**Setup jenkins CICD pipeline using freestyle job**

**2) Setup jenkins job using freestyle job using below code.**

**https://github.com/betawins/Techie\_horizon\_Login\_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**
2. **Sonarqube Integration**

Prerequisite to Install jenkins is **java**.

Download jenkins Repo:

=====================

sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo

Import the jenkins key:

======================

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

Update ec2:

==========

sudo yum upgrade

Add required dependencies for the jenkins package:

=================================================

sudo amazon-linux-extras install java-openjdk11

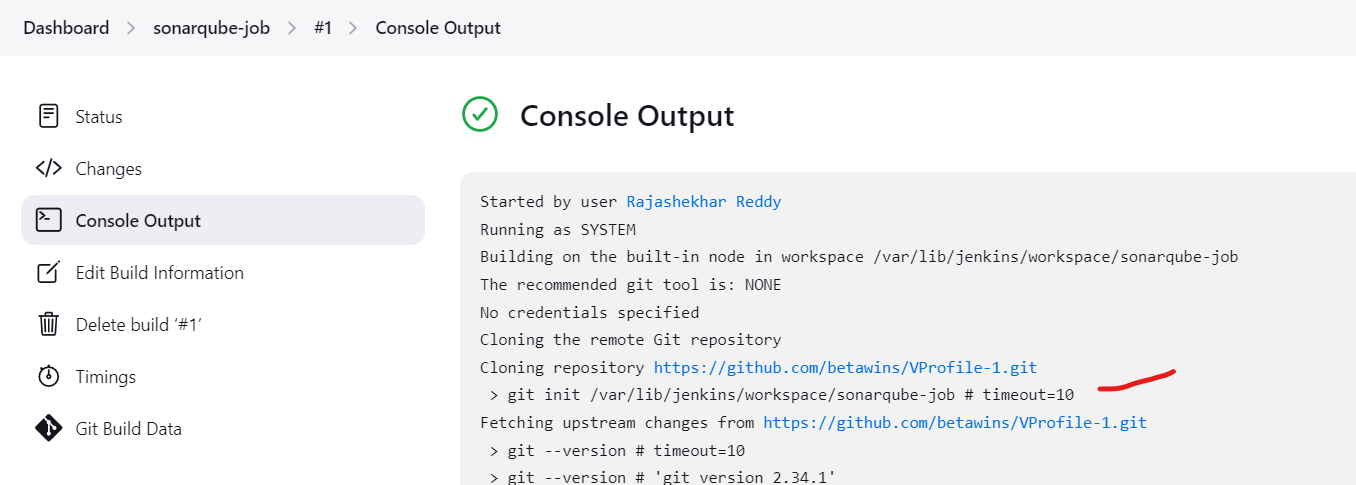
Install Jenkins:

===============

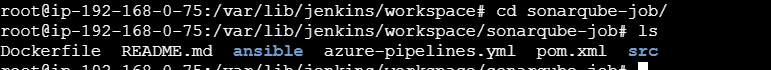
sudo yum install jenkins

\* First Access to my Jenkins-master Dashboard then I created new job (sonarqube-job) in added a git URL

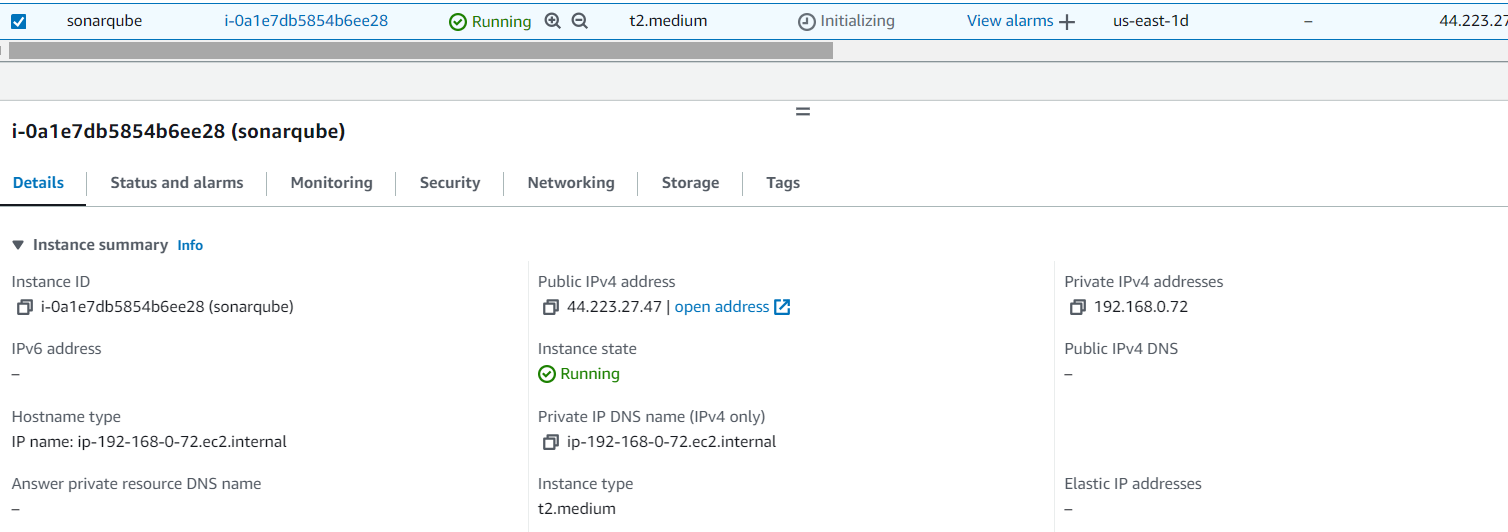
Url: “https://github.com/betawins/VProfile-1.git”



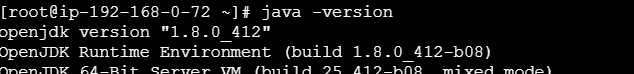
\* Once created the job build it, after build the job workspace will created



\* First created a new instance with t2.medium and with storage of 20 GB



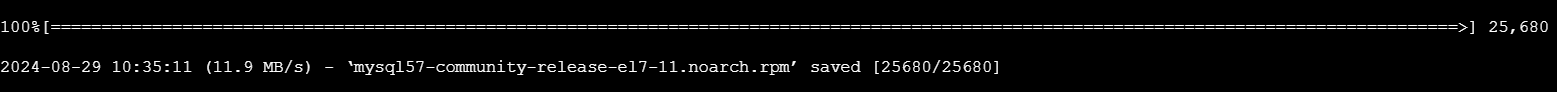
\* install Java using “ yum install java-1.8\*



\* Download mysql package using

sudo wget https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm

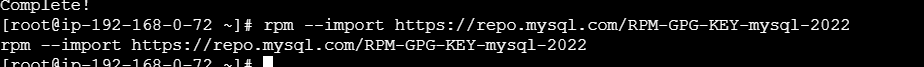
sudo yum localinstall mysql57-community-release-el7-11.noarch.rpm





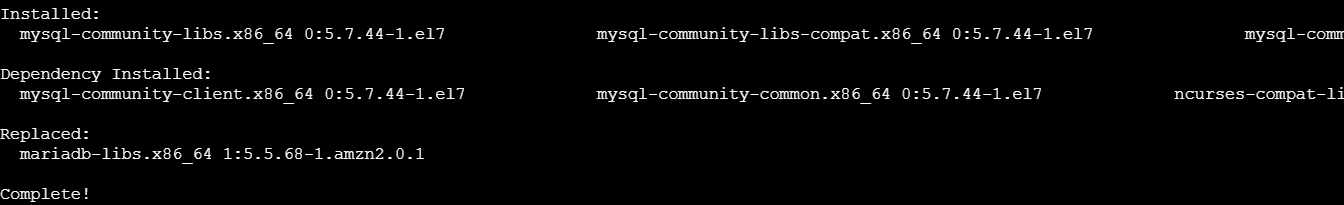
\* Import the key using

rpm --import <https://repo.mysql.com/RPM-GPG-KEY-mysql-2022>



\* Install the mysql community-server

sudo yum install mysql-community-server



\* Then start the mysql using

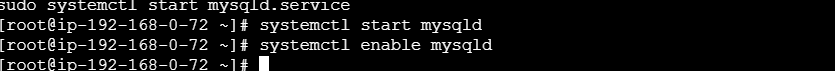
sudo systemctl start mysqld.service



Start MySQL and Enable Start at Boot Time using below commands

systemctl start mysqld

systemctl enable mysqld



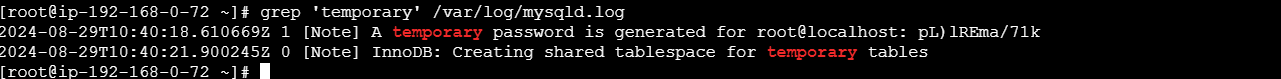
\* I have checked that mysql port is running or not using netstat -na | grep 3306



Configure the MySQL Root Password

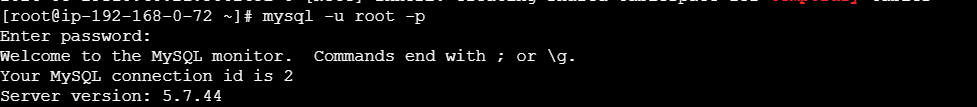
We will see default MySQL root password

grep 'temporary' /var/log/mysqld.log



Login to mysql using the default password

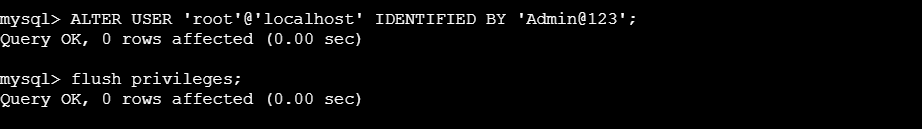
mysql -u root -p



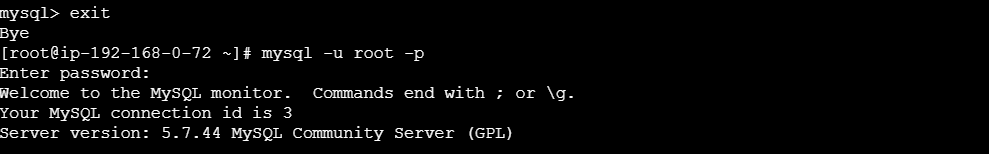
Now replace the default password with a new and strong password

ALTER USER 'root'@'localhost' IDENTIFIED BY 'Admin@123';

flush privileges;



Tested Using new password , alogged in with new password

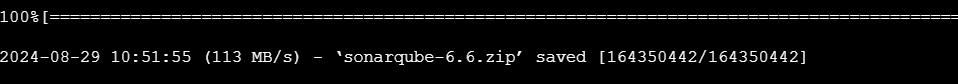


Download & unzip SonarQube 6.0

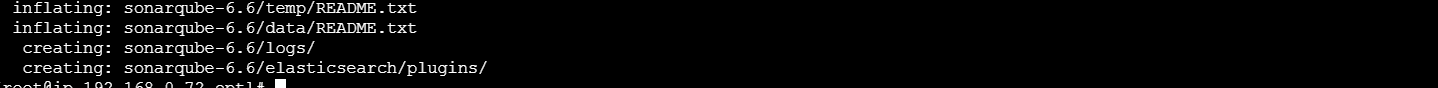
\* change to “**/opt”**

Then Download sonarQube

wget <https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-6.6.zip>

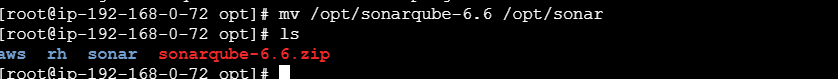


unzip sonarqube-6.6.zip



\* Rename the folder name

mv /opt/sonarqube-6.6 /opt/sonar



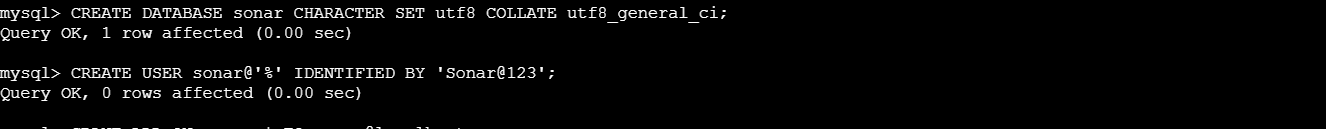
Login to mysql

mysql -u root -p

\* Create a local and a remote user

CREATE DATABASE sonar CHARACTER SET utf8 COLLATE utf8\_general\_ci;

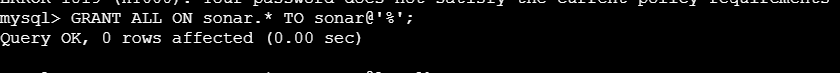
CREATE USER sonar@'%' IDENTIFIED BY 'Sonar@123';



Grant database access permissions to users

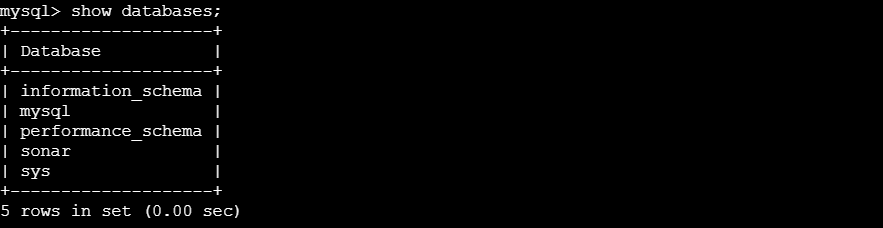
GRANT ALL ON sonar.\* TO sonar@localhost;

GRANT ALL ON sonar.\* TO sonar@'%';

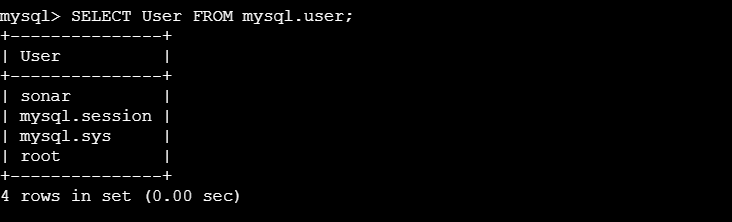


**check users and databases**

show databases;



SELECT User FROM mysql.user;



FLUSH PRIVILEGES;



**#ON EC2-Instance**

Edit sonar properties file to uncomment and provide required information for below properties.

- File Name: /opt/sonar/conf/sonar.properties

- sonar.jdbc.username=`sonar`

- sonar.jdbc.password=`Sonar@123`

- sonar.jdbc.url=jdbc:mysql://`localhost:3306`/sonar?useUnicode=true&characterEncoding=utf8&rewriteBatchedStatements=true&useConfigs=maxPerformance&useSSL=false

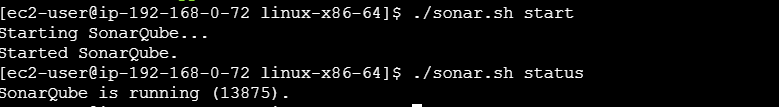
- sonar.web.host=`0.0.0.0`

- sonar.web.context=`/sonar

Start SonarQube service

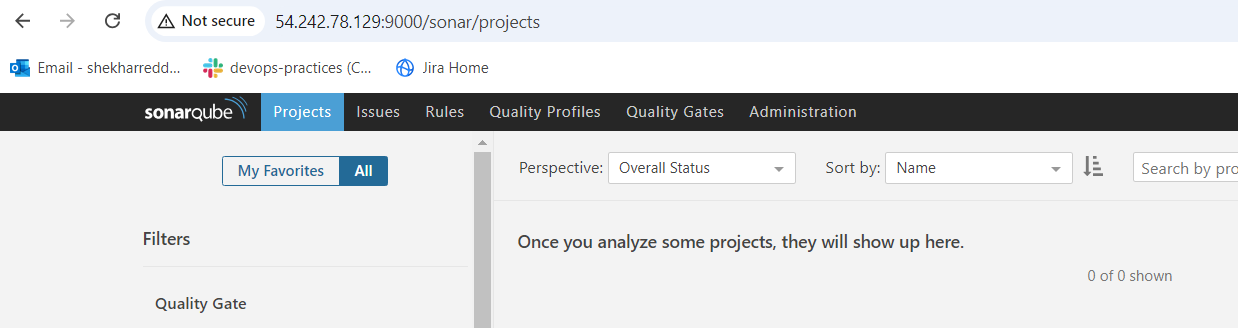
# cd /opt/sonar/bin/linux-x86-64/

# ./sonar.sh start

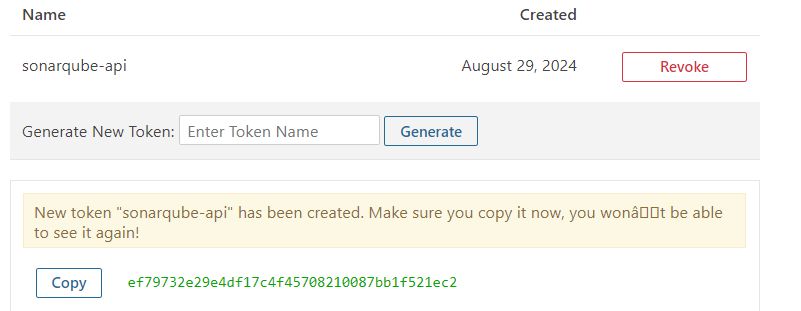


Logged into sonarqube

http://54.242.78.129:9000/sonar

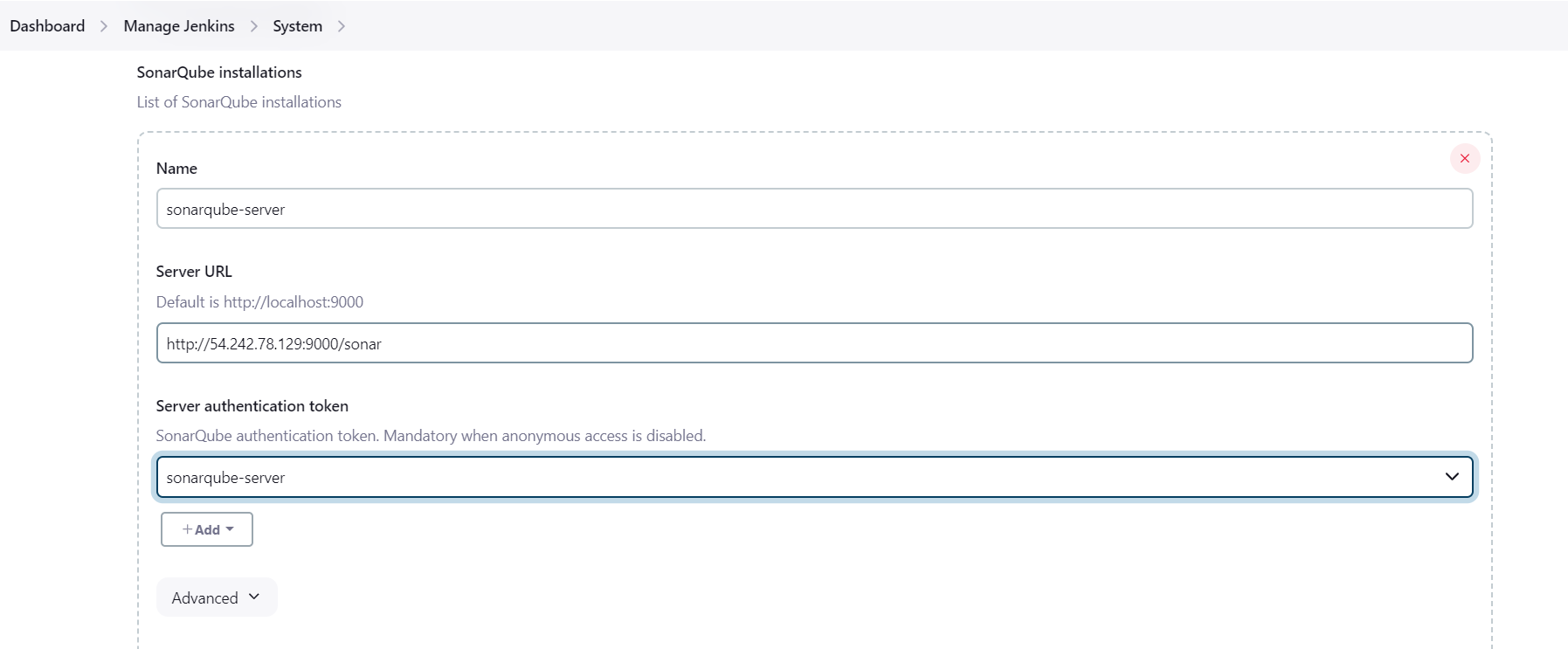


Then create a token in sonarqube



\* To add the Global credentials of Sonarqube

\* Jenkins [Dashboard](http://54.165.95.20:8081/)→[Manage Jenkins](http://54.165.95.20:8081/manage/)→[Credentials](http://54.165.95.20:8081/manage/credentials/)→[System](http://54.165.95.20:8081/manage/credentials/store/system/)→[Global credentials (unrestricted)](http://54.165.95.20:8081/manage/credentials/store/system/domain/_/)



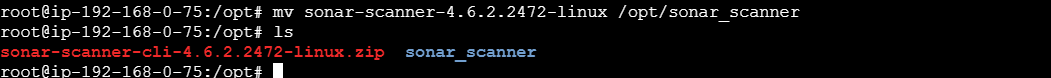
\* Download the sonarqube-scanner package at “opt” folder in jenkins-server

wget https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-4.6.2.2472-linux.zip

unzip sonar-scanner-cli-4.6.2.2472-linux.zip



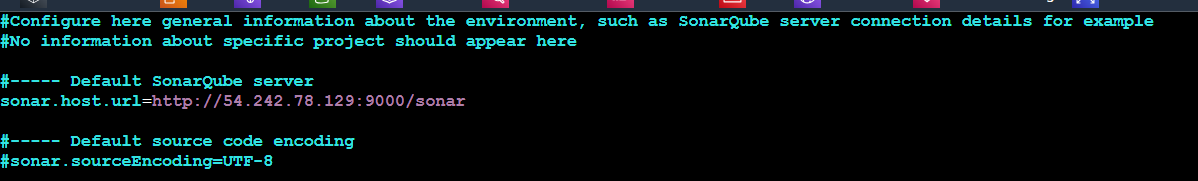
\* Renamed mv sonar-scanner-4.6.2.2472-linux /opt/sonar\_scanner



Set SonarQube server details in sonar-scanner property file

Sonar properties file: /opt/sonar\_scanner/conf/sonar-scanner.properties

- sonar.host.url=http://`<SONAR\_SERVER\_IP>`:9000/sonar

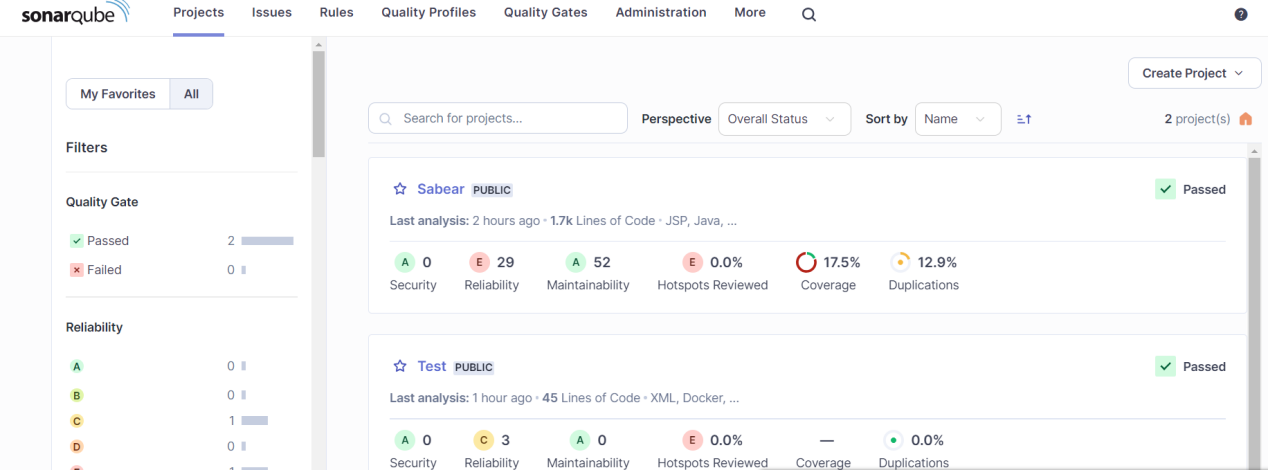


Configuring SonarQube scanner home path

**Manage Jenkins` > `Global Tool Configuration` > `SonarQube Scanner`**

**-** Name  **: `sonar\_scanner`**

**-** SONAR\_RUNNER\_HOME **: `/opt/sonar\_scanner/bin`**



**3)Maven Compilation**

sudo apt install fontconfig openjdk-17-jre

#How to install maven:

1. Change to opt direcotry

cd /opt

2) Download maven

Wget https://mirrors.estointernet.in/apache/maven/maven-3/3.6.3/binaries/apache-maven-3.6.3-bin.tar.gz

3) Extract the maven tar file

tar xvf apache-maven-3.6.3

1. Export maven homepath in “.bash\_profile”

# .bash\_profile

# Get the aliases and functions

if [ -f ~/.bashrc ]; then

. ~/.bashrc

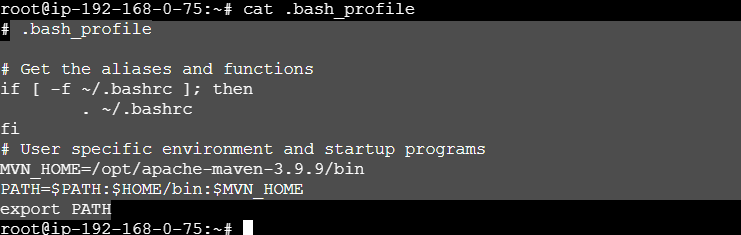
fi

# User specific environment and startup programs

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

PATH=$PATH:$HOME/bin:$MVN\_HOME

export PATH

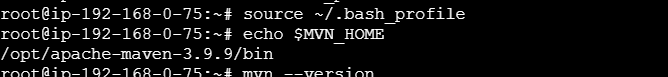


5) Restart the .bash\_profile

source ~/.bash\_profile

6) Check for MVN\_HOME

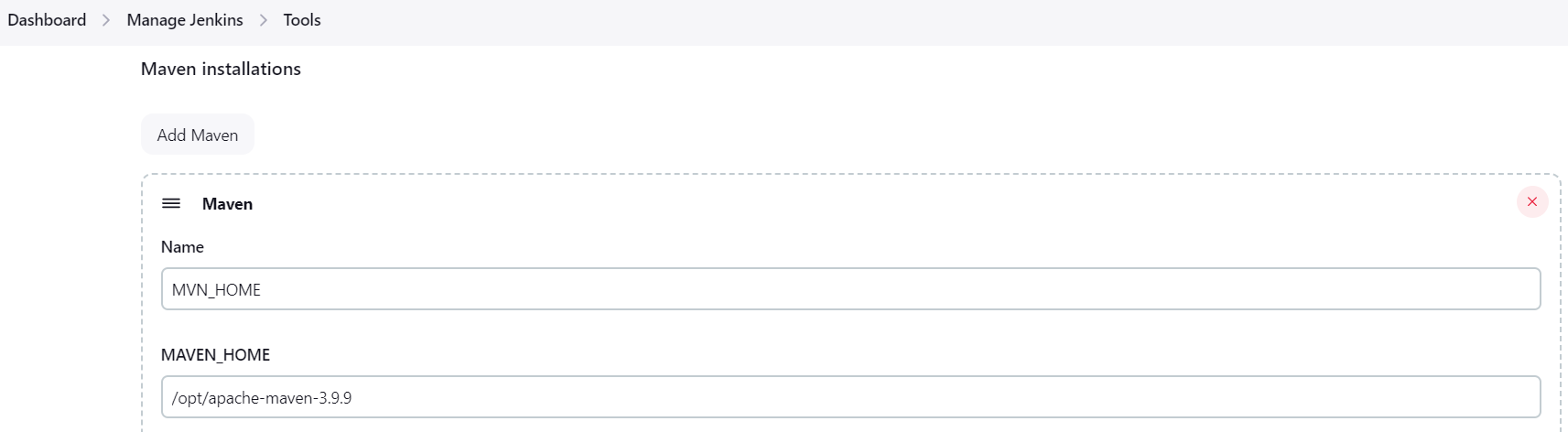
echo $MVN\_HOME

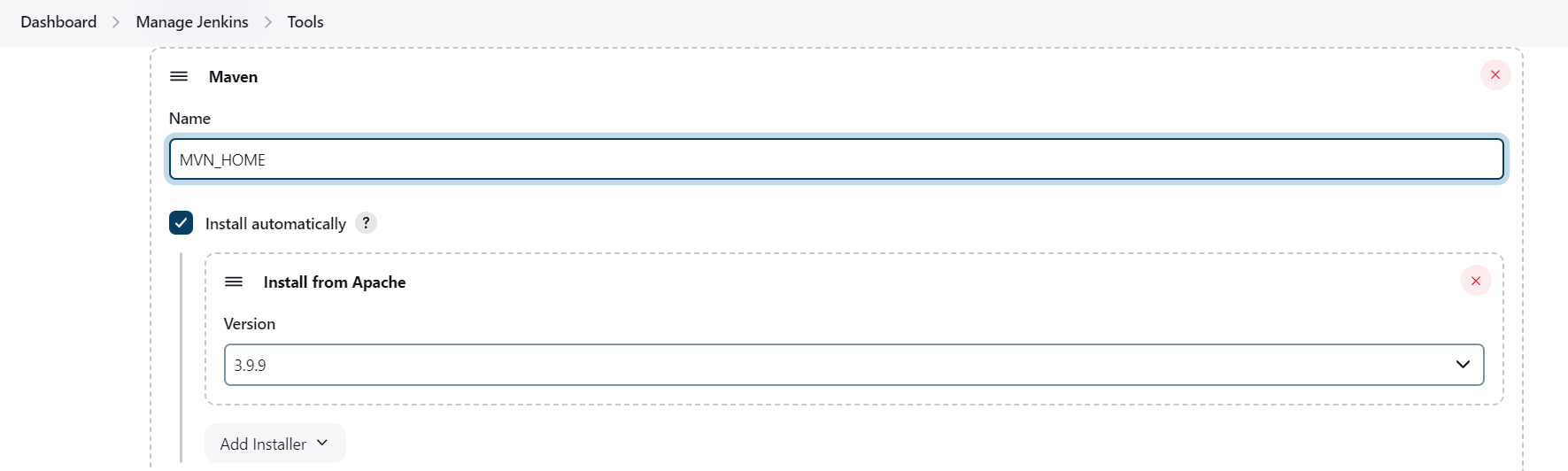


\* To check the version of maven “ **mvn --version”**

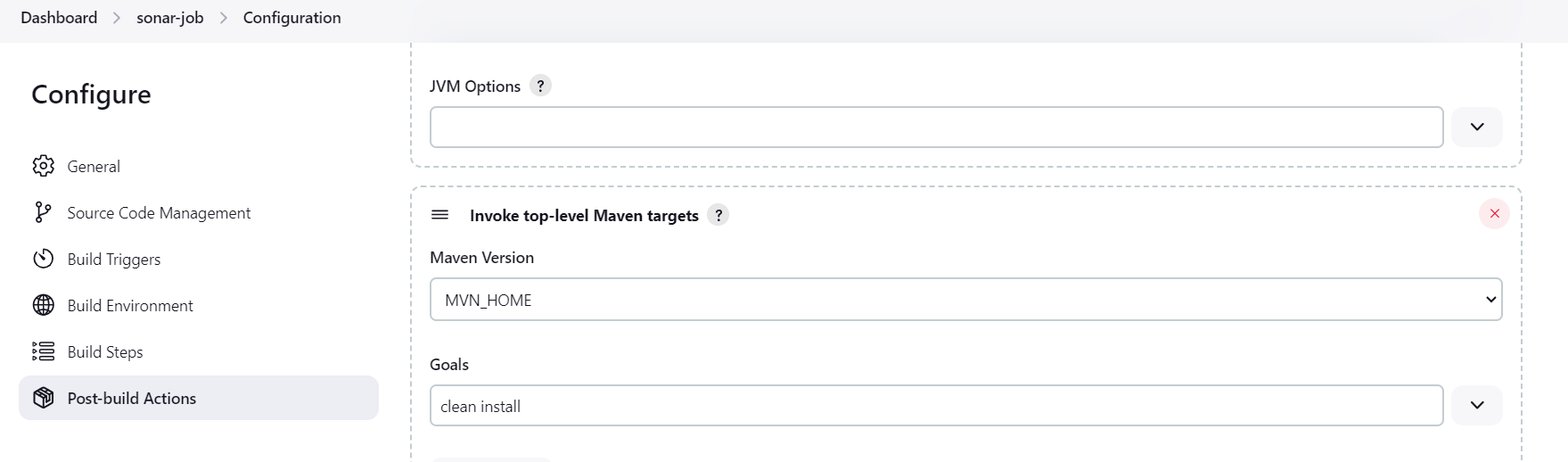


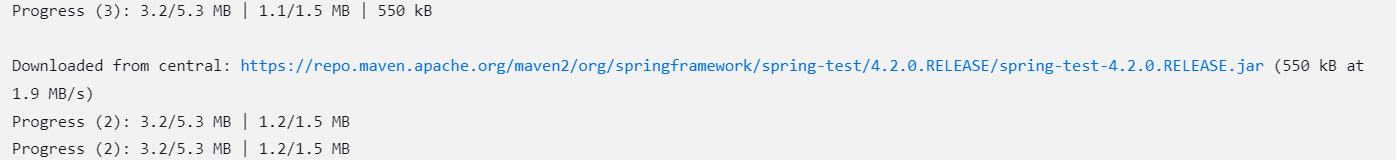
·Go to Jenkins [Dashboard](http://54.90.95.135:8081/)→ [Manage Jenkins](http://54.90.95.135:8081/manage/)→ [Tools](http://54.90.95.135:8081/manage/configureTools/) to integrate to the jenkins



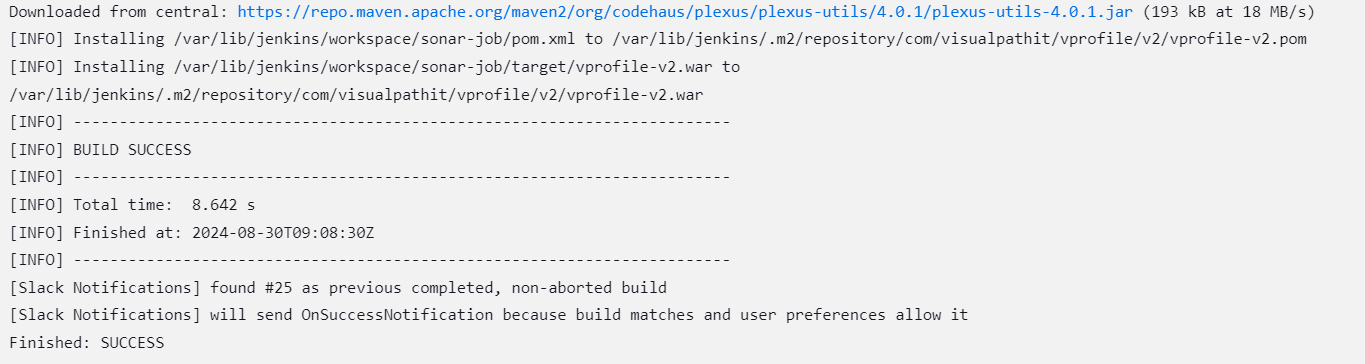


\* configured the job to save the packages in the given path

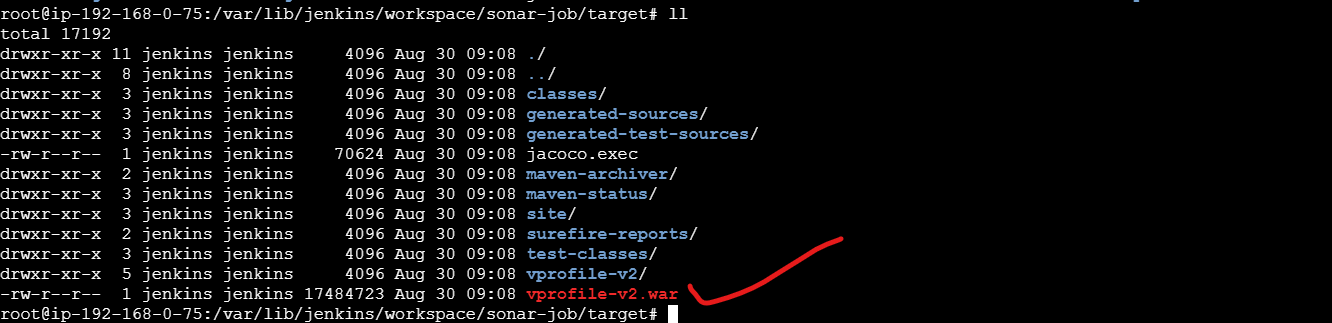




After completion of maven integration build successfully executed



Once complete the maven compilation then only TARGET folder will create, where this war packages will store



\* we can find a folder”.**m2**” , in this repository file will downloaded all the dependencies of maven



1. **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

```

As a good security practice, it is not advised to run nexus service as root. so create new user called nexus and grant sudo access to manage nexus services

sudo adduser nexus

# visudo \\ nexus ALL=(ALL) NOPASSWD: ALL

sudo chown -R nexus:nexus /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

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

Add nexus as a service at boot time

sudo ln -s /opt/nexus/bin/nexus /etc/init.d/nexus

Login as a nexus user and start service

su - nexus

service nexus start

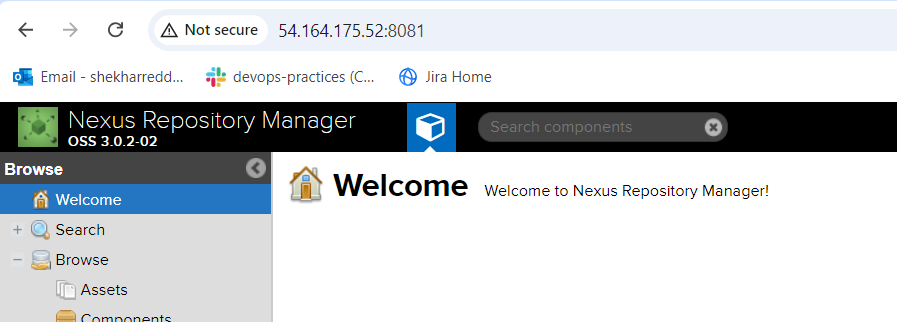
Login nexus server from browser on port 8081

http://<Nexus\_server>:8081

Use default credentials to login

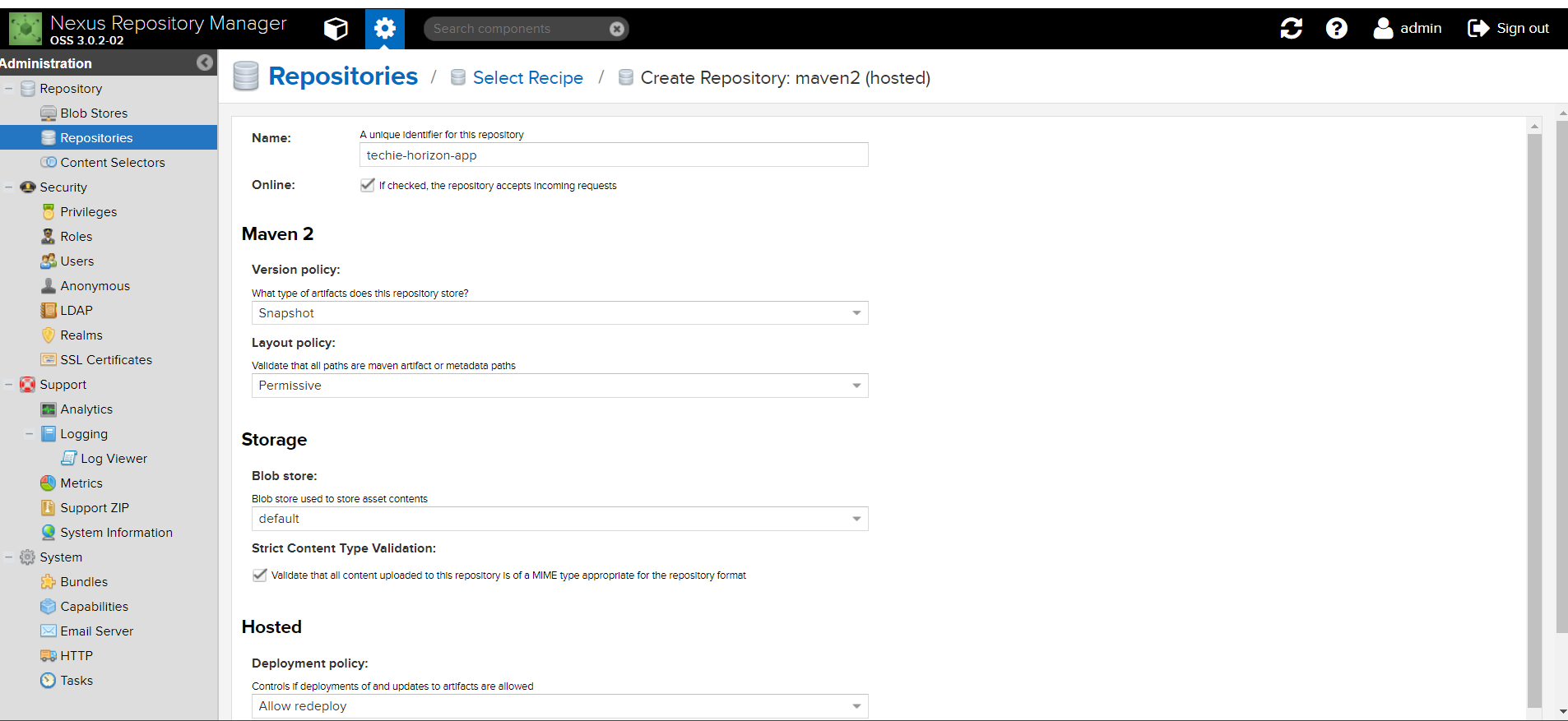
username : admin

password : admin123

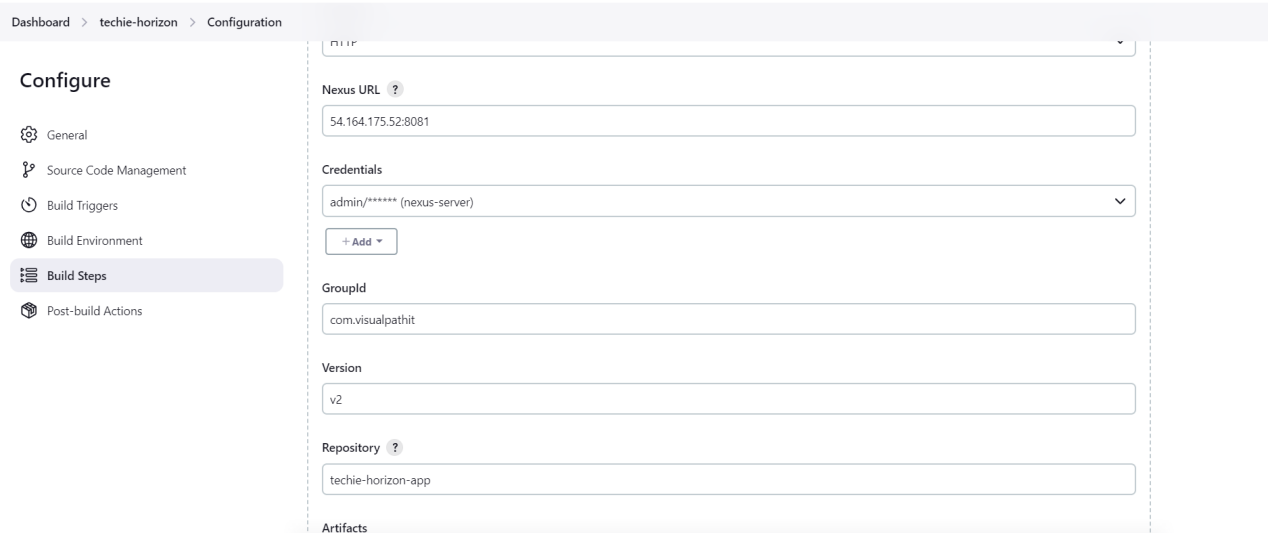


**\*** Create a repository in maven

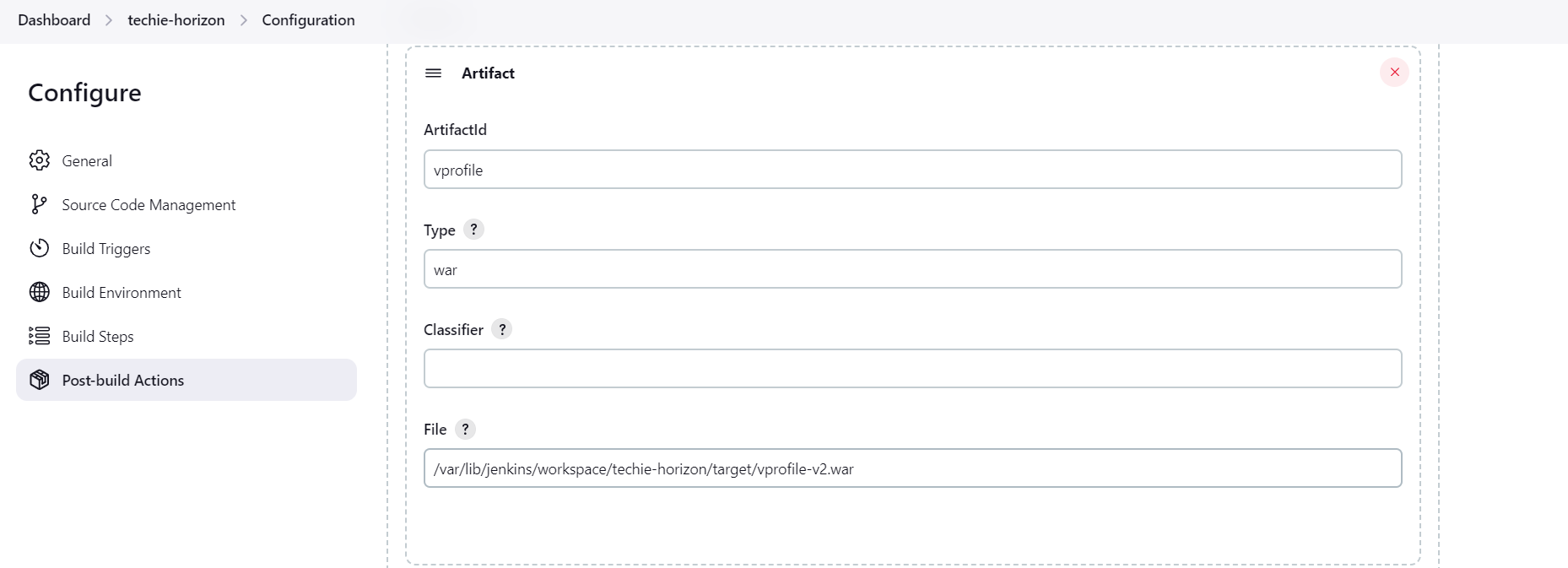
Go to settings→Go to repository→click on create repository



\* Configured credentials of nexus server

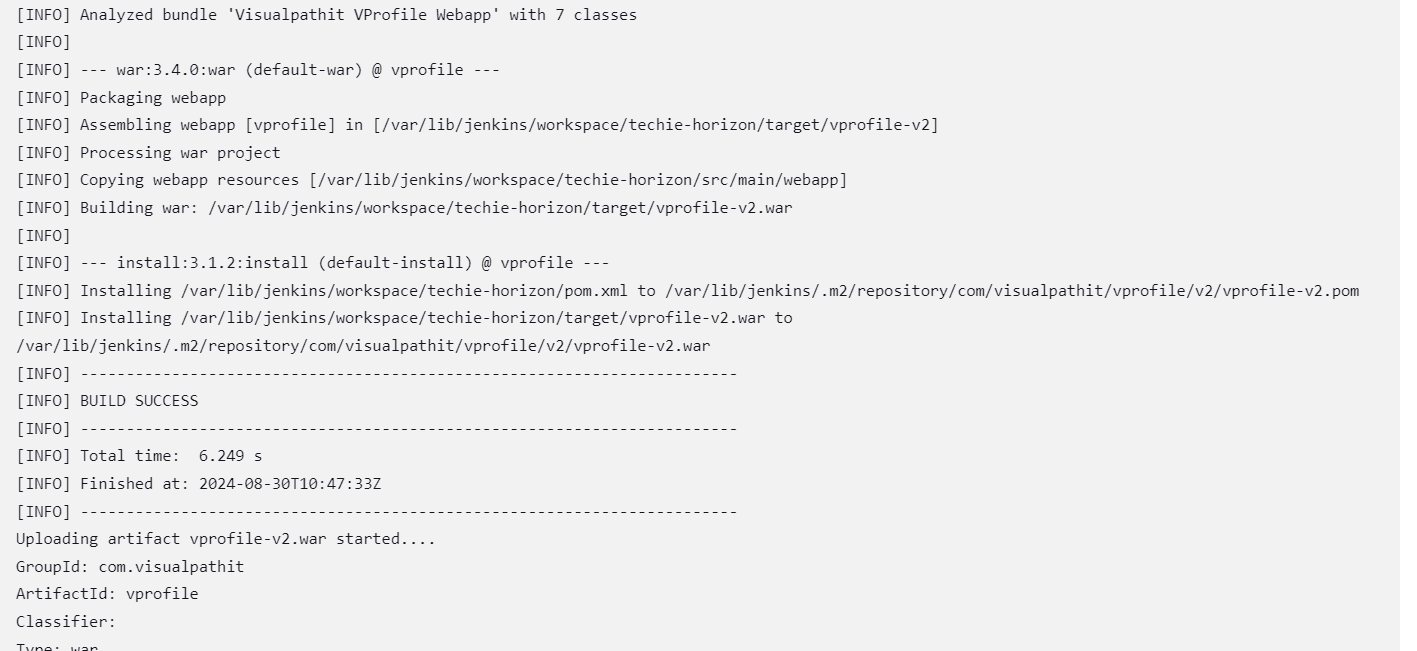


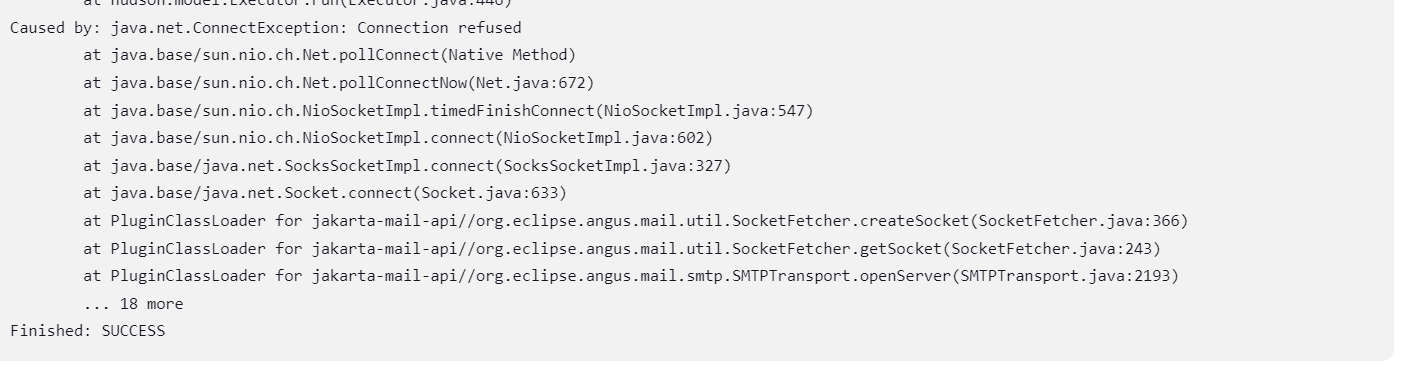
\* we can add artifactory details to store the source code packages in nexus artifactory, which are configured in jenkins using github url(https://github.com/betawins/Techie\_horizon\_Login\_app.git), which are provided by developer



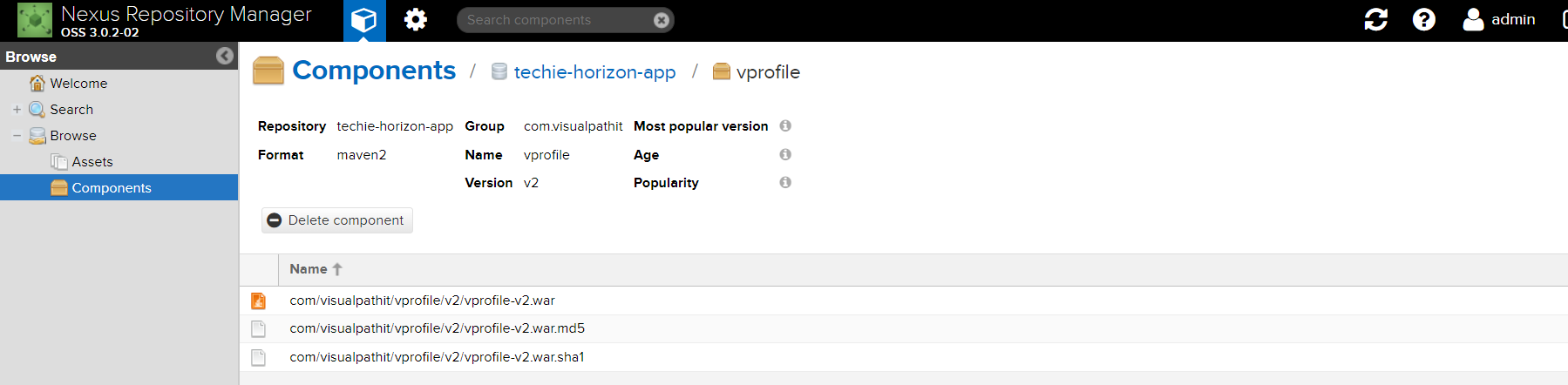
\* Now build the job for confirmation



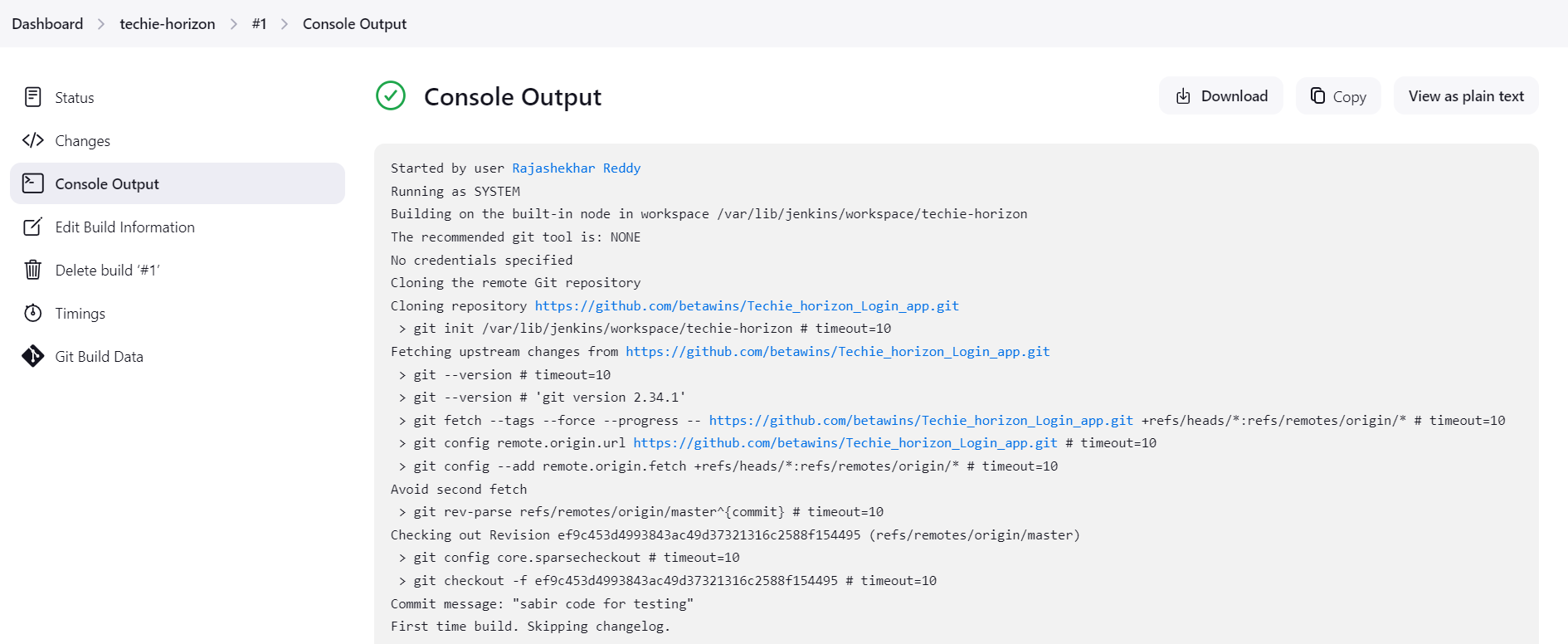


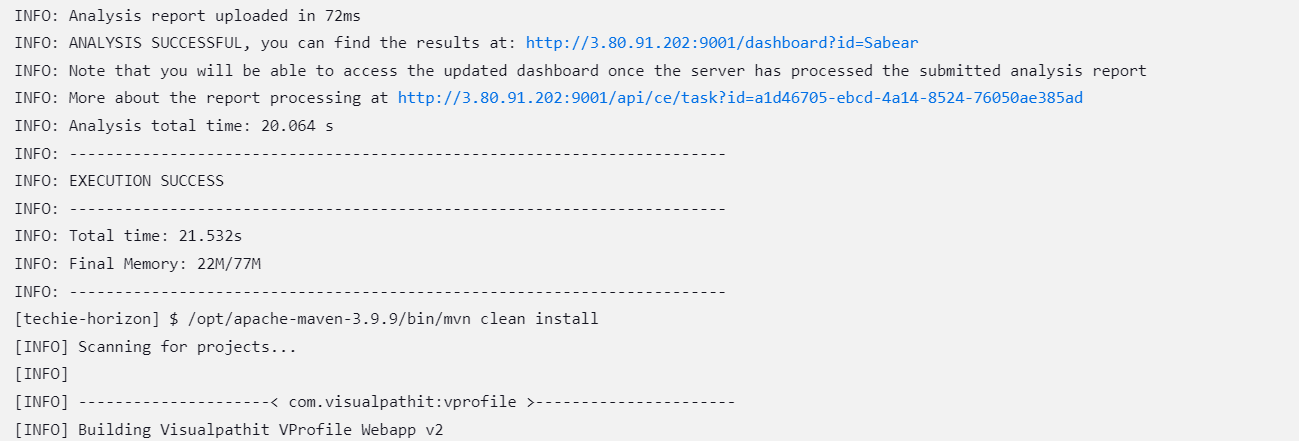


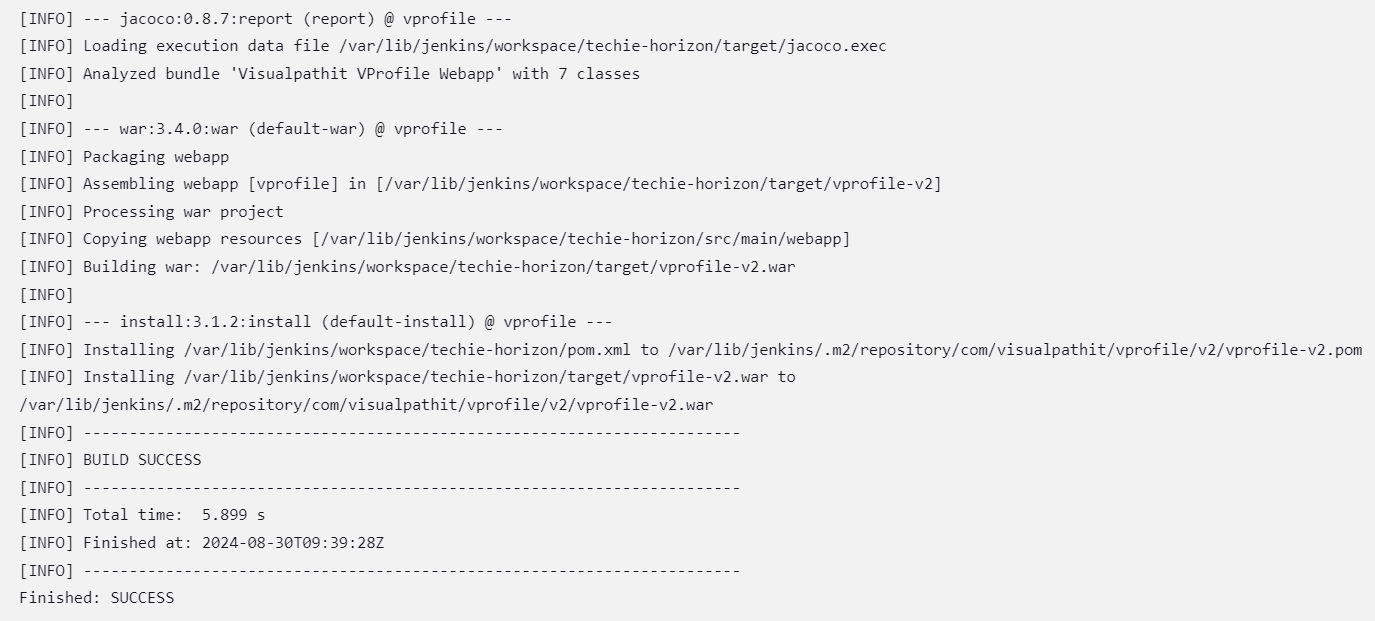
\*Now we can find the versions of the packages

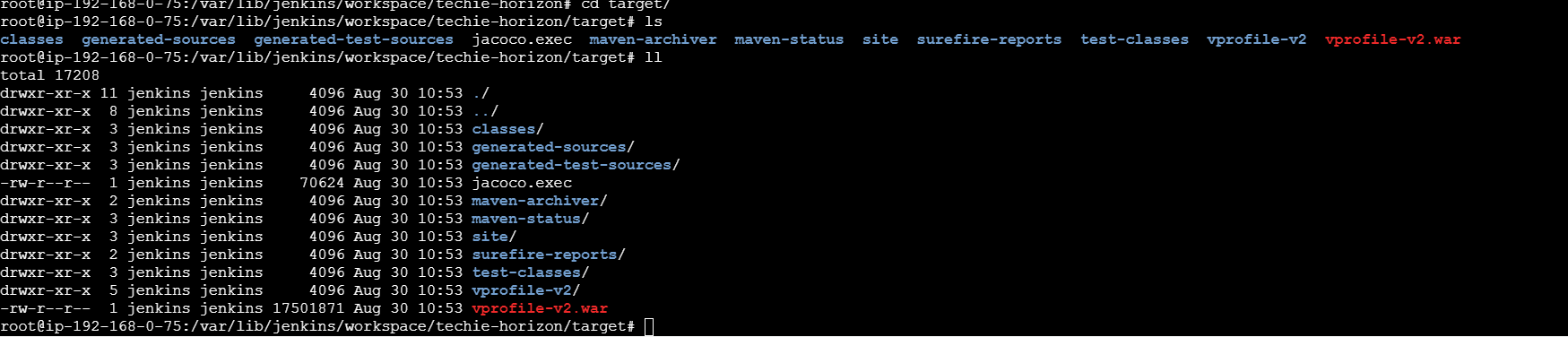






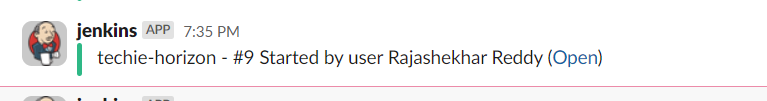






1. **Slack Notification**

We can find the slack



**6)Deployment of Tomcat**

Install a plugin called **deploy a container** in jenkins.

Create an instance and install java with 1.8 version

Sudo yum -y install java-1.8\*

Go to /opt/ folder then Download tomcat

<https://downloads.apache.org/tomcat/tomcat-9/v9.0.93/bin/apache-tomcat-9.0.93.tar.gz.asc>

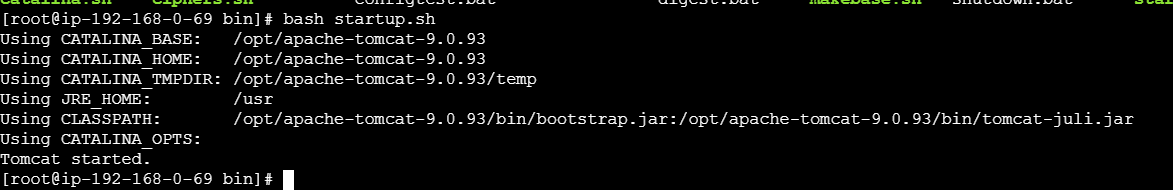
Extract the file

Tar xvf apache---

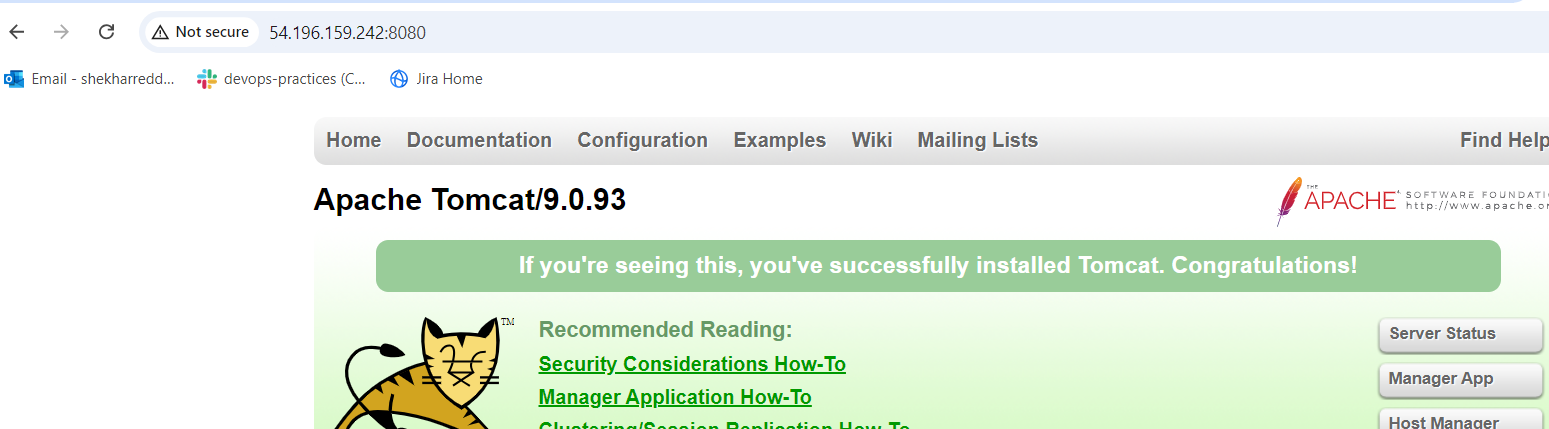
Open the tomcat folder and go to bin folder

Then start the tomcat service

"**./startup.sh**”



\* deployed on browser



above command gives 3 context.xml files. comment (<!-- & -->) `Value ClassName` field on files which are under webapp directory.

After that restart tomcat services to effect these changes

**tomcatdown**

**Tomcatup**

now application is accessible on port 8090. but tomcat application doesnt allow to login from browser. changing a default parameter in context.xml does address this issue

find / -name context.xml

/opt/apache-tomcat-9.0.93/conf/context.xml

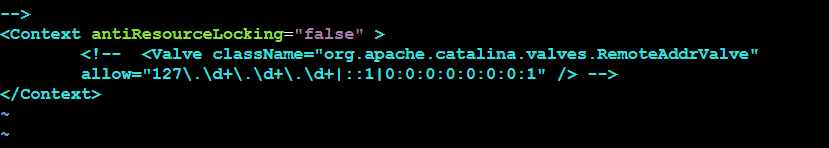
/opt/apache-tomcat-9.0.93/webapps/docs/META-INF/context.xml

/opt/apache-tomcat-9.0.93/webapps/examples/META-INF/context.xml

/opt/apache-tomcat-9.0.93/webapps/host-manager/META-INF/context.xml

/opt/apache-tomcat-9.0.93/webapps/manager/META-INF/context.xml

Like below image we need to address the parameter at above mentioned files



Update below users information in tomcat-users.xml , which is available in /opt/apache-tomcat-9.0.93/conf

<role rolename="manager-gui"/>

<role rolename="manager-script"/>

<role rolename="manager-jmx"/>

<role rolename="manager-status"/>

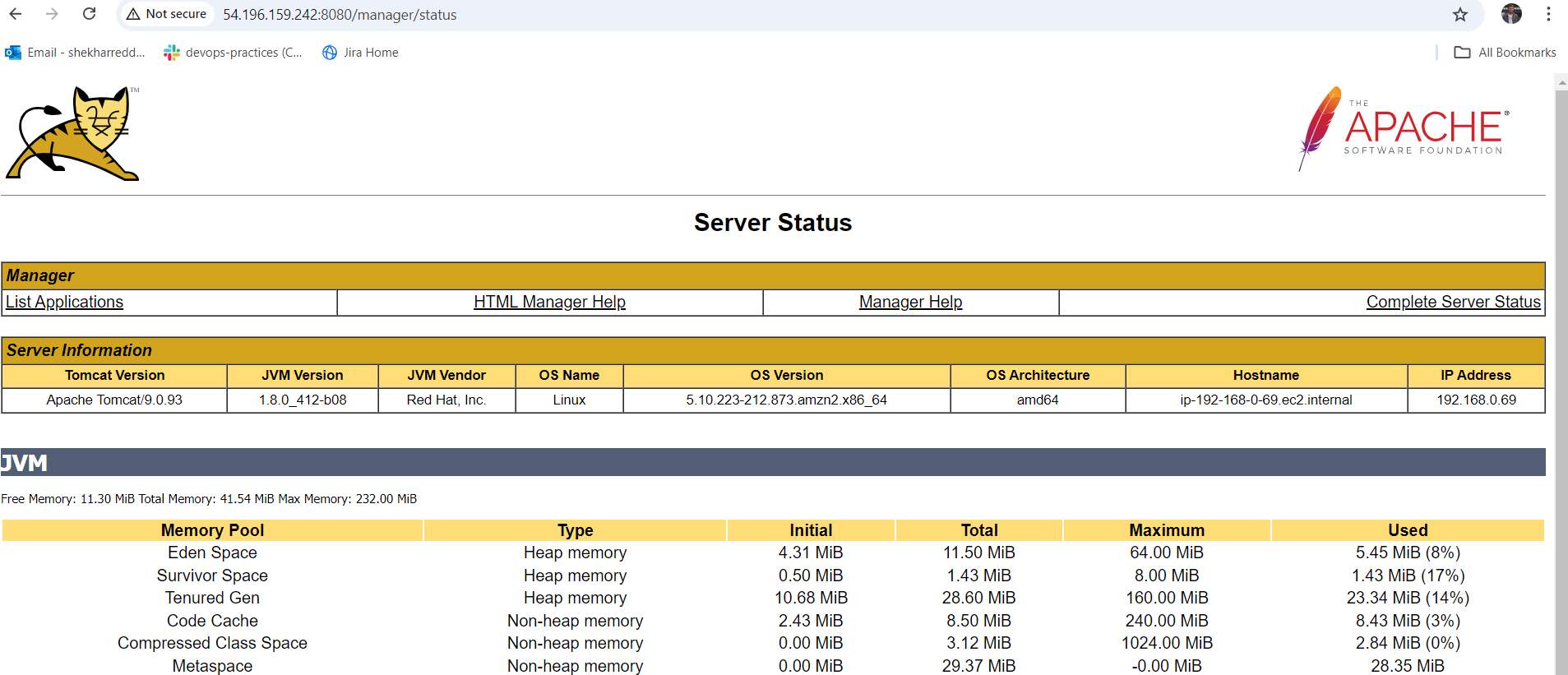
<user username="admin" password="admin" roles="manager-gui, manager-script, manager-jmx, manager-status"/>

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

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

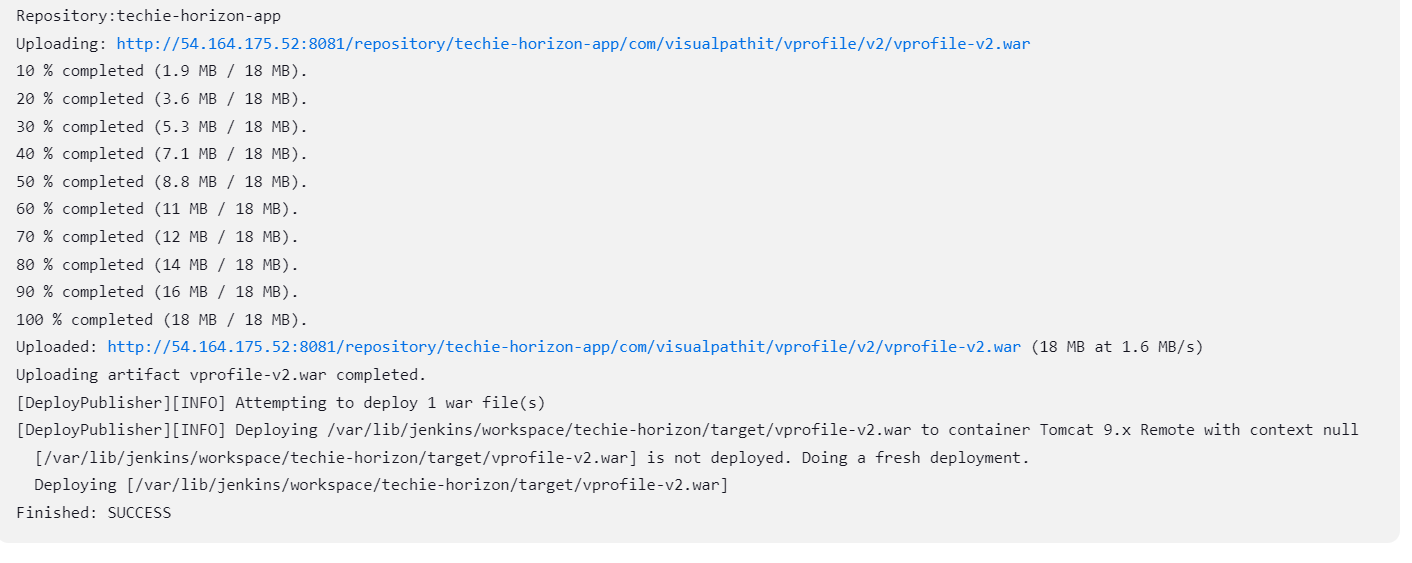
\*Restart serivce and try to login to tomcat application from the browser. This time it should be Successful

Collapse

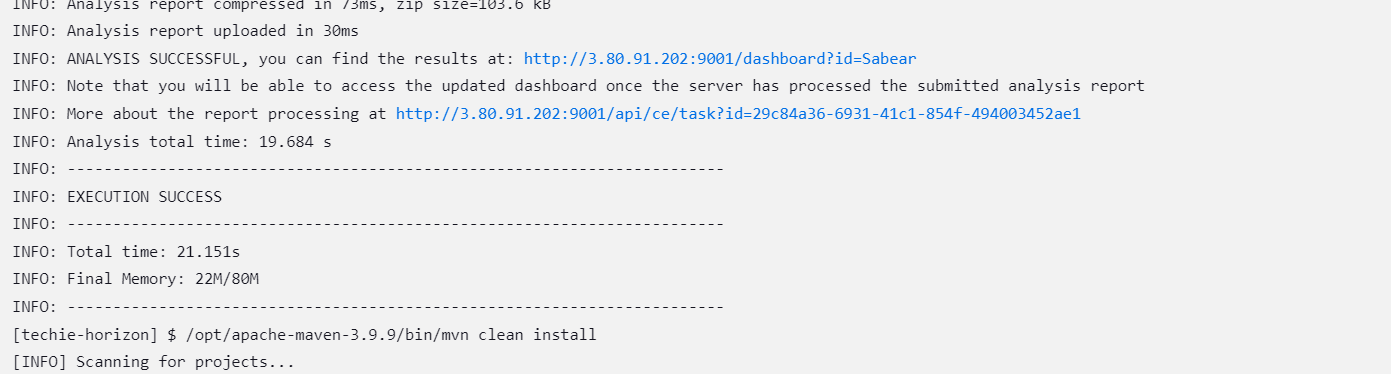


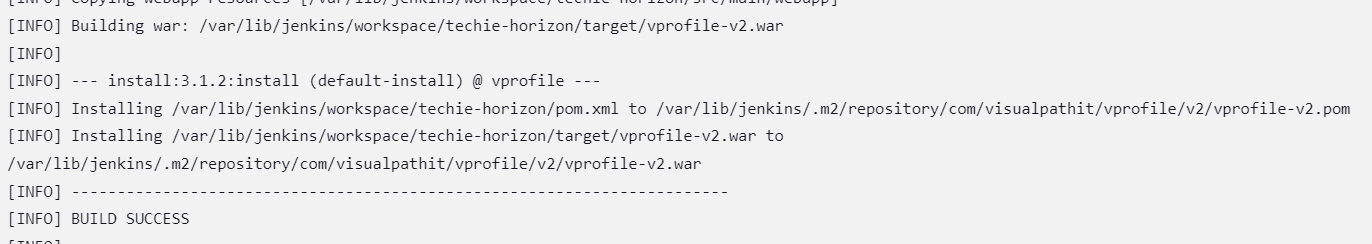
We can add the credentials for integrating to the jenkins





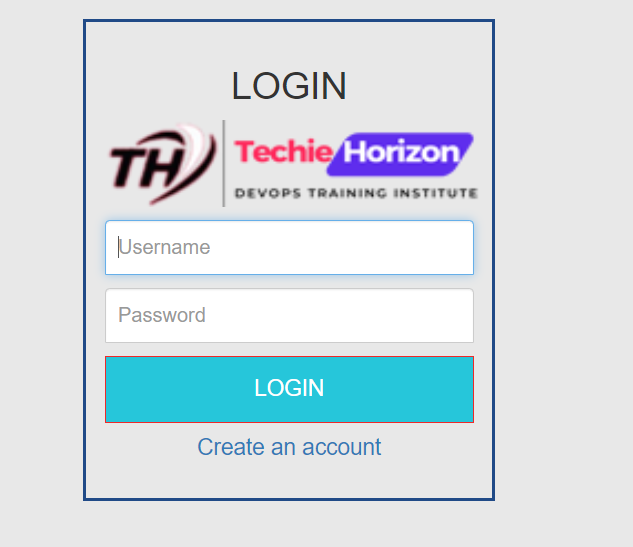
After completing the all configurations Build the job, Then all stages will successfully executed exepting slack notification







Using <http://54.196.159.242:8080/vprofile-v2> access to the browser, we can accessible to the server



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

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

**Stages:**

**1) Git Clone**

**2) Sonarqube Integration**

**3) Maven Compilation**

**4) Nexus Artifactory**

1. **Git Clone**
2. **Sonarqube Integration**

Prerequisite to Install jenkins is **java**.

Download jenkins Repo:

=====================

sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo

Import the jenkins key:

======================

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

Update ec2:

==========

sudo yum upgrade

Add required dependencies for the jenkins package:

=================================================

sudo amazon-linux-extras install java-openjdk11

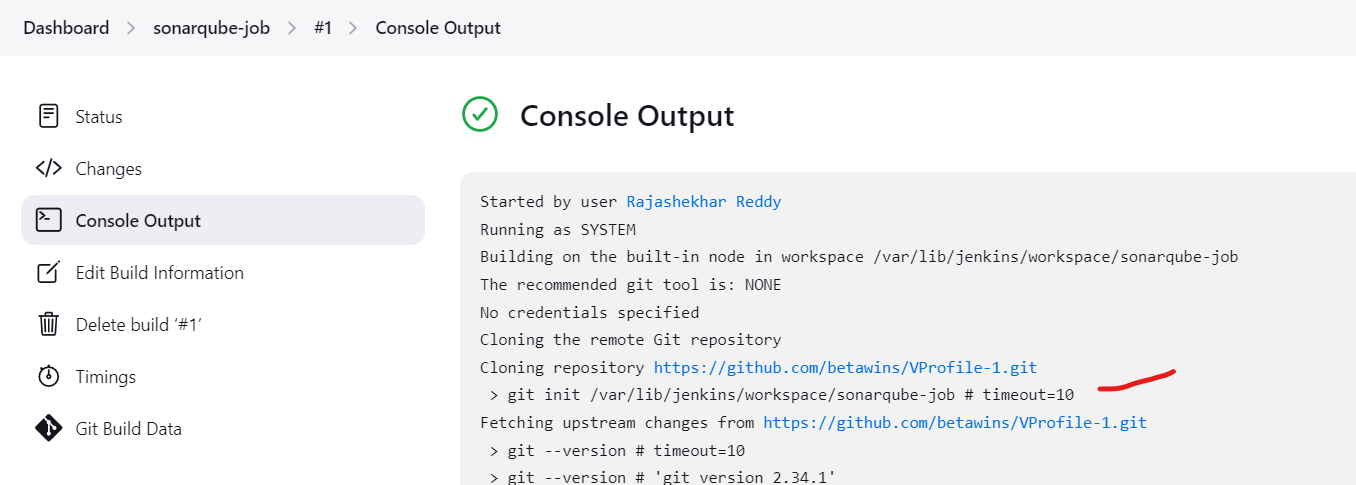
Install Jenkins:

===============

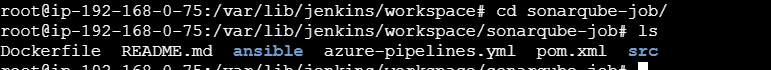
sudo yum install jenkins

\* First Access to my Jenkins-master Dashboard then I created new job (sonarqube-job) in added a git URL

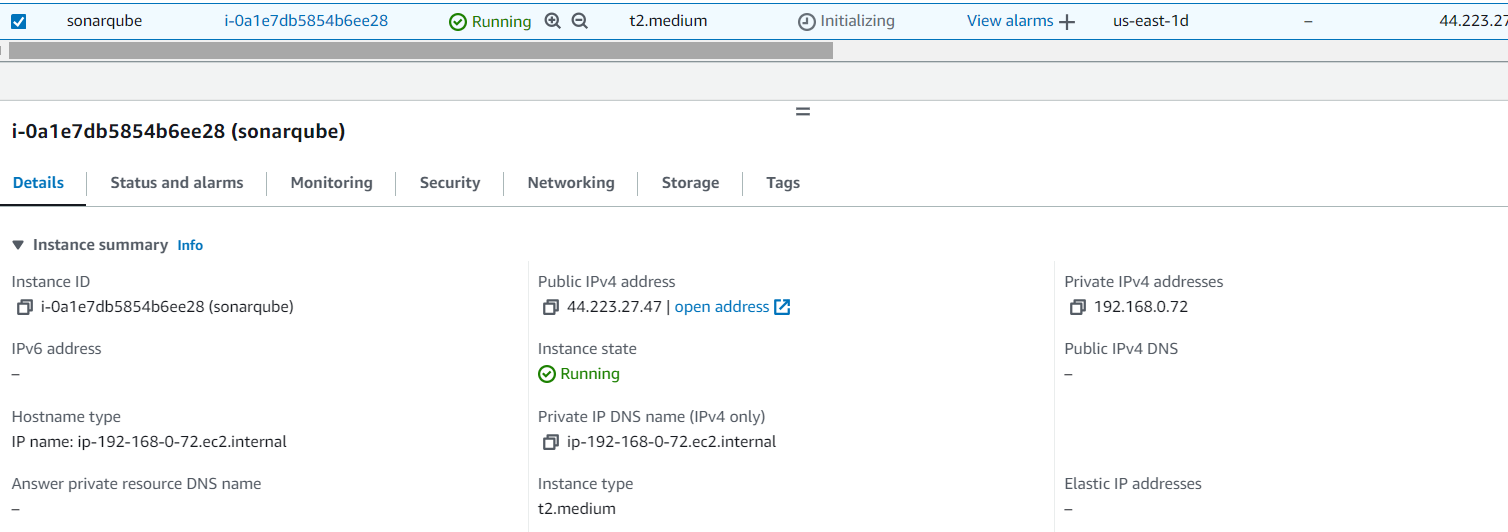
Url: “https://github.com/betawins/VProfile-1.git”



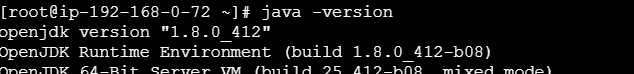
\* Once created the job build it, after build the job workspace will created



\* First created a new instance with t2.medium and with storage of 20 GB



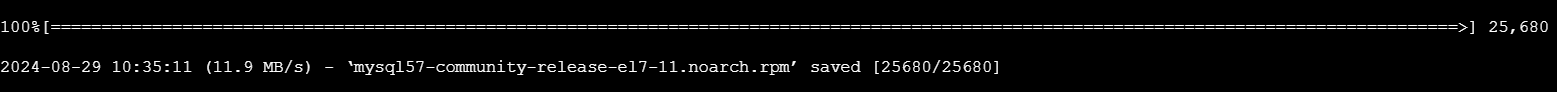
\* install Java using “ yum install java-1.8\*



\* Download mysql package using

sudo wget https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm

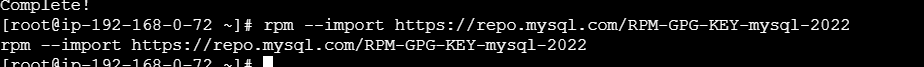
sudo yum localinstall mysql57-community-release-el7-11.noarch.rpm





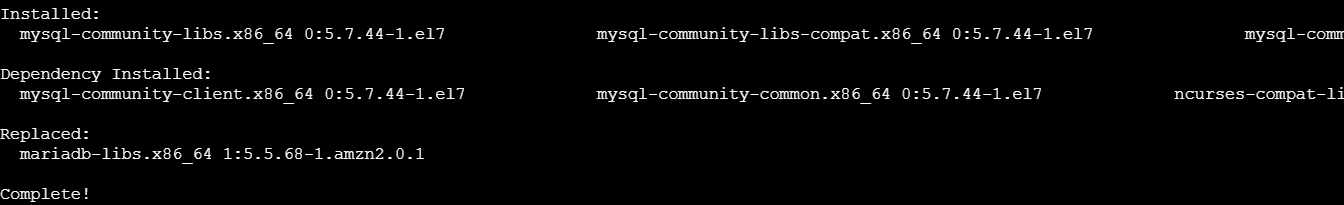
\* Import the key using

rpm --import <https://repo.mysql.com/RPM-GPG-KEY-mysql-2022>



\* Install the mysql community-server

sudo yum install mysql-community-server



\* Then start the mysql using

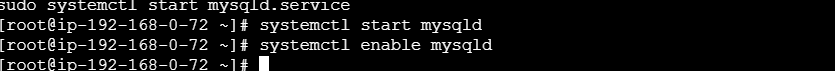
sudo systemctl start mysqld.service



Start MySQL and Enable Start at Boot Time using below commands

systemctl start mysqld

systemctl enable mysqld



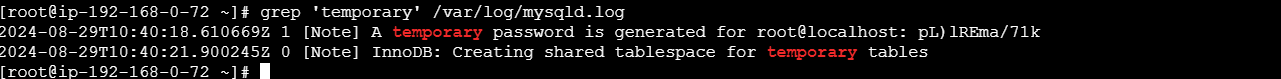
\* I have checked that mysql port is running or not using netstat -na | grep 3306



Configure the MySQL Root Password

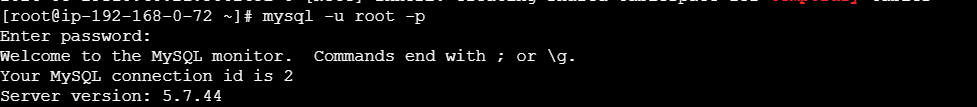
We will see default MySQL root password

grep 'temporary' /var/log/mysqld.log



Login to mysql using the default password

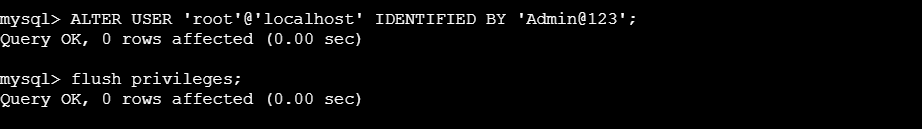
mysql -u root -p



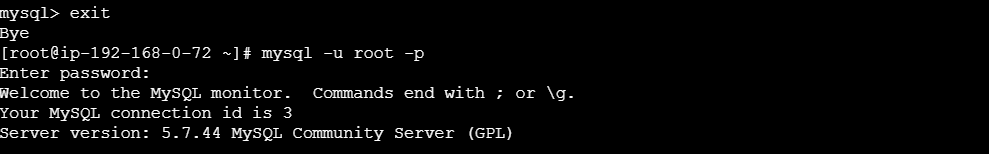
Now replace the default password with a new and strong password

ALTER USER 'root'@'localhost' IDENTIFIED BY 'Admin@123';

flush privileges;



Tested Using new password , alogged in with new password

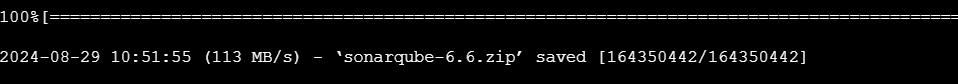


Download & unzip SonarQube 6.0

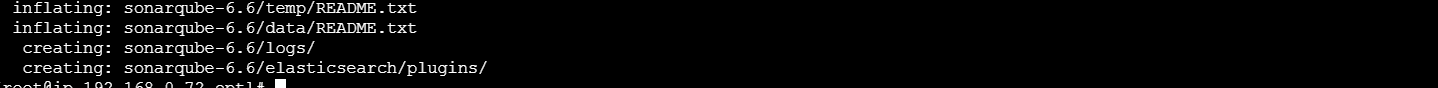
\* change to “**/opt”**

Then Download sonarQube

wget <https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-6.6.zip>

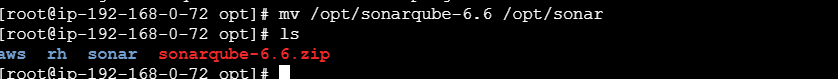


unzip sonarqube-6.6.zip



\* Rename the folder name

mv /opt/sonarqube-6.6 /opt/sonar



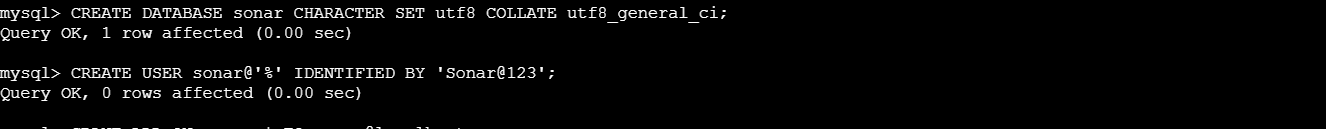
Login to mysql

mysql -u root -p

\* Create a local and a remote user

CREATE DATABASE sonar CHARACTER SET utf8 COLLATE utf8\_general\_ci;

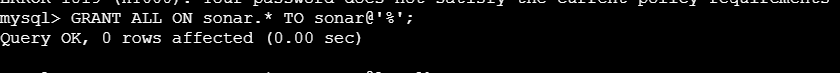
CREATE USER sonar@'%' IDENTIFIED BY 'Sonar@123';



Grant database access permissions to users

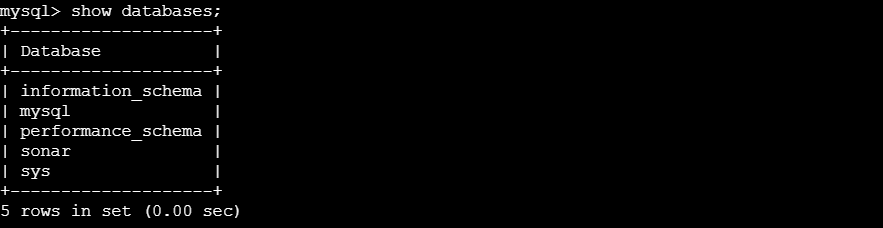
GRANT ALL ON sonar.\* TO sonar@localhost;

GRANT ALL ON sonar.\* TO sonar@'%';

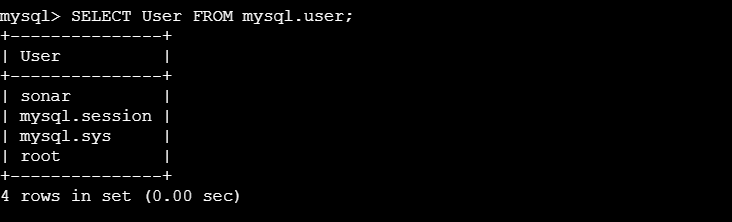


**check users and databases**

show databases;



SELECT User FROM mysql.user;



FLUSH PRIVILEGES;



**#ON EC2-Instance**

Edit sonar properties file to uncomment and provide required information for below properties.

- File Name: /opt/sonar/conf/sonar.properties

- sonar.jdbc.username=`sonar`

- sonar.jdbc.password=`Sonar@123`

- sonar.jdbc.url=jdbc:mysql://`localhost:3306`/sonar?useUnicode=true&characterEncoding=utf8&rewriteBatchedStatements=true&useConfigs=maxPerformance&useSSL=false

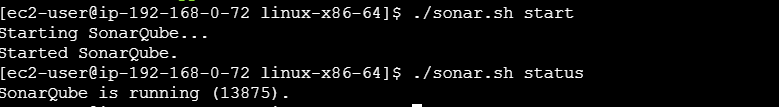
- sonar.web.host=`0.0.0.0`

- sonar.web.context=`/sonar

Start SonarQube service

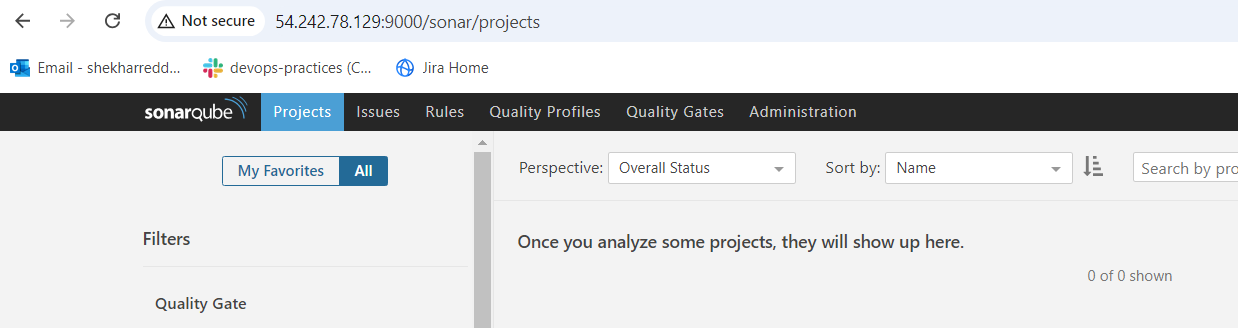
# cd /opt/sonar/bin/linux-x86-64/

# ./sonar.sh start

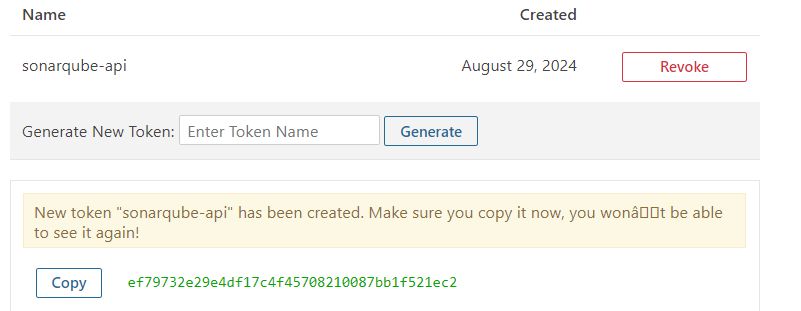


Logged into sonarqube

http://54.242.78.129:9000/sonar

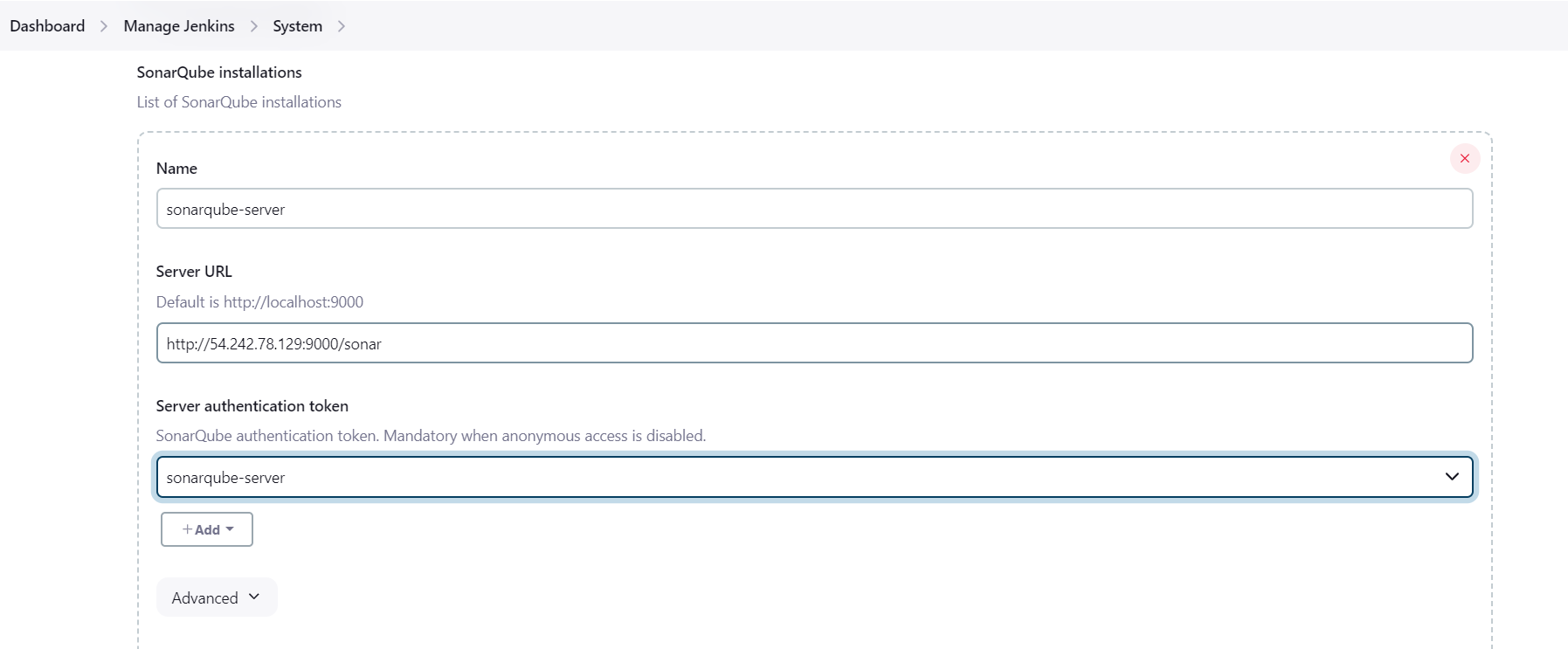


Then create a token in sonarqube



\* To add the Global credentials of Sonarqube

\* Jenkins [Dashboard](http://54.165.95.20:8081/)→[Manage Jenkins](http://54.165.95.20:8081/manage/)→[Credentials](http://54.165.95.20:8081/manage/credentials/)→[System](http://54.165.95.20:8081/manage/credentials/store/system/)→[Global credentials (unrestricted)](http://54.165.95.20:8081/manage/credentials/store/system/domain/_/)



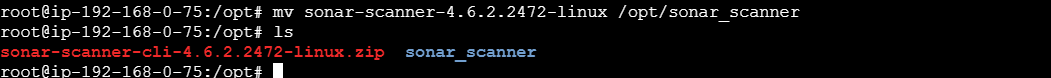
\* Download the sonarqube-scanner package at “opt” folder in jenkins-server

wget https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-4.6.2.2472-linux.zip

unzip sonar-scanner-cli-4.6.2.2472-linux.zip



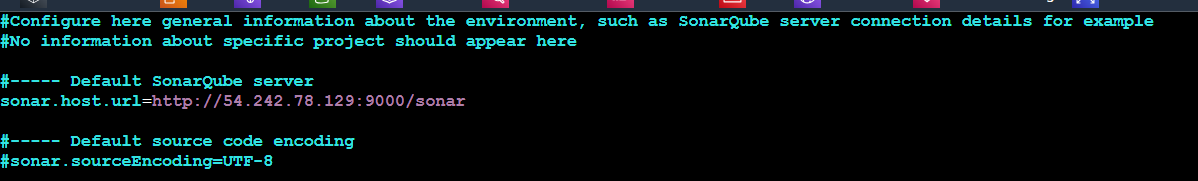
\* Renamed mv sonar-scanner-4.6.2.2472-linux /opt/sonar\_scanner



Set SonarQube server details in sonar-scanner property file

Sonar properties file: /opt/sonar\_scanner/conf/sonar-scanner.properties

- sonar.host.url=http://`<SONAR\_SERVER\_IP>`:9000/sonar

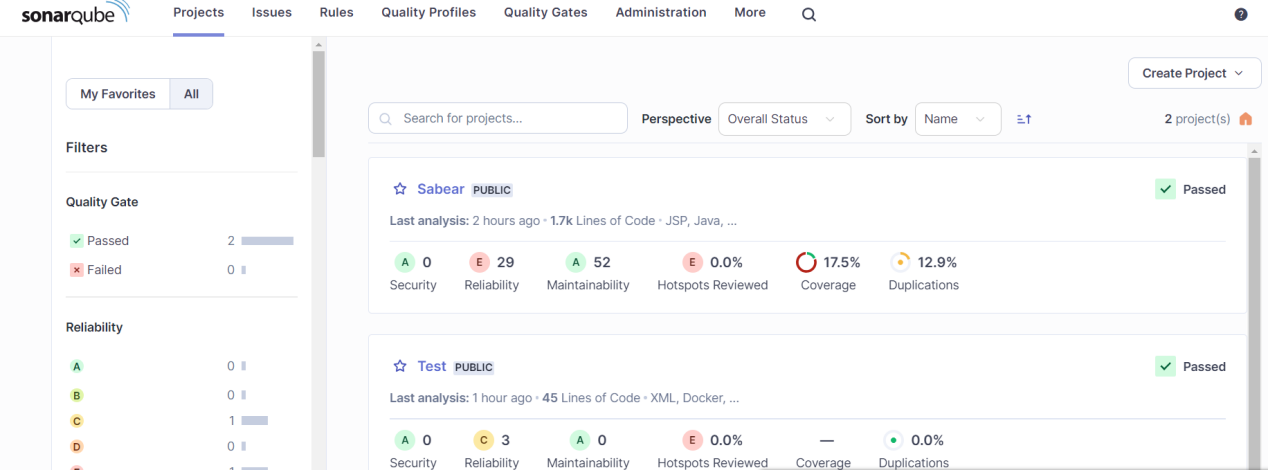


Configuring SonarQube scanner home path

**Manage Jenkins` > `Global Tool Configuration` > `SonarQube Scanner`**

**-** Name  **: `sonar\_scanner`**

**-** SONAR\_RUNNER\_HOME **: `/opt/sonar\_scanner/bin`**



**3)Maven Compilation**

sudo apt install fontconfig openjdk-17-jre

#How to install maven:

1. Change to opt direcotry

cd /opt

2) Download maven

Wget https://mirrors.estointernet.in/apache/maven/maven-3/3.6.3/binaries/apache-maven-3.6.3-bin.tar.gz

3) Extract the maven tar file

tar xvf apache-maven-3.6.3

1. Export maven homepath in “.bash\_profile”

# .bash\_profile

# Get the aliases and functions

if [ -f ~/.bashrc ]; then

. ~/.bashrc

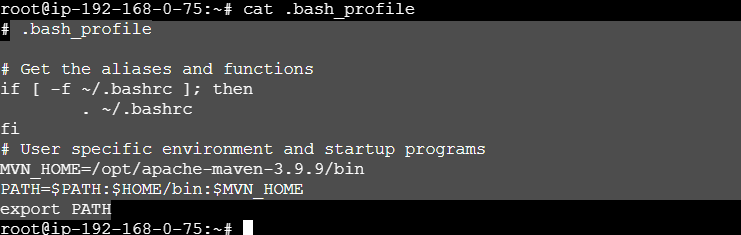
fi

# User specific environment and startup programs

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

PATH=$PATH:$HOME/bin:$MVN\_HOME

export PATH

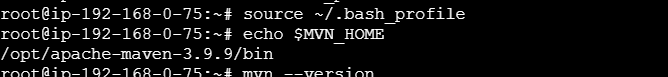


5) Restart the .bash\_profile

source ~/.bash\_profile

6) Check for MVN\_HOME

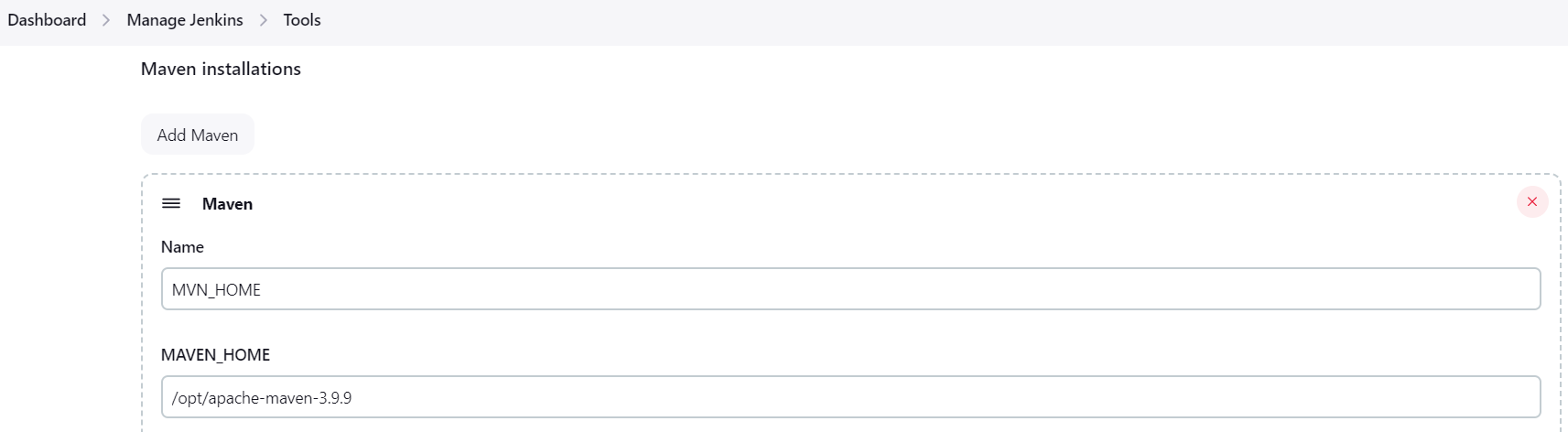
echo $MVN\_HOME

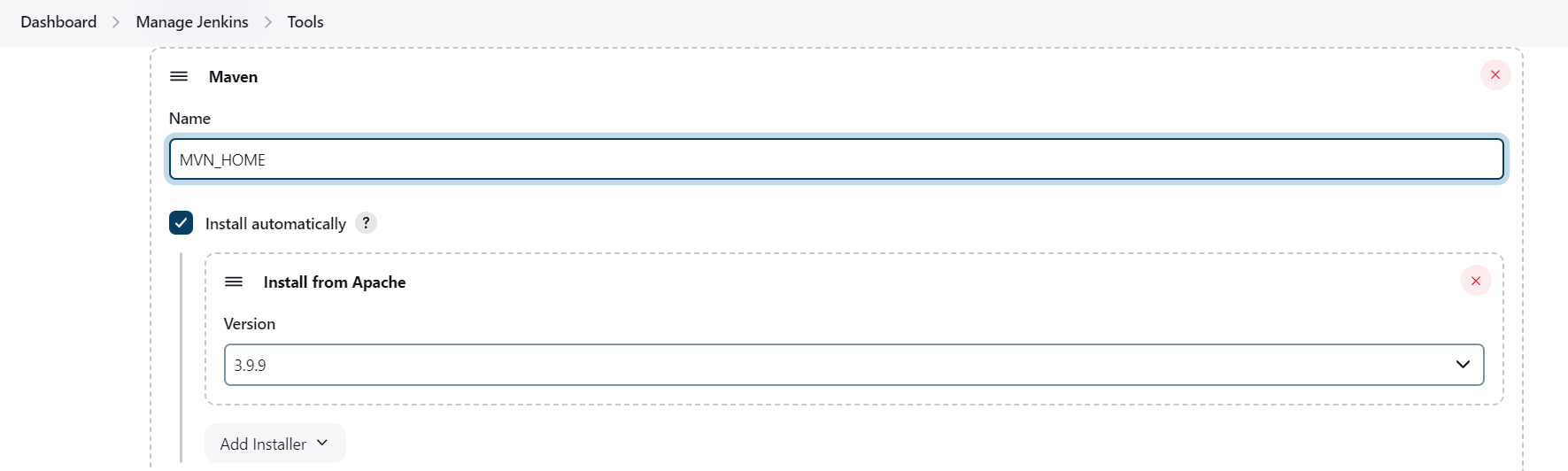


\* To check the version of maven “ **mvn --version”**

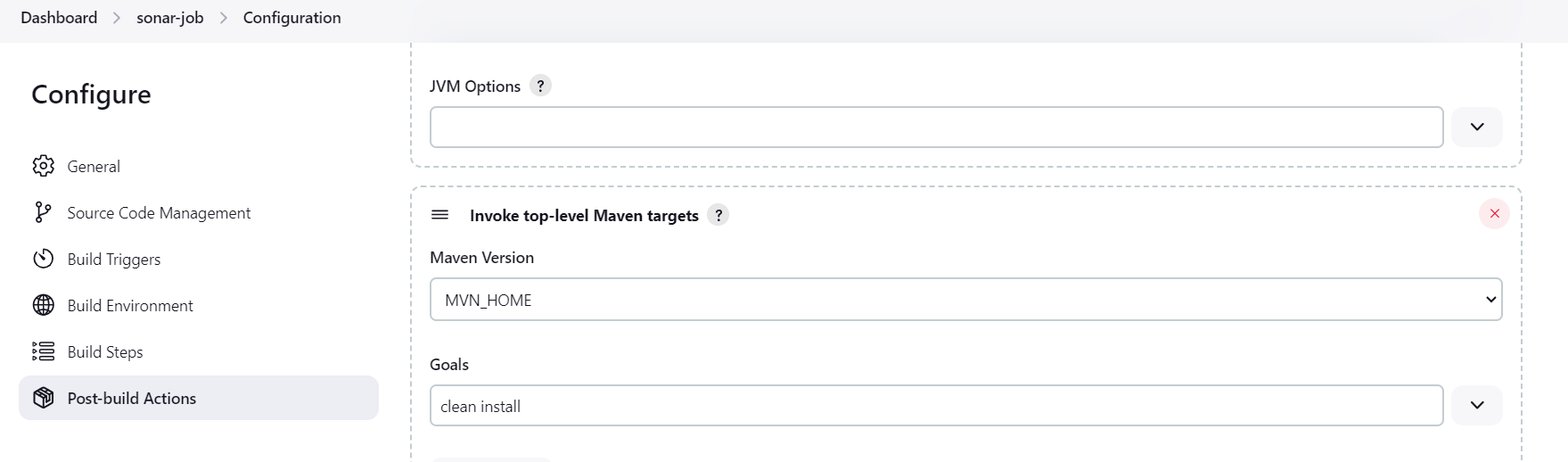


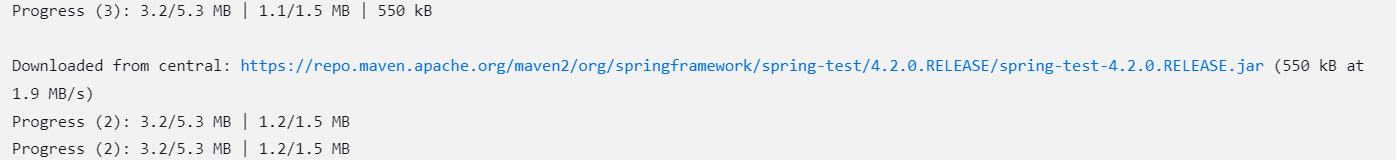
·Go to Jenkins [Dashboard](http://54.90.95.135:8081/)→ [Manage Jenkins](http://54.90.95.135:8081/manage/)→ [Tools](http://54.90.95.135:8081/manage/configureTools/) to integrate to the jenkins



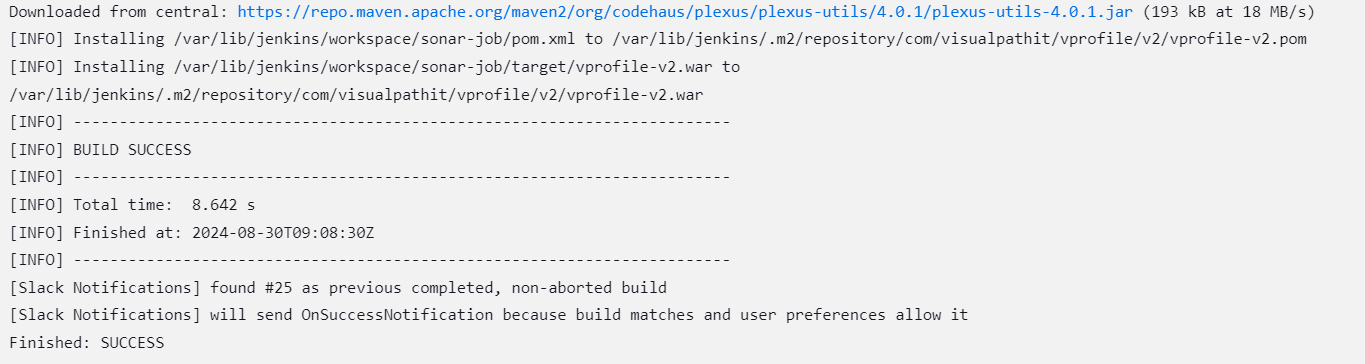


\* configured the job to save the packages in the given path

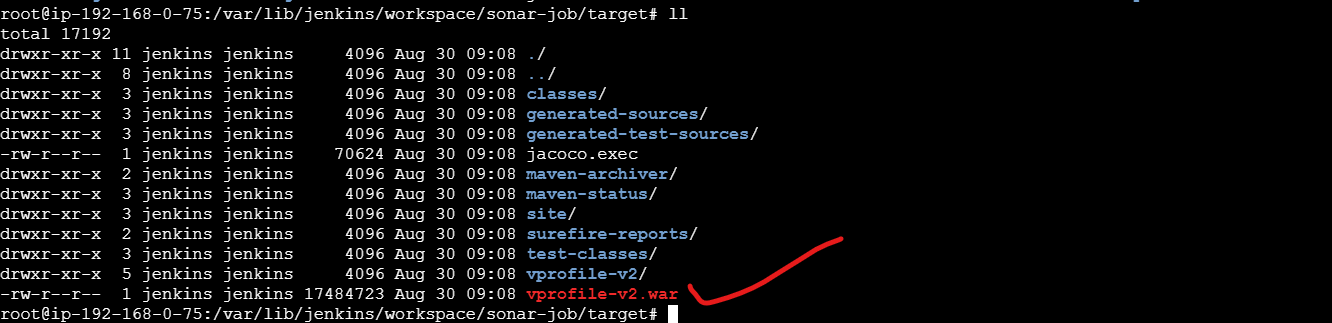




After completion of maven integration build successfully executed



Once complete the maven compilation then only TARGET folder will create, where this war packages will store



\* we can find a folder”.**m2**” , in this repository file will downloaded all the dependencies of maven



1. **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

```

As a good security practice, it is not advised to run nexus service as root. so create new user called nexus and grant sudo access to manage nexus services

sudo adduser nexus

# visudo \\ nexus ALL=(ALL) NOPASSWD: ALL

sudo chown -R nexus:nexus /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

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

Add nexus as a service at boot time

sudo ln -s /opt/nexus/bin/nexus /etc/init.d/nexus

Login as a nexus user and start service

su - nexus

service nexus start

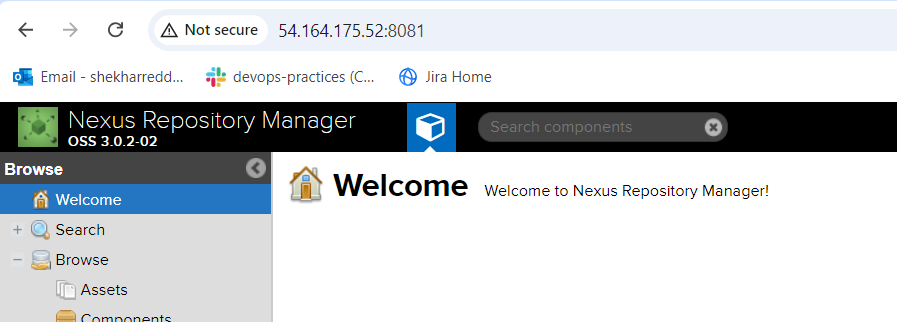
Login nexus server from browser on port 8081

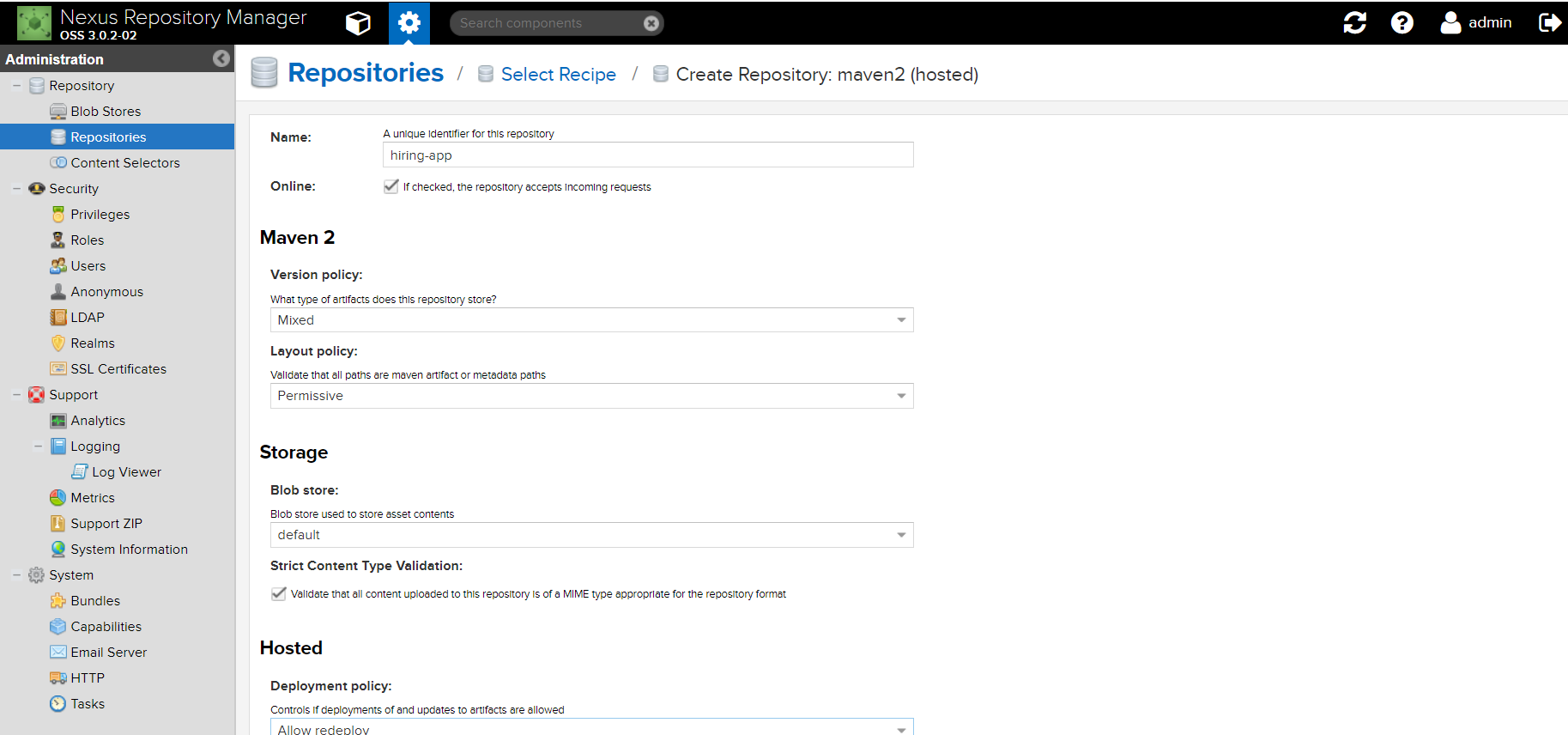
http://<Nexus\_server>:8081

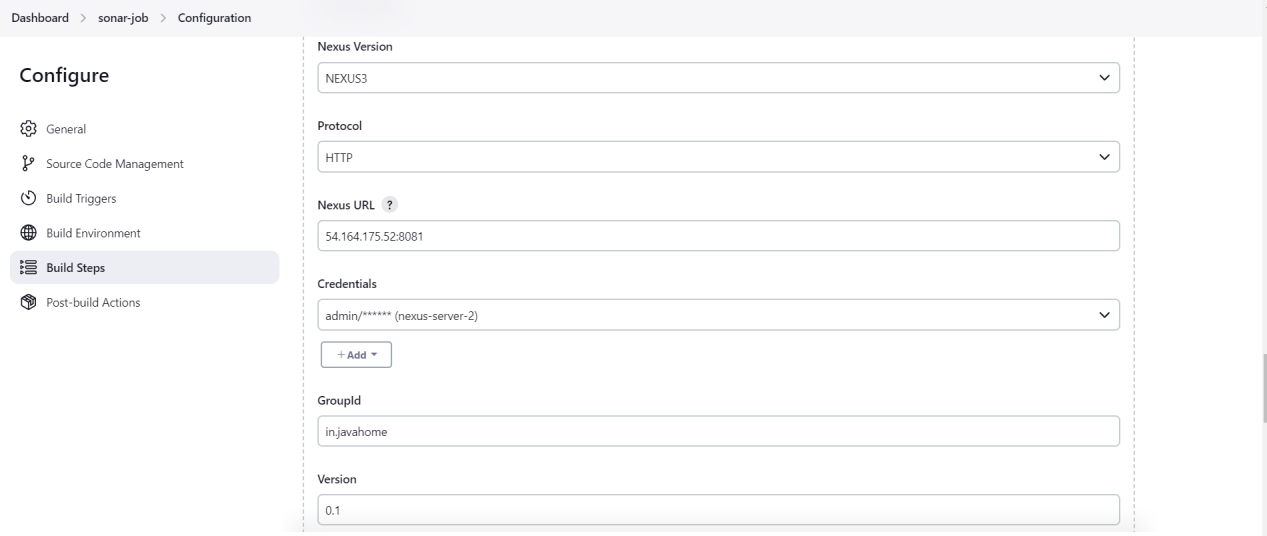
Use default credentials to login

username : admin

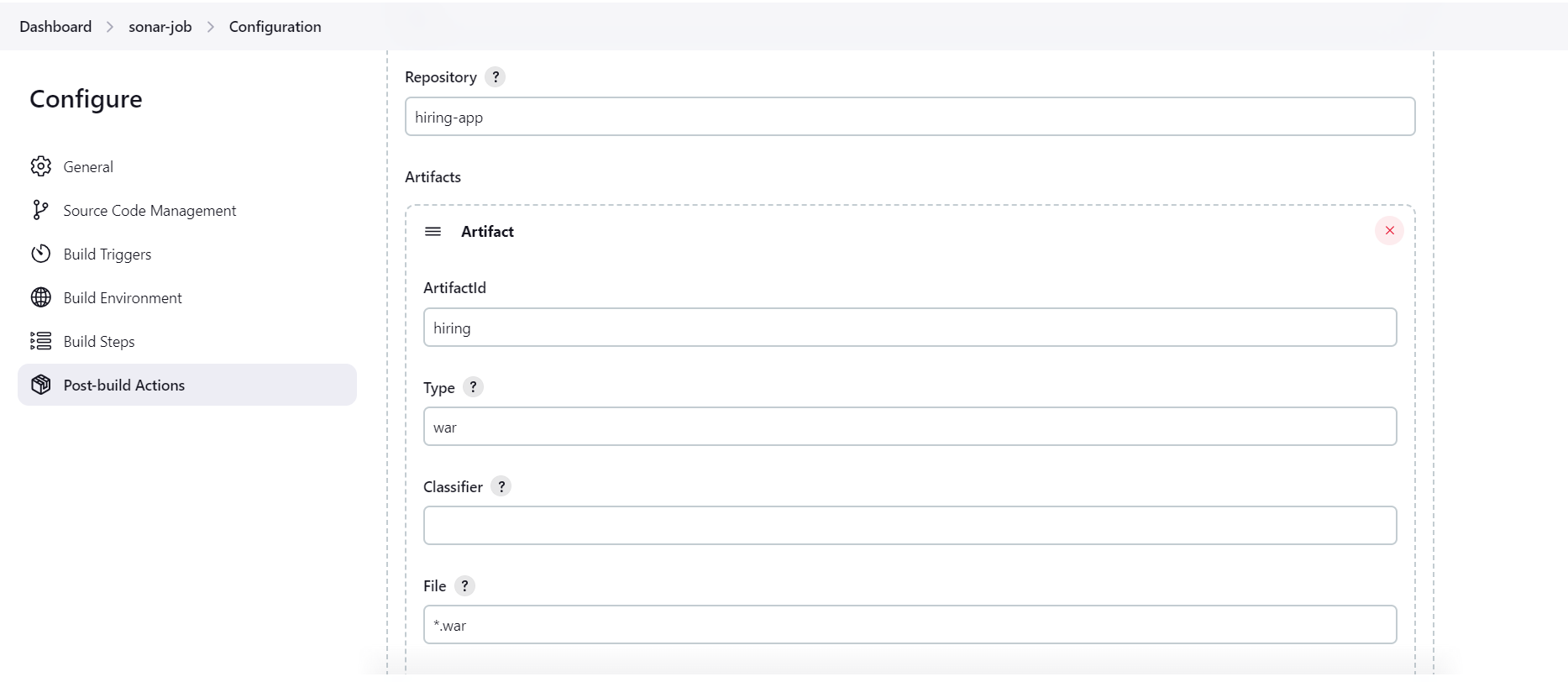
password : admin123

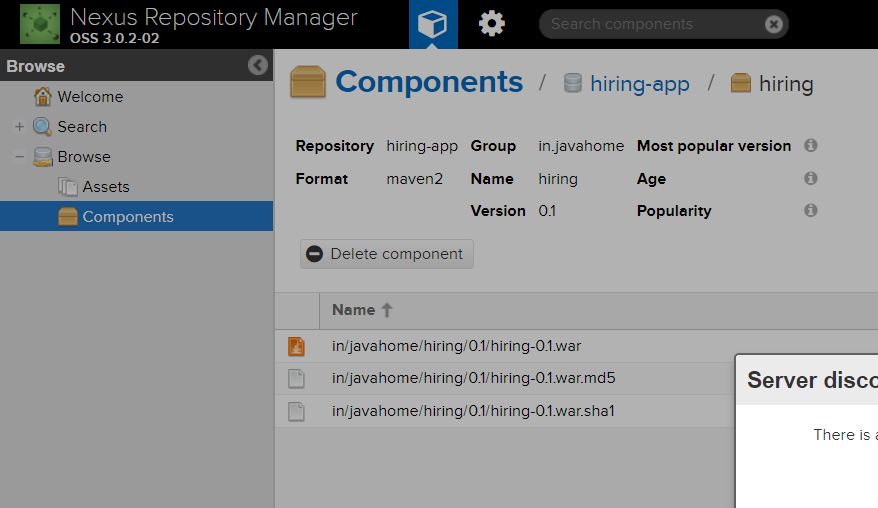




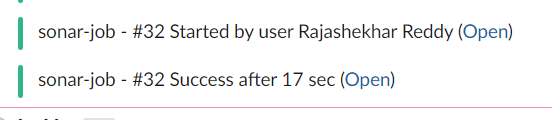


\* Integrated with nexus artifactory

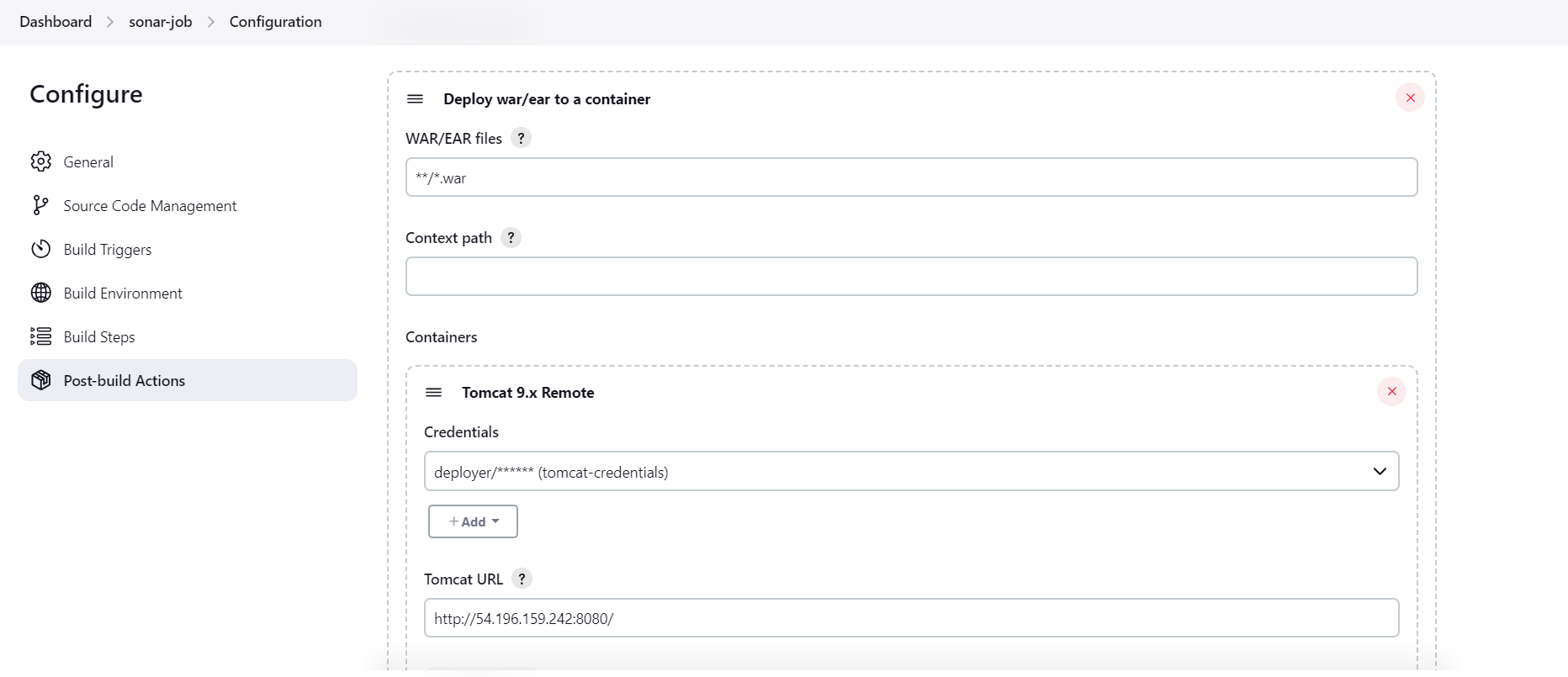


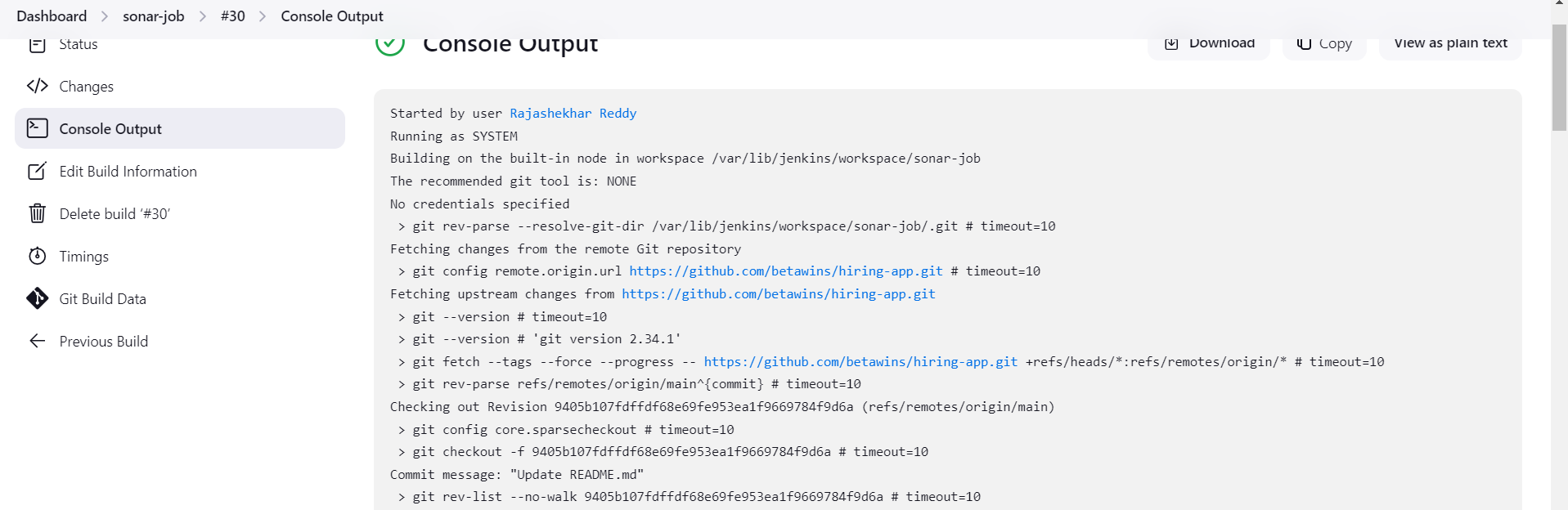


