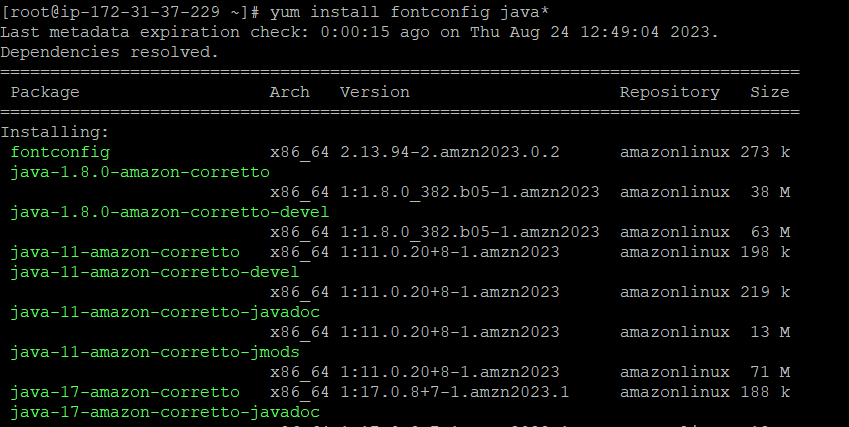
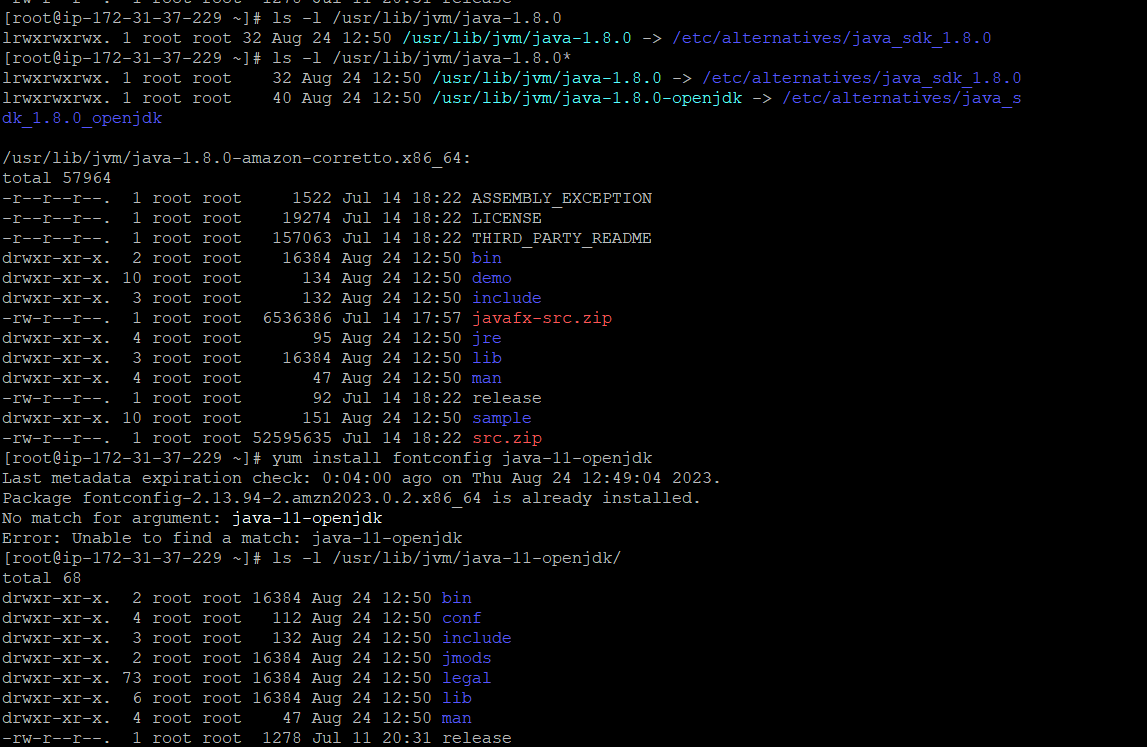
sudo rpm --import <https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key>



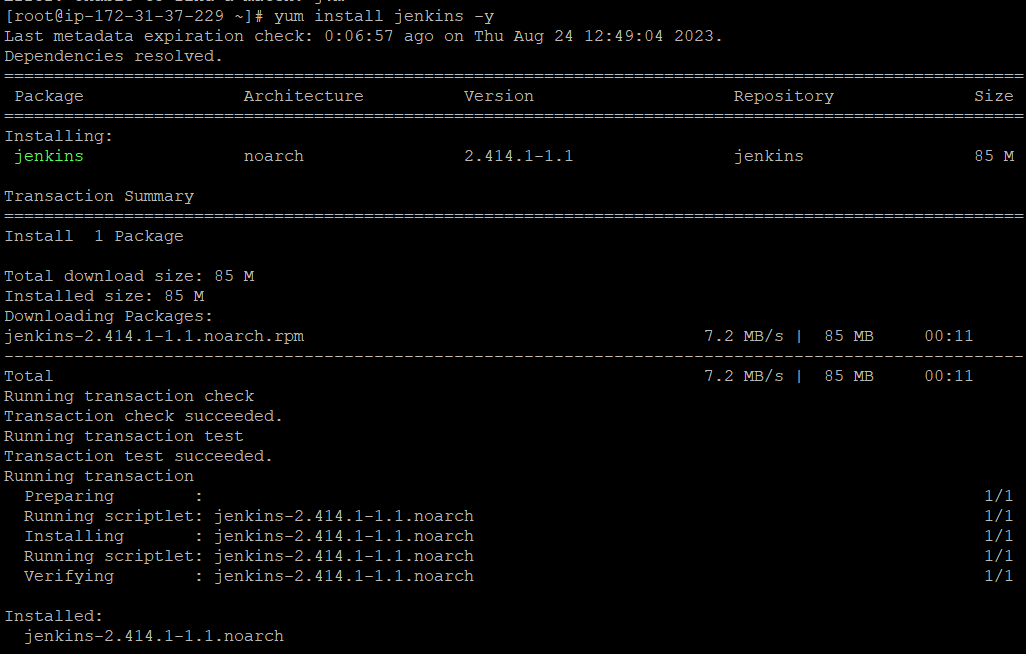
#yum install fontconfig java\*



ls -l /usr/lib/jvm/java-1.8.0\*



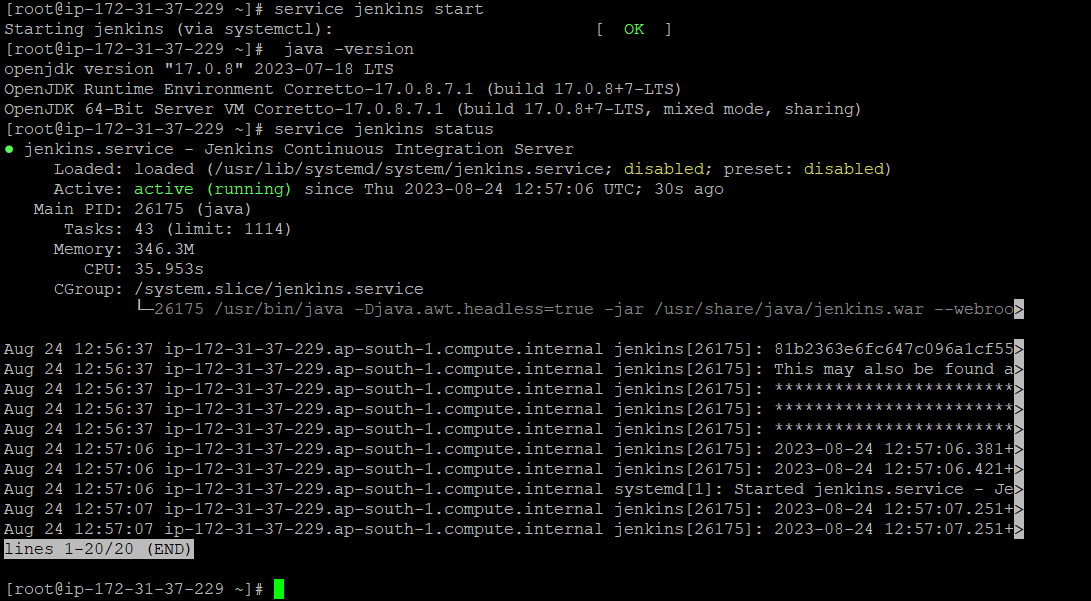
#yum install Jenkins -y



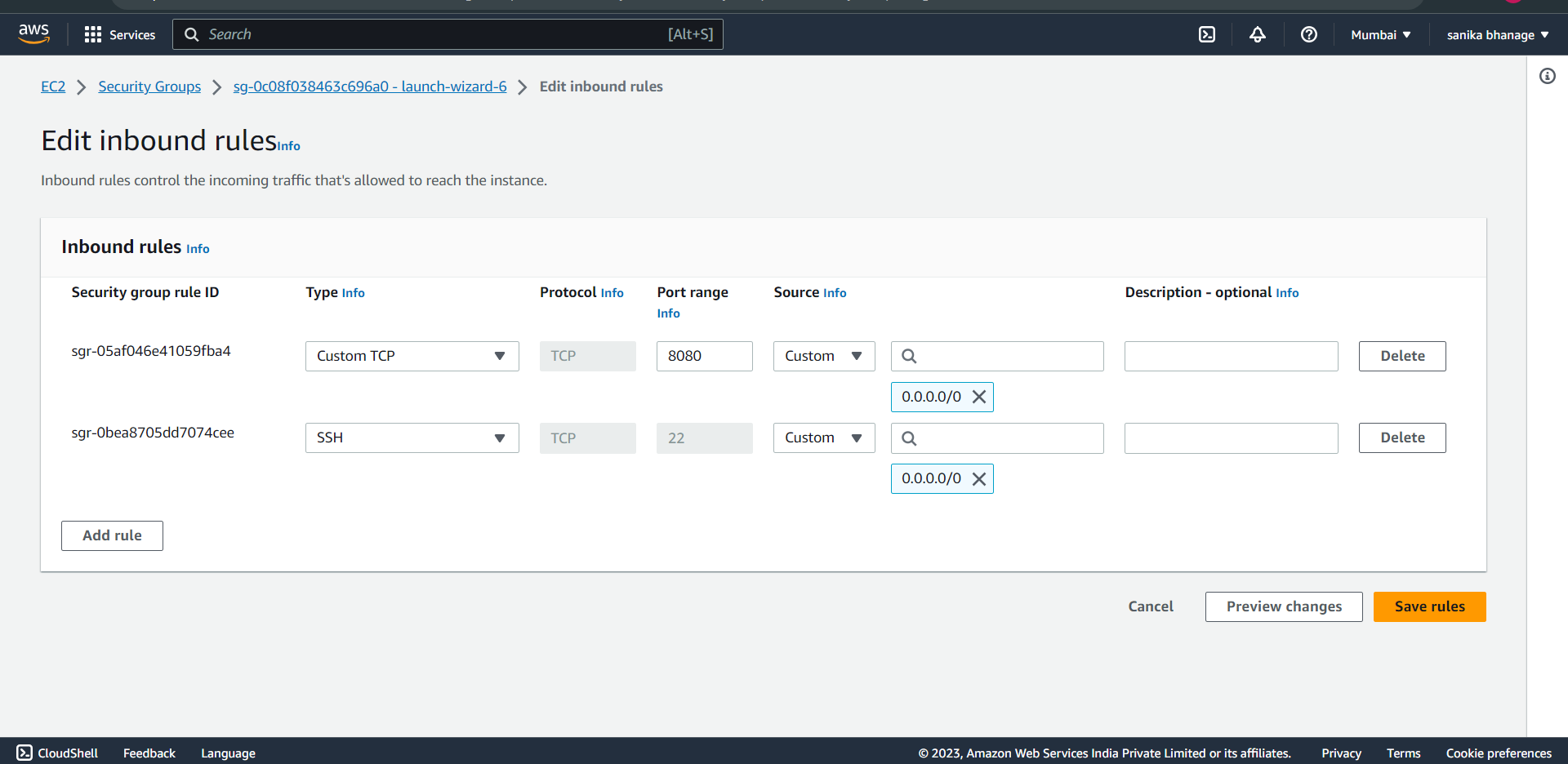
#service jenkins start

#java -version

#service jenkins status

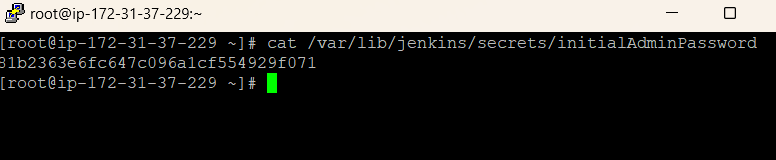


Check inbound rule for security group 8080 port must be added first for browsing jenkins

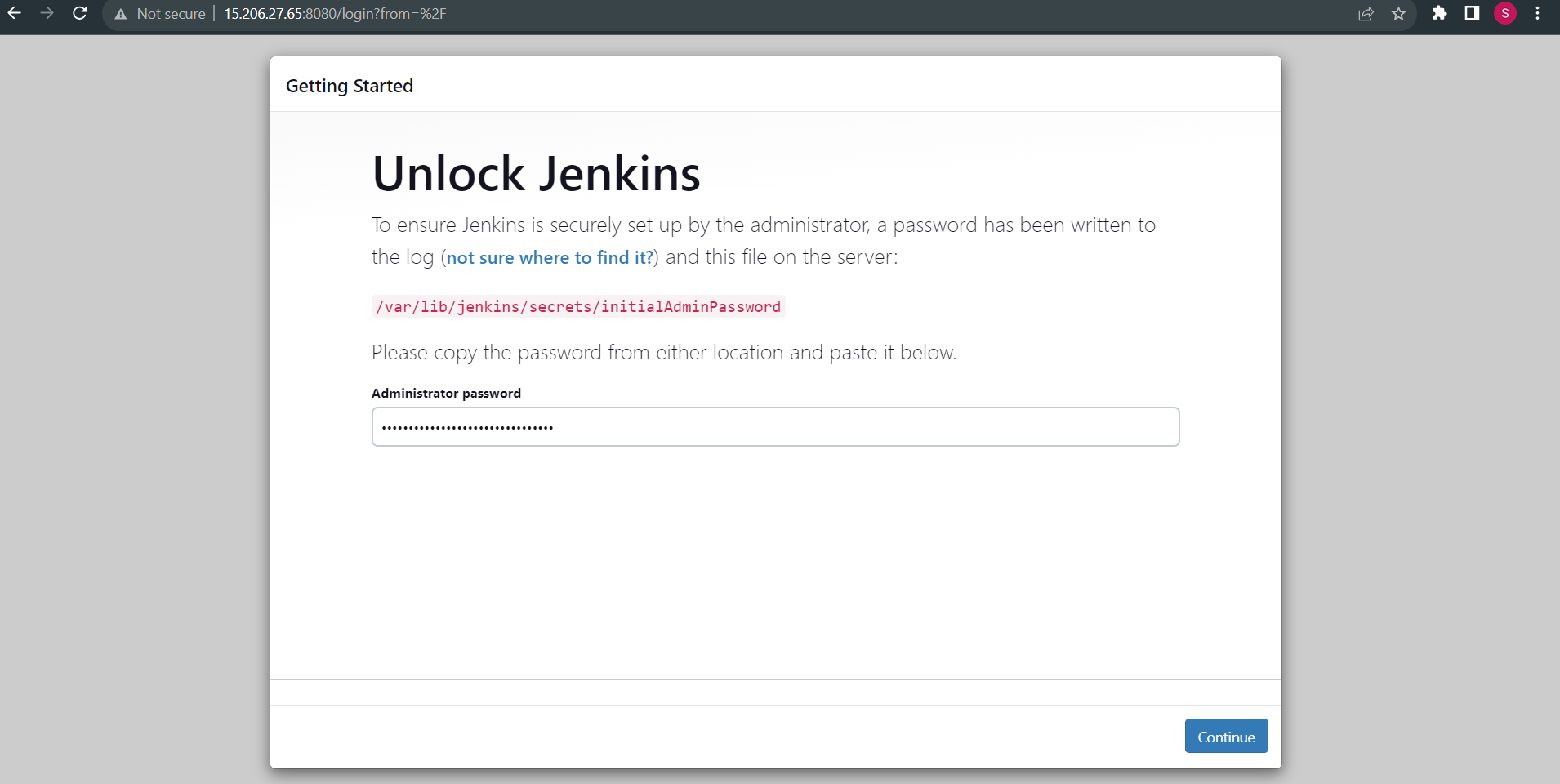


On browser, search for http://publicipv4:8080

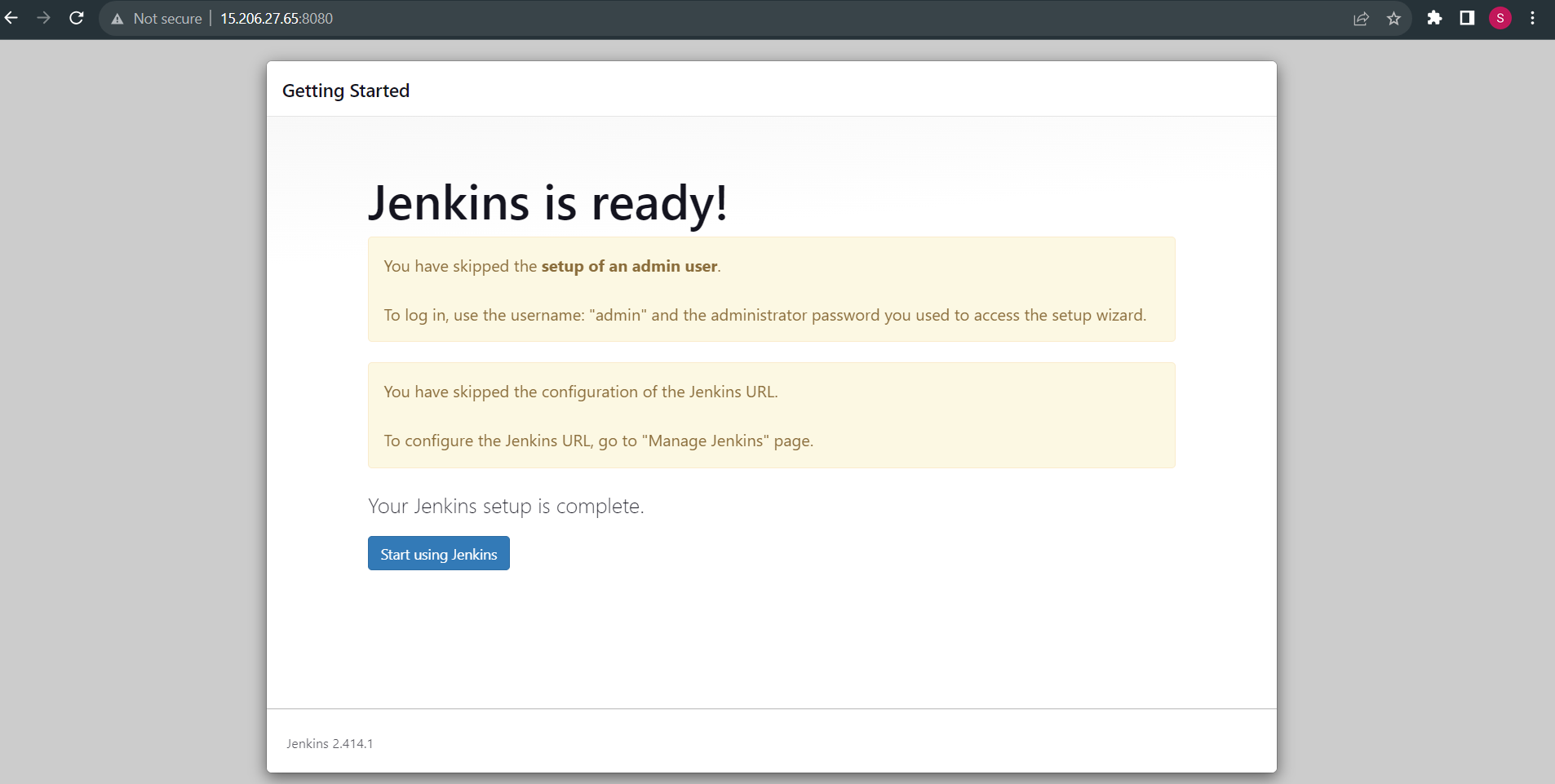
#cat /var/lib/jenkins/secrets/initialAdminPassword



