**­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)

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

Create a app.yaml file with the following content:  
  
runtime: python39

handlers:

- url: /.\*

script: auto

1. **Deploy the Application**:

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

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

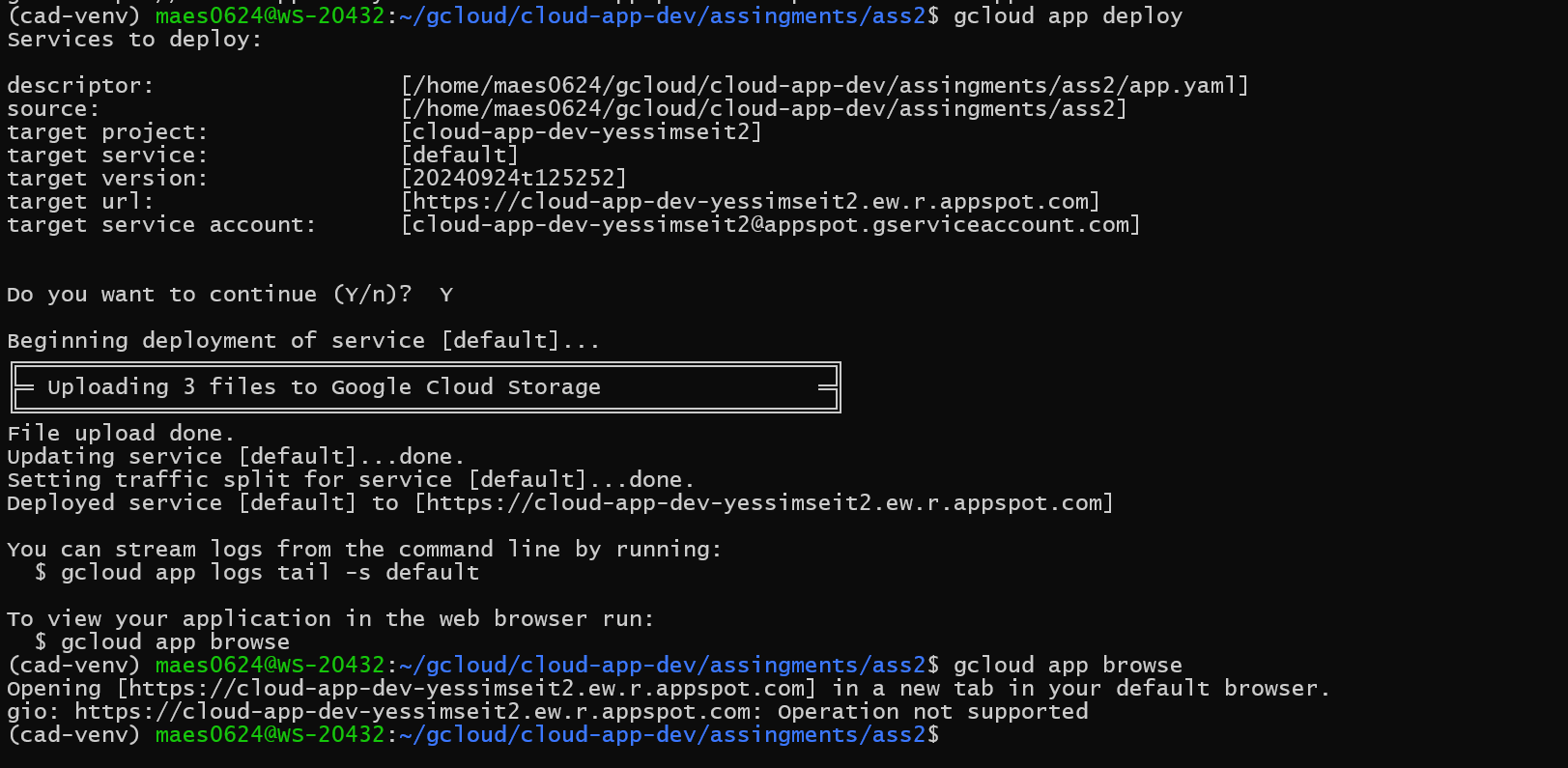
**Deliverables**:

