GitHub Repository Setup, CI/CD Pipeline, GCP Deployment, and Email Notifications

Here's a comprehensive guide to achieve all your requirements:

1. GitHub Repository Setup

First, let's set up your GitHub repository properly:

1. Create a new repository on GitHub (don't initialize with README)
2. In your local project directory, run these commands:

# Initialize git

git init

# Create a .gitignore file

echo "\_\_pycache\_\_/" >> .gitignore

echo "\*.pyc" >> .gitignore

echo ".env" >> .gitignore

echo "venv/" >> .gitignore

# Add files and commit

git add .

git commit -m "Initial commit"

# Connect to GitHub and push

git remote add origin https://github.com/yourusername/your-repo-name.git

git branch -M main

git push -u origin main

2. CI/CD Pipeline with GitHub Actions

Create a .github/workflows/main.yml file with this content:

name: CI/CD Pipeline

on:

push:

branches: [ main ]

pull\_request:

branches: [ main ]

jobs:

test:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v2

- name: Set up Python

uses: actions/setup-python@v2

with:

python-version: '3.9'

- name: Install dependencies

run: |

python -m pip install --upgrade pip

pip install -r api/Requirements.txt

pip install pytest pytest-mock locust

- name: Run tests

run: |

cd api

pytest test.api\_endpoints.py -v

- name: Run performance tests

run: |

cd api

locust -f test\_performance.py --headless -u 10 -r 5 --run-time 30s --host=http://localhost:5000

deploy:

needs: test

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v2

- uses: google-github-actions/setup-gcloud@v0

with:

service\_account\_key: ${{ secrets.GCP\_SA\_KEY }}

project\_id: ${{ secrets.GCP\_PROJECT\_ID }}

- run: gcloud app deploy api/app.yaml –quiet

3. GCP Deployment Setup

Create these files in your project:

api/app.yaml:

runtime: python39

entrypoint: gunicorn -b :$PORT main:app

env\_variables:

OPENWEATHERMAP\_API\_KEY: 'e200c72831ddbb5f765391b6a3b620f4'

handlers:

- url: /.\*

script: auto

Requiremnts.txt

Flask==2.0.1

gunicorn==20.1.0

requests==2.26.0

pytest==6.2.5

locust==2.5.1

pytest-mock==3.6.1

GCP Setup Steps:

1. Create a Google Cloud Project
2. Enable App Engine
3. Create a Service Account with App Engine Admin and Storage Admin roles
4. Download the JSON key for the service account
5. Add these secrets to your GitHub repository:
   * GCP\_SA\_KEY: The content of the service account JSON file
   * GCP\_PROJECT\_ID: Your GCP project ID

4. Email Notifications Setup

Add this to your .github/workflows/main.yml file (at the end):

- name: Send email notification

if: always()

uses: dawidd6/action-send-mail@v3

with:

server\_address: smtp.gmail.com

server\_port: 465

username: ${{ secrets.MAIL\_USERNAME }}

password: ${{ secrets.MAIL\_PASSWORD }}

subject: ${{ job.status }}: CI/CD Pipeline for Weather API

to: your-email@example.com

from: GitHub Actions

body: |

The CI/CD pipeline for the Weather API has completed with status: ${{ job.status }}

View the logs at: https://github.com/${{ github.repository }}/actions/runs/${{ github.run\_id }}

Add these secrets to your GitHub repository:

* MAIL\_USERNAME: Your email address
* MAIL\_PASSWORD: Your email password or app-specific password

Final Folder Structure

Your final folder structure should look like this:

U-API-PROJECT/

├── .github/

│ └── workflows/

│ └── main.yml

├── apenv/

├── api/

│ ├── \_\_pycache\_\_/

│ ├── \_\_init\_\_.py

│ ├── config.py

│ ├── main.py

│ ├── test.api\_endpoints.py

│ ├── test\_performance.py

│ ├── weather.py

│ ├── pytest.ini

│ ├── Requirements.txt

│ ├── setup.py

│ └── app.yaml

└── .gitignore

Additional Recommendations

1. **Security**: Don't hardcode your API key in config.py. Use environment variables or a secrets manager.
2. **Monitoring**: Consider adding monitoring endpoints in your Flask app.
3. **Logging**: Add proper logging to your application.
4. **Docker**: Consider containerizing your application for more consistent deployments.

Would you like me to elaborate on any specific part of this setup?