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

3) Maven Compilation

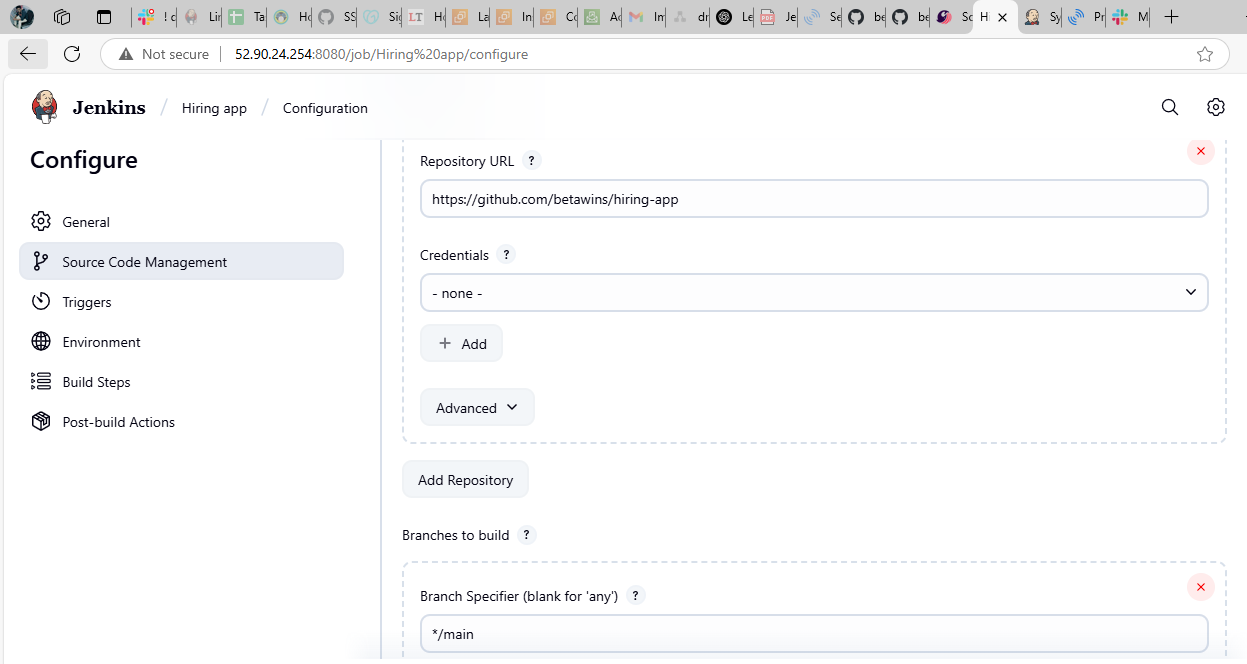
4) Nexus Artifactory

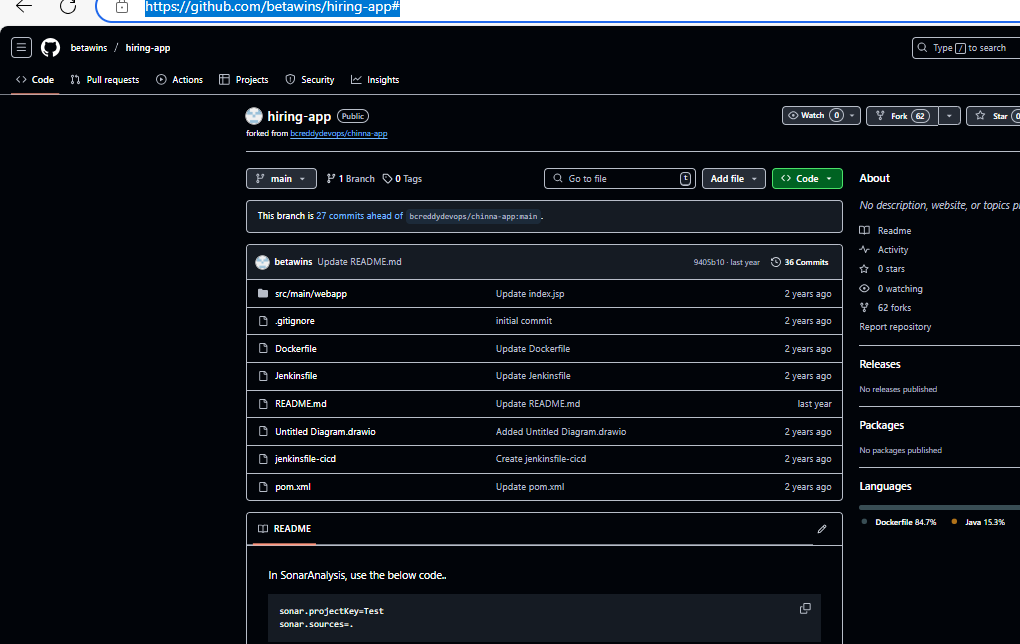
5) Slack Notification

6) Deploy On tomcat

Stage 1: Git clone

Create new project add git hub url and set branch as main and then in the buid steps copy the text from source code and install git plugin





Stage 2: Sonarqube Integration

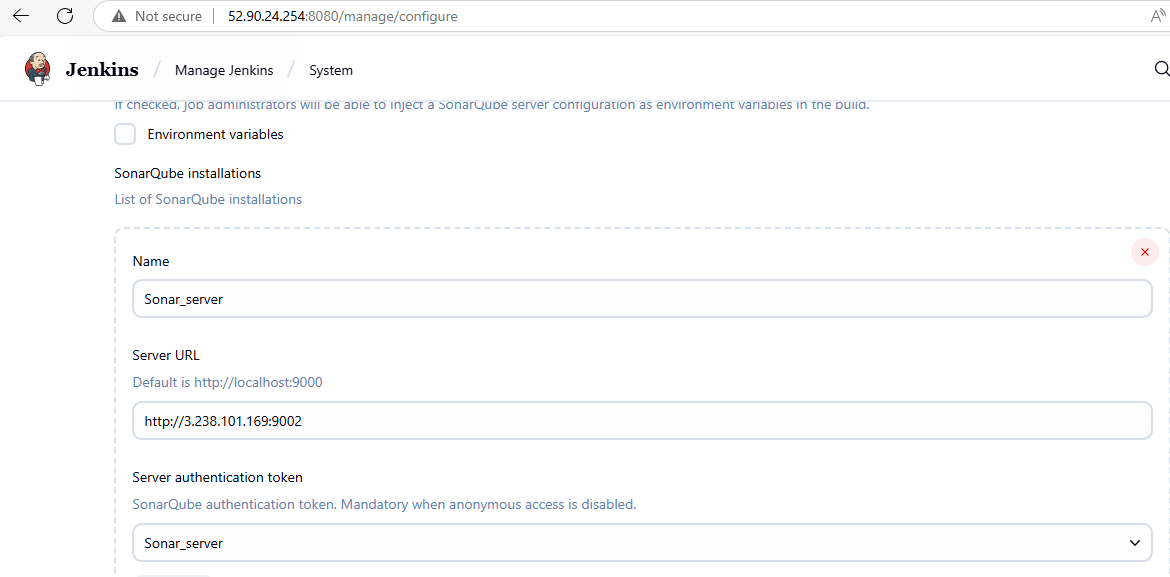
Create new instance with t2.large and 20 Gb storage and connect to it

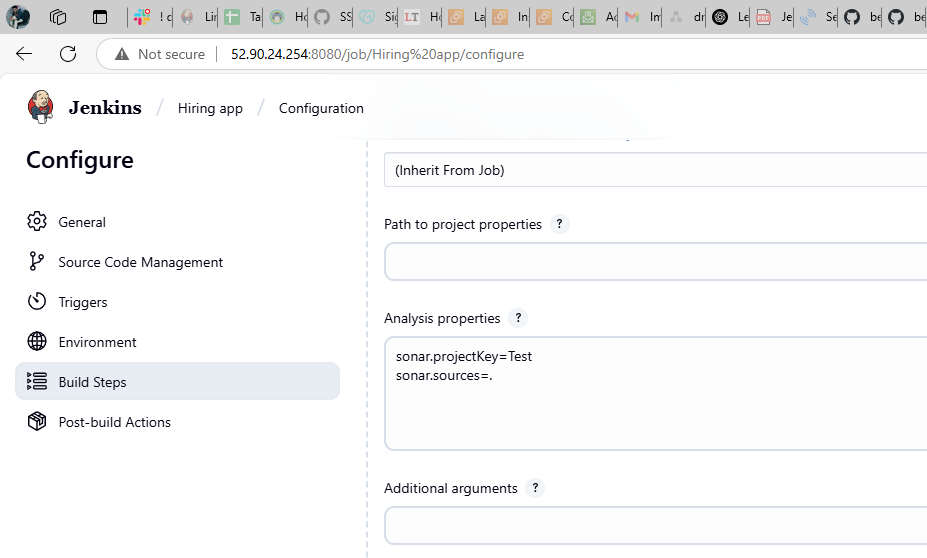
Install sonarqube server using Docker and access on the browser with “public IP:9000”

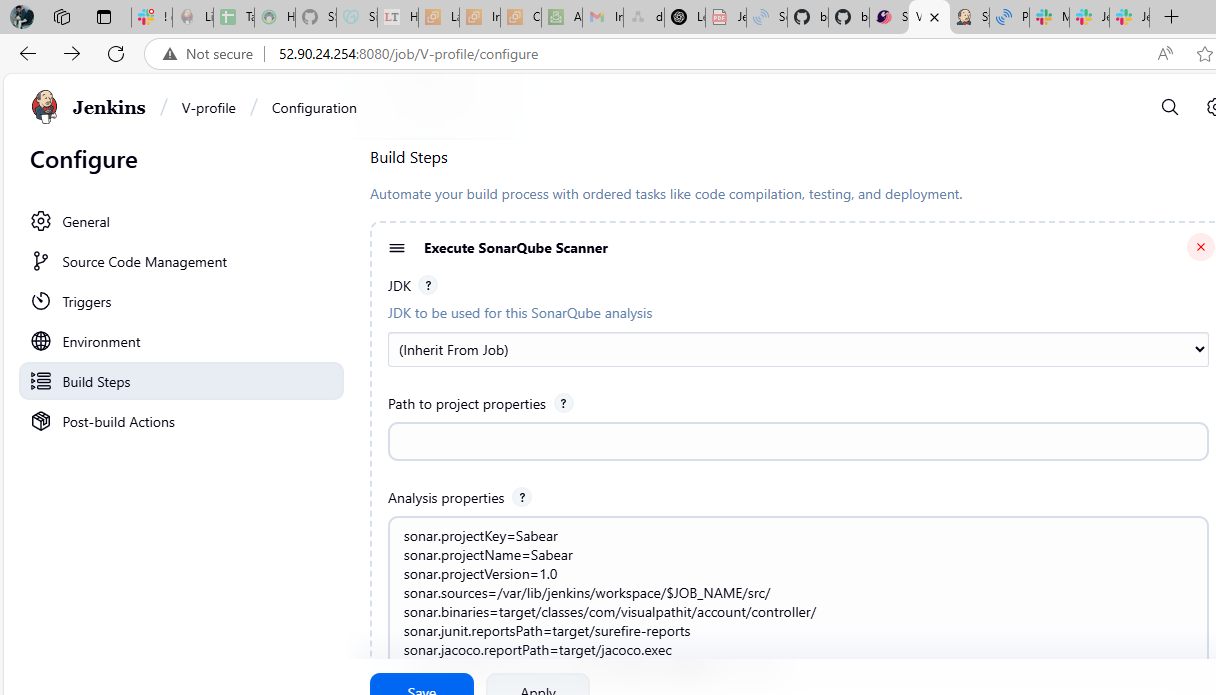
Create tokens in sonarqube:

🡪admin 🡪my account 🡪security 🡪name and generate tokens

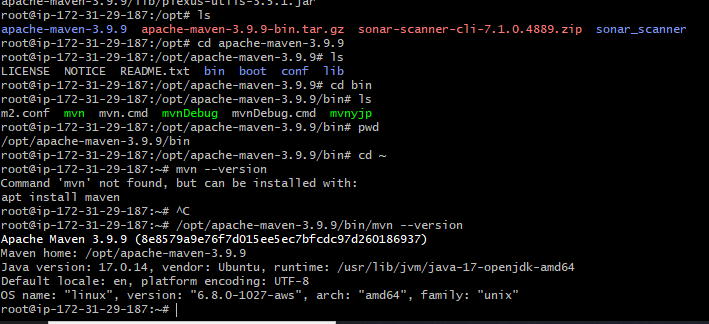
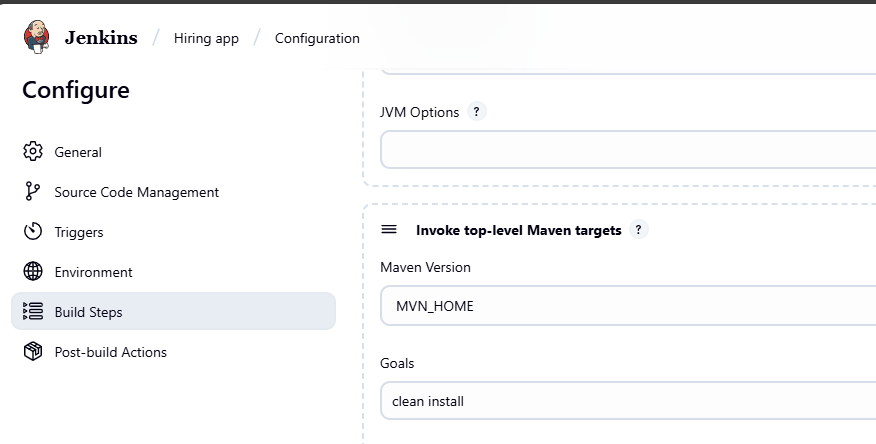
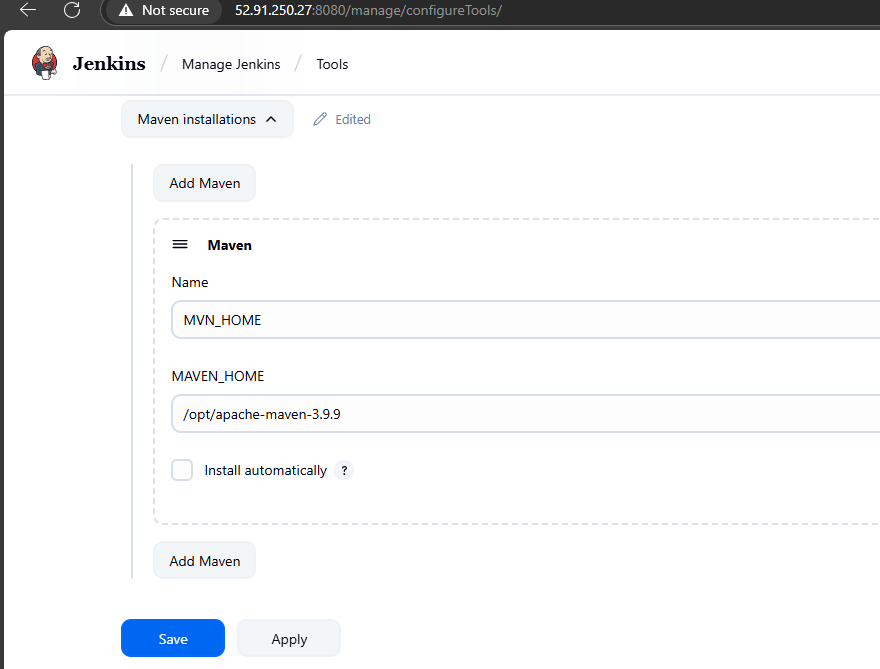
Copy this token in the “*secret”* option while adding credentials to jenkins





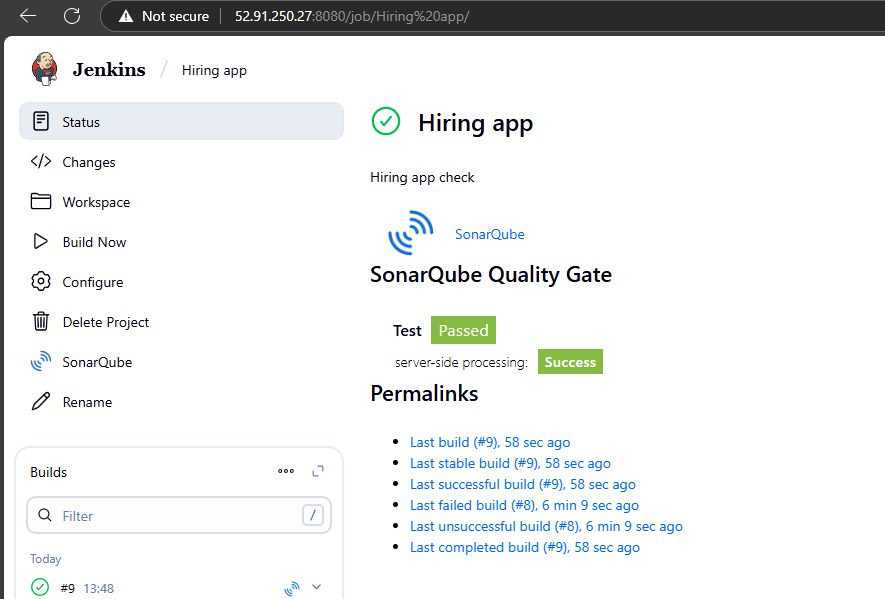


Stage 3:Maven compilation

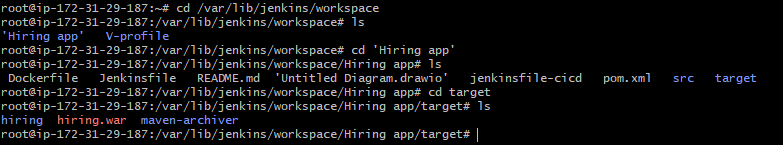
In Jenkins server goto /opt directory and download Maven   

Now check the directory with Job name in the workspace directory



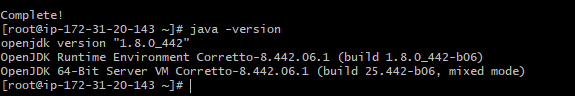


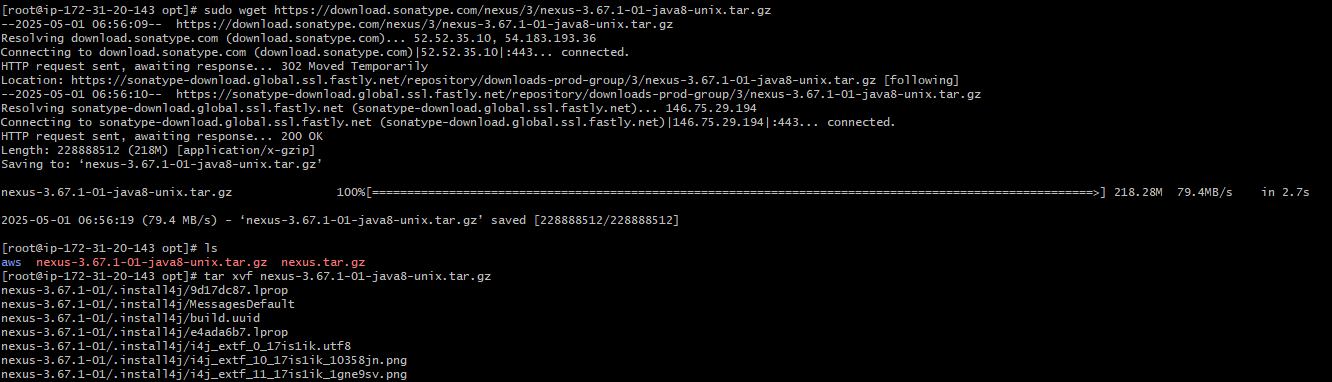
Now we can see pom.xml file & *target* directory created in workspace/Hiring\_app



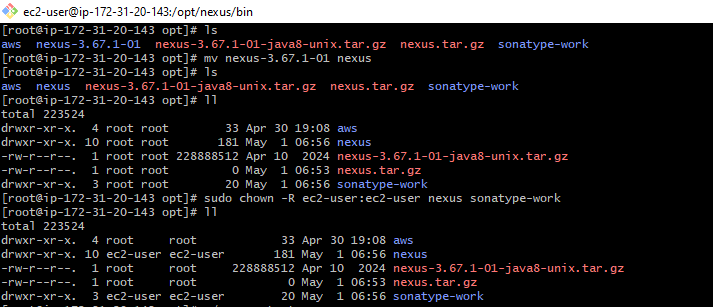
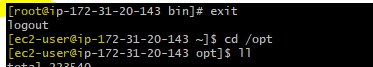
Stage 4:Nexus Artifactory