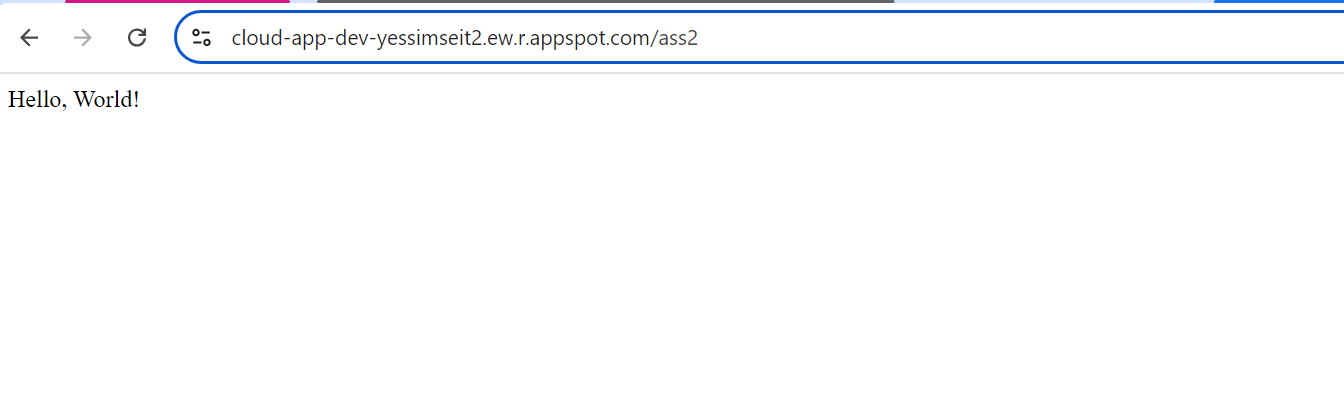
* A deployed web application on Google App Engine.
* A screenshot of the running application.

Firstly we create all files by instructions.

Then use command gcloud app deploy.

After command I have url and go to google .

* 



### **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!');

};

1. **Deploy the Function**:

Use the following command to deploy the function:  
  
gcloud functions deploy helloWorldFunction --runtime nodejs18 --trigger-http

1. **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 \

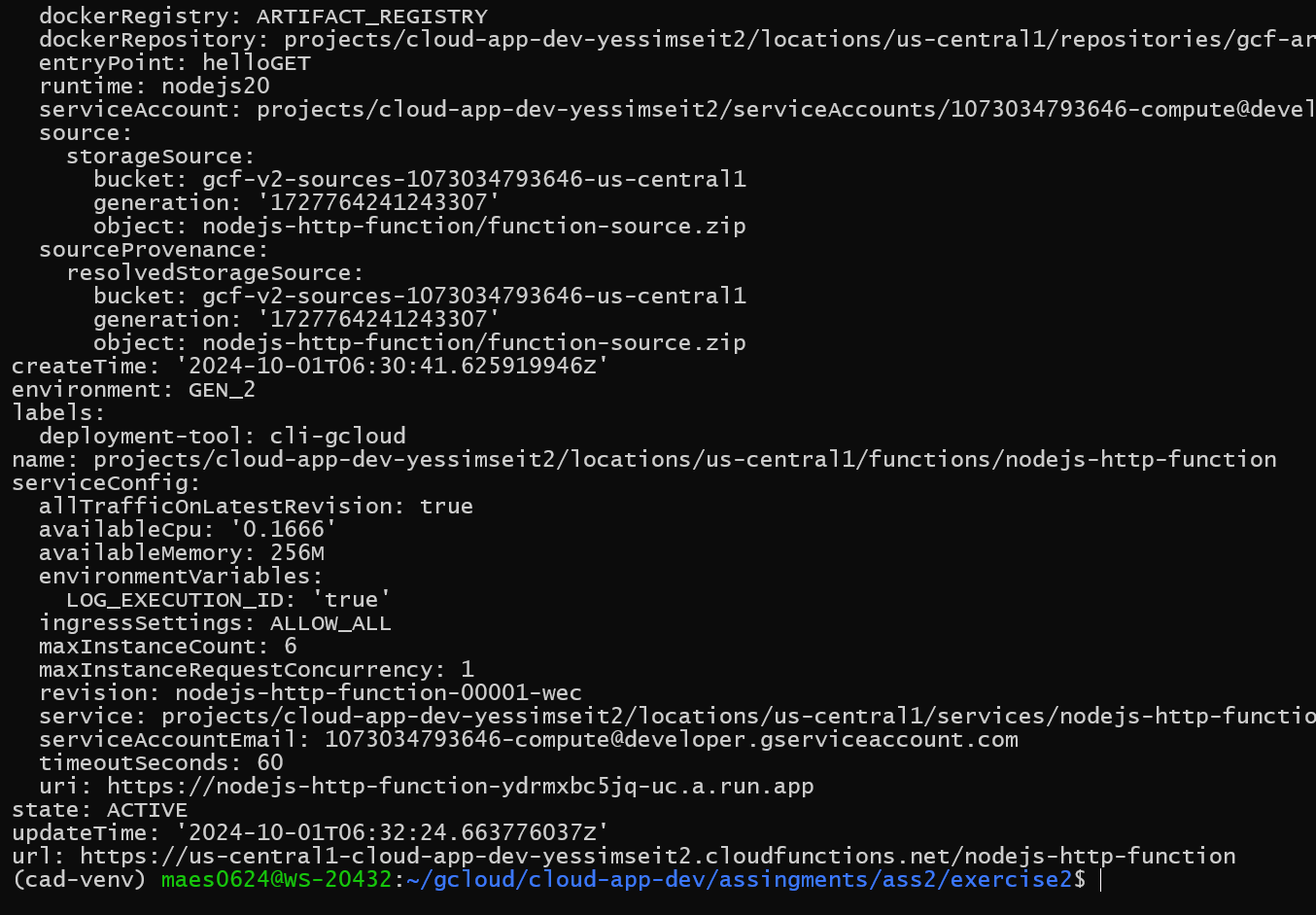
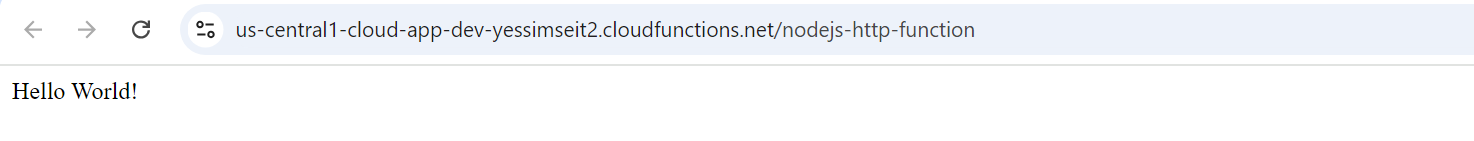
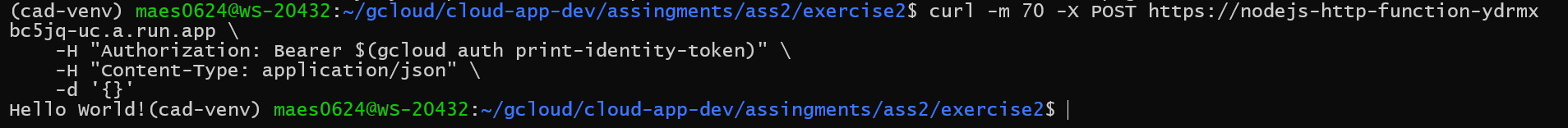
--runtime=nodejs20 \

