**Deploy an application in 4 different environments by using Jenkins with yes or no condition**

In this project we going to deploying an application in Dev, Test, Production using Jenkins pipeline CI/CD.

For deploying web application I going to using the tomcat web server.

To do this we need 4 servers

One for Jenkins.

One for development tomcat server.

One for testing tomcat server.

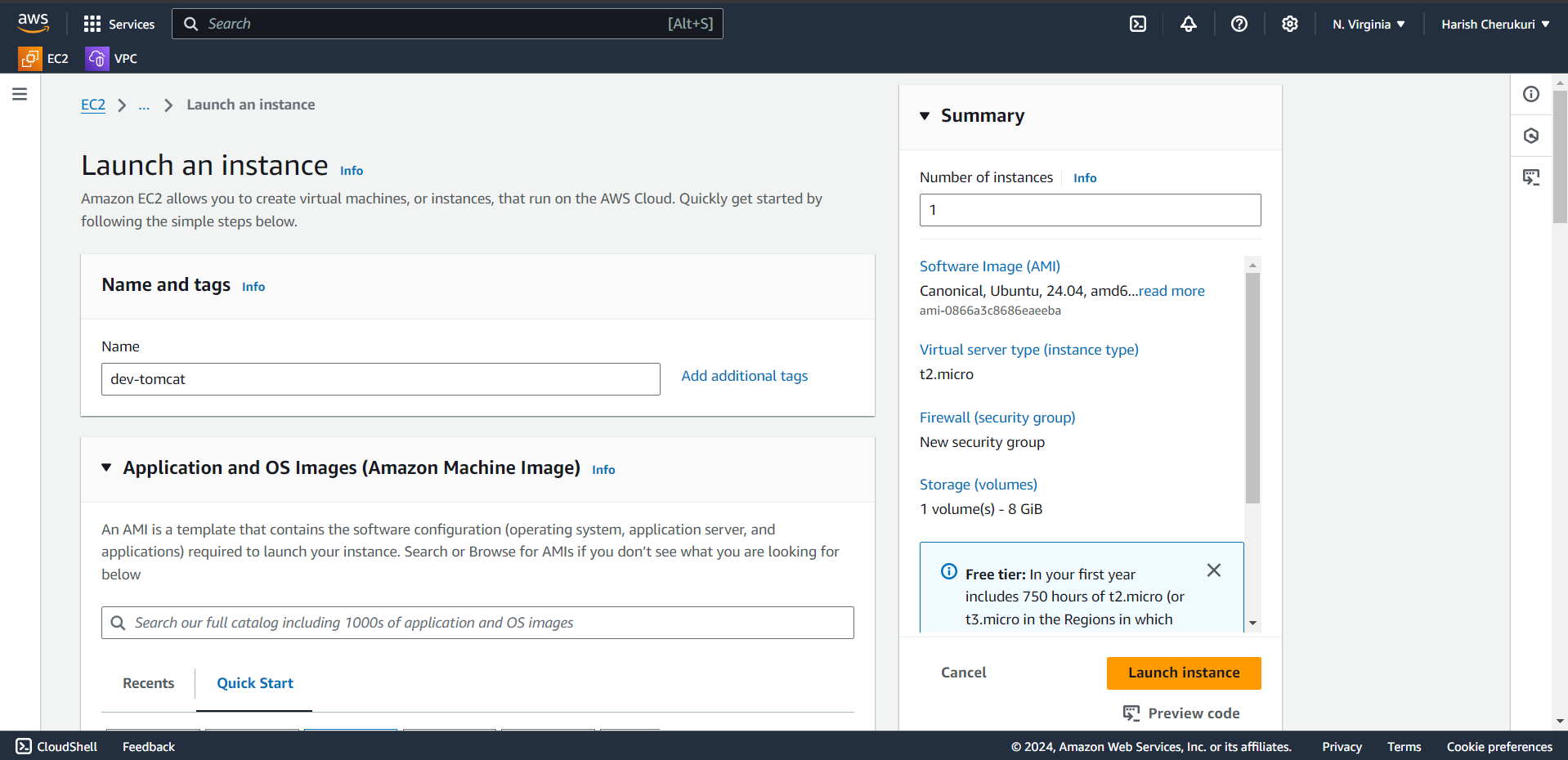
One for production tomcat server.

**Step-1**

First launch 3 tomcat server for dev ,test , production.

Login to aws.

Click on launch instance.



Name it as dev-tomcat.

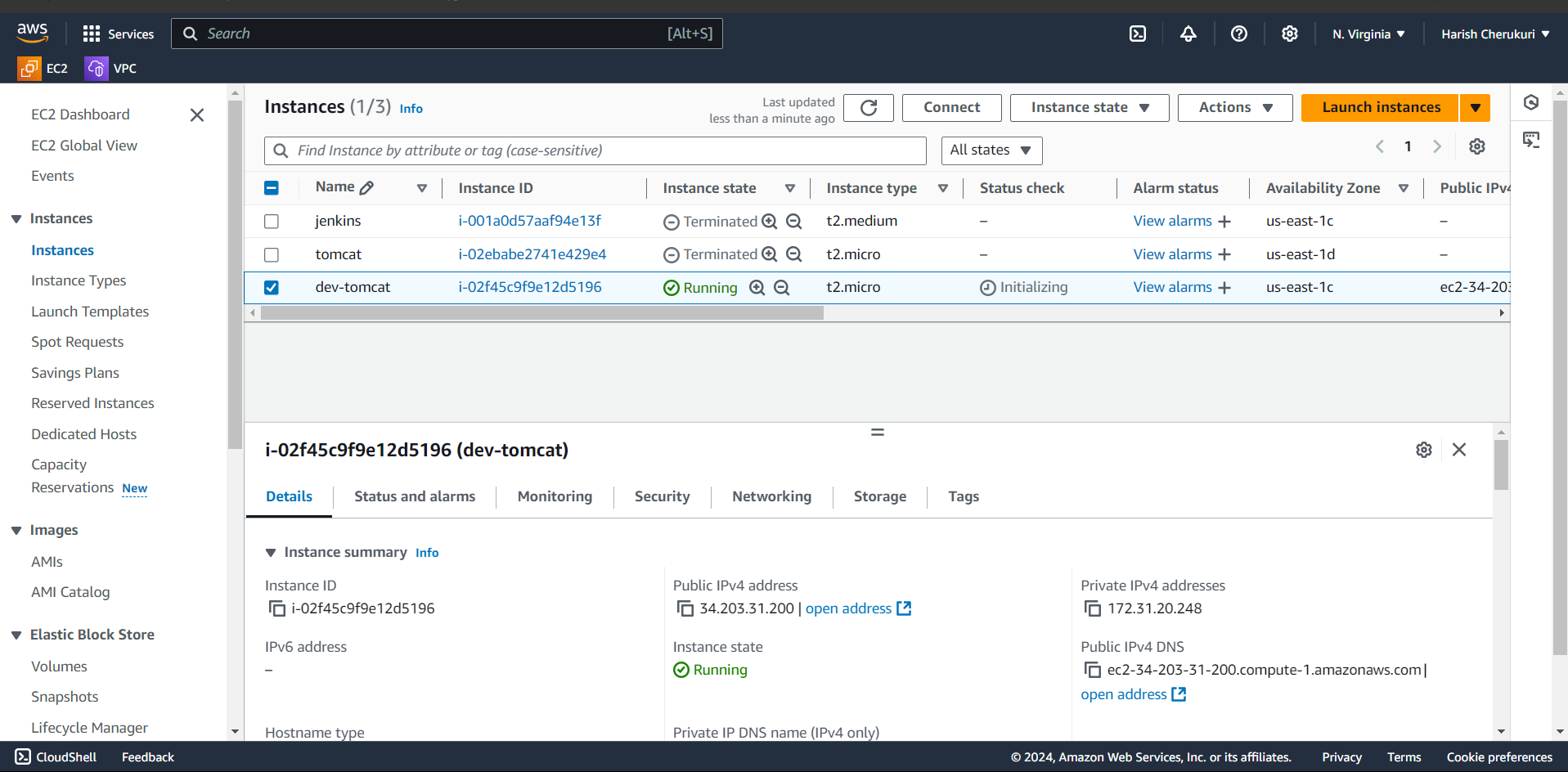
Select the ami id.

Select the instance id.

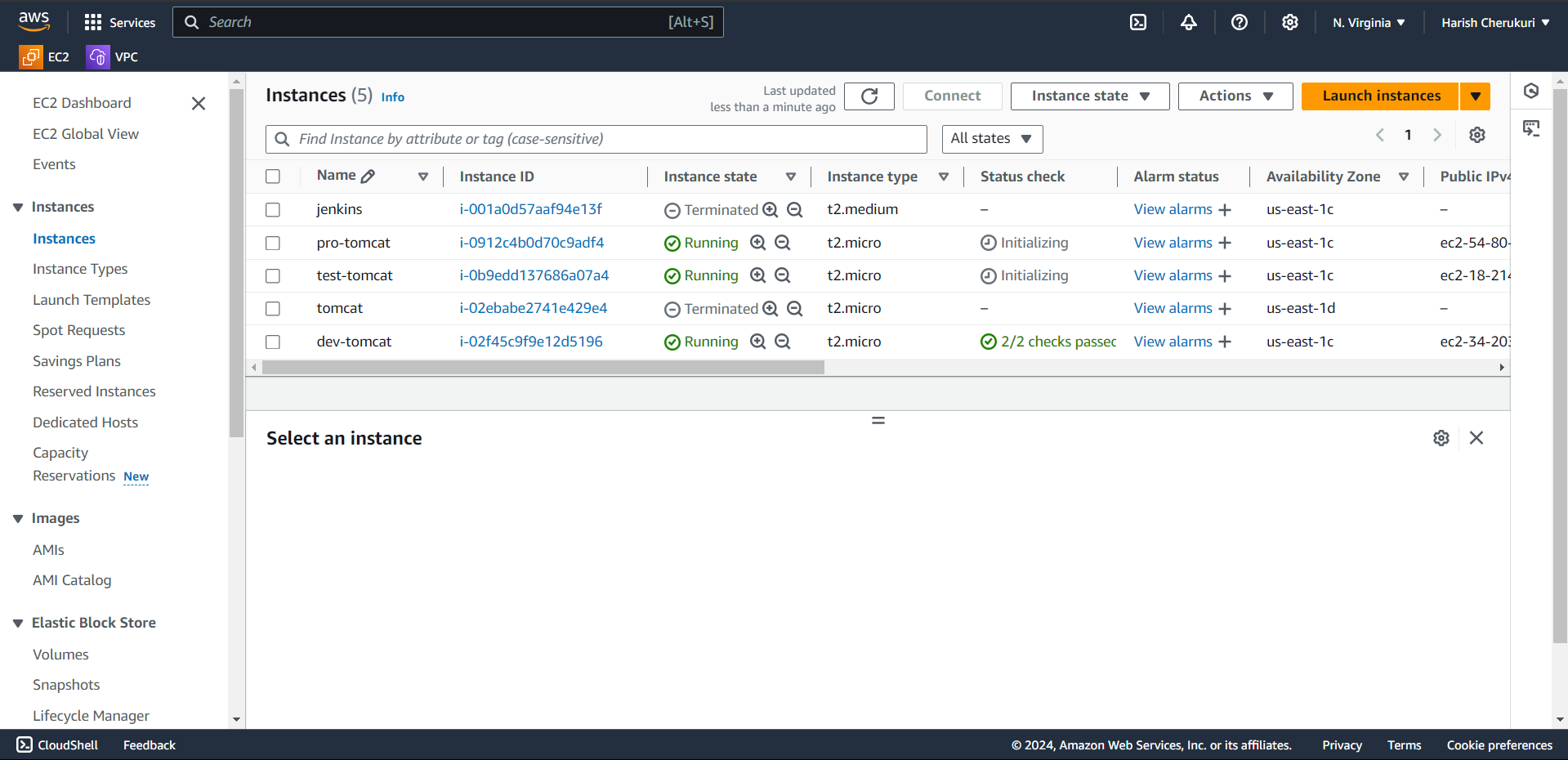
Select the key pair.

In security group go to inbound rules add tomcat port number 8080.

Click launch instance.



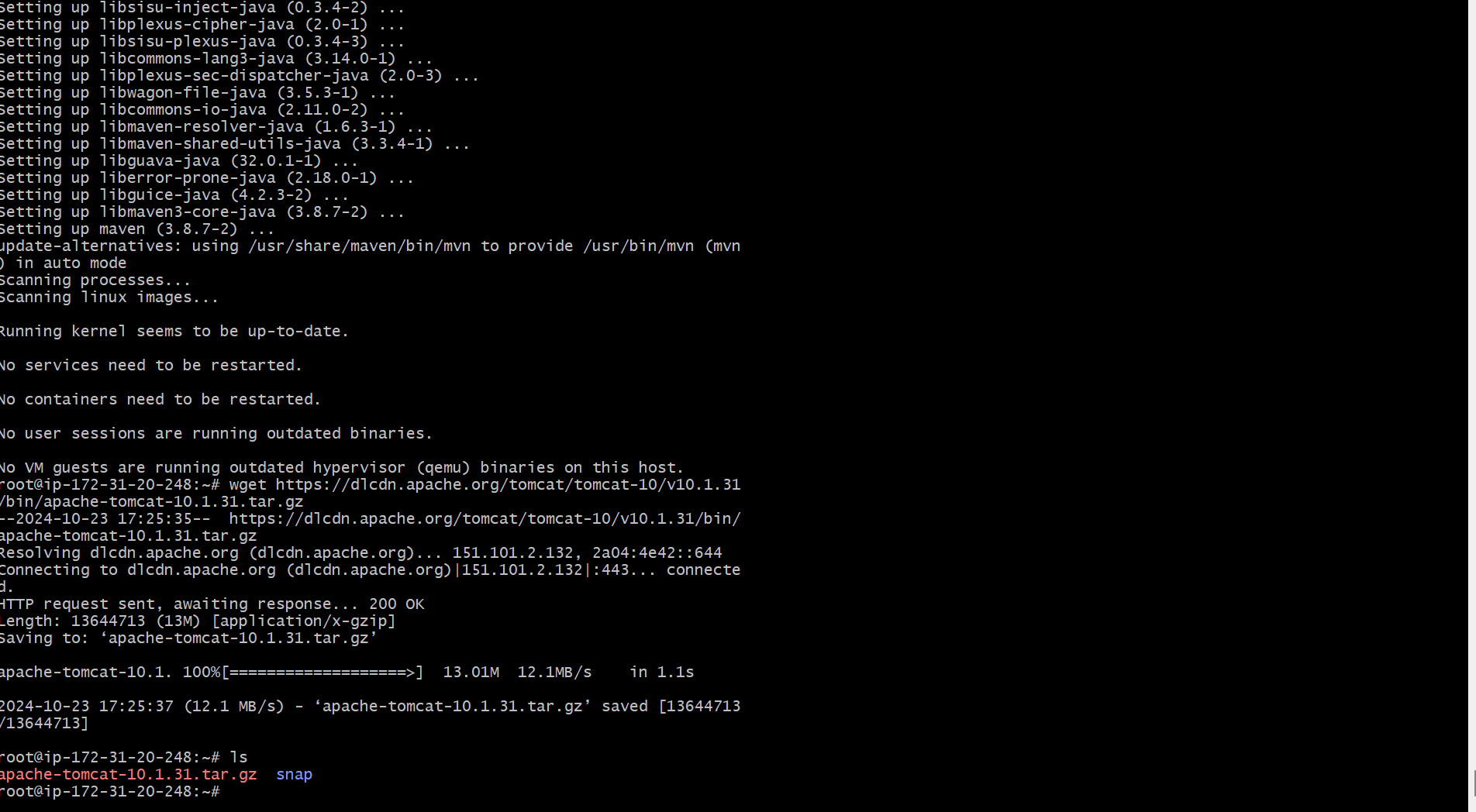
Like this launch remaining two instances.



Now ssh to dev-tomcat server.

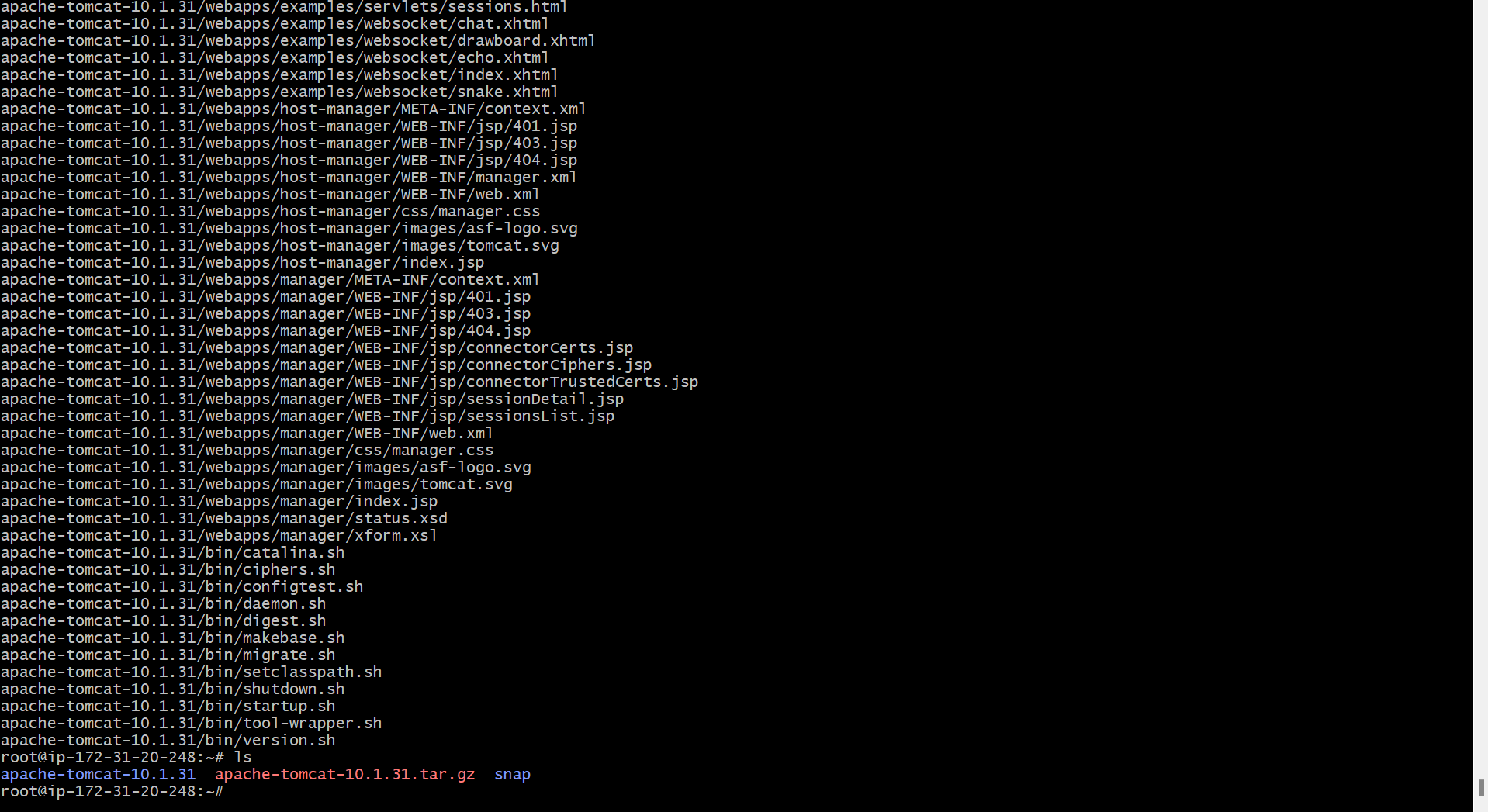
After ssh to server.

