Assignment 2, Cloud Application Development

Dony by: Kabdrakhmanov Altair, 21B030829

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.

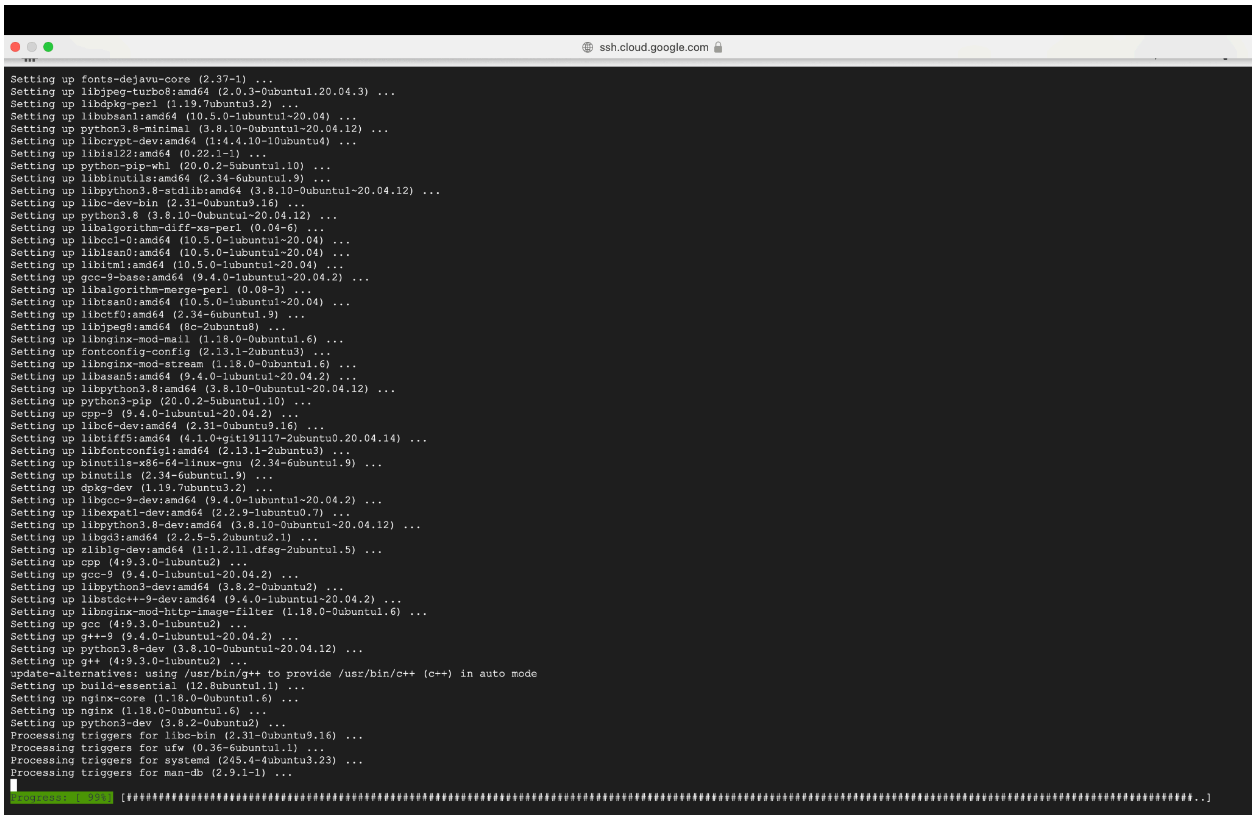
gcloud app deploy screenshot

Running application on internet

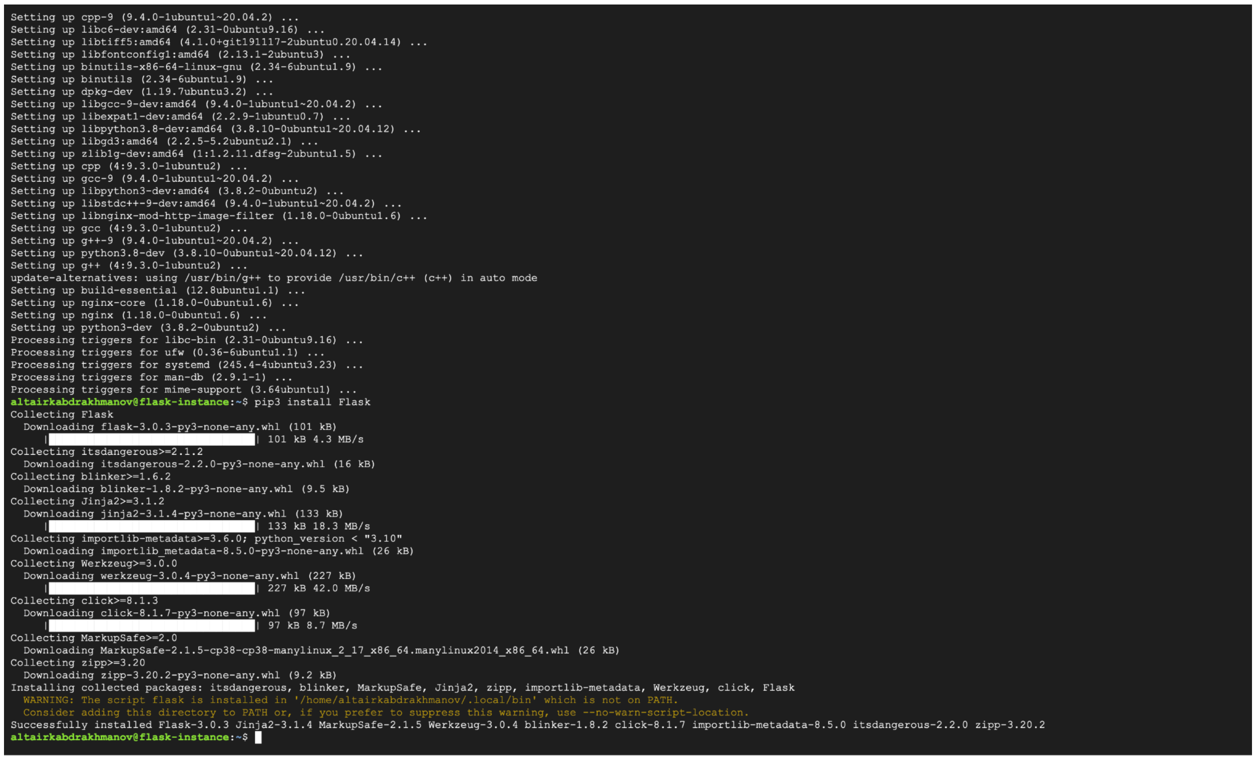
My steps:

- Create new project
- Enable Compute Engine in API&Services
- Create VM instance e-micro with allowed HTTP and HTTPS traffic

