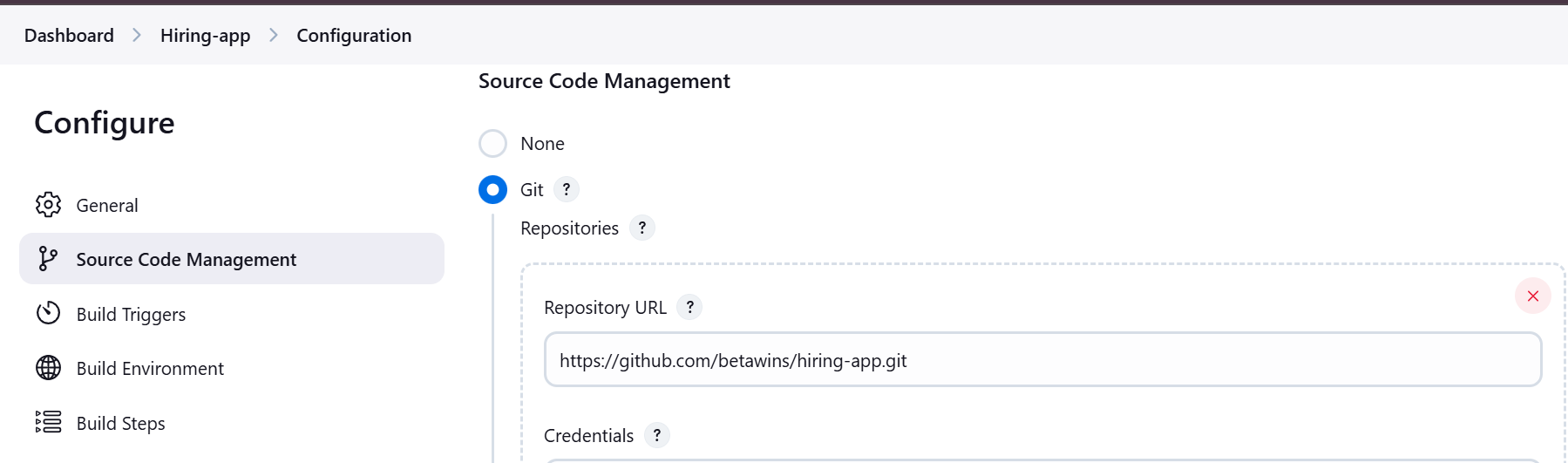
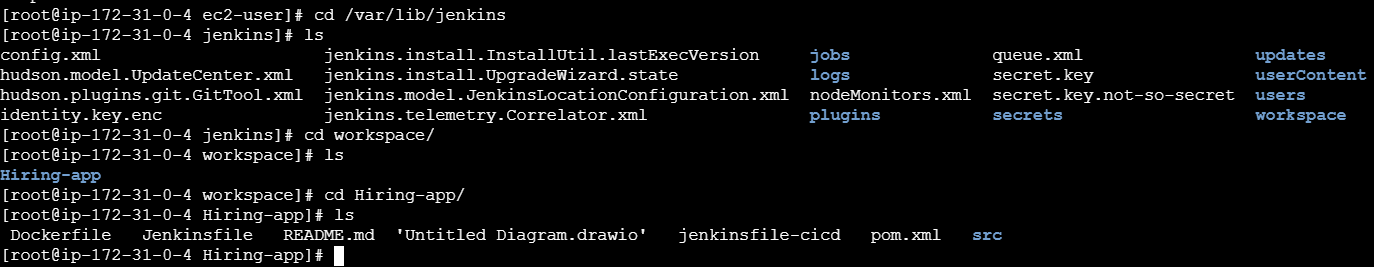
JENKINS-04

1**) Setup jenkins CICD pipeline using freestyle job using Docker containers using below code.**

