**Jenkins-04**

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

[**https://github.com/betawins/hiring-app.git**](https://github.com/betawins/hiring-app.git)

***Stages:***

***1) Git Clone***

***2) Sonarqube Integration***

***3) Maven Compilation***

***4) Nexus Artifactory***

***5) Slack Notification***

***6) Deploy On tomcat***

1. **Git Clone**

Jenkins Freestyle Job Configuration

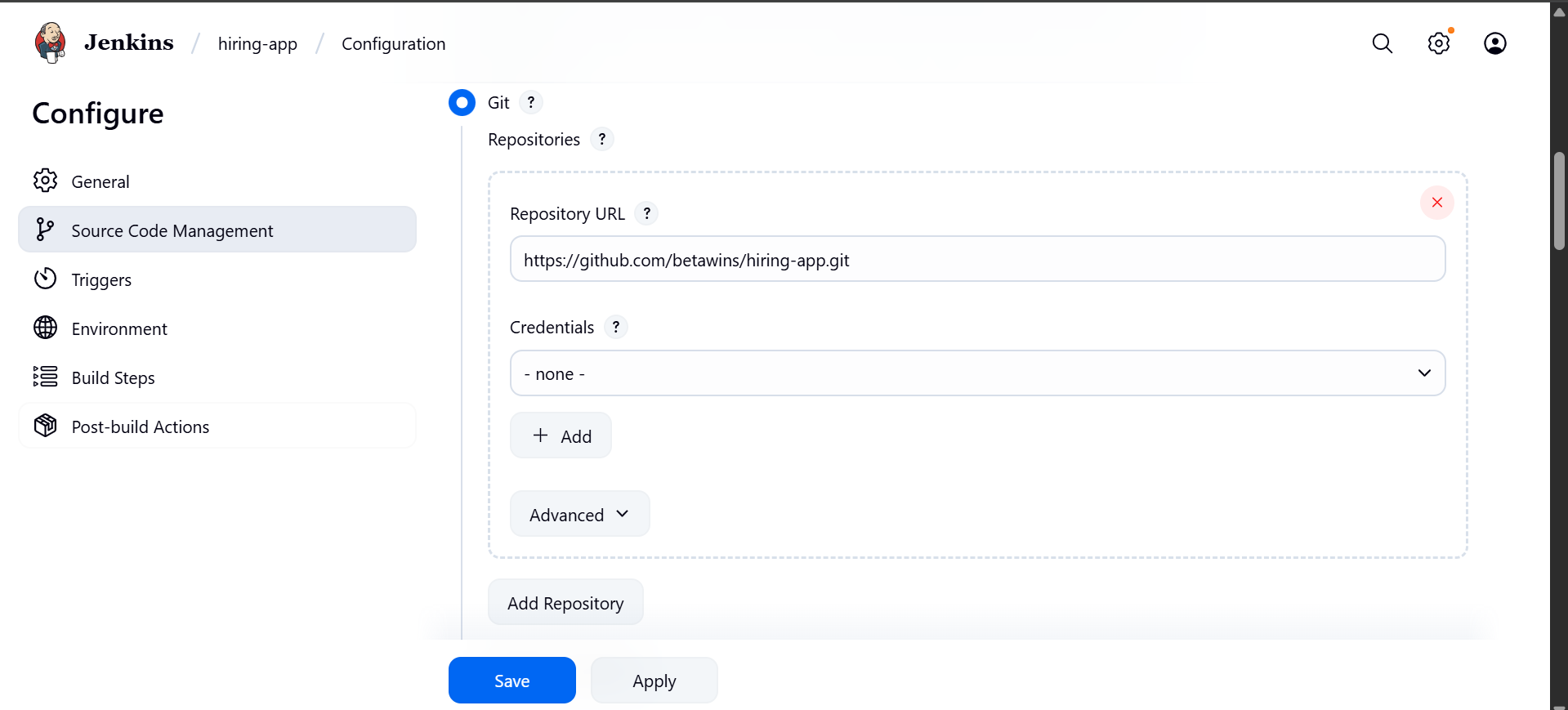
**Create Job**

* Go to Jenkins → New Item → select **Freestyle Project** → name it hiring-app

**Git Clone**

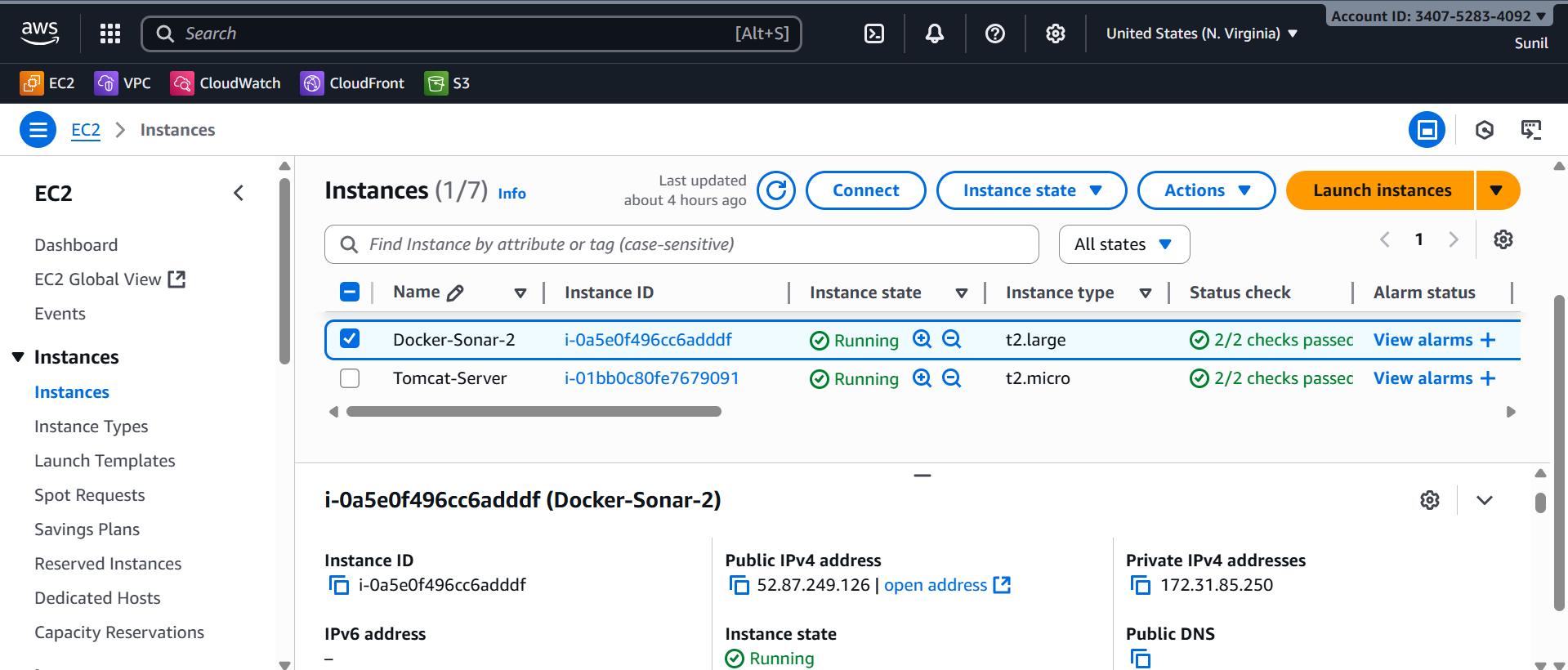
* In job configuration → Source Code Management → select **Git**.
* Repository URL:

<https://github.com/betawins/hiring-app.git>

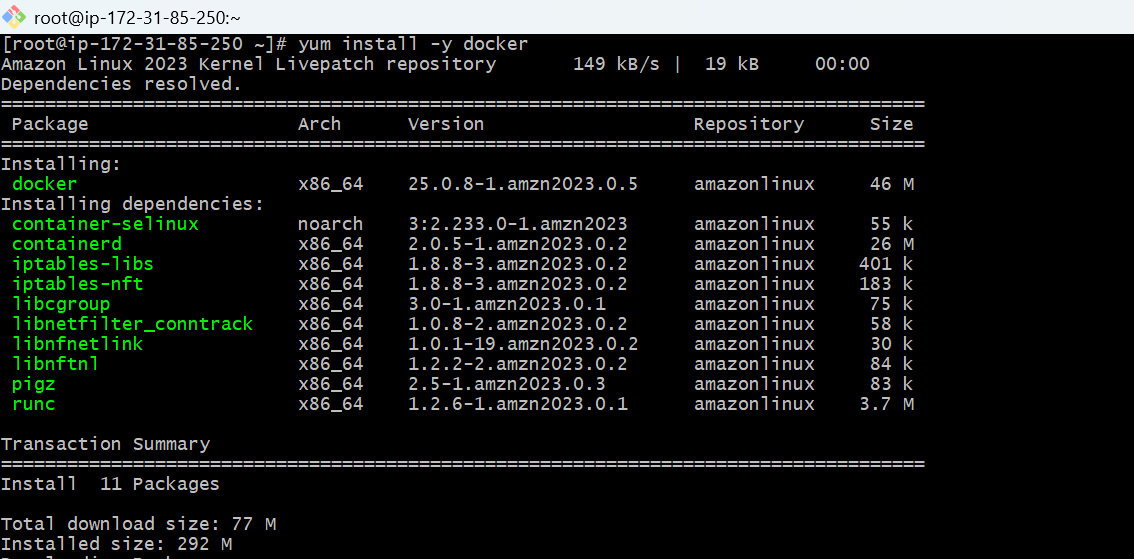


**2.** **SonarQube Integration --using Docker**

**Create a sonarqube-ec2 take t2.large and installed docker and pull image of sonarqube.**

****

**yum -y install docker**

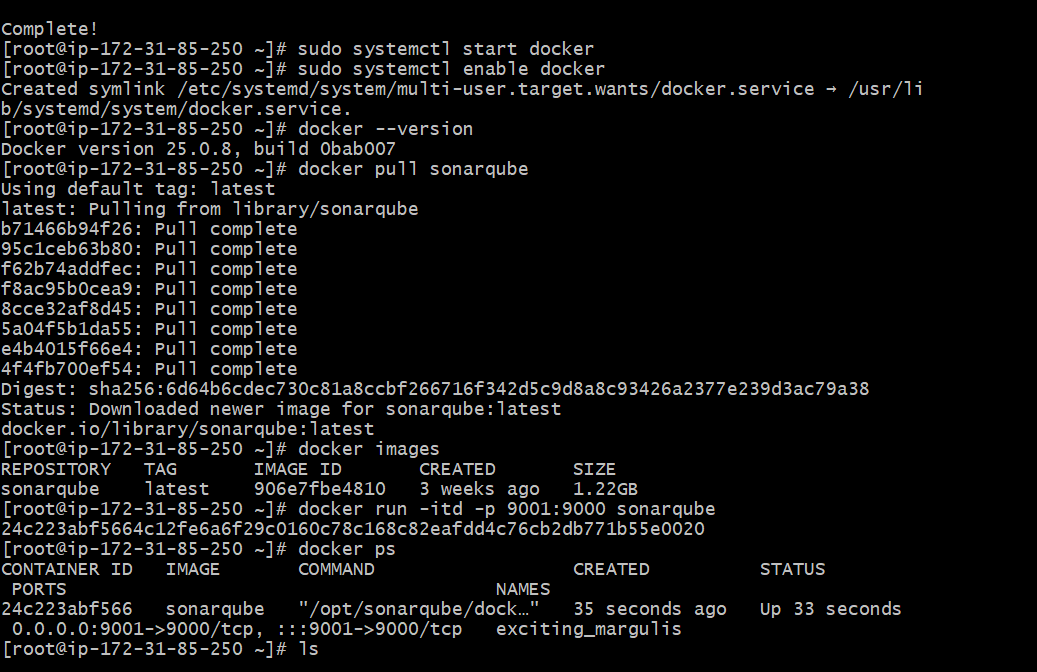


**systemctl start docker**

**docker pull sonarqube**

**docker run -d --name sonarqube -p 9001:9000 sonarqube:lts**

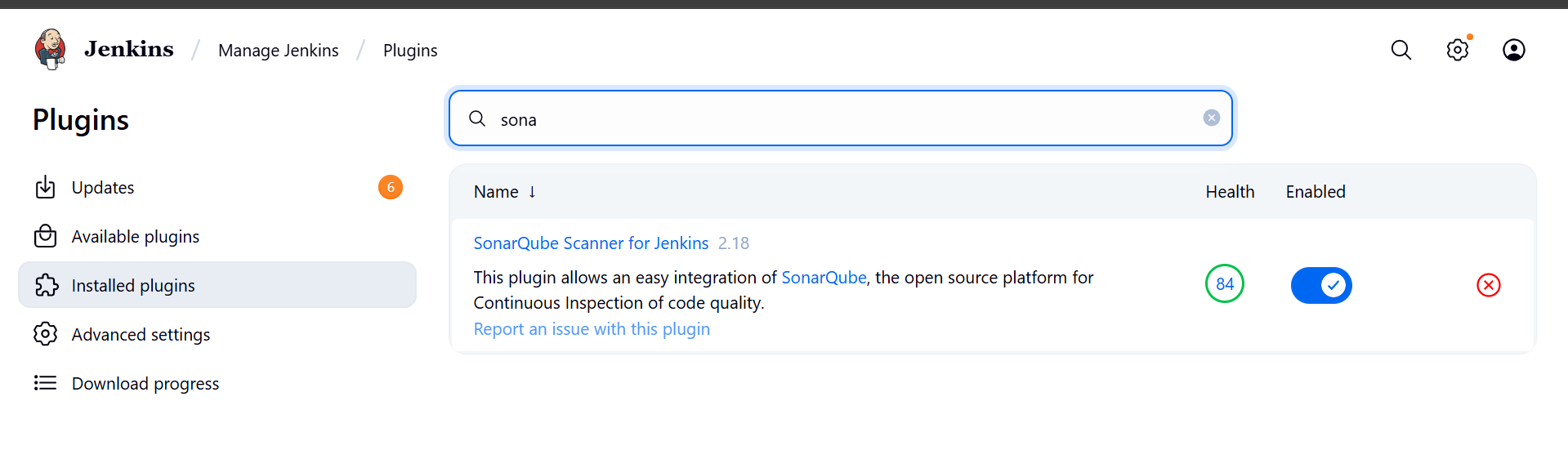
**Docker ps**



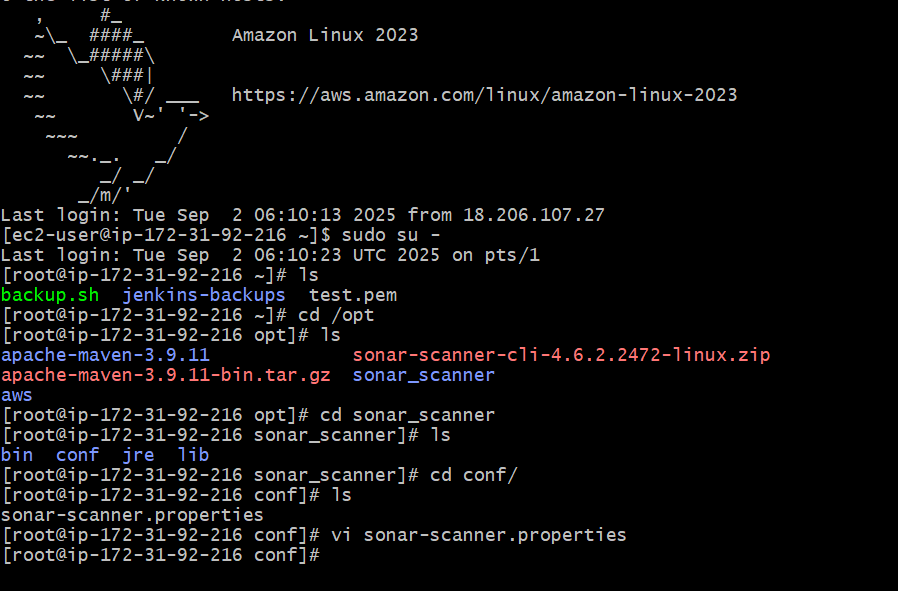
Install sonrQube Scanner Plugin in Jenkins

