**Step-by-Step Guide: Build and Deploy a Simple App using Jenkins**

This guide will walk you through **building and deploying** a simple application using **Jenkins CI/CD pipeline**.

Before you start, make sure you have the following:  
**Jenkins installed** (locally, on a VM, or on Docker)  
**Git installed** on Jenkins server  
**Docker installed** if deploying with containers  
A **GitHub repository** with a simple application (e.g., Python, Java, or Node.js)

**Step 1: Install Required Plugins**

1️. Open **Jenkins Dashboard** → **Manage Jenkins** → **Manage Plugins**.  
2️. Install the following plugins:

* **Pipeline**
* **Git Plugin**
* **Docker Pipeline Plugin** (if using Docker)

**Step 2: Create a New Jenkins Pipeline Job**

1️. In Jenkins, click **New Item** → **Pipeline** → Give it a name → Click **OK**.  
2️. Scroll down to the **Pipeline** section and select **Pipeline script from SCM**.  
3️. In the **SCM** field, select **Git**, then enter your **GitHub repository URL**.  
4️. In the **Branch Specifier**, enter \*/main or \*/master.  
5️. In the **Script Path**, enter Jenkinsfile (which we will create in the next step).

**Step 3: Create a Jenkinsfile (CI/CD Pipeline)**

In your GitHub repository, create a new file named **Jenkinsfile** and add the following code:

groovy

CopyEdit

pipeline {

agent any

stages {

stage('Clone Repository') {

steps {

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

}

}

stage('Build') {

steps {

sh 'echo "Building the application..."'

}

}

stage('Test') {

steps {

sh 'echo "Running tests..."'

}

}

stage('Deploy') {

steps {

sh 'echo "Deploying application..."'

}

}

}

}

**Explanation:**

* **Stage 1: Clone Repository** → Pulls the latest code from GitHub.
* **Stage 2: Build** → Builds the application (placeholder command).
* **Stage 3: Test** → Runs tests (placeholder command).
* **Stage 4: Deploy** → Deploys the app (placeholder command).

**Step 4: Build and Run the Pipeline**

1️. Go back to **Jenkins Dashboard** → Click on your job.  
2️. Click **Build Now**.  
3️. Click **Console Output** to see the progress.

If everything is successful, you will see all **stages completed** in the output.

**Step 5: Deploy with Docker (Optional)**

If you want to deploy your application as a **Docker container**, modify your **Jenkinsfile** like this:

groovy

CopyEdit

pipeline {

agent any

environment {

IMAGE\_NAME = 'your-dockerhub-username/my-app'

}

stages {

stage('Clone Repository') {

steps {

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

}

}

stage('Build Docker Image') {

steps {

sh 'docker build -t $IMAGE\_NAME .'

}

}

stage('Push Docker Image') {

steps {

withDockerRegistry([credentialsId: 'docker-hub-credentials']) {

sh 'docker push $IMAGE\_NAME'

}

}

}

stage('Deploy') {

steps {

sh 'docker run -d -p 8080:80 $IMAGE\_NAME'

}

}

}

}

**Explanation:**

* **Builds a Docker image** of your application.
* **Pushes it to Docker Hub** (requires credentials setup in Jenkins).
* **Deploys the container** on a local machine or server.
* You have successfully set up **Jenkins CI/CD** to **build, test, and deploy** a simple app!
* If using **Docker**, your app is now running on **http://localhost:8080**.