**https://github.com/betawins/hiring-app.git**

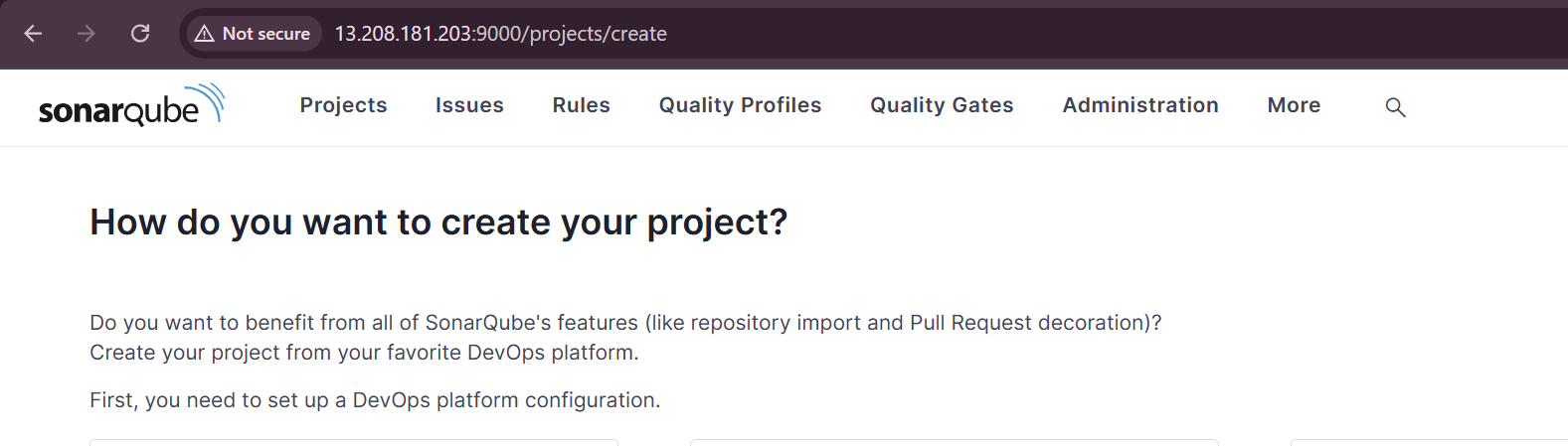
**Stages:**

**1) Git Clone**  

**2) Sonarqube Integration**

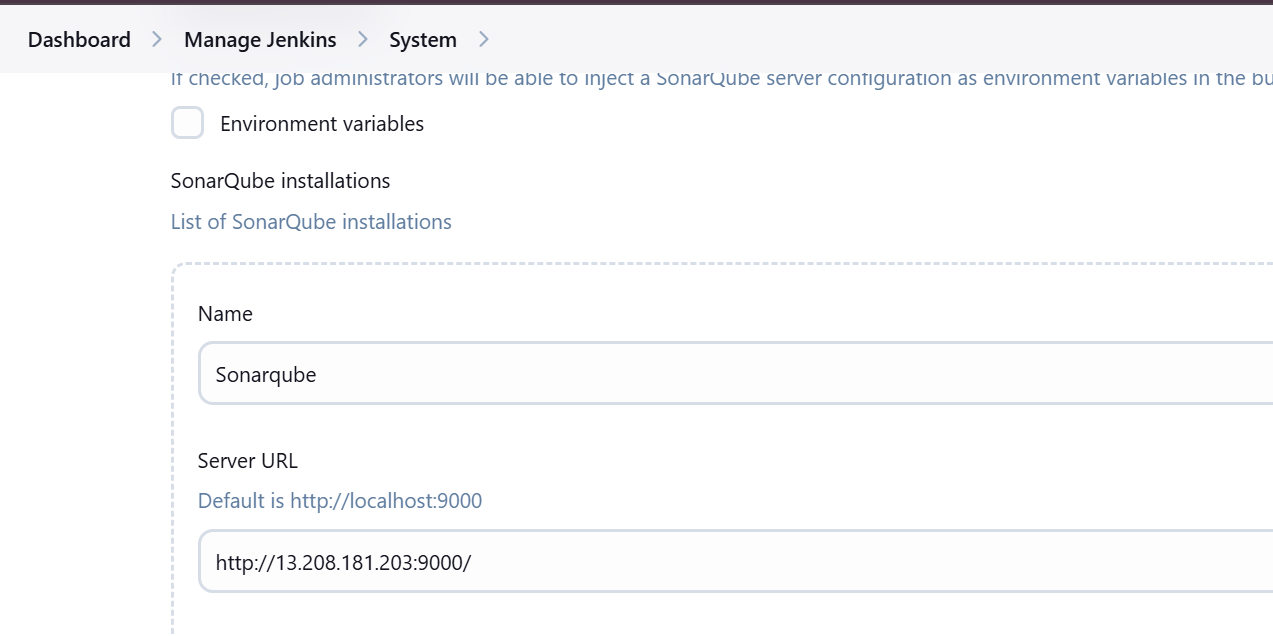
Using Docker for sonarqube



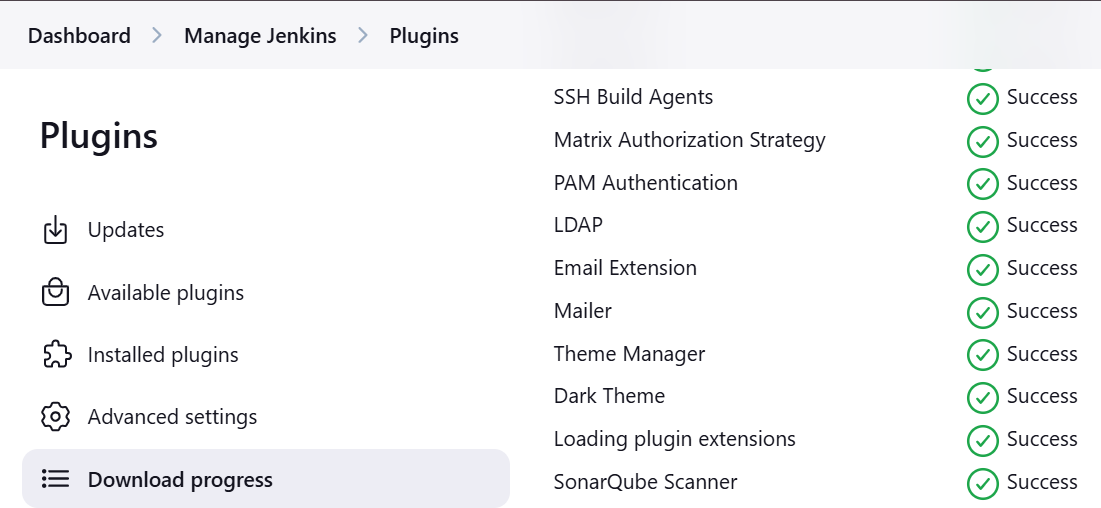


Go to Jenkins

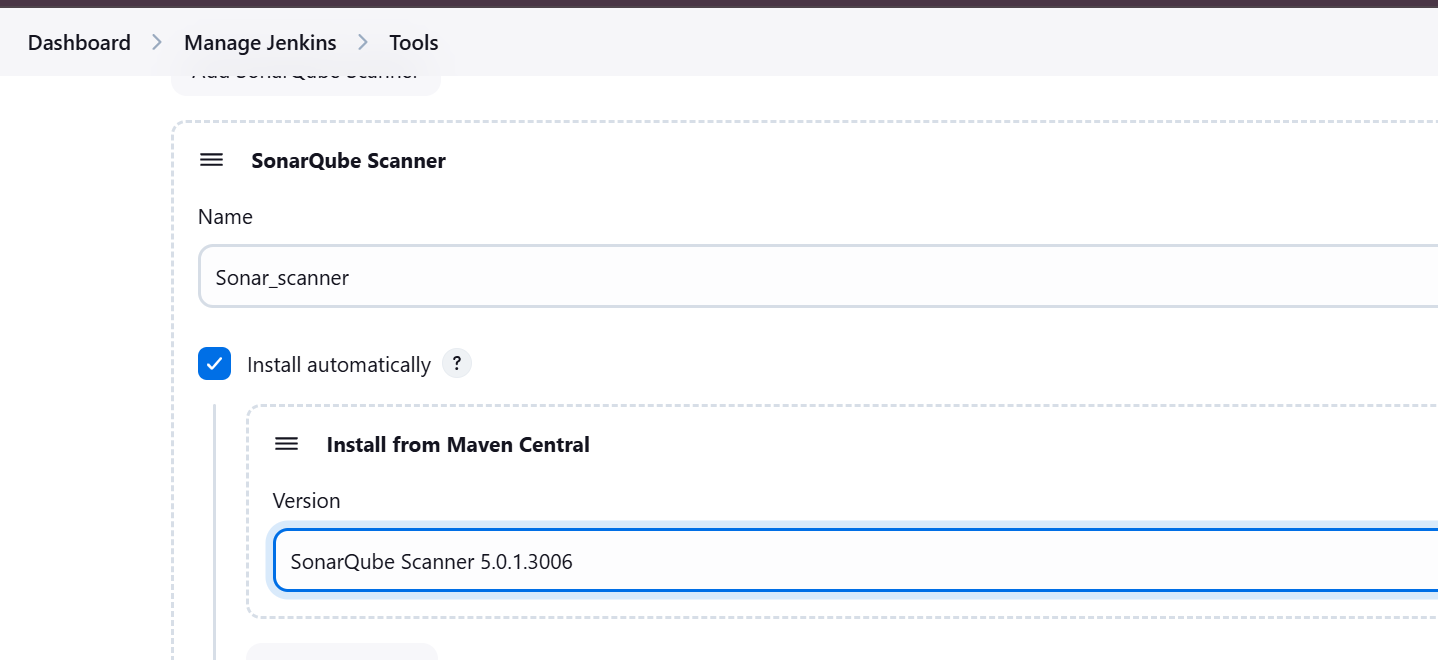
Dashboard>Manage Jenkins>system>Sonarqube Installation



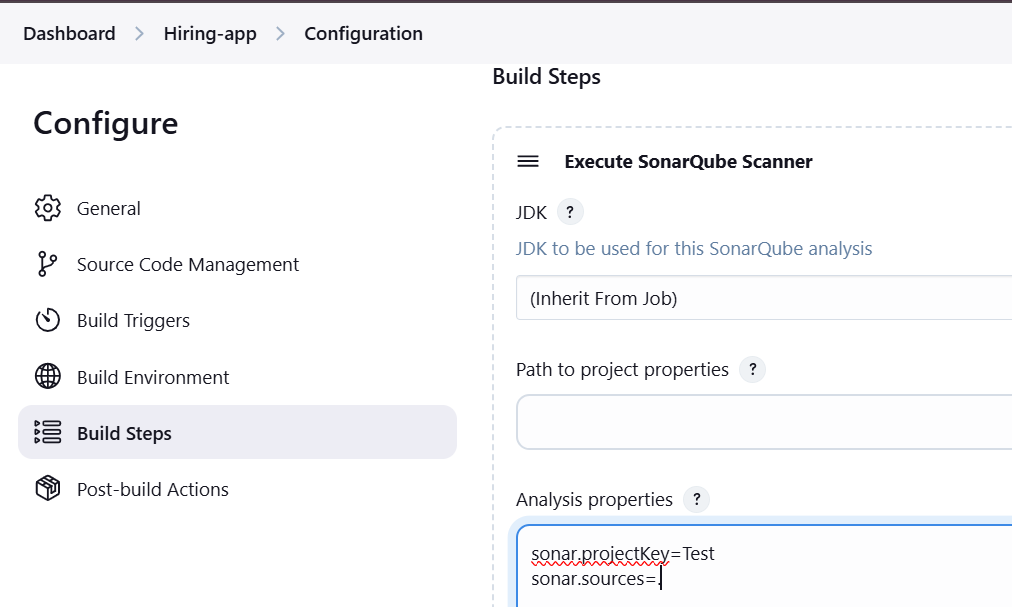
**G**o to Jenkins and download the Jenkins Scanner plugin



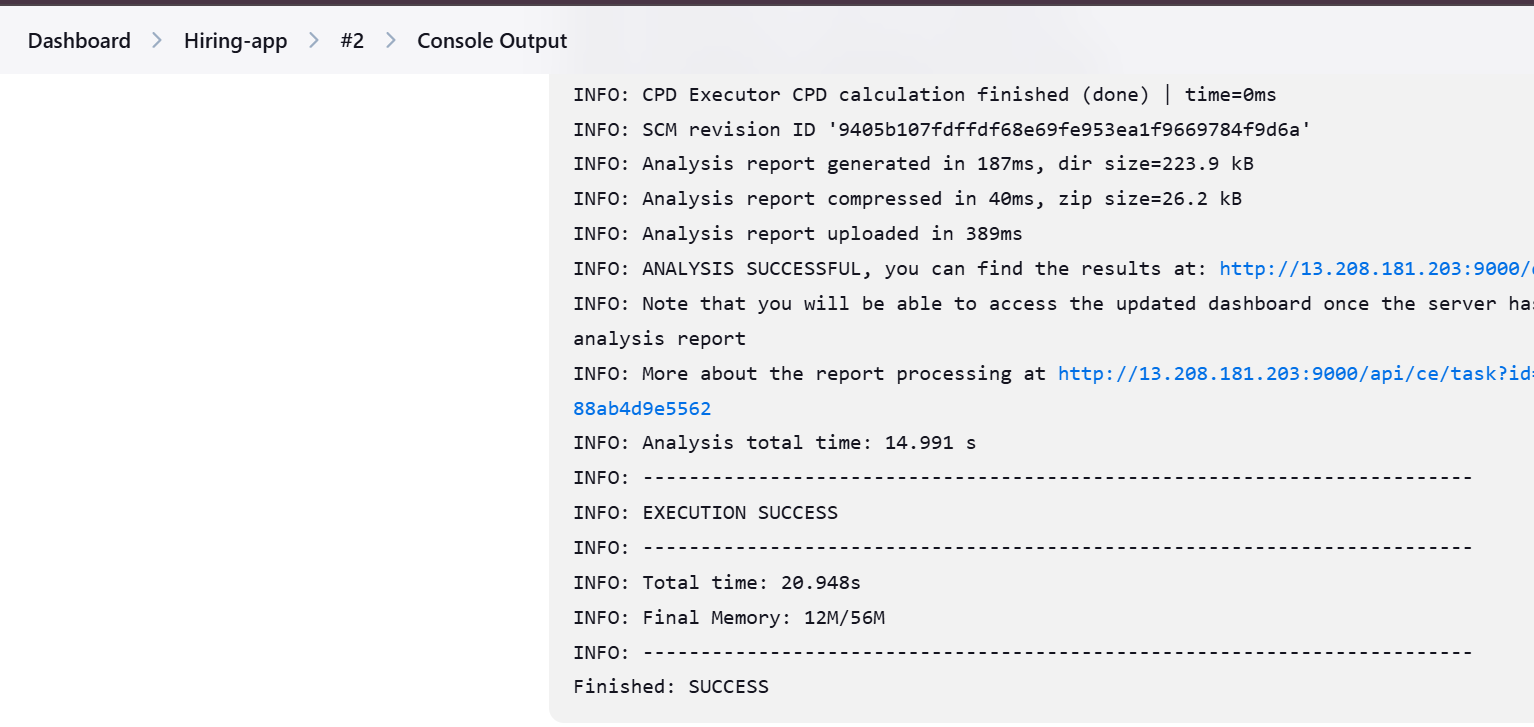
**Now go to the Dashboard>manage Jenkins>Tools**



