Jenkins Scripted Pipeline (Groovy)

--> For scripted pipeline, we will be using Groovy language

--> Flexibility

--> Customization

--> Reusable components

Syntax:

node {

stage(‘git clone’){

echo ‘git cloning’

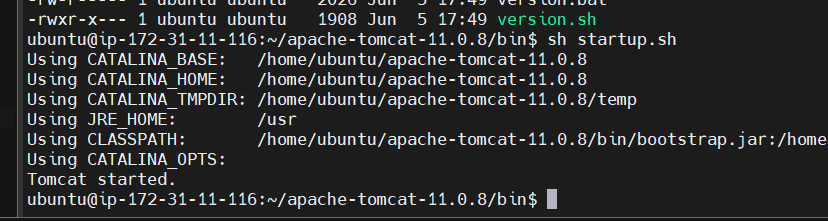
}

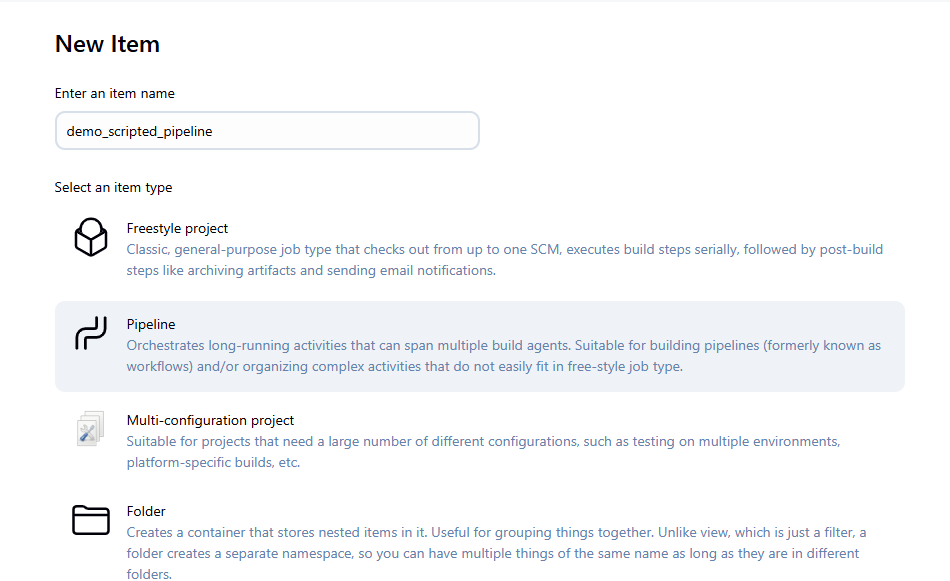
stage(‘build’){

echo ‘build…’

}

}







node {

stage('Cloning') {

echo 'git repo cloning'