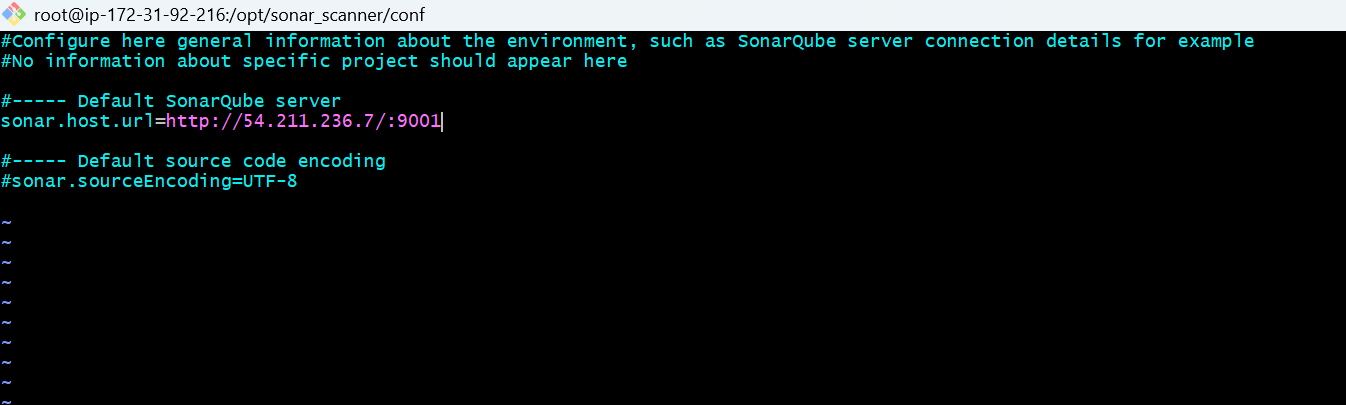
In Jenkins Dashboard:

* Go to **Manage Jenkins → Plugins → Available Plugins**.
* Search & install:
* **SonarQube Scanner for Jenkins**

****

In Jenkins Master edit the sonar properties





: Configure SonarQube Server

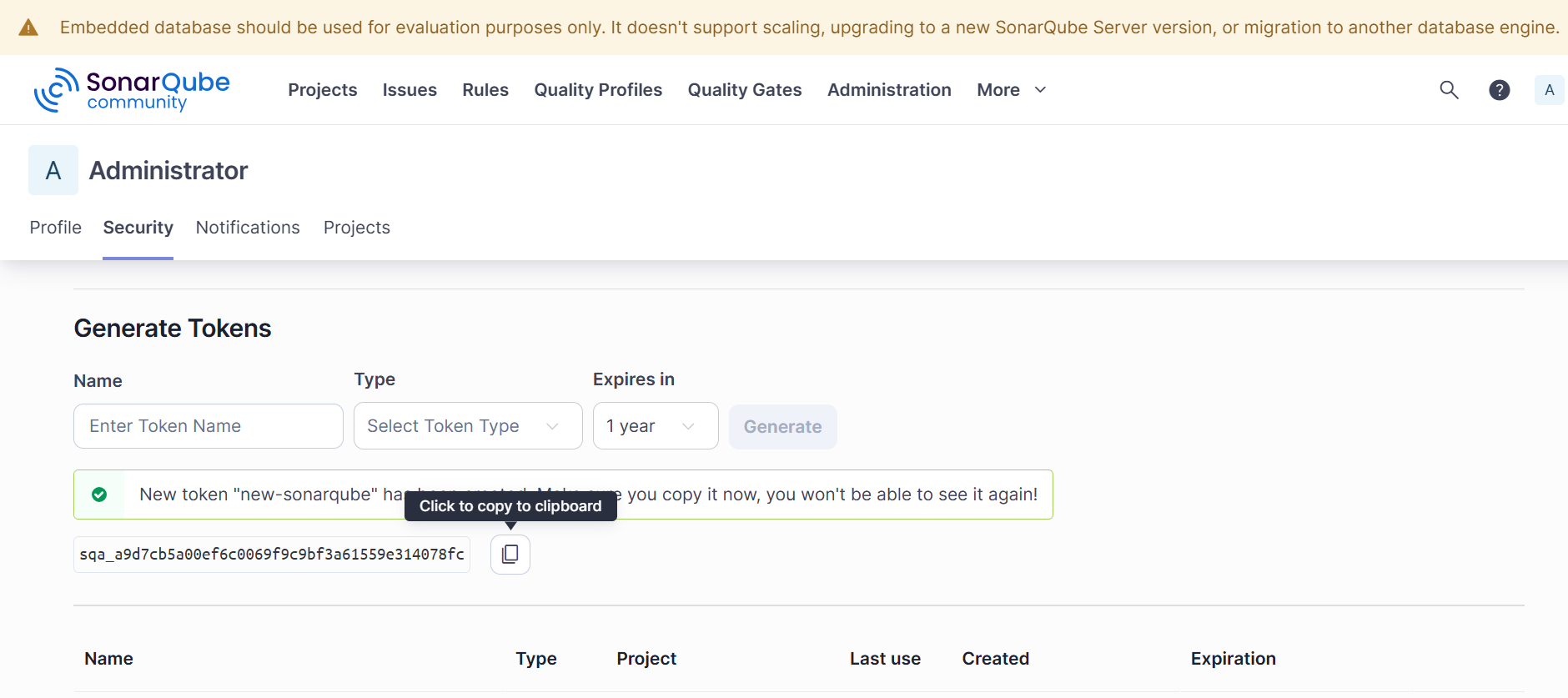
* Go to **Manage Jenkins → System**
* Scroll to **SonarQube servers** → Click **Add SonarQube**.



http://sonarqube-ip:9001

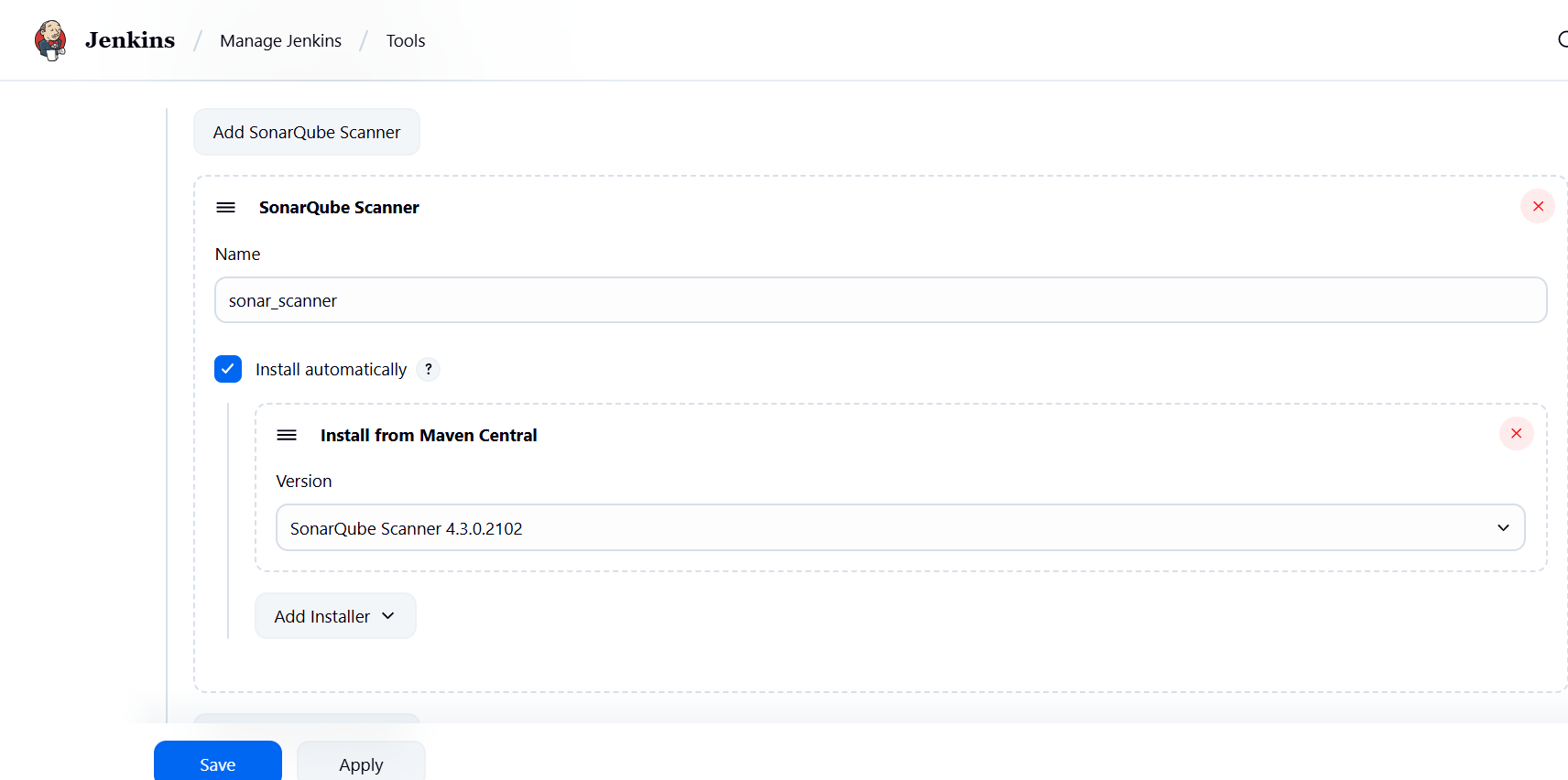
**Server authentication token**:

* Login to SonarQube UI → My Account → Security → Generate Token.
* Add it as a **Secret Text Credential** in Jenkins.



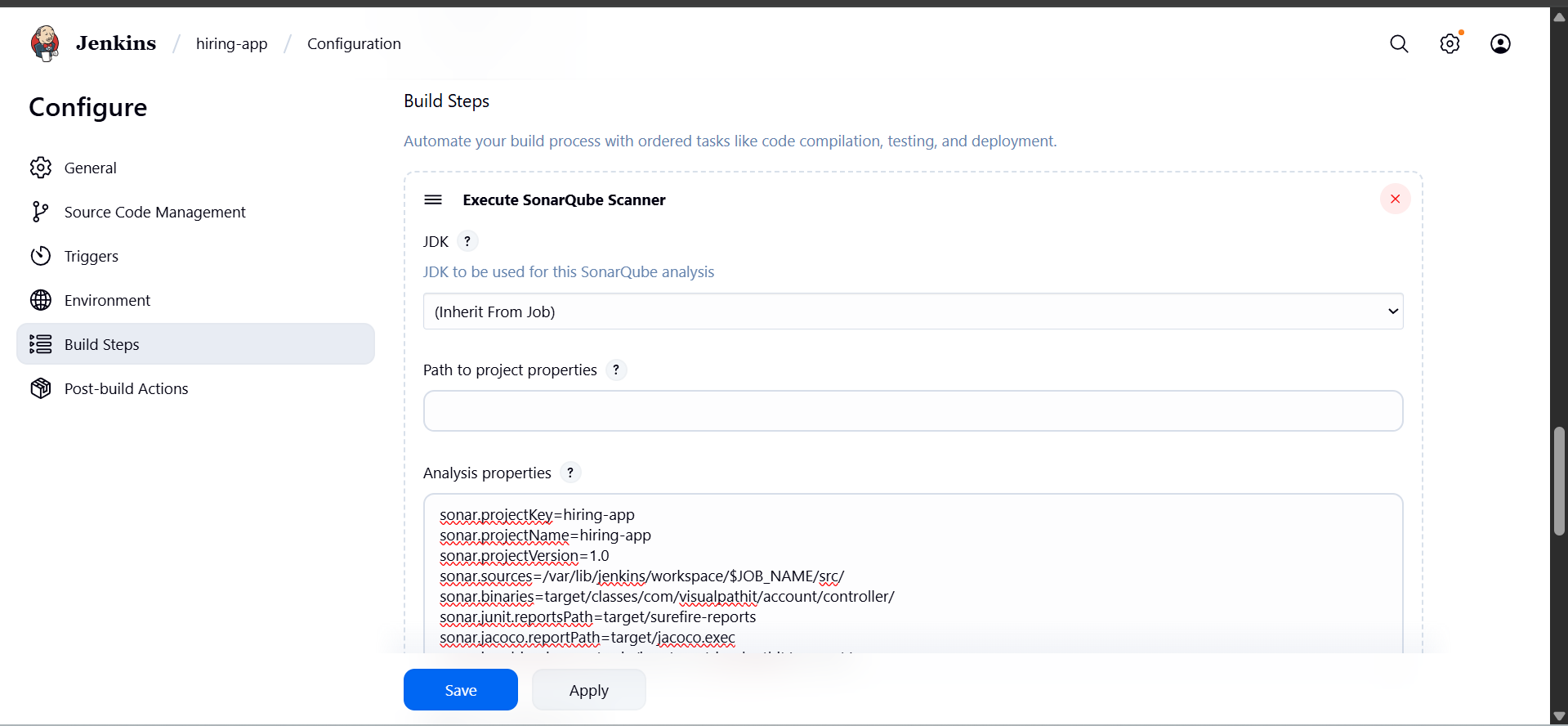
### Configure SonarQube Scanner

* Go to **Manage Jenkins → Tools**
* Under **SonarQube Scanner installations** → Add → Name it sonar­\_scanner.