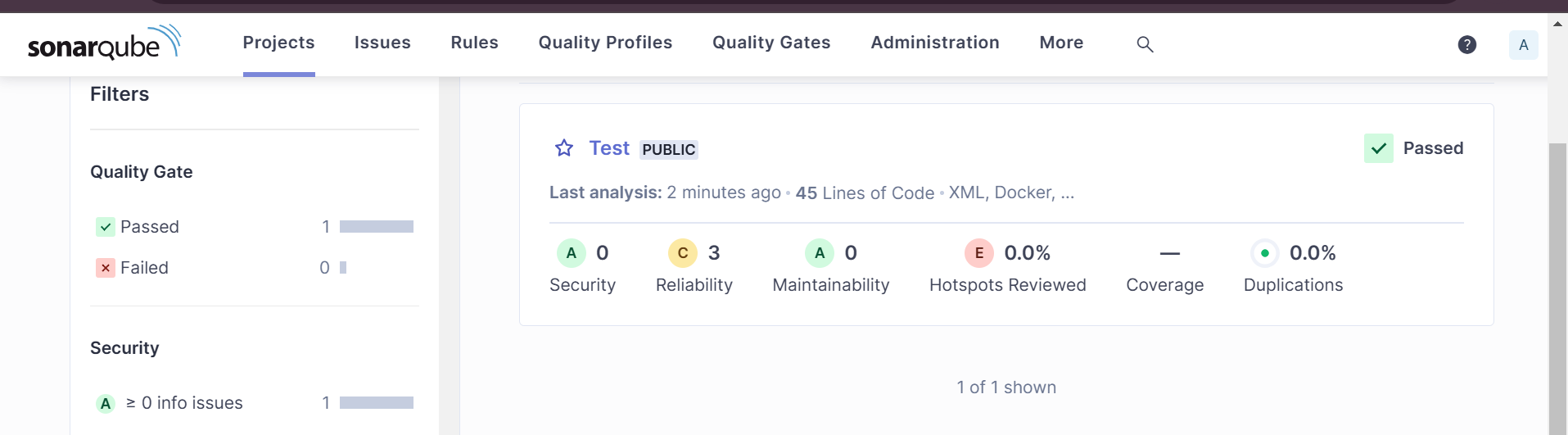
Now



Console output



In soanrqube server we can see the quality of the project

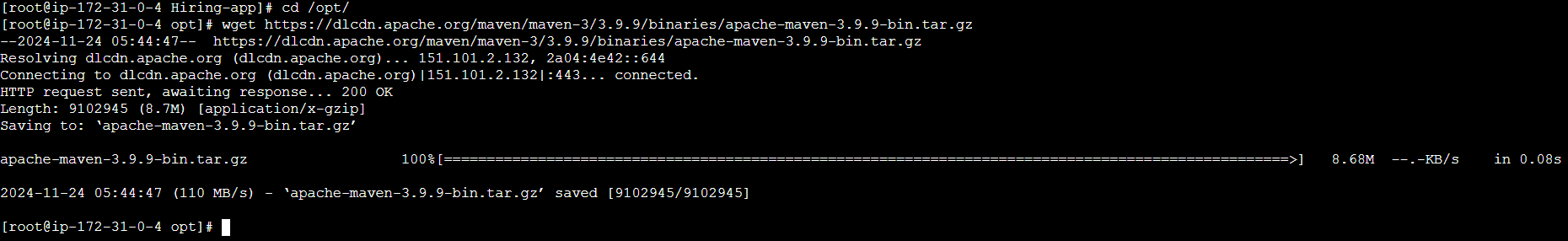


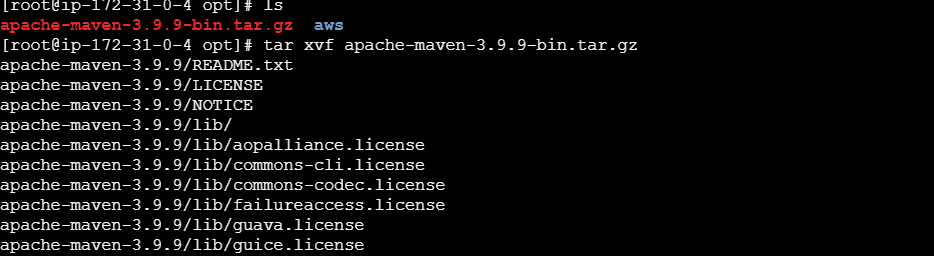
**3) Maven Compilation**

Go to the Jenkins sever

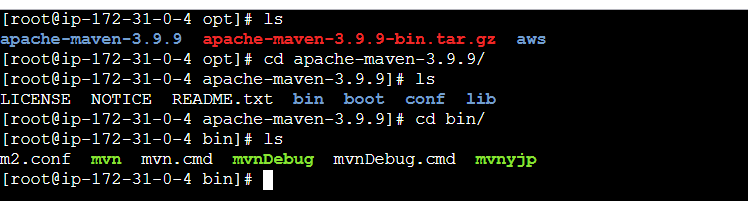
Switch to /opt

Download the maven using the link

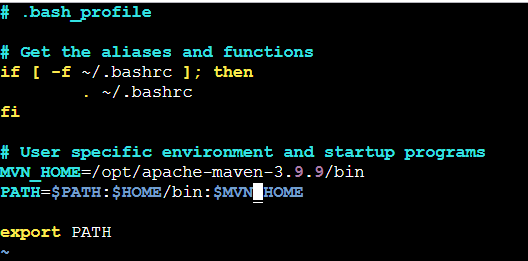
<https://dlcdn.apache.org/maven/maven-3/3.9.9/binaries/apache-maven-3.9.9-bin.tar.gz> 

