```



cloud-app-dev-yessimseit2.ew.r.appspot.com/ass2

Hello, World!

Exercise 2: Building with Google Cloud Functions

Objective: Create a Google Cloud Function that processes HTTP requests.

Instructions:

1. **Setup**:
   - Ensure you have a Google Cloud account.
   - Install the Google Cloud SDK on your local machine.
2. **Create a Function**:
   - Create a new Google Cloud Function using the following configuration:
     - **Name**: helloWorldFunction
     - **Trigger**: HTTP
     - **Runtime**: Node.js 18 (or another supported runtime)
     - **Entry Point**: helloWorld
3. **Write the Code**:
   - Write a simple function that returns "Hello, World!" when accessed via HTTP.

Example index.js:

```
exports.helloWorld = (req, res) => {
  res.send('Hello, World!');
};
```

   ○

4. **Deploy the Function**:

Use the following command to deploy the function:

```
gcloud functions deploy helloWorldFunction --runtime nodejs18 --trigger-http
```

5. **Invoke the Function**:

- Once deployed, use the provided URL to test the function by accessing it via a web browser or curl.

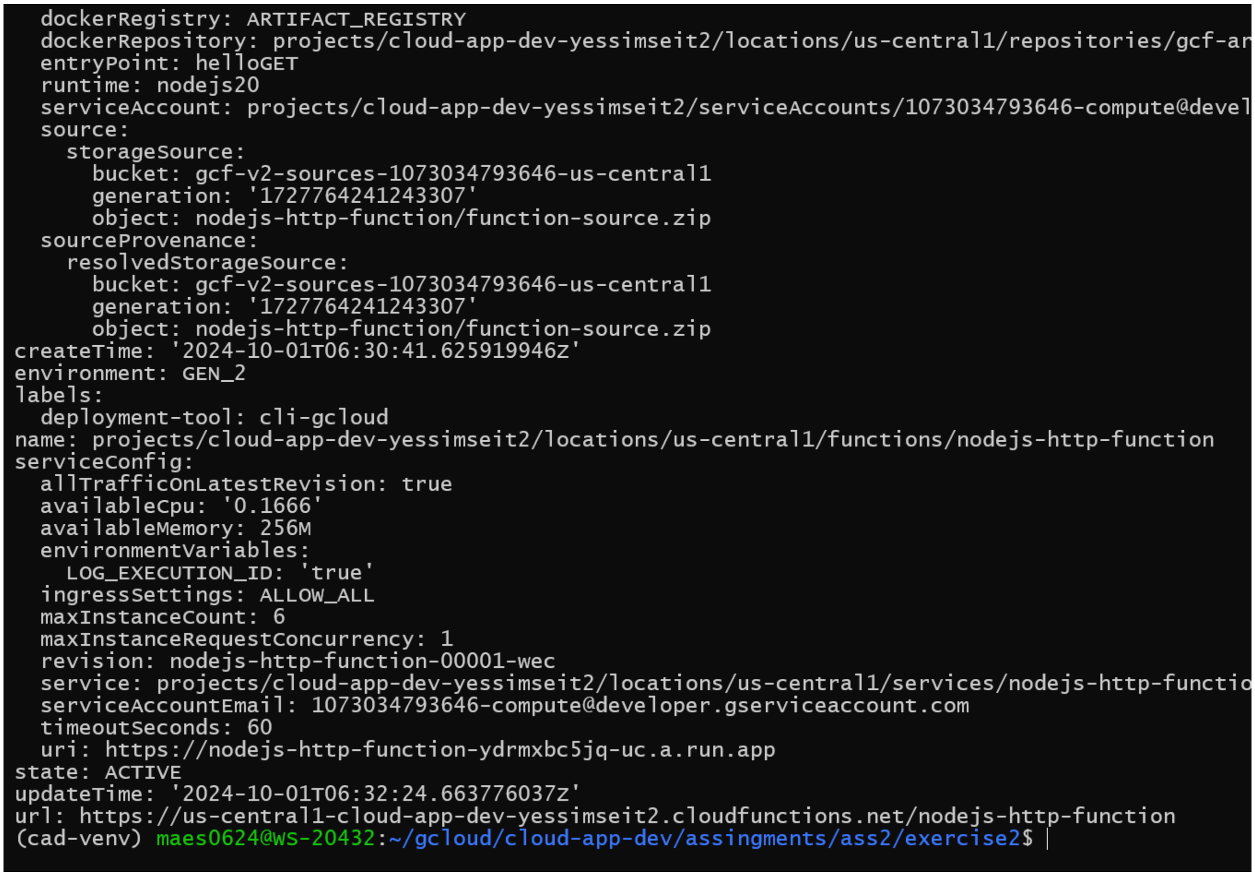
Deliverables:

- A deployed Google Cloud Function.
- A screenshot showing the response from the function.

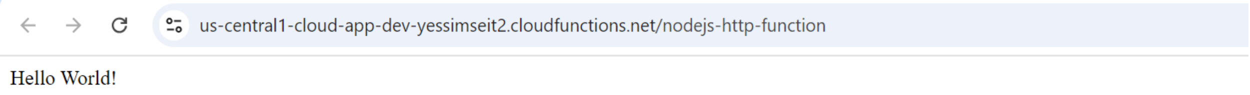
We create index.js and package.json.Then deploy function to google cloud function.

