Flask App Deployment on EC2 with Docker and GitHub Actions

1. Introduction

This document outlines the step-by-step process followed to deploy a Flask application on an EC2 instance using Docker and GitHub Actions for CI/CD.

2. Forking the Repository

1. Fork the original repository on GitHub.
2. Clone the forked repository:
   ```bash
   git clone <your-forked-repo-url>
   ```
3. Navigate to the project directory:
   ```bash
   cd photo-gallery-python-flask
   ```

3. Setting up EC2 Instance

1. Launch an Ubuntu EC2 instance on AWS.
2. SSH into the instance:
   ```bash
   ssh -i your-key.pem ubuntu@<EC2-PUBLIC-IP>
   ```
3. Install Docker:
   ```bash
   sudo apt update -y
   sudo apt install -y docker.io
   sudo systemctl start docker
   sudo systemctl enable docker
   ```

4. Building and Running Docker Container

1. Build the Docker image:
   ```bash
   docker build -t flask-app .
   ```
2. Run the container:
   ```bash
   docker run -d -p 5000:5000 --name flask-app flask-app
   ```

5. Setting up GitHub Actions Workflow

A GitHub Actions workflow (`.github/workflows/deploy.yml`) was created to automate deployment:

```
name: CI/CD Pipeline for Flask App with Docker

on:
  push:
    branches:
      - staging
      - master

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout Code
        uses: actions/checkout@v3

      - name: Set Up Docker Buildx
        uses: docker/setup-buildx-action@v2

      - name: Login to Docker Hub
        uses: docker/login-action@v2
        with:
          username: ${{ secrets.DOCKER_HUB_USERNAME }}
          password: ${{ secrets.DOCKER_HUB_PASSWORD }}

      - name: Build and Push Docker Image
        uses: docker/build-push-action@v4
        with:
          context: .
          file: ./Dockerfile
          push: true
          tags: |
            ${{ secrets.DOCKER_HUB_USERNAME }}/flask-app:latest
            ${{ secrets.DOCKER_HUB_USERNAME }}/flask-app:${{ github.sha }}

  deploy:
    needs: build
    runs-on: ubuntu-latest
    if: github.ref == 'refs/heads/master'
```

```
  steps:
    - name: Deploy to EC2
      uses: appleboy/ssh-action@master
      with:
        host: ${{ secrets.EC2_HOST }}
        username: ${{ secrets.EC2_USER }}
        key: ${{ secrets.EC2_SSH_KEY }}
        script: |
          sudo apt update -y
          sudo apt install -y docker.io
          sudo systemctl start docker
          sudo systemctl enable docker
          docker stop flask-app || true
          docker rm flask-app || true
          docker image prune -af || true
          docker pull ${{ secrets.DOCKER_HUB_USERNAME }}/flask-app:latest
          docker run -d --name flask-app -p 5000:5000 --restart unless-stopped ${{
secrets.DOCKER_HUB_USERNAME }}/flask-app:latest
```

6. GitHub Secrets

In GitHub Repository -> Settings -> Secrets and Variables -> Actions, add the following secrets:

- `DOCKER_HUB_USERNAME`
- `DOCKER_HUB_PASSWORD`
- `EC2_HOST`
- `EC2_USER`
- `EC2_SSH_KEY`
- `SSH_PRIVATE_KEY`
- `STAGING_SERVER_IP`

7. Configuring Nginx

1. Open the Nginx configuration file:
   ```bash
   sudo nano /etc/nginx/sites-available/default
   ```
2. Update the configuration to proxy requests to the Flask app running on port 5000:
   ```nginx
   server {
      listen 80;
      location / {
         proxy_pass http://127.0.0.1:5000;
         proxy_set_header Host $host;
```

```
        proxy_set_header X-Real-IP $remote_addr;
    }
}
```