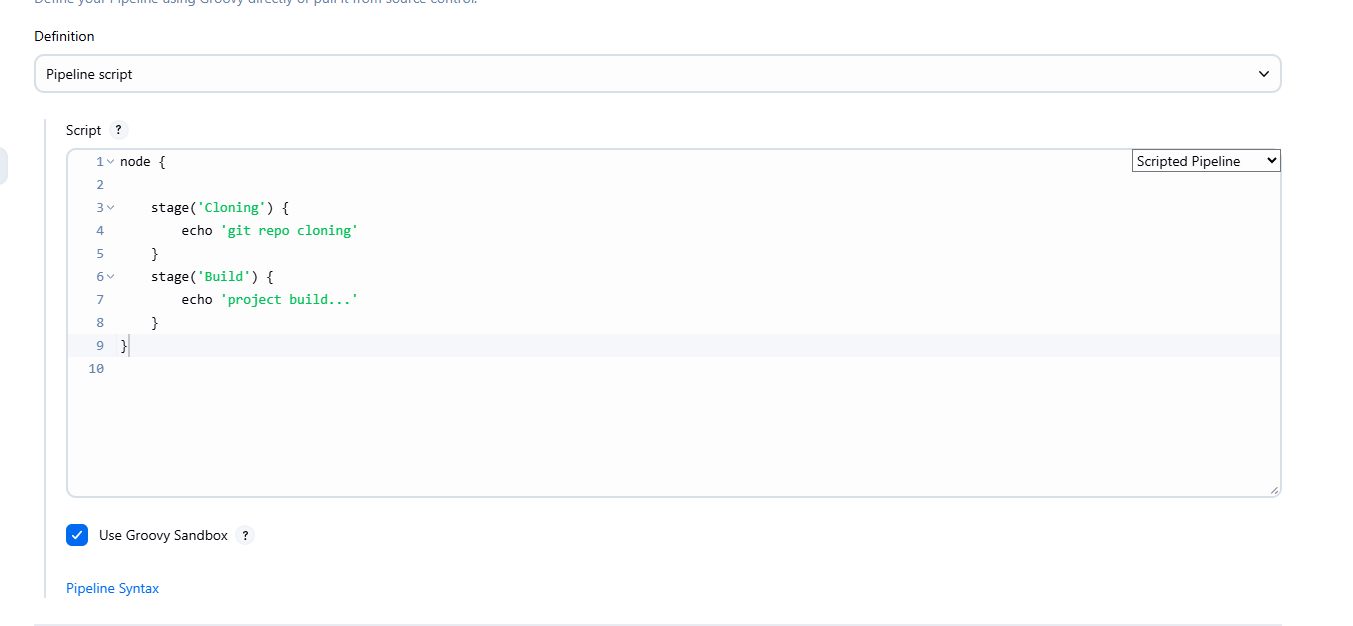
}

stage('Build') {

echo 'project build...'

}

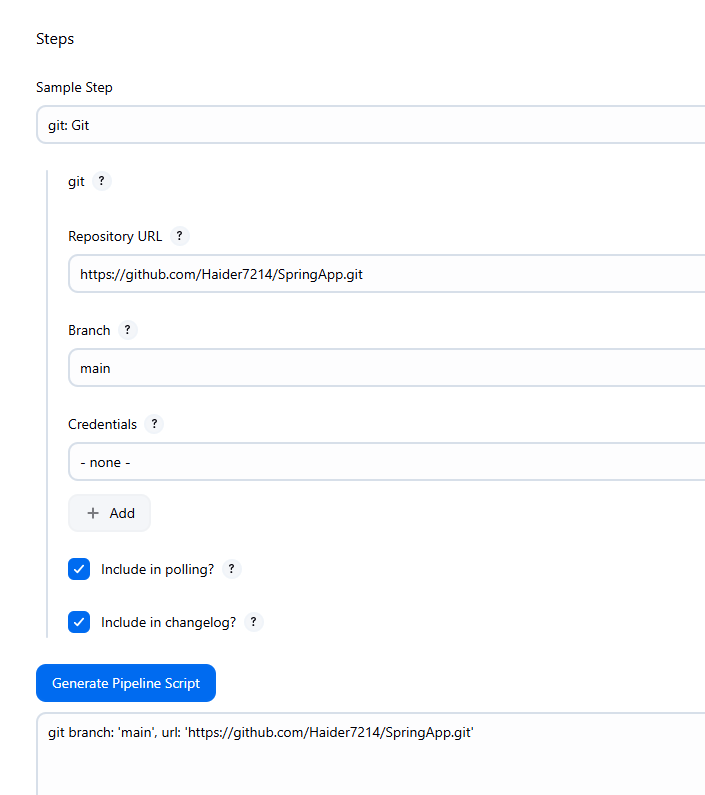
}



Apply and Save



Go back to Pipeline --> Configure



For Declarative it is different, for Scripted it is totally different

git branch: 'main', url: 'https://github.com/Haider7214/SpringApp.git'

node {

def mvnPath

stage('git clone') {

git branch: 'main', url: 'https://github.com/Haider7214/SpringApp.git'

}

stage('Maven Build') {

def mvnHome = tool name:'maven:3.9.10', type='maven';

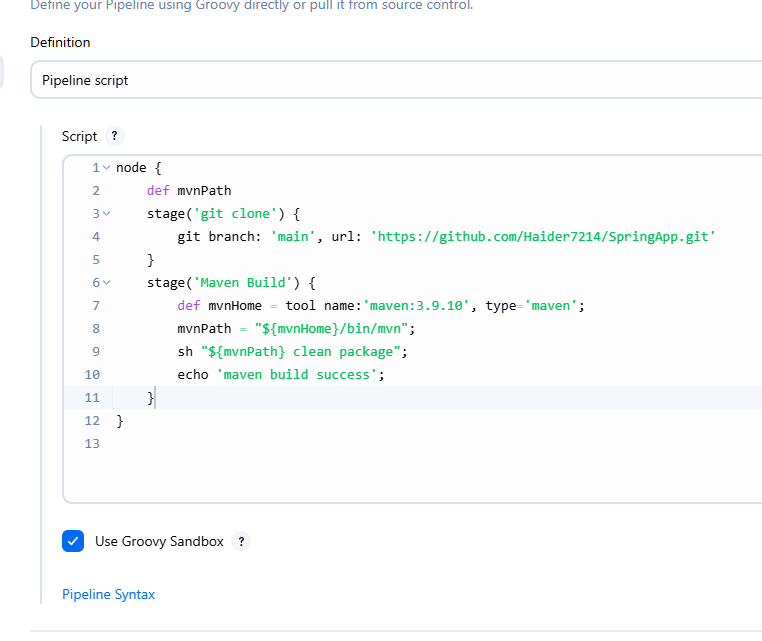
mvnPath = "${mvnHome}/bin/mvn";

sh "${mvnPath} clean package";

echo 'maven build success';

}

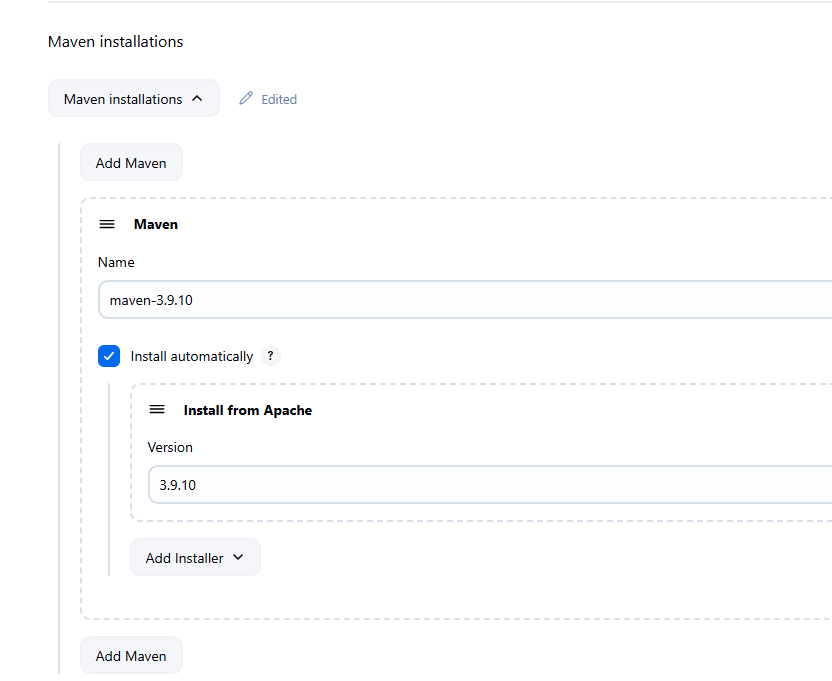
}



Apply and Save

Manage Jenkins --> Tools

That’s the name we have given for Maven



Failed



I change the script

node {

def mvnPath

stage('git clone') {

git branch: 'main', url: 'https://github.com/Haider7214/SpringBoot.git'

}

stage('Maven Build') {

def mvnHome = tool name:'maven-3.9.10', type='maven';

mvnPath = "${mvnHome}/bin/mvn";

sh "${mvnPath} clean package";

echo 'maven build success';

}

}

Again Build failed

node {

def mvnPath

stage('git clone') {

git branch: 'main', url: 'https://github.com/Haider7214/SpringBoot.git'

}

stage('Maven Build') {

def mvnHome=tool name:'maven-3.9.10', type:'maven';

mvnPath="${mvnHome}/bin/mvn";

sh "${mvnPath} clean package";

echo 'maven build success';

}

}

This time build was successful