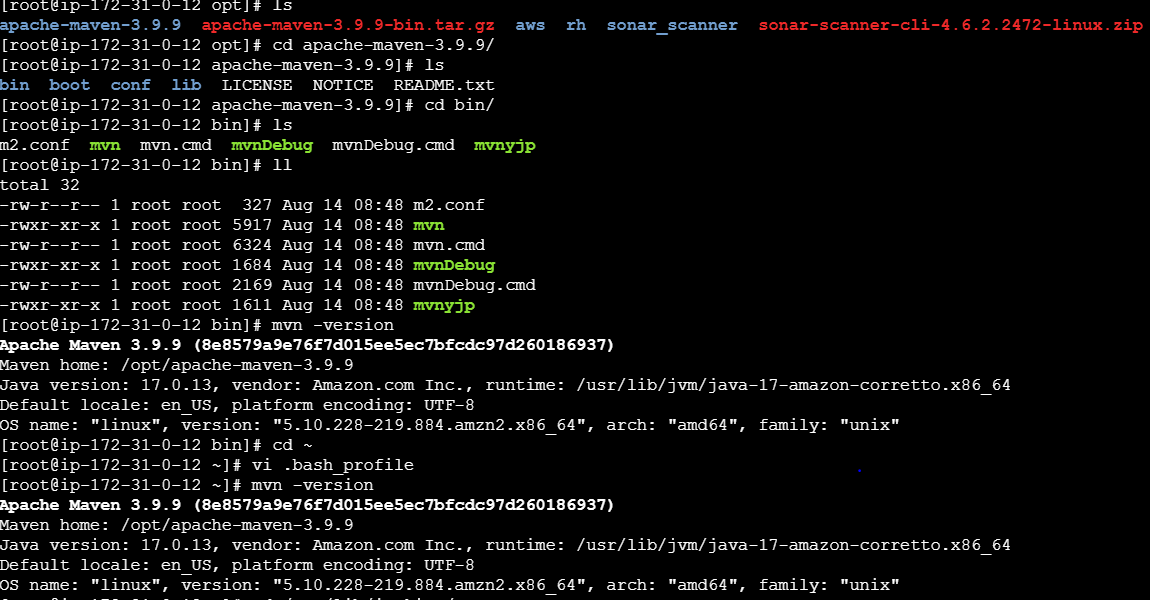
Extract the file 

We can see the extracted file

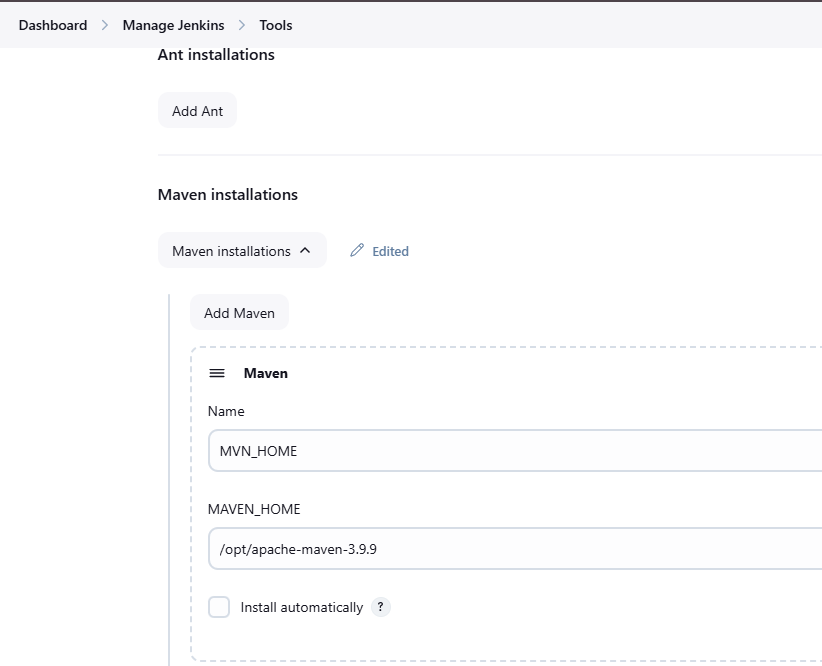


Declaring the global variable





Now,Go to Dashboard>Manage Jenkins>Tool>Maven Installations

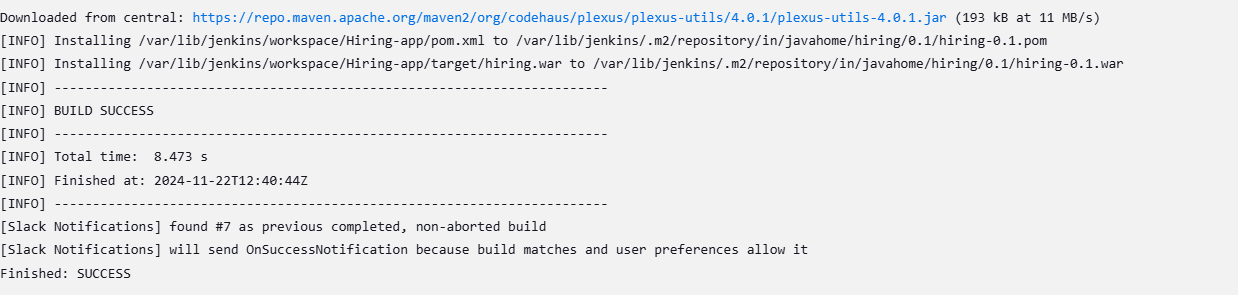


Go to DashBoard>Job>Build step>invoke top-level Maven target



Save the changes and execute the job

We can check that job status



In Jenkins server

Go to : cd /var/lib/Jenkins/workspace

We can see a target file is created

**4) Nexus Artifactory**

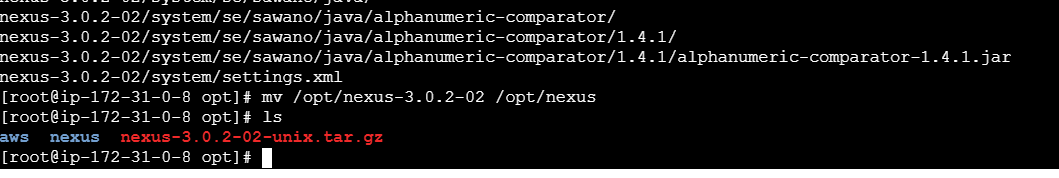
Download and setup nexus stable version

cd /opt

wget

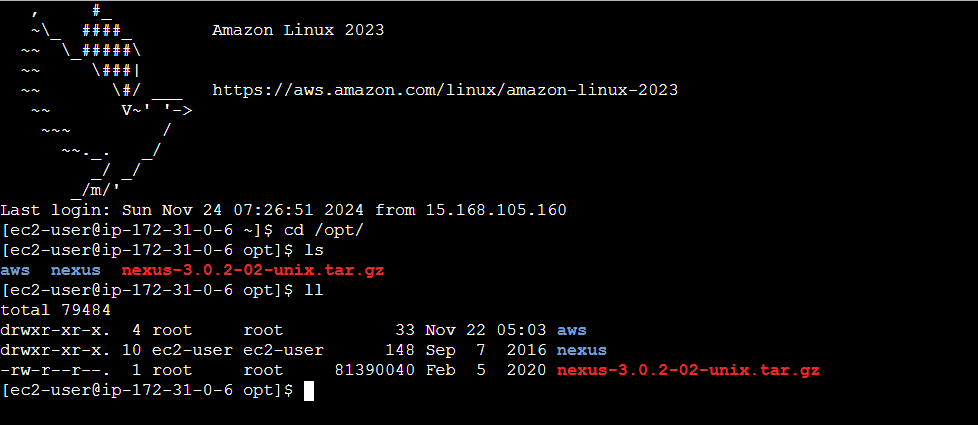
https://sonatype-download.global.ssl.fastly.net/nexus/3/nexus-3.0.2-02-unix.tar.gz

tar -zxvf nexus-3.0.2-02-unix.tar.gz

mv /opt/nexus-3.0.2-02 /opt/nexus

now[changing the permissions of nexus file]

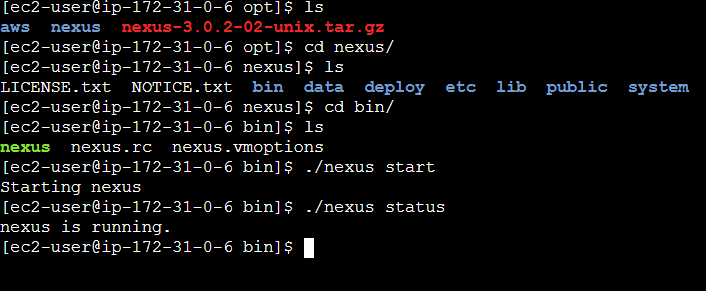
chown –R ec2-user:ec2-user nexus

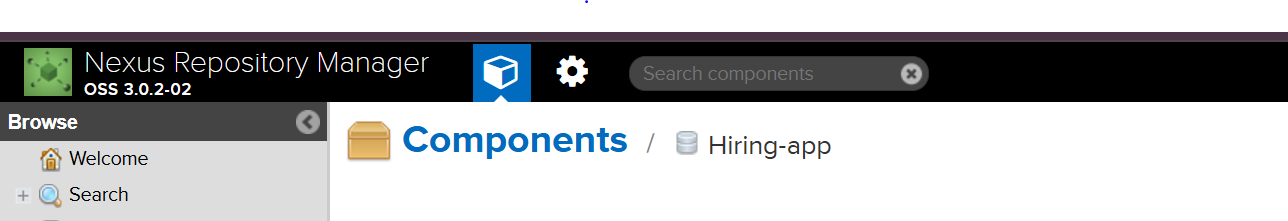


vi /opt/nexus/bin/nexus.rc

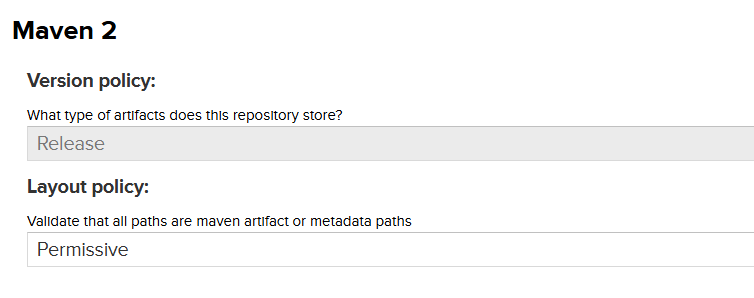
run\_as\_user="nexus" (file shold have only this line)