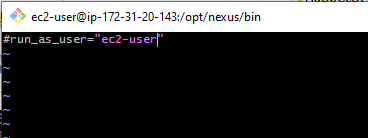
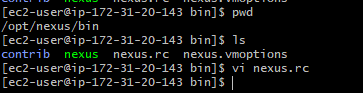
Create a new instance for nexus and connect to the instance and install

java-1.8\* and then goto /opt directory and 



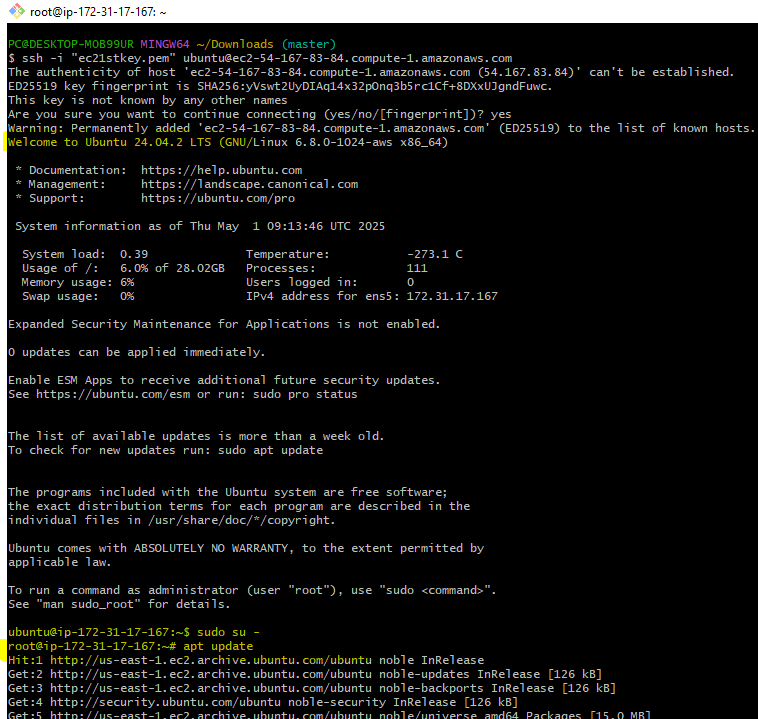
### Now untar the file and rename as nexus and change ownership of nexus directory and for sonatype-work also

 **Use** “**exit**” **command and change** from “ root to ec2-user ” 

Now goto cd /opt/nexus/bin >uncomment & make run\_as\_user to “ec2-user” 

Now start nexus file using ./nexus start and check status using ./nexus status

We can INSTALL AND START NEXUS BY USING DOCKER ALSO

Launch an ec2 instance with t3.medium and Ubuntu AMI with EBS volume of 30 GB and connect to it . 

Now switch root user using sudo su –

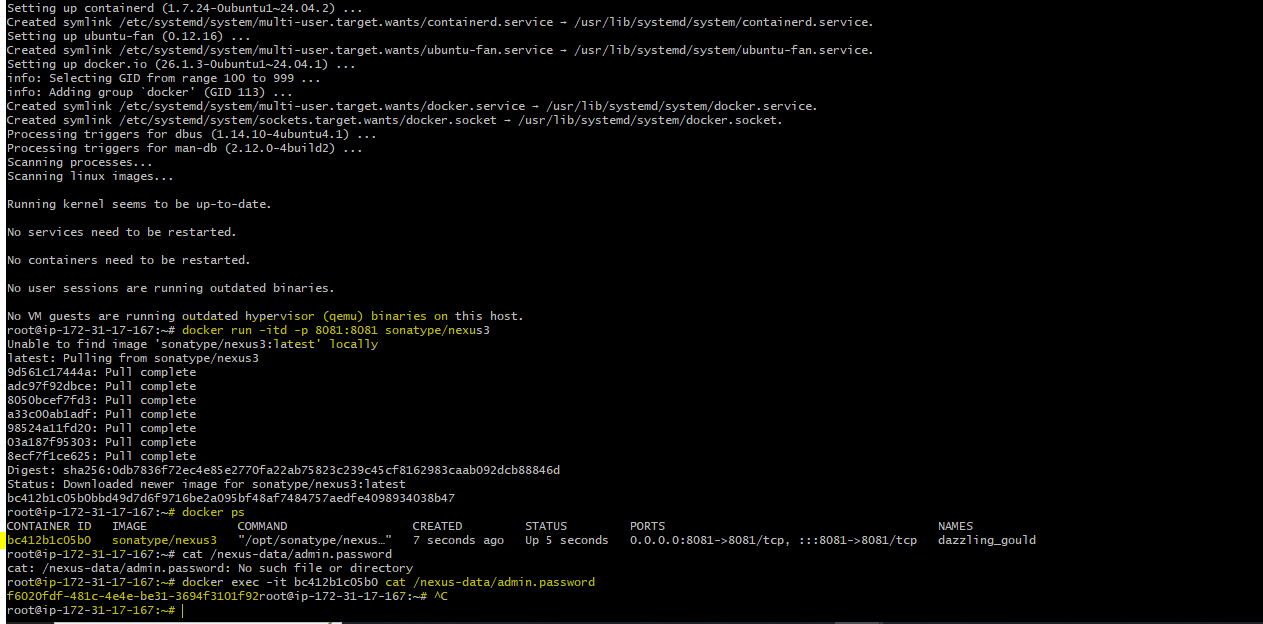
UPDATE PACKAGES using *apt update*

To download docker image for sonatype nexus >>docker run -itd -p 8081:8081 sonatype/nexus3

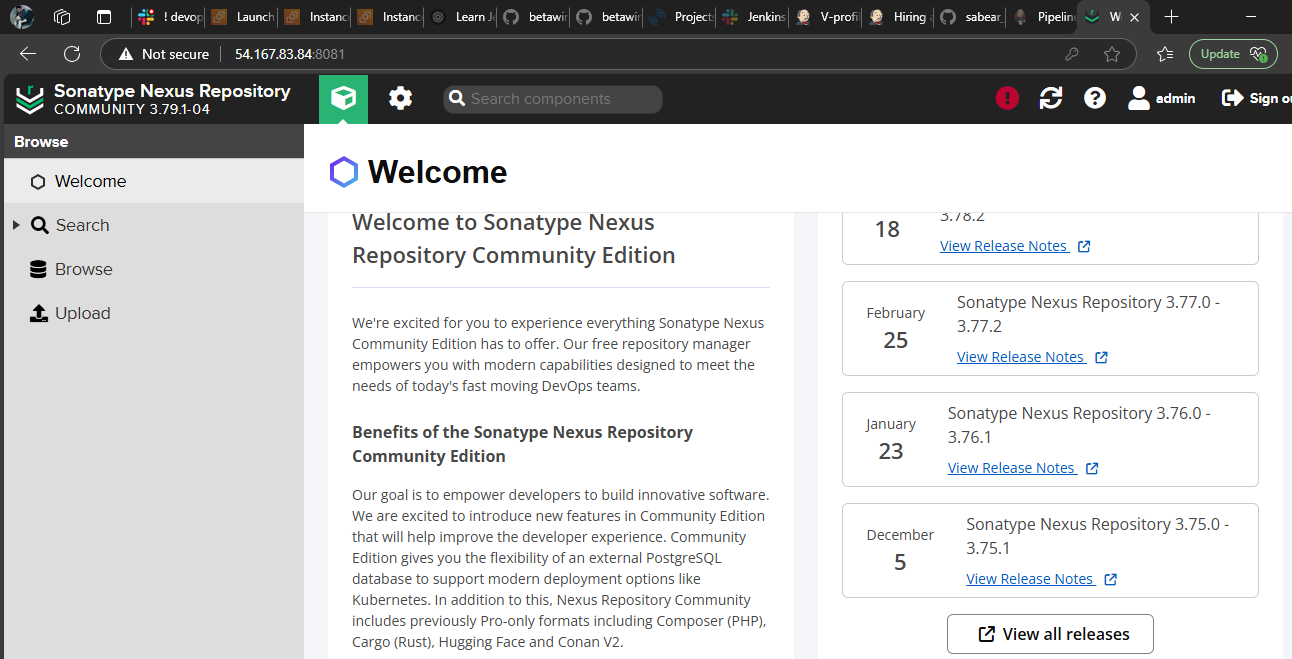
To list all images use docker ps

It will take few min.s to start nexus after that check on browser using public IP:8081 you will redirected nexus sign in page

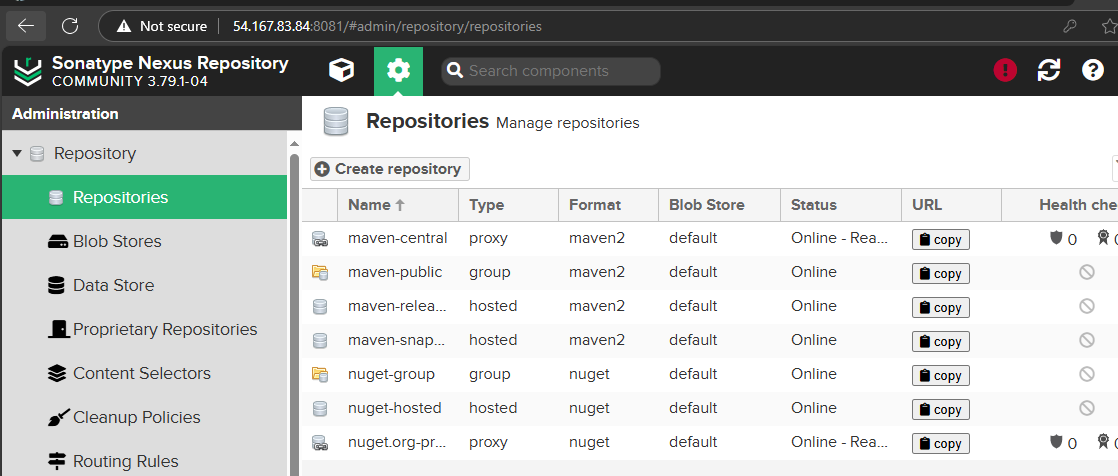
Use this command to get intialAdminPassword docker exec -it <container ID> cat /nexus-data/admin.password and default username is **admin**