3. Restart Nginx:
```bash
sudo systemctl restart nginx
```

8. Testing Deployment

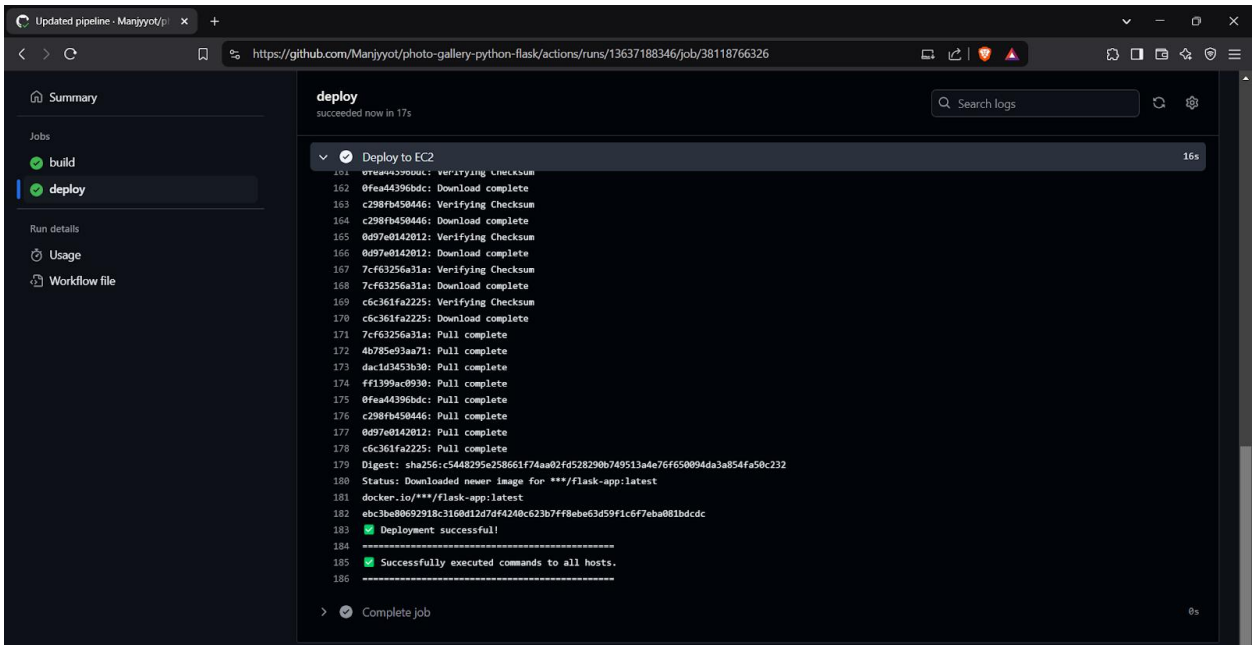
1. Ensure the application is running:
```bash
docker ps
```
2. Check logs if errors occur:
```bash
docker logs flask-app
```
3. Open the application in a browser:
```
http://<EC2-PUBLIC-IP>
```