after,run the nexus

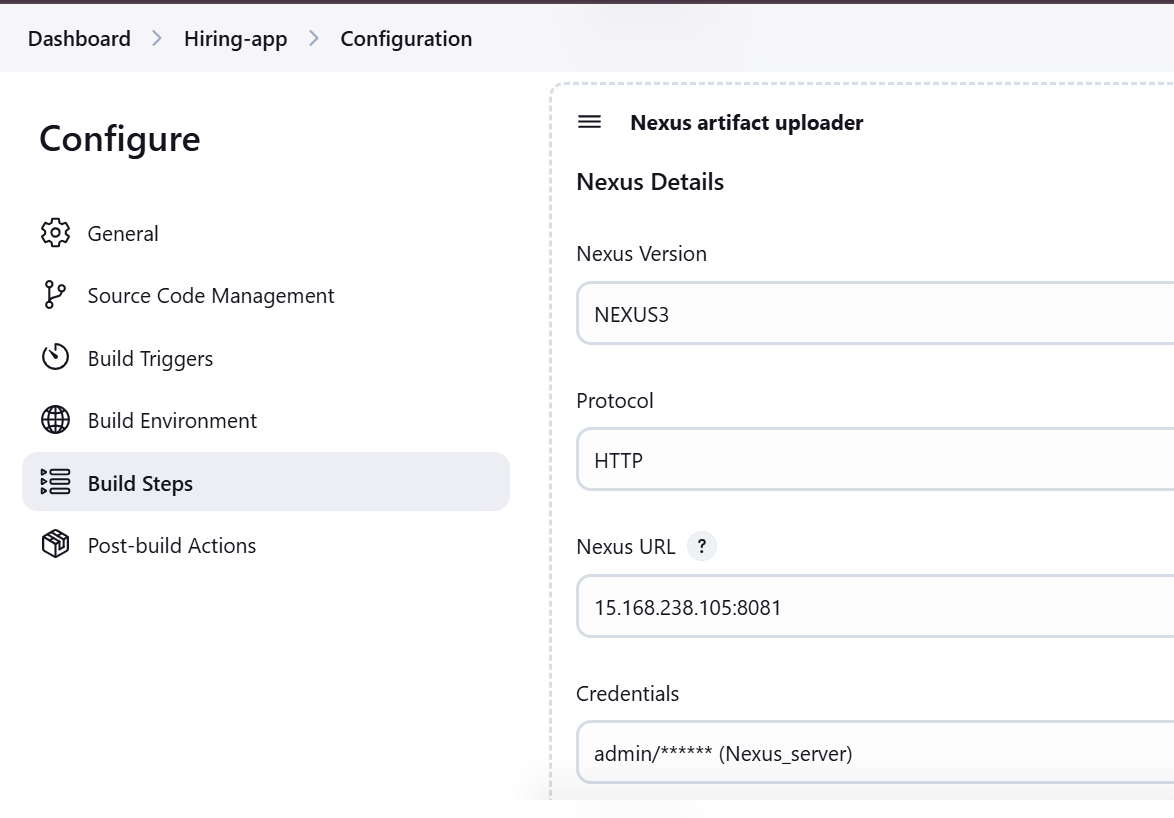


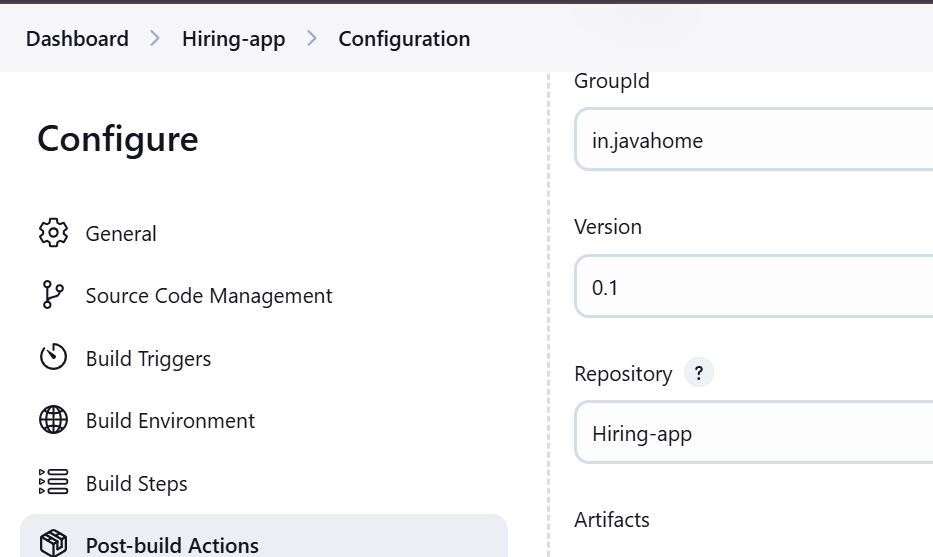


We need to crete a repository in the nexus

Go to settings>repositories>create 

Save



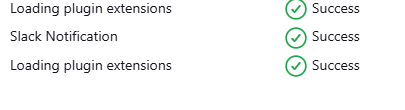
 

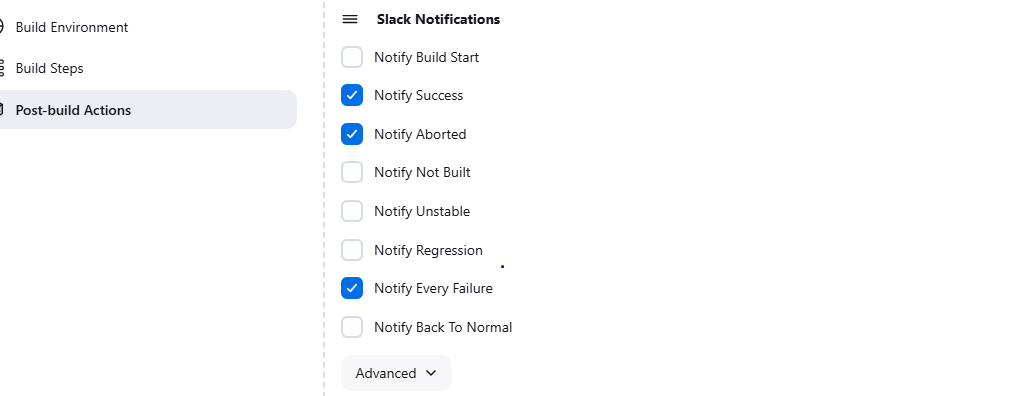
Execute the job

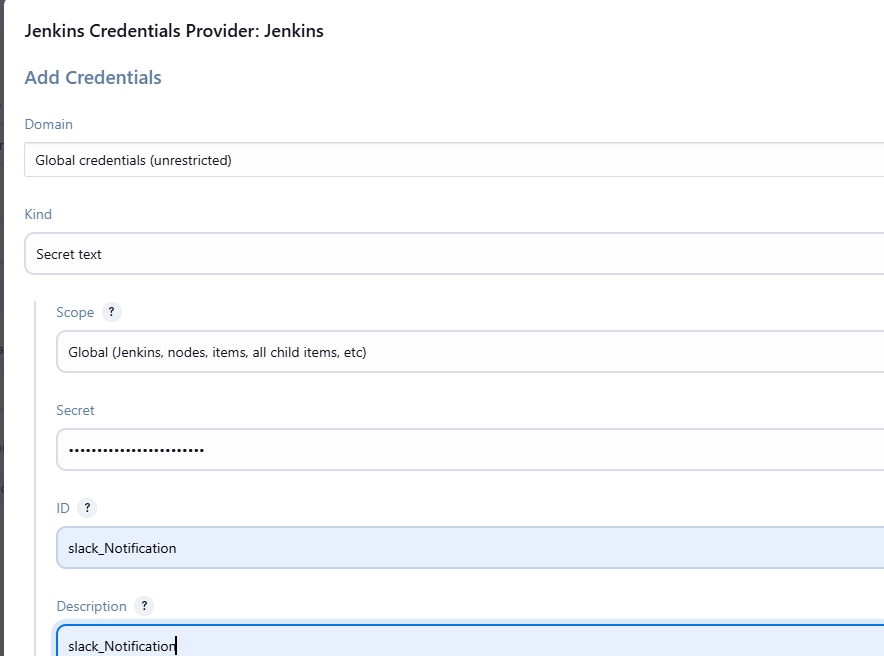


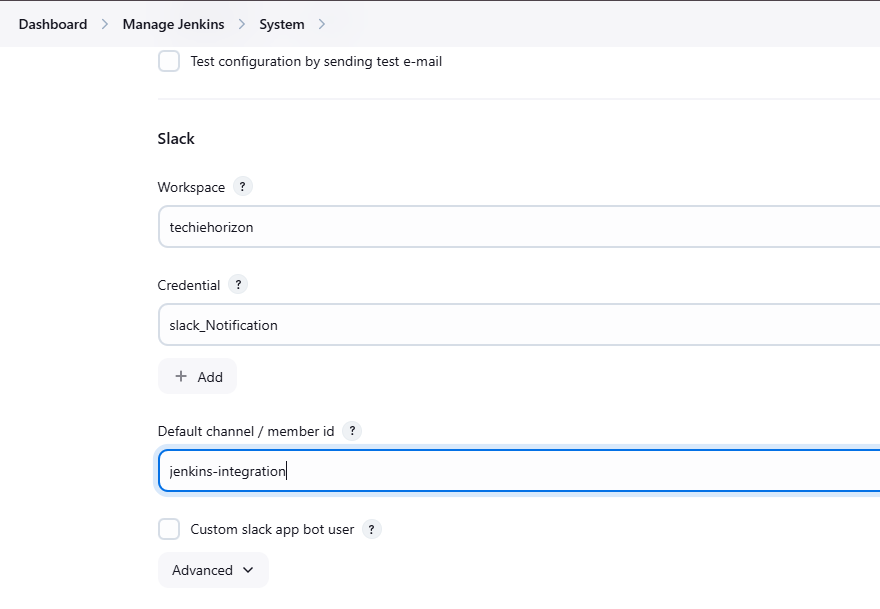
**5) Slack Notification**

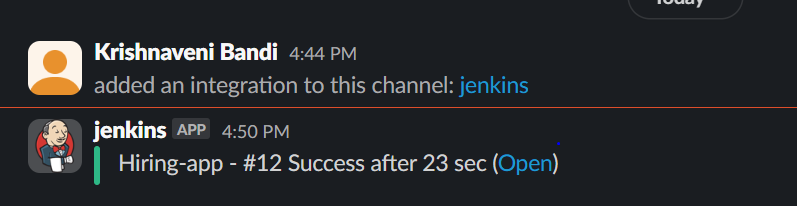
Gp to the dashboard>plugin>slackNotification