- Install python3, and pip3 install Flask



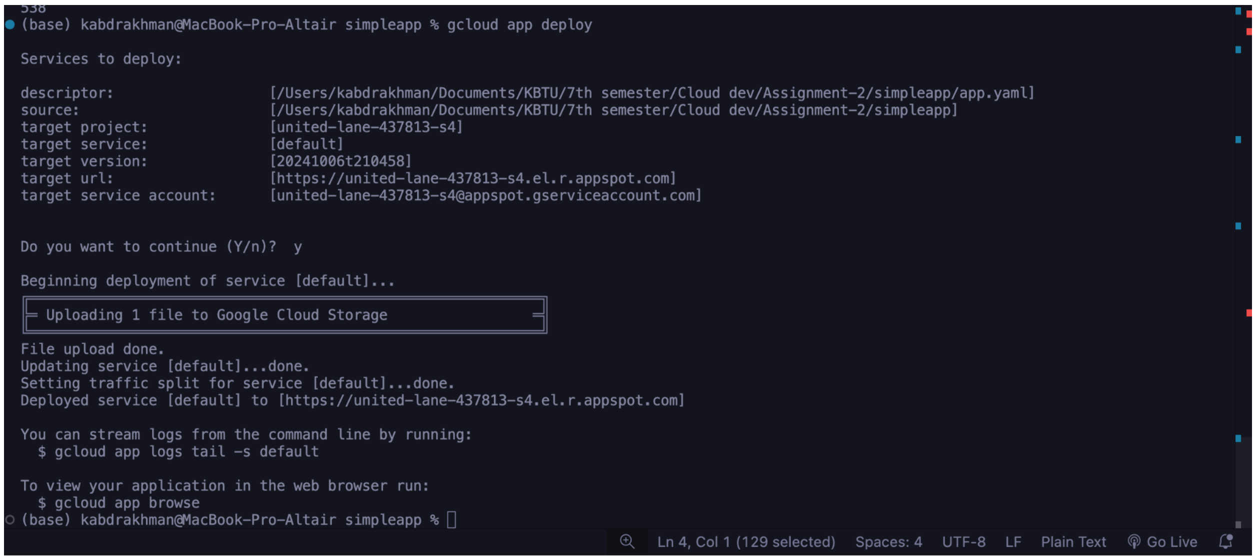
```
Setting up fonts-dejavu-core (2.37-1) ...
Setting up libjpeg-turbo8:amd64 (2.0.3-0ubuntu1.20.04.3) ...
Setting up libdpkg-perl (1.19.7ubuntu3.2) ...
Setting up libubsan1:amd64 (10.5.0-1ubuntu1~20.04) ...
Setting up python3.8-minimal (3.8.10-0ubuntu1~20.04.12) ...
Setting up libcrypt-dev:amd64 (1:4.4.10-10ubuntu4) ...
Setting up libisl22:amd64 (0.22.1-1) ...
Setting up python-pip-whl (20.0.2-5ubuntu1.10) ...
Setting up libbinutils:amd64 (2.34-6ubuntu1.9) ...
Setting up libpython3.8-stdlib:amd64 (3.8.10-0ubuntu1~20.04.12) ...
Setting up libc-dev-bin (2.31-0ubuntu9.16) ...
Setting up python3.8 (3.8.10-0ubuntu1~20.04.12) ...
Setting up libalgorithm-diff-xs-perl (0.04-6) ...
Setting up libcc1-0:amd64 (10.5.0-1ubuntu1~20.04) ...
Setting up liblsan0:amd64 (10.5.0-1ubuntu1~20.04) ...
Setting up libitm1:amd64 (10.5.0-1ubuntu1~20.04) ...
Setting up gcc-9-base:amd64 (9.4.0-1ubuntu1~20.04.2) ...
Setting up libalgorithm-merge-perl (0.08-3) ...
Setting up libtsan0:amd64 (10.5.0-1ubuntu1~20.04) ...
Setting up libctf0:amd64 (2.34-6ubuntu1.9) ...
Setting up libjpeg8:amd64 (8c-2ubuntu8) ...
Setting up libnginx-mod-mail (1.18.0-0ubuntu1.6) ...
Setting up fontconfig-config (2.13.1-2ubuntu3) ...
Setting up libnginx-mod-stream (1.18.0-0ubuntu1.6) ...
Setting up libasan5:amd64 (9.4.0-1ubuntu1~20.04.2) ...
Setting up libpython3.8:amd64 (3.8.10-0ubuntu1~20.04.12) ...
Setting up python3-pip (20.0.2-5ubuntu1.10) ...
Setting up cpp-9 (9.4.0-1ubuntu1~20.04.2) ...
Setting up libc6-dev:amd64 (2.31-0ubuntu9.16) ...
Setting up libtiff5:amd64 (4.1.0+git191117-2ubuntu0.20.04.14) ...
Setting up libfontconfig1:amd64 (2.13.1-2ubuntu3) ...
Setting up binutils-x86-64-linux-gnu (2.34-6ubuntu1.9) ...
Setting up binutils (2.34-6ubuntu1.9) ...
Setting up dpkg-dev (1.19.7ubuntu3.2) ...
Setting up libgcc-9-dev:amd64 (9.4.0-1ubuntu1~20.04.2) ...
Setting up libexpat1-dev:amd64 (2.2.9-1ubuntu0.7) ...
Setting up libpython3.8-dev:amd64 (3.8.10-0ubuntu1~20.04.12) ...
Setting up libgd3:amd64 (2.2.5-5.2ubuntu2.1) ...
Setting up zlib1g-dev:amd64 (1:1.2.11.dfsg-2ubuntu1.5) ...
Setting up cpp (4:9.3.0-1ubuntu2) ...
Setting up gcc-9 (9.4.0-1ubuntu1~20.04.2) ...
Setting up libpython3-dev:amd64 (3.8.2-0ubuntu2) ...
Setting up libstdc++-9-dev:amd64 (9.4.0-1ubuntu1~20.04.2) ...
Setting up libnginx-mod-http-image-filter (1.18.0-0ubuntu1.6) ...
Setting up gcc (4:9.3.0-1ubuntu2) ...
Setting up g++-9 (9.4.0-1ubuntu1~20.04.2) ...
Setting up python3.8-dev (3.8.10-0ubuntu1~20.04.12) ...
Setting up g++ (4:9.3.0-1ubuntu2) ...
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mode
Setting up build-essential (12.8ubuntu1.1) ...
Setting up nginx-core (1.18.0-0ubuntu1.6) ...
Setting up nginx (1.18.0-0ubuntu1.6) ...
Setting up python3-dev (3.8.2-0ubuntu2) ...
Processing triggers for libc-bin (2.31-0ubuntu9.16) ...
Processing triggers for ufw (0.36-6ubuntu1.1) ...
Processing triggers for systemd (245.4-4ubuntu3.23) ...
Processing triggers for man-db (2.9.1-1) ...

Progress: [ 99%] [###########################################################################################################################################################..]