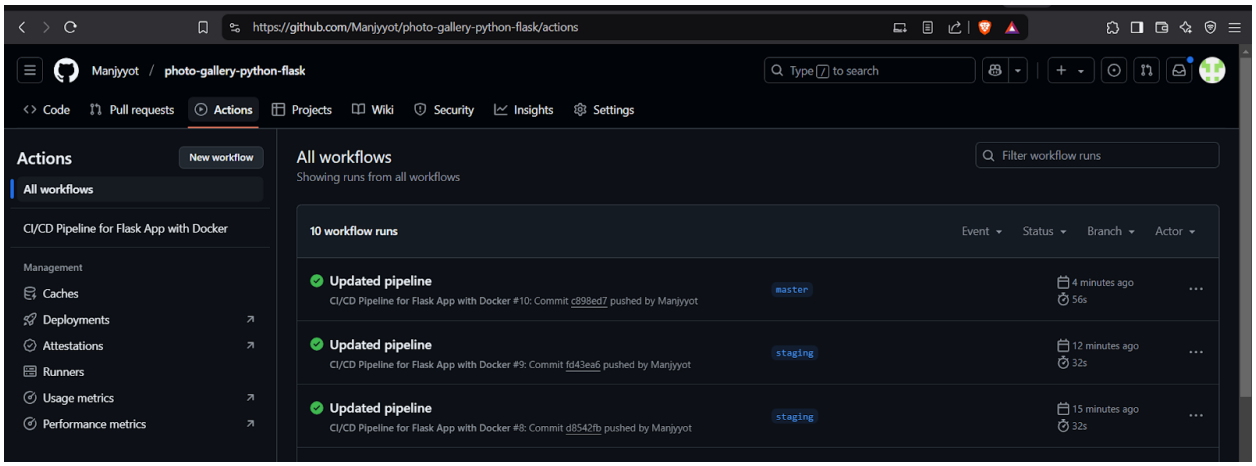
9. Summary of Steps

- Forked the repository
- Set up an **EC2 instance** (Ubuntu)
- Installed **Docker** and configured Nginx
- Created **GitHub Actions** workflow for **CI/CD**
- Set up **secrets and variables** in GitHub Actions
- Deployed the application successfully

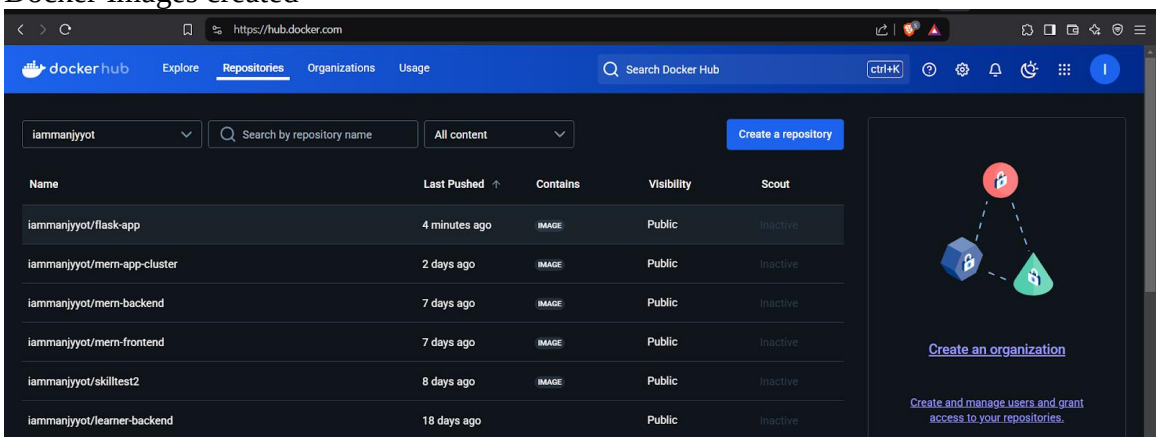
Media:

Pipeline result on pushing to the Master branch





Docker Images created



Docker Containers

```
ubuntu@ip-10-0-0-58:~/photo-gallery-python-flask$ docker ps
CONTAINER ID   IMAGE                              COMMAND           CREATED          STATUS          PORTS                                                NAMES
f2932b2a3cd5   iammanjyyot/flask-app:latest   "python main.py"   37 seconds ago   Up 35 seconds   0.0.0.0:5000->5000/tcp, :::5000->5000/tcp   flask-app
ubuntu@ip-10-0-0-58:~/photo-gallery-python-flask$ docker images
REPOSITORY             TAG       IMAGE ID       CREATED              SIZE
iammanjyyot/flask-app   latest    6380719f0021   About a minute ago   154MB
ubuntu@ip-10-0-0-58:~/photo-gallery-python-flask$
```

Github Actions, Secrets and variables