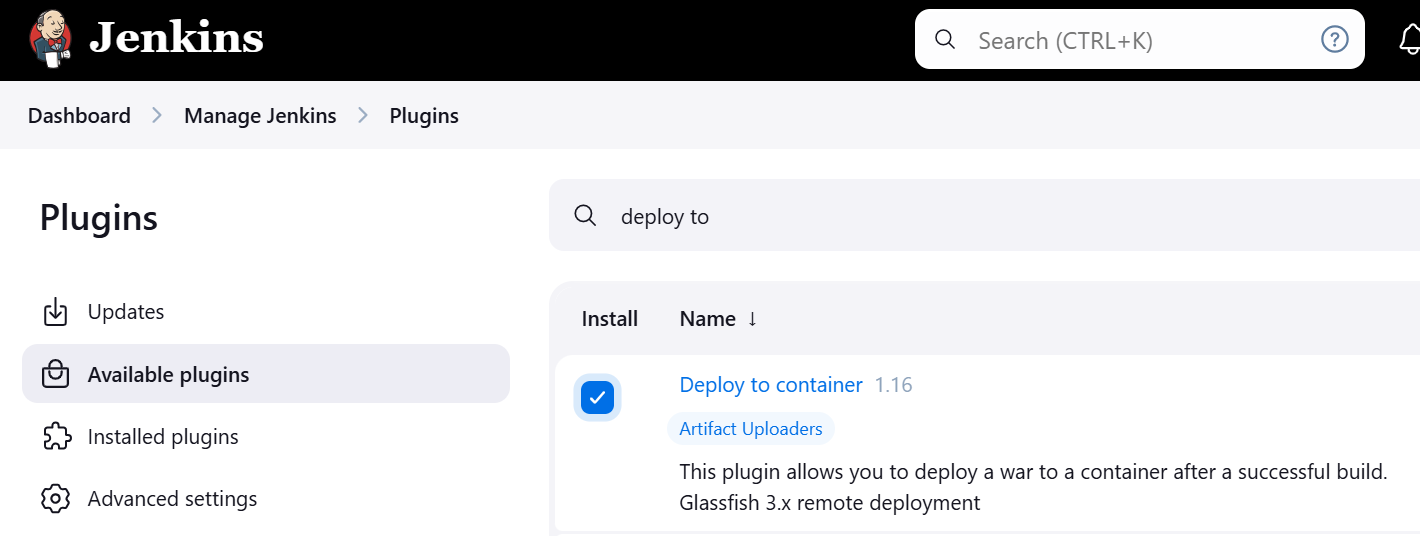
 Post build actions

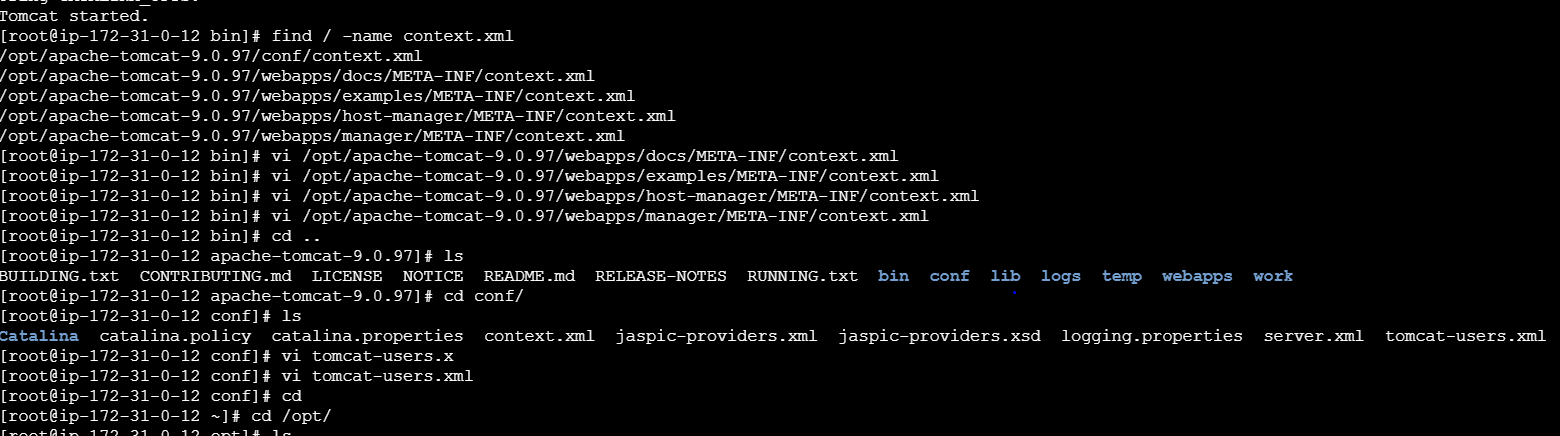








**6) Deploy On tomcat** 

find / -name context.xml 

Update users information in the tomcat-users.xml file

goto tomcat home directory and Add below users to conf/tomcat-user.xml file

<role rolename="manager-gui"/>

<role rolename="manager-script"/>

<role rolename="manager-jmx"/>

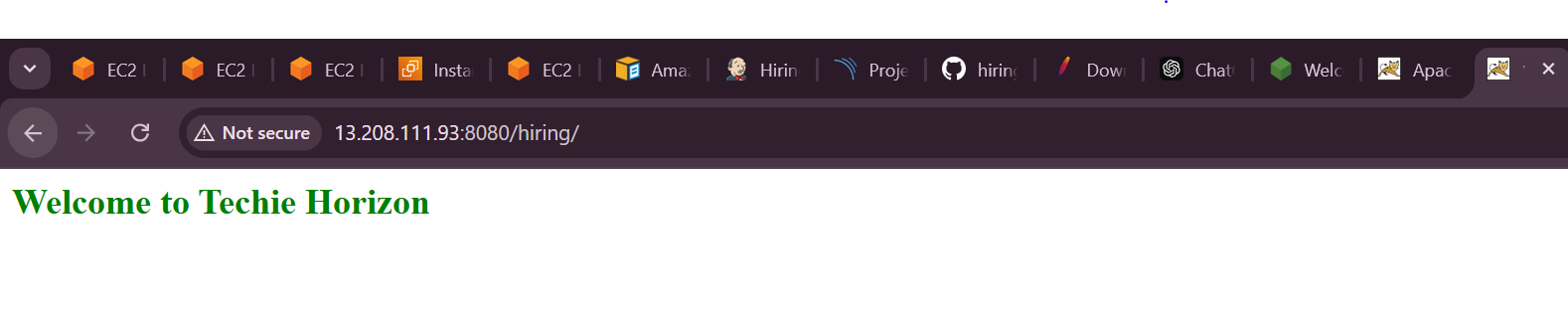
<role rolename="manager-status"/>

<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"/> 





**4) Write sample skeleton of pipelines.**

pipeline {

agent any // Run the pipeline on any available agent

environment {

// Define global variables, if any

MVN\_HOME = "/opt/apache-maven-3.9.9/bin"

}

stages {

stage('Checkout') {

steps {

echo 'Checking out code from SCM...'

checkout scm // Checkout the source code from the configured repository

}

}

stage('Build') {

steps {

echo 'Building the project...'

sh "${MVN\_HOME}/mvn clean package -DskipTests" // Build the project and create the artifact

}

}

stage('Test') {

steps {

echo 'Running tests...'

sh "${MVN\_HOME}/mvn test" // Run unit tests

}

}

stage('Code Quality') {

steps {

echo 'Performing code quality checks...'

// Add code quality tools like SonarQube

sh "sonar-scanner -Dsonar.projectKey=your\_project\_key -Dsonar.host.url=http://your-sonarqube-server"

}

}

stage('Artifact Upload') {

steps {

echo 'Uploading artifact to Nexus...'

sh "${MVN\_HOME}/mvn deploy" // Deploy artifact to Nexus or similar artifact repository

}

}

stage('Deploy to Tomcat') {

steps {

echo 'Deploying the application to Tomcat...'

sh """

scp target/hiring.war user@tomcat-server:/path/to/tomcat/webapps/

ssh user@tomcat-server 'systemctl restart tomcat'

"""

}

}

}

post {

always {

echo 'Pipeline execution completed.'

}

success {

echo 'Pipeline executed successfully!'

}

failure {

echo 'Pipeline failed. Investigate the issues.'