****

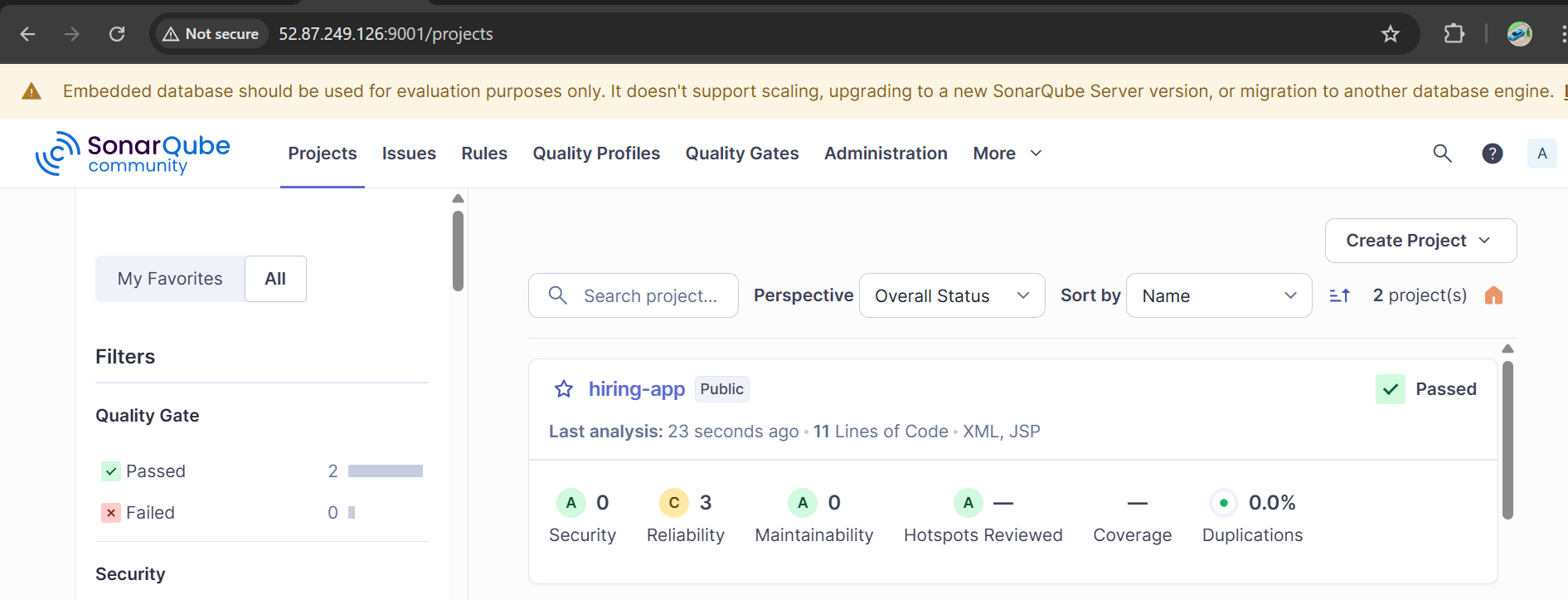
### ****SonarQube Integration****

* In Jenkins: **Manage Jenkins → Configure System** → set SonarQube server details.
* In build step: Add **Execute SonarQube Scanner** with properties:

****

* This will analyze code quality and send results to SonarQube container.
* Now start build

**And check the sonarqube for code quality**

****

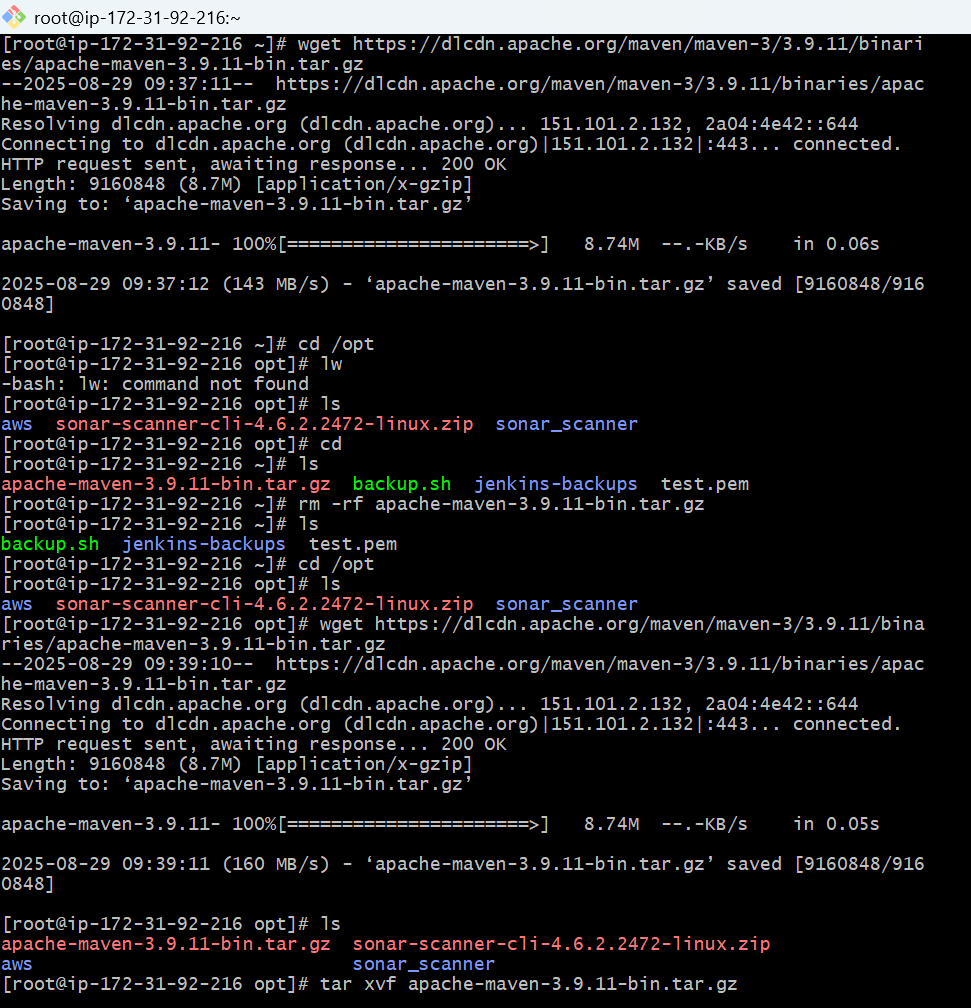
**3. Maven Compilation**

Install & Configure Maven in Jenkins

download apche maven in jenkins master

cd /opt

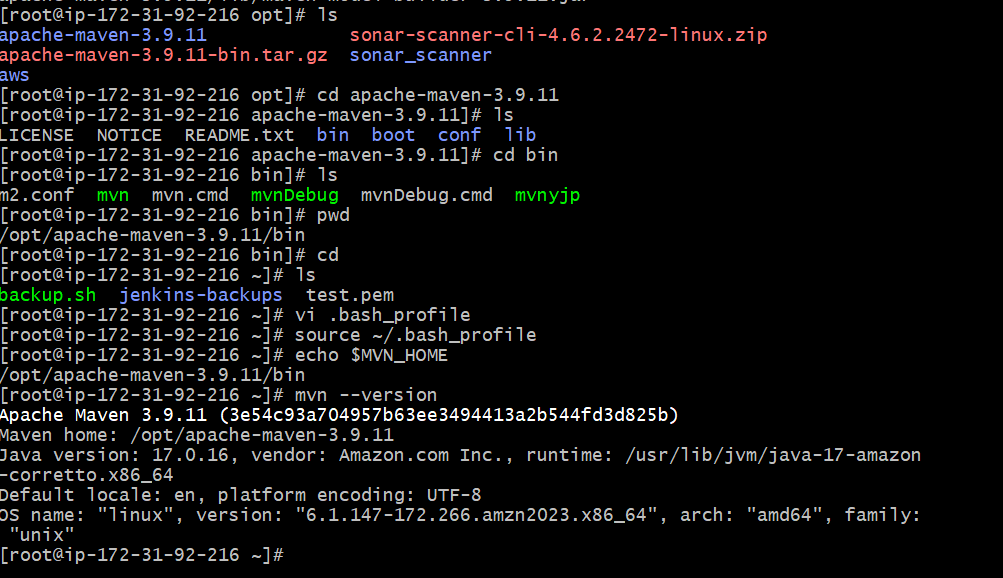
wget <https://dlcdn.apache.org/maven/maven-3/3.9.11/binaries/apache-maven-3.9.11-bin.tar.gz>



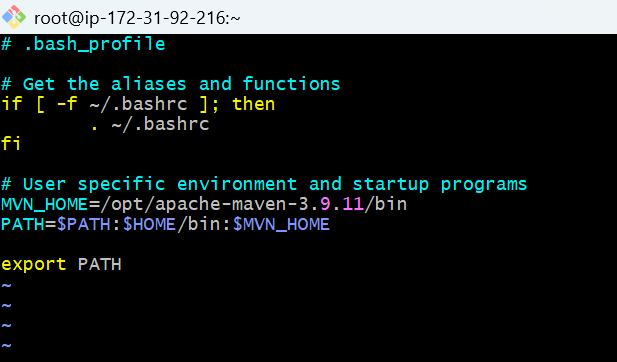
**tar -xvf apache-maven-3.9.11-bin.tar.gz**

**cd apache-maven-3.9.11/bin**

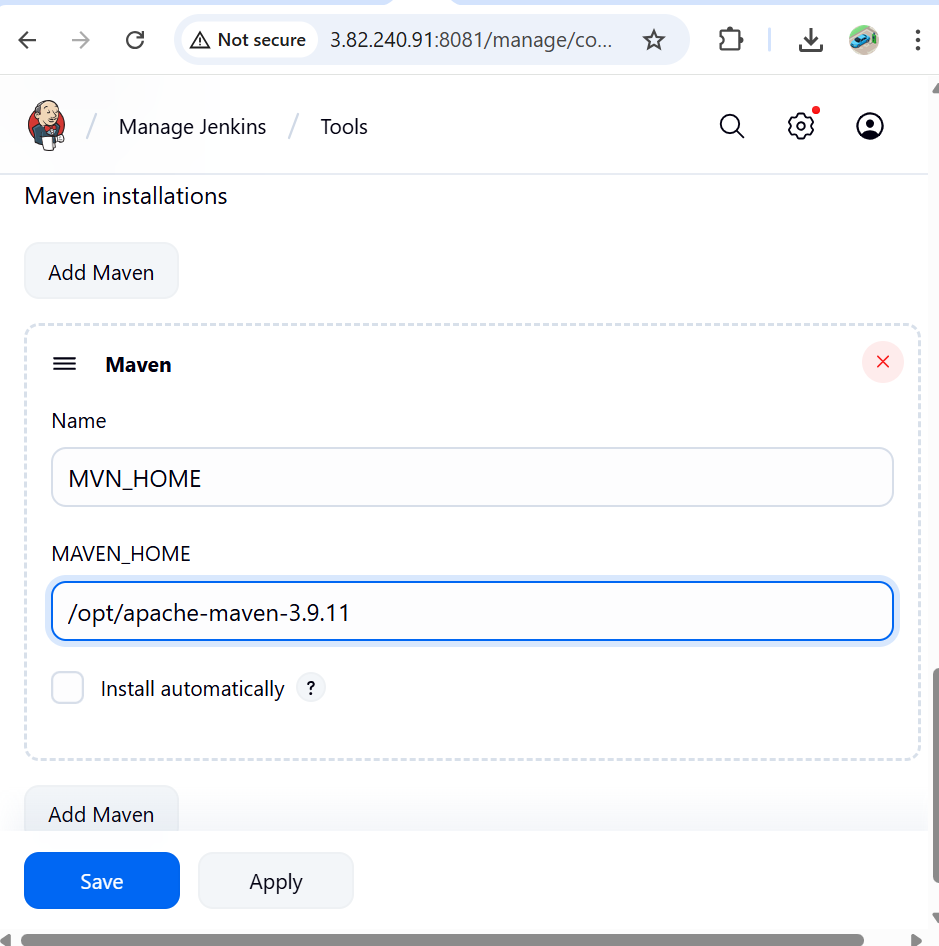
**Store global variable in script to access maven –version**