```



```
Setting up cpp-9 (9.4.0-1ubuntu1~20.04.2) ...
Setting up libc6-dev:amd64 (2.31-0ubuntu9.16) ...
Setting up libtiff5:amd64 (4.1.0+git191117-2ubuntu0.20.04.14) ...
Setting up libfontconfig1:amd64 (2.13.1-2ubuntu3) ...
Setting up binutils-x86-64-linux-gnu (2.34-6ubuntu1.9) ...
Setting up binutils (2.34-6ubuntu1.9) ...
Setting up dpkg-dev (1.19.7ubuntu3.2) ...
Setting up libgcc-9-dev:amd64 (9.4.0-1ubuntu1~20.04.2) ...
Setting up libexpat1-dev:amd64 (2.2.9-1ubuntu0.7) ...
Setting up libpython3.8-dev:amd64 (3.8.10-0ubuntu1~20.04.12) ...
Setting up libgd3:amd64 (2.2.5-5.2ubuntu2.1) ...
Setting up zlib1g-dev:amd64 (1:1.2.11.dfsg-2ubuntu1.5) ...
Setting up cpp (4:9.3.0-1ubuntu2) ...
Setting up gcc-9 (9.4.0-1ubuntu1~20.04.2) ...
Setting up libpython3-dev:amd64 (3.8.2-0ubuntu2) ...
Setting up libstdc++-9-dev:amd64 (9.4.0-1ubuntu1~20.04.2) ...
Setting up libnginx-mod-http-image-filter (1.18.0-0ubuntu1.6) ...
Setting up gcc (4:9.3.0-1ubuntu2) ...
Setting up g++-9 (9.4.0-1ubuntu1~20.04.2) ...
Setting up python3.8-dev (3.8.10-0ubuntu1~20.04.12) ...
Setting up g++ (4:9.3.0-1ubuntu2) ...
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mode
Setting up build-essential (12.8ubuntu1.1) ...
Setting up nginx-core (1.18.0-0ubuntu1.6) ...
Setting up nginx (1.18.0-0ubuntu1.6) ...
Setting up python3-dev (3.8.2-0ubuntu2) ...
Processing triggers for libc-bin (2.31-0ubuntu9.16) ...
Processing triggers for ufw (0.36-6ubuntu1.1) ...
Processing triggers for systemd (245.4-4ubuntu3.23) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for mime-support (3.64ubuntu1) ...
altairkabdrakhmanov@flask-instance:~$ pip3 install Flask
Collecting Flask
  Downloading flask-3.0.3-py3-none-any.whl (101 kB)
     |████████████████████████████████| 101 kB 4.3 MB/s
Collecting itsdangerous>=2.1.2
  Downloading itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Collecting blinker>=1.6.2
  Downloading blinker-1.8.2-py3-none-any.whl (9.5 kB)
Collecting Jinja2>=3.1.2
  Downloading jinja2-3.1.4-py3-none-any.whl (133 kB)
     |████████████████████████████████| 133 kB 18.3 MB/s
Collecting importlib-metadata>=3.6.0; python_version < "3.10"
  Downloading importlib_metadata-8.5.0-py3-none-any.whl (26 kB)
Collecting Werkzeug>=3.0.0
  Downloading werkzeug-3.0.4-py3-none-any.whl (227 kB)
     |████████████████████████████████| 227 kB 42.0 MB/s
Collecting click>=8.1.3
  Downloading click-8.1.7-py3-none-any.whl (97 kB)
     |████████████████████████████████| 97 kB 8.7 MB/s
Collecting MarkupSafe>=2.0
  Downloading MarkupSafe-2.1.5-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (26 kB)
Collecting zipp>=3.20
  Downloading zipp-3.20.2-py3-none-any.whl (9.2 kB)
Installing collected packages: itsdangerous, blinker, MarkupSafe, Jinja2, zipp, importlib-metadata, Werkzeug, click, Flask
  WARNING: The script flask is installed in '/home/altairkabdrakhmanov/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed Flask-3.0.3 Jinja2-3.1.4 MarkupSafe-2.1.5 Werkzeug-3.0.4 blinker-1.8.2 click-8.1.7 importlib-metadata-8.5.0 itsdangerous-2.2.0 zipp-3.20.2
altairkabdrakhmanov@flask-instance:~$
```