Install tomcat ,java in server .

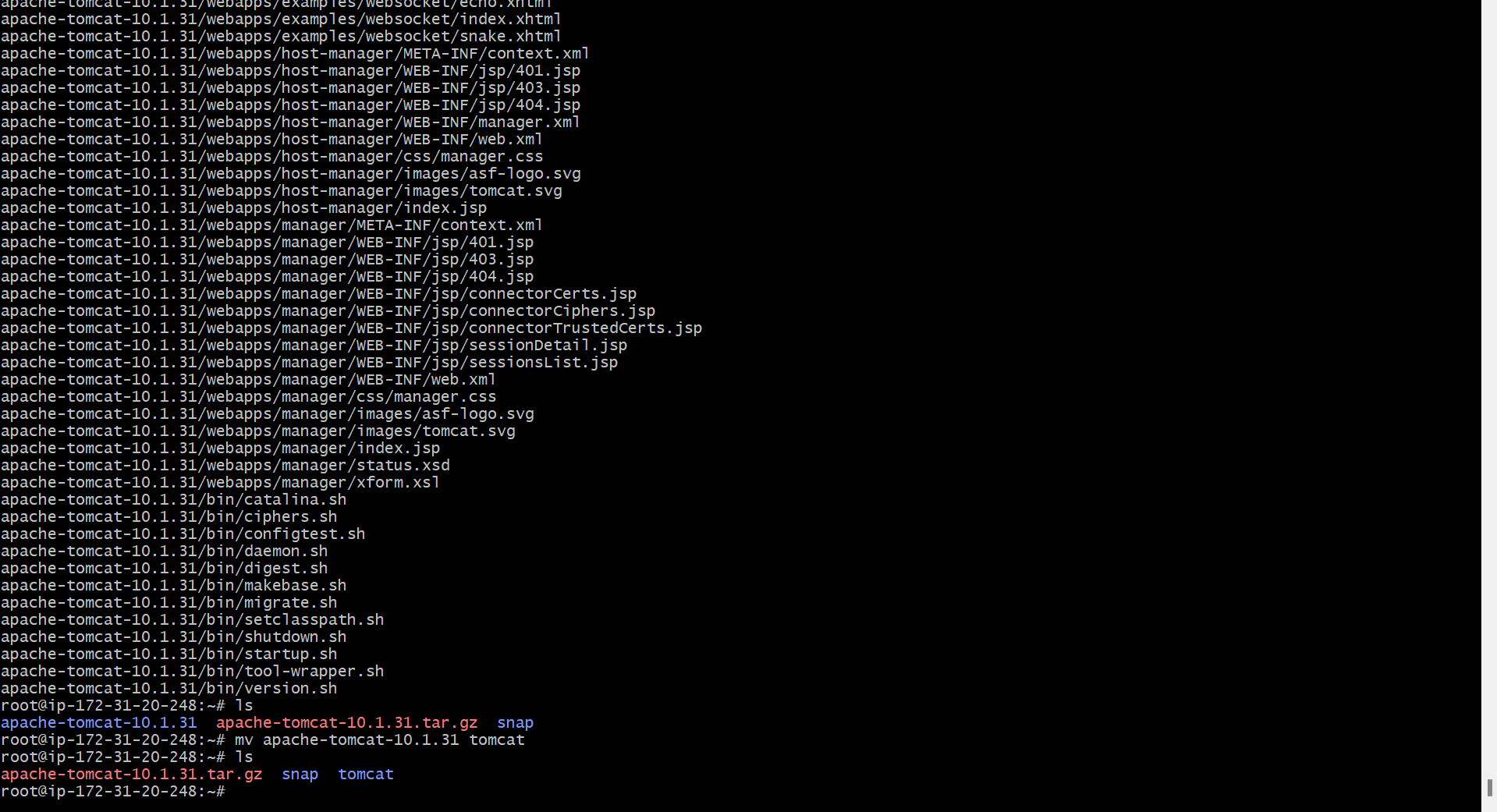


Tomcat is installed now we need to untar the installed file.

By tar -xvzf file name.

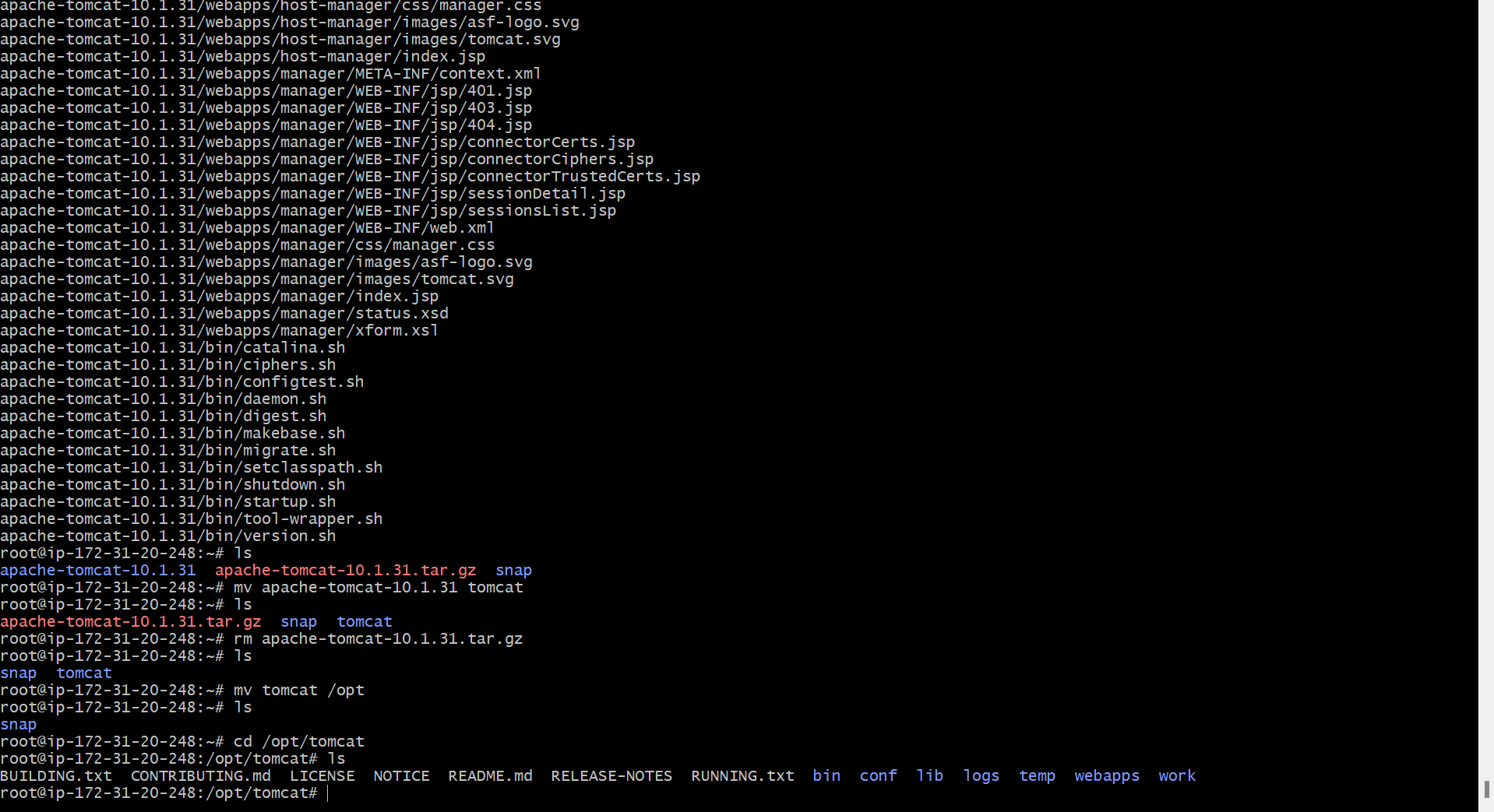


Rename the untared file to tomcat.



Mv the tomcat to /opt folder.

Now go to /opt/tomcat path.



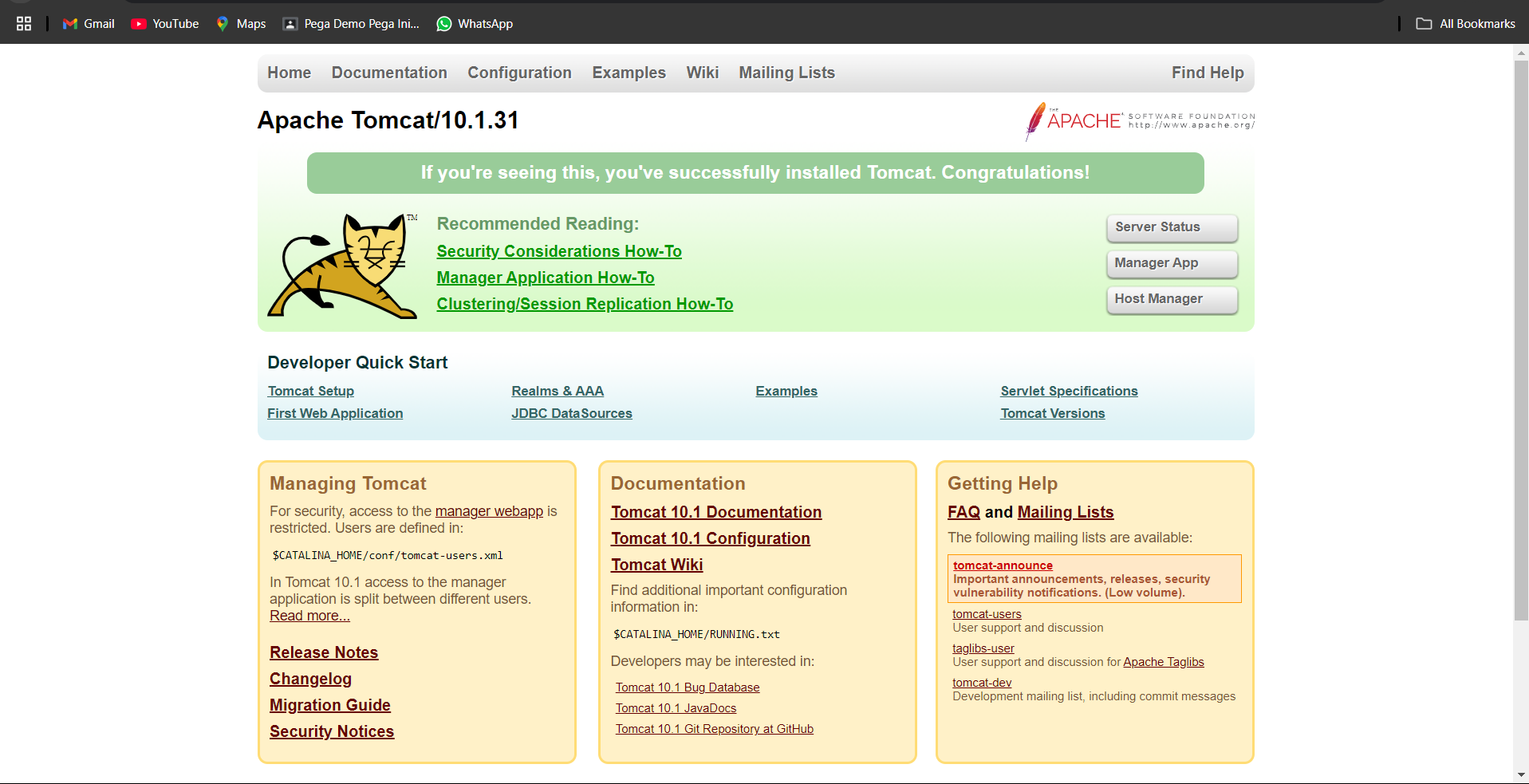
You can find the file and folders inside the tomcat.

Go to bin and startup the tomcat server.

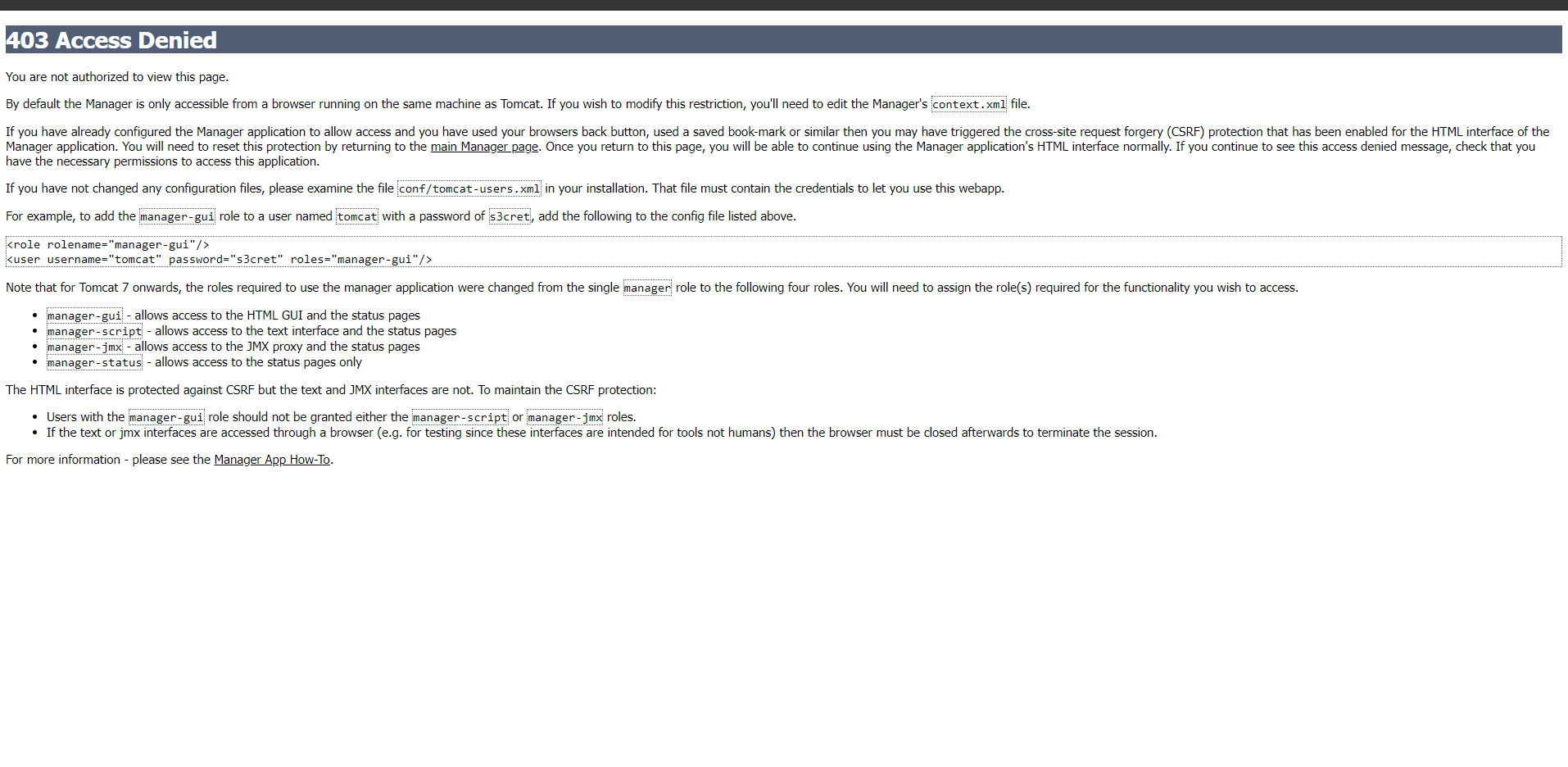
After starting the server copy the dev-tomcat server public ip paste it browser with port number.

34.203.31.200:8080

It will open the tomcat server.



Now if click on any button you may got the 403 error.



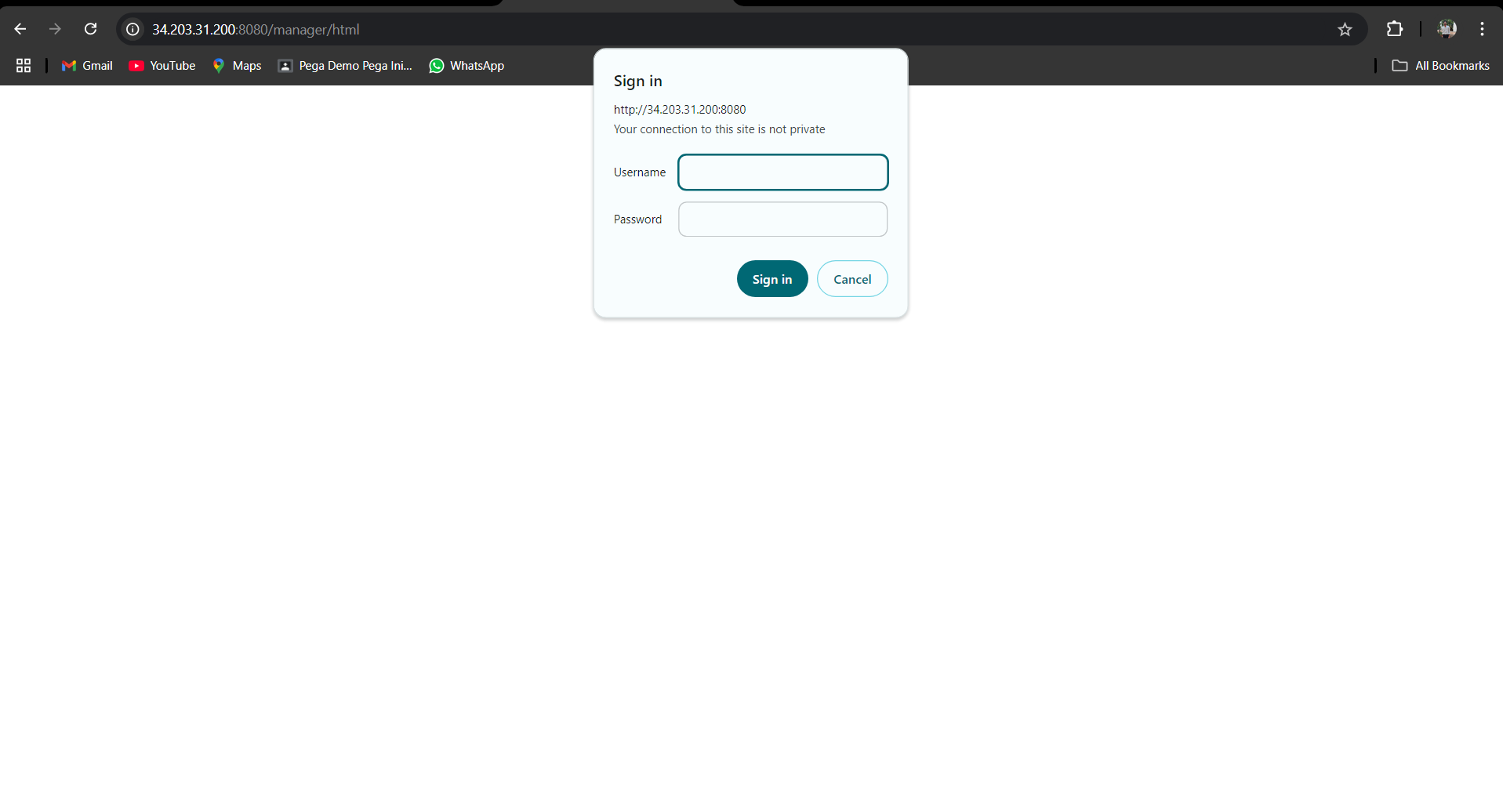
To over come this we need change some correction in host-manager,manager and also adding the permissions in conf.

