Great project idea! You can absolutely **showcase a simple, real-world CI/CD pipeline** for Python using only free tools and AWS free tier resources. Here’s a step-by-step guide to **set up a modern CI/CD pipeline** for your “Hello World ~ Pratul” Python program, complete with testing and deployment to an AWS EC2 instance.

**Tech Stack & Free Tools**

* **GitHub** (for code hosting and CI/CD workflow with [GitHub Actions](https://github.com/features/actions))
* **AWS EC2 Free Tier** (for deployment target)
* **Python** (the app)
* **pytest** (for test case)
* **GitHub Secrets** (to store AWS credentials securely for deployment)

**Step 1: Simple Python App**

hello.py

def hello():

return "Hello World ~ Pratul"

if \_\_name\_\_ == "\_\_main\_\_":

print(hello())

**Step 2: Simple Test Case**

test\_hello.py

from hello import hello

def test\_hello():

assert hello() == "Hello World ~ Pratul"

**Step 3: Requirements File**

requirements.txt

pytest

**Step 4: GitHub Actions Workflow for CI/CD**

Create the following file in your repo:  
.github/workflows/cicd.yml

name: CI/CD Pipeline

on:

push:

branches:

- main

jobs:

test:

runs-on: ubuntu-latest

steps:

- name: Checkout code

uses: actions/checkout@v4

- name: Set up Python

uses: actions/setup-python@v5

with:

python-version: '3.10'

- name: Install dependencies

run: pip install -r requirements.txt

- name: Run tests

run: pytest

deploy:

needs: test

runs-on: ubuntu-latest

if: github.ref == 'refs/heads/main'

steps:

- name: Checkout code

uses: actions/checkout@v4

- name: Install AWS CLI

run: |

sudo apt-get update

sudo apt-get install -y awscli

- name: Configure AWS credentials

uses: aws-actions/configure-aws-credentials@v4

with:

aws-access-key-id: ${{ secrets.AWS\_ACCESS\_KEY\_ID }}

aws-secret-access-key: ${{ secrets.AWS\_SECRET\_ACCESS\_KEY }}

aws-region: us-east-1

- name: Launch EC2 instance and deploy

run: |

INSTANCE\_ID=$(aws ec2 run-instances \

--image-id ami-0c94855ba95c71c99 \ # Ubuntu 22.04 LTS (update with latest AMI in your region)

--instance-type t2.micro \

--key-name ${{ secrets.AWS\_KEY\_PAIR\_NAME }} \

--security-group-ids ${{ secrets.AWS\_SECURITY\_GROUP\_ID }} \

--subnet-id ${{ secrets.AWS\_SUBNET\_ID }} \

--count 1 \

--output text \

--query 'Instances[0].InstanceId')

echo "Launched EC2 instance: $INSTANCE\_ID"

aws ec2 wait instance-running --instance-ids $INSTANCE\_ID

PUBLIC\_IP=$(aws ec2 describe-instances --instance-ids $INSTANCE\_ID --query "Reservations[0].Instances[0].PublicIpAddress" --output text)

echo "Public IP: $PUBLIC\_IP"

sleep 60 # Give the instance a moment to boot

# Copy the code to the EC2 instance and run

scp -i ${{ secrets.AWS\_SSH\_PRIVATE\_KEY\_PATH }} -o StrictHostKeyChecking=no hello.py ubuntu@$PUBLIC\_IP:/home/ubuntu/

ssh -i ${{ secrets.AWS\_SSH\_PRIVATE\_KEY\_PATH }} -o StrictHostKeyChecking=no ubuntu@$PUBLIC\_IP "sudo apt update && sudo apt install -y python3 && python3 /home/ubuntu/hello.py"

env:

AWS\_ACCESS\_KEY\_ID: ${{ secrets.AWS\_ACCESS\_KEY\_ID }}

AWS\_SECRET\_ACCESS\_KEY: ${{ secrets.AWS\_SECRET\_ACCESS\_KEY }}

**What you need to prepare:**

1. **AWS Account** (use free tier: t2.micro)
2. **Create a Key Pair** in AWS EC2 (for SSH)
3. **Create Security Group** to allow SSH (port 22) from your IP
4. **Store secrets in GitHub repo**:
   * AWS\_ACCESS\_KEY\_ID
   * AWS\_SECRET\_ACCESS\_KEY
   * AWS\_KEY\_PAIR\_NAME
   * AWS\_SSH\_PRIVATE\_KEY\_PATH (upload private key as secret or use an encrypted runner)
   * AWS\_SECURITY\_GROUP\_ID
   * AWS\_SUBNET\_ID

**How the Pipeline Works**

* **Step 1:** On every push to main branch:
  + Runs tests with pytest.
* **Step 2:** If tests pass:
  + Spins up a free-tier EC2, waits for it to boot.
  + Copies your Python script over SSH, runs it, and prints output.

**What to Show Employers**

* **A real GitHub repo** with the above code and workflow.
* **CI/CD badge** from GitHub Actions showing status.
* **Screenshot or logs** from GitHub Actions showing auto-deployment.

**Let me know if you want:**

* Step-by-step setup for AWS/GitHub secrets
* A video guide
* Ready-to-fork demo repo
* Any other customizations!

**Would you like me to make a sample repo for you, or do you want help with the AWS setup or GitHub Actions YAML?**