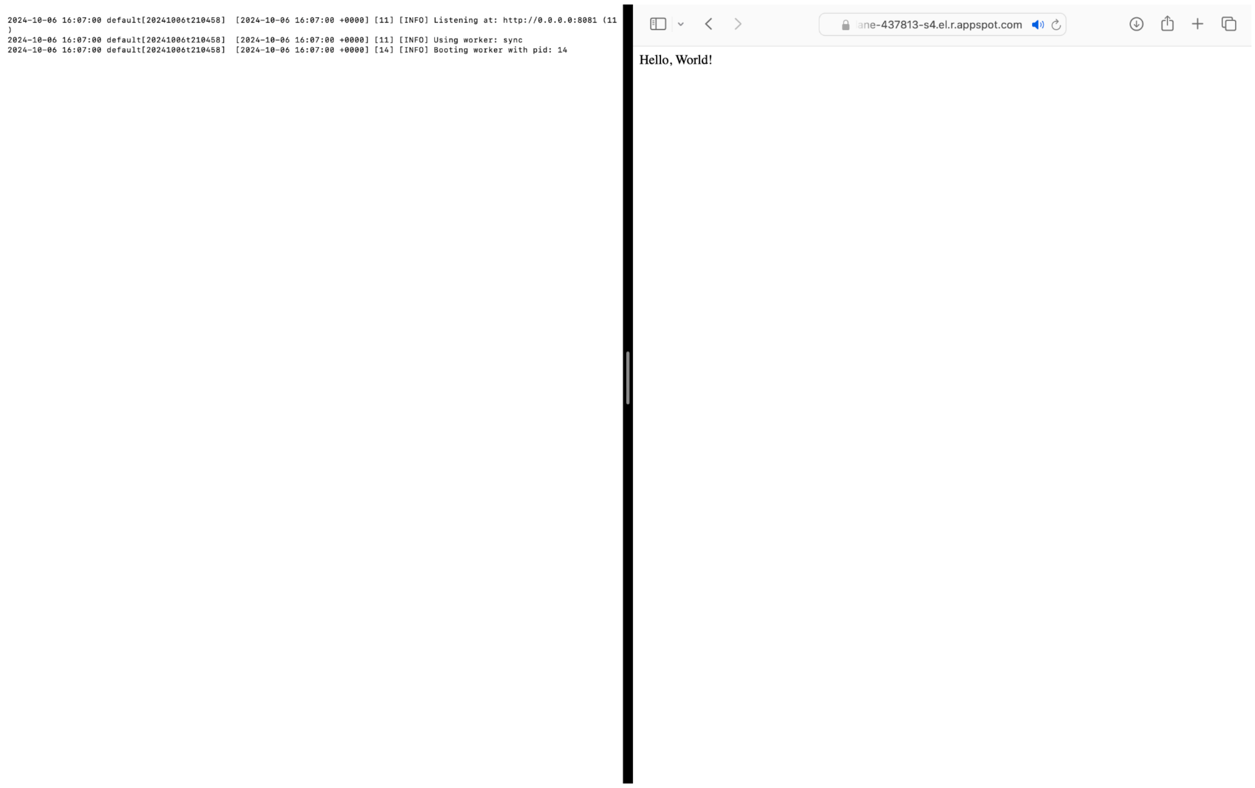
- On local machine I create folder “myapp” with app.py, app.yaml and requirements.txt
- The requirements.txt has the following:

    - Flask==2.2.3
    - gunicorn==20.1.0
    - Werkzeug==2.2.2
- The I applied gcloud app deploy



```
538
(base) kabdrakhman@MacBook-Pro-Altair simpleapp % gcloud app deploy

Services to deploy:

descriptor:                  [/Users/kabdrakhman/Documents/KBTU/7th semester/Cloud dev/Assignment-2/simpleapp/app.yaml]
source:                      [/Users/kabdrakhman/Documents/KBTU/7th semester/Cloud dev/Assignment-2/simpleapp]
target project:              [united-lane-437813-s4]
target service:              [default]
target version:              [20241006t210458]
target url:                  [https://united-lane-437813-s4.el.r.appspot.com]
target service account:      [united-lane-437813-s4@appspot.gserviceaccount.com]


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

Beginning deployment of service [default]...
╔════════════════════════════════════════════════════════════╗
╠═ Uploading 1 file to Google Cloud Storage                 ═╣
╚════════════════════════════════════════════════════════════╝
File upload done.
Updating service [default]...done.
Setting traffic split for service [default]...done.
Deployed service [default] to [https://united-lane-437813-s4.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
(base) kabdrakhman@MacBook-Pro-Altair simpleapp %
```

- Have “Hello, World!” on https://united-lane-437813-s4.el.r.appspot.com/



```
2024-10-06 16:07:00 default[20241006t210458]  [2024-10-06 16:07:00 +0000] [11] [INFO] Listening at: http://0.0.0.0:8081 (11
)
2024-10-06 16:07:00 default[20241006t210458]  [2024-10-06 16:07:00 +0000] [11] [INFO] Using worker: sync
2024-10-06 16:07:00 default[20241006t210458]  [2024-10-06 16:07:00 +0000] [14] [INFO] Booting worker with pid: 14
```

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`.