**In jenkins server** edit the file and some details → **vi .bash\_profile**

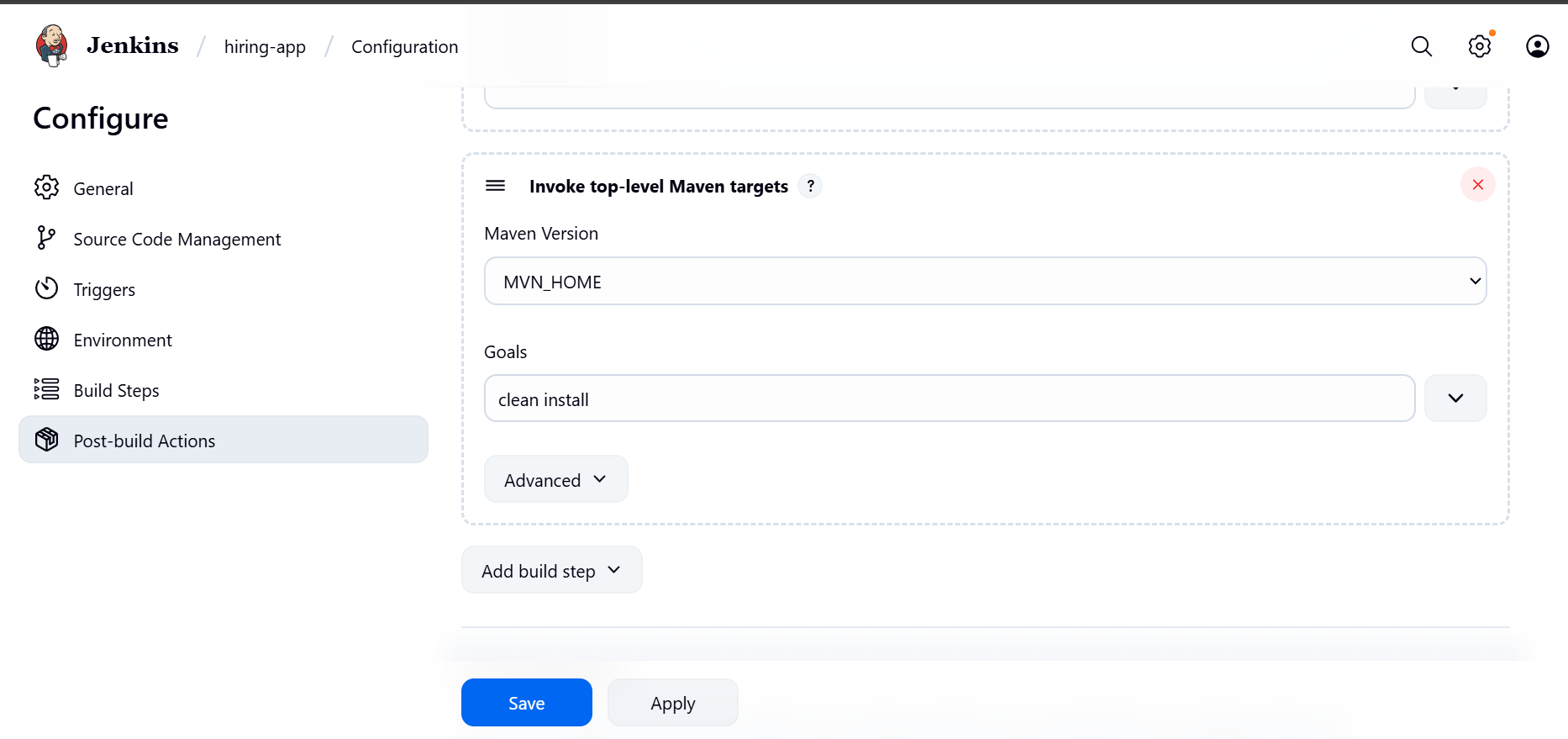


Now go to manage jenkins --> tools --> maven --> /opt/apache-maven-3.9.11

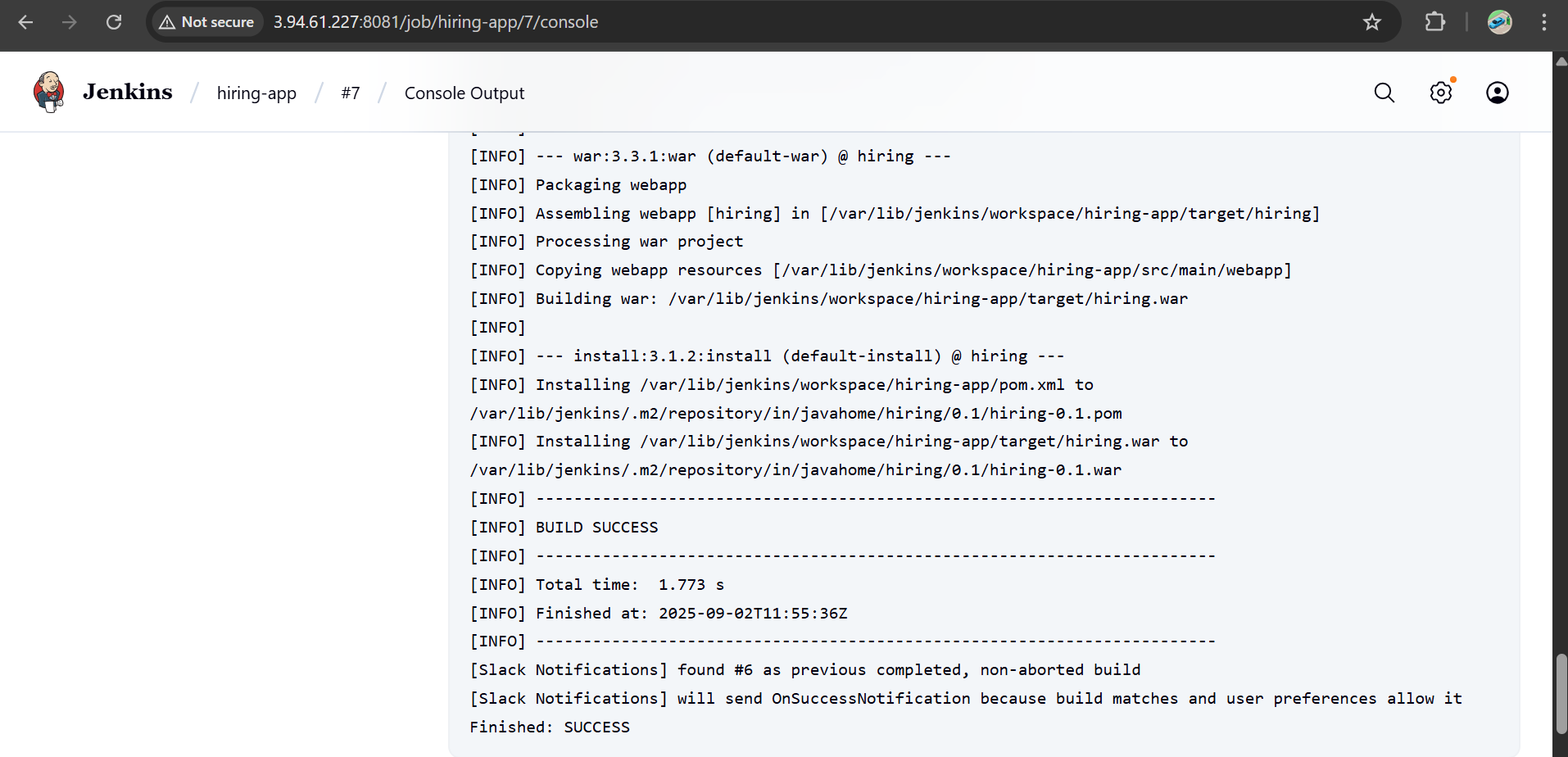


Add build step → **Invoke top-level Maven targets**.

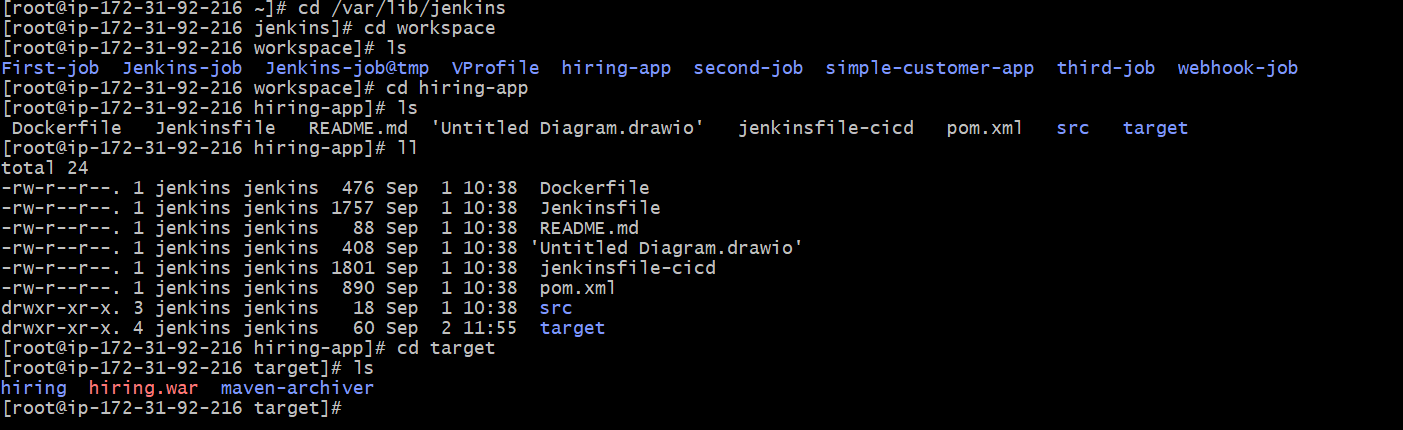
Goals: clean



Now run the build then git clone done, sonarqube integration done, then maven created a war file.

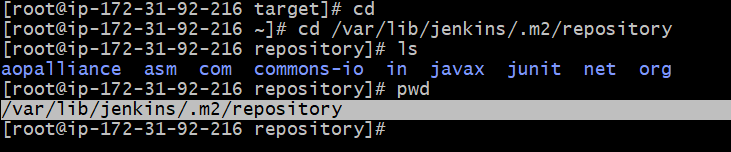


**In jenkins server → cd /var/lib/jenkins/workspace/hiring-app/target**

****

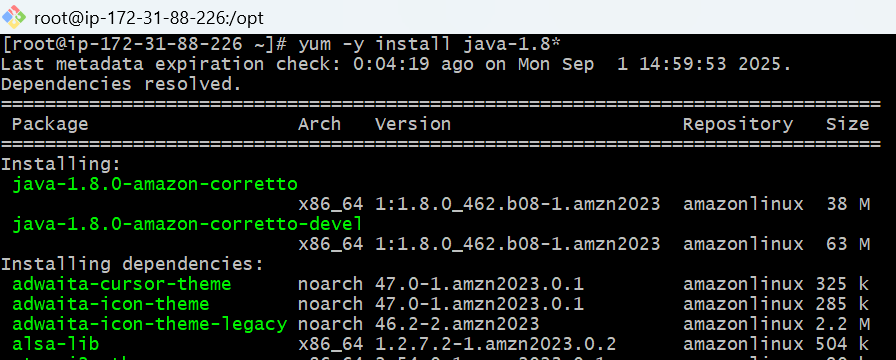
We can check dependencies in jenkins server

cd /var/lib/jenkins/.m2/repository



**4. Nexus Artifactory**

1. Now create nexus-ec2 with t2.large
2. Install java → **yum -y install java-1.8\***

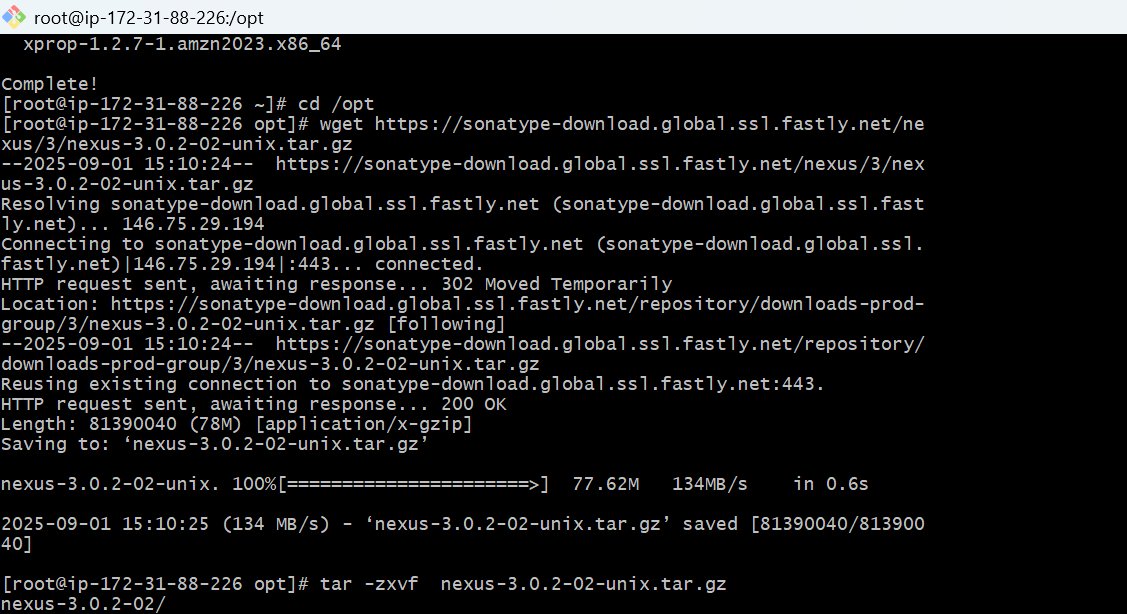


1. And now download nexus

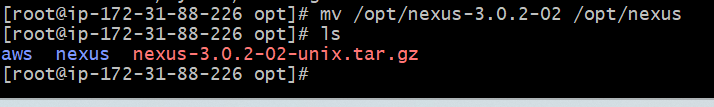
cd /opt

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

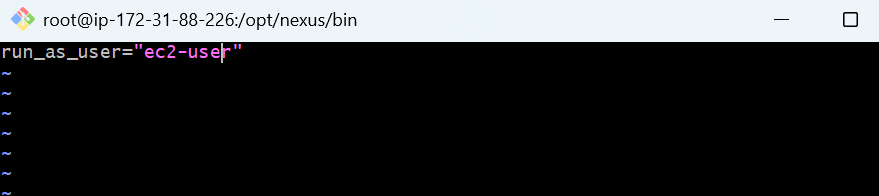


Open /opt/nexus/bin/nexus.rc file,

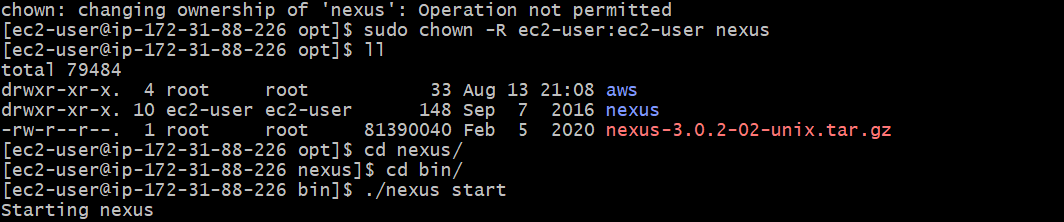
uncomment run\_as\_user parameter and set it as following.

vi /opt/nexus/bin/nexus.rc

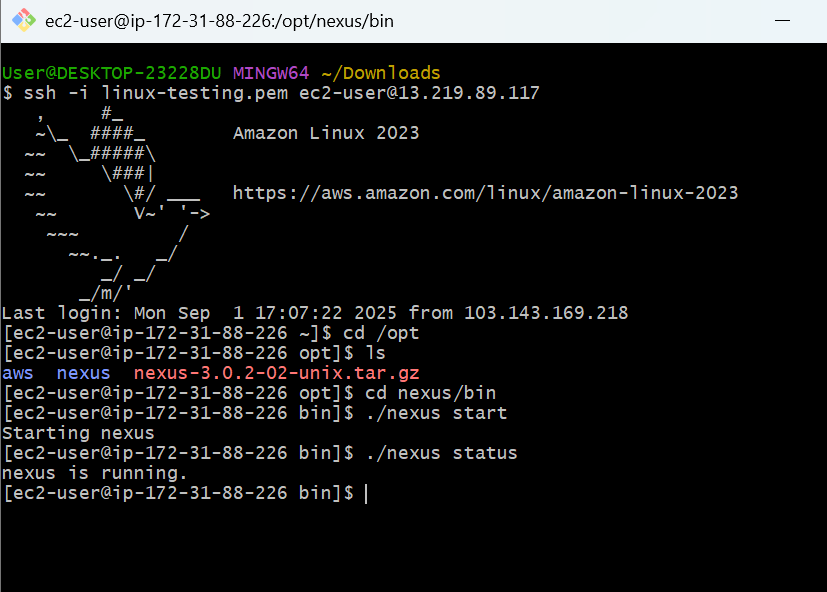
Remove # and Add → run\_as\_user="**ec2-user**"



* Now switch to ec2-user give ownership and start and check status.
* Cd /opt
* Give permissions → **sudo** **chown -R ec2-user:ec2-user nexus**