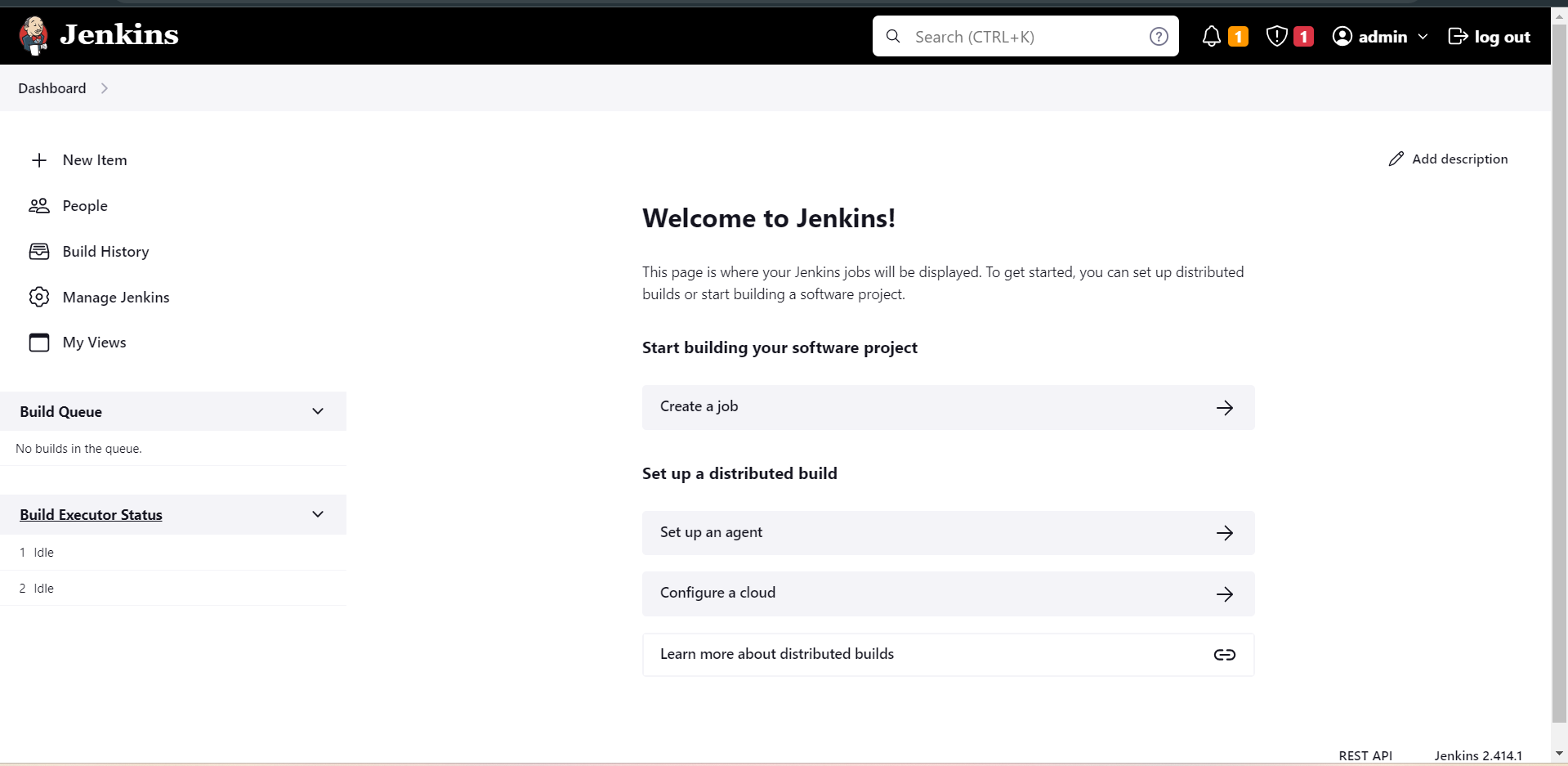
Copy this password and paste it in browser,



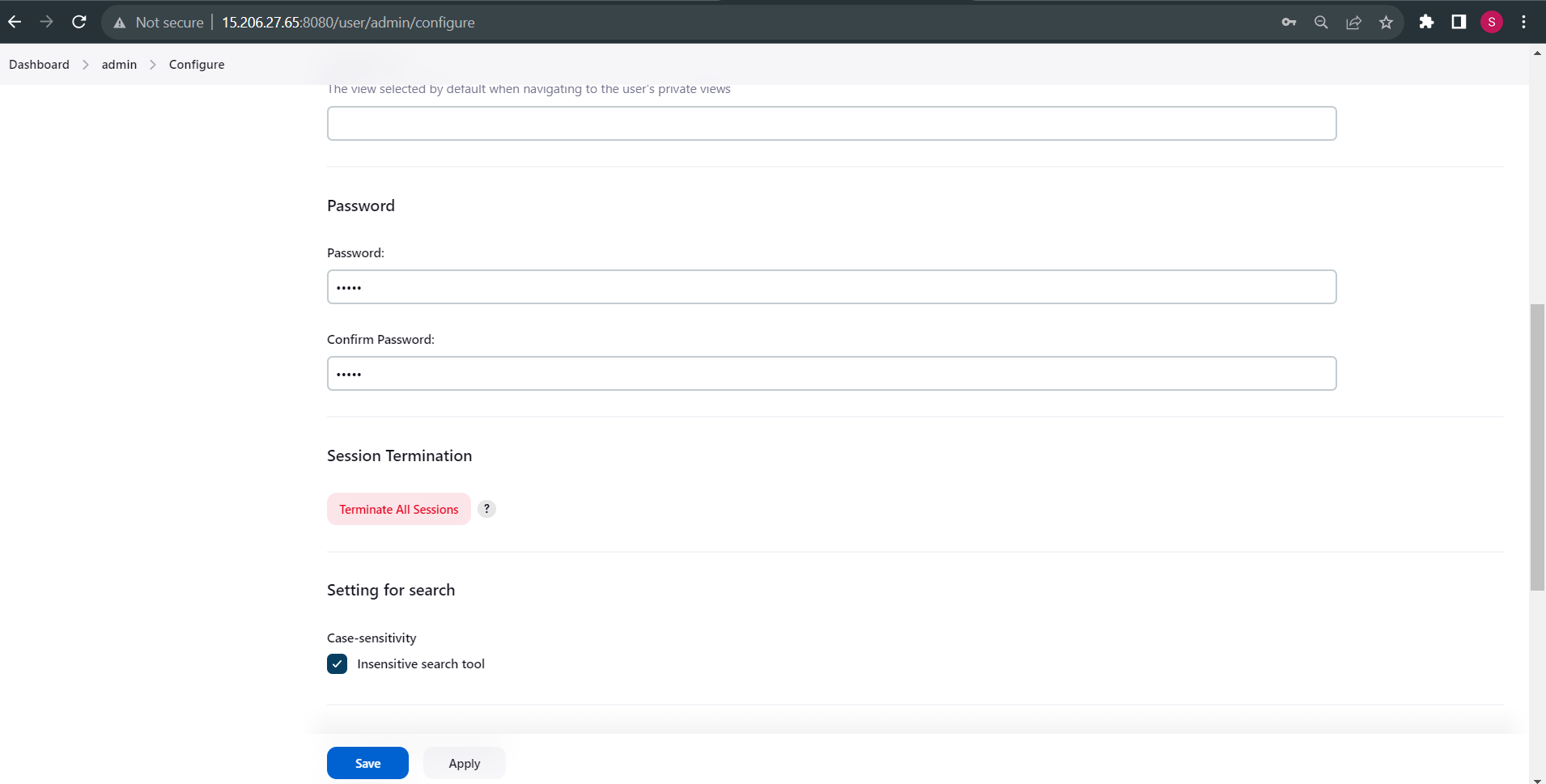
Not installation of any plugin just start jenkins,



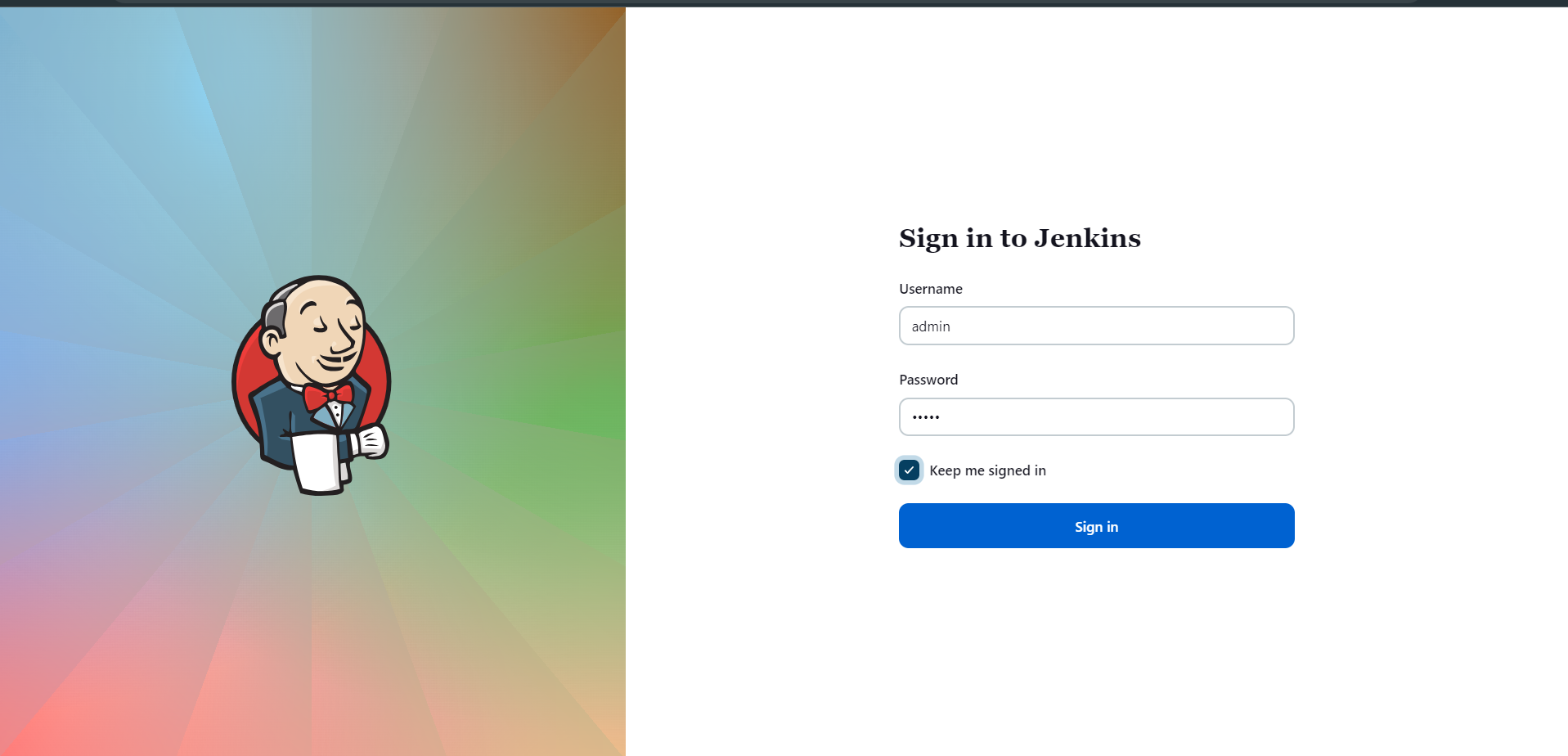
Jenkins dashboard,



Change the jenkins default password, dashboard -> admin->configure

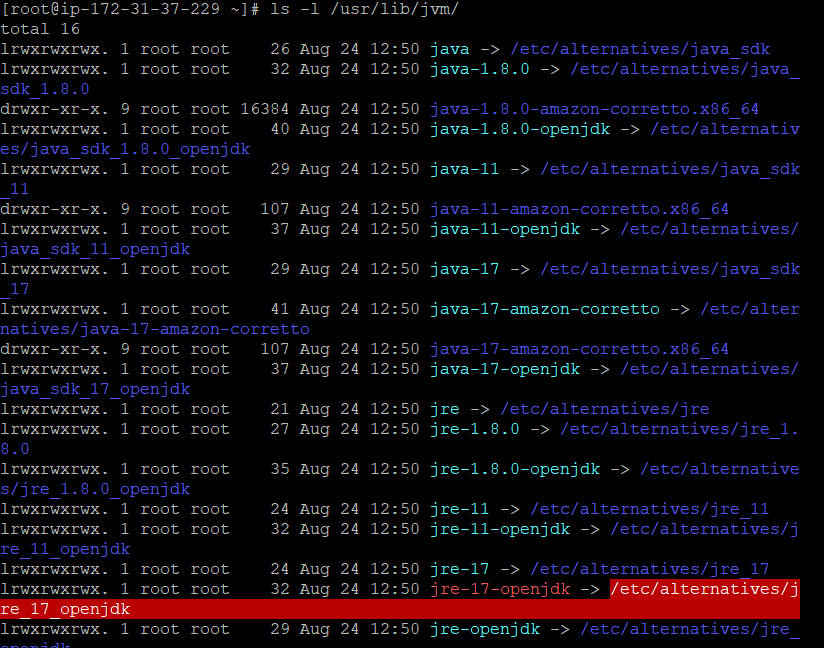


Again login with new password,

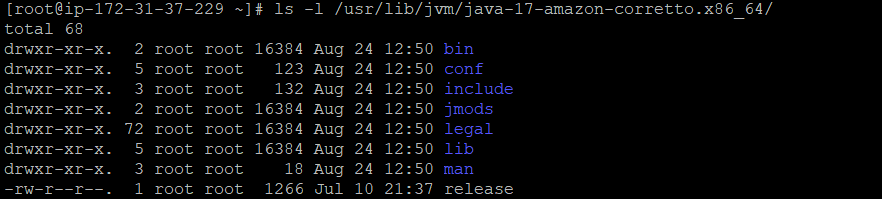


Set JAVA HOME PATH

#ls -l /usr/lib/jvm/



#ls -l /usr/lib/jvm/java-17-amazon-corretto.x86\_64/

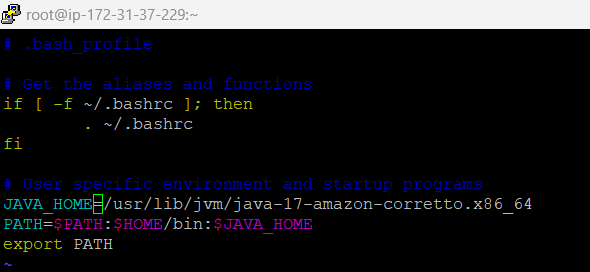


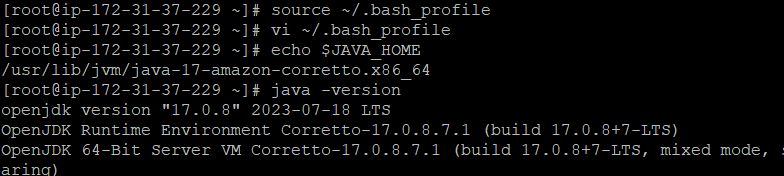
#vi ~/.bash\_profile paste the path add JAVA\_HOME