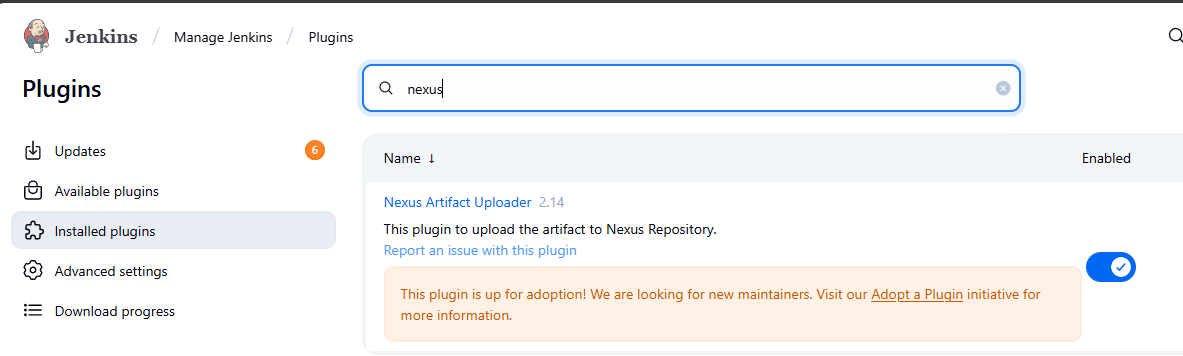


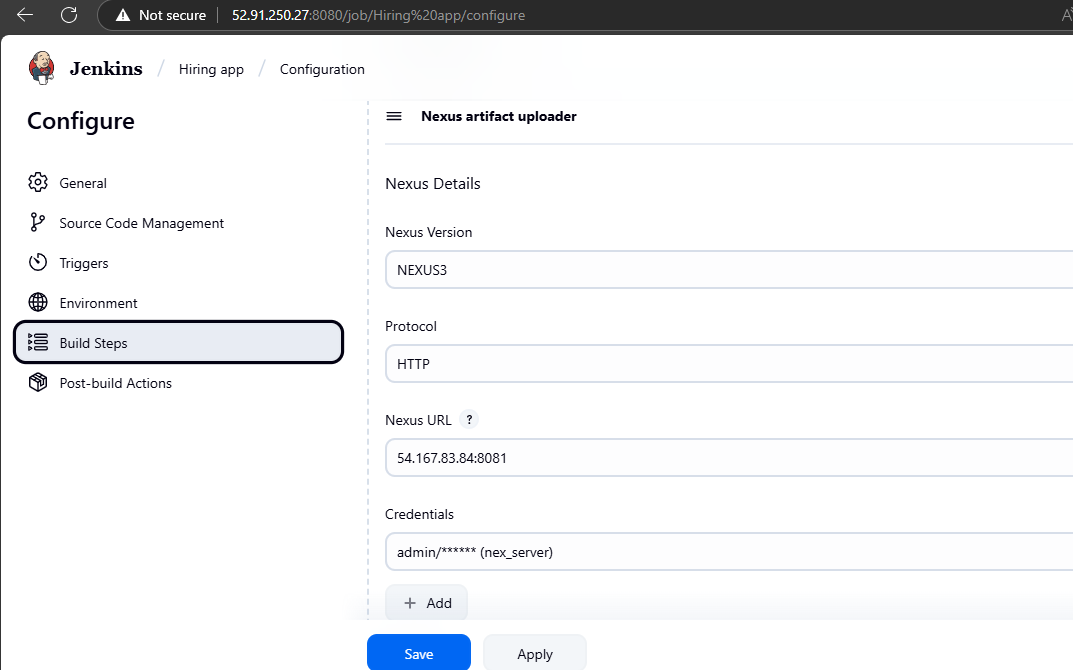
Sign and change password

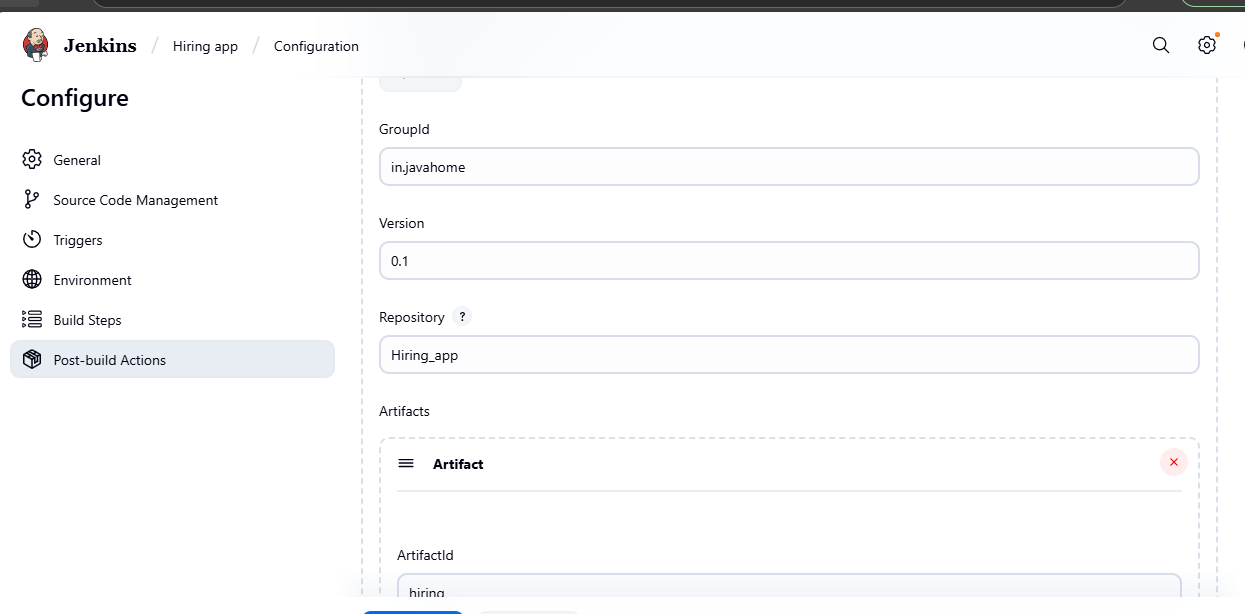
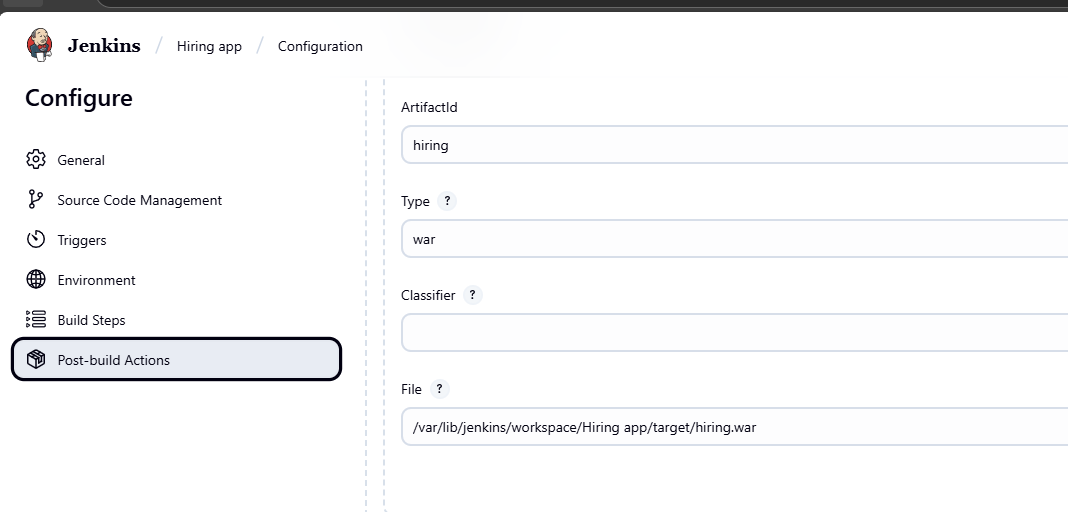


Settings > Repositories > Create repository > maven 2(hosted)



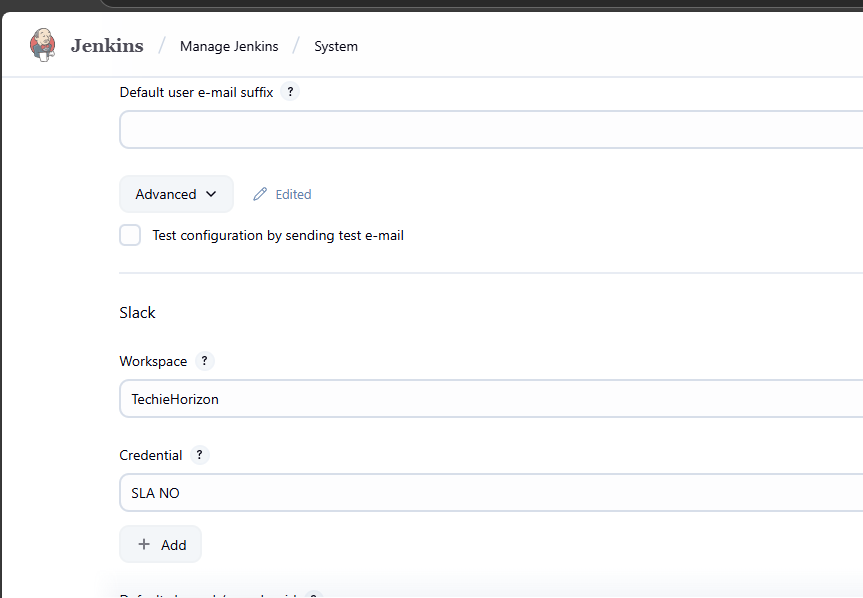
Dashboard > Manage jenkins > Plugins > Available plugins > search for nexus artificat and install >install without restartingGoto Job > Configire > Build steps > Add Build step > Choose Nexus artifact uploader