* **Cd /opt/nexus/bin**
* **./nexus start**
* **./nexus status**



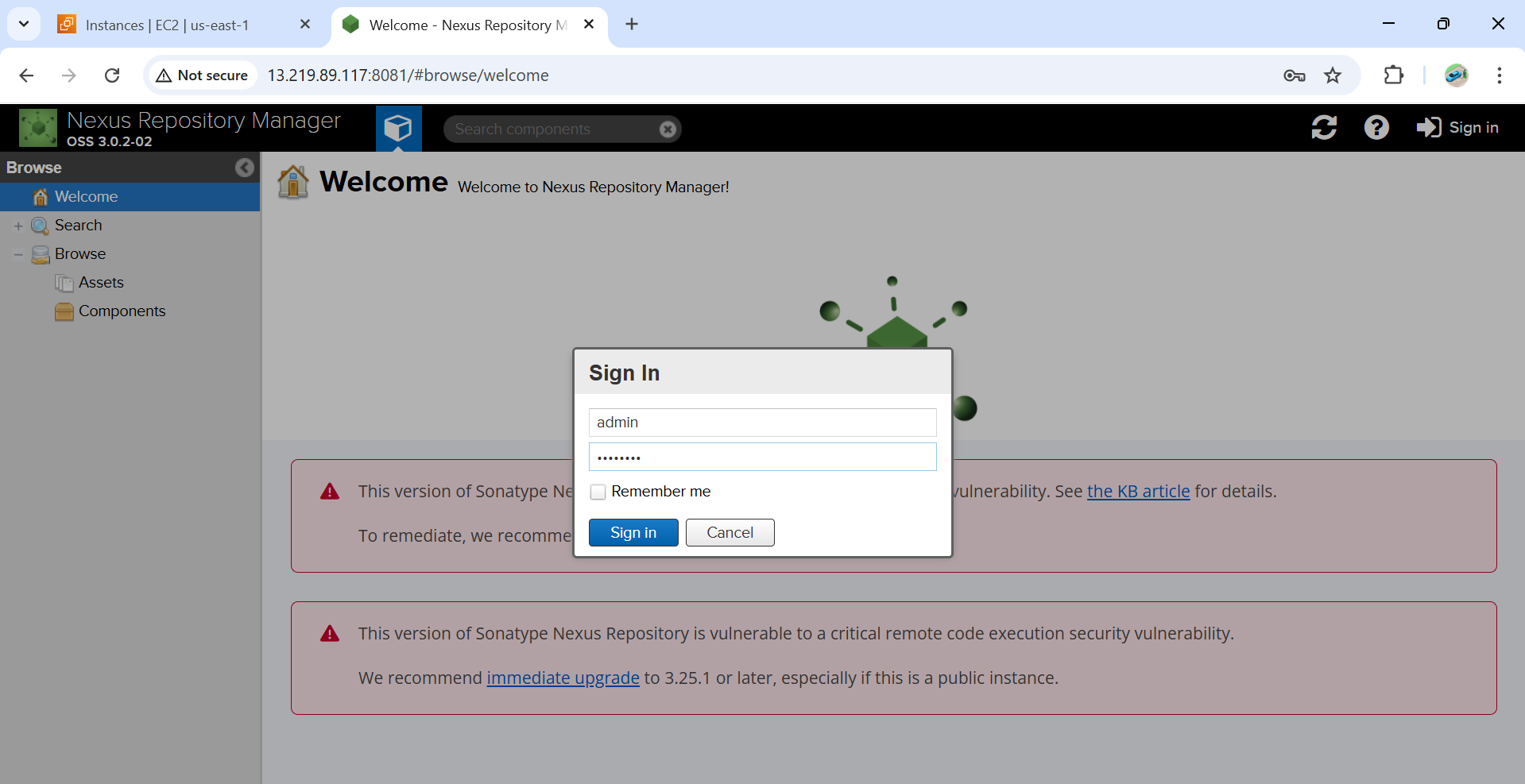
Login nexus server from browser on port 8081