# sourch ~/.bash\_profile

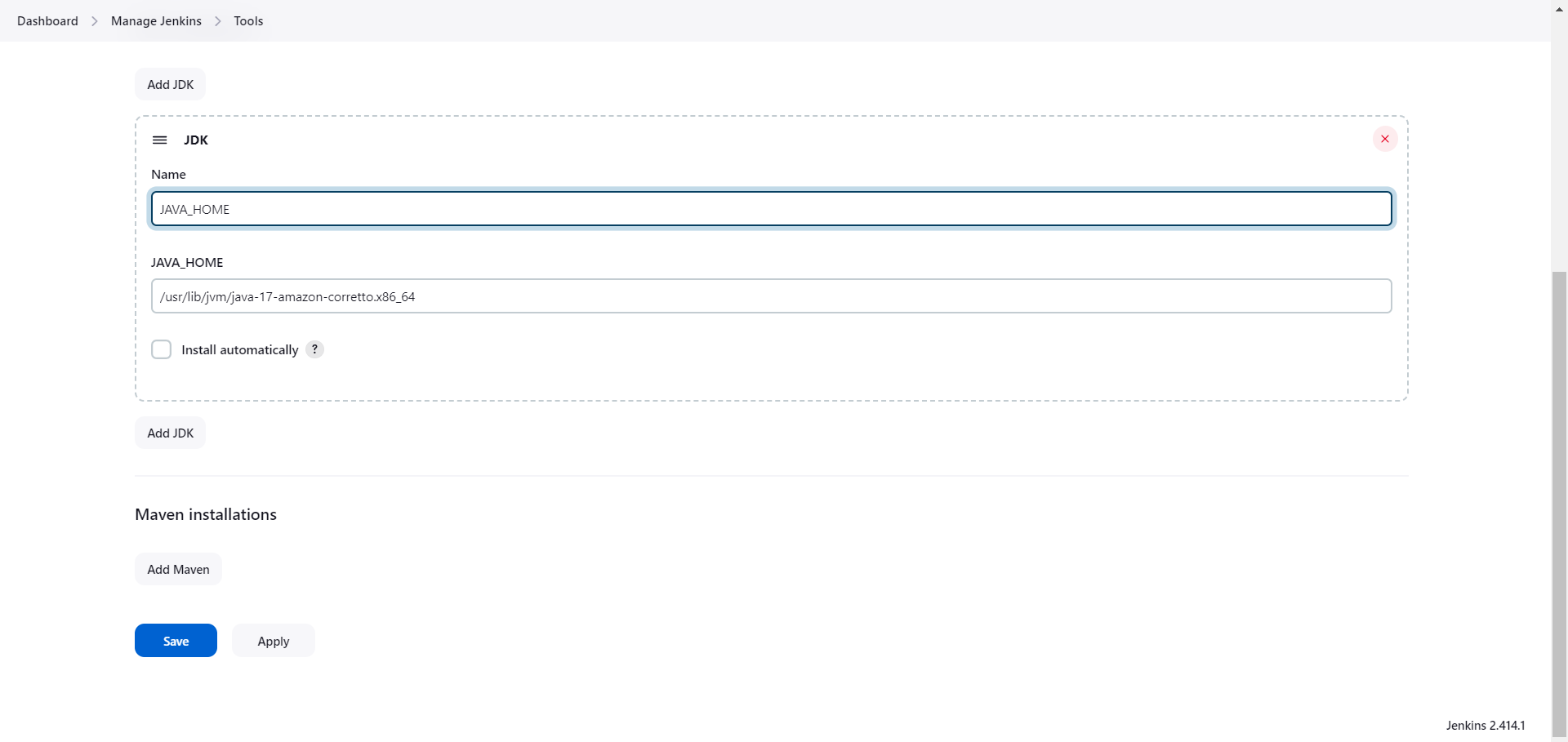
#java -version

#echo $JAVA\_HOME



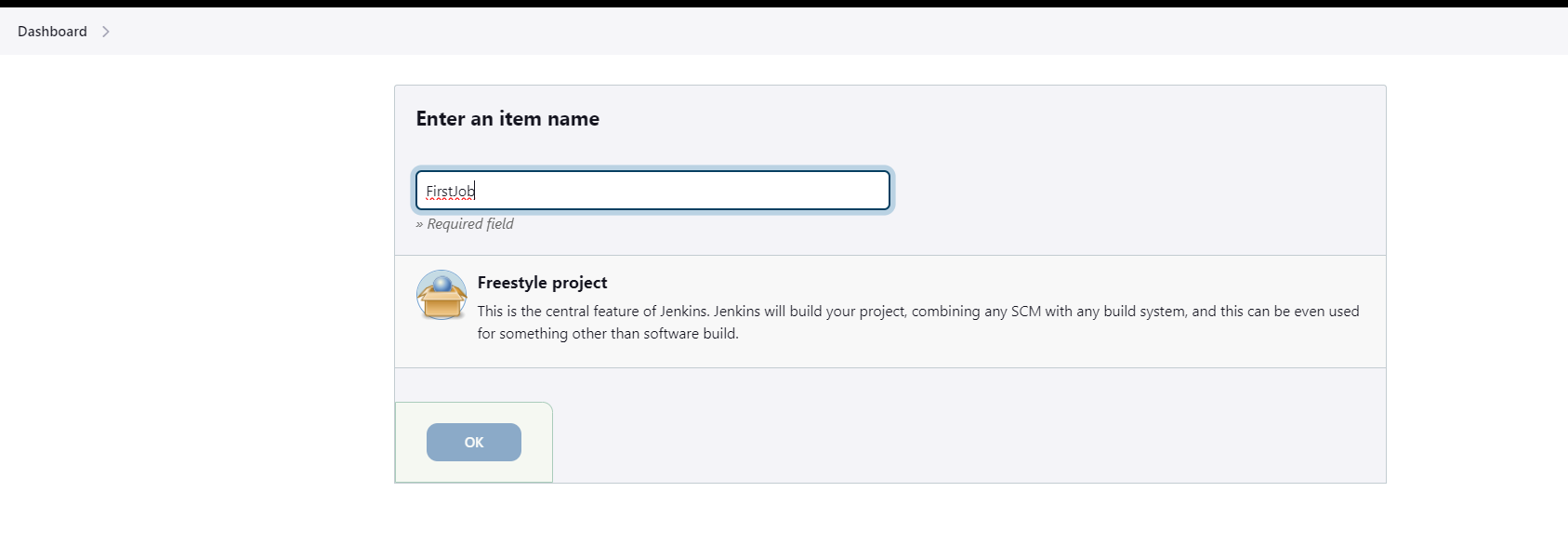


Now, we have to set java home path for that go to Manage jenkins->Tools-> Add JDK click apply & save.

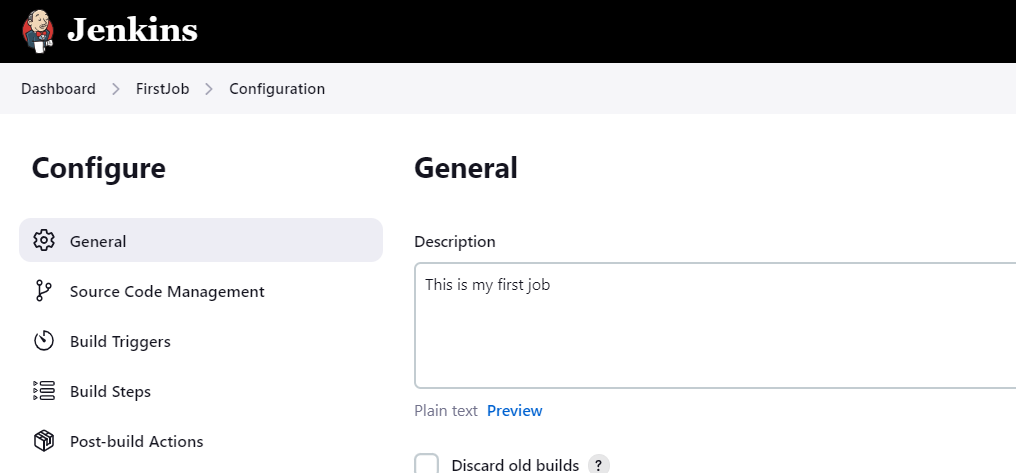


Run the first jenkins job,

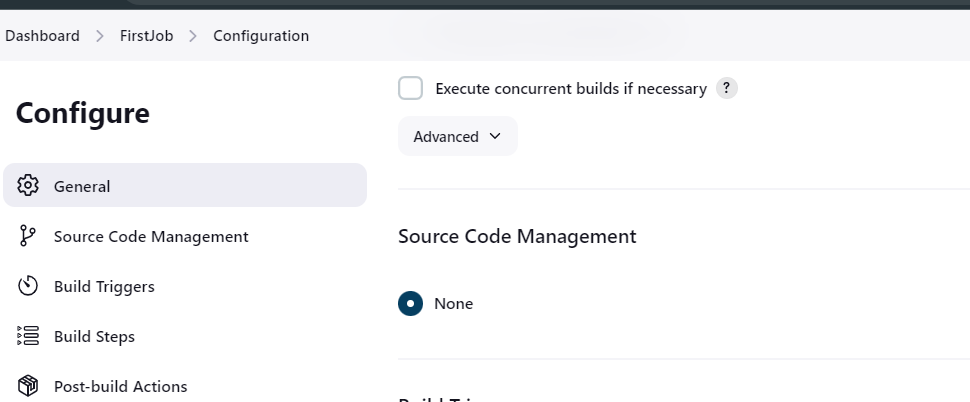
Dashboard -> new item -> add item name -> click free style project -> ok



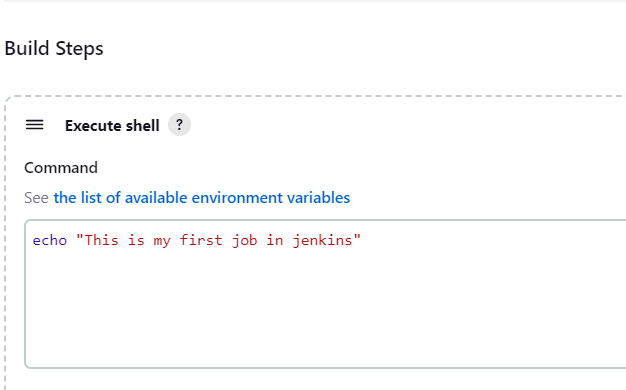
Give description



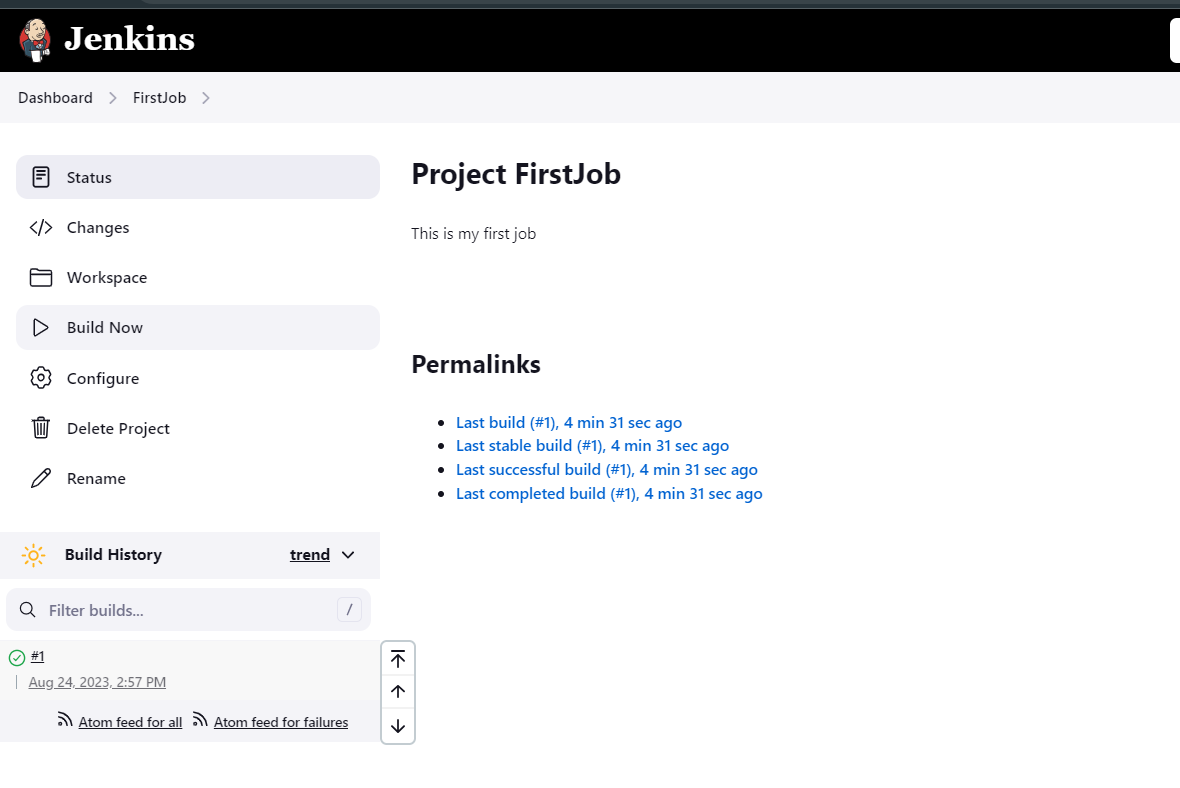
Select source code management -> none



Select Execute Shell and click on apply and save.

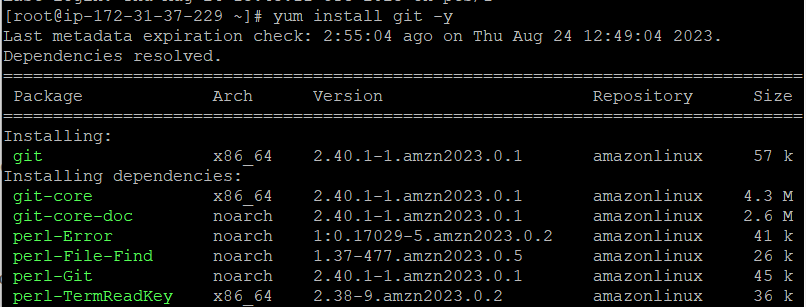


Click on Build Now and see the first job status



**Setup Git**

#yum install git -y

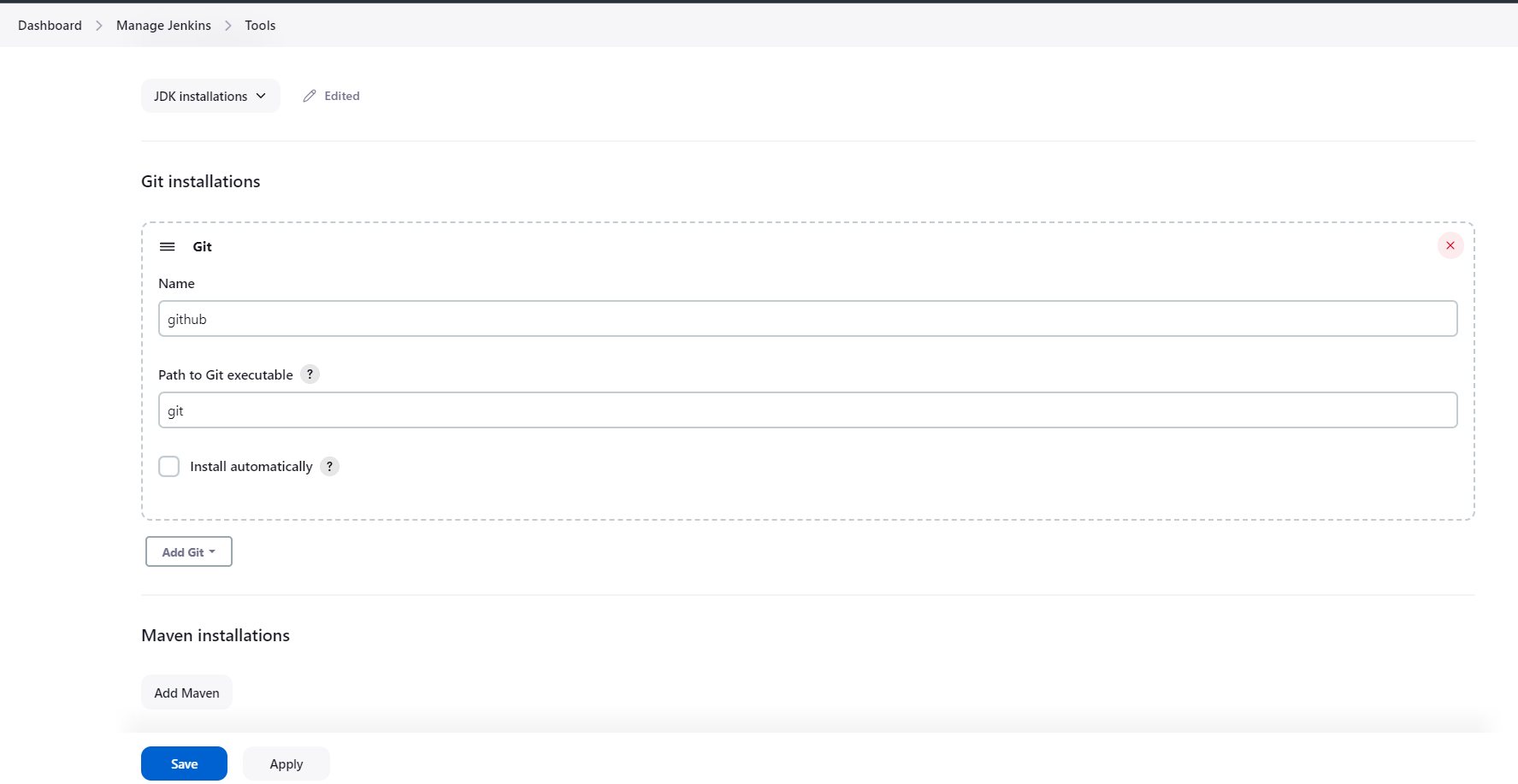


Manage Jenkins > Jenkins Plugins > available > github click on install



After the successful installation go to the Dashboard -> Manage jenkins -> Tools

Add git path and click on apply and save.



**Setup Maven**

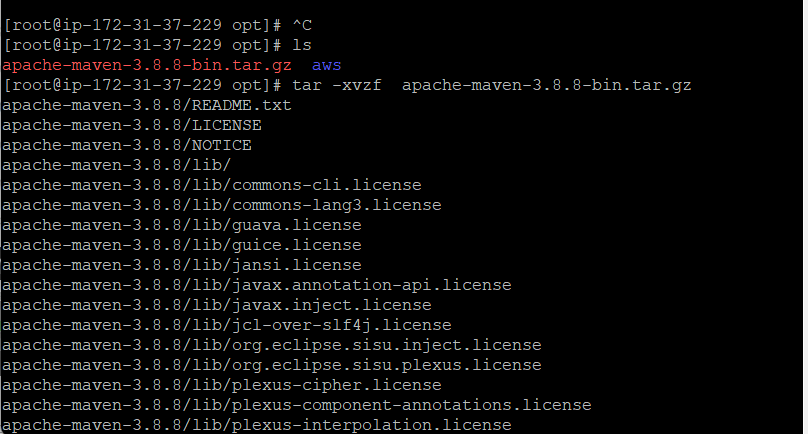
cd /opt

ls -ltr

wget https://dlcdn.apache.org/maven/maven-3/3.8.8/binaries/apache-maven-3.8.8-bin.tar.gz