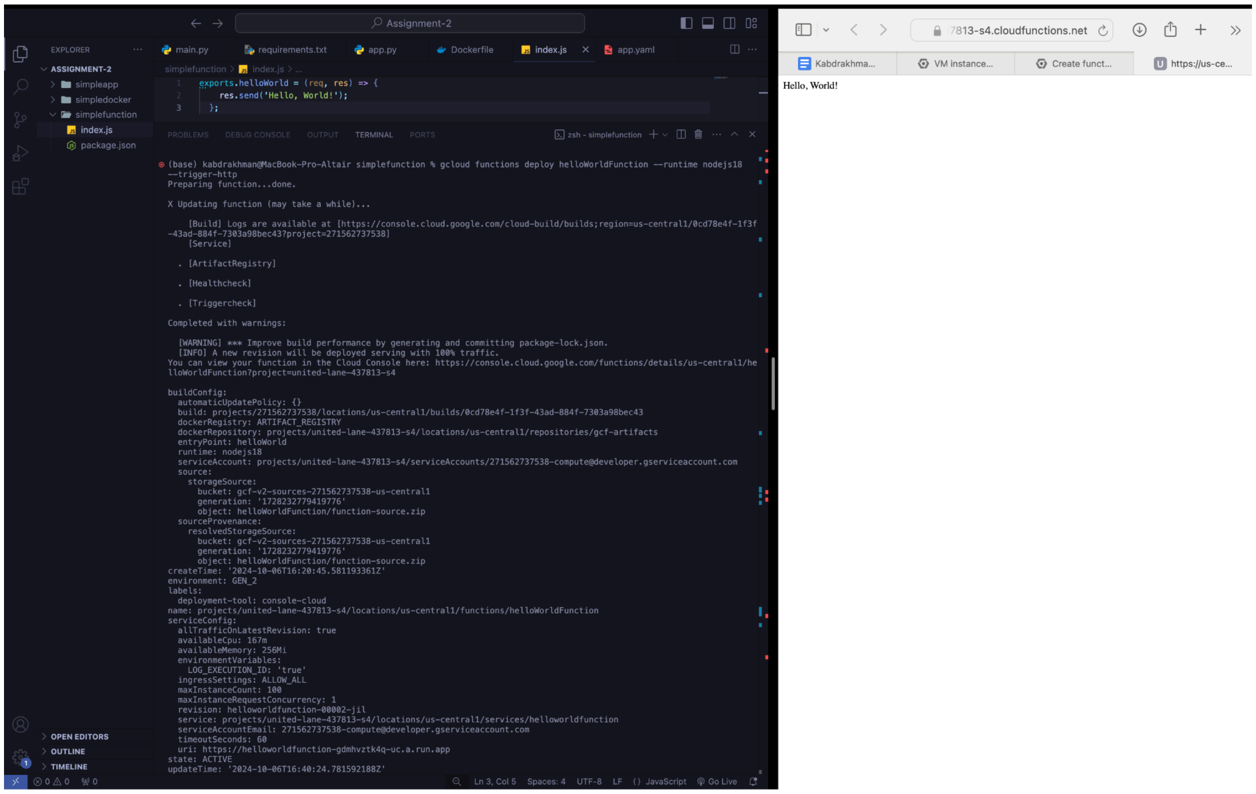
Deliverables:

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

My steps:

- Enabled cloud run functions
- Then create index.js with given code

- Then applied npm init -y
- Applied this command: `gcloud functions deploy helloWorldFunction --runtime nodejs18 --trigger-http`
- Make curl request to this url https://us-central1-united-lane-437813-s4.cloudfunctions.net/helloWorldFunction
- Screenshot