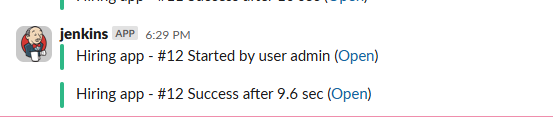
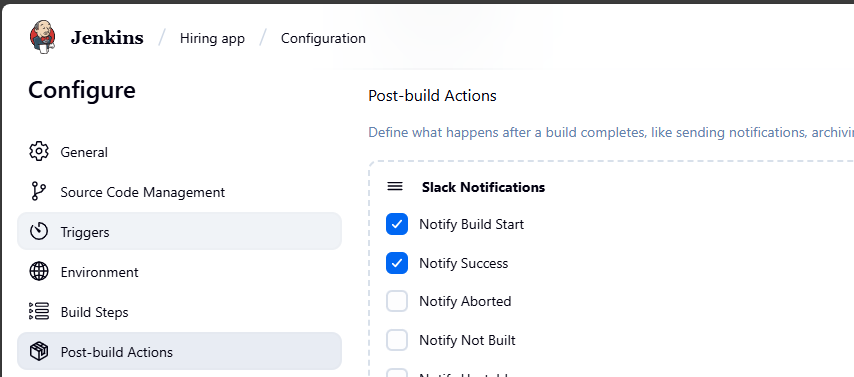
 In the credentials use Usename with passwords and give username as admin (default for nexus ) and password as (you have given for nexus )

below fields are Fill and Update from pom.xml 

Stage5:Slack notification

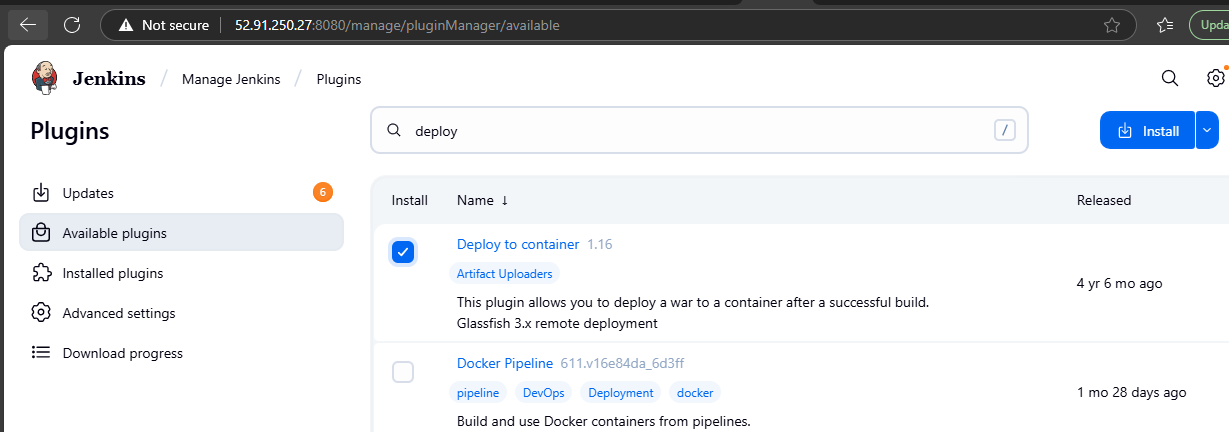
* Goto jenkins Dashboard > plugins > Available plugins > search for Slack notification > Install> Install without restarting
* Dashboard > manage Jenkins > System >Slack > give workspacce name and add credentials
* Goto slack > More > Apps > Jenkins CI > Configuration> Add to slack > You will get secret text
* Create credentials for Slack > use kind as secret text > Paste the text in the Secret > give ID,description and save
* Goto Job > Post build Actions > Enable required options > save



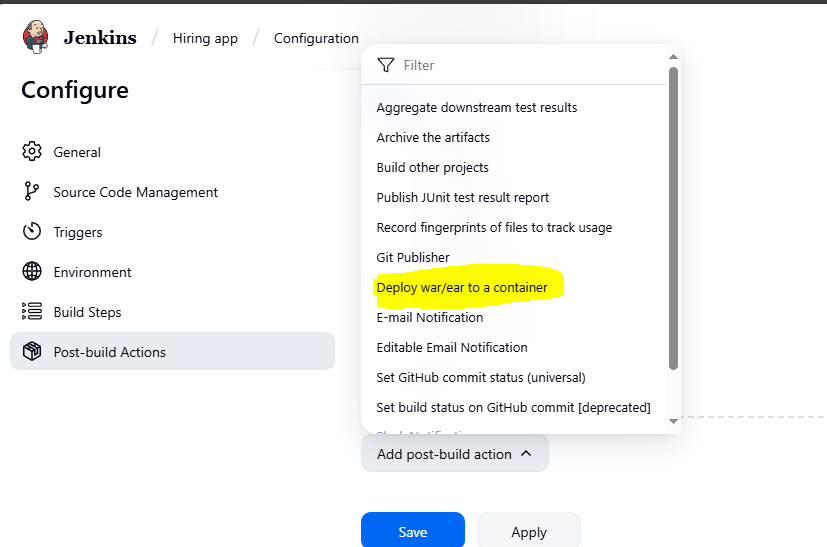


Stage6: Deploy to Tomcat

Dashboard > Manage Jenkins > Available plugins > search for Deploy to conatainer > Install > Install without restart



Dashboard > Manage Jenkins > Post Build Actions >choose Deploy war/ear to a conatiner



Launch an ec2 for Tomcat installation and connect to it.

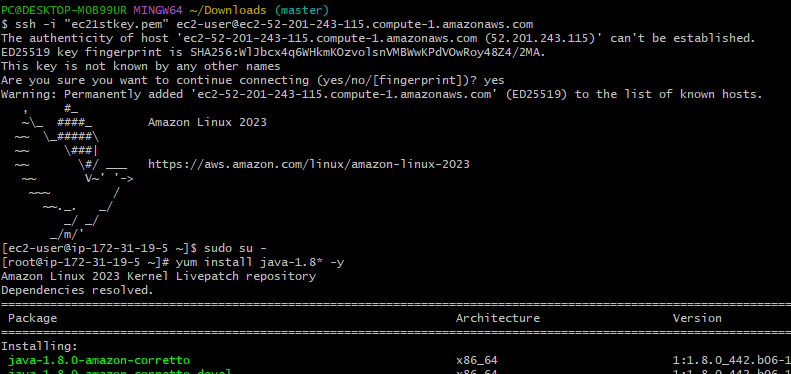
Switch to super user by using “Sudo su – ”

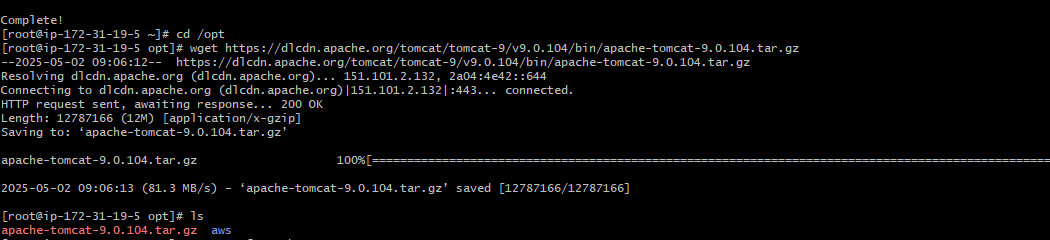
Install java 1.8 version as java is the prerequisite for Apache tomcat

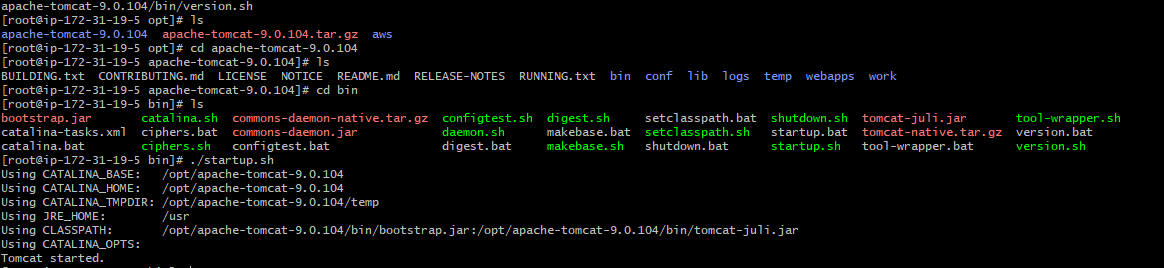
yum install java-1.8\* -y

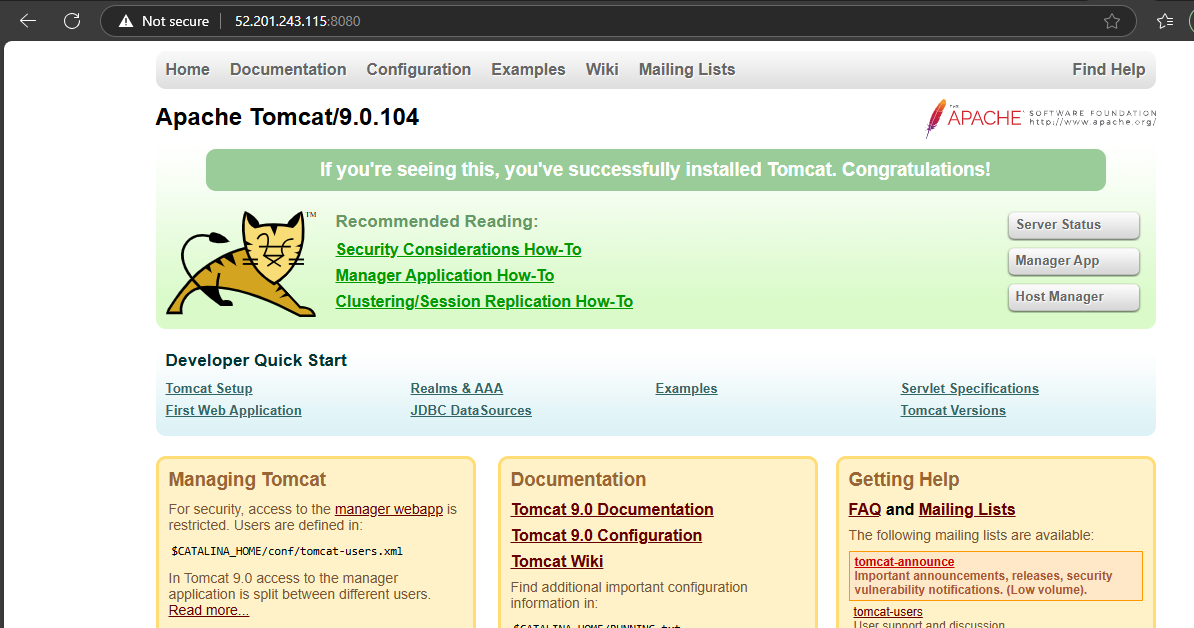
cd/opt > wget <https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.104/bin/apache-tomcat-9.0.104.tar.gz>

next untar the downloaded file > goto bin >run script statrtup.sh

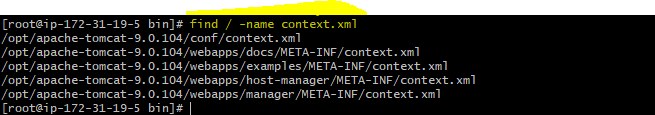




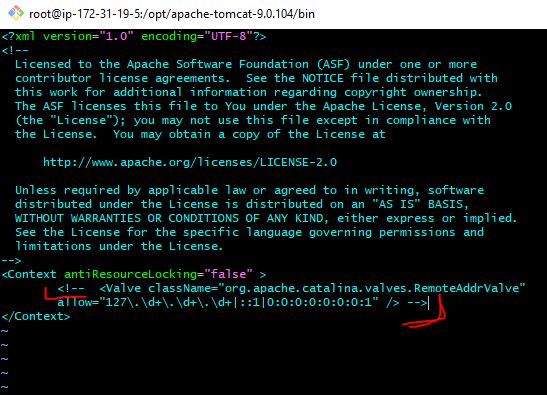




Goto *bin* directory search for context.xml by using find / -name conext.xml



Open and edit all 4 files with related to webapps

Add <!- - and - - > wherever you find ValveclassName 

Now goto conf directory and open tomcat–users.xml and update user info by adding <role rolename="manager-gui"/>