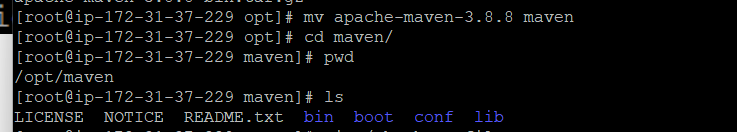
tar -xvzf apache-maven-3.8.8-bin.tar.gz



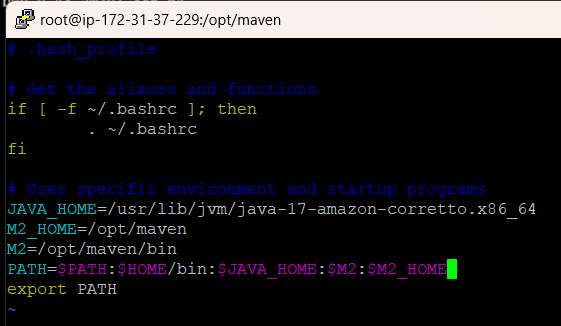
mv apache-maven-3.8.8 maven

cd maven/

pwd

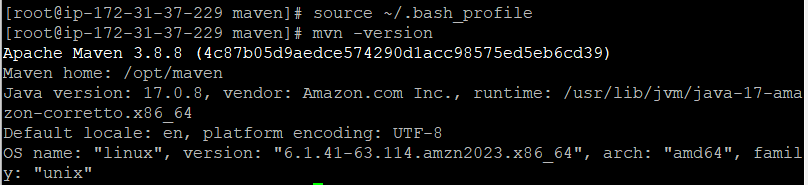


vi ~/.bash\_profile



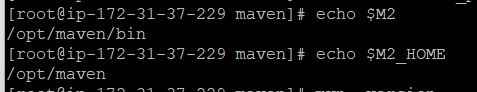
source ~/.bash\_profile

mvn -version



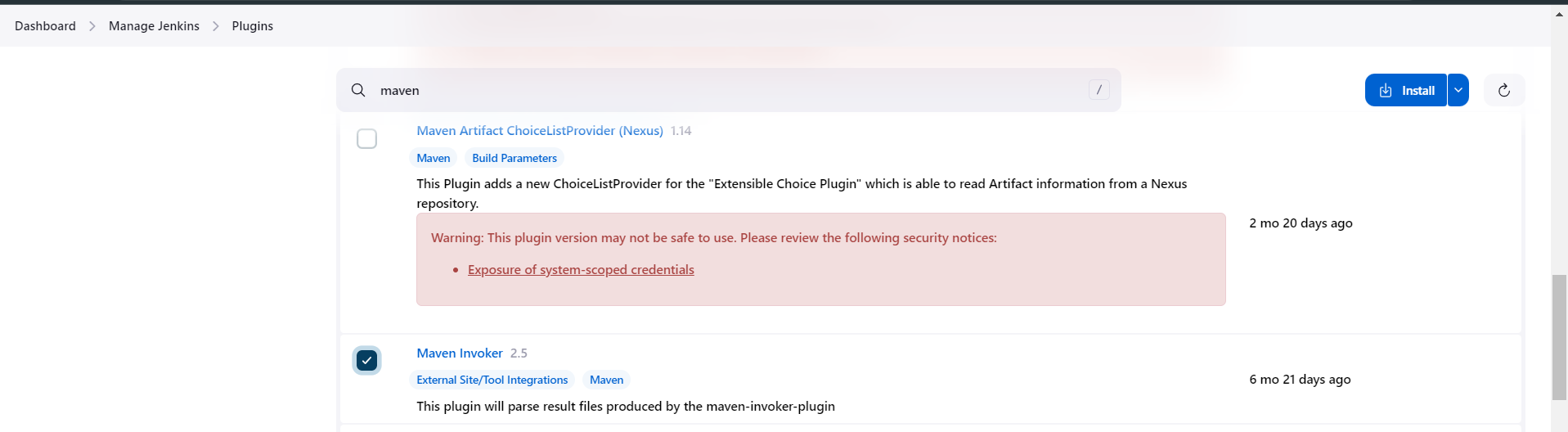
#echo $M2

#echo $M2\_HOME



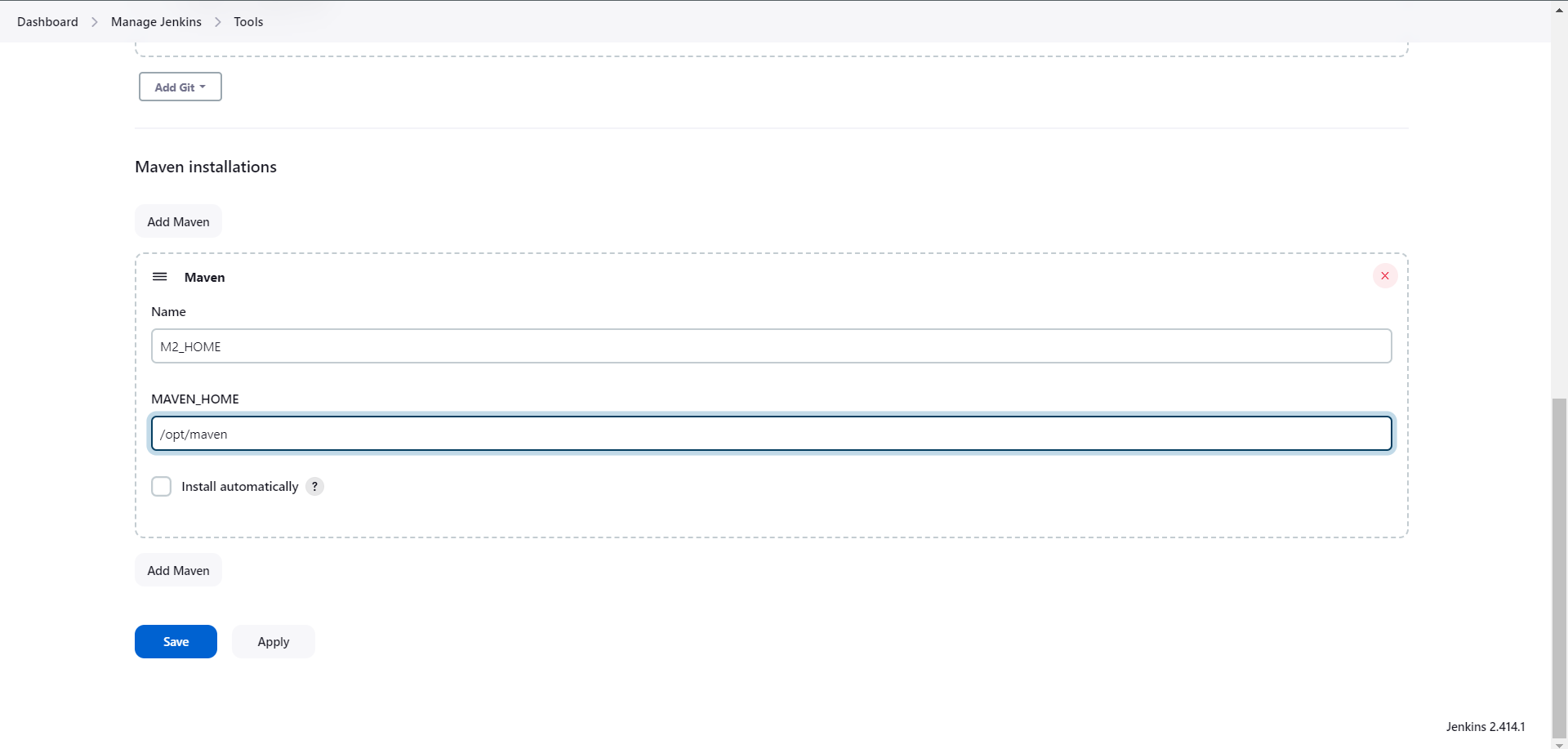
In the Manage jenkins -> plugins -> available plugin search maven and select “Maven Integration” & “Maven Invoker” click Install.





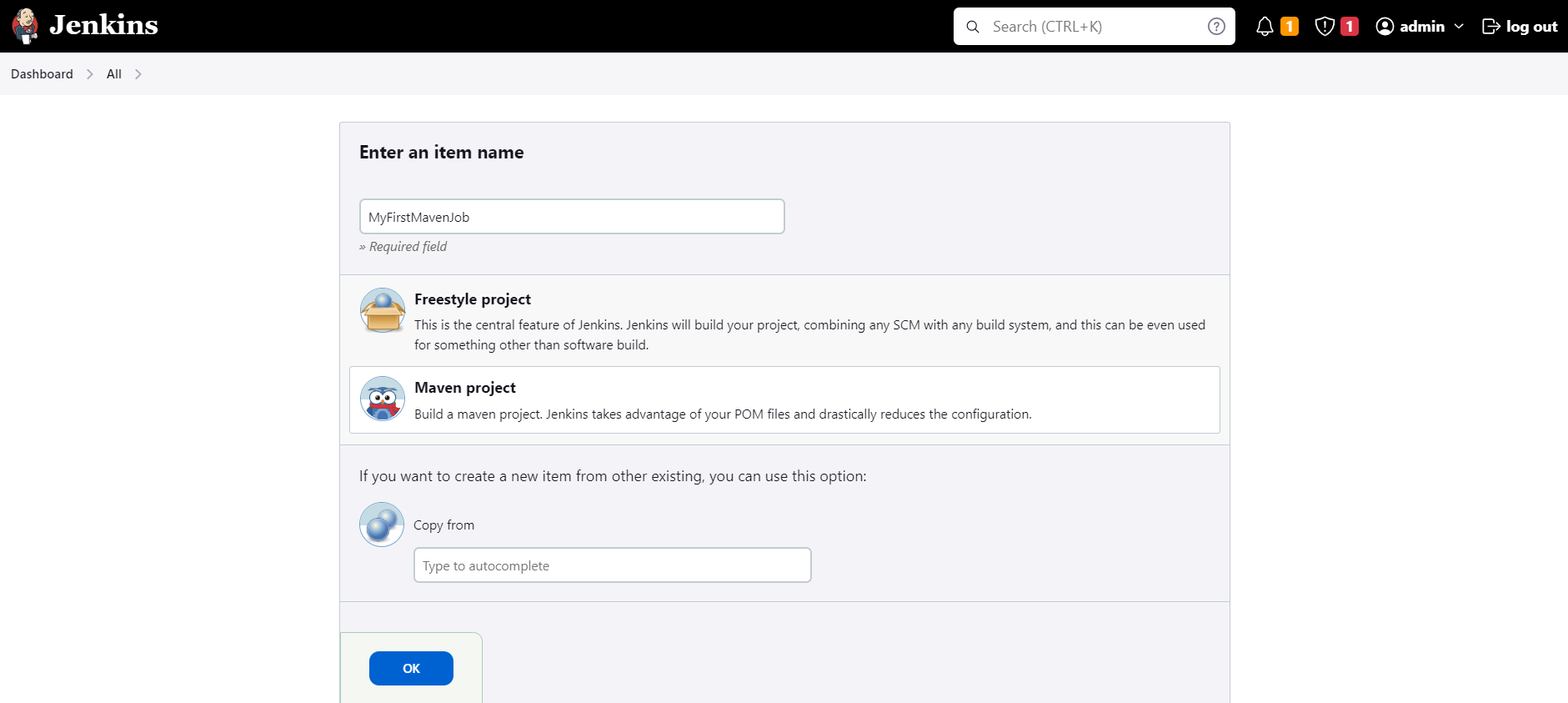
Now set maven path

Manage jenkins-> Tools -> maven -> add path click on apply and save

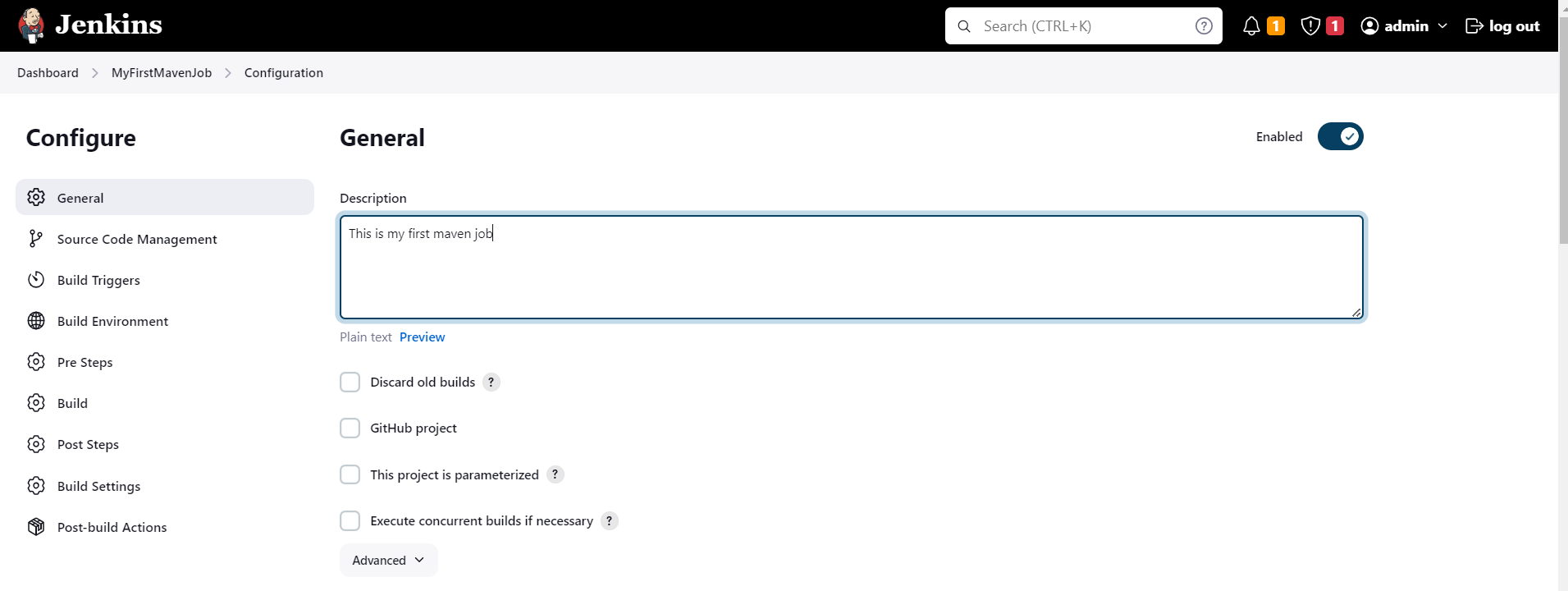


Create first maven job

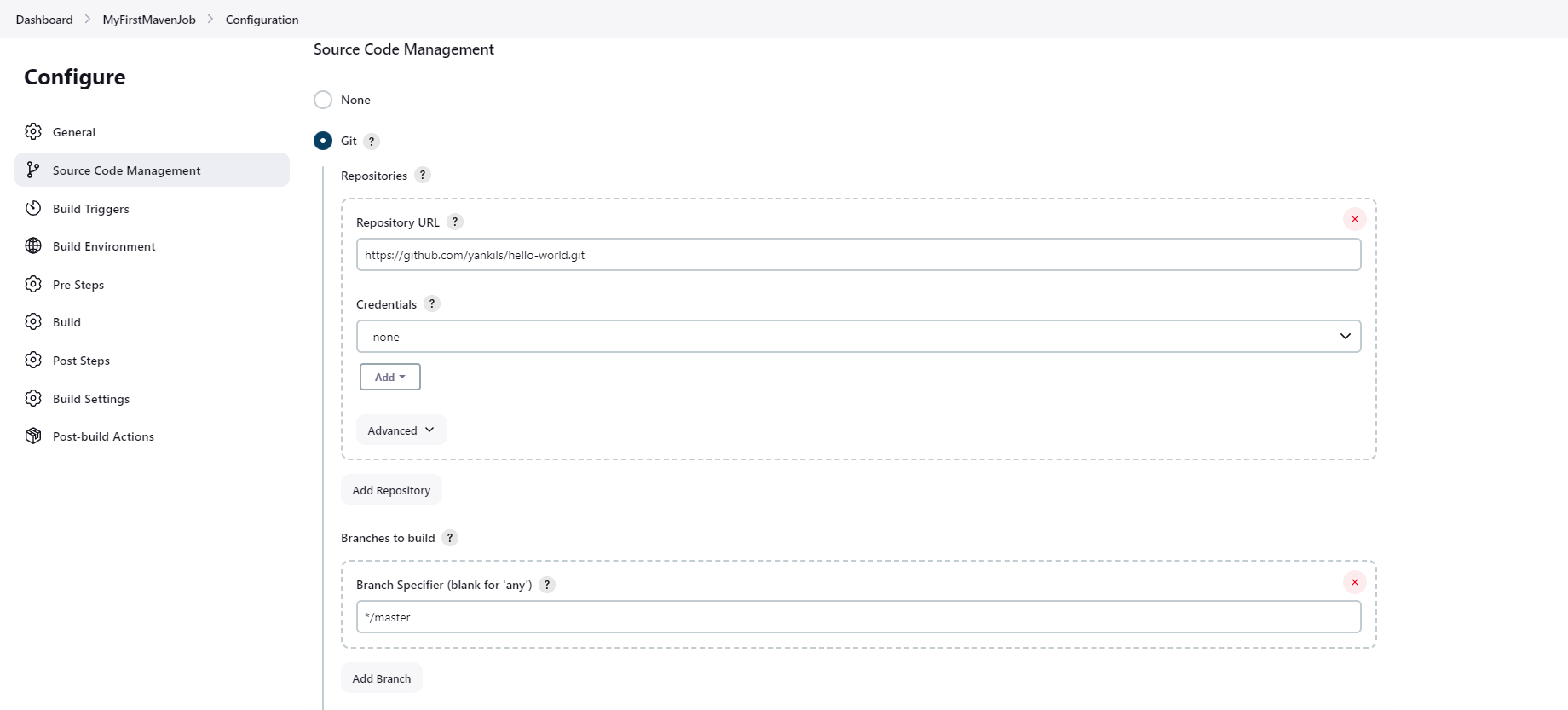
Dashboard->Add new Item-> give item name -> select maven project->click on ok