--region=*us-central1* \

--source=. \

--entry-point=helloGET \  
--trigger-http

We deployed success.

* 
* 
* We use special uri from google
* 

### **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!")

1. **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"]

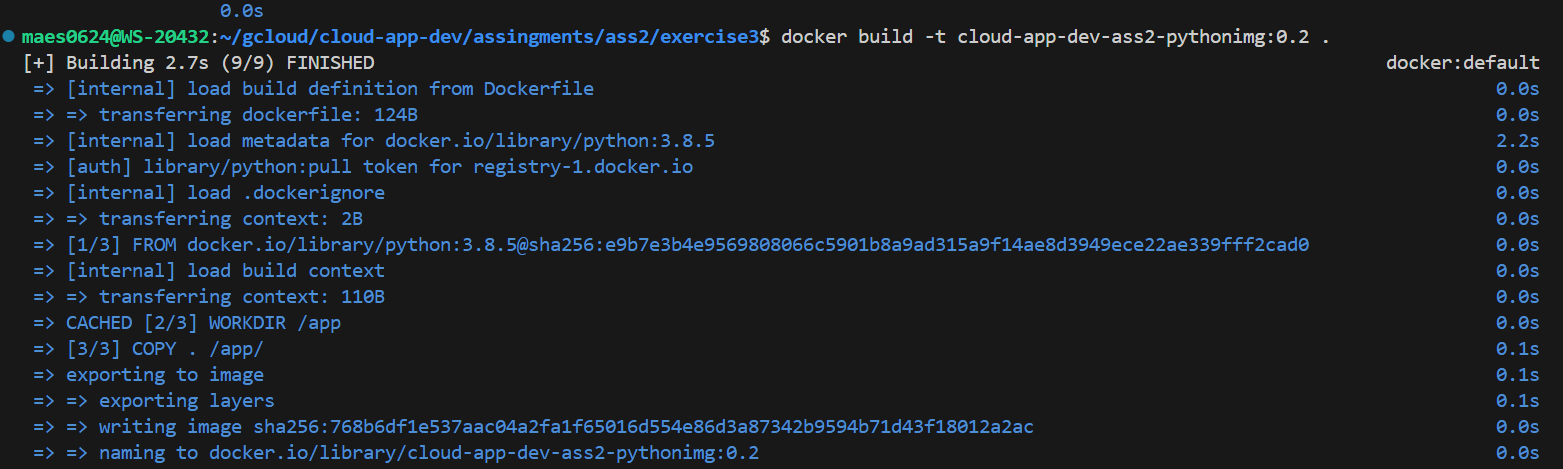
1. **Build the Docker Image**:

Build the Docker image using the following command:  
  
docker build -t hello-world-app .

1. **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!"
* 
* 