Exercise 1: Google App Engine

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

Step-by-Step Explanation:

1. **Setup Google Cloud**:
    - **Step**: Ensure you have a Google Cloud account.
    - **Install Google Cloud SDK**: Download and install the SDK following the official documentation. This is required to manage your app deployment.
2. **Create a Project**:
    - **Step**: In the Google Cloud Console, create a new project by navigating to console.cloud.google.com. Click "Select a project" > "New Project" > Name it > Click "Create."
3. **Prepare the Application**:

Step: Write a simple Flask app in Python. Here's your code:
python
Копировать код

```python
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)
```

    -
    - This is a basic Flask application that responds with “Hello, World!” when accessed.
4. **Create App Engine Configuration**:

Step: In the root directory of your project, create a file called `app.yaml` with the following content:
yaml
Копировать код

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

    -
    - This file tells Google App Engine how to run your application.
5. **Deploy the Application**:

Command: Open a terminal and run the following command:
Копировать код

```
gcloud app deploy
```

-
    - This command deploys your Flask application to Google App Engine.
6. **Access the Application**:
    - **Step**: After deployment, Google App Engine provides a URL where your app can be accessed. Copy that URL and open it in a browser to see your "Hello, World!" message.

Deliverables:

- **Screenshot**: Take a screenshot of your running application with the "Hello, World!" message displayed.

Exercise 2: Building with Google Cloud Functions

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

Step-by-Step Explanation:

1. **Setup Google Cloud**:
    - **Step**: Make sure you have a Google Cloud account.
    - **Install Google Cloud SDK**: If you haven't installed the SDK, follow the instructions here.
2. **Create a Cloud Function**:
    - **Step**: In the Google Cloud Console, navigate to “Cloud Functions” and click “Create Function.”
    - **Configuration**:
        - **Name**: `helloWorldFunction`
        - **Trigger**: HTTP
        - **Runtime**: Choose Node.js 18
        - **Entry Point**: `helloWorld`
3. **Write the Code**:

Step: Create a file `index.js` with the following code:
javascript
Копировать код
```
exports.helloWorld = (req, res) => {
    res.send('Hello, World!');
};
```

    -
    - This function responds with "Hello, World!" when accessed via HTTP.
4. **Deploy the Function**:

Command: In your terminal, use the following command to deploy the function:
css
Копировать код
```
gcloud functions deploy helloWorldFunction --runtime nodejs18 --trigger-http
```

    -
5. **Invoke the Function**:
    - **Step**: Once deployed, Google Cloud provides a URL for the function. Open this URL in a web browser to test it. You should see "Hello, World!" in the response.

Deliverables:

- **Screenshot**: Capture a screenshot of the response from the function showing "Hello, World!"

Exercise 3: Containerizing Applications

Objective: Containerize a simple application using Docker.

Step-by-Step Explanation:

1. **Install Docker**:
    - **Step**: Ensure Docker is installed by following the instructions on docker.com.
2. **Create a Simple Application**:

Step: Write a simple Python script (`app.py`) that prints "Hello from inside the container!" when run:
python
Копировать код
```
print("Hello from inside the container!")
```

    -
3. **Create a Dockerfile**:

Step: In the same directory as your `app.py`, create a file named `Dockerfile` with the following content:
Dockerfile
Копировать код
```
# 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**:

Command: In your terminal, navigate to the directory containing your `Dockerfile` and `app.py`, then run:
Копировать код
```
docker build -t hello-world-app .
```

    -
    - This command builds a Docker image named `hello-world-app`.
5. **Run the Docker Container**:

Command: Run the container using:
arduino
Копировать код
```
docker run --rm hello-world-app
```

    -
    - You should see the output "Hello from inside the container!" in your terminal.

Deliverables:

- **Screenshot**: Take a screenshot of the terminal showing the output "Hello from inside the container!"

Summary for Submission:

1. **Deployments**:
    - You need to deploy a Google App Engine app, a Google Cloud Function, and a Dockerized app.
2. **Screenshots**:
    - For each task, take screenshots of your final results (running web app, function response, and Docker container output) and include them in your report.
3. **Upload to GitHub**:
    - Ensure all your deliverables (code and screenshots) are uploaded to a GitHub repository.