****

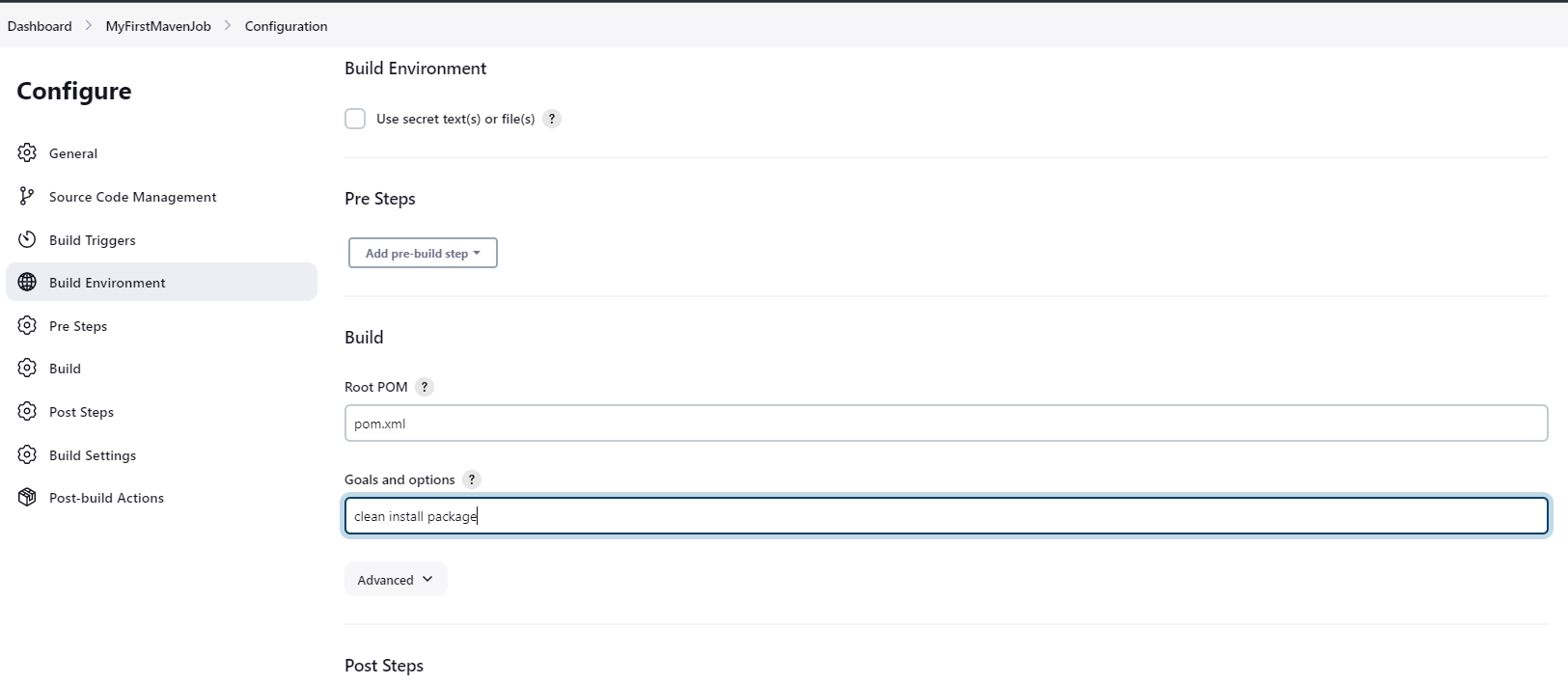
Add general description of job

****

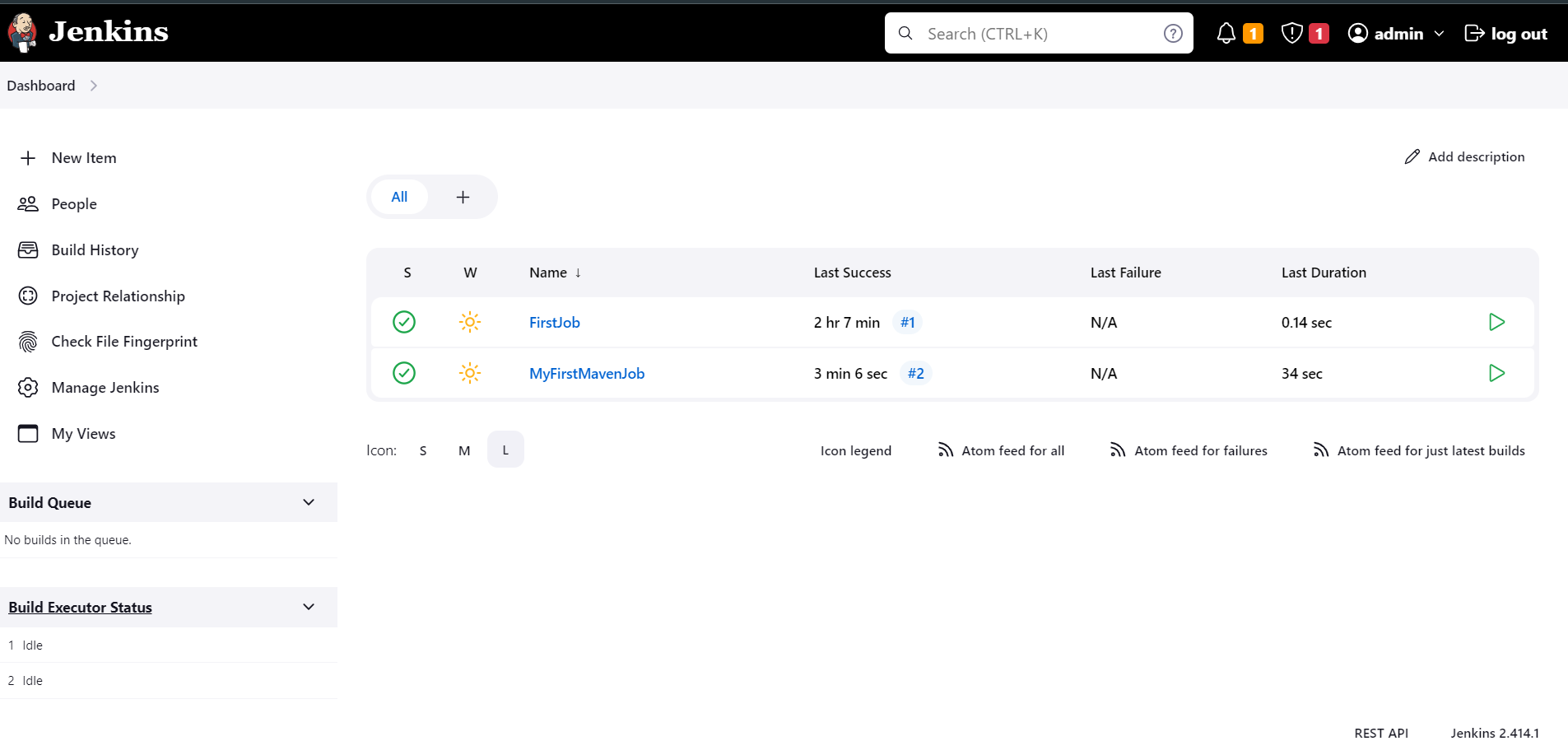
Select source code management -> git -> add git project repo url -> if it is private then only add credentials. If you have to pull from master branch then use “\*/master” instead of master you can specify your required branch, from which project will be pull inside jenkins.



Pom.xml required to build job of maven project. Hence this file essential in your github repo . click on apply and save and click on Build Now



It will change the status to green if job is builded successfully.



You can see the successfully builded job repos here also with webapp.war file required for maven.

cd /var/lib/jenkins/workspace/

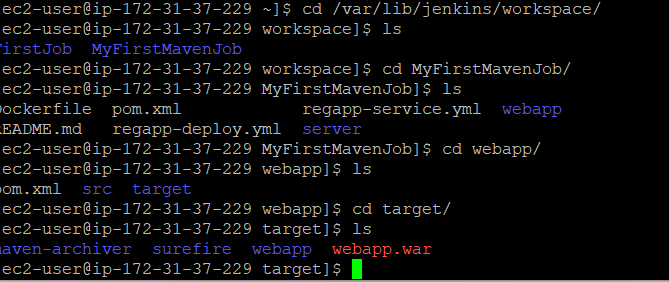
cd MyFirstMavenJob/

ls

cd webapp/

cd target/

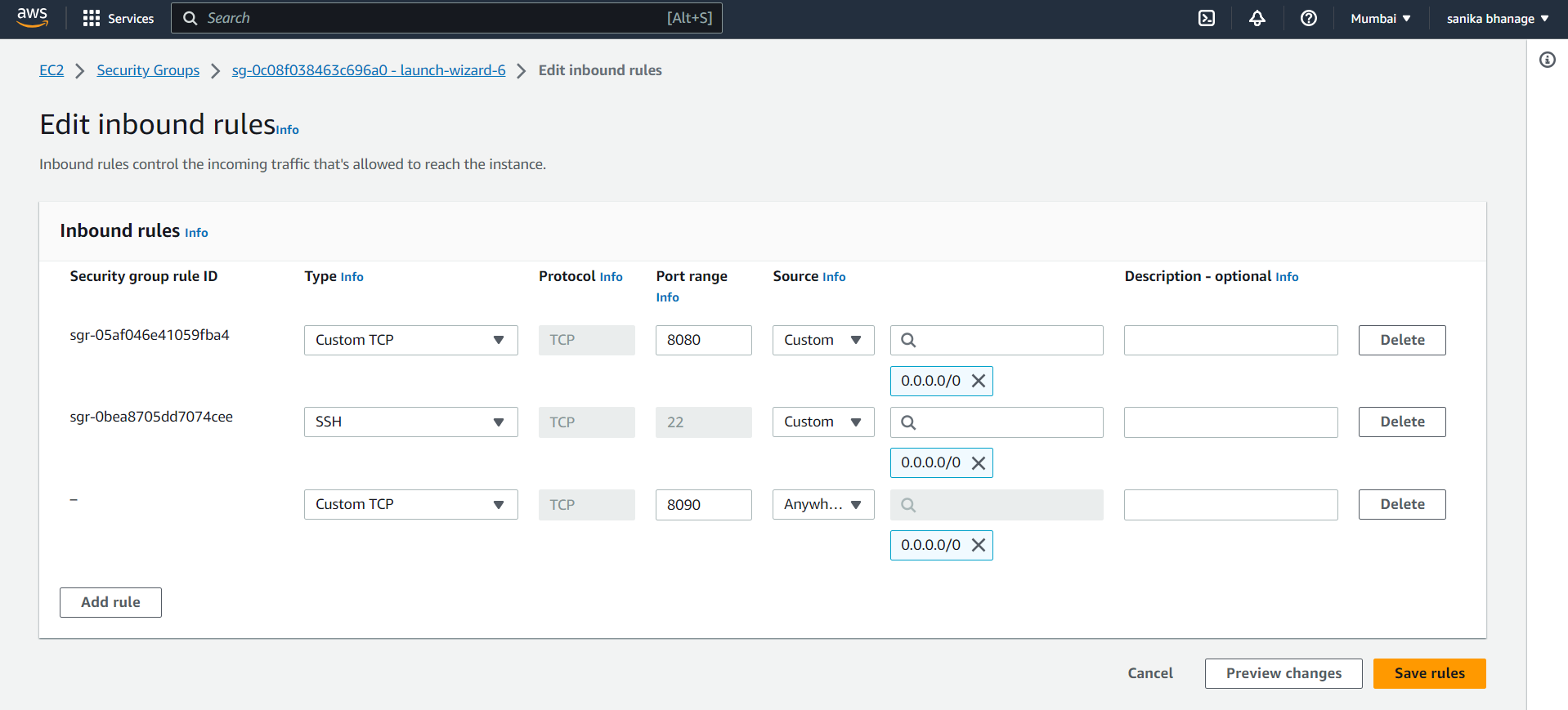
ls



To host this webapp.war we need tomacat server ,

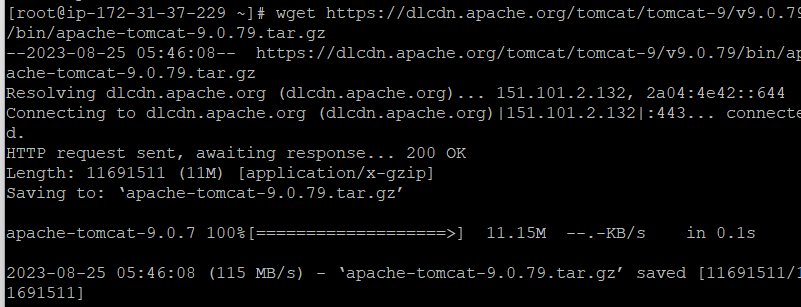
Setup Tomcat Server

Add the port 8090 in the security group as tomcat and jenkins both have same port number

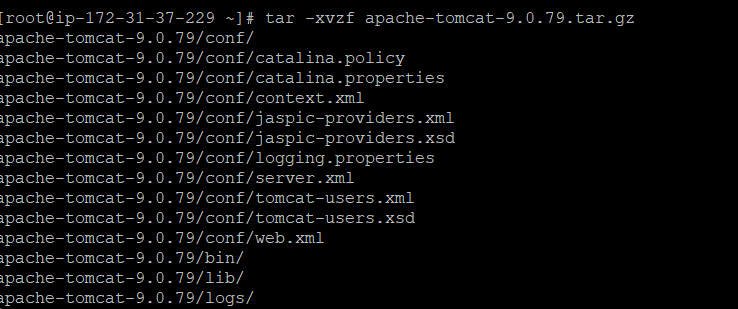


Download the tomcat latest version from its site

wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.79/bin/apache-tomcat-9.0.79.tar.gz



tar -xvzf apache-tomcat-9.0.79.tar.gz



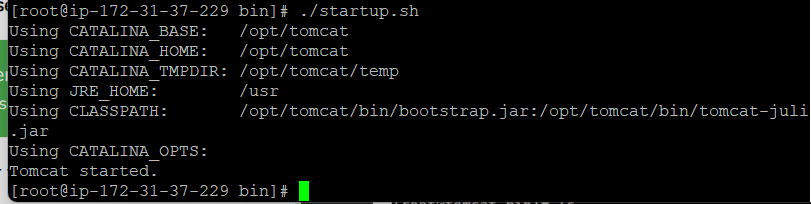
#mv apache-tomcat-9.0.79 tomcat

Got to the cd tomcat/; cd bin/; ls; you can see shutdown.sh , startup.sh Check if these files are executable or not if no then change there permissions.

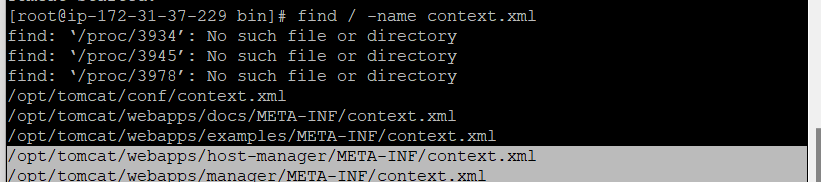


To start the tomcat server ,

./startup.sh



By default , tomcat allows login from local server. Hence to enable login and to change the default port to 8090 we need to find context.xml we need to change same line in both files /opt/tomcat/webapps/manager/META-INF/context.xml & /opt/tomcat/webapps/host-manager/META-INF/context.xml



In this file you will find the line that allows only local server to login for tomcat but if we comment it out then we can able to login from out side also.

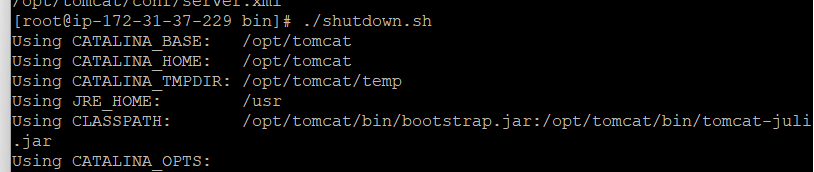
<Valve className="org.apache.catalina.valves.RemoteAddrValve"

allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" />



To change the default port , go to /opt/tomcat/conf/server.xml change 8080 -> 8090. Before that run shutdown.sh

./shutdown.sh





Added required roles in /opt/tomcat/conf/tomcat-users.xml.

<role rolename="manager-gui"/>

<role rolename="manager-script"/>

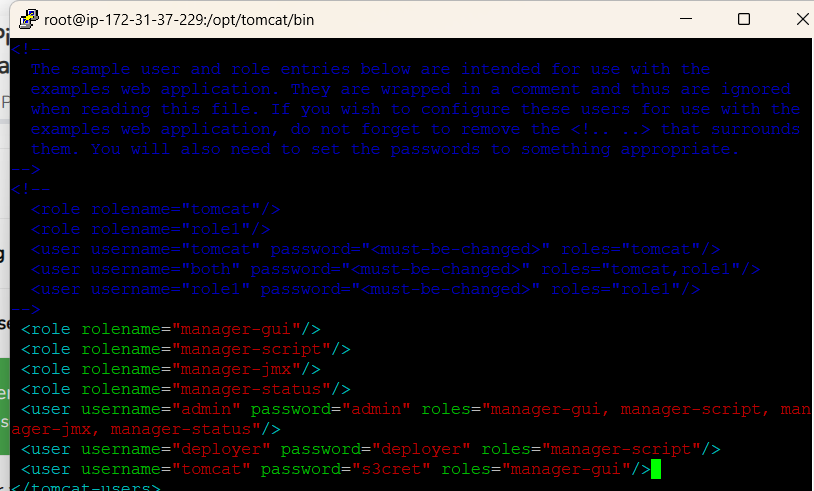
<role rolename="manager-jmx"/>

<role rolename="manager-status"/>

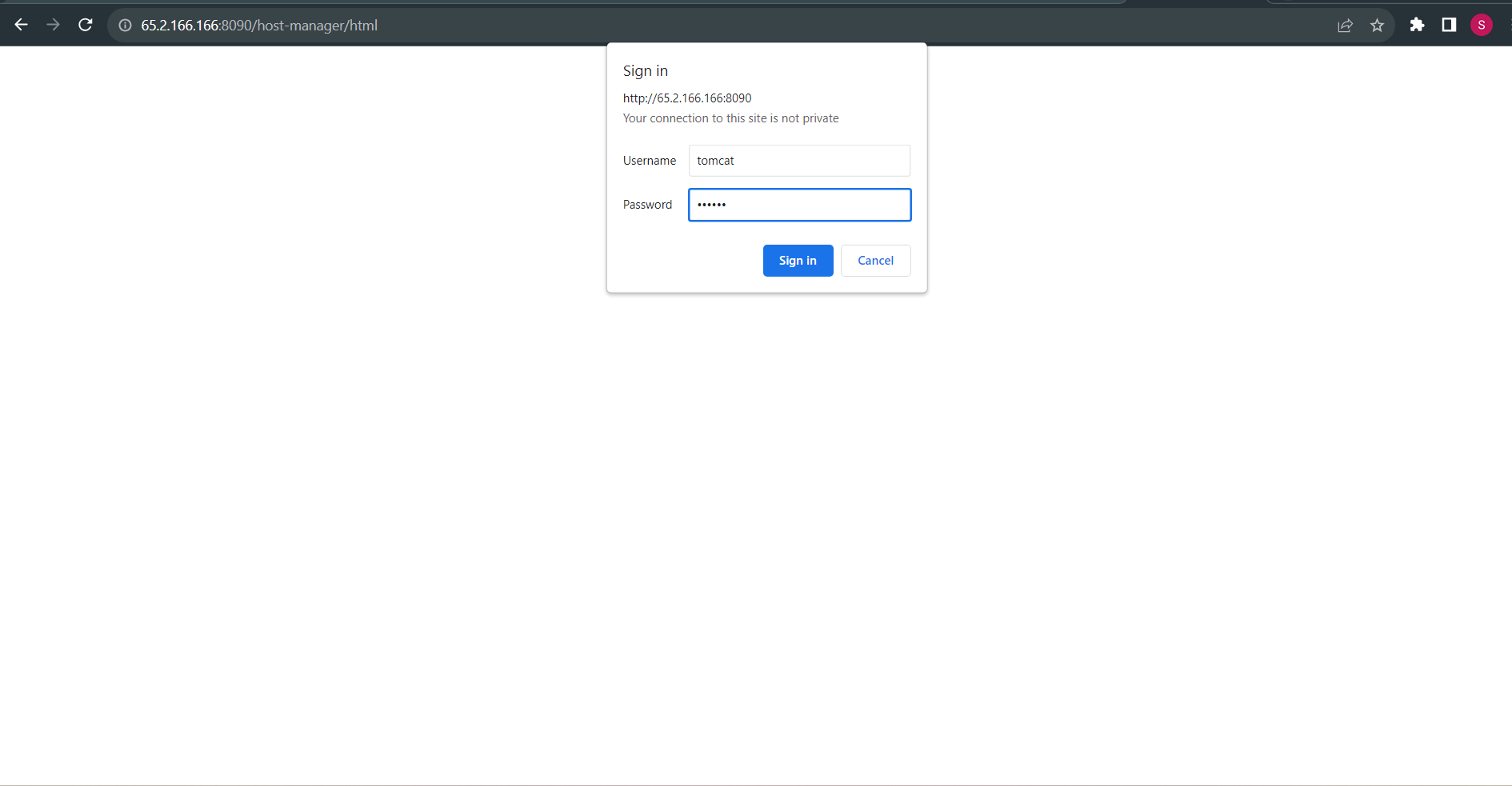
<user username="admin" password="admin" roles="manager-gui, manager-script, manager-jmx, manager-status"/>