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

```
(base) kabdrakhman@MacBook-Pro-Altair simplefunction % gcloud functions deploy helloWorldFunction --runtime nodejs18
--trigger-http
Preparing function...done.

X Updating function (may take a while)...

    [Build] Logs are available at [https://console.cloud.google.com/cloud-build/builds;region=us-central1/0cd78e4f-1f3f
-43ad-884f-7303a98bec43?project=271562737538]
    [Service]

  . [ArtifactRegistry]

  . [Healthcheck]

  . [Triggercheck]

Completed with warnings:

  [WARNING] *** Improve build performance by generating and committing package-lock.json.
  [INFO] A new revision will be deployed serving with 100% traffic.
You can view your function in the Cloud Console here: https://console.cloud.google.com/functions/details/us-central1/he
lloWorldFunction?project=united-lane-437813-s4

buildConfig:
  automaticUpdatePolicy: {}
  build: projects/271562737538/locations/us-central1/builds/0cd78e4f-1f3f-43ad-884f-7303a98bec43
  dockerRegistry: ARTIFACT_REGISTRY
  dockerRepository: projects/united-lane-437813-s4/locations/us-central1/repositories/gcf-artifacts
  entryPoint: helloWorld
  runtime: nodejs18
  serviceAccount: projects/united-lane-437813-s4/serviceAccounts/271562737538-compute@developer.gserviceaccount.com
  source:
    storageSource:
      bucket: gcf-v2-sources-271562737538-us-central1
      generation: '1728232779419776'
      object: helloWorldFunction/function-source.zip
  sourceProvenance:
    resolvedStorageSource:
      bucket: gcf-v2-sources-271562737538-us-central1
      generation: '1728232779419776'
      object: helloWorldFunction/function-source.zip
createTime: '2024-10-06T16:20:45.581193361Z'
environment: GEN_2
labels:
  deployment-tool: console-cloud
name: projects/united-lane-437813-s4/locations/us-central1/functions/helloWorldFunction
serviceConfig:
  allTrafficOnLatestRevision: true
  availableCpu: 167m
  availableMemory: 256Mi
  environmentVariables:
    LOG_EXECUTION_ID: 'true'
  ingressSettings: ALLOW_ALL
  maxInstanceCount: 100
  maxInstanceRequestConcurrency: 1
  revision: helloworldfunction-00002-jil
  service: projects/united-lane-437813-s4/locations/us-central1/services/helloworldfunction
  serviceAccountEmail: 271562737538-compute@developer.gserviceaccount.com
  timeoutSeconds: 60
  uri: https://helloworldfunction-gdmhvztk4q-uc.a.run.app
state: ACTIVE
updateTime: '2024-10-06T16:40:24.781592188Z'
```

Hello, World!

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`:

```python
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
```

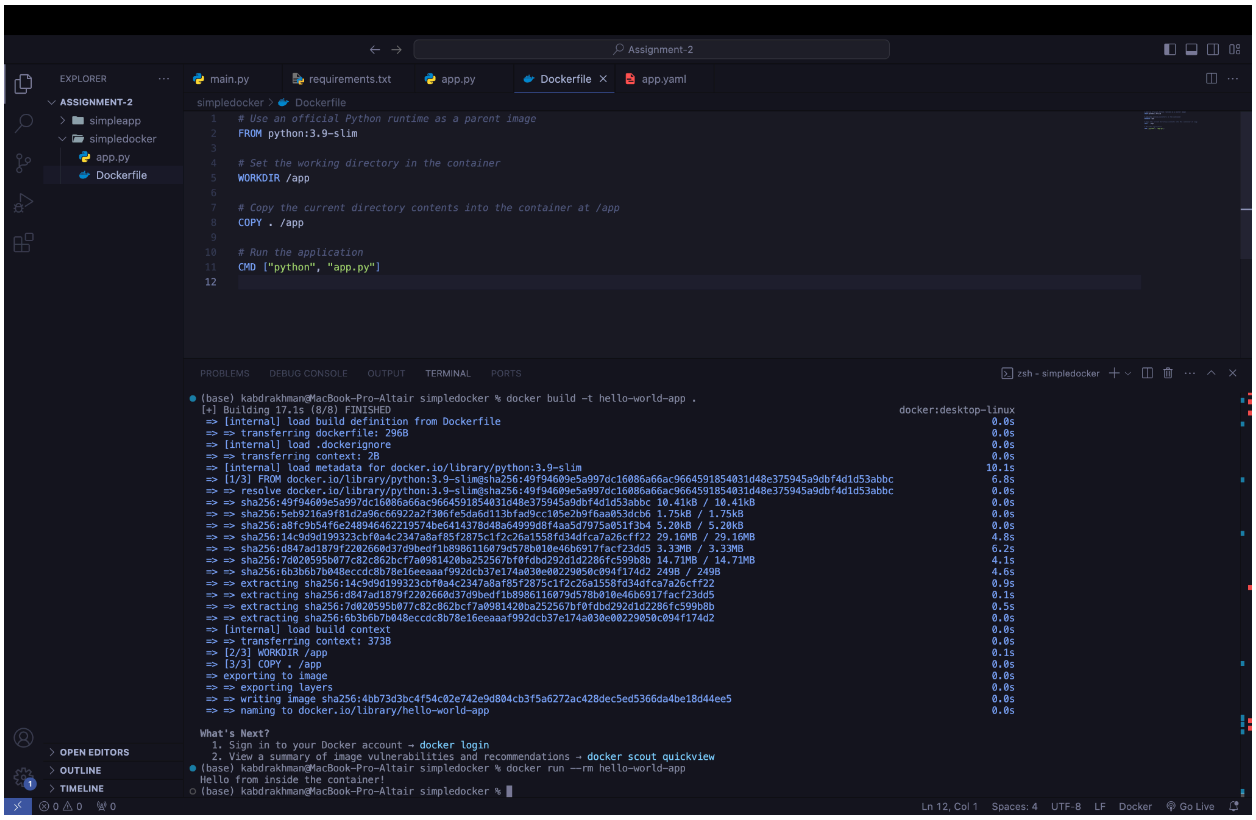
Deliverables:

