CI/CD Deployment of FastAPI using Docker, GitHub Actions, and AWS EC2 (Amazon Linux 2)

1. Launch EC2 Instance (Amazon Linux 2)

- Go to AWS EC2 dashboard > Launch Instance.
- Choose Amazon Linux 2 AMI.
- Choose instance type (e.g., t2.micro for free tier).
- Configure security group to allow:
  - SSH (port 22) - for GitHub Action and your access.
  - HTTP (port 8000) - for web app.
- Download and save the key pair (.pem).

2. Connect to EC2 via SSH

Use Git Bash or terminal:

```
ssh -i "your-key.pem" ec2-user@<EC2-PUBLIC-IP>
```

3. Install Docker on EC2 (Amazon Linux 2)

```
sudo yum update -y

sudo yum install docker -y

sudo service docker start

sudo usermod -a -G docker ec2-user

# Logout and reconnect to apply group permissions
```

4. Add Git

```
sudo yum install git -y
```

5. GitHub Setup

- Go to GitHub > Settings > Secrets and variables > Actions > New repository secret:
  - EC2_SSH_KEY (your private SSH key contents)
  - EC2_HOST (public DNS or IP of EC2)
  - AWS_ACCESS_KEY (IAM user access key)
  - AWS_SECRET_KEY (IAM user secret key)

6. Create GitHub Actions Workflow (deploy.yml)

File path: .github/workflows/deploy.yml

```
name: Deploy to EC2

on:
  push:
    branches: [main]

jobs:
  deploy:
    name: Deploy on EC2
    runs-on: ubuntu-latest

    steps:
    - name: Checkout code
      uses: actions/checkout@v3

    - name: Set up SSH
      uses: webfactory/ssh-agent@v0.7.0
      with:
          ssh-private-key: ${{ secrets.EC2_SSH_KEY }}

      - name: Deploy to EC2 and run Docker
        run: |
          ssh -o StrictHostKeyChecking=no ec2-user@${{ secrets.EC2_HOST }} << 'EOF'
            cd CICD-Model-Deployement-AWS || git clone https://github.com/<your-username>/CICD-Model-Deployement-AWS.git && cd CICD-Model-Deployement-AWS
            git pull origin main
            echo "AWS_ACCESS_KEY=${{ secrets.AWS_ACCESS_KEY }}" > .env
            echo "AWS_SECRET_KEY=${{ secrets.AWS_SECRET_KEY }}" >> .env
            docker stop tip-app || true
            docker rm tip-app || true
            docker build -t tip-app .
            docker run -d -p 8000:8000 --env-file .env --name tip-app tip-app
          EOF
```

7. Trigger the CI/CD

- Push any change to the main branch of your GitHub repository.
- It will automatically run the deploy.yml workflow.
- Visit: http://<EC2-PUBLIC-IP>:8000/

8. Clean Up

- To avoid charges:
  - Terminate EC2 instance.
  - Check and release unused Elastic IPs (if any).
  - Clear Secrets if no longer needed.