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

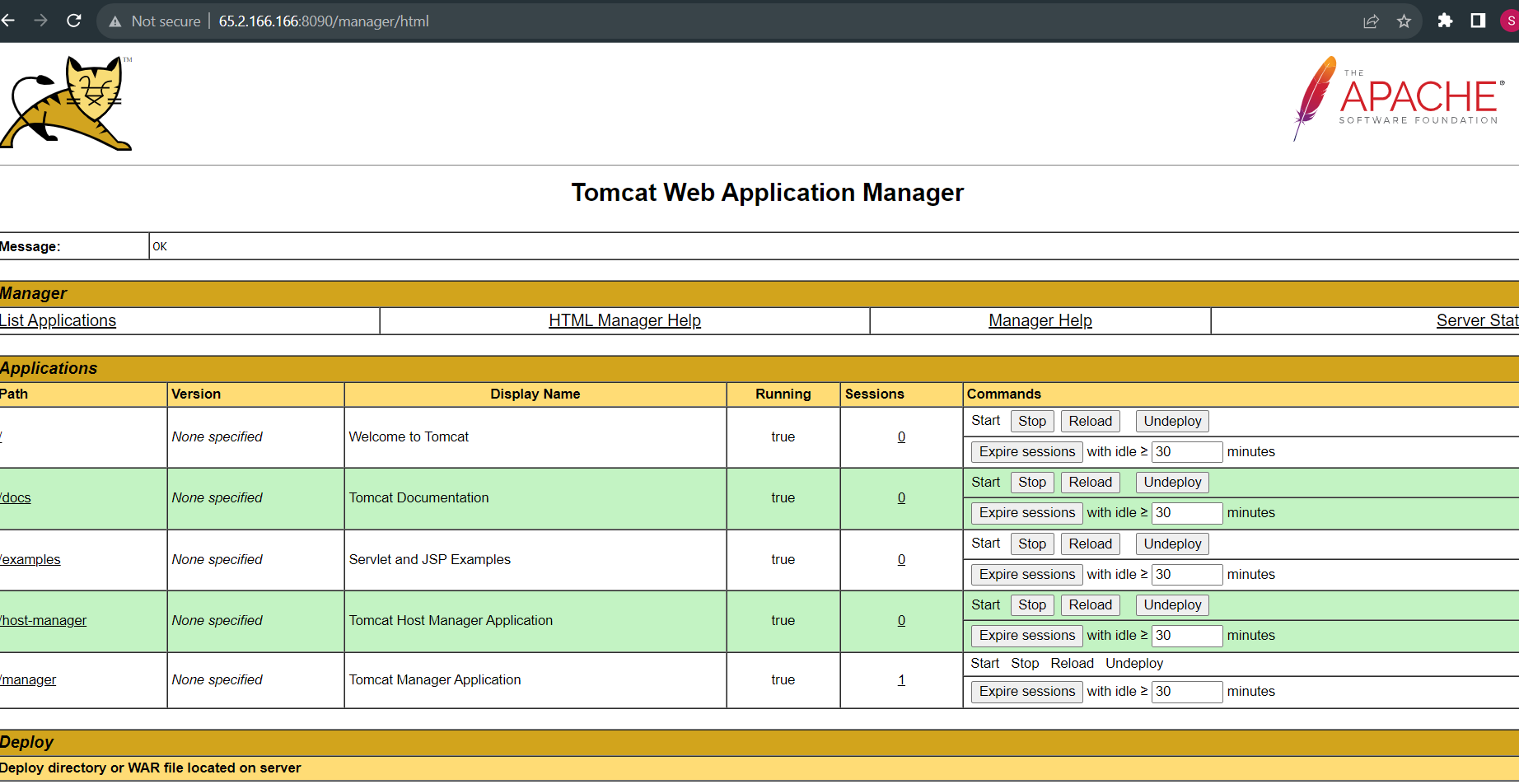
<user username="tomcat" password="s3cret" roles="manager-gui"/>



Again run shutdown.sh and startup.sh and in browser add credentials for manager-gui.

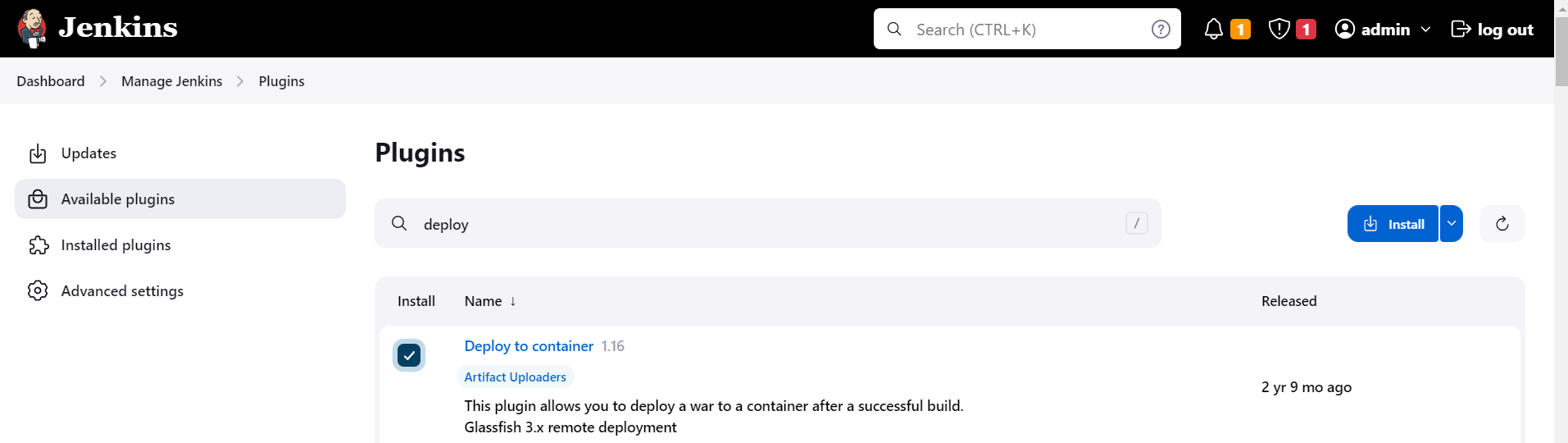


After login,

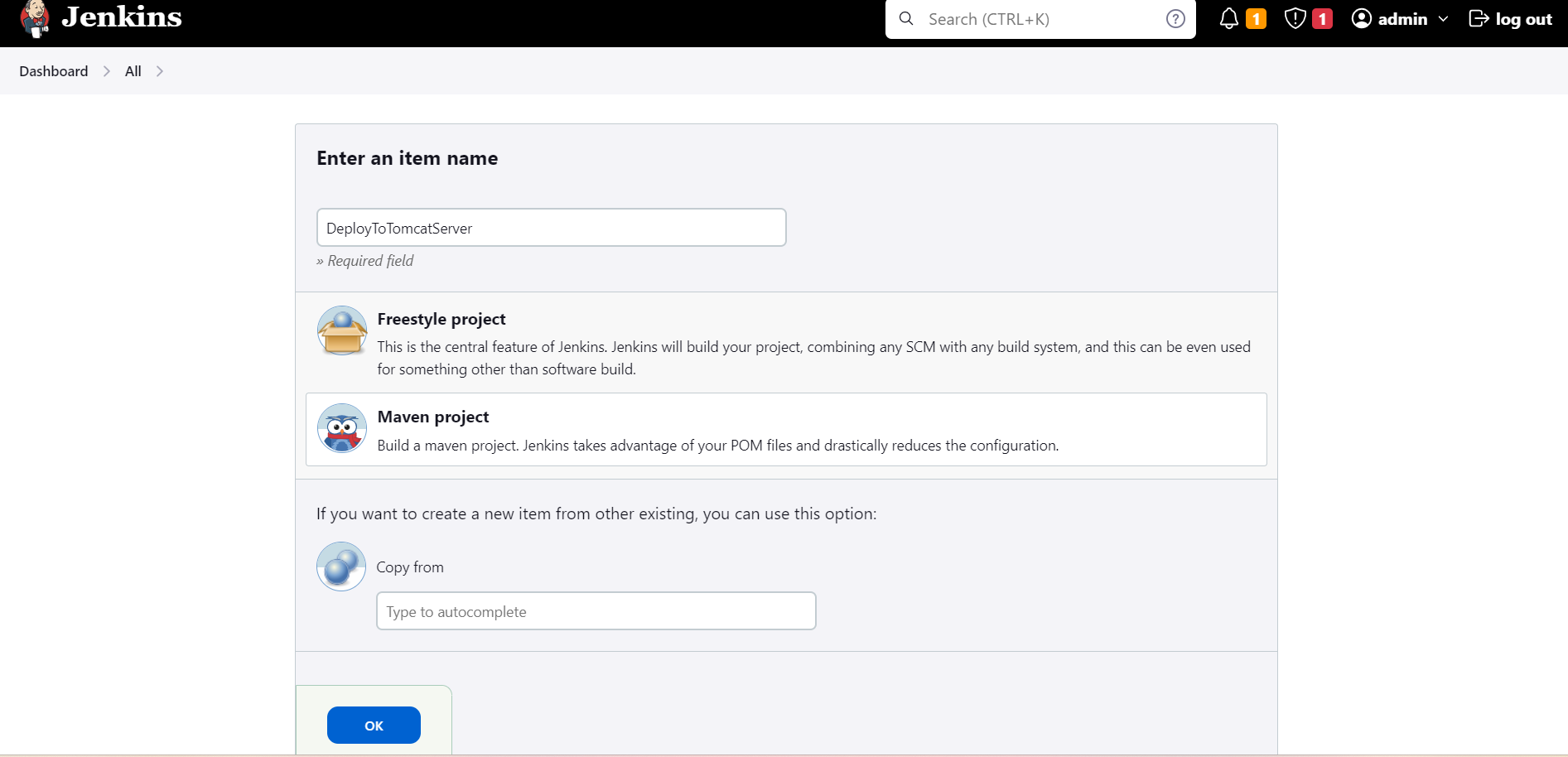


To integrate tomcat with jenkins we required plugin for tomcat in jenkins. Manage jenkins -> plugins -> Available plugins

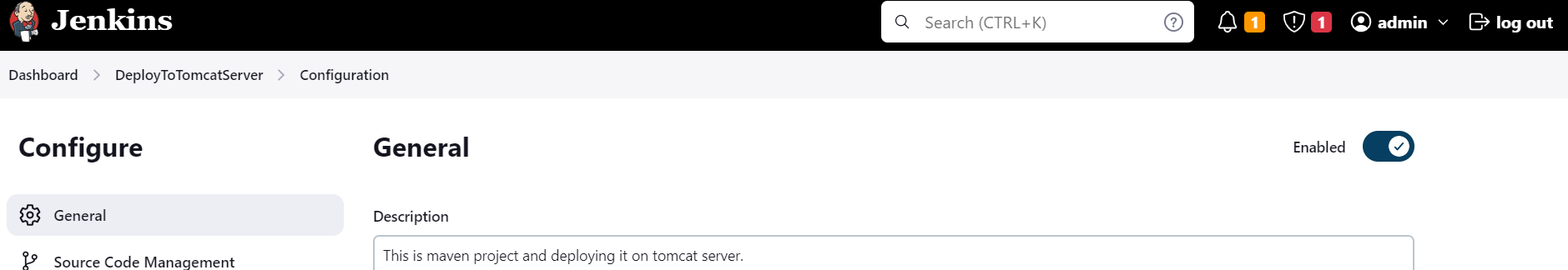
To deploy project on any vm machine we required “deploy to container” plugin. Click on install.



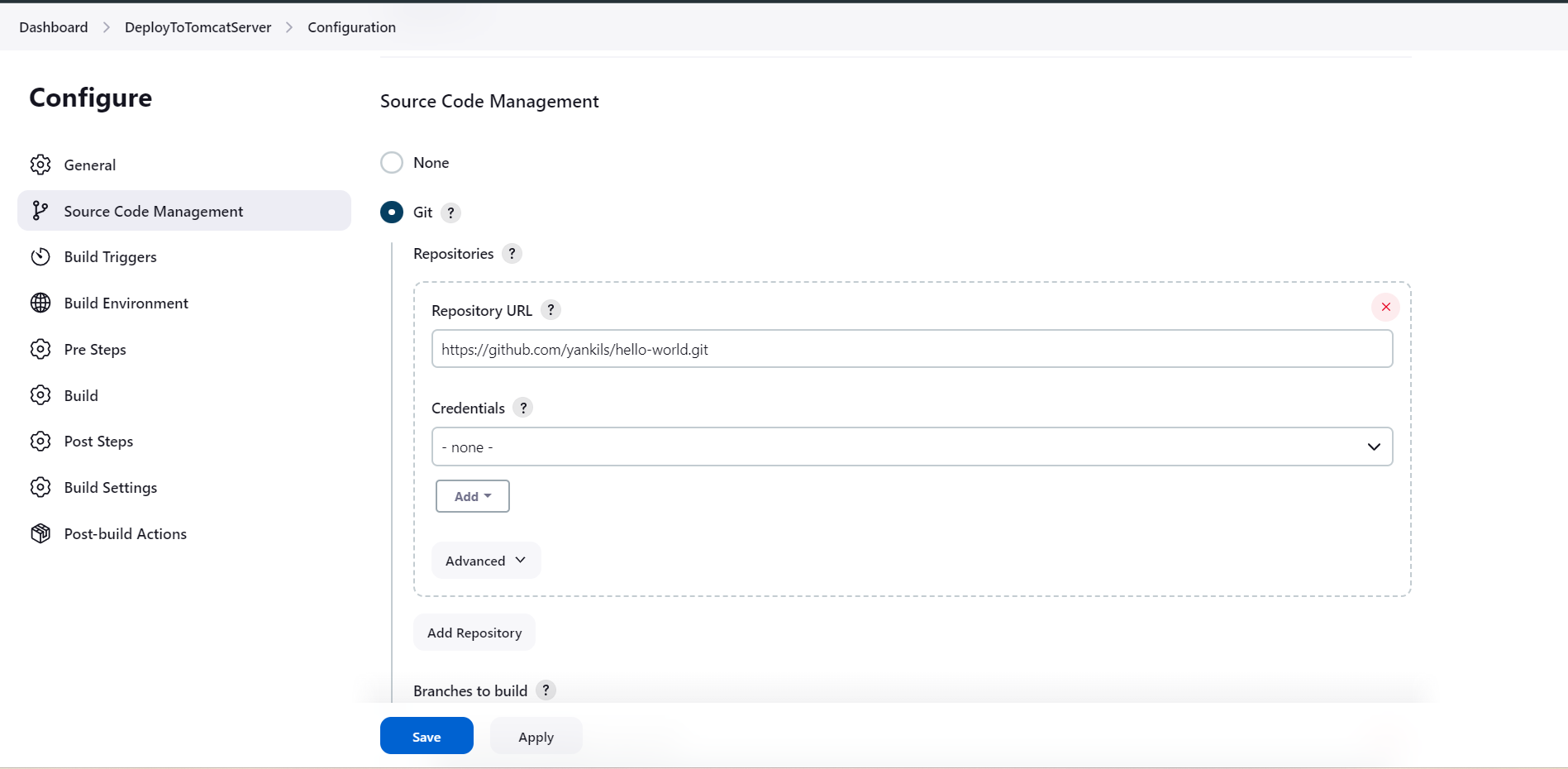
After installation build the maven job and deploy it on tomcat server. Select maven project.



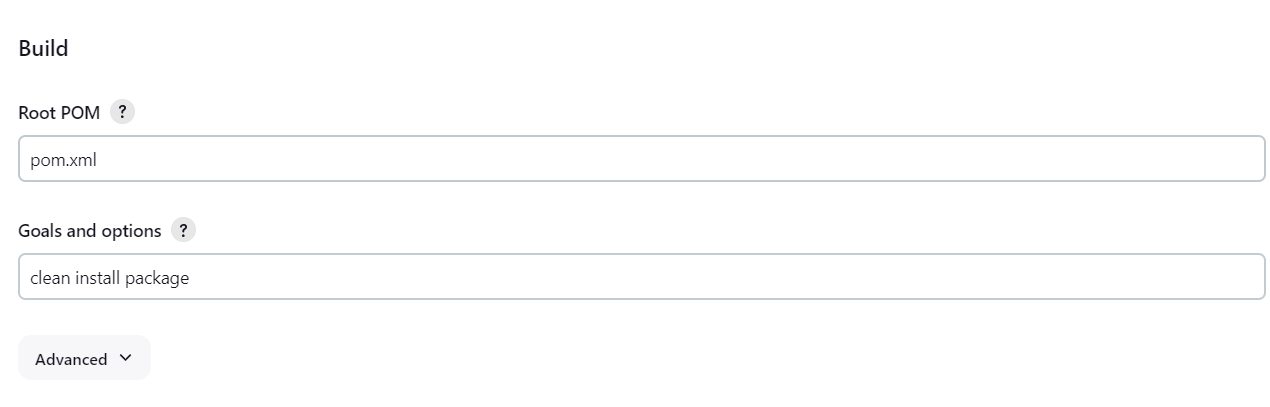
Add general job description



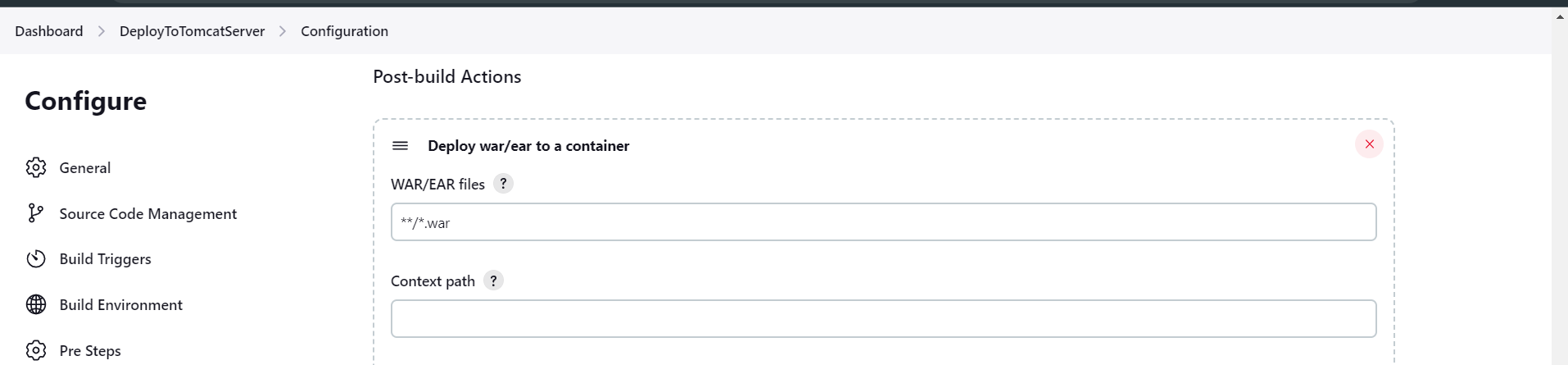
Add the github repo url



For build, pom.xml is required



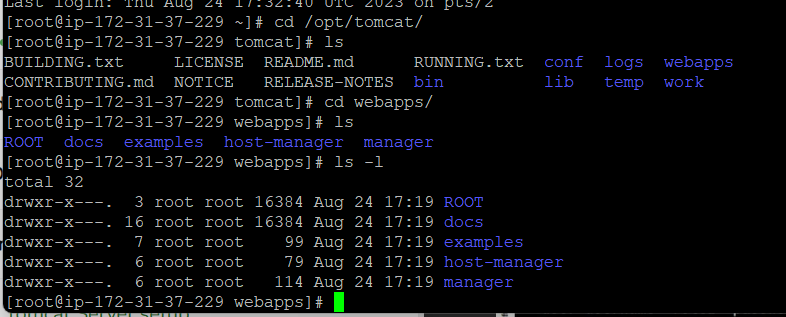
Add the war file details. If you don’t know file name simply add \*\*/\*.war