```
gcloud functions deploy nodejs-http-function \
--gen2 \
--runtime=nodejs20 \
--region=us-central1 \
--source=. \
--entry-point=helloGET \
--trigger-http
```

We deployed success.

- 

```
  dockerRegistry: ARTIFACT_REGISTRY
  dockerRepository: projects/cloud-app-dev-yessimseit2/locations/us-central1/repositories/gcf-ar
  entryPoint: helloGET
  runtime: nodejs20
  serviceAccount: projects/cloud-app-dev-yessimseit2/serviceAccounts/1073034793646-compute@devel
  source:
    storageSource:
      bucket: gcf-v2-sources-1073034793646-us-central1
      generation: '1727764241243307'
      object: nodejs-http-function/function-source.zip
  sourceProvenance:
    resolvedStorageSource:
      bucket: gcf-v2-sources-1073034793646-us-central1
      generation: '1727764241243307'
      object: nodejs-http-function/function-source.zip
createTime: '2024-10-01T06:30:41.625919946Z'
environment: GEN_2
labels:
  deployment-tool: cli-gcloud
name: projects/cloud-app-dev-yessimseit2/locations/us-central1/functions/nodejs-http-function
serviceConfig:
  allTrafficOnLatestRevision: true
  availableCpu: '0.1666'
  availableMemory: 256M
  environmentVariables:
    LOG_EXECUTION_ID: 'true'
  ingressSettings: ALLOW_ALL
  maxInstanceCount: 6
  maxInstanceRequestConcurrency: 1
  revision: nodejs-http-function-00001-wec
  service: projects/cloud-app-dev-yessimseit2/locations/us-central1/services/nodejs-http-functio
  serviceAccountEmail: 1073034793646-compute@developer.gserviceaccount.com
  timeoutSeconds: 60
  uri: https://nodejs-http-function-ydrmxbc5jq-uc.a.run.app
state: ACTIVE
updateTime: '2024-10-01T06:32:24.663776037Z'
url: https://us-central1-cloud-app-dev-yessimseit2.cloudfunctions.net/nodejs-http-function
(cad-venv) maes0624@WS-20432:~/gcloud/cloud-app-dev/assingments/ass2/exercise2$ |
```

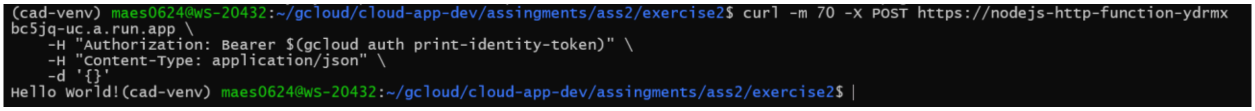


us-central1-cloud-app-dev-yessimseit2.cloudfunctions.net/nodejs-http-function

Hello World!

-

- We use special uri from google



```
(cad-venv) maes0624@WS-20432:~/gcloud/cloud-app-dev/assingments/ass2/exercise2$ curl -m 70 -X POST https://nodejs-http-function-ydrmx
bc5jq-uc.a.run.app \
    -H "Authorization: Bearer $(gcloud auth print-identity-token)" \
    -H "Content-Type: application/json" \
    -d '{}'
Hello World!(cad-venv) maes0624@WS-20432:~/gcloud/cloud-app-dev/assingments/ass2/exercise2$ |
```

Exercise 3: Containerizing Applications

Objective: Containerize a simple application using Docker.

Instructions:

1. **Setup**:
    - ○ Ensure Docker is installed on your local machine.
2. **Create a Simple Application**:
    - ○ Write a simple Python application.

Example app.py:

