**📘 Jenkins Deployment Pipeline: Step-by-Step Guide**

**🎯 Objective**

Create a Jenkins pipeline that **builds**, **tests**, and **deploys** a sample web application (e.g., a Python Flask app or Node.js app) to a target server (like a staging or prod server).

**🏗️ Prerequisites**

**✅ Mandatory Requirements**

| **Item** | **Description** |
| --- | --- |
| Jenkins installed | Master with at least 1 agent |
| GitHub repo | Sample project source code |
| Target server | SSH access (e.g., staging or production EC2) |
| Jenkins plugins | Git, Pipeline, SSH Agent, Credentials Binding |
| Credentials | SSH key or username/password stored in Jenkins |

**🧱 Sample Use Case**

* GitHub repository: https://github.com/example/flask-app
* Tech: Python Flask app
* Goal: Deploy to remote Ubuntu server over SSH

**📁 Sample Jenkinsfile**

pipeline {

agent any

environment {

REMOTE\_HOST = 'your-server-ip'

REMOTE\_USER = 'ubuntu'

DEPLOY\_DIR = '/opt/myapp'

}

stages {

stage('Checkout') {

steps {

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

}

}

stage('Build') {

steps {

sh 'pip install -r requirements.txt'

}

}

stage('Test') {

steps {

sh 'pytest tests/'

}

}

stage('Deploy') {

steps {

sshagent(['my-ssh-credentials']) {

sh """

ssh -o StrictHostKeyChecking=no $REMOTE\_USER@$REMOTE\_HOST '

mkdir -p $DEPLOY\_DIR &&

rm -rf $DEPLOY\_DIR/\*'

scp -r \* $REMOTE\_USER@$REMOTE\_HOST:$DEPLOY\_DIR/

"""

}

}

}

}

}

**🧾 Jenkins Deployment Flow Explanation**

**🔹 Step 1: Create a Jenkins Pipeline Job**

* Go to **Jenkins Dashboard > New Item**
* Name: flask-app-deploy
* Type: **Pipeline**
* Under "Pipeline script", paste the above Jenkinsfile

**🔹 Step 2: Configure Credentials**

* Go to **Manage Jenkins > Credentials**
* Add **SSH Username with Private Key**
* ID: my-ssh-credentials

**🔹 Step 3: Connect Git Repository**

* Jenkinsfile can be:
  + Inline (in job config)
  + Or from Git: use **Pipeline from SCM** and link GitHub repo

**🔹 Step 4: Install Required Plugins**

Make sure these are installed:

| **Plugin** | **Purpose** |
| --- | --- |
| Git | Clone source code |
| Pipeline | Enable Jenkinsfile |
| SSH Agent | Securely connect to target server |
| Credentials Binding | Use secrets/SSH keys securely |

**🔹 Step 5: Run the Job**

* Click **Build Now**
* Check Console Output
* Verify code is copied and running in target server directory

**✅ Result**

Your Jenkins pipeline:

* Pulls code from GitHub
* Installs dependencies
* Runs tests
* Deploys code over SSH to target server

**🔐 Security Notes**

* Never hard-code passwords or IPs in Jenkinsfile
* Use environment variables and credentials
* Set up proper RBAC for Jenkins users

**🧼 Optional Enhancements**

* Add email or Slack notification on success/failure
* Add rollback logic
* Use Docker container for deployment
* Add health check step post-deploy