**Part 1: Deploy Flask and Express on a Single EC2 Instance**

**1. Provision the EC2 Instance**

* Launch an **Ubuntu Server (20.04 or 22.04)** EC2 instance (free-tier eligible: t2.micro).
* Open security group rules to allow:
  + Port 22 (SSH)
  + Port 3000 (Express)
  + Port 5000 (Flask)
  + Port 8080 (Jenkins)
* SSH into your instance:

bash

ssh -i "your-key.pem" ubuntu@<EC2-PUBLIC-IP>

**2. Install Required Software**

bash

CopyEdit

# Update

sudo apt update && sudo apt upgrade -y

# Python and pip for Flask

sudo apt install python3 python3-pip -y

# Node.js and npm for Express

curl -sL https://deb.nodesource.com/setup\_18.x | sudo -E bash -

sudo apt install -y nodejs

# Git

sudo apt install git -y

# pm2 (for process management)

sudo npm install pm2 -g

**3. Clone and Set Up Applications**

bash

# Clone Flask and Express apps

git clone https://github.com/your-username/your-flask-repo.git

git clone https://github.com/your-username/your-express-repo.git

# Flask

cd your-flask-repo

pip3 install -r requirements.txt

# Update Flask app to run on 0.0.0.0:5000 if needed

pm2 start flask\_app.py --interpreter=python3 --name flask-app

# Express

cd ../your-express-repo

npm install

# Ensure it listens on 0.0.0.0:3000

pm2 start app.js --name express-app

**4. Architecture Overview**

**Diagram (Textual):**

less

[User] --> [EC2 Public IP:3000] --> Express App

--> [EC2 Public IP:5000] --> Flask App

|

[EC2 Instance]

┌────────────────────┐

| Flask (port 5000) |

| Express (port 3000)|

| Jenkins (port 8080)|

└────────────────────┘

**Part 2: CI/CD Pipeline Using Jenkins**

**1. Install Jenkins**

bash

# Java (required for Jenkins)

sudo apt install openjdk-11-jdk -y

# Jenkins

wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -

sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'

sudo apt update

sudo apt install jenkins -y

sudo systemctl start jenkins

sudo systemctl enable jenkins

# Allow Jenkins port

sudo ufw allow 8080

* Access Jenkins at http://<EC2-PUBLIC-IP>:8080
* Get the initial admin password:

bash

sudo cat /var/lib/jenkins/secrets/initialAdminPassword

**2. Install Jenkins Plugins**

* Required:
  + Git
  + NodeJS
  + Python
  + Pipeline
  + PM2 integration (optional)

**3. Create Pipelines**

You’ll create two pipelines:

**Jenkinsfile for Flask App**

groovy

pipeline {

agent any

environment {

APP\_DIR = "/home/ubuntu/your-flask-repo"

}

stages {

stage('Clone') {

steps {

git 'https://github.com/your-username/your-flask-repo.git'

}

}

stage('Install Dependencies') {

steps {

sh 'pip3 install -r requirements.txt'

}

}

stage('Restart App') {

steps {

sh "pm2 restart flask-app || pm2 start flask\_app.py --interpreter=python3 --name flask-app"

}

}

}

}

**Jenkinsfile for Express App**

groovy

pipeline {

agent any

environment {

APP\_DIR = "/home/ubuntu/your-express-repo"

}

stages {

stage('Clone') {

steps {

git 'https://github.com/your-username/your-express-repo.git'

}

}

stage('Install Dependencies') {

steps {

sh 'npm install'

}

}

stage('Restart App') {

steps {

sh "pm2 restart express-app || pm2 start app.js --name express-app"

}

}

}

}

**4. Set Up GitHub Webhooks**

* Go to GitHub → Repository → Settings → Webhooks
* Payload URL: http://<EC2-PUBLIC-IP>:8080/github-webhook/
* Content type: application/json
* Events: Just the push event.

**Optional Enhancements**

* Add stages like:
  + Testing (using pytest or mocha)
  + Linting
* Store secrets using Jenkins credentials.
* Install dotenv or use .env securely.

**✅ Deliverables Checklist**

| **Deliverable** | **Description** |
| --- | --- |
| ✅ GitHub Links | Public repos for Flask and Express |
| ✅ EC2 Instance | Running Flask on 5000, Express on 3000 |
| ✅ Screenshot | Browser accessing Flask and Express via EC2 public IP |
| ✅ Jenkins Screenshots | Successful build logs for both apps |
| ✅ Jenkinsfiles | Located in root of each repo |
| ✅ README.md | Contains setup steps, architecture, CI/CD pipeline details |

**✅ Sample README.md Structure**

markdown

CopyEdit

# Flask + Express CI/CD Deployment on AWS EC2

## Part 1: Deployment

### Architecture

- Flask runs on port 5000

- Express runs on port 3000

- Both hosted on a single EC2 instance

### How to Access

- `http://<EC2-PUBLIC-IP>:5000` → Flask

- `http://<EC2-PUBLIC-IP>:3000` → Express

## Part 2: CI/CD with Jenkins

### Jenkins Pipelines

- Flask: Pulls repo, installs dependencies, restarts with PM2

- Express: Same as above

### Trigger

- GitHub webhooks trigger Jenkins on code push

## Setup Guide

- [x] EC2 provisioning

- [x] Jenkins setup

- [x] Node.js, Python, PM2 installation

- [x] CI/CD pipeline