```
print("Hello from inside the container!")
```

○

3. **Create a Dockerfile**:
    - ○ Write a Dockerfile to containerize the application.

Example Dockerfile:

```
# Use an official Python runtime as a parent image
FROM python:3.9-slim

# Set the working directory in the container
WORKDIR /app

# Copy the current directory contents into the container at /app
COPY . /app

# Run the application
CMD ["python", "app.py"]
```

4. **Build the Docker Image**:

Build the Docker image using the following command:

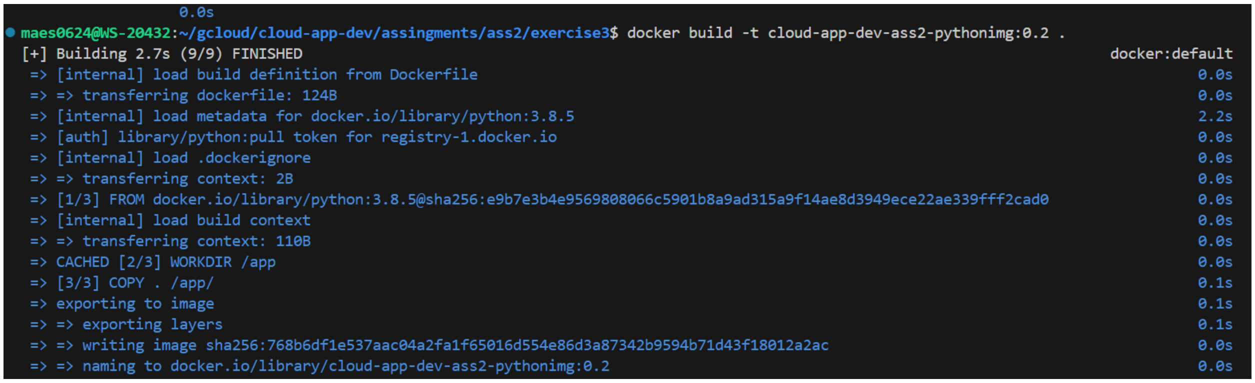
docker build -t hello-world-app .

5. **Run the Docker Container**:

Run the container using the following command:
docker run --rm hello-world-app

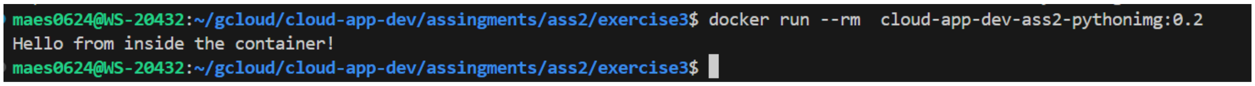
Deliverables:

- A Docker image that runs a simple application.
- A screenshot of the container output showing "Hello from inside the container!"
-



```
                    0.0s
● maes0624@WS-20432:~/gcloud/cloud-app-dev/assingments/ass2/exercise3$ docker build -t cloud-app-dev-ass2-pythonimg:0.2 .
 [+] Building 2.7s (9/9) FINISHED                                                                        docker:default
  => [internal] load build definition from Dockerfile                                                             0.0s
  => => transferring dockerfile: 124B                                                                             0.0s
  => [internal] load metadata for docker.io/library/python:3.8.5                                                  2.2s
  => [auth] library/python:pull token for registry-1.docker.io                                                    0.0s
  => [internal] load .dockerignore                                                                                0.0s
  => => transferring context: 2B                                                                                  0.0s
  => [1/3] FROM docker.io/library/python:3.8.5@sha256:e9b7e3b4e9569808066c5901b8a9ad315a9f14ae8d3949ece22ae339fff2cad0   0.0s
  => [internal] load build context                                                                                0.0s
  => => transferring context: 110B                                                                                0.0s
  => CACHED [2/3] WORKDIR /app                                                                                    0.0s
  => [3/3] COPY . /app/                                                                                           0.1s
  => exporting to image                                                                                           0.1s
  => => exporting layers                                                                                          0.1s
  => => writing image sha256:768b6df1e537aac04a2fa1f65016d554e86d3a87342b9594b71d43f18012a2ac                     0.0s
  => => naming to docker.io/library/cloud-app-dev-ass2-pythonimg:0.2                                              0.0s
```

-



```
maes0624@WS-20432:~/gcloud/cloud-app-dev/assingments/ass2/exercise3$ docker run --rm  cloud-app-dev-ass2-pythonimg:0.2
Hello from inside the container!
maes0624@WS-20432:~/gcloud/cloud-app-dev/assingments/ass2/exercise3$ █
```