http://<Nexus\_server>:8081

Use default credentials to login

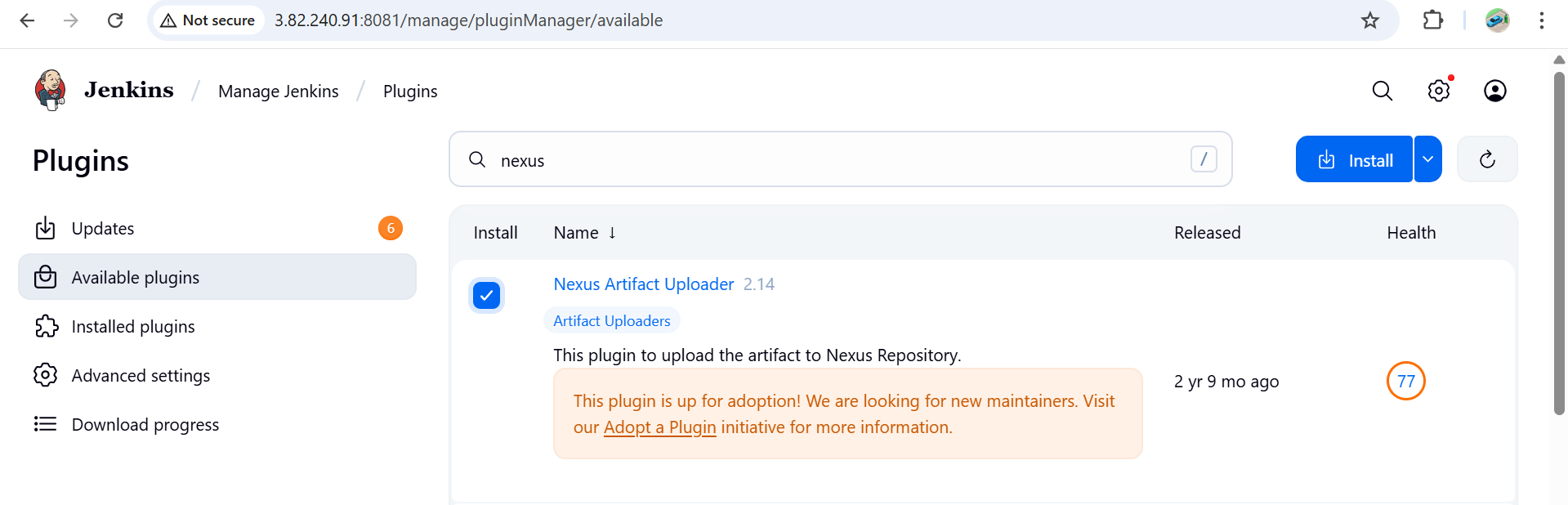
username : admin

password : admin123



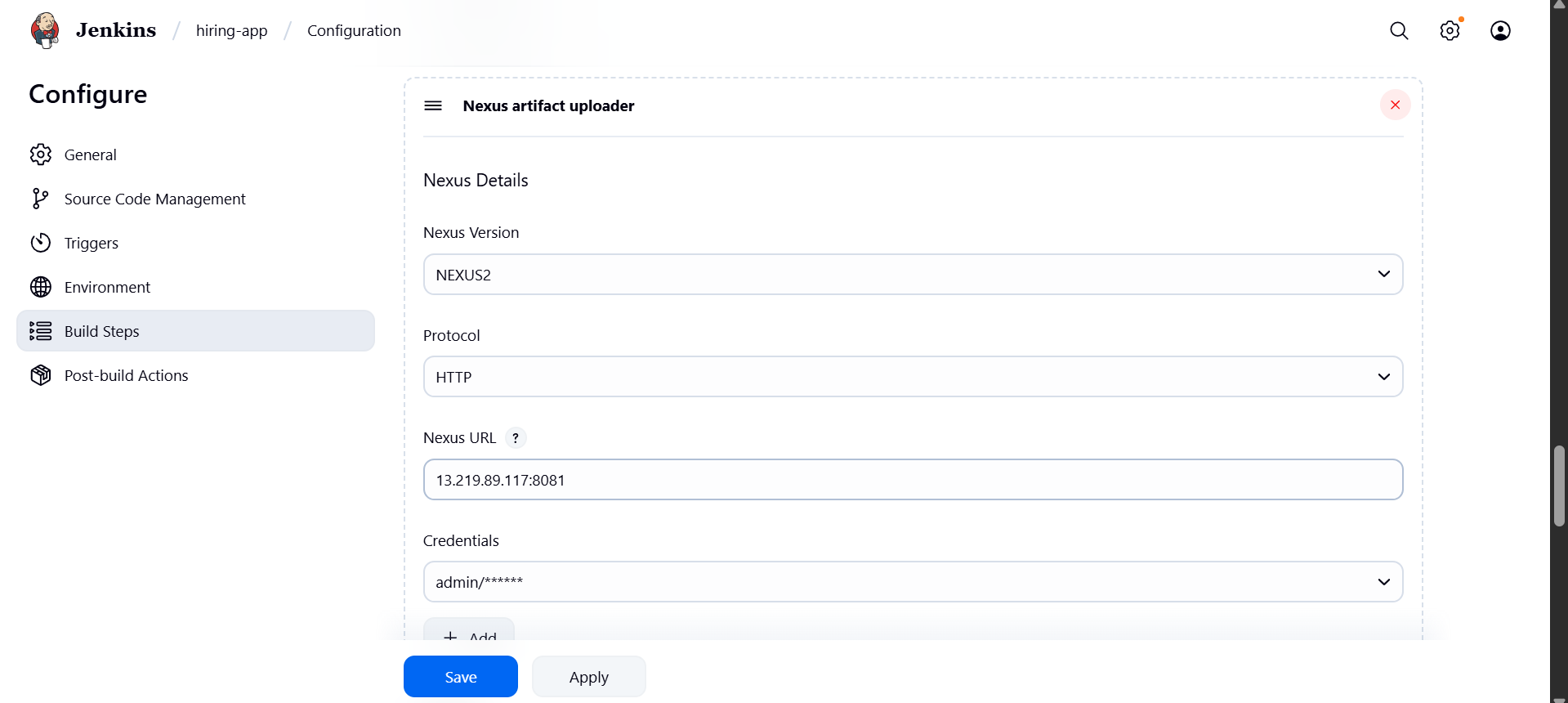
**Integrate jenkins with nexus**

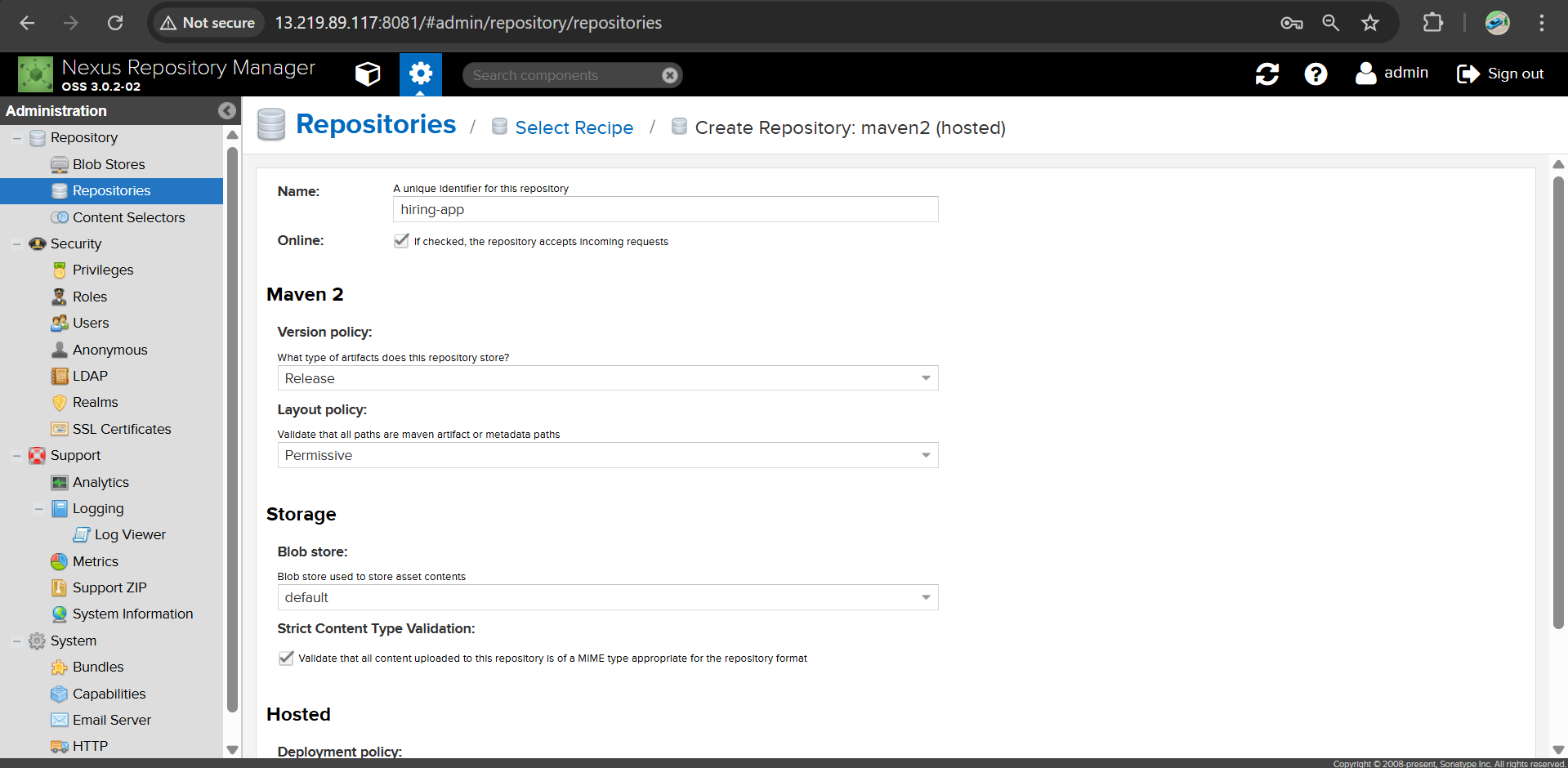
Install Nexus plugin

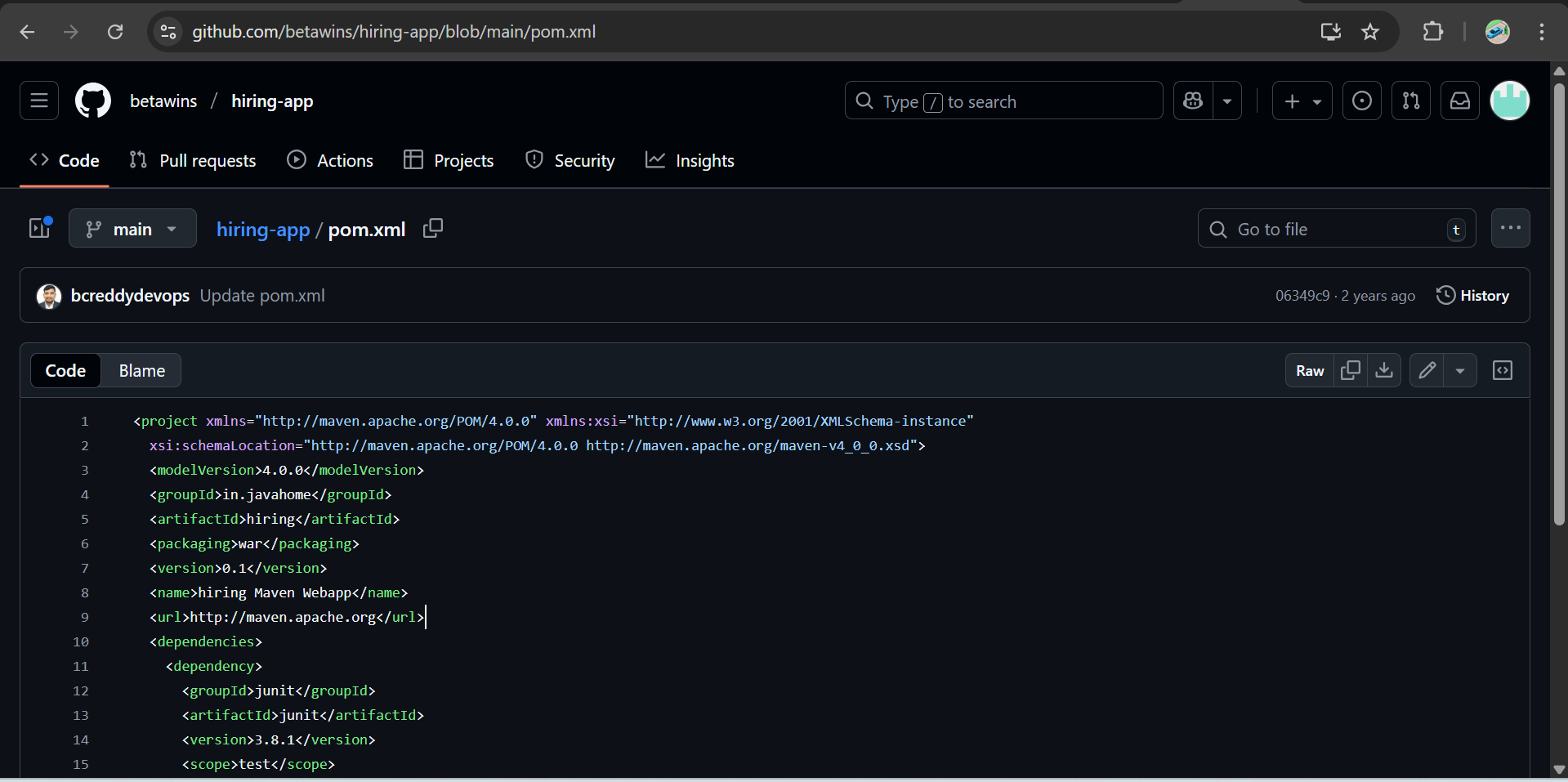


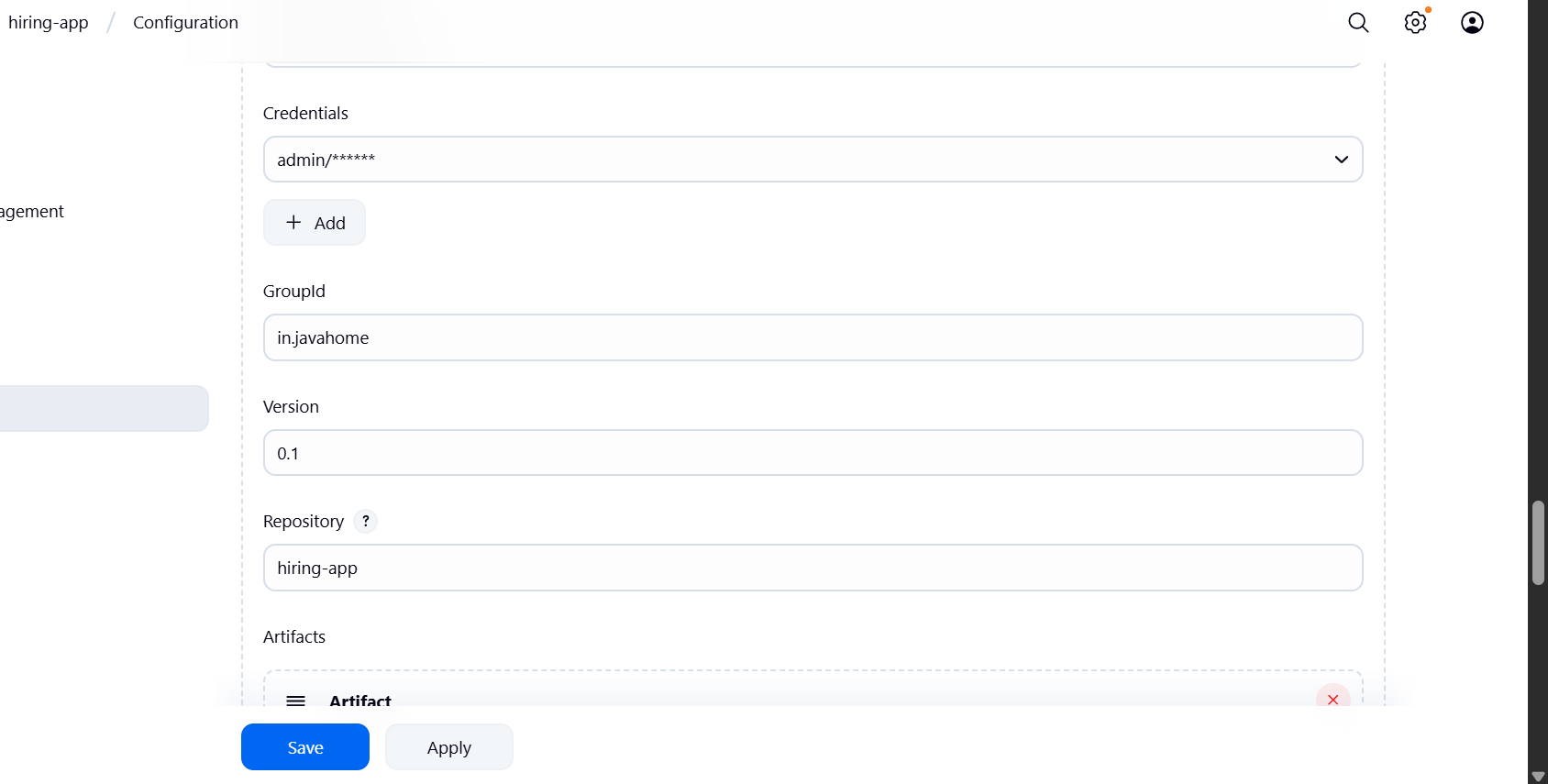
**Now go to your job → configure → build step → nexus**

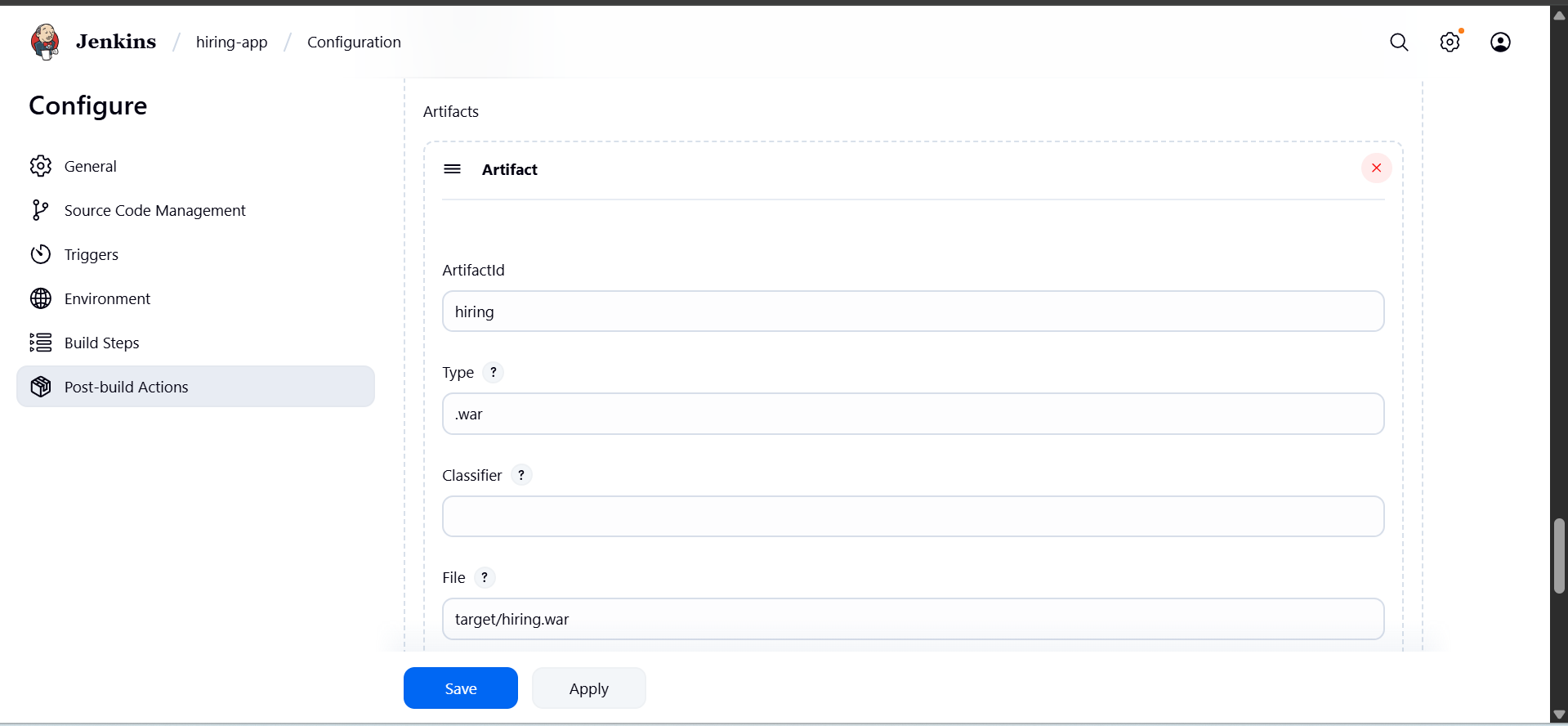
**In git hub repo in pom.xml file we have some info group id and version, bring and enter here and goto nexus create repository → maven2(host) and take release.**