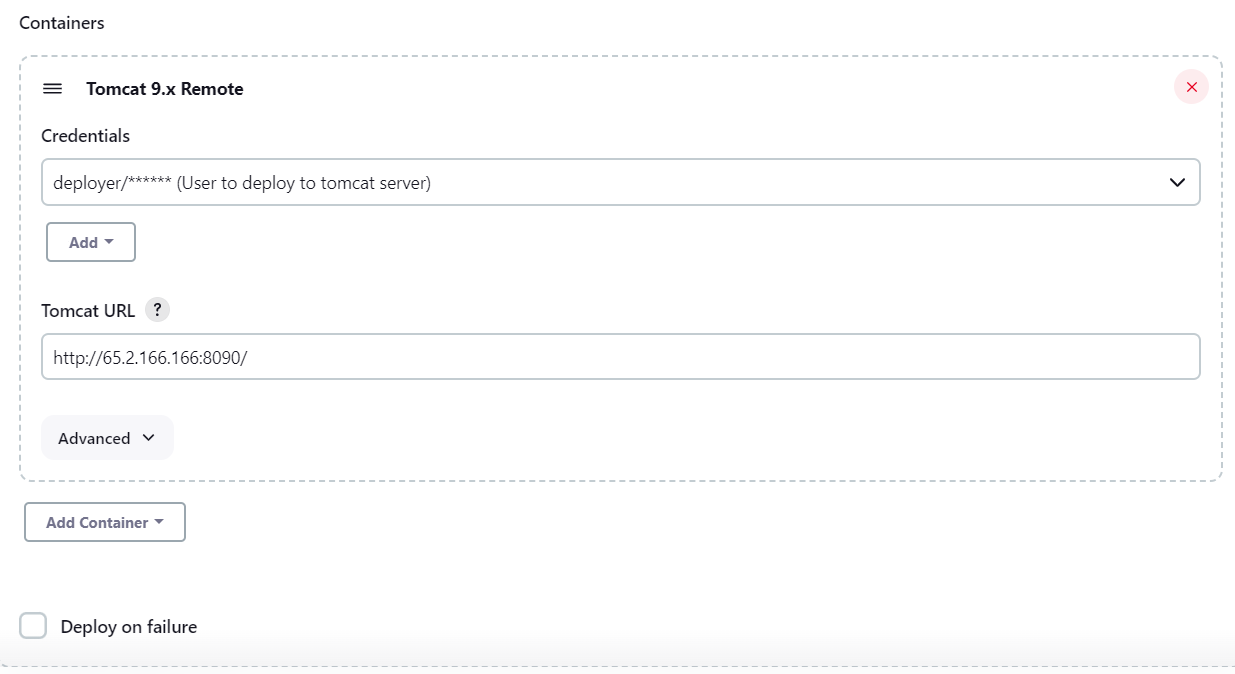
Select tomcat9 server for deployment. We have to add deployer user credential that we added already in the conf/tomcat-users.xml file. Username : deployer Password: deployer Click on save.



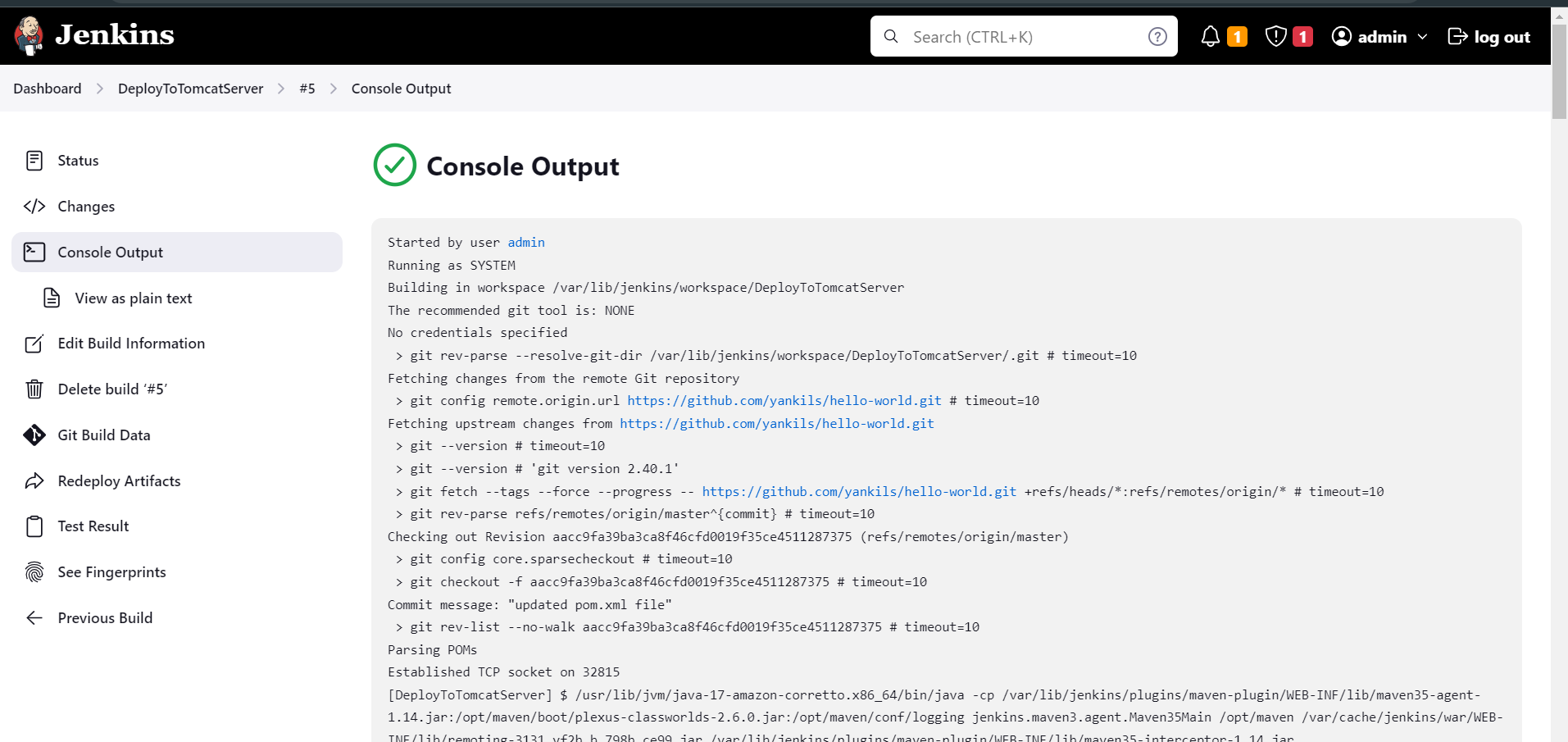
Also add tomcat url and click on apply & save and build now. In vm we can check the path for webapp where war file.



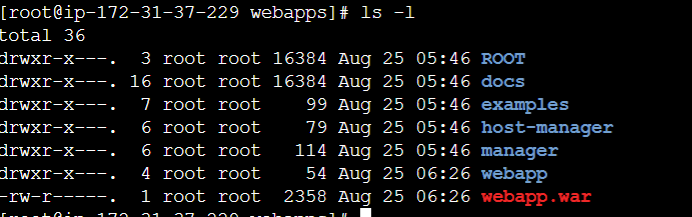
Click on apply and save and build now.



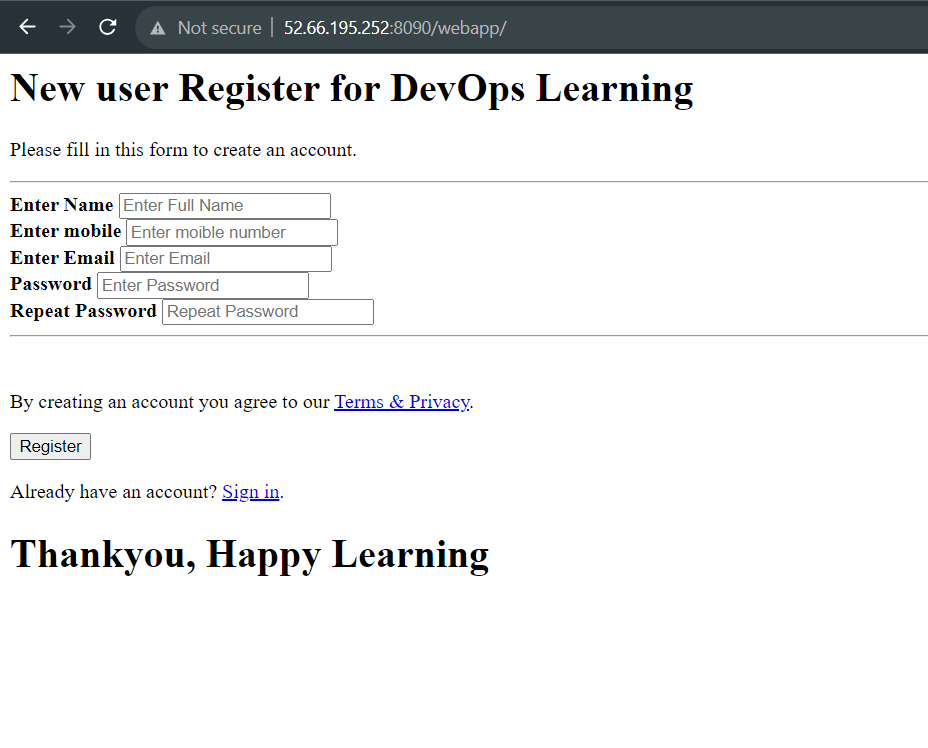
After successfully job build is done,



In the tomcat/webapps we can see the webapp.war file which is required project deployment.

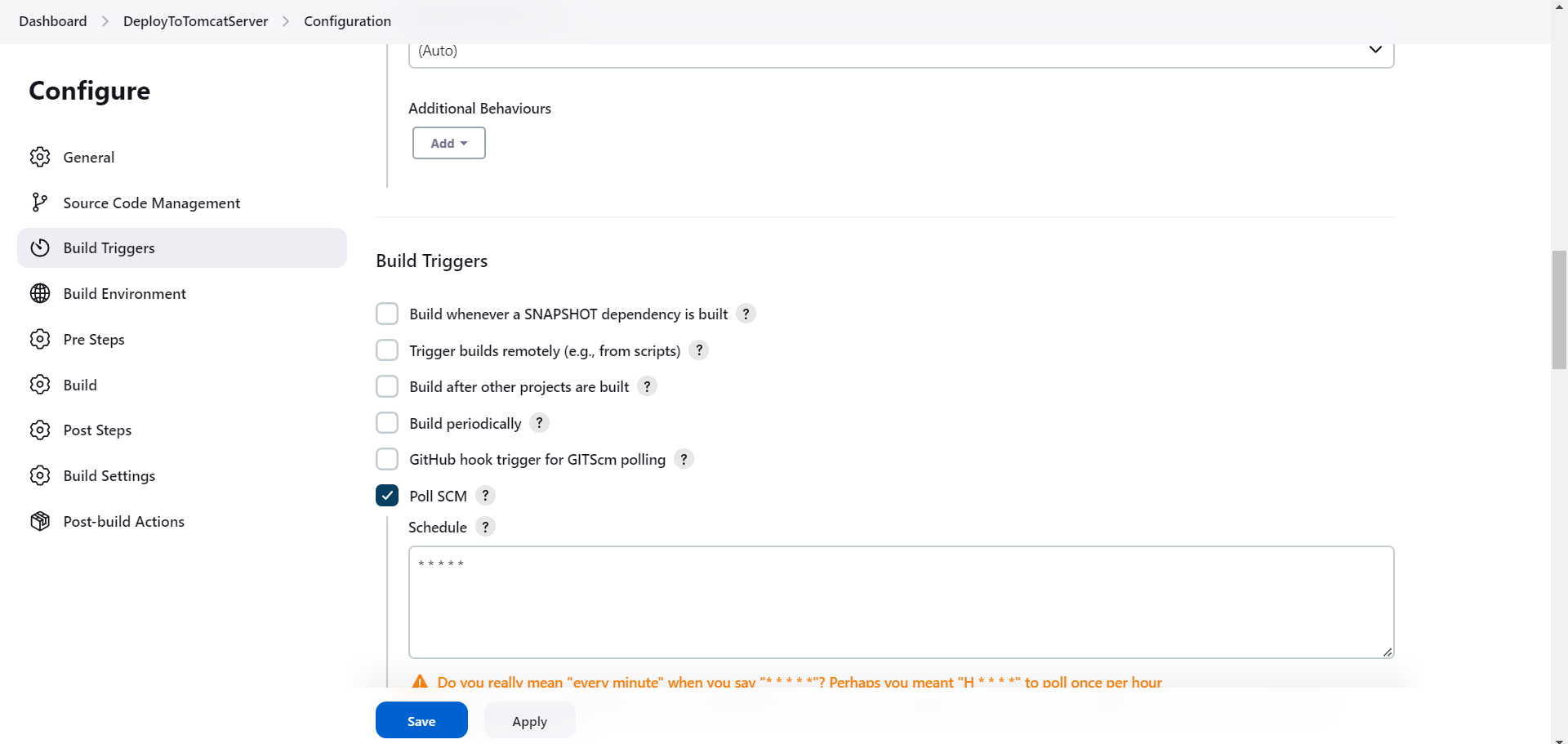


In the browser , search http://<public-ip>:8090/webapp You can see project deployed successfully on tomcat server.



Deployment of project using POLLSCM, means job will run after specified period of time

In the Build trigger select POLL SCM \*\*\*\*\* (every minute,every hour,every day,every month,every week) click apply save and build now

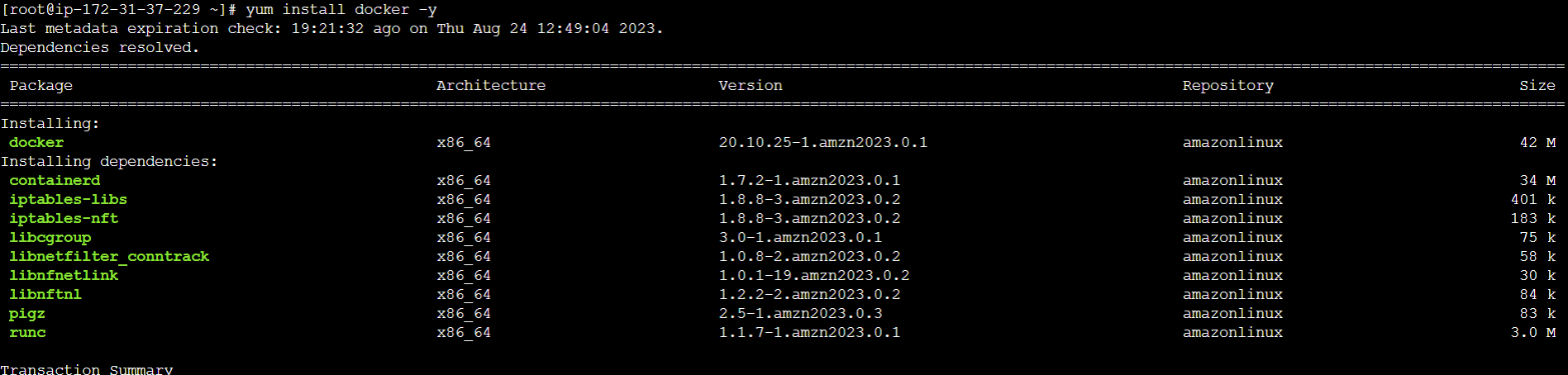


Commit the new changes in the maven project and push them to github you will see new changes occurred in webpage,