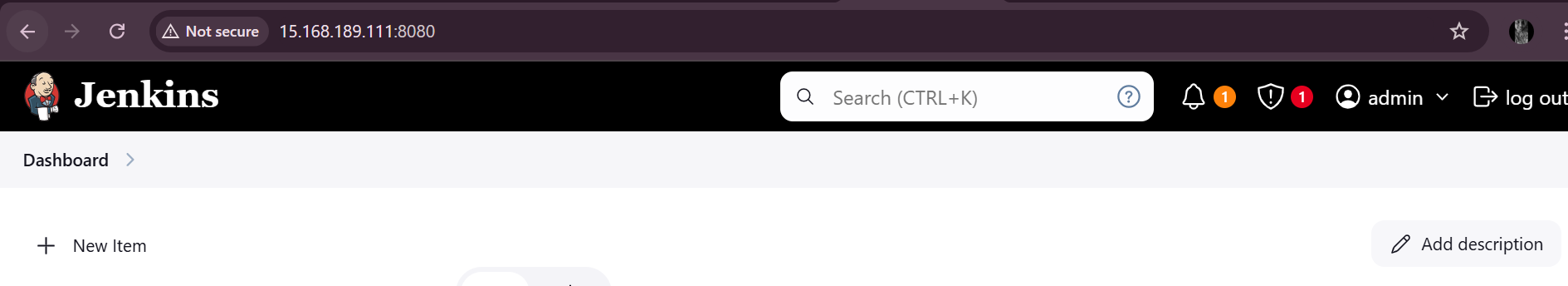
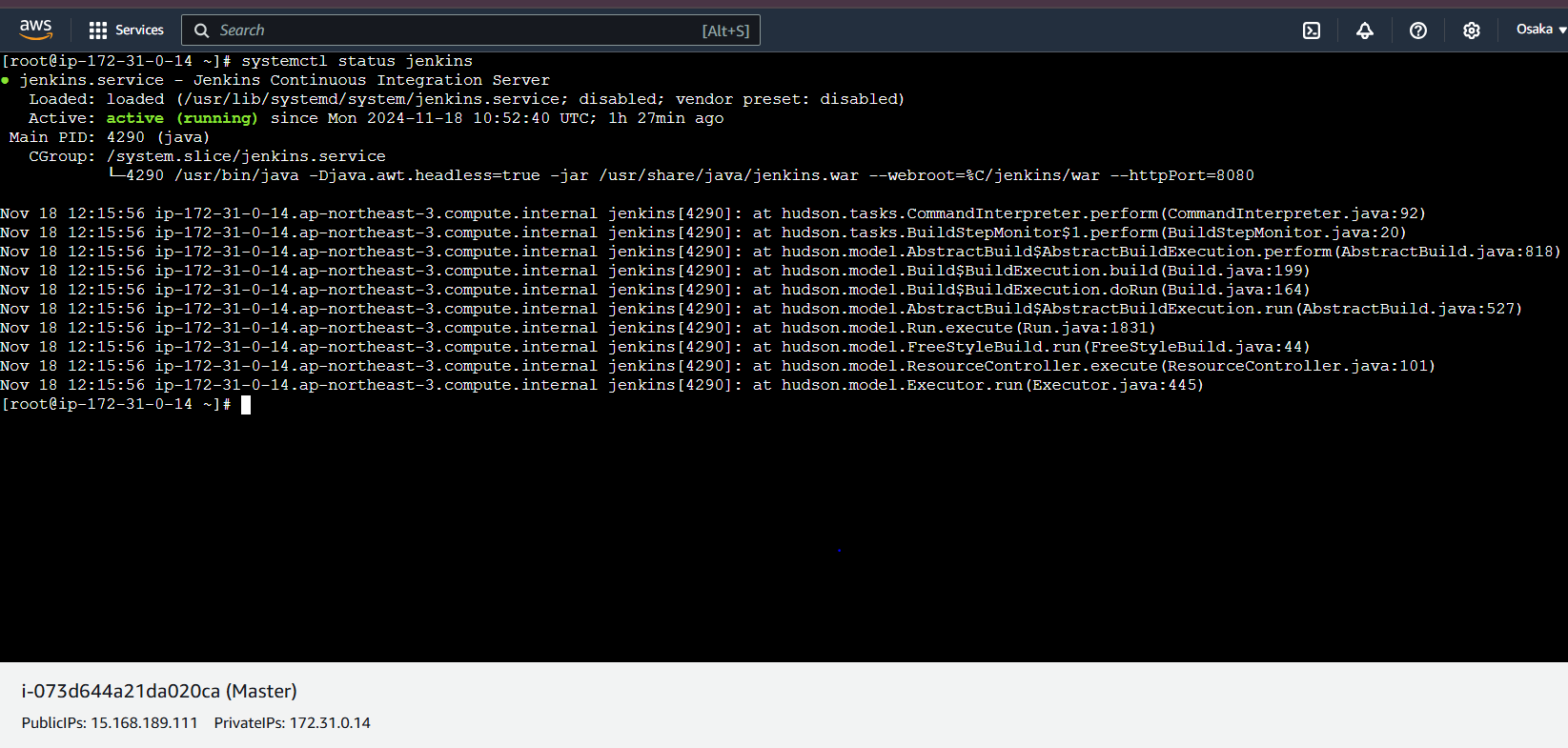
}

}

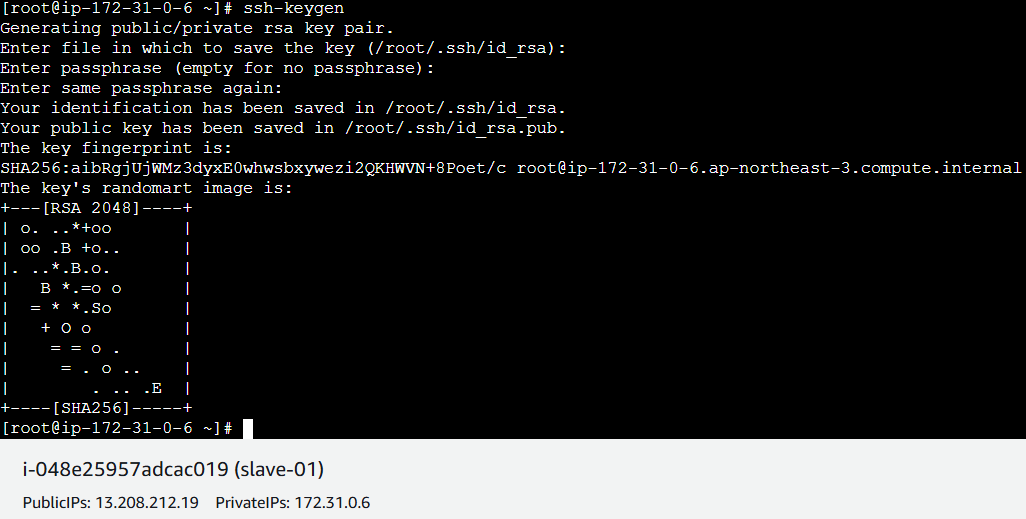
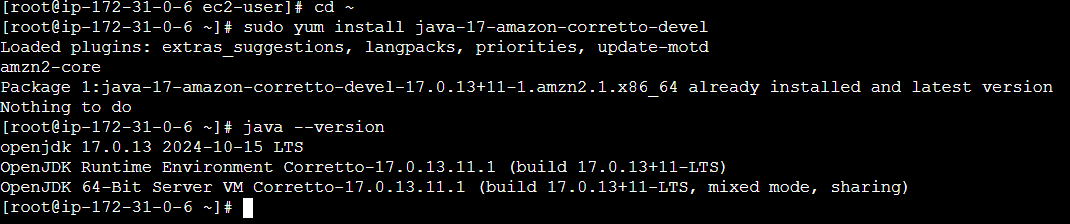
}

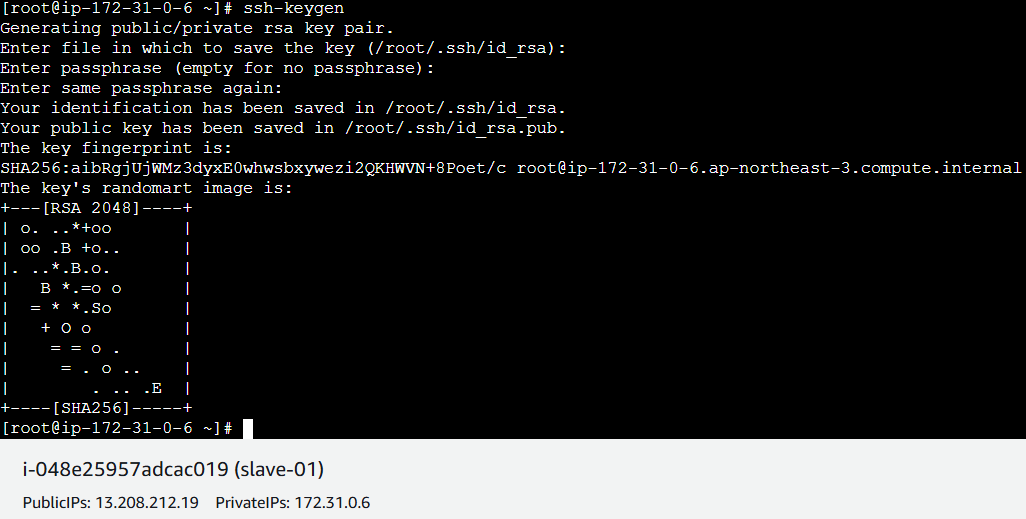
**6) Setup one slave machine for jenkins**.