<role rolename="manager-gui"/>

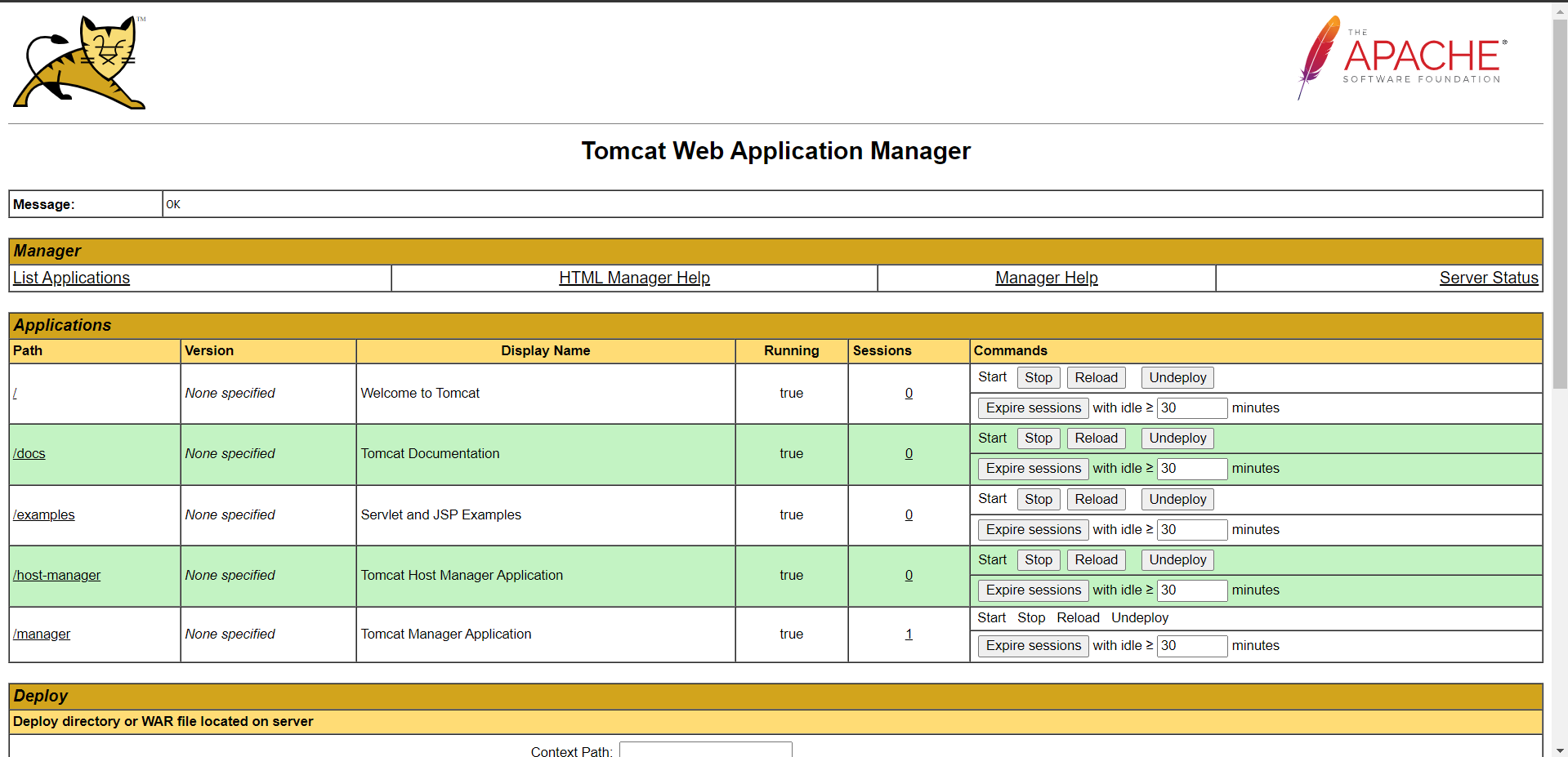
<role rolename="manager-script"/>

<user username="admin" password="admin" roles="manager-gui"/>

<user username="dev" password="dev" roles="manager-script"/>



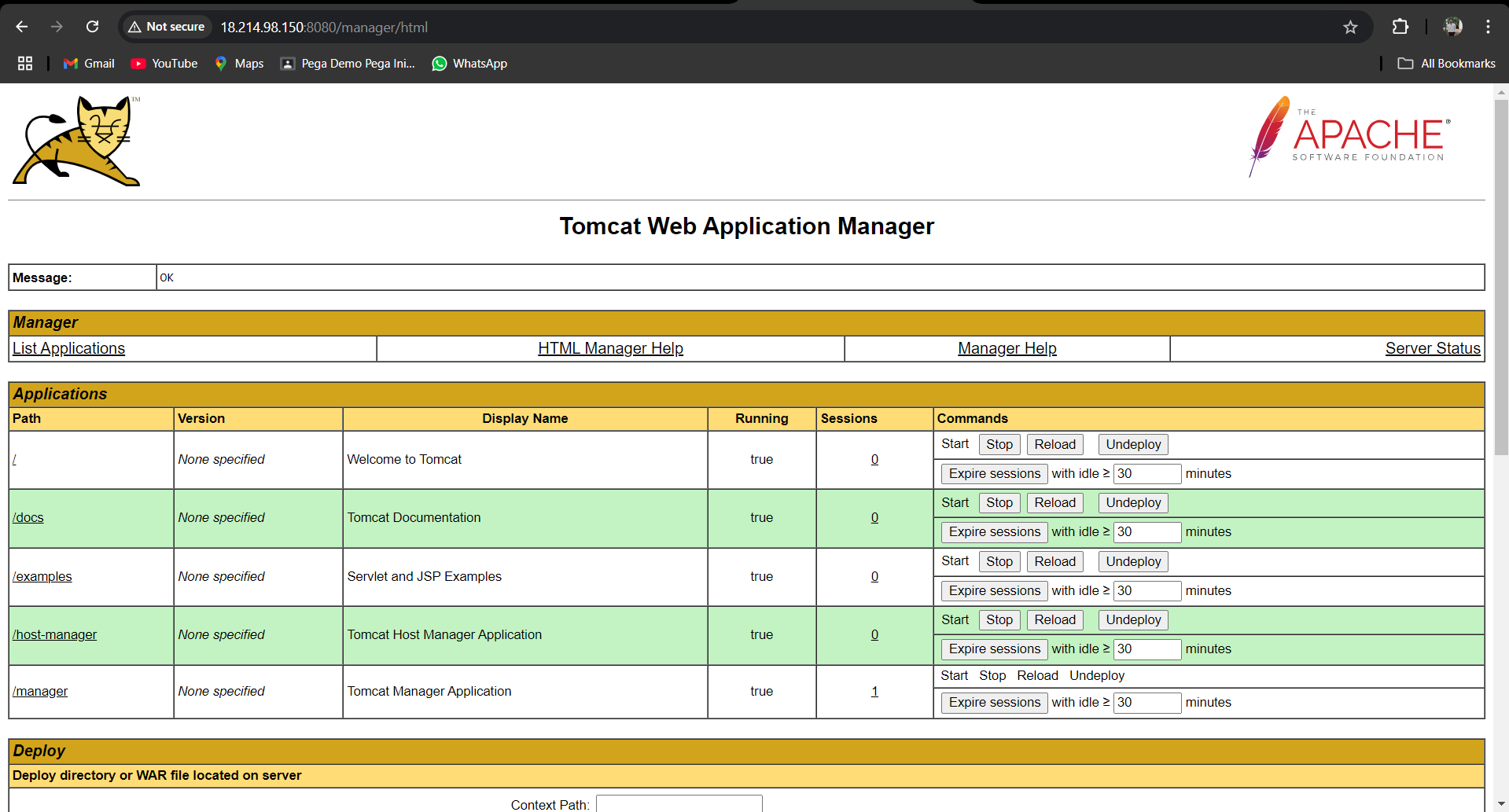
Give the login details.



We enter to tomcat server.

Like this do it for test,pro tomcat servers.

We got the test tomcat server.



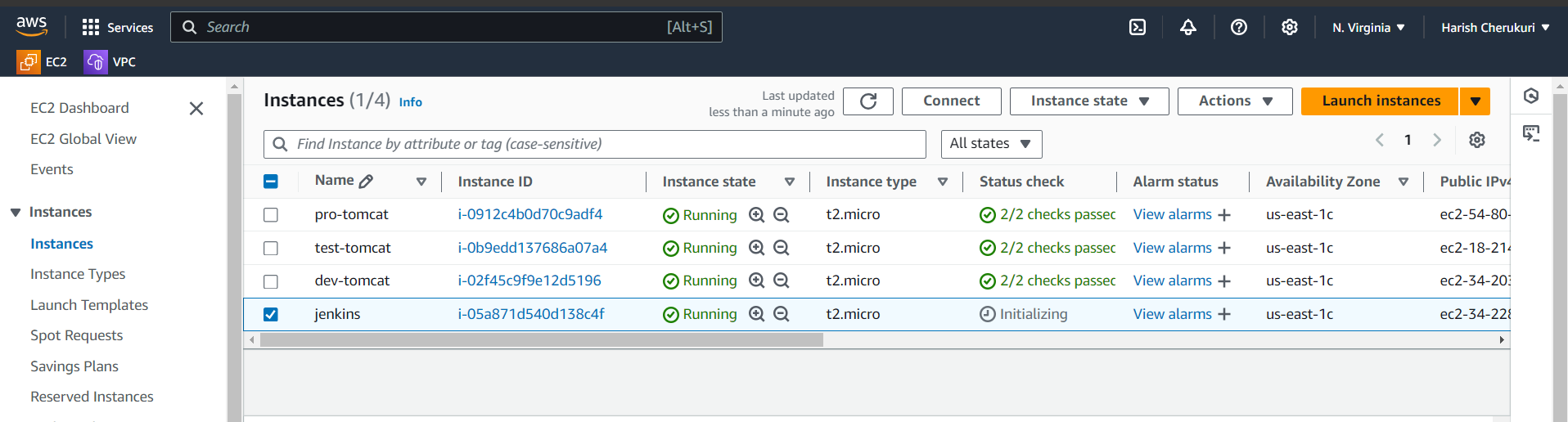
Now pro-tomcat server.

We got the pro-tomcat server.



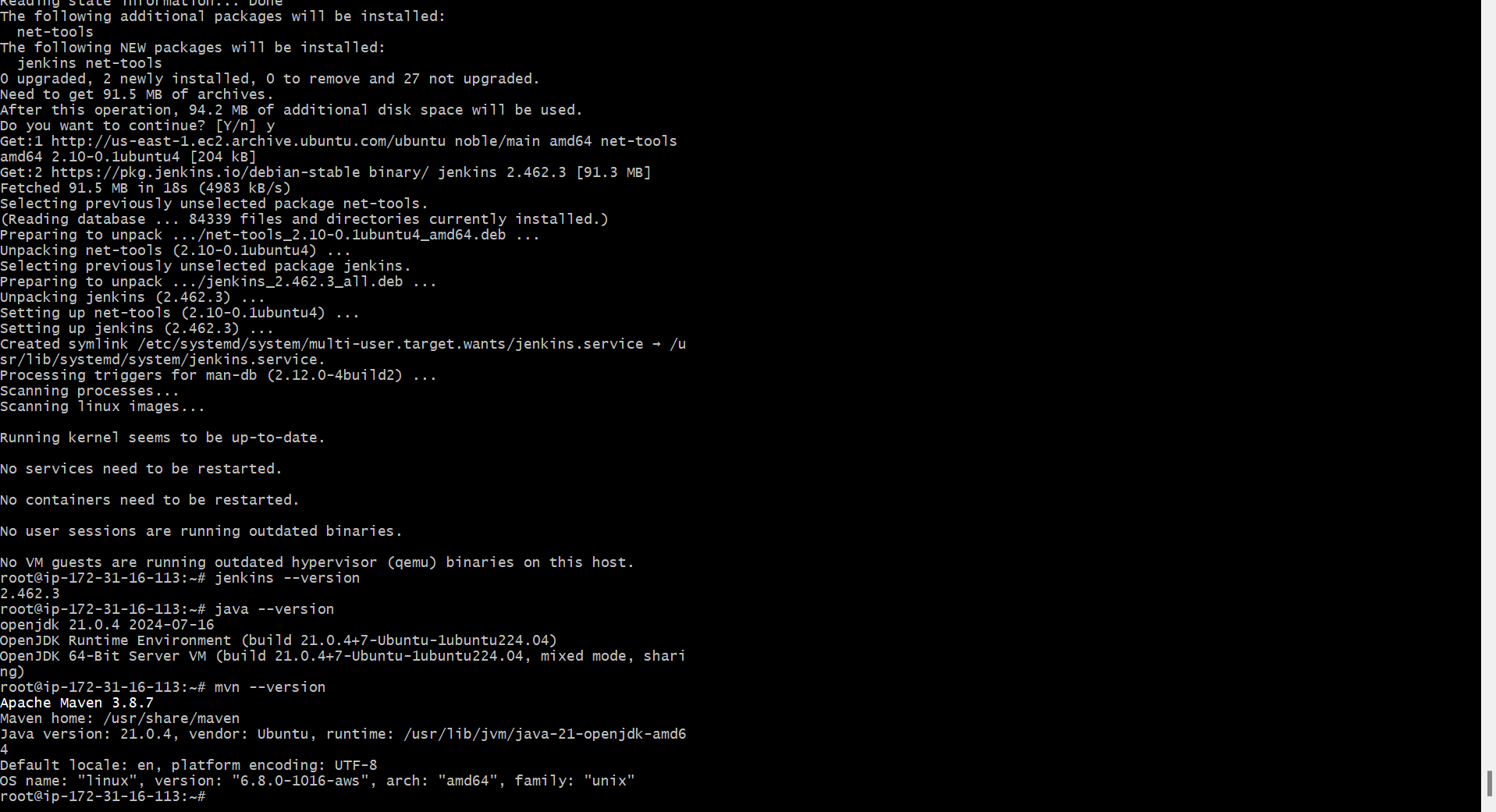
**Step-2**

launch the Jenkins server.



Ssh to Jenkins server.

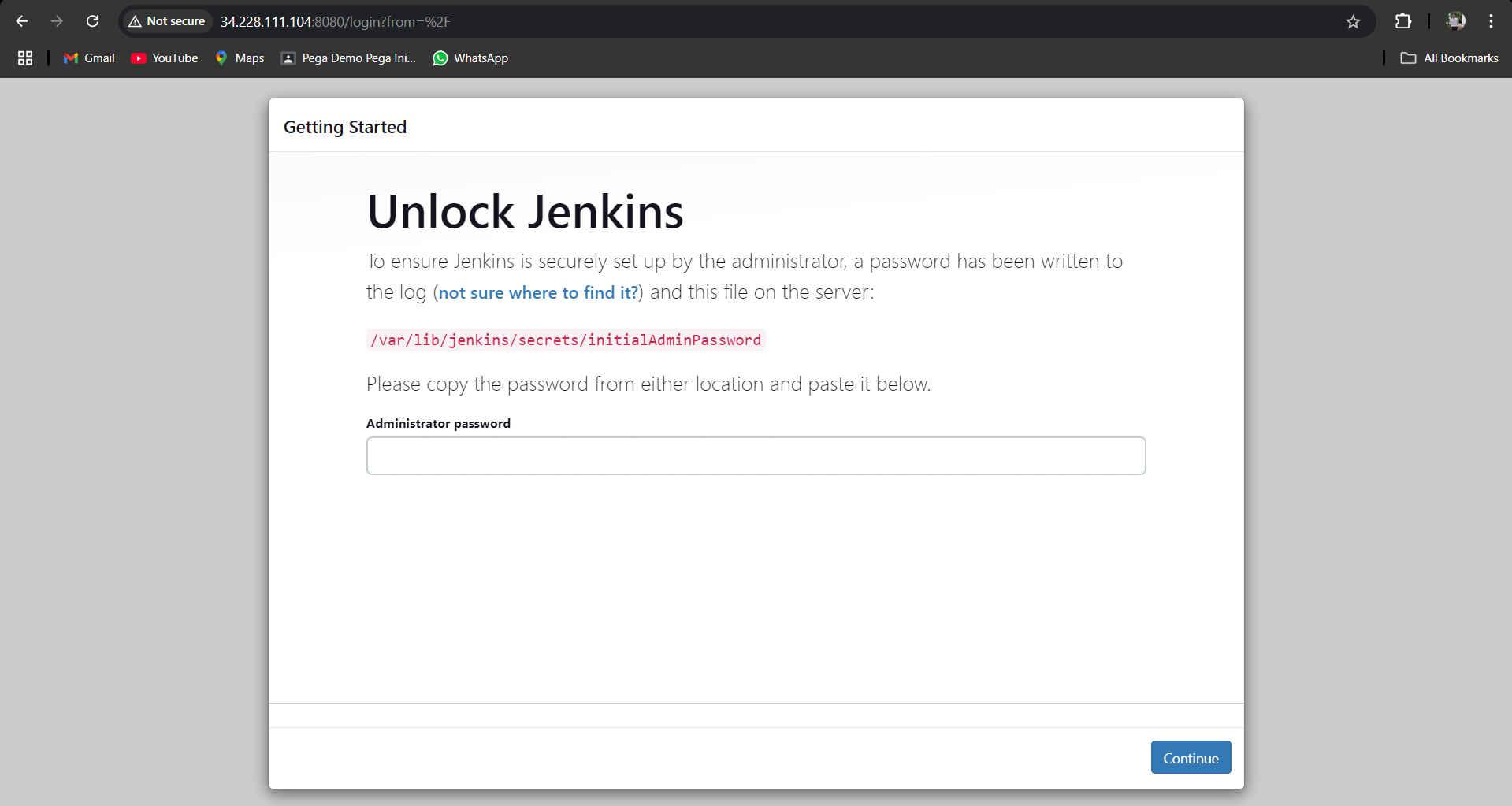
Install java , maven and Jenkins.



Installation is over.

Paste the Jenkins server public ip and it port number in browser.

34.228.111.104:8080.

