**Project 4: CI/CD from Scratch – Flask + Jenkins + Docker**

**✅ Project Objective:**

To **build a complete CI/CD pipeline** that automatically fetches code from GitHub, builds a Docker image, runs tests, and deploys a Flask app using **Jenkins**.

**🧰 Tools & Technologies Used:**

* **Flask (Python Web Framework)**
* **GitHub (Code Repository)**
* **Jenkins (CI/CD Server)**
* **Docker (Containerization)**
* **Linux OS (Ubuntu/CentOS)**
* **Git**

**⚙️ Project Workflow:**

1. **Push Flask App Code to GitHub**
2. **Jenkins Pulls the Code via Webhook**
3. **Build Docker Image in Jenkins Pipeline**
4. **Run Flask App in Docker Container**
5. **Automatic Redeploy on Code Changes**

**🧱 Sample Folder Structure:**

Copy code

flask-app/

├── app.py

├── requirements.txt

├── Dockerfile

└── Jenkinsfile

**🔁 CI/CD Pipeline Stages:**

**1. Clone GitHub Repository**

groovy

Copy code

stage('Clone') {

steps {

git 'https://github.com/username/flask-app.git'

}

}

**2. Build Docker Image**

groovy

Copy code

stage('Build') {

steps {

sh 'docker build -t flask-ci-cd-app .'

}

}

**3. Run Docker Container**

groovy

Copy code

stage('Deploy') {

steps {

sh 'docker run -d -p 5000:5000 flask-ci-cd-app'

}

}

**🐳 Sample Dockerfile:**

dockerfile

Copy code

FROM python:3.9-slim

WORKDIR /app

COPY . /app

RUN pip install -r requirements.txt

CMD ["python", "app.py"]

**✅ Expected Output:**

* Jenkins job automatically deploys your Flask app inside a Docker container on every push.
* Access app at http://<your\_server\_ip>:5000/

**💡 Benefits:**

* Fully automated deployment process
* Easy rollback with version control
* Combines the power of GitHub + Jenkins + Docker
* Eliminates manual builds and reduces human error