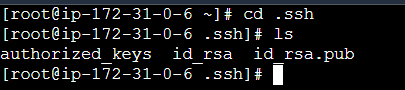


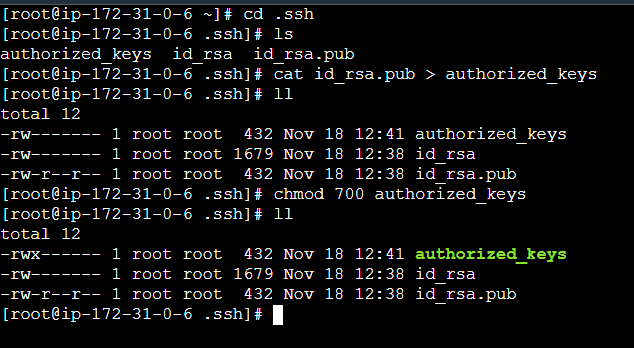


Creating a slave[with the java package

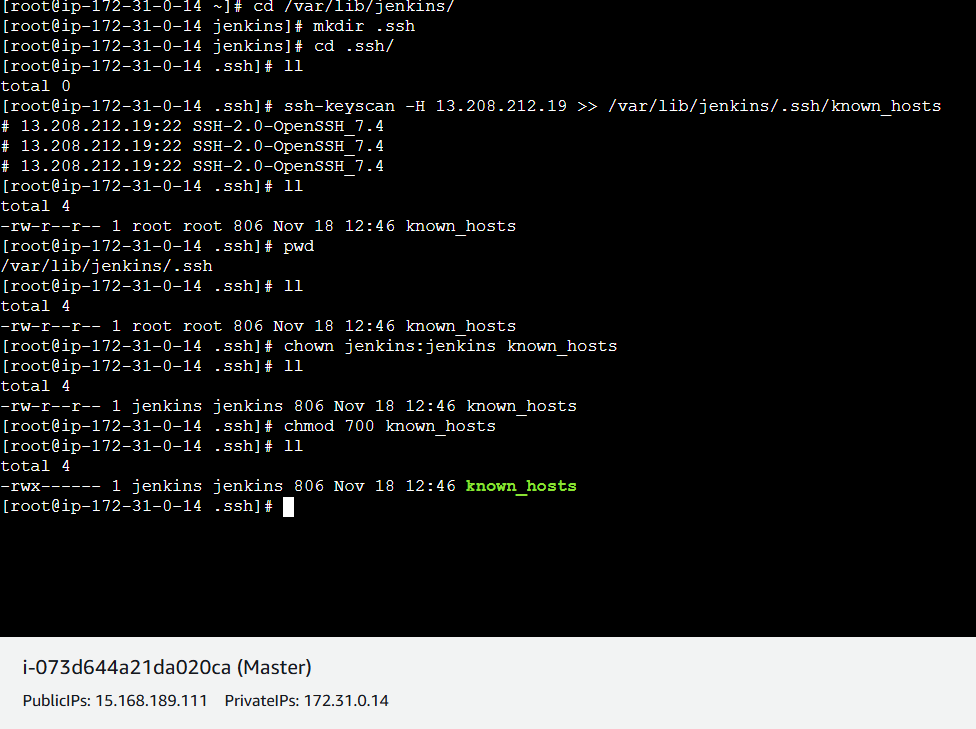


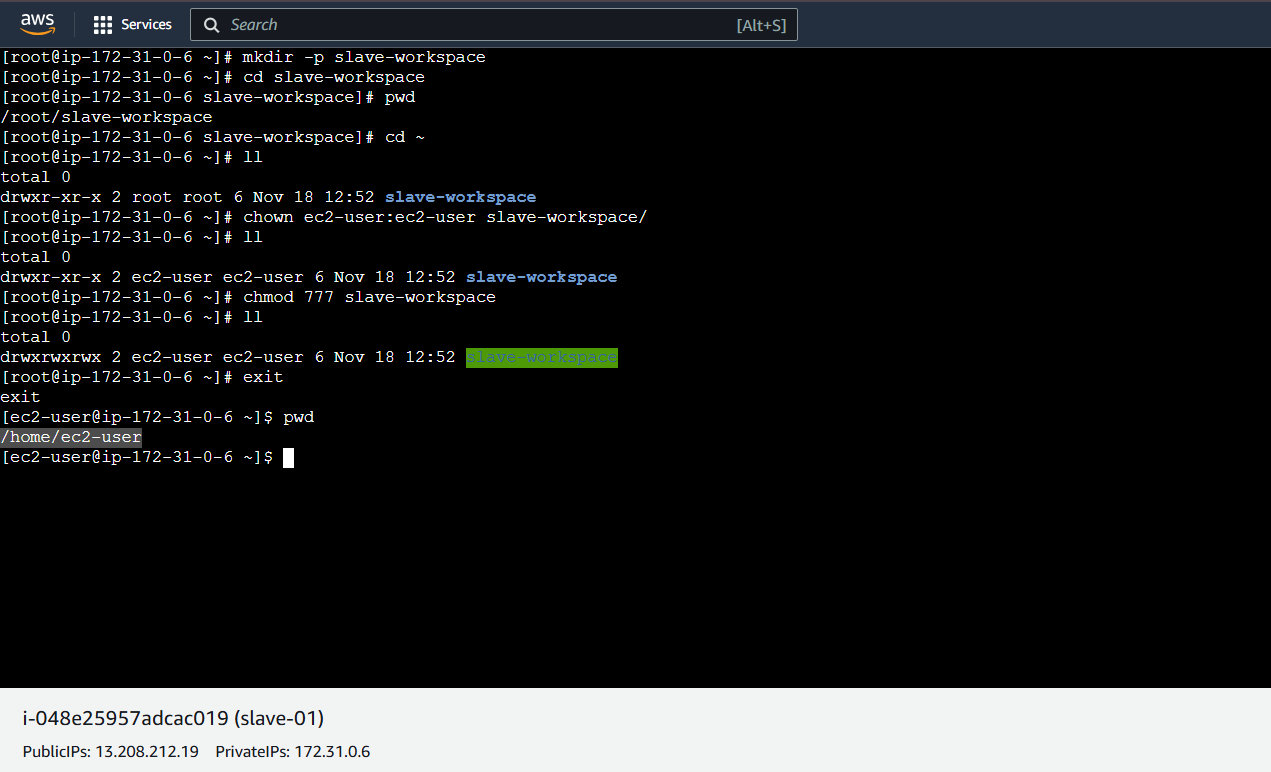
Creating passswordless authentication

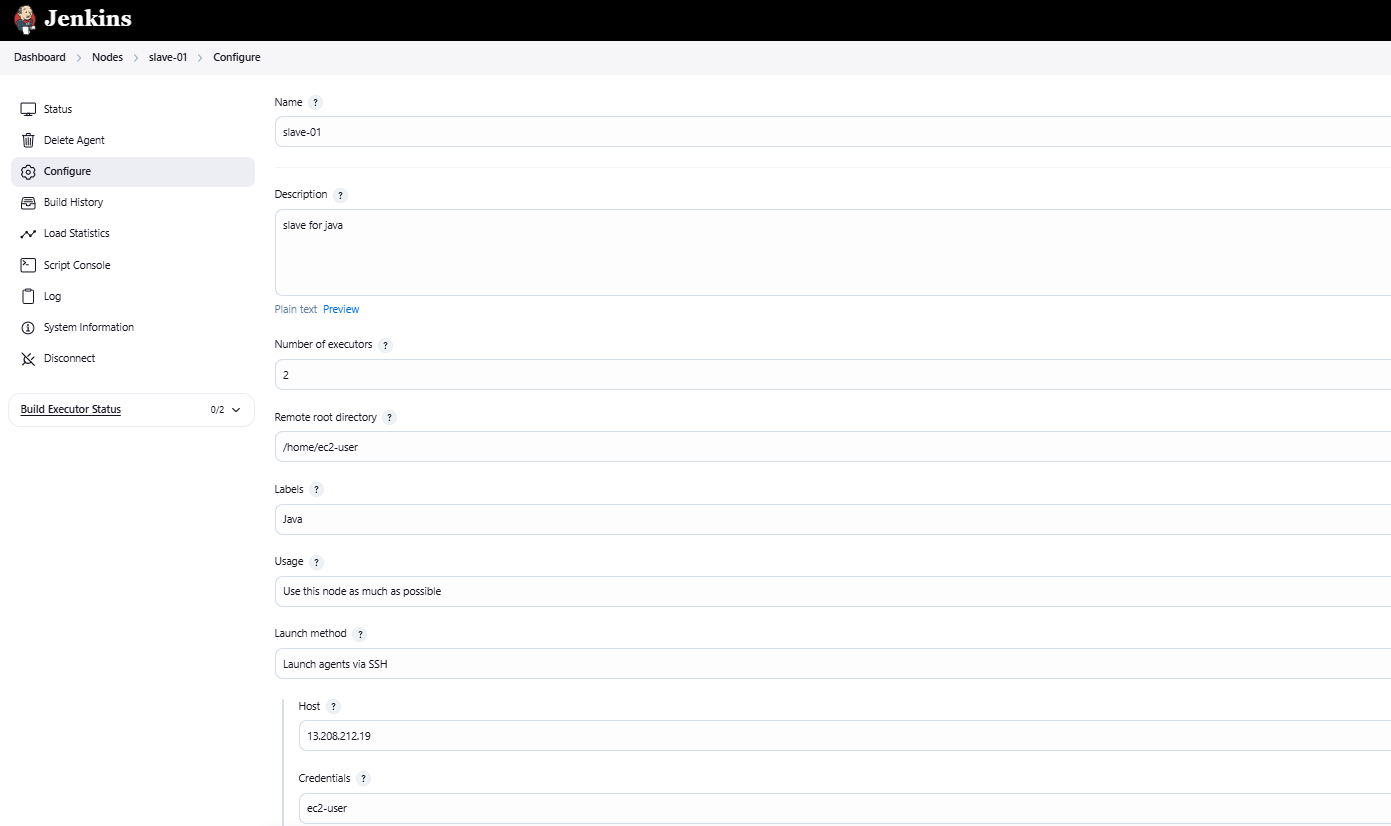




Now goto Master







Console output

