**Jenkins + Docker Pipeline:**

**Step-by-Step Guide**

**Objective**

Automate the **build, test, and deployment** of a **Dockerized application** using **Jenkins Pipeline**.

**Prerequisites**

Ensure you have installed the following on your Jenkins server:  
✅ **Jenkins** (Latest version)  
✅ **Docker** (For containerization)  
✅ **Git** (For version control)  
✅ **Jenkins Plugins:**

* **Pipeline Plugin**
* **Docker Pipeline Plugin**
* **Git Plugin**
* **Blue Ocean Plugin** (for better visualization)

**🔹 Step 1: Install Docker on Jenkins Server**

**1.1 Install Docker**

bash

CopyEdit

sudo apt update

sudo apt install docker.io -y

**1.2 Start and Enable Docker**

bash

CopyEdit

sudo systemctl start docker

sudo systemctl enable docker

**1.3 Add Jenkins User to Docker Group**

bash

CopyEdit

sudo usermod -aG docker jenkins

✔️ **Restart Jenkins** to apply changes:

bash

CopyEdit

sudo systemctl restart jenkins

**1.4 Verify Docker Installation**

bash

CopyEdit

docker --version

**🔹 Step 2: Create a Simple Web Application**

**2.1 Clone Sample App**

bash

CopyEdit

git clone https://github.com/your-username/docker-jenkins-app.git

cd docker-jenkins-app

**2.2 Create a Dockerfile**

Inside the project folder, create a Dockerfile:

dockerfile

CopyEdit

# Use an official Nginx image

FROM nginx:latest

COPY index.html /usr/share/nginx/html/

EXPOSE 80

CMD ["nginx", "-g", "daemon off;"]

**2.3 Create an index.html**

Inside the project folder:

html

CopyEdit

<!DOCTYPE html>

<html>

<head>

<title>Jenkins + Docker Pipeline</title>

</head>

<body>

<h1>Deployment Successful with Jenkins and Docker!</h1>

</body>

</html>

**2.4 Create a Jenkinsfile**

Inside the project folder:

groovy

CopyEdit

pipeline {

agent any

environment {

DOCKER\_IMAGE = "my-jenkins-app"

DOCKER\_TAG = "latest"

DOCKER\_REPO = "your-dockerhub-username/my-jenkins-app"

}

stages {

stage('Clone Repository') {

steps {

git 'https://github.com/your-username/docker-jenkins-app.git'

}

}

stage('Build Docker Image') {

steps {

script {

sh "docker build -t ${DOCKER\_IMAGE}:${DOCKER\_TAG} ."

}

}

}

stage('Run Container Locally') {

steps {

script {

sh "docker run -d -p 8080:80 --name my-container ${DOCKER\_IMAGE}:${DOCKER\_TAG}"

}

}

}

stage('Push to Docker Hub') {

steps {

script {

docker.withRegistry('https://index.docker.io/v1/', 'docker-hub-credentials') {

sh "docker tag ${DOCKER\_IMAGE}:${DOCKER\_TAG} ${DOCKER\_REPO}:${DOCKER\_TAG}"

sh "docker push ${DOCKER\_REPO}:${DOCKER\_TAG}"

}

}

}

}

stage('Deploy to Server') {

steps {

script {

sh "ssh user@remote-server 'docker pull ${DOCKER\_REPO}:${DOCKER\_TAG} && docker run -d -p 80:80 ${DOCKER\_REPO}:${DOCKER\_TAG}'"

}

}

}

}

}

**🔹 Step 3: Push Code to GitHub**

**3.1 Initialize Git Repository**

bash

CopyEdit

git init

git add .

git commit -m "Initial Commit for Jenkins + Docker Pipeline"

git branch -M main

git remote add origin https://github.com/your-username/docker-jenkins-app.git

git push -u origin main

**🔹 Step 4: Configure Jenkins for Docker Pipeline**

**4.1 Add Docker Credentials in Jenkins**

1. **Go to Jenkins Dashboard → Manage Jenkins → Manage Credentials**
2. Select **Global Credentials → Add Credentials**
3. **Type:** Username and Password
4. **ID:** docker-hub-credentials
5. Click **Save**

**4.2 Create a New Jenkins Pipeline Job**

1. **Go to Jenkins Dashboard → New Item**
2. **Enter Job Name:** Docker-Pipeline
3. Select **Pipeline** and click **OK**

**4.3 Configure Pipeline**

1. Select **"Pipeline script from SCM"**
2. **SCM:** Git
3. **Repository URL:** https://github.com/your-username/docker-jenkins-app.git
4. **Branch:** main
5. **Script Path:** Jenkinsfile
6. Click **Save**

**🔹 Step 5: Run the Jenkins Pipeline**

1. Click **"Build Now"**
2. Monitor the **Console Output**
3. If successful, the output will show:

pgsql

CopyEdit

Successfully built Docker image

Successfully pushed image to Docker Hub

Deployed container on remote server

**🔹 Step 6: Verify Deployment**

1. Check running containers:

bash

CopyEdit

docker ps

1. Open the deployed application in a browser:

cpp

CopyEdit

http://<remote-server-ip>

**✅ Summary**

🎯 **Jenkins automates:**  
✔️ **Pulling the latest code from GitHub**  
✔️ **Building a Docker image**  
✔️ **Running the container locally**  
✔️ **Pushing the image to Docker Hub**  
✔️ **Deploying the app on a remote server**