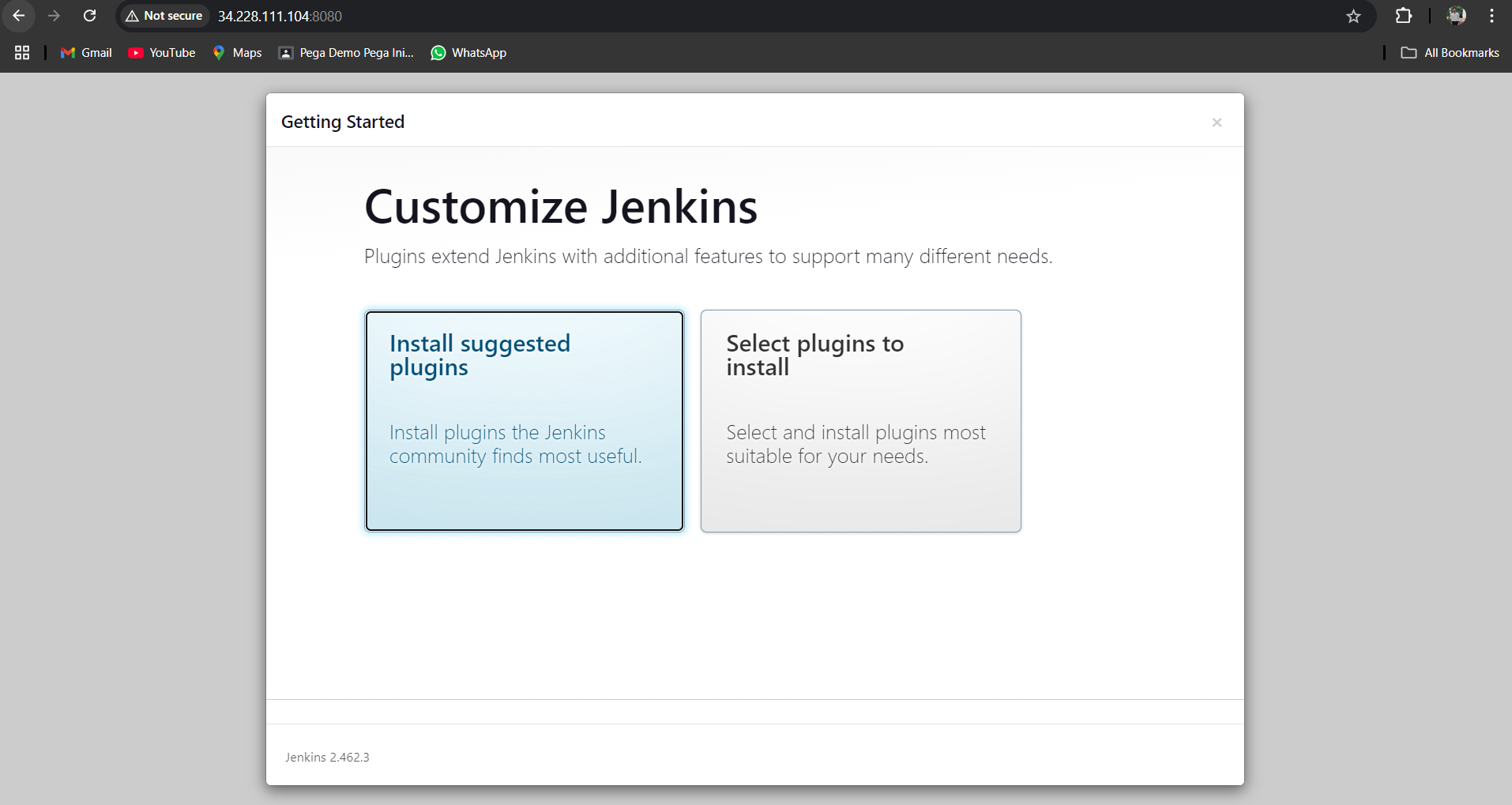


We got like this to enter into Jenkins we need password.

To know that go to that path we can find the password.

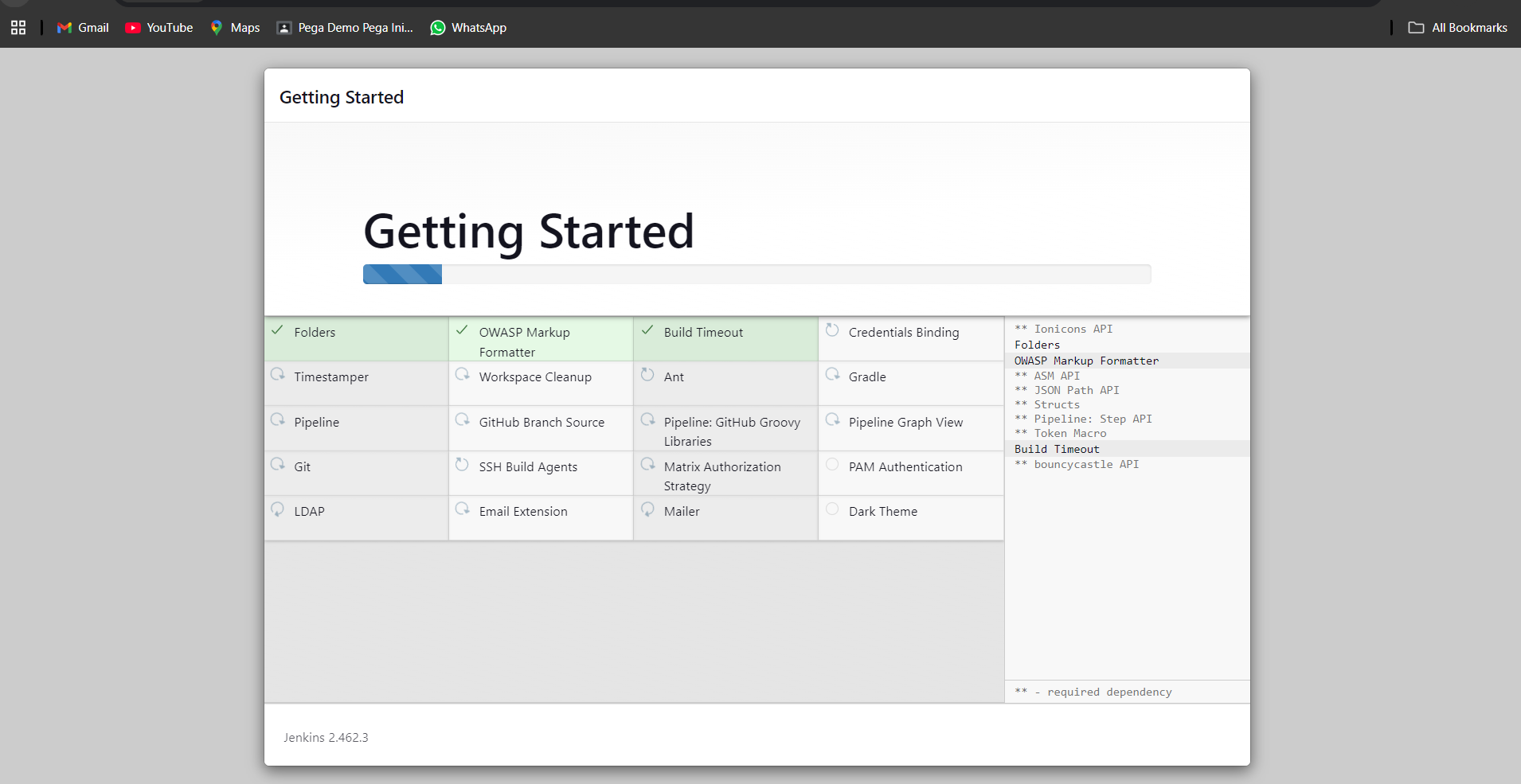
By using cat command.

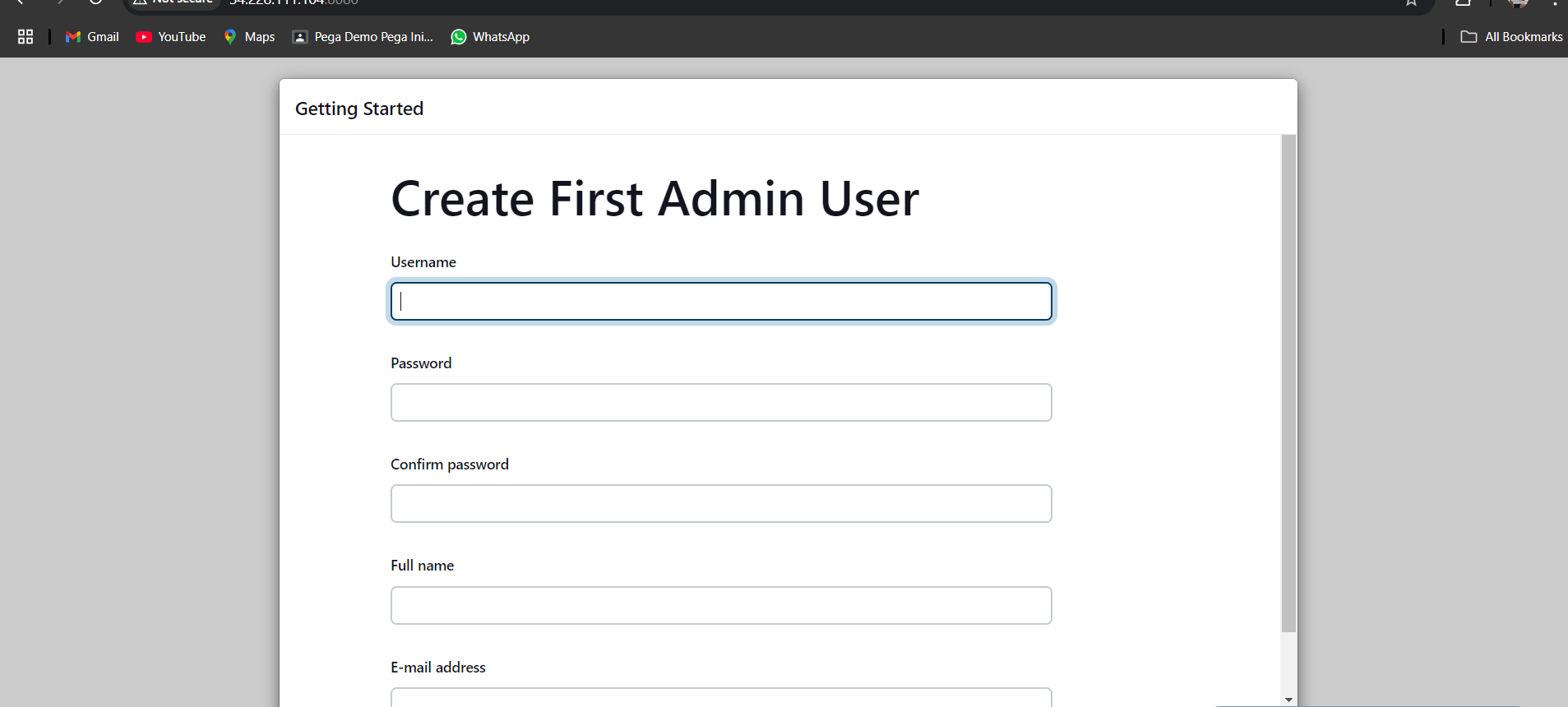
Click on continue.



Click on install suggested plugins.

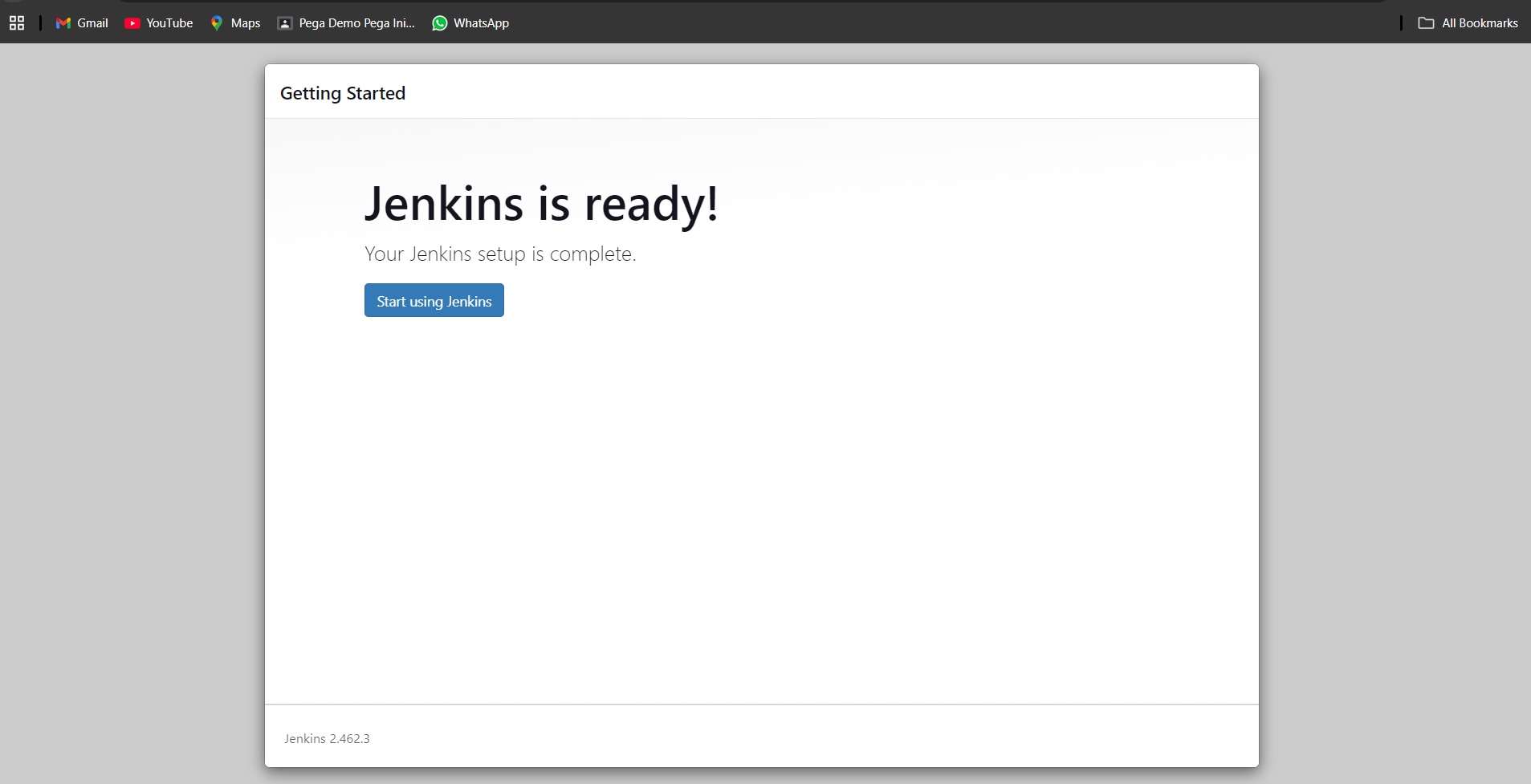
It will going to install the plugins .





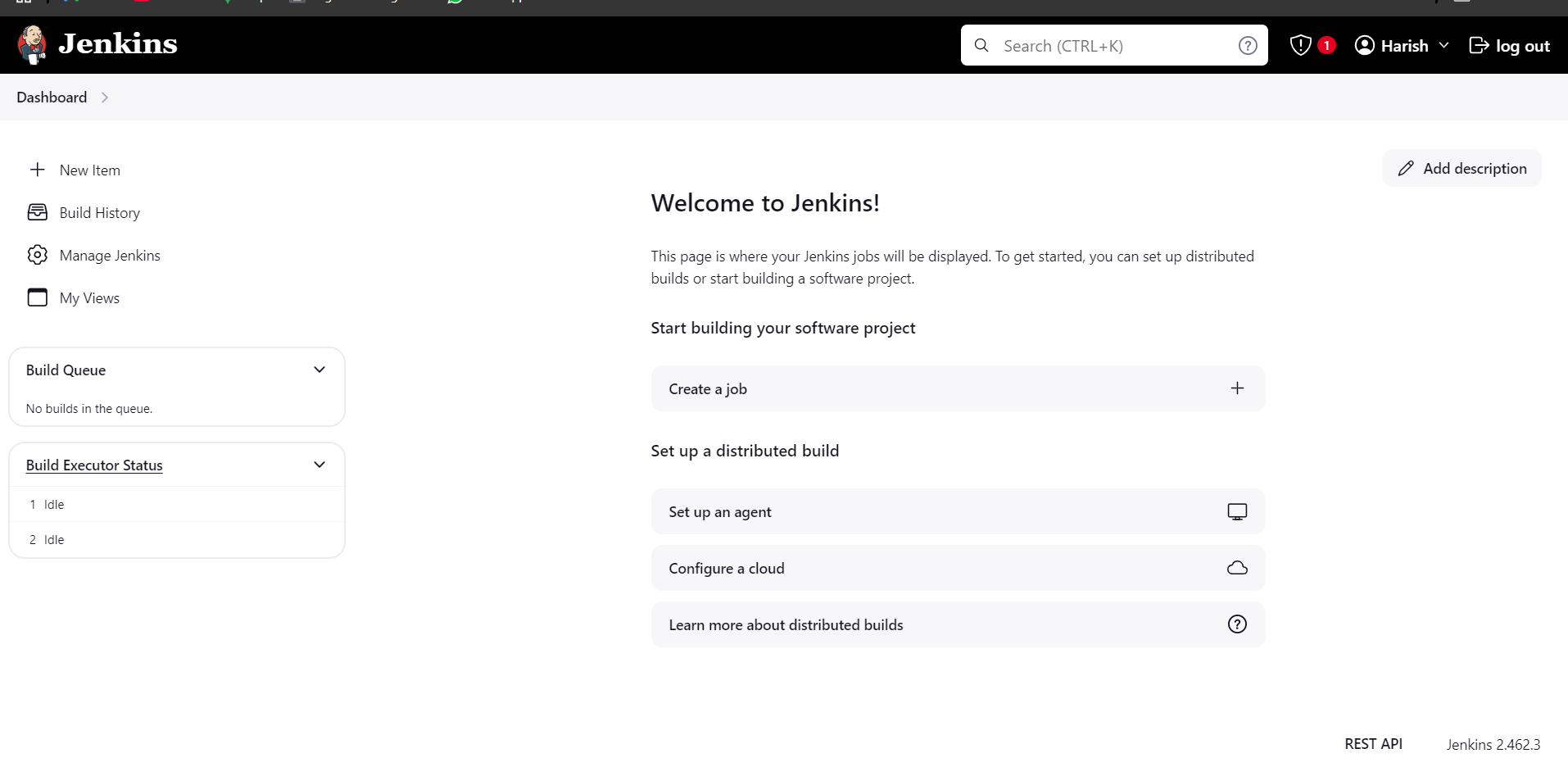
Give username , password , and email id.

Click on save and continue.