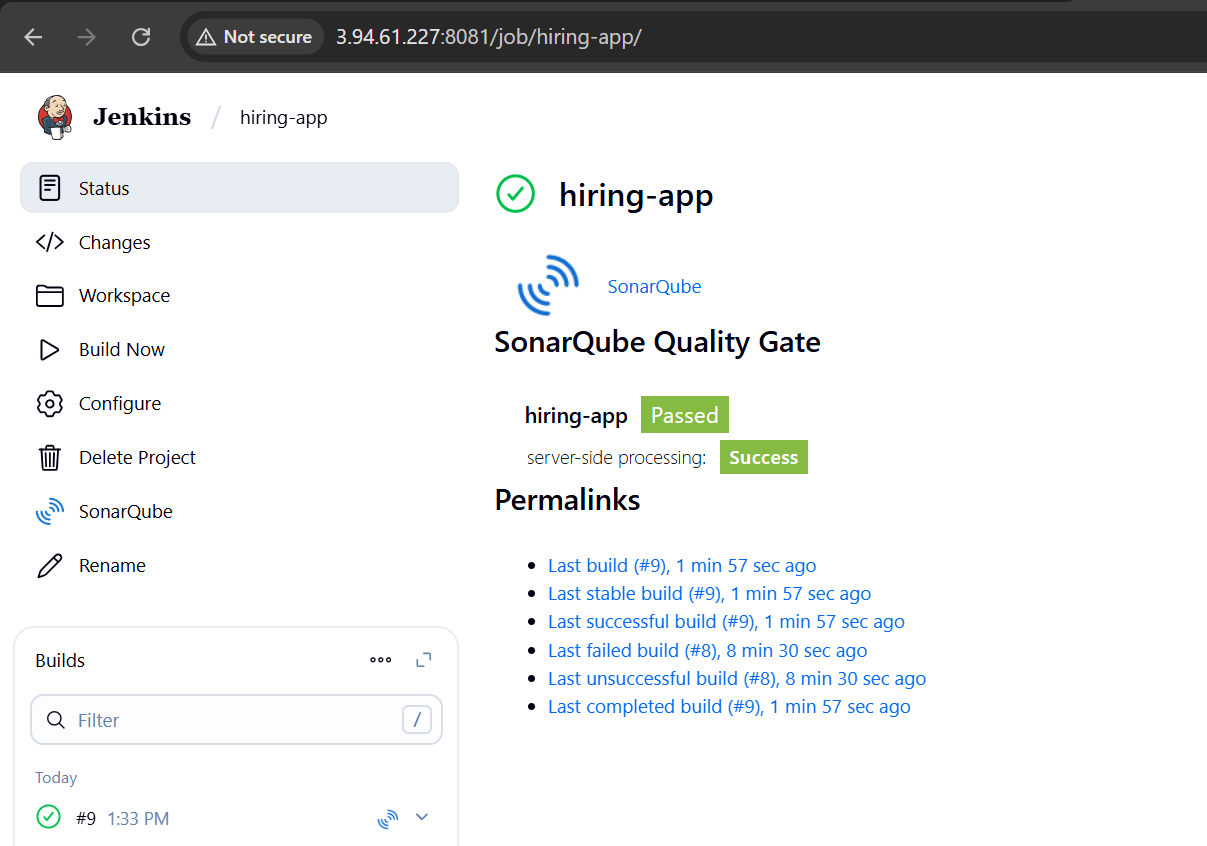




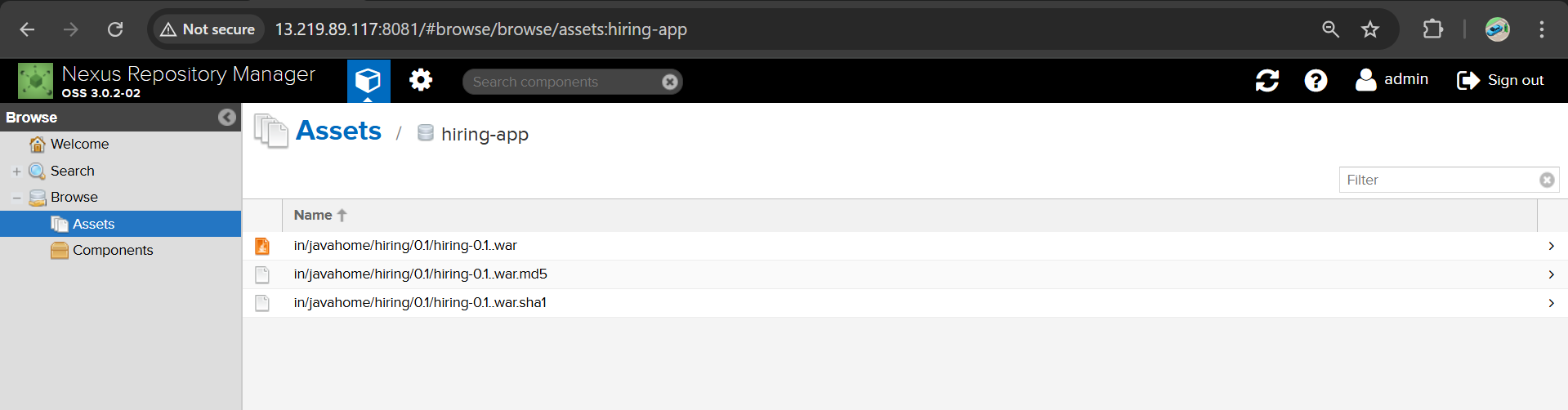




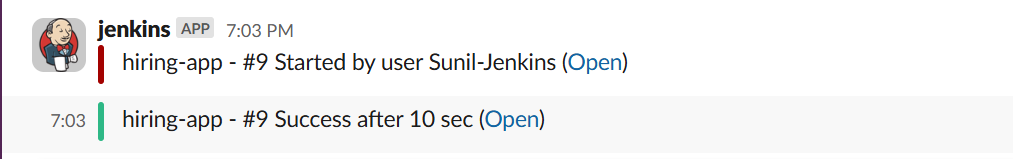
**Now run build**



Check Nexus for war file

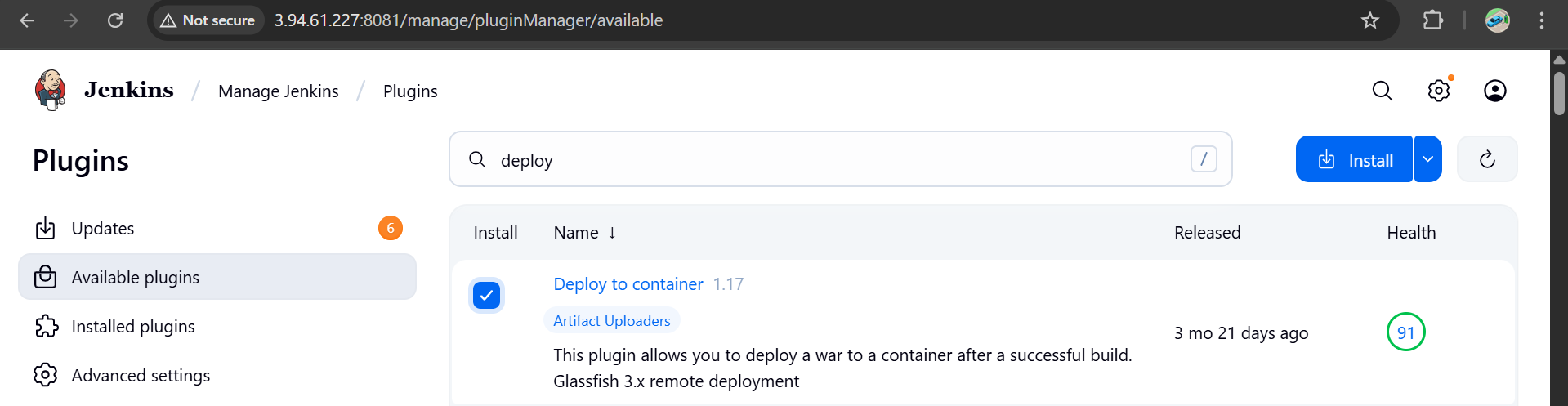


**5) Slack Notification**

****

**6) Deploy On tomcat**

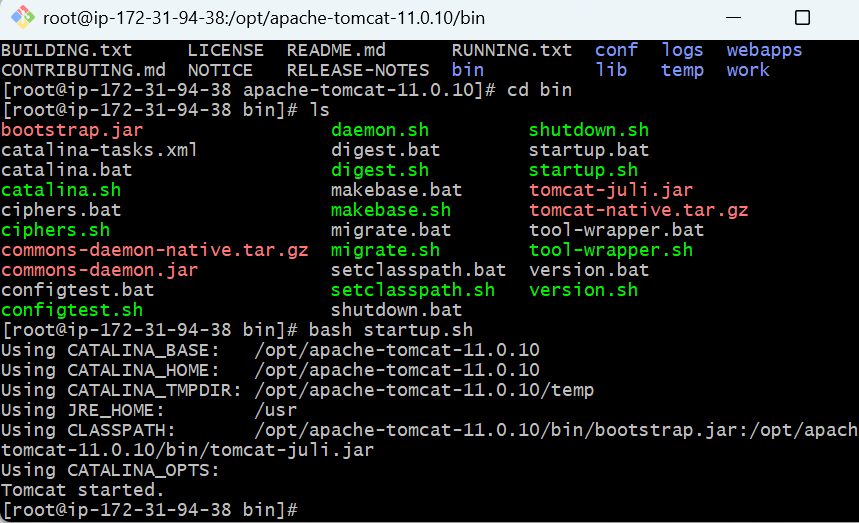
**Go to jenkins page Install plugin → deploy to container**

****

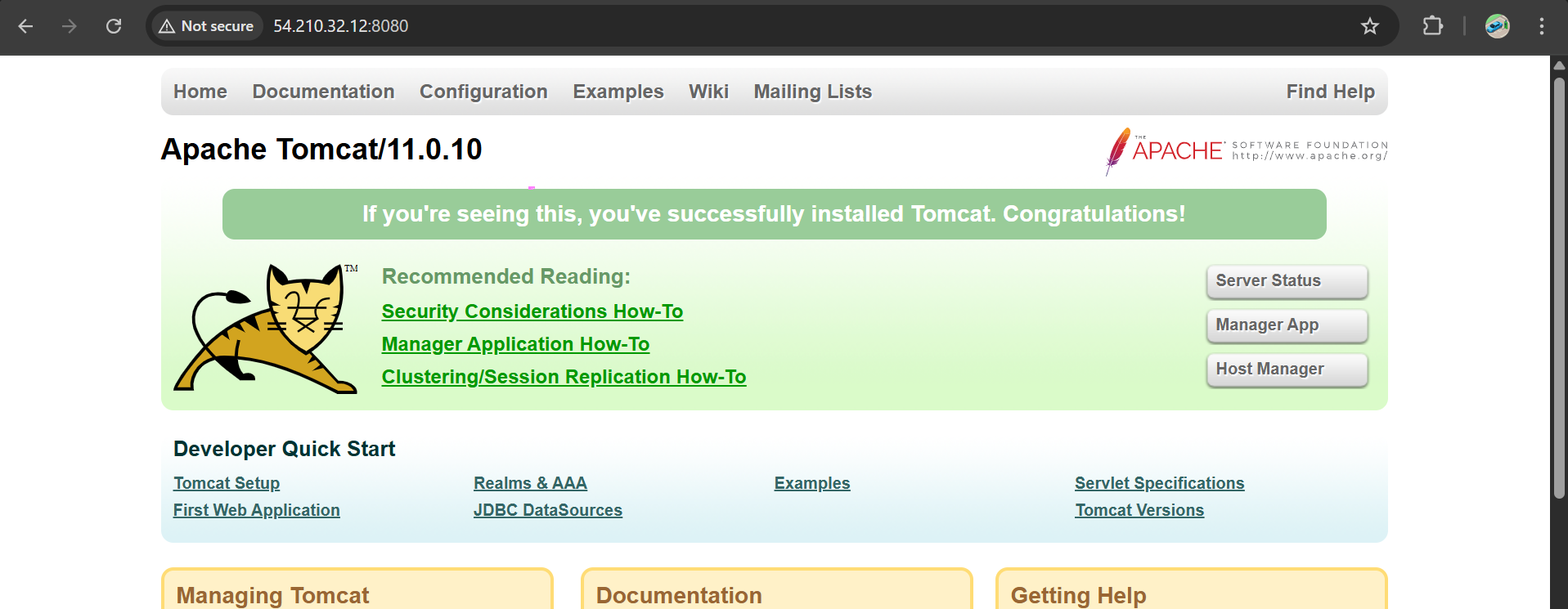
**Now launch an ec2 instance**

**Install java and install tomcat in it**

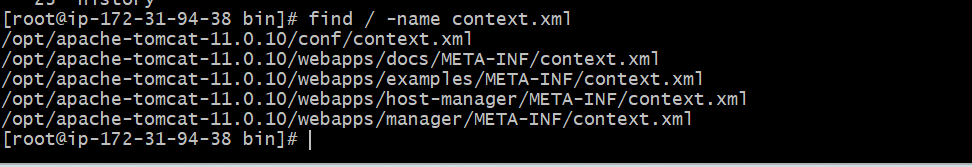
**And start**

****

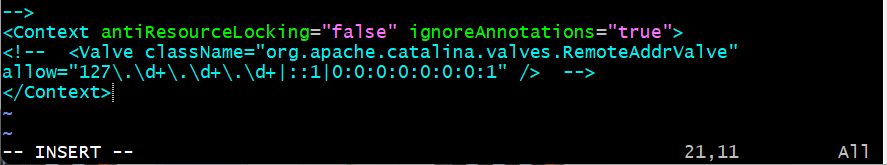
Check on the web with the public ip

****

* **Now create user in tomcat.**
* **Take file webapps and edit for uncomment the four files**
* **find / -name context.xml**



vi /opt/apache-tomcat-11.0.10/webapps/docs/META-INF/context.xml

****

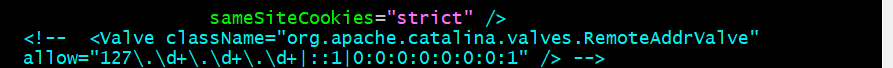
vi /opt/apache-tomcat-11.0.10/webapps/examples/META-INF/context.xml

****

vi /opt/apache-tomcat-11.0.10/webapps/host-manager/META-INF/context.xml

****

vi /opt/apache-tomcat-11.0.10/webapps/manager/META-INF/context.xml

****

**Now go to conf in xml file add the below user details as shown in box.**

**cd /opt/** apache-tomcat-11.0.10/conf/

**vi tomcat-users.xml**

*<role rolename="manager-gui"/>*

*<role rolename="manager-script"/>*

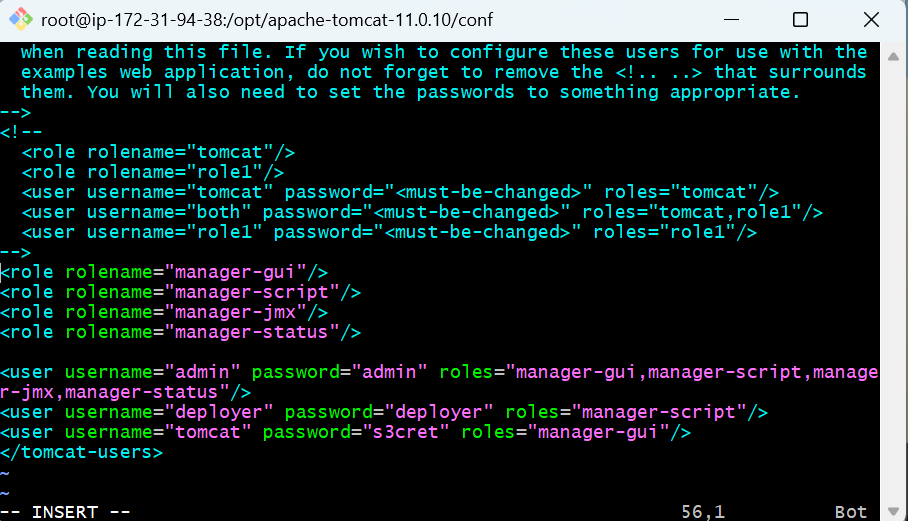
*<role rolename="manager-jmx"/>*

*<role rolename="manager-status"/>*

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

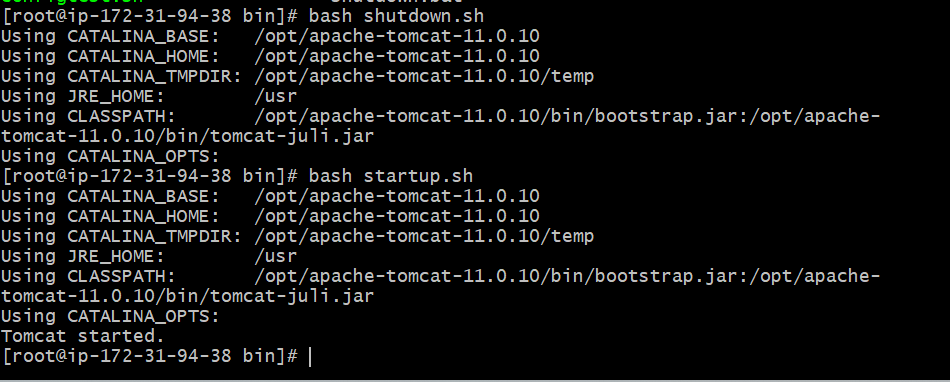
*<user username="deployer" password="deployer" roles="manager-script"/>*

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

****

**After editing**

**Shutdown and start the tomcat**

****

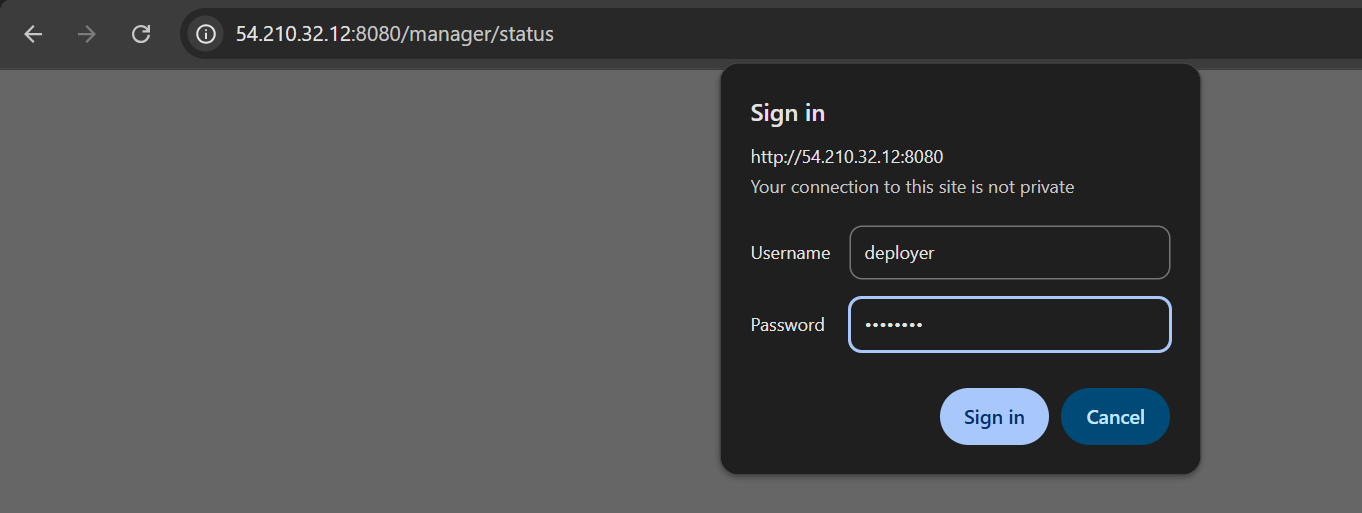
**Now in web page click on server status**

