**Part 2 – Apache Website (Single Machine)**

1. **Create Instance with 22, 80, 8080, 9090, 9100, 9091, 9093, 3000**
2. **Commands**

# Update

$ sudo apt update && sudo apt upgrade -y

# Install Docker

$ sudo apt install -y docker.io docker-compose

$ sudo systemctl enable docker --now

$ sudo usermod -aG docker ubuntu # log out & in again

# Install required tools

$ sudo apt update

$ sudo apt install -y ca-certificates curl gnupg

# Create keyrings folder if not exists

$ sudo mkdir -p /usr/share/keyrings

# Download Jenkins key and store it properly

curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | \

sudo tee /usr/share/keyrings/jenkins-keyring.asc > /dev/null

# Add Jenkins apt repo

echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \

https://pkg.jenkins.io/debian-stable binary/" | \

sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null

$ sudo apt update

$ sudo apt install -y fontconfig openjdk-17-jre Jenkins

$ sudo systemctl enable jenkins

$ sudo systemctl start jenkins

$ sudo systemctl status Jenkins

Open Jenkins in browser

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

1. **Install Plugins**

**Git plugin**

**Docker Pipeline**

**Pipeline**

**Blue Ocean**

1. **New Item 🡪 name: apache-site-pipeline 🡪 Select Pipeline → click OK.**

**Pipeline 🡪 Definition → choose Pipeline script.**

pipeline {

agent any

stages {

stage('Hello') {

steps {

echo 'Hello, Jenkins is working!'

}

}

stage('List Files') {

steps {

sh 'ls -la'

}

}

}

}

1. **Next (Real Repo)**

In git hub

$ nano Jenkinsfile

then keep this code there

Once you confirm this works, replace the script with your real pipeline, **e.g.:**

pipeline {

agent any

stages {

stage('Checkout Code') {

steps {

git 'https://github.com/your-repo/apache-site.git'

}

}

stage('Build Docker Image') {

steps {

sh 'docker build -t apache-test .'

}

}

stage('Run Container') {

steps {

sh 'docker run -d -p 8080:80 apache-test'

}

}

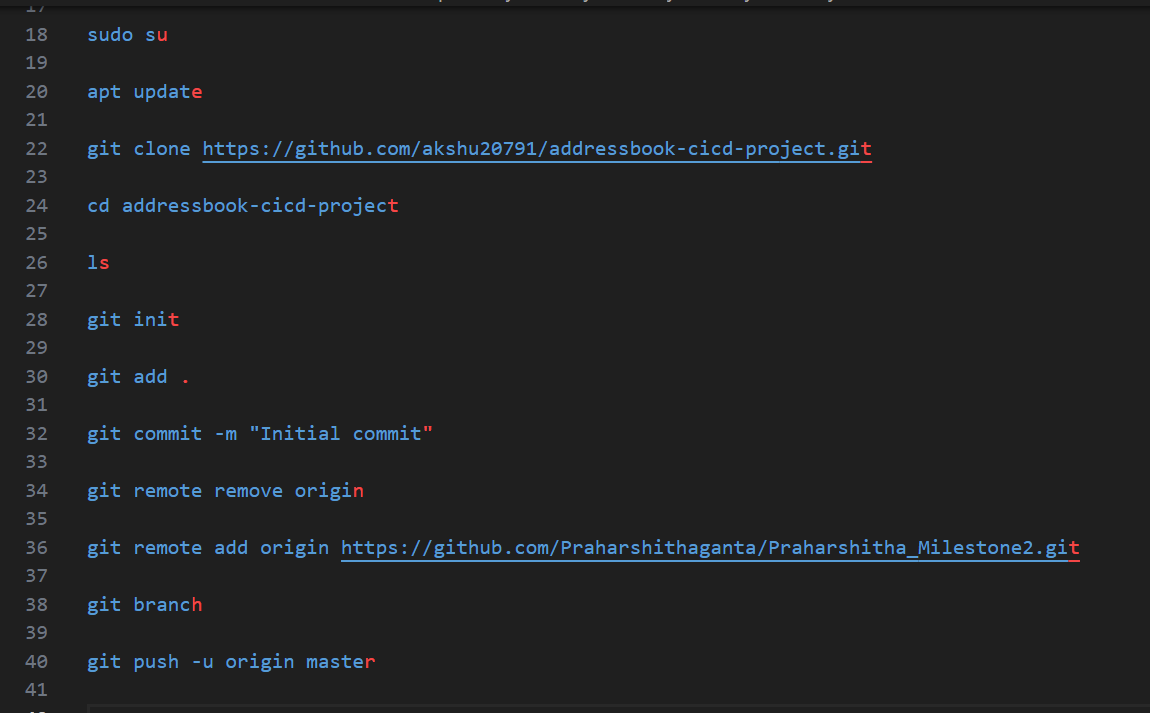
}

}

**Then try this**

* **Pipeline Job**
* **Defination 🡪 Pipeline script from SCM**
* **SCM: Git**
* **Paste your GitHub repo URL.**
* **Script Path: Jenkinsfile.**
* **Save and click Build Now.**

1. **Git**



$ git branch -M main

$ git push -u origin main

$ mkdir public-html

$ nano public-html/index.html

Paste this

<!DOCTYPE html>

<html>

<head>

<title>My Apache Website</title>

</head>

<body>

<h1>Hello from Jenkins CI/CD!</h1>

</body>

</html>

1. **Add Dockerfile**

$ nano Dockerfile

Paste this

FROM httpd:2.4

COPY ./public-html/ /usr/local/apache2/htdocs/

1. **Add Jenkins**

$ nano Jenkinsfile

Paste this

pipeline {

agent any

environment {

DOCKERHUB = "your-dockerhub-username"

IMAGE = "apache-site"

}

stages {

stage('Checkout') {

steps {

git 'https://github.com/your-repo/apache-site.git'

}

}

stage('Build Docker') {

steps {

sh 'docker build -t $DOCKERHUB/$IMAGE:$BUILD\_NUMBER .'

}

}

stage('Push Docker') {

steps {

sh 'echo $DOCKERHUB\_PASS | docker login -u $DOCKERHUB --password-stdin'

sh 'docker push $DOCKERHUB/$IMAGE:$BUILD\_NUMBER'

}

}

stage('Deploy Container') {

steps {

sh 'docker rm -f apache-site || true'

sh 'docker run -d --name apache-site -p 8080:80 $DOCKERHUB/$IMAGE:$BUILD\_NUMBER'

}

}

stage('Health Check') {

steps {

sh 'curl -f http://localhost:8080 || exit 1'

}

}

}

}

1. **Monitoring (Prometheus + Grafana + Node Exporter)**

**This is in Word file Doc**

1. **Create DockerHub Access**

Go to DockerHub.

Log in.

At top-right, click your profile → Account Settings.

Personal Access Token.

Name: jenkins-token

Permissions: Read/Write

Copy the token (this is your DockerHub password substitute).

1. **Add Credentials in Jenkins**

**Open Jenkins UI → Dashboard.**

**Go to Manage Jenkins → Credentials.**

**Click on (global) under “Stores scoped to Jenkins”.**

**Click Add Credentials.**

**Kind:** Username with password

**Username: praharshitha10**

**Password:** the access token you created

**ID: DOCKERHUB\_PASS (important, because you’ll use this in Jenkinsfile)**

**Description:** DockerHub credentials for Jenkins

**Click Save.**

1. **Update Jenkinsfile**

Modify your Jenkinsfile so Jenkins uses the credentials when building & pushing:

pipeline {

agent any

environment {

DOCKERHUB\_CREDENTIALS = credentials('DOCKERHUB\_PASS')

DOCKER\_IMAGE = " DOCKERHUB\_PASS /apache-site"

}

stages {

stage('Checkout Code') {

steps {

git 'https://github.com/Praharshithaganta/apache-site.git'

}

}

stage('Build Docker Image') {

steps {

sh 'docker build -t $DOCKER\_IMAGE:latest .'

}

}

stage('Push to DockerHub') {

steps {

sh 'echo $DOCKERHUB\_CREDENTIALS\_PSW | docker login -u $DOCKERHUB\_CREDENTIALS\_USR --password-stdin'

sh 'docker push $DOCKER\_IMAGE:latest'

}

}

stage('Deploy Container') {

steps {

sh 'docker rm -f apache-site || true'

sh 'docker run -d -p 8080:80 --name apache-site $DOCKER\_IMAGE:latest'

}

}

}

}

1. **Step 1: Run the Pipeline**

* **Pipeline Job**
* **Defination 🡪 Pipeline script from SCM**
* **SCM: Git**
* **Paste your GitHub repo URL.**
* **Script Path: Jenkinsfile.**
* **Save and click Build Now.**

**Click Build Now.**

**Watch the console output for each stage:**

**Checkout Code → should say it cloned your GitHub repo.**

**Build Docker Image → should show Docker build logs.**

**Push to DockerHub → should show Login Succeeded then Pushed.**

**Deploy Container → should show docker run -d ... succeeded.**

**Step 2: Verify on DockerHub**

**Log in to DockerHub.**

**Go to Repositories.**

**You should see your repo:**

**<your-dockerhub-username>/apache-site**

**Inside, you should see the latest tag uploaded.**

**Step 3: Verify Container on EC2**

**Run on your EC2:**

**docker ps**