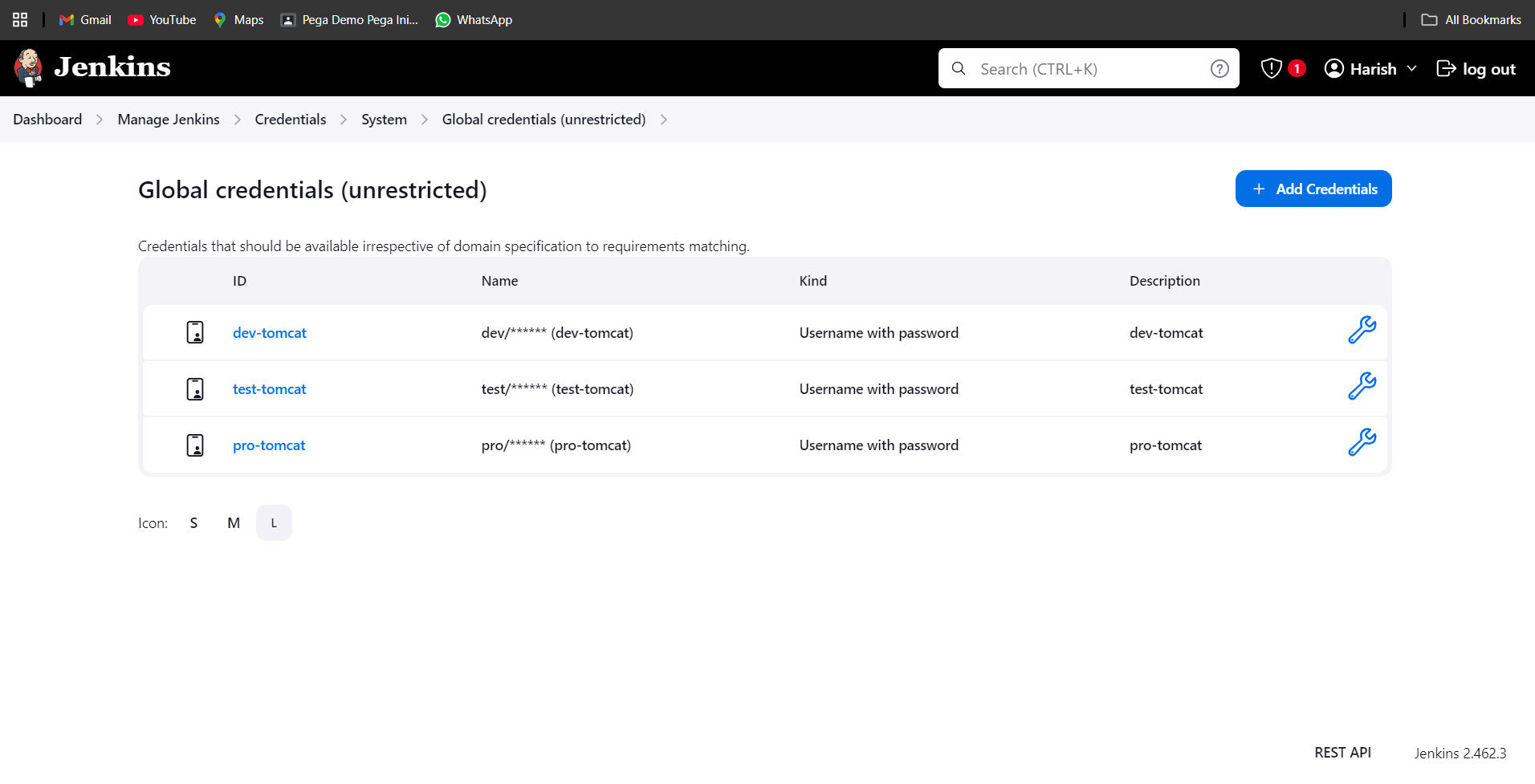
Our Jenkins is ready.

Click on start using Jenkins.



Jenkins dashboard looks like this.

Now first add the dev,test,pro tomcat server username and password in Jenkins credentials.



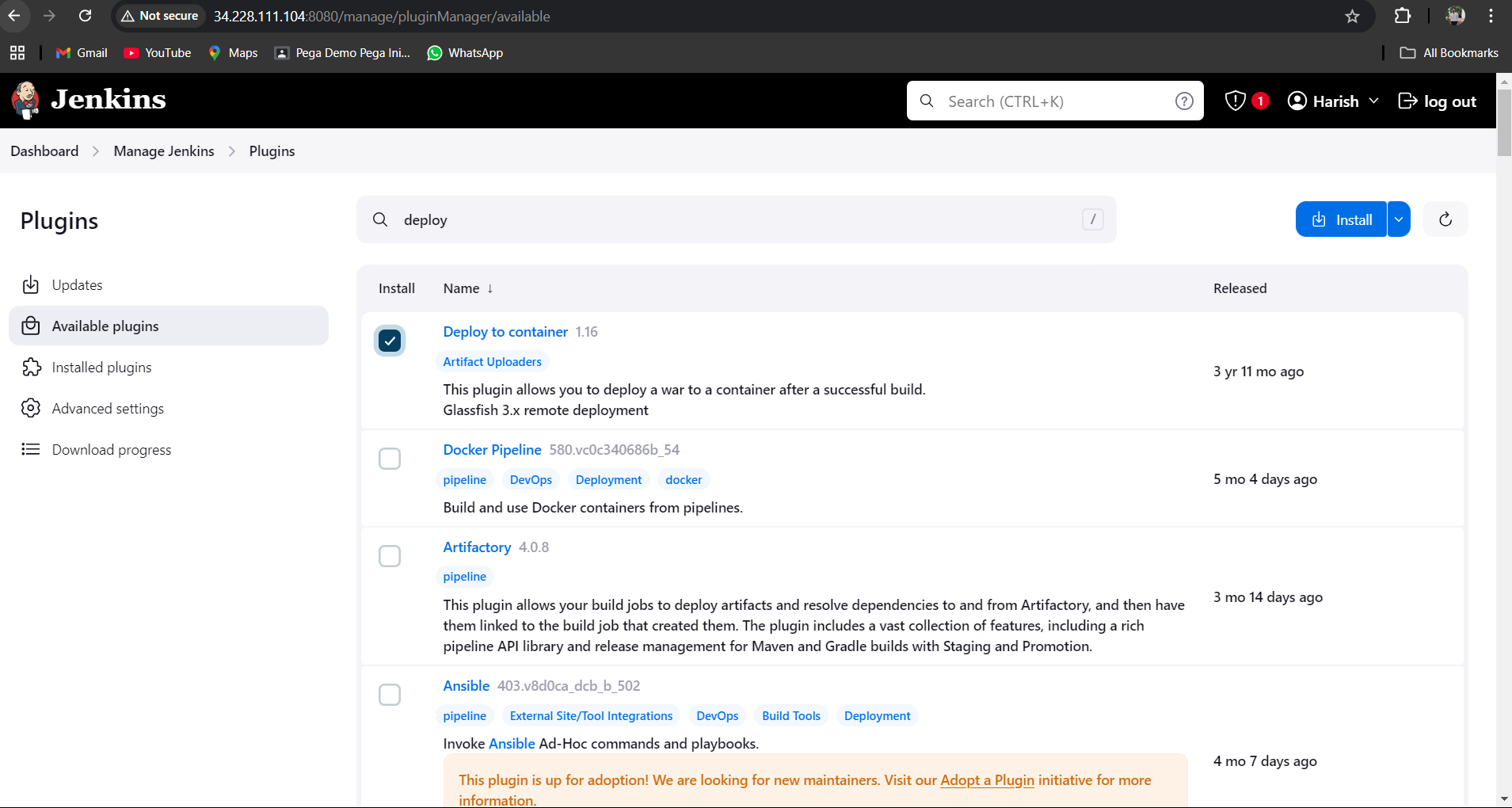
Credentials is added.

To deploy application in web server we need one plugin to install.

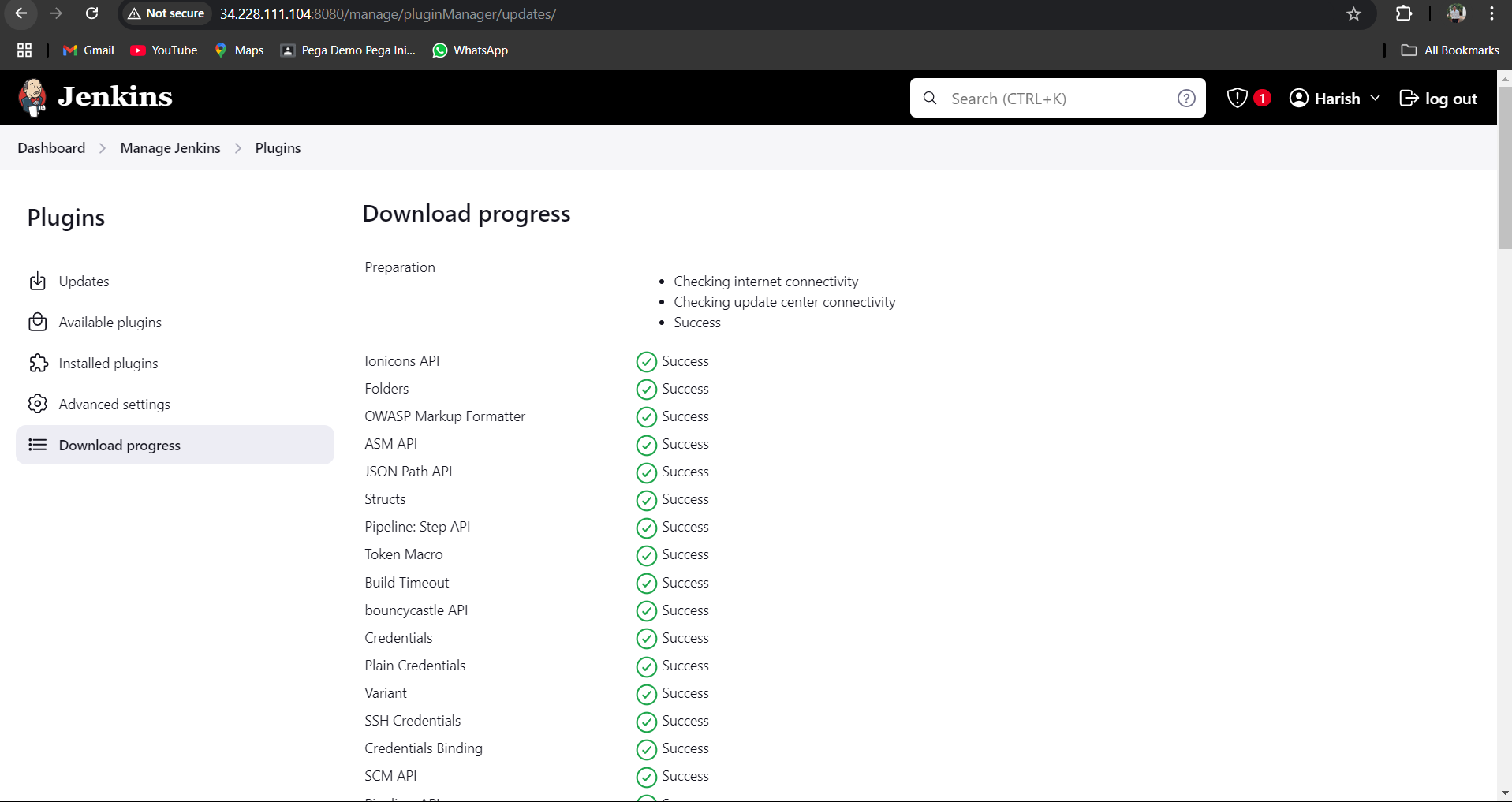
Go to manage Jenkins – plugins – available plugins.

Search for deploy to container and install.

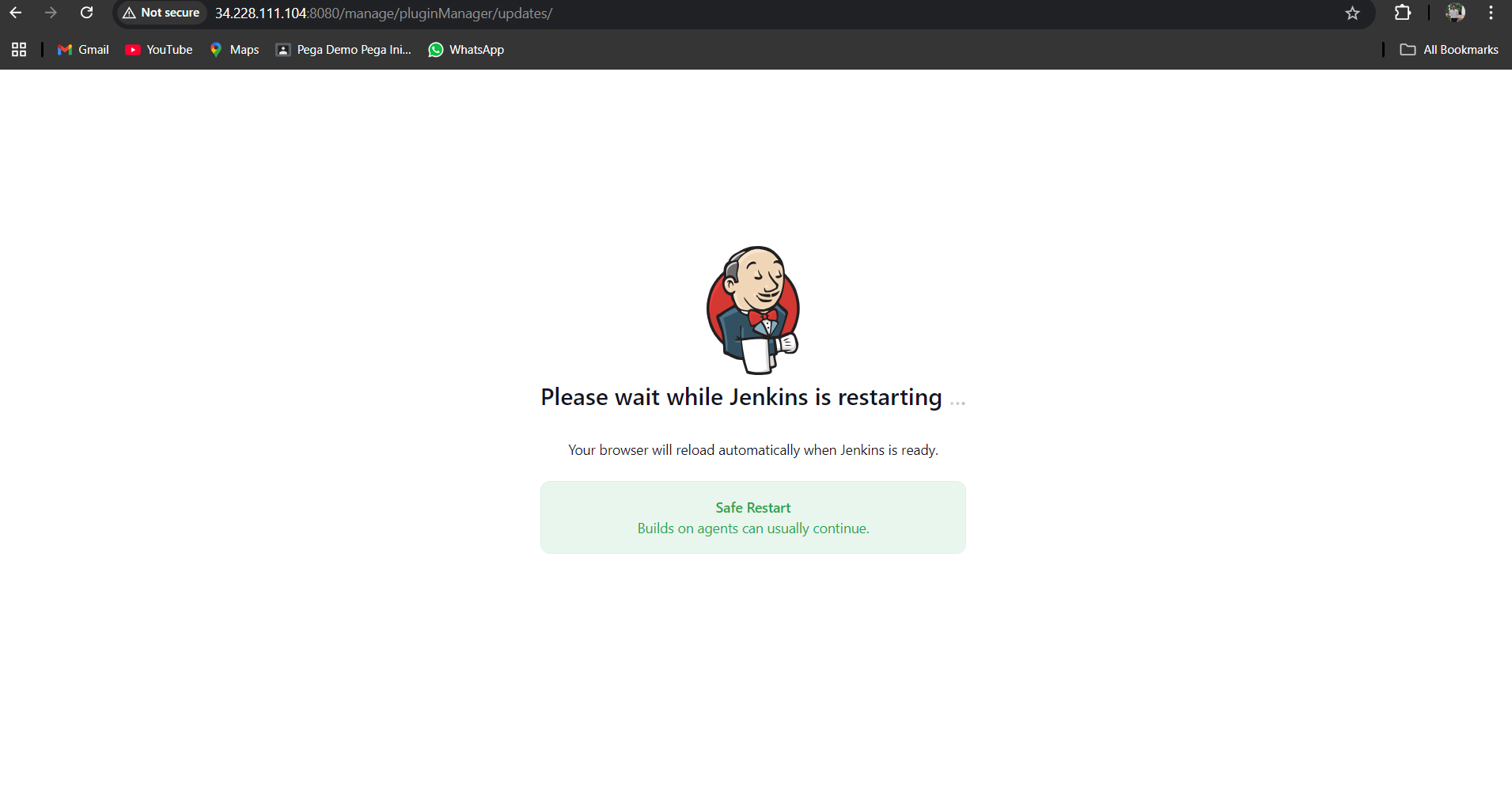
After installation is completed restart the Jenkins.



Click on install.

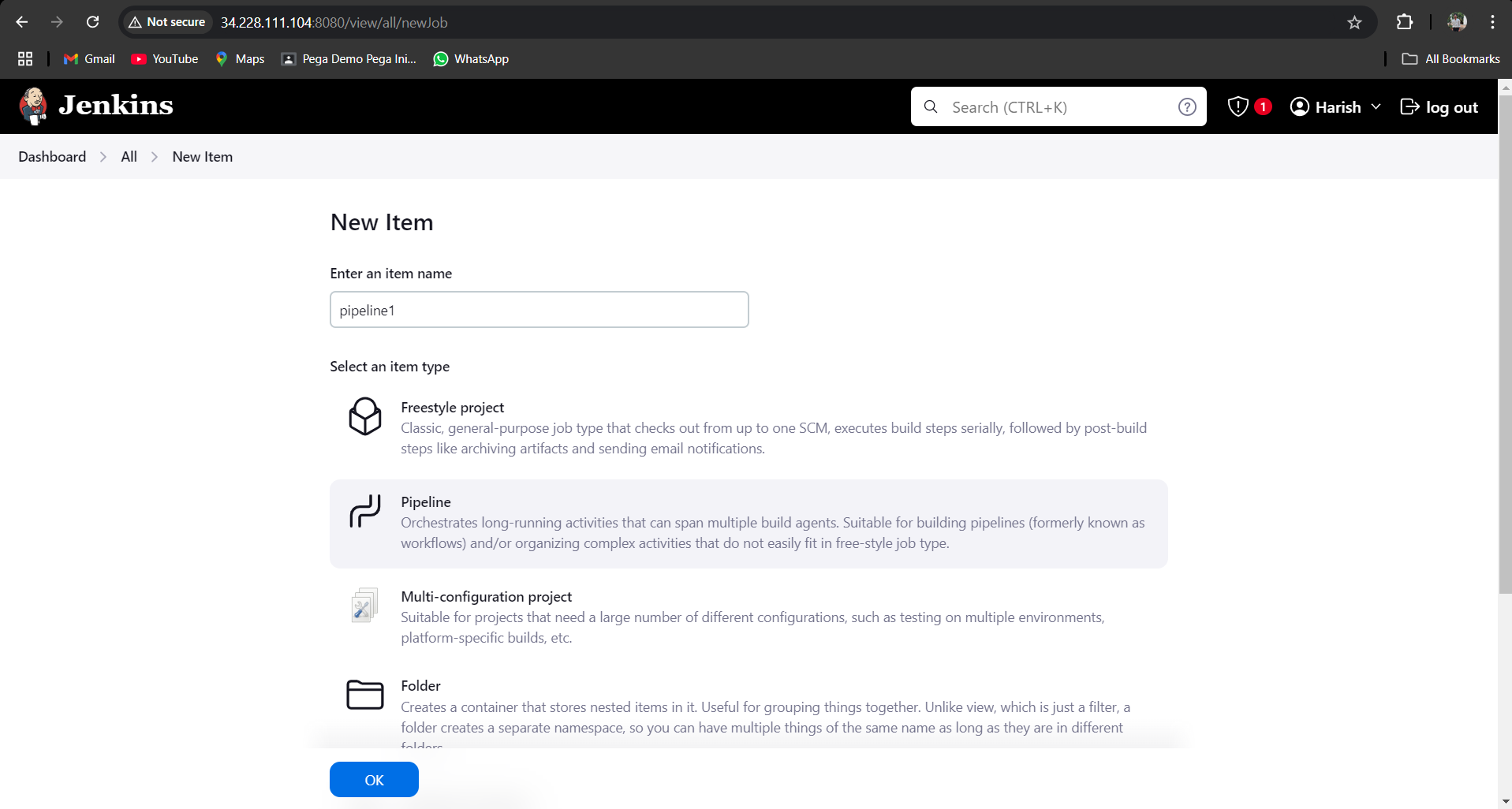


Installing the plugin.



Restart the Jenkins.

Now create the new item.