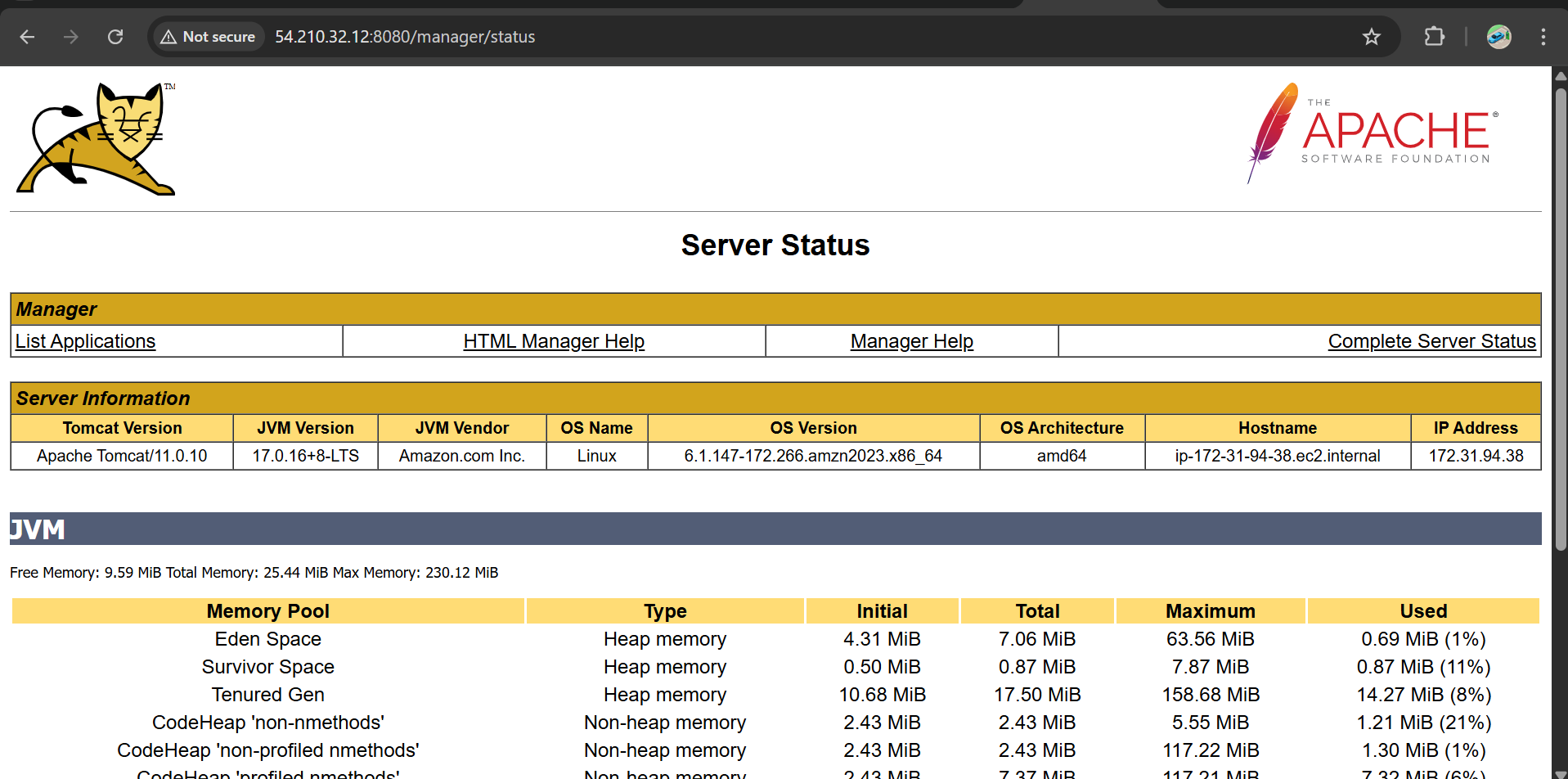
****

**Goto the tomcat page click on server status**

**Username – deployer**

**Password – deployer**

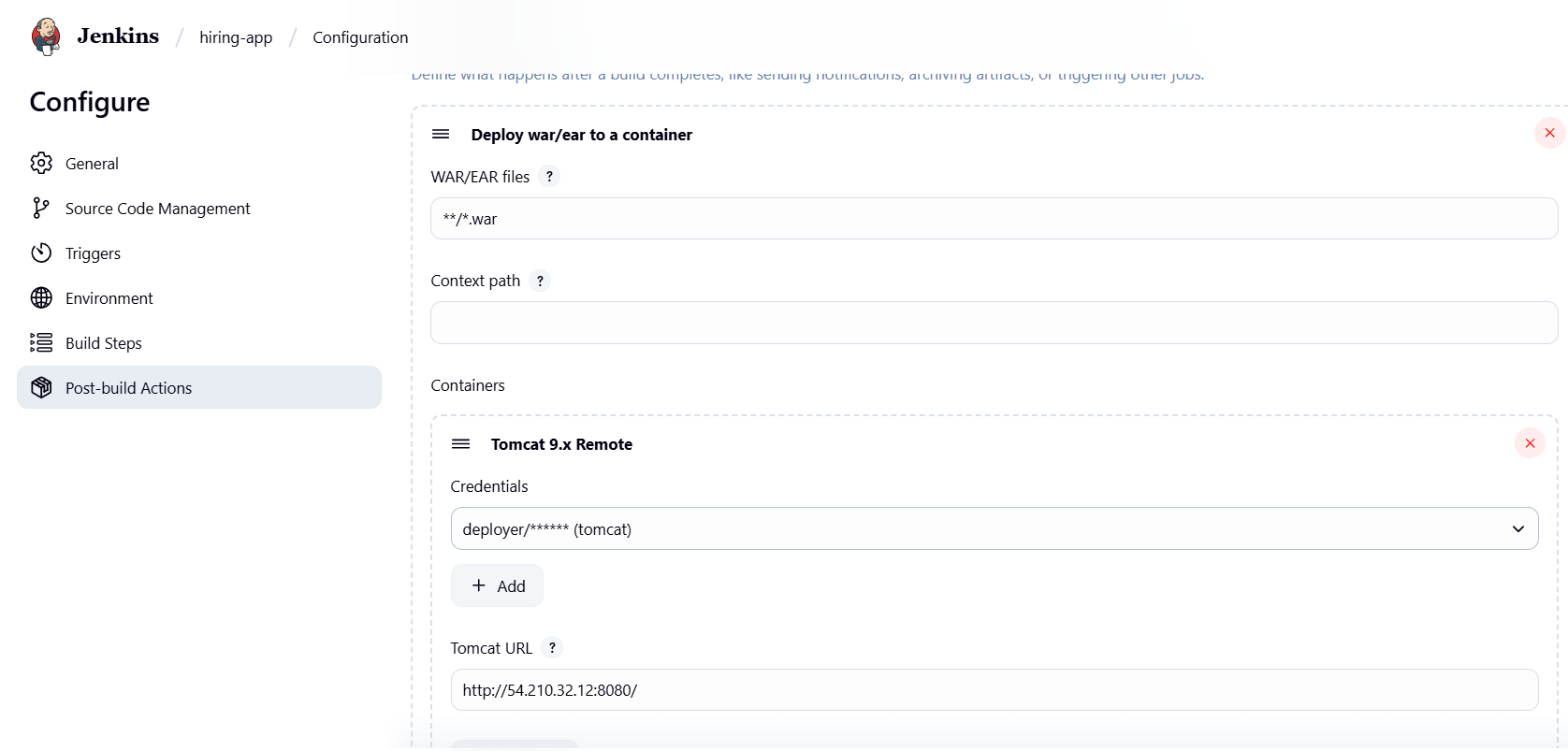
****

****

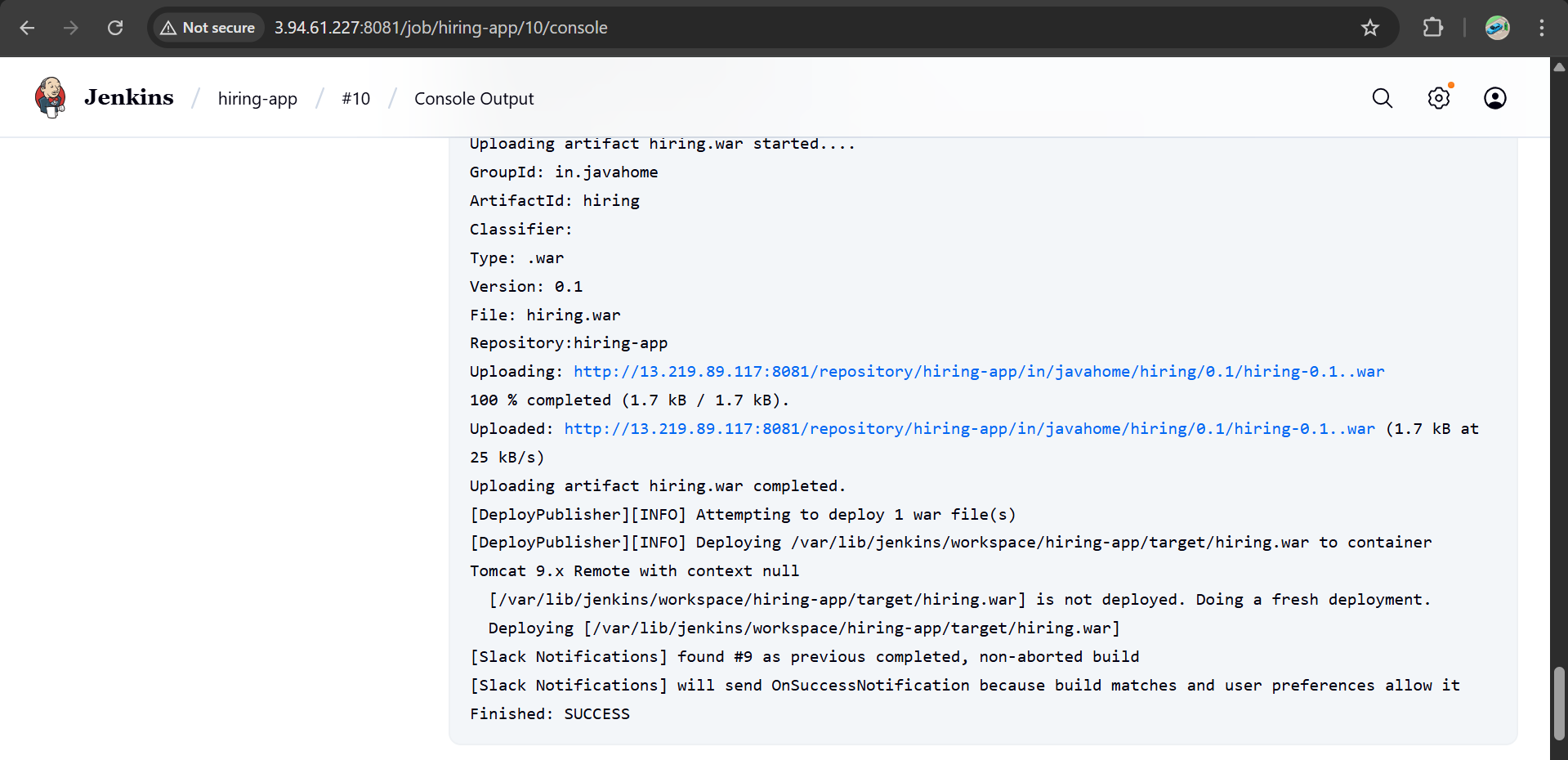
**Just go to jenkins add these credentials deployer and take WAR/EAR files as \*\*/\*.war**

**Add the same credentials**

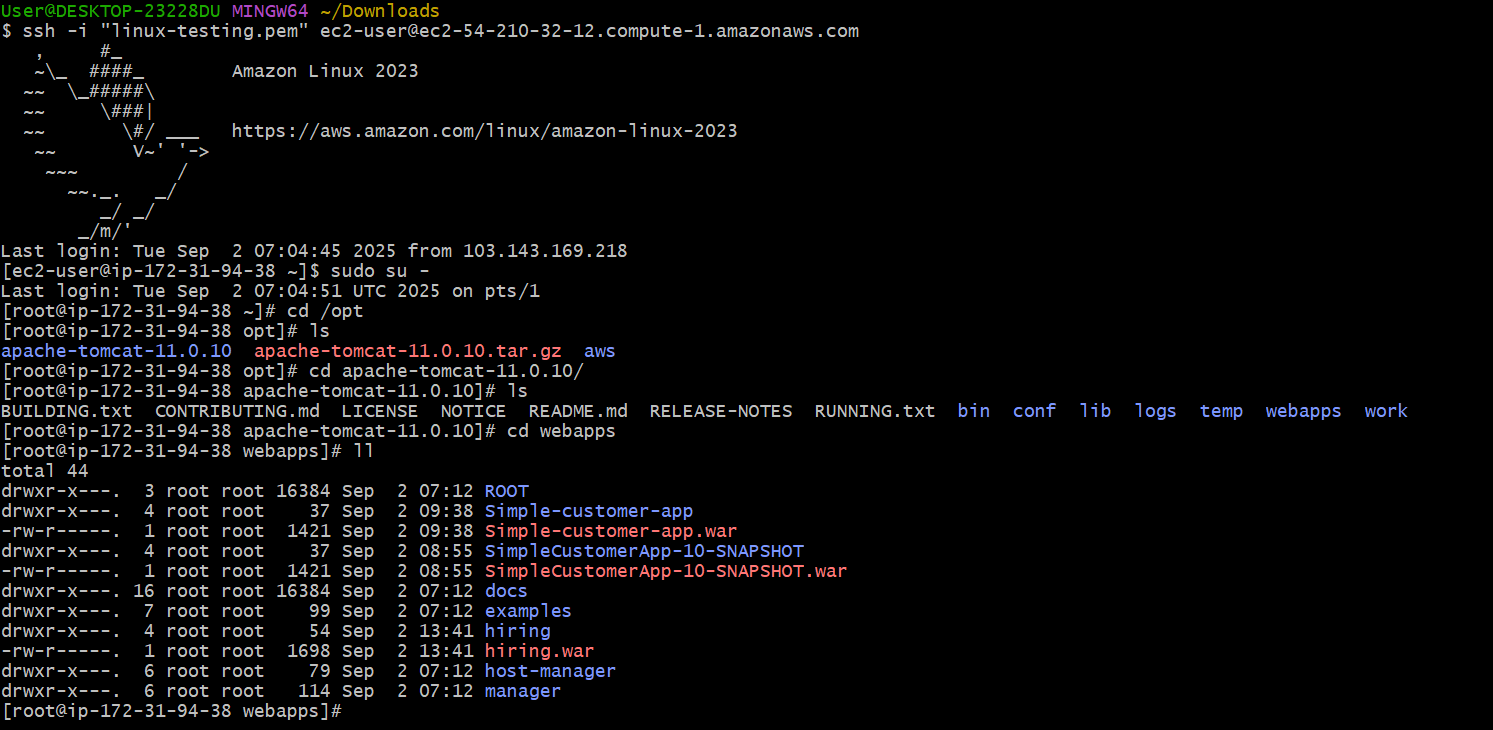
**Username – deployer , Password – deployer**



After save Run Build



* **Now our deployment successfully done**
* Now you can check in tomcat cli webapps one file is shown
* Cd /opt/tomcat/webapps/hiring



**And you can check in tomcat at browser we got our application is live.**

<http://54.210.32.12:8080/hiring/>

