CI/CD pipeline

**A. For the Backend (FastAPI, Dockerized, on EC2):**

1. Code Push:

Developer pushes code to the main branch on GitHub (or GitLab, Bitbucket, etc.).

2. CI Pipeline:

* Lint & Test: Run Python linters (e.g., flake8, black) and unit tests (pytest).
* Build Docker Image: Build a Docker image for the FastAPI app.
* Push to Registry: Push the image to Docker Hub or AWS ECR.

3. CD Pipeline:

* Deploy to EC2:
  + SSH into the EC2 instance (or use AWS SSM).
  + Pull the latest Docker image.
  + Stop the old container and start the new one.
* (Optional) Use AWS CodeDeploy or Ansible for zero-downtime deployment.

**B. For the Frontend (React, S3 + CloudFront):**

1. Code Push:

Developer pushes code to the main branch.

2. CI Pipeline:

* Lint & Test: Run JS/TS linters and frontend tests.
* Build: Build the static React app (npm run build).

CD Pipeline:

* Sync to S3: Upload the build output to the S3 bucket.
* Invalidate CloudFront Cache: Invalidate the distribution to serve the latest files.

**C. Tools:**

* CI/CD Platform: GitHub Actions, GitLab CI, Bitbucket Pipelines, CircleCI, or AWS CodePipeline.
* Docker Registry: Docker Hub or AWS ECR.
* Deployment: SSH, AWS CLI, or tools like Ansible/Terraform for infrastructure changes.

**D. GitHub Actions Workflow (Backend)**

**Yaml file for the GitHub Actions Workflow**

name: CI/CD Backend

on:

push:

branches: [main]

jobs:

build-and-deploy:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v3

- name: Set up Python

uses: actions/setup-python@v4

with:

python-version: '3.10'

- name: Install dependencies

run: pip install -r backend/requirements.txt

- name: Run tests

run: pytest

- name: Build Docker image

run: docker build -t myrepo/fastapi-backend:latest ./backend

- name: Log in to Docker Hub

run: echo "${{ secrets.DOCKER\_PASSWORD }}" | docker login -u "${{ secrets.DOCKER\_USERNAME }}" --password-stdin

- name: Push Docker image

run: docker push myrepo/fastapi-backend:latest

- name: Deploy to EC2

uses: appleboy/ssh-action@v0.1.10

with:

host: ${{ secrets.EC2\_HOST }}

username: ubuntu

key: ${{ secrets.EC2\_SSH\_KEY }}

script: |

docker pull myrepo/fastapi-backend:latest

docker stop fastapi || true

docker rm fastapi || true

docker run -d --name fastapi -p 80:8000 myrepo/fastapi-backend:latest