## **Multi-Branch Project**

### **Project Overview**

* Create a simple Java Maven project.
* Version-control the project using Git with multiple branches.
* Set up Jenkins multi-branch pipeline for automated build and deployment.

## **Project Deliverables**

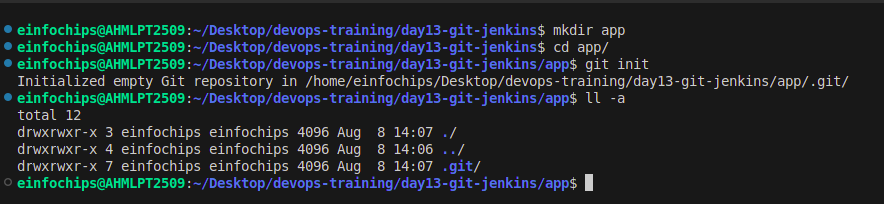
### **1.Git Repository:**

* Local Git repository initialized.

mkdir app

cd app

git init



* Branches: development, staging, and production.

git branch development

git branch staging

git branch production

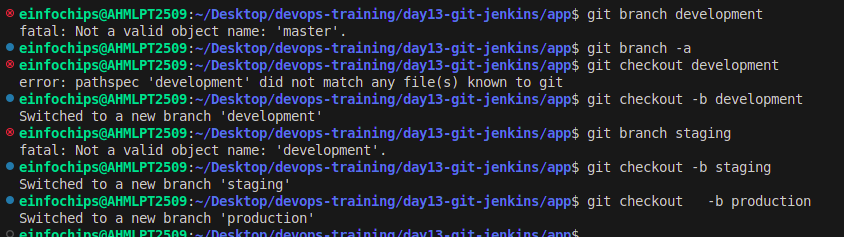
Create branches

git checkout -b development

git checkout -b staging

git checkout -b production

Repository pushed to remote Git server (e.g., GitHub, GitLab, Bitbucket).



### **2.Maven Project:**

* Simple Java Maven project created (HelloWorld application).

Create file src/main/java/com/example/App.java

mkdir -p src/main/java/com/example/

cd src/main/java/com/example/

nano App.java

package com.example;

public class App {

public static void main(String[] args) {

System.out.println("Hello World !");

}

}

* pom.xml with dependencies and build configurations.

nano pom.xml

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/POM/4.0.0/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>my-java-app</artifactId>

<version>1.0-SNAPSHOT</version>

<properties>

<maven.compiler.source>1.8</maven.compiler.source>

<maven.compiler.target>1.8</maven.compiler.target>

</properties>

</project>

# git add all code development , staging , production:

git branch

git checkout -b development

git add .

git commit -m "Initial commit on development branch"

git push -u origin development

git branch

git checkout -b staging

git checkout staging

git add .

git commit -m "Initial commit on staging branch"

git push -u origin staging

git checkout -b production

git checkout production

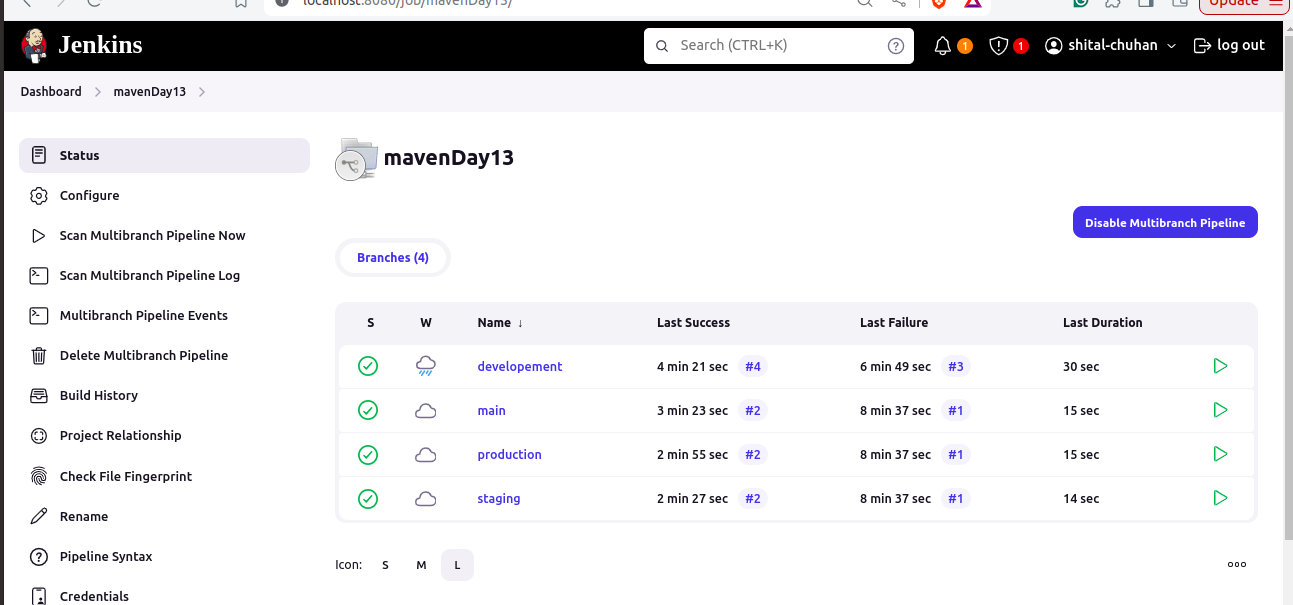
git add .