Integrating Docker into Jenkins

#yum install docker -y

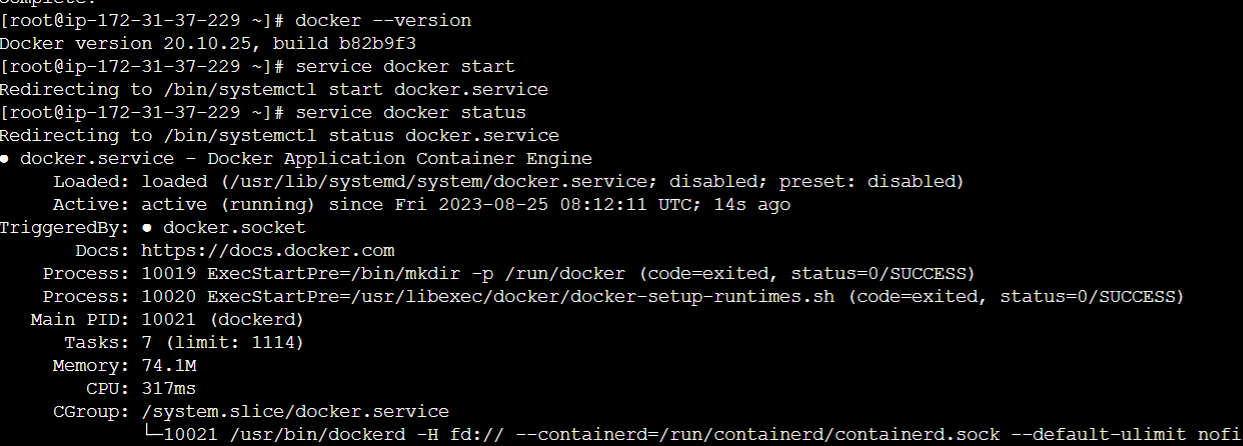


#docker -version

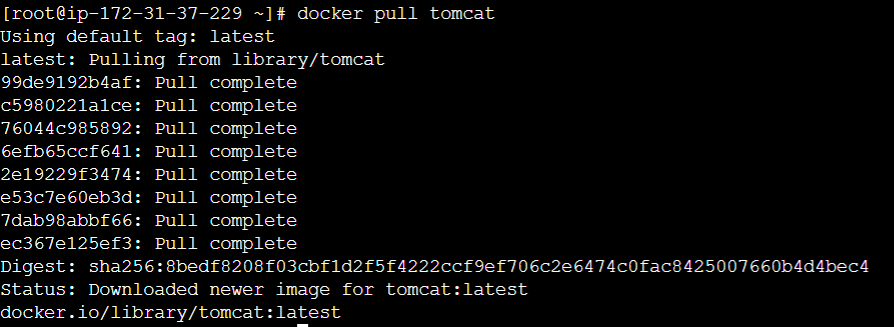
#service docker start

#service docker status

#systemctl enable docker

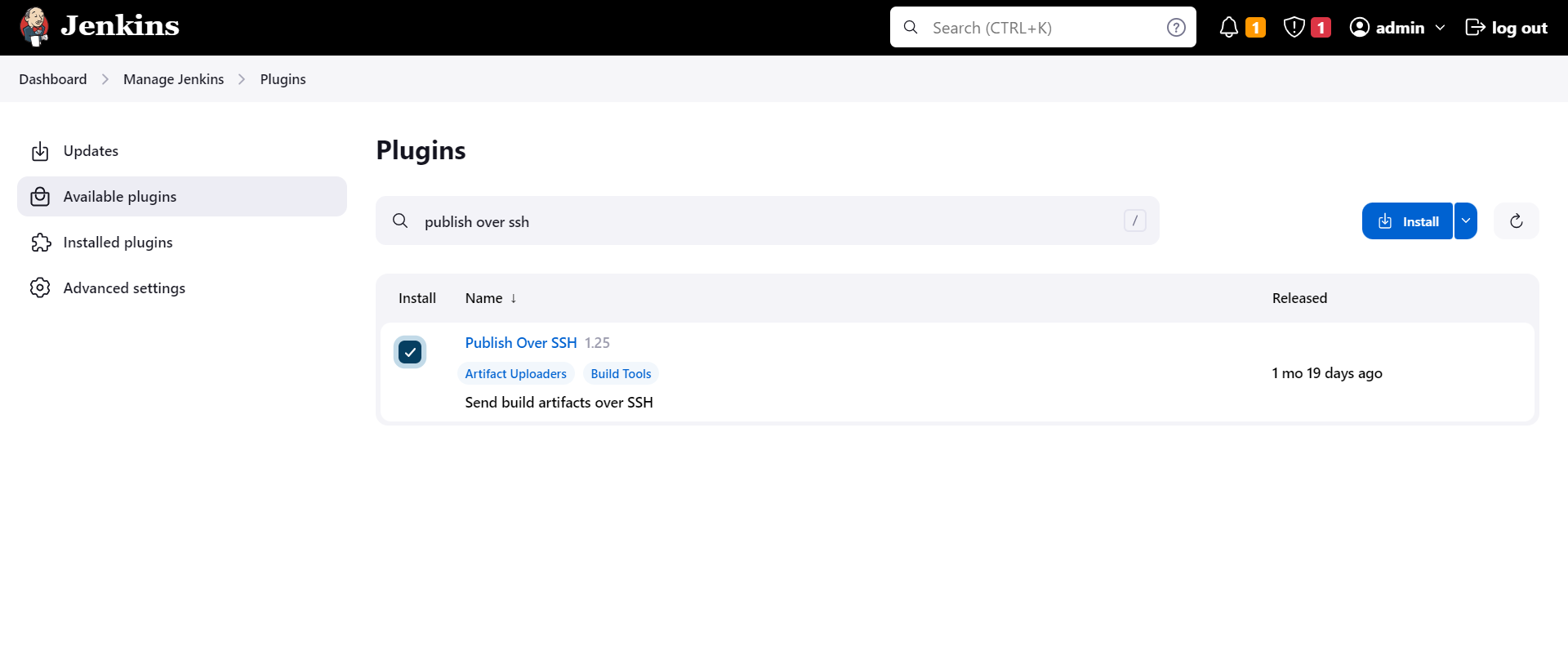


#docker pull tomcat



Create container using this docker image

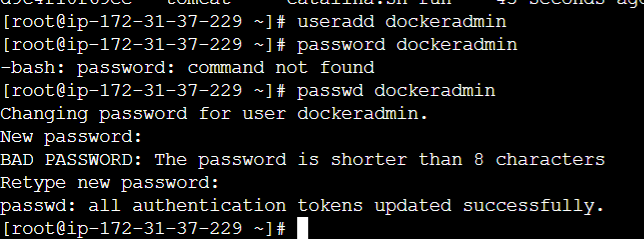
To integrate docker into Jenkins , search plugin called as “publish over ssh” and install.



Add docker credentials

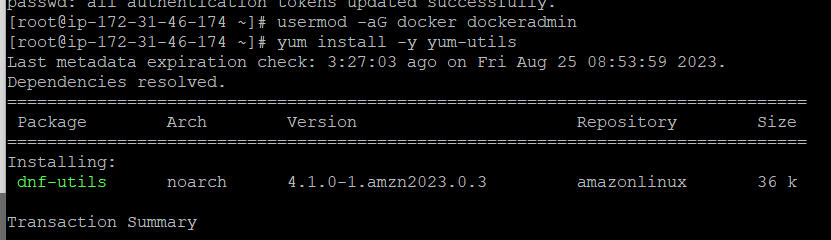
#useradd dockeradmin

#passwd dockeradmin



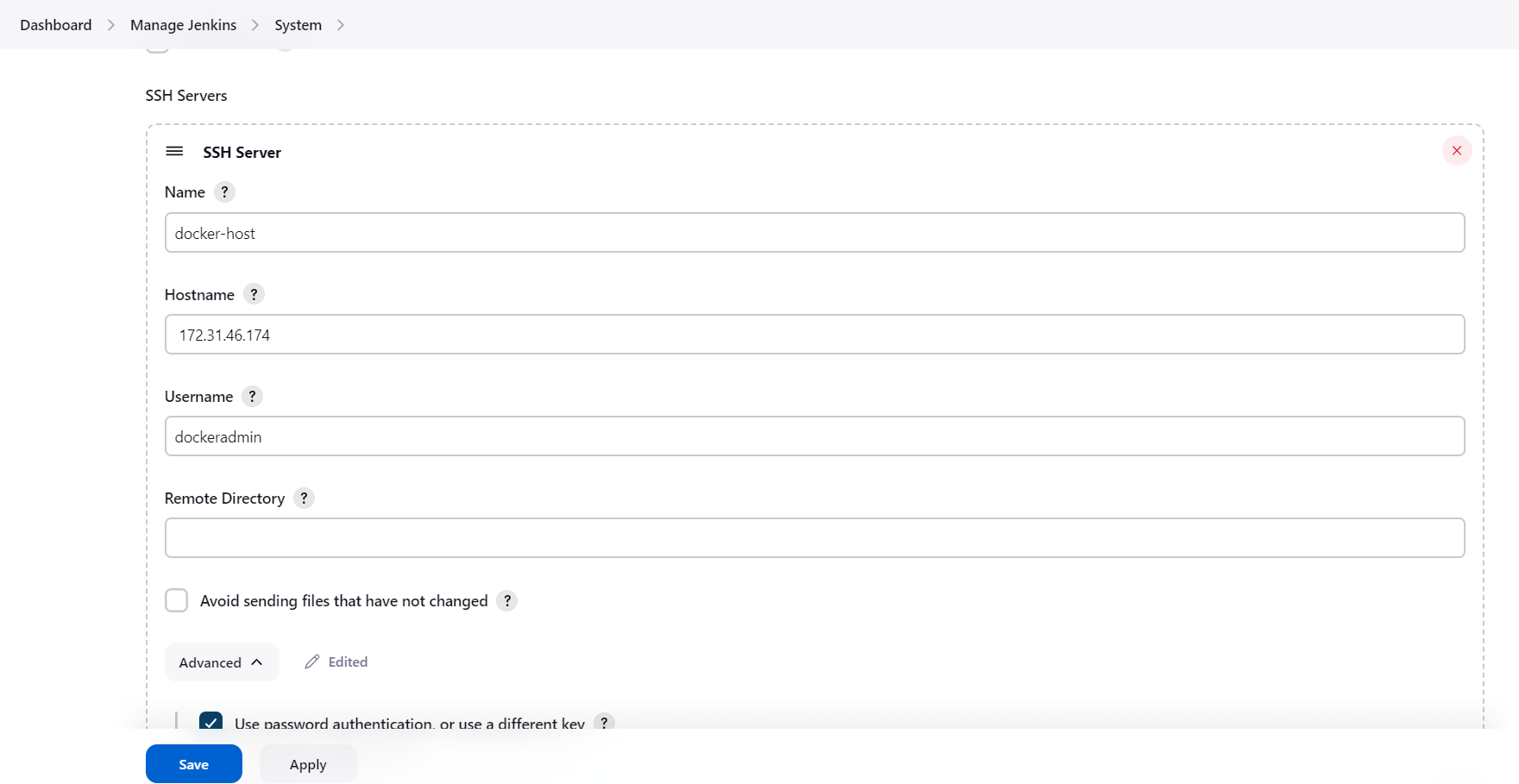
#usermod -aG docker dockeradmin

#yum install yum-utils -y

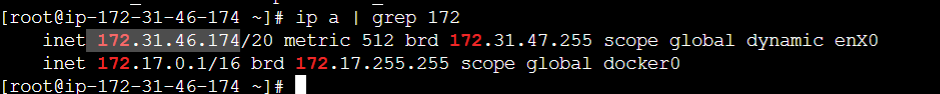


Setup docker into Jenkins

Goto the Manage Jenkins->configure->Publish over ssh-> click on add SSH Servers, provide you ip address as a hostname, dockeradmin as a user and name for the service.

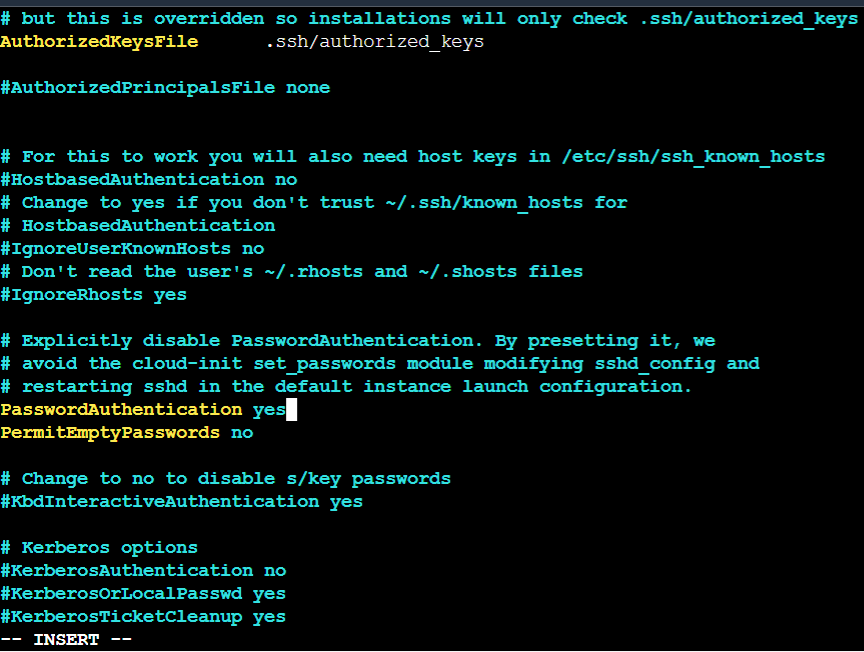


Check for ip address copy it and paste

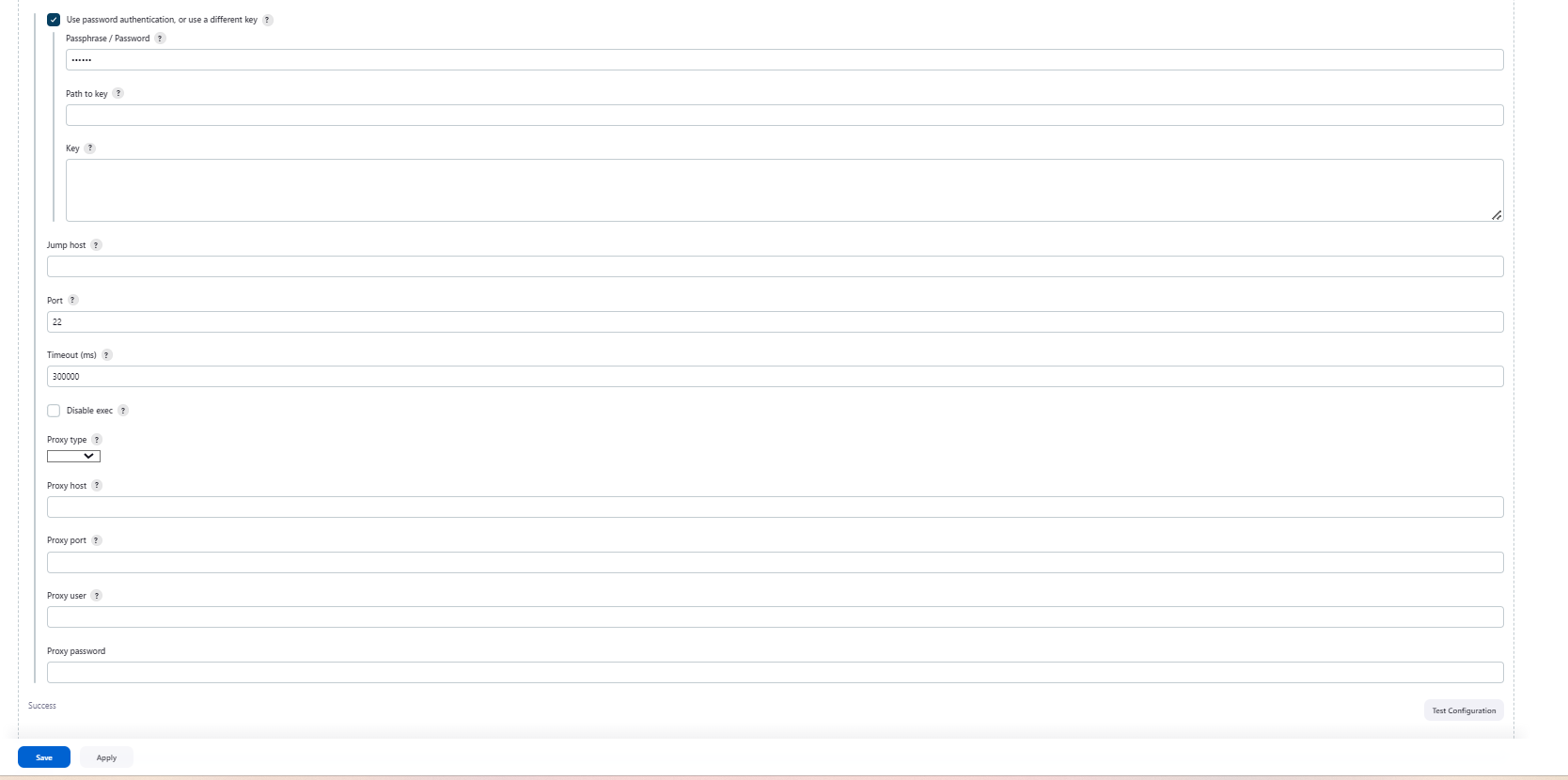


Now, we have to configure password authentication for the docker service. Hence goto your docker

In the file /etc/ssh/sshd\_config change PasswordAuthentication to yes from no.



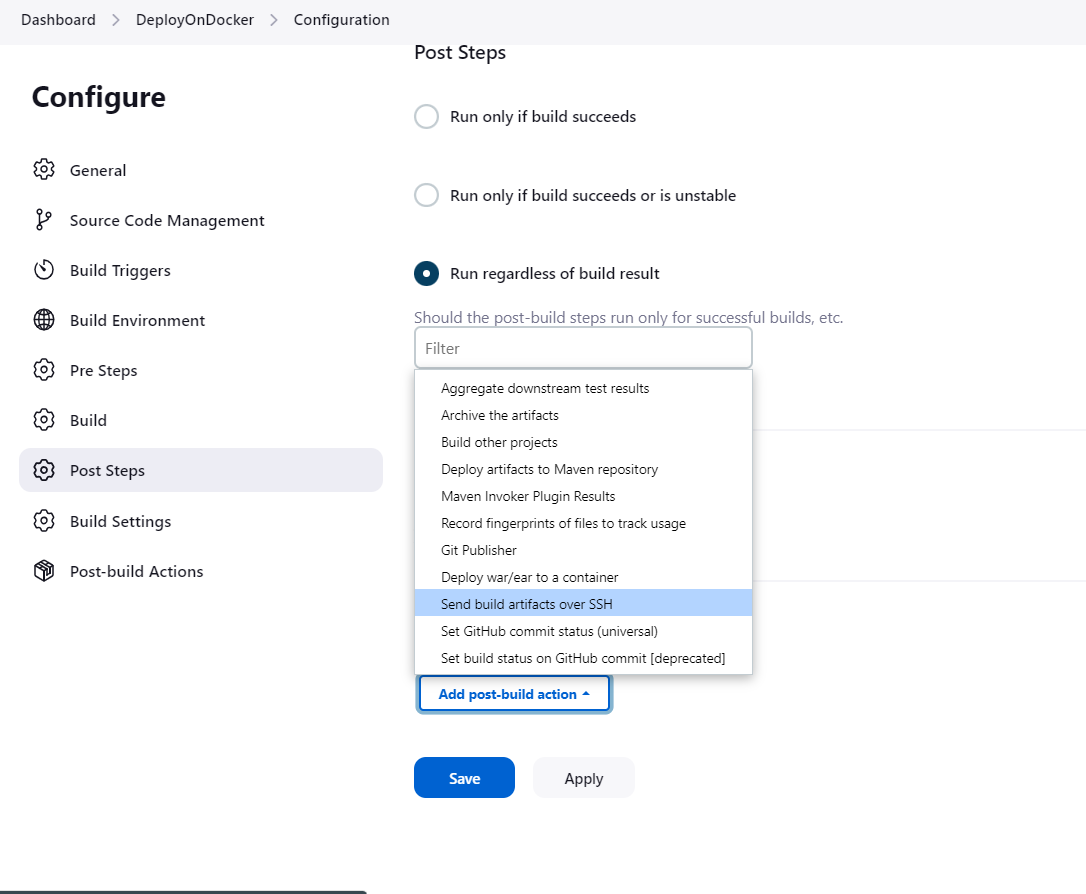
Now in Jenkins, select password authentication, provide password. And test the connection if it is successful then click on apply and save.



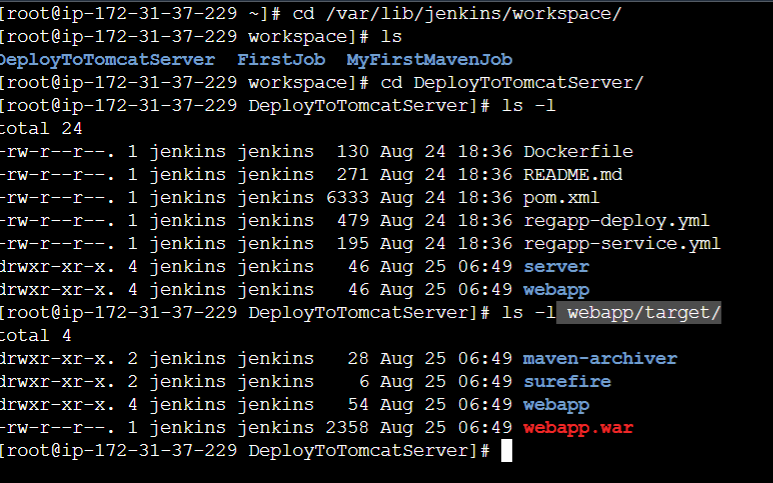
Now, to test deployment on docker create new job



In this job, we have to build artifacts over ssh hence remove tomcat9 deployment and select build artifact over SSH.



Copy the war file path and paste it , you can see that the path is in previous job build



Select our SSH server , give source files webapp/target/\*.war and Remote Directory is “.”