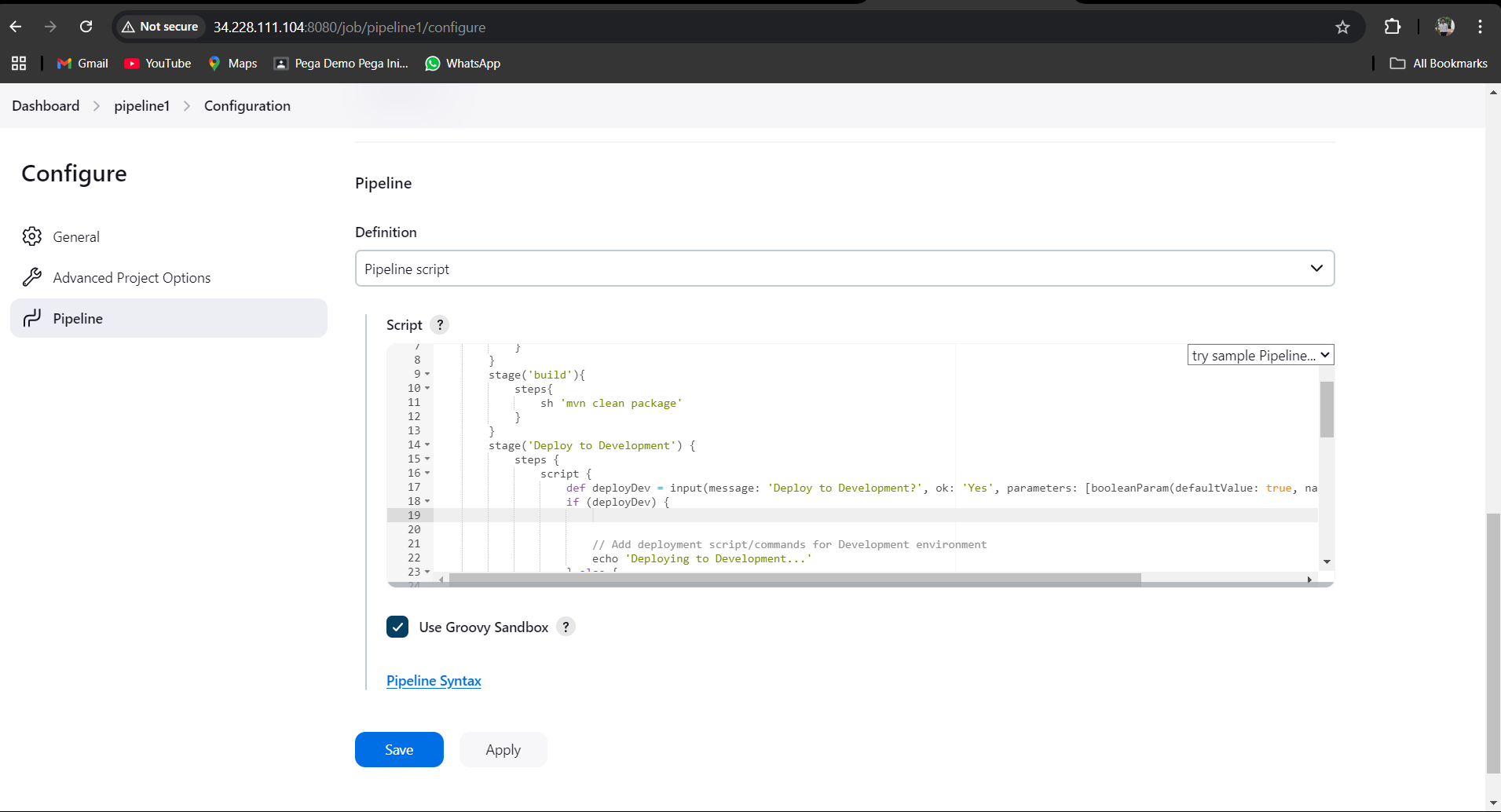


Give the pipeline name.

Select the pipeline click on ok.



Here we have to write the CICD pipeline to build and deploy.



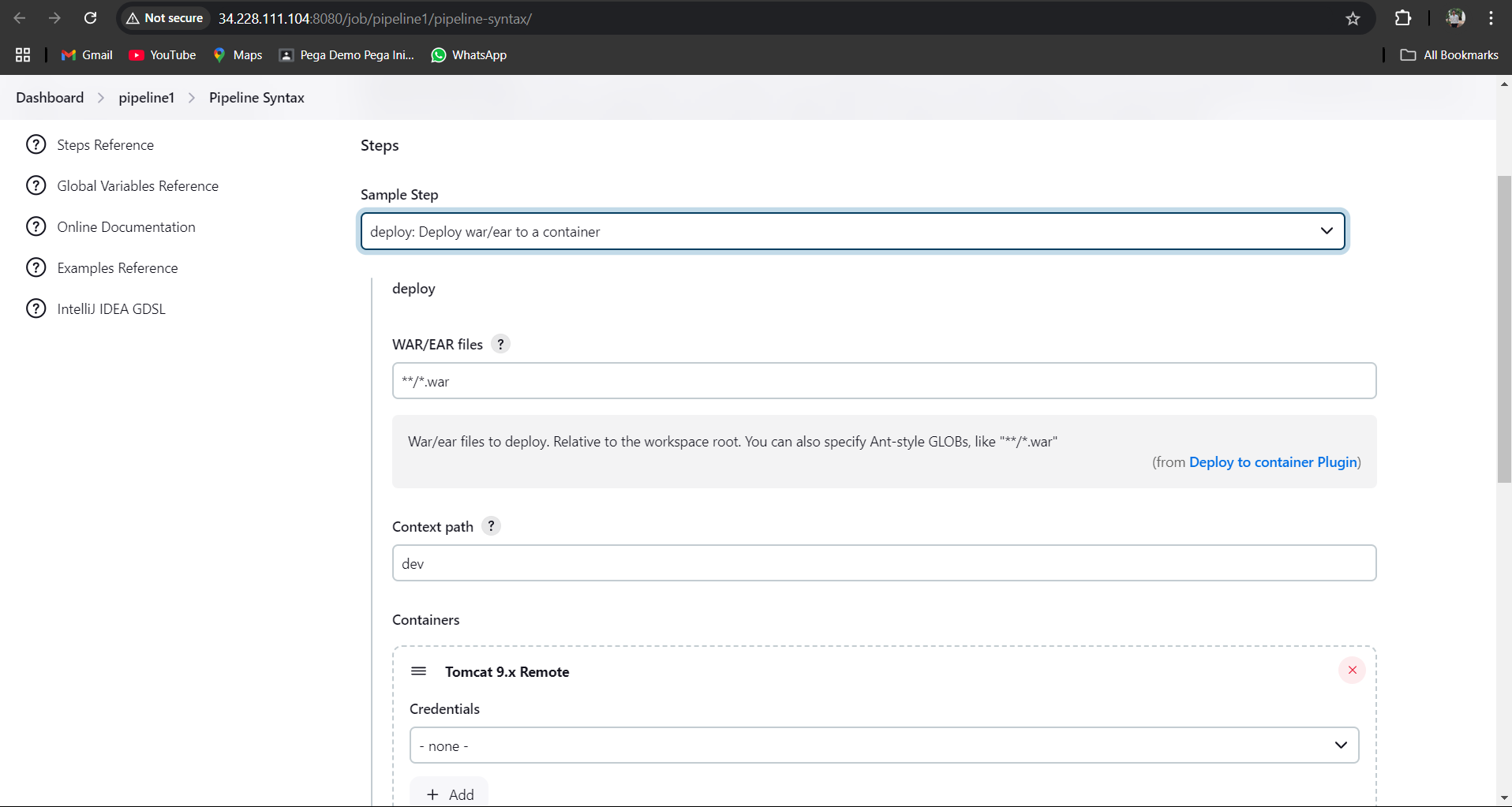
We cloning the code from git by using checkout from version control plugin.

It will generate the syntax paste it in pipeline.

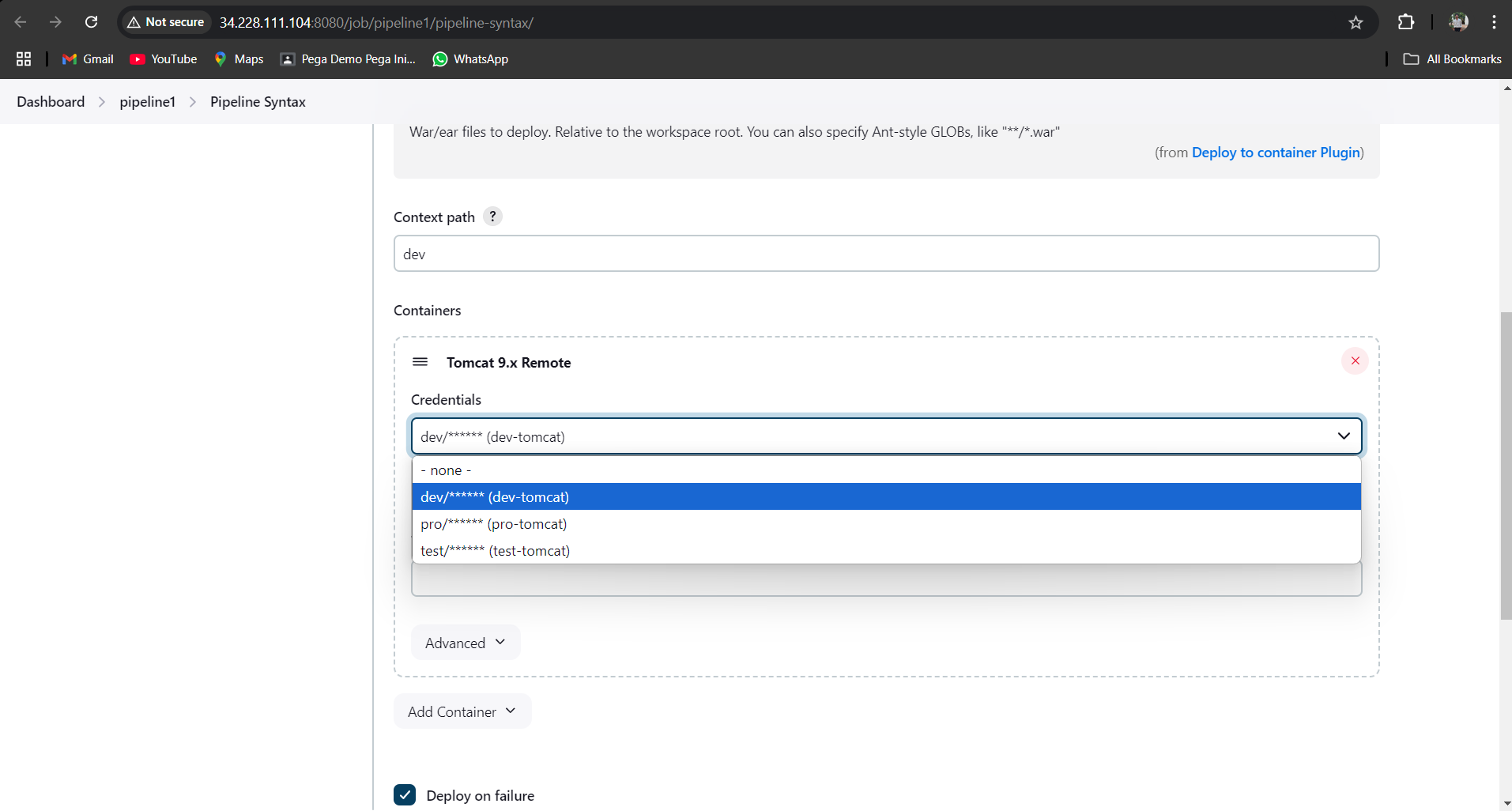
Then build and package process by maven tool. It generate the .war file.

After the war file is generated now we want to deploy it to web server.

For that syntax we are using deploy to container plugin.

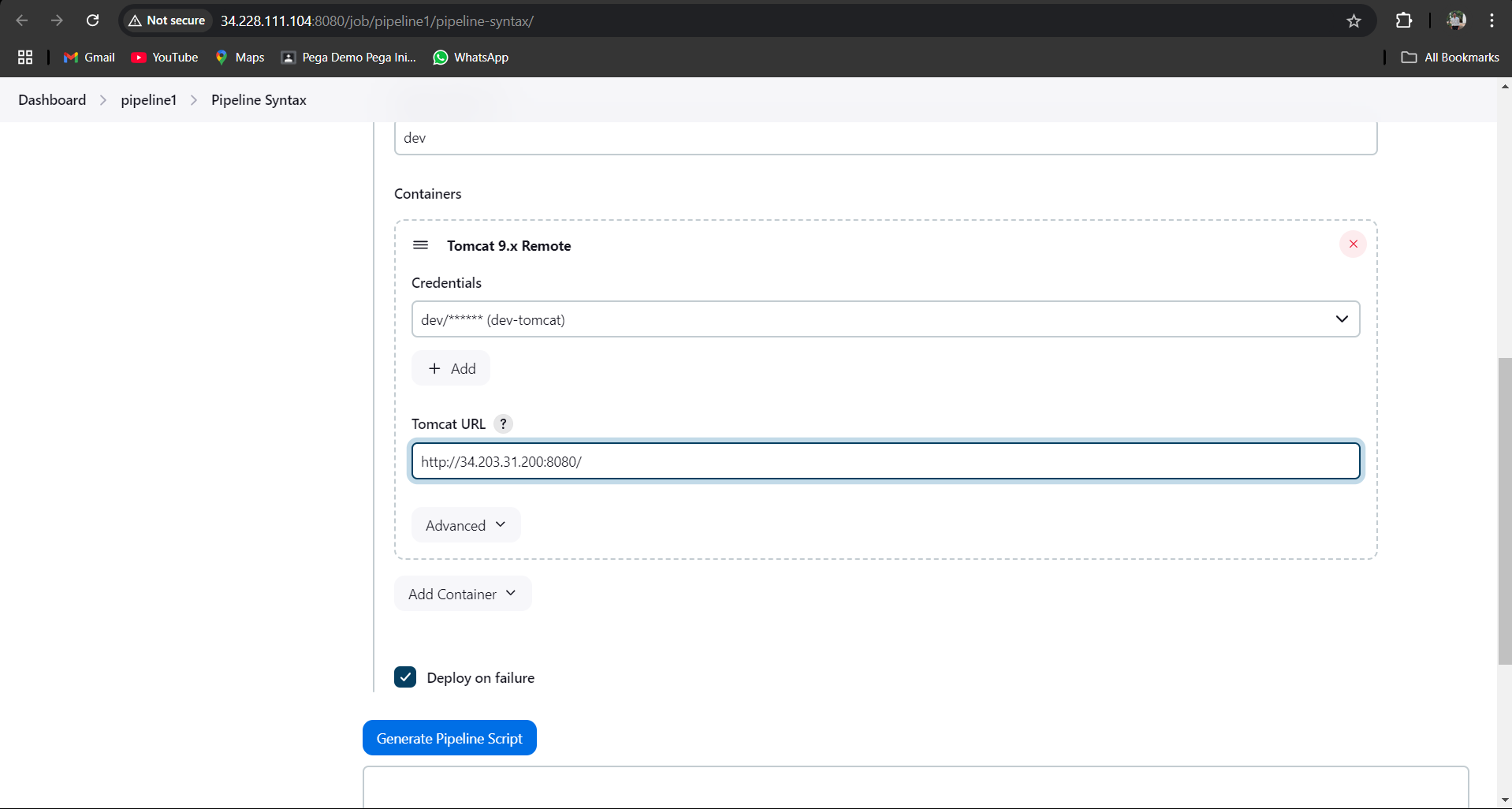


We have to generate three syntax’s for dev , test , pro.



It will show like this .

I am selecting the dev credentials.



In tomcat url you have to give dev-tomcat server url.

Click on generate pipeline syntax.

Like that we have to do both test and pro to generate pipeline syntax.

pipeline {

agent any

stages {

stage('clone'){

steps{

checkout scmGit(branches: [[name: '\*/master']], extensions: [], userRemoteConfigs: [[url: 'https://github.com/Venn1991/train-ticket-reservation.git']])

}

}

stage('build'){

steps{

sh 'mvn clean package'

}

}

stage('Deploy to Development') {

steps {

script {

def deployDev = input(message: 'Deploy to Development?', ok: 'Yes', parameters: [booleanParam(defaultValue: true, name: 'Deploy')])

if (deployDev) {

deploy adapters: [tomcat9(credentialsId: 'dev-tomcat', path: '', url: 'http://34.203.31.200:8080/')], contextPath: 'dev', war: '\*\*/\*.war'

// Add deployment script/commands for Development environment

echo 'Deploying to Development...'

} else {

echo 'Skipping Development deployment.'

}

}

}

}

stage('Deploy to Testing') {

steps {

script {

def deployTest = input(message: 'Deploy to Testing?', ok: 'Yes', parameters: [booleanParam(defaultValue: true, name: 'Deploy')])

if (deployTesting) {

deploy adapters: [tomcat9(credentialsId: 'test-tomcat', path: '', url: 'http://18.214.98.150:8080/')], contextPath: 'test', war: '\*\*/\*.war'

// Add deployment script/commands for Testing environment

echo 'Deploying to Testing...'

} else {

echo 'Skipping Testing deployment.'

}

}

}

}

stage('Deploy to Production') {

steps {

script {

def deployProd = input(message: 'Deploy to Production?', ok: 'Yes', parameters: [booleanParam(defaultValue: true, name: 'Deploy')])

if (deployProd) {

deploy adapters: [tomcat9(credentialsId: 'pro-tomcat', path: '', url: 'http://54.80.193.148:8080/')], contextPath: 'pro', war: '\*\*/\*.war'

// Add deployment script/commands for Production environment

echo 'Deploying to Production...'

} else {

echo 'Skipping Production deployment.'