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

My steps:

- Created simpledocker folder with app.py and Dockerfle
- Write the given code

- Run the command to build image: `docker build -t hello-world-app .`
- Run the command to run container: `docker run --rm hello-world-app`
- `The screenshot`



```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"]

```

```
(base) kabdrakhman@MacBook-Pro-Altair simpledocker % docker build -t hello-world-app .
[+] Building 17.1s (8/8) FINISHED                                                                                        docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                                                     0.0s
 => => transferring dockerfile: 296B                                                                                                     0.0s
 => [internal] load .dockerignore                                                                                                        0.0s
 => => transferring context: 2B                                                                                                          0.0s
 => [internal] load metadata for docker.io/library/python:3.9-slim                                                                      10.1s
 => [1/3] FROM docker.io/library/python:3.9-slim@sha256:49f94609e5a997dc16086a66ac9664591854031d48e375945a9dbf4d1d53abbc                 6.8s
 => => resolve docker.io/library/python:3.9-slim@sha256:49f94609e5a997dc16086a66ac9664591854031d48e375945a9dbf4d1d53abbc                 0.0s
 => => sha256:49f94609e5a997dc16086a66ac9664591854031d48e375945a9dbf4d1d53abbc 10.41kB / 10.41kB                                         0.0s
 => => sha256:5eb9216a9f81d2a96c66922a2f306fe5da6d113bfad9cc105e2b9f6aa053dcb6 1.75kB / 1.75kB                                           0.0s
 => => sha256:a8fc9b54f6e248946462219574be6414378d48a64999d8f4aa5d7975a051f3b4 5.20kB / 5.20kB                                           0.0s
 => => sha256:14c9d9d199323cbf0a4c2347a8af85f2875c1f2c26a1558fd34dfca7a26cff22 29.16MB / 29.16MB                                         4.8s
 => => sha256:d847ad1879f2202660d37d9bedf1b8986116079d578b010e46b6917facf23dd5 3.33MB / 3.33MB                                           6.2s
 => => sha256:7d020595b077c82c862bcf7a0981420ba252567bf0fdbd292d1d2286fc599b8b 14.71MB / 14.71MB                                         4.1s
 => => sha256:6b3b6b7b048eccdc8b78e16eeaaaf992dcb37e174a030e00229050c094f174d2 249B / 249B                                               4.6s
 => => extracting sha256:14c9d9d199323cbf0a4c2347a8af85f2875c1f2c26a1558fd34dfca7a26cff22                                                0.9s
 => => extracting sha256:d847ad1879f2202660d37d9bedf1b8986116079d578b010e46b6917facf23dd5                                                0.1s
 => => extracting sha256:7d020595b077c82c862bcf7a0981420ba252567bf0fdbd292d1d2286fc599b8b                                                0.5s
 => => extracting sha256:6b3b6b7b048eccdc8b78e16eeaaaf992dcb37e174a030e00229050c094f174d2                                                0.0s
 => [internal] load build context                                                                                                        0.0s
 => => transferring context: 373B                                                                                                        0.0s
 => [2/3] WORKDIR /app                                                                                                                   0.1s
 => [3/3] COPY . /app                                                                                                                    0.0s
 => exporting to image                                                                                                                   0.0s
 => => exporting layers                                                                                                                  0.0s
 => => writing image sha256:4bb73d3bc4f54c02e742e9d804cb3f5a6272ac428dec5ed5366da4be18d44ee5                                             0.0s
 => => naming to docker.io/library/hello-world-app                                                                                       0.0s

What's Next?
  1. Sign in to your Docker account → docker login
  2. View a summary of image vulnerabilities and recommendations → docker scout quickview
(base) kabdrakhman@MacBook-Pro-Altair simpledocker % docker run --rm hello-world-app
Hello from inside the container!
(base) kabdrakhman@MacBook-Pro-Altair simpledocker % 
```