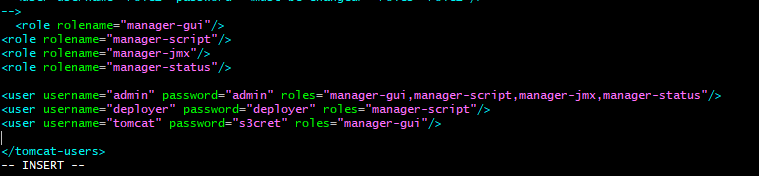
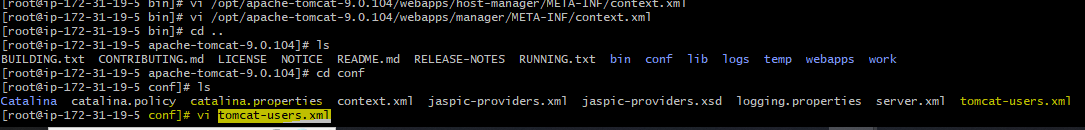
<role rolename="manager-script"/>

<role rolename="manager-jmx"/>

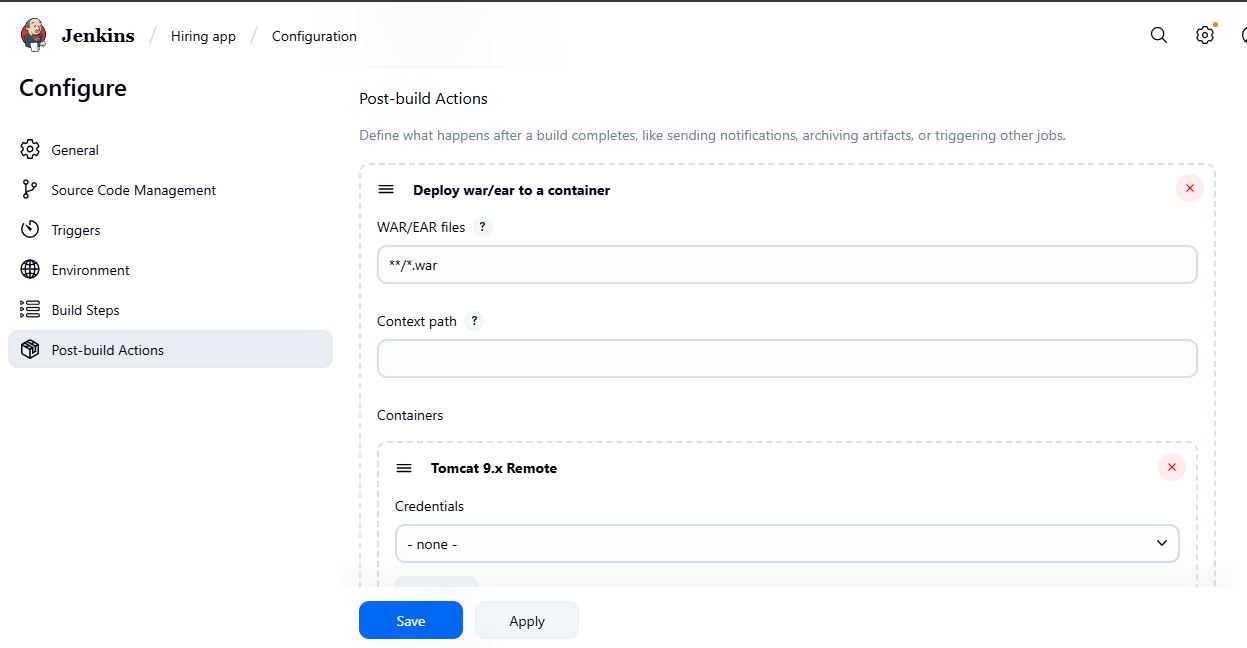
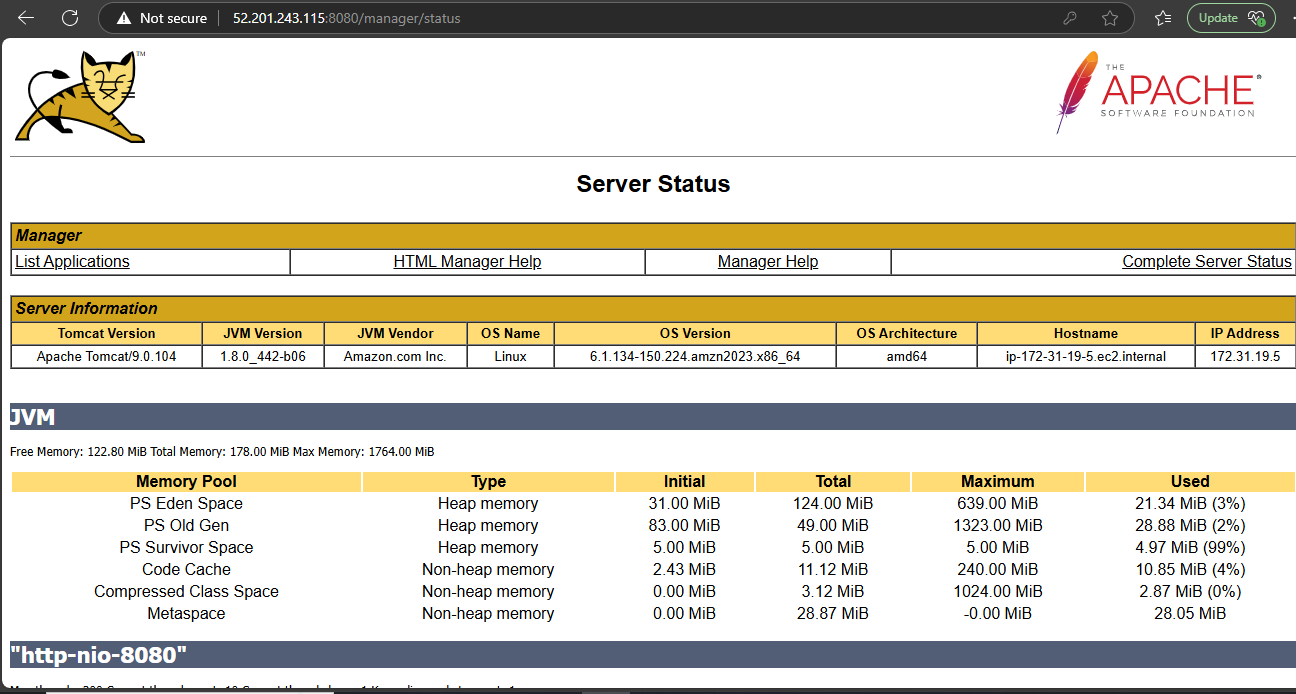
<role rolename="manager-status"/>

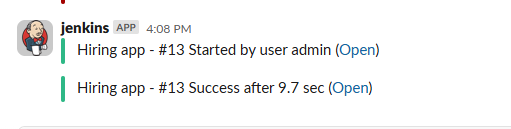
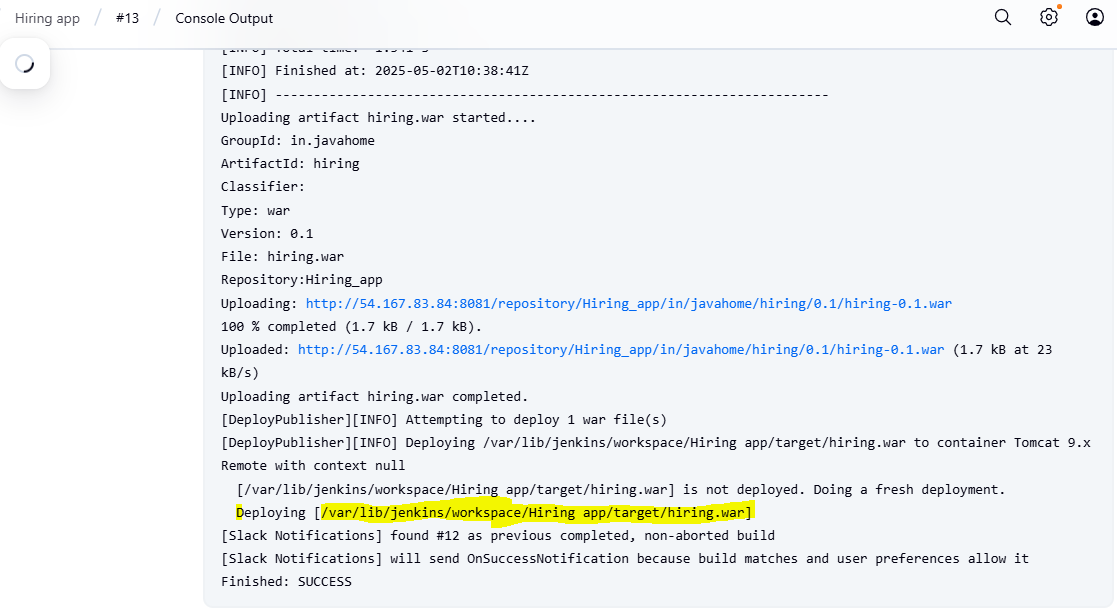
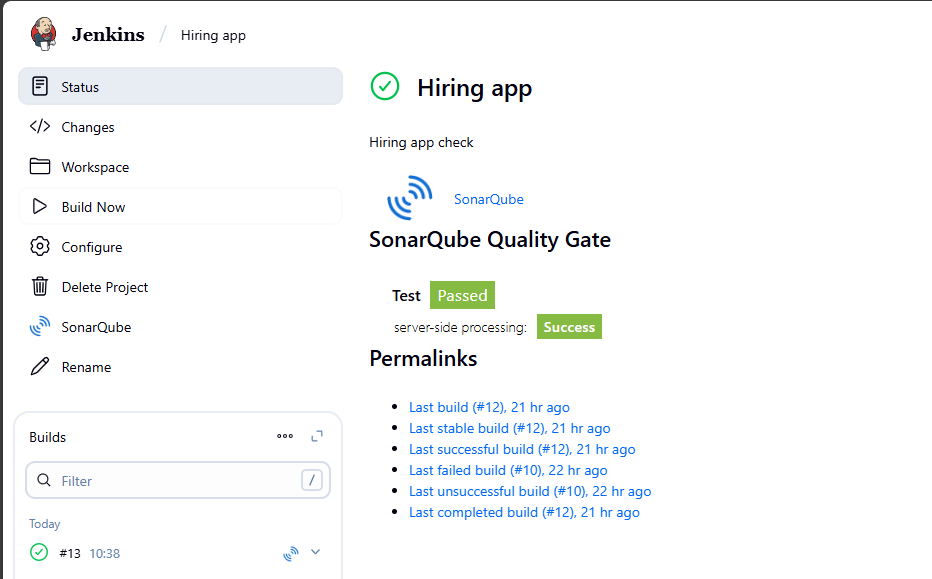
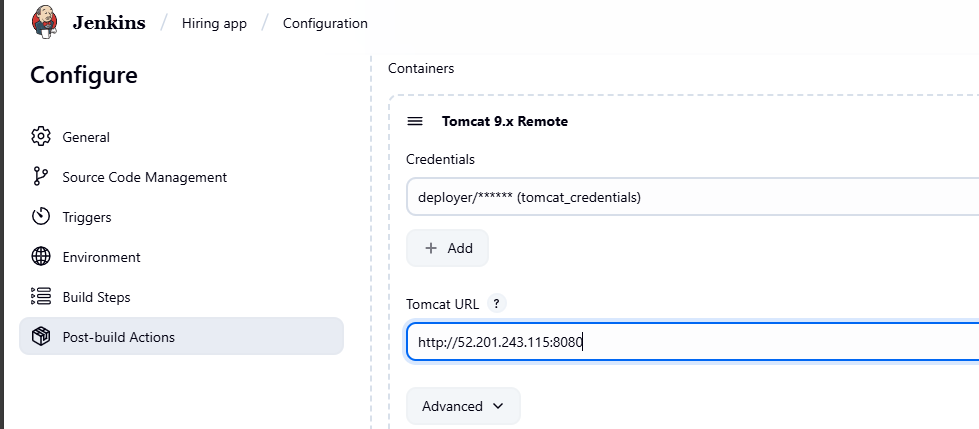
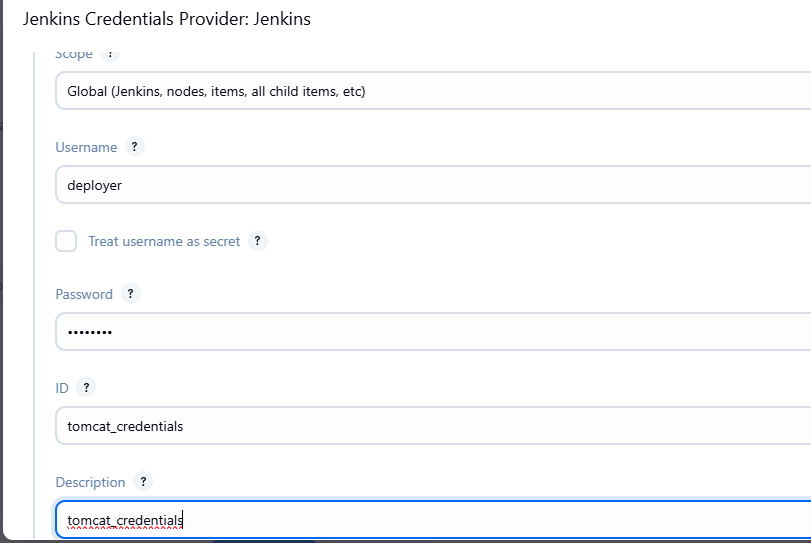
<user username="admin" password="admin" roles="manager-gui,manager-script,manager-jmx,manager-status"/>

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

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

*Now Restart Apache - tomcat* : cd bin > bash shutdown.sh and bash startup.sh

After that sigin to tomcat website: click on Server Status give *both* Username and password as deployer ,use these credentials for Jenkins ,save and click Build now



2) Setup a jenkins CICD pipeline using Declarative pipeline using feature-1.1 branch.

<https://github.com/betawins/sabear_simplecutomerapp/tree/feature-1.1>

stages:

1) Git Clone

2) Sonarqube Integration

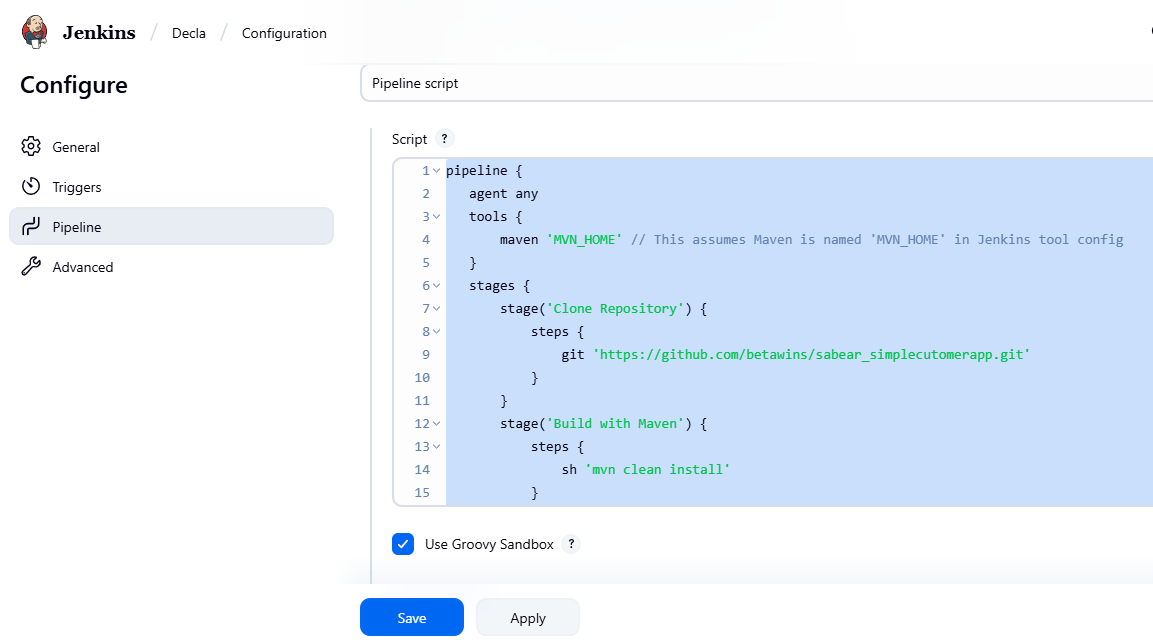
3) Maven Compilation

4) Nexus Artifactory

5) Slack Notification

6) Deploy On tomcat

# Dashboard > new item > give name > select pipeline



pipeline {

agent any

tools {

maven 'MVN\_HOME' // This assumes Maven is named 'MVN\_HOME' in Jenkins tool config

}

stages {

stage('Clone Repository') {

steps {

git 'https://github.com/betawins/sabear\_simplecutomerapp.git'

}

}

stage('Build with Maven') {

steps {

sh 'mvn clean install'

}

}

}

post {

success {

echo ':white\_tick: Build completed successfully!'

}

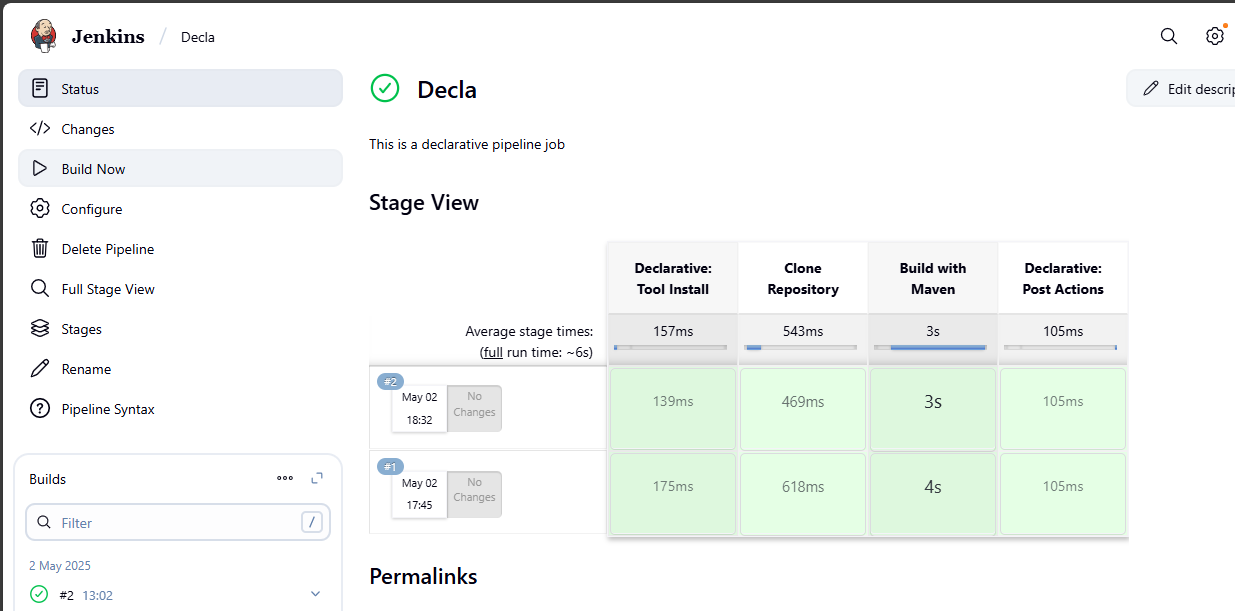
failure {

echo ':x: Build failed.'

}

}

}



3) Setup a jenkins CICD pipeline using Scripted pipeline using feature-1.1 branch.

<https://github.com/betawins/sabear_simplecutomerapp/tree/feature-1.1>

stages:

1) Git Clone

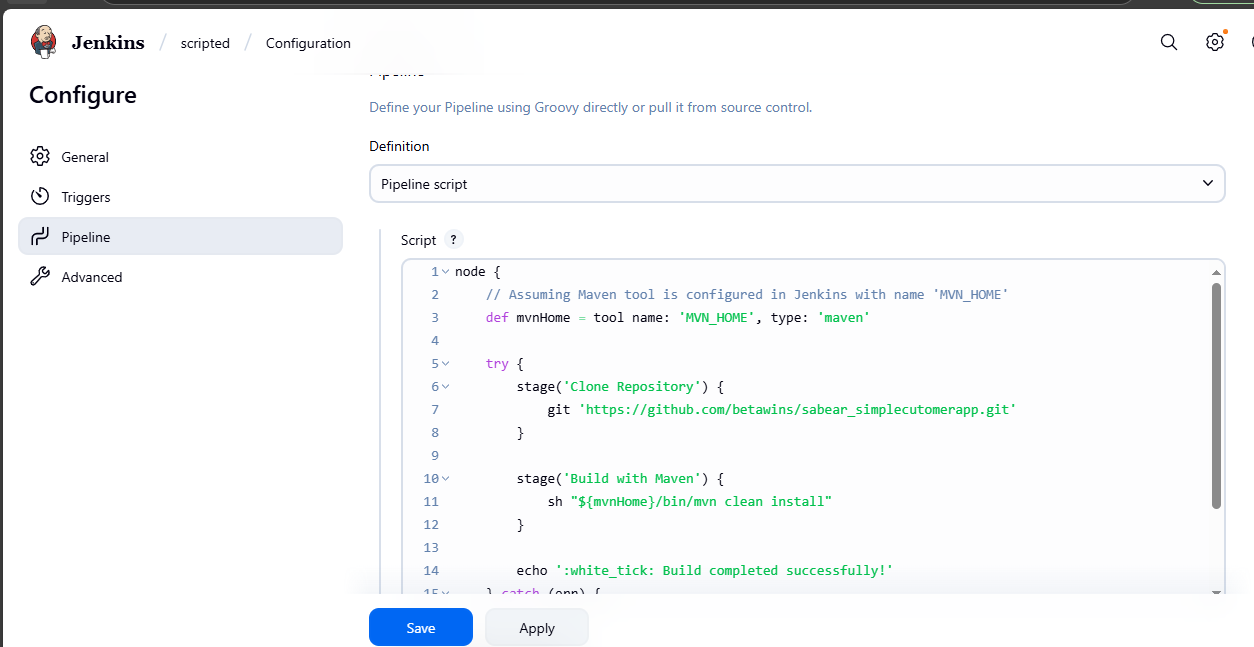
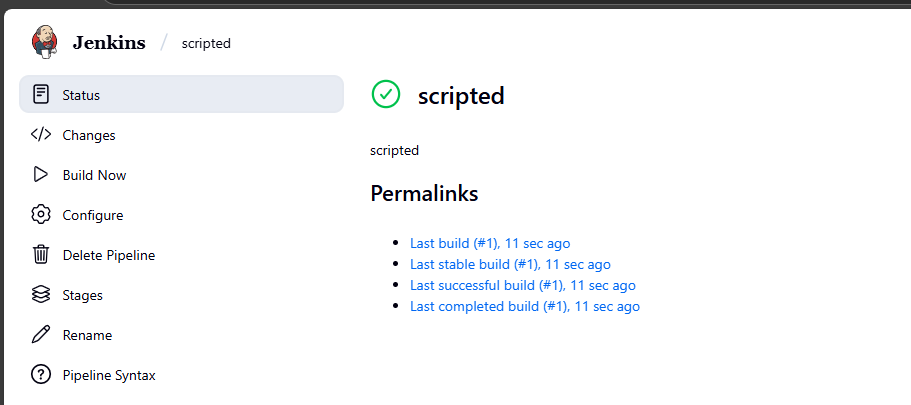
2) Sonarqube Integration

3) Maven Compilation

4) Nexus Artifactory

5) Slack Notification

6) Deploy On tomcat

4) Write sample skeleton of pipelines.

# Declarative pipeline:

pipeline {

agent {label ”gnAN”}

Stages{

Stage(“Code”){

Steps{}

echo “This is cloning the code”

}

stage(“Build”){

Steps{}

echo “This is building the code”

}

stage(“Test”){

Steps{}

echo “This is Testing the code”

}

stage(“Deploy”){

Steps{}

echo “This is Deploying the code”

}

}

}

# Scripted pipeline:

node('gnAN') {

stage('Code') {

echo 'This is cloning the code'

}

stage('Build') {

echo 'This is building the code'

}

stage('Test') {

echo 'This is Testing the code'

}

stage('Deploy') {

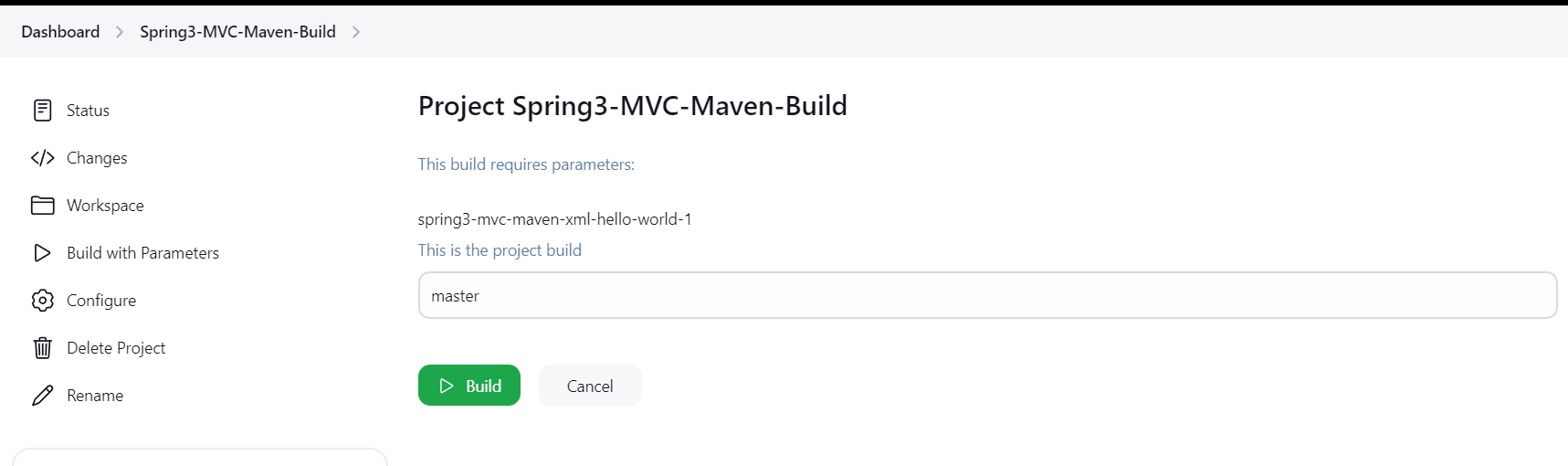
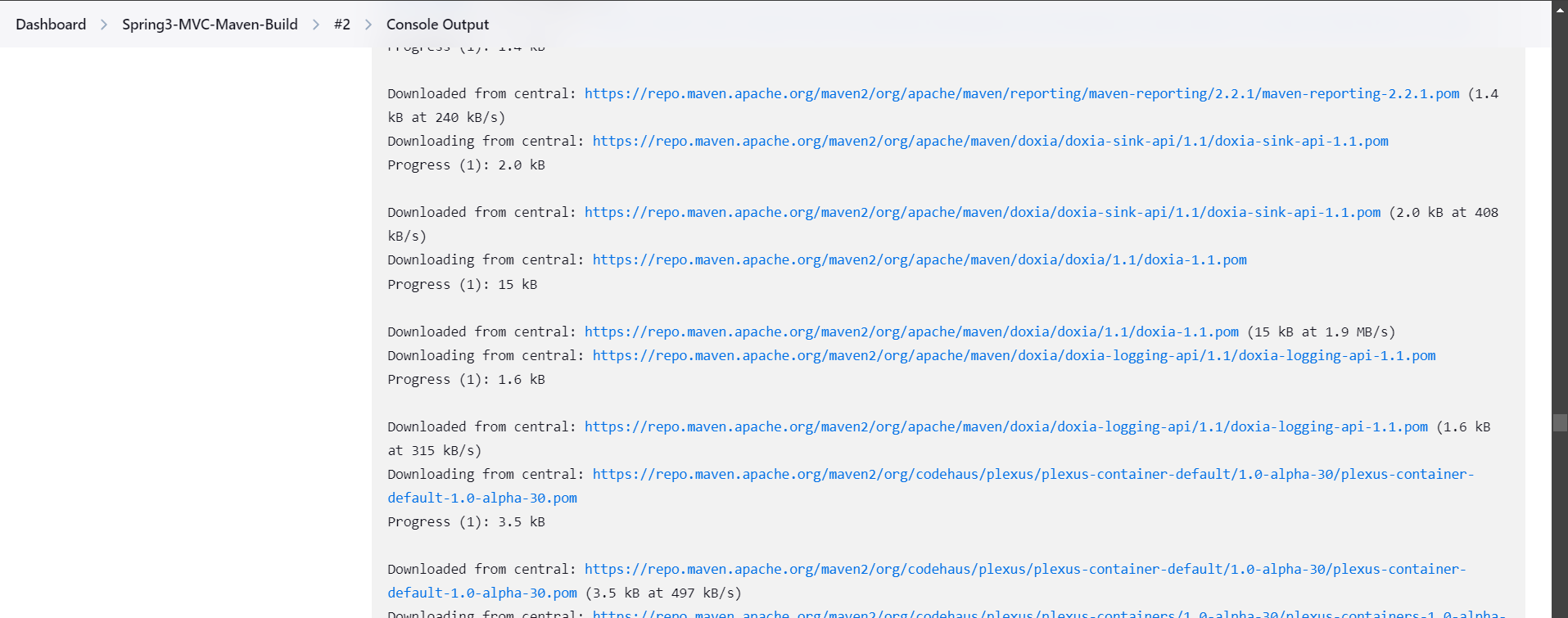
echo 'This is Deploying the code'

}

}

5) Create a parametirized job in jenkins.

<https://github.com/betawins/spring3-mvc-maven-xml-hello-world-1.git>

6) Setup one slave machine for jenkins.

GitHubGitHub

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

Contribute to betawins/hiring-app development by creating an account on GitHub. (33 kB)

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

GitHubGitHub

[GitHub - betawins/sabear\_simplecutomerapp at feature-1.1](https://github.com/betawins/sabear_simplecutomerapp/tree/feature-1.1)

Contribute to betawins/sabear\_simplecutomerapp development by creating an account on GitHub. (35 kB)

<https://github.com/betawins/sabear_simplecutomerapp/tree/feature-1.1>

GitHubGitHub

[GitHub - betawins/spring3-mvc-maven-xml-hello-world-1: Maven + Spring 3 MVC hello world example (XML)](https://github.com/betawins/spring3-mvc-maven-xml-hello-world-1.git)

Maven + Spring 3 MVC hello world example (XML). Contribute to betawins/spring3-mvc-maven-xml-hello-world-1 development by creating an account on GitHub. (53 kB)

[https](https://github.com/betawins/spring3-mvc-maven-xml-hello-world-1.git)