}

}

}

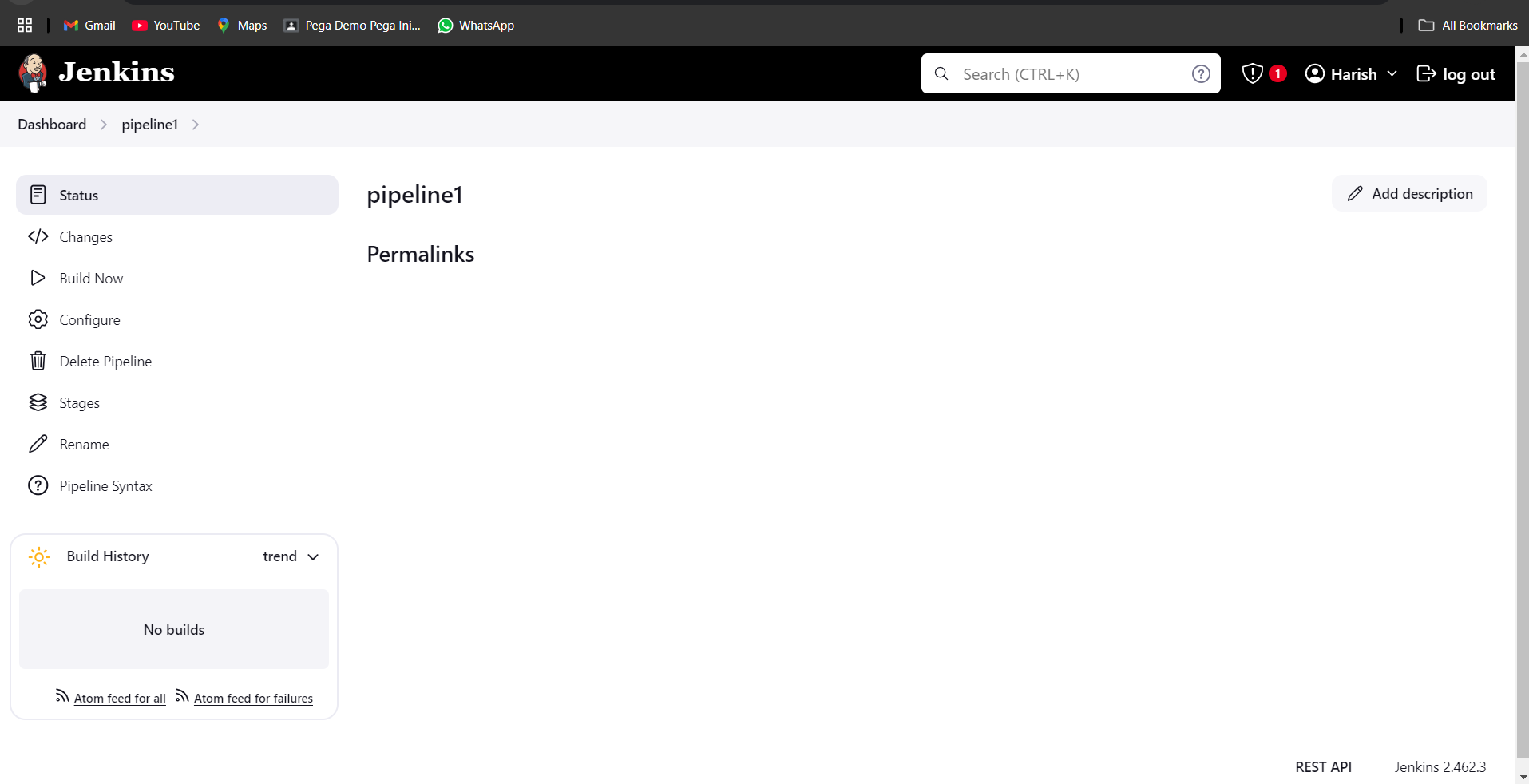
}

}

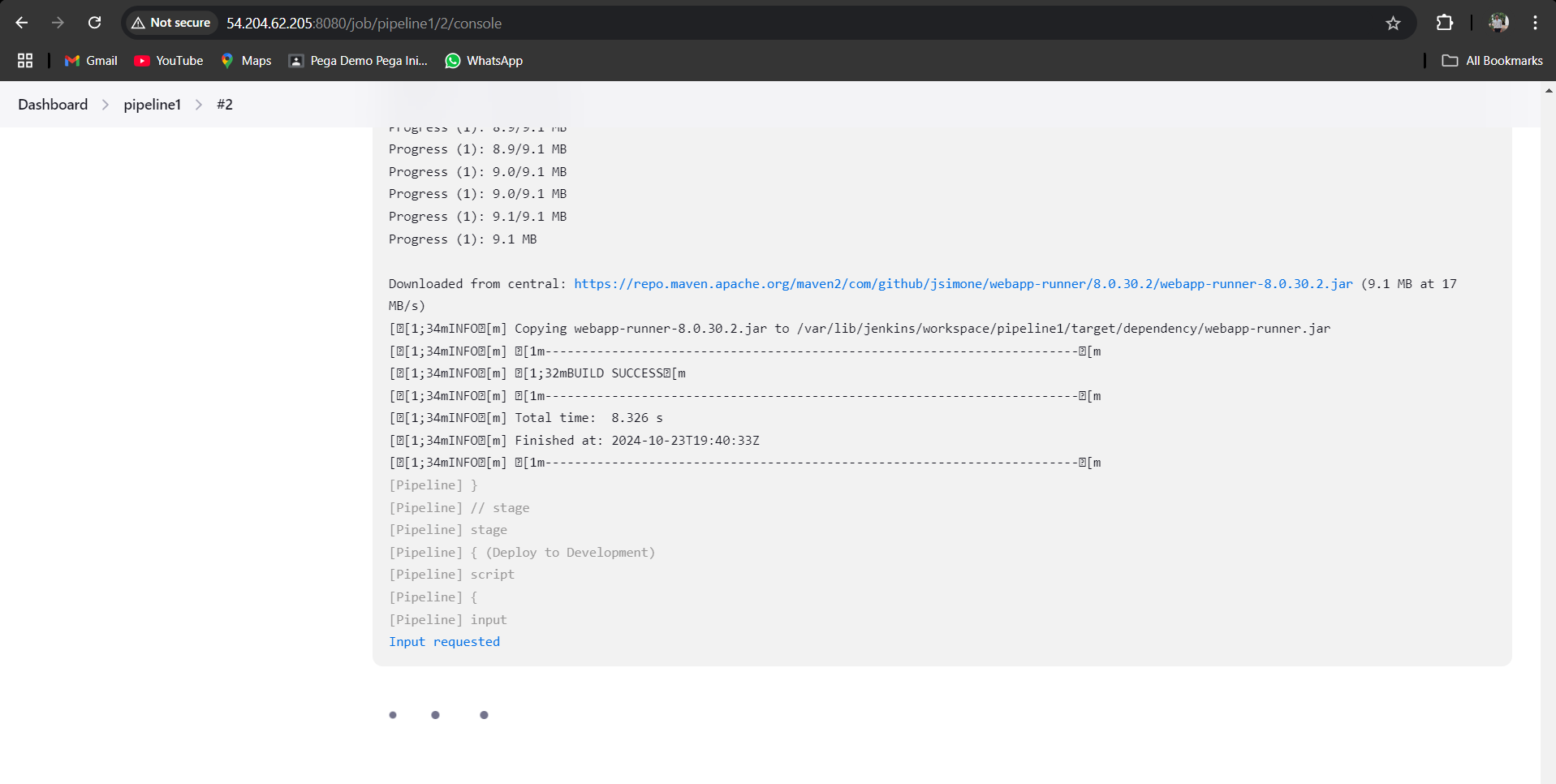
}

This is pipeline.

Now click on apply and save.

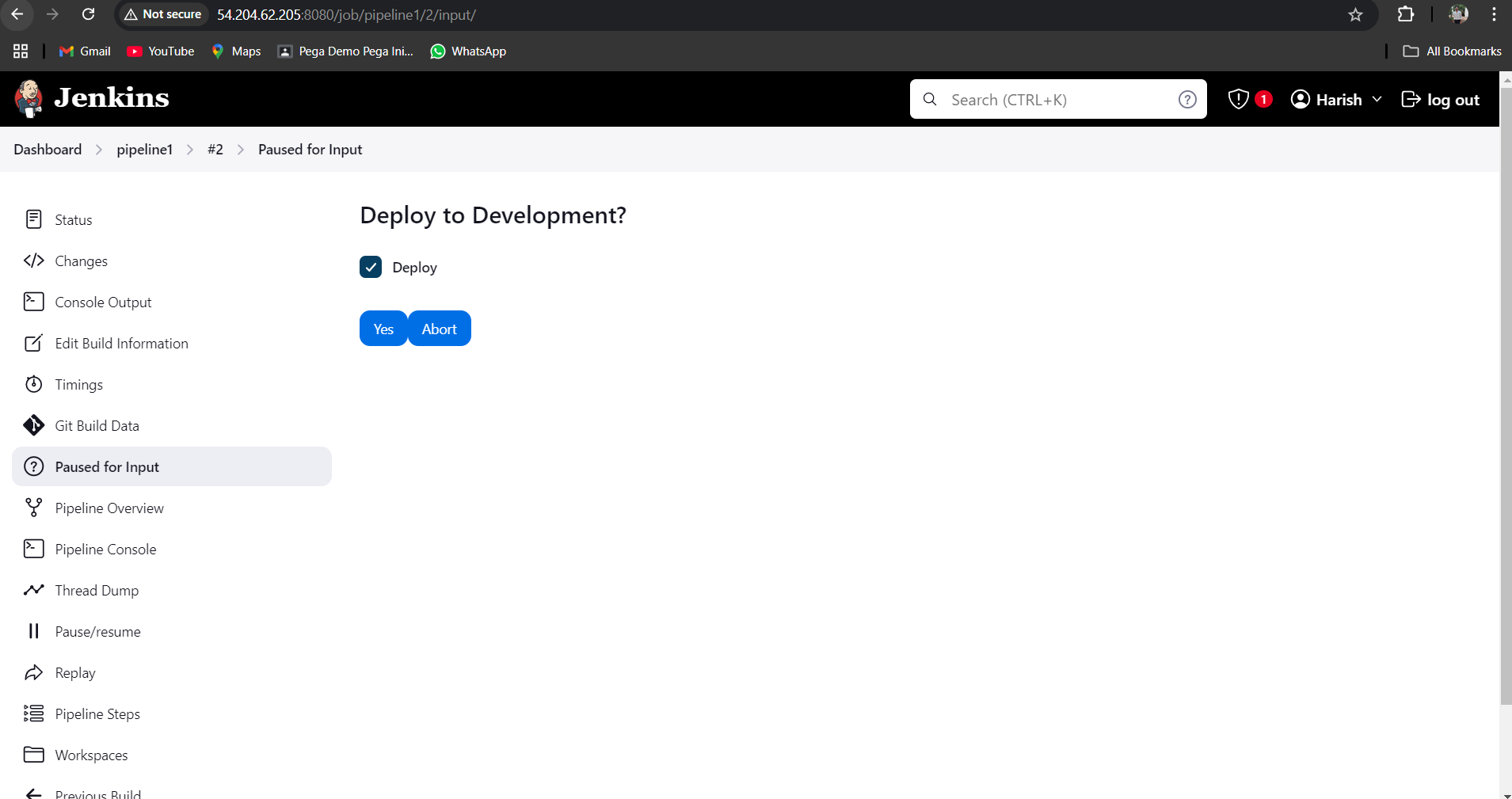


Click on build now.



It will ask for input requested.

Click on that



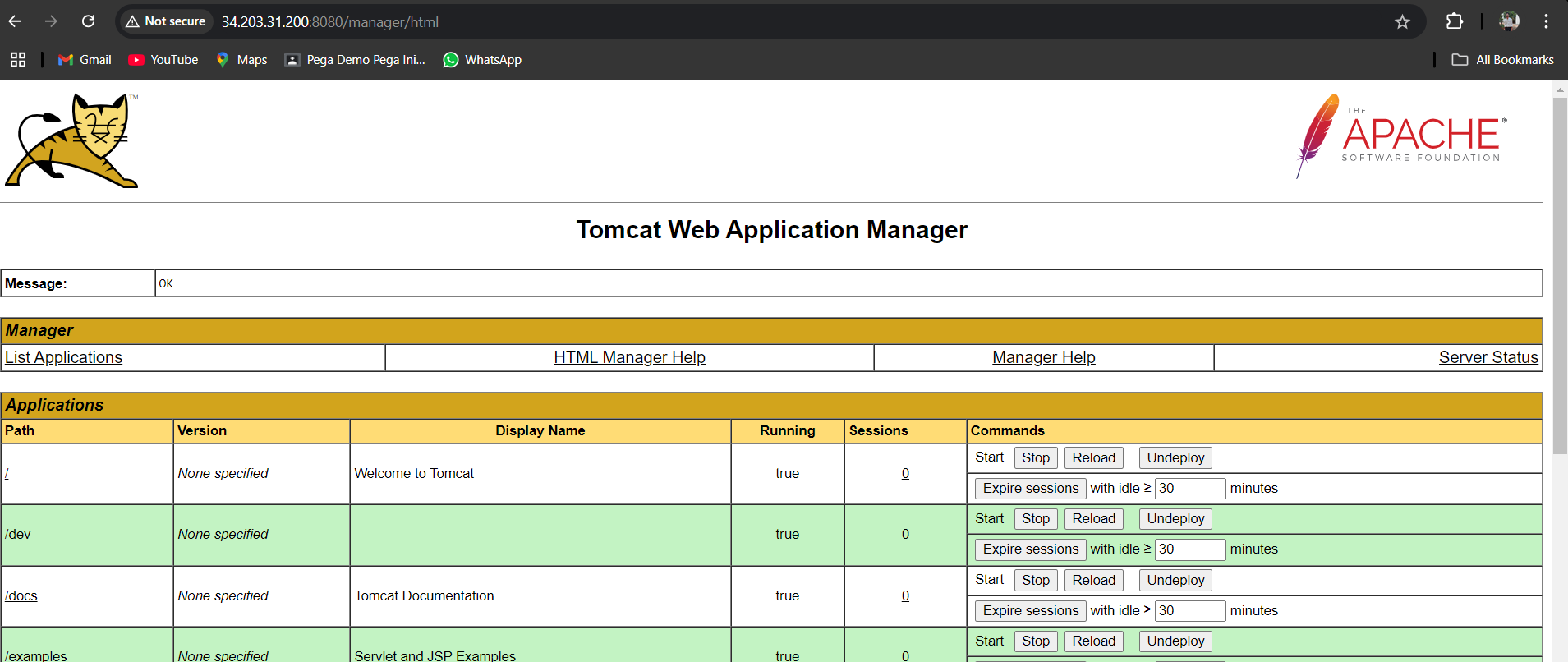
It is asking to deploy to dev

Yes or abort.

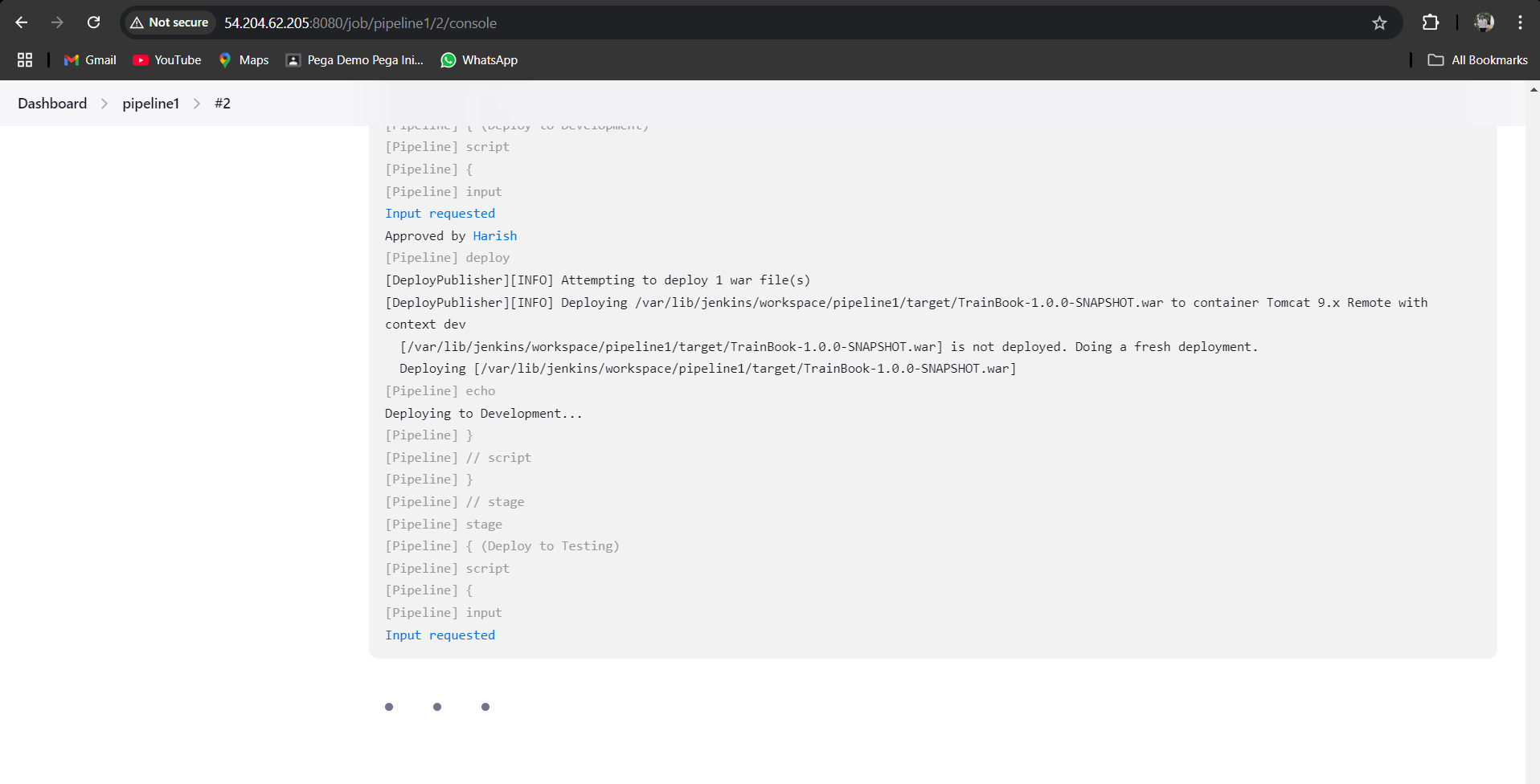
Click on yes.

Now go to dev-tomcat server and refresh it.

We can see the dev application on tomcat server.



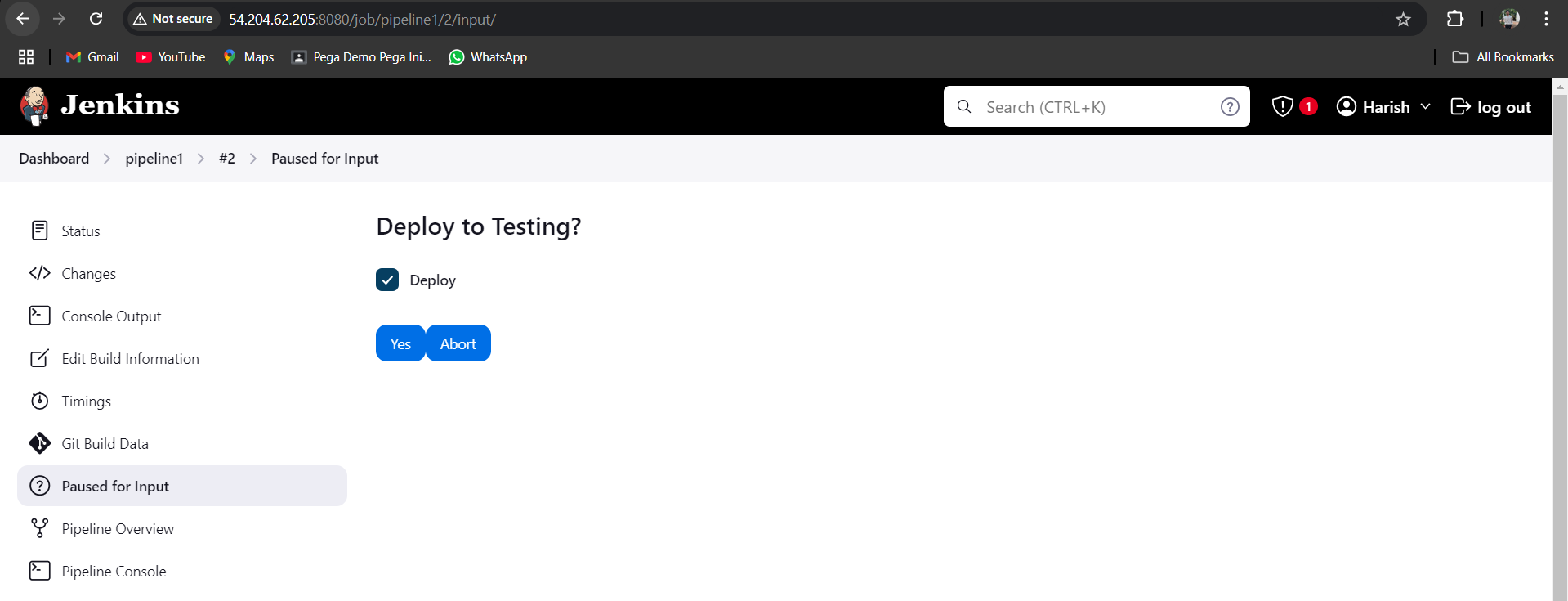
After that it will ask for input requested to deploy to testing .