git commit -m "Initial commit on production branch"

git push -u origin production

### **3.Jenkins Setup:**

Multi-branch pipeline job configured in Jenkins.



* Jenkinsfile defining build and deployment steps.

pipeline {

agent any

tools{

maven 'maven-3.9.8'

}

stages {

stage('Checkout') {

steps {

git url: 'https://github.com/cshital/maven.git', branch: env.BRANCH

}

}

stage('Build') {

steps {

script {

echo "Building pull request branch: ${env.BRANCH}"

sh 'mvn clean install'

}

}

}

stage('Test') {

steps {

script {

echo "Running tests on pull request branch: ${env.BRANCH}"

sh 'mvn test'

}

}

}

stage('Output') {

steps{

script{

sh 'java src/main/java/com/example/App.java'

}

}

}

stage('Archive Artifacts') {

steps {

archiveArtifacts artifacts: '\*\*/target/\*.jar', allowEmptyArchive: true

}

}

}

post {

always {

echo 'Pipeline finished.'

}

success {

echo 'Pipeline succeeded.'

}

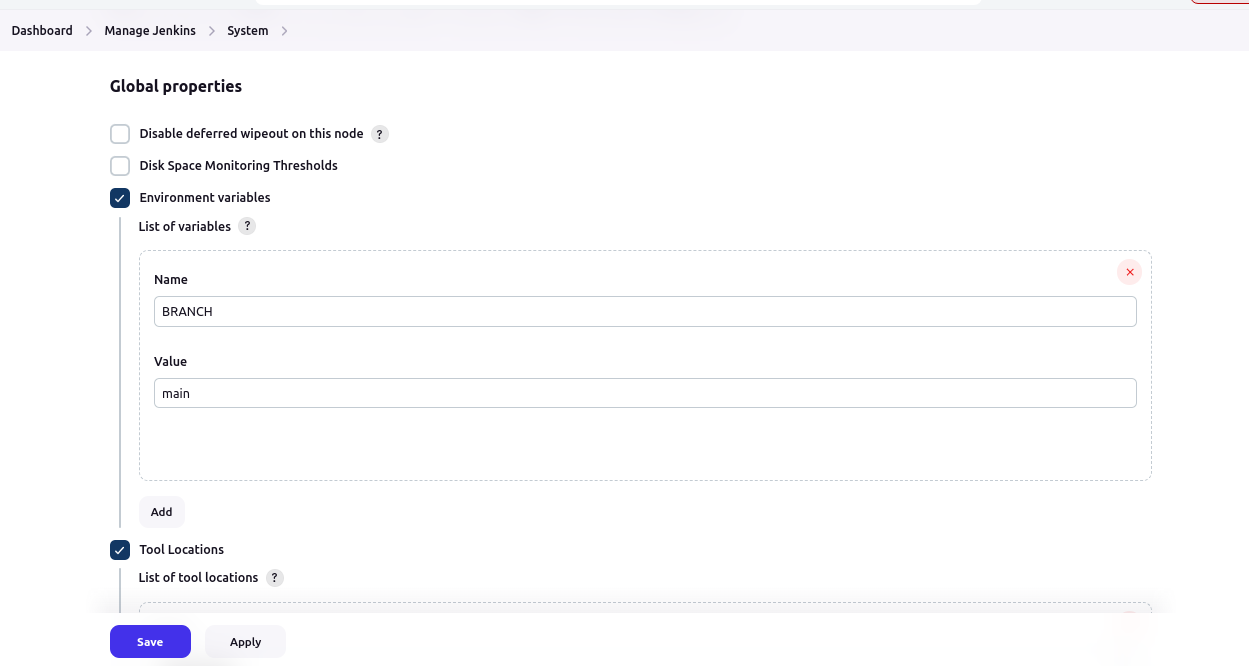
failure {

echo 'Pipeline failed.'

}

}

}



Output