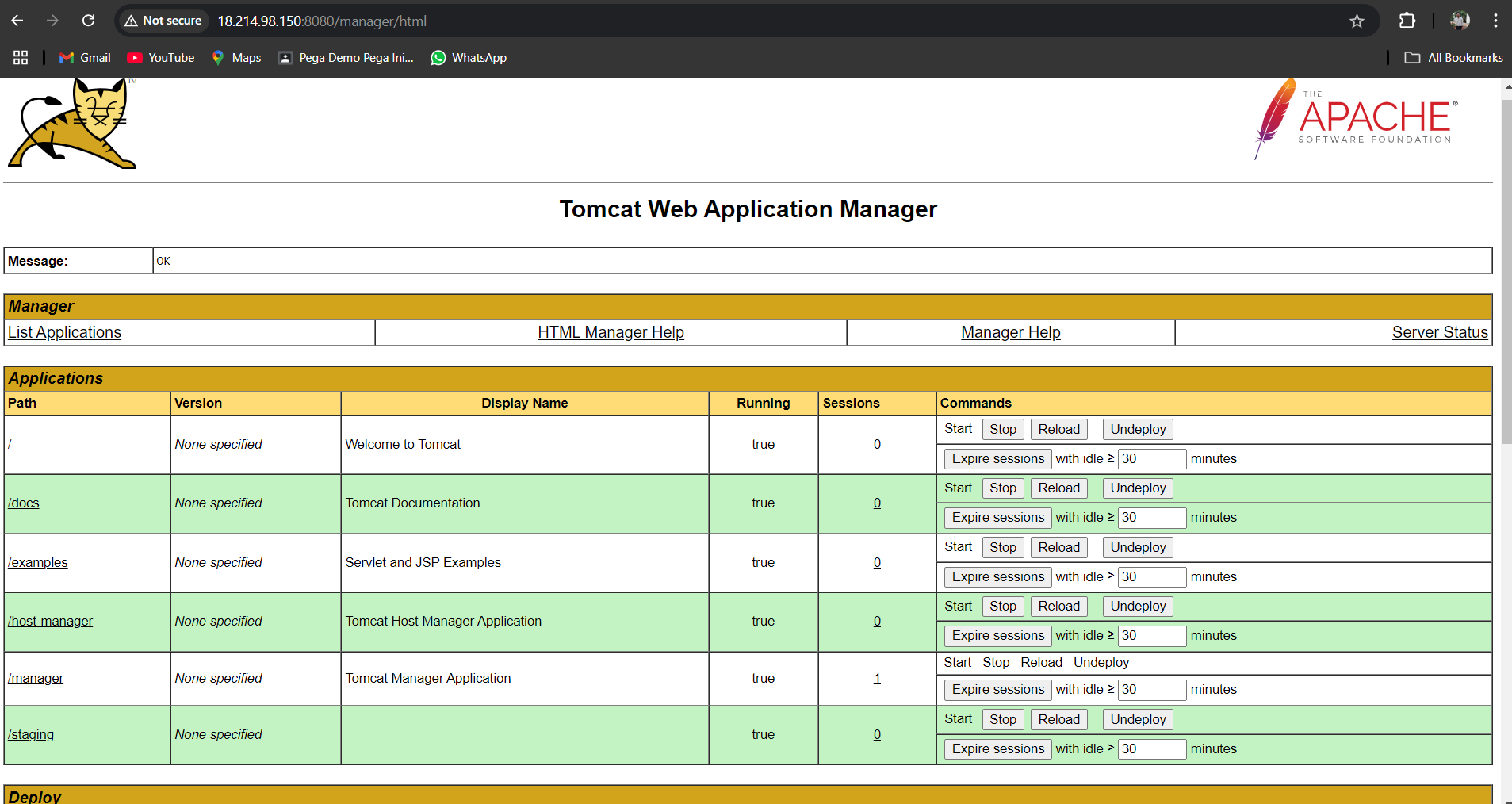
It will wait until we give input to it.

Give the input as yes.

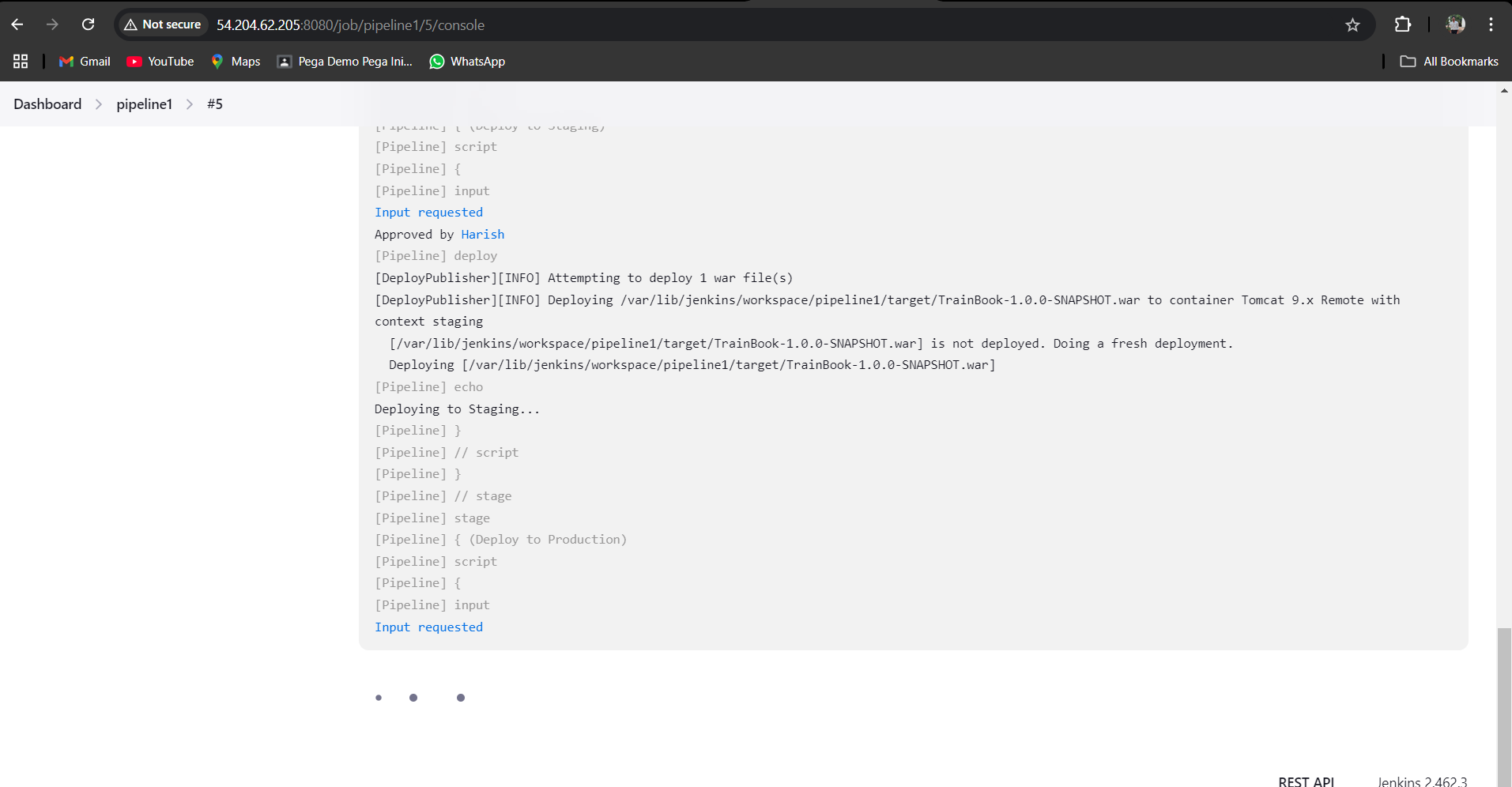
Go and refresh the test-tomcat server.



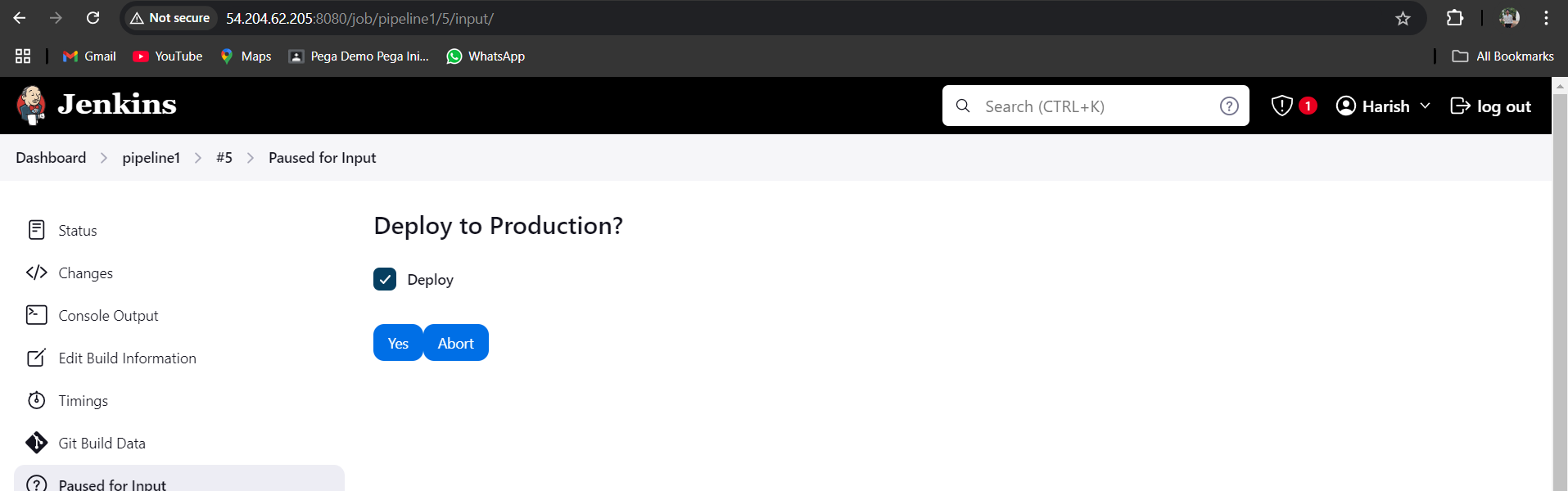
I was change the name of test to staging.

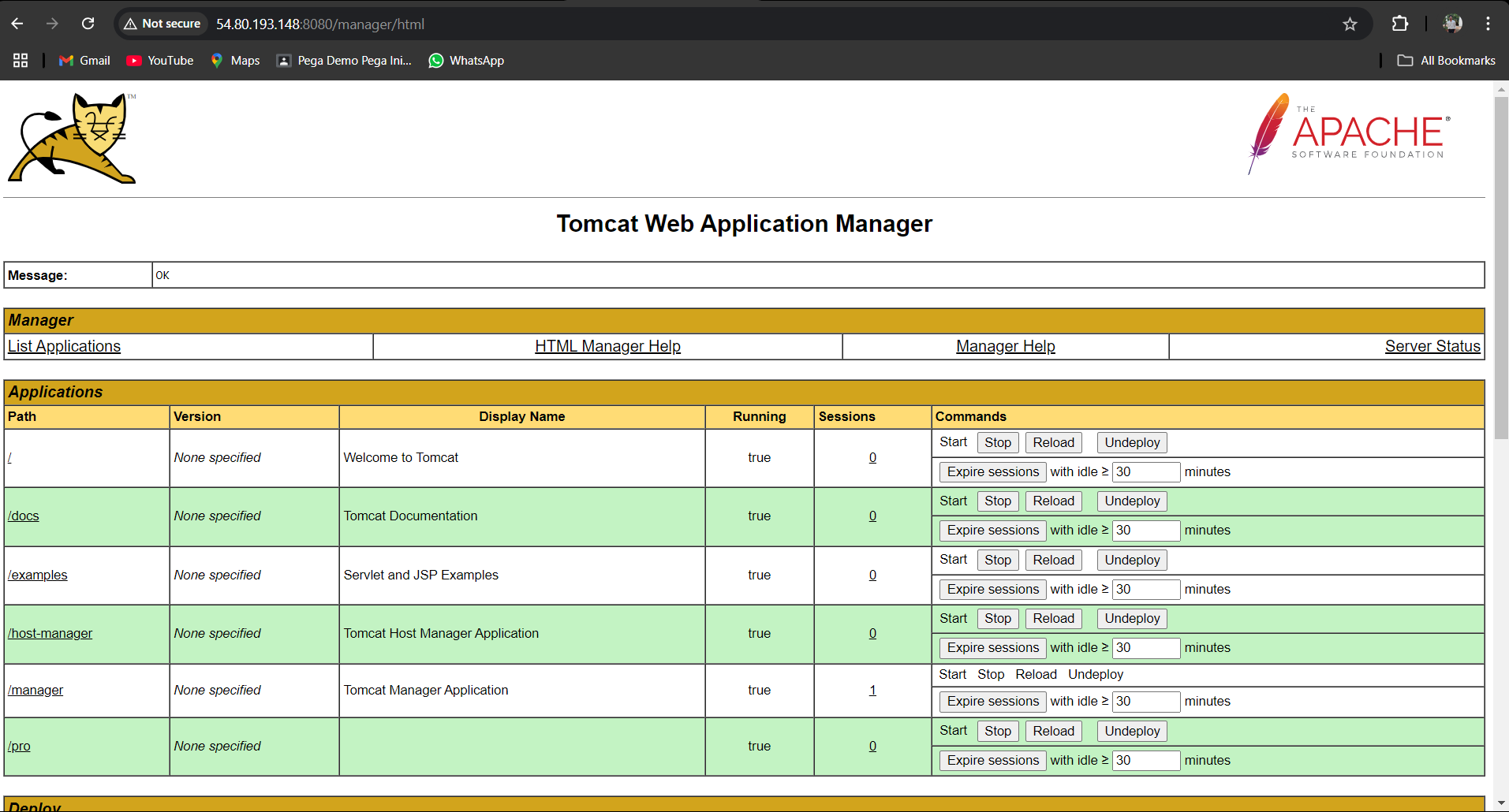


We got the staging deployed to test-tomcat server.

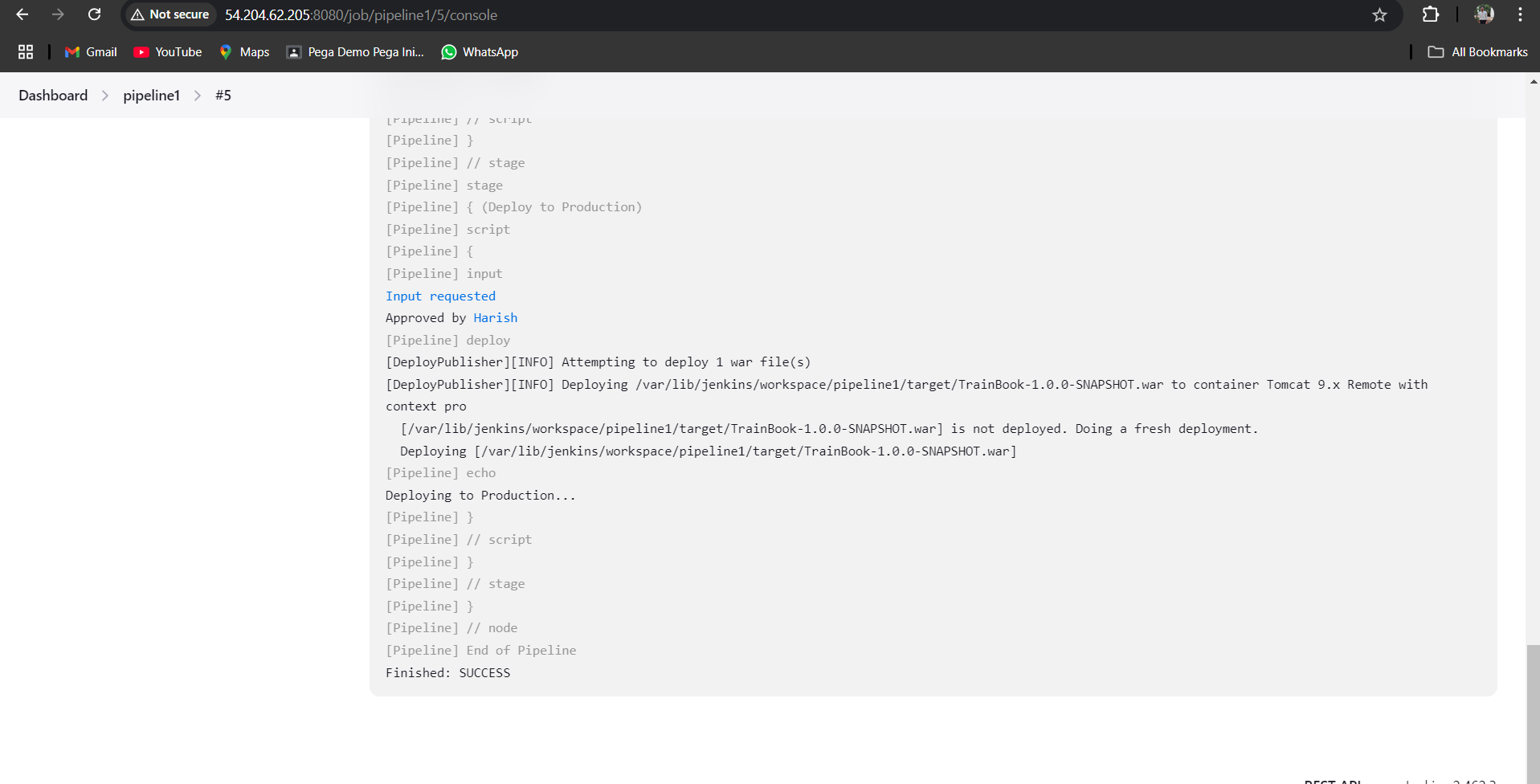


Then it is asking for production deployment .

click on yes



We got the production application on pro-tomcat server.



So, the pipeline is success.

**Conclusion**

By using Jenkins CICD pipeline we deployed an application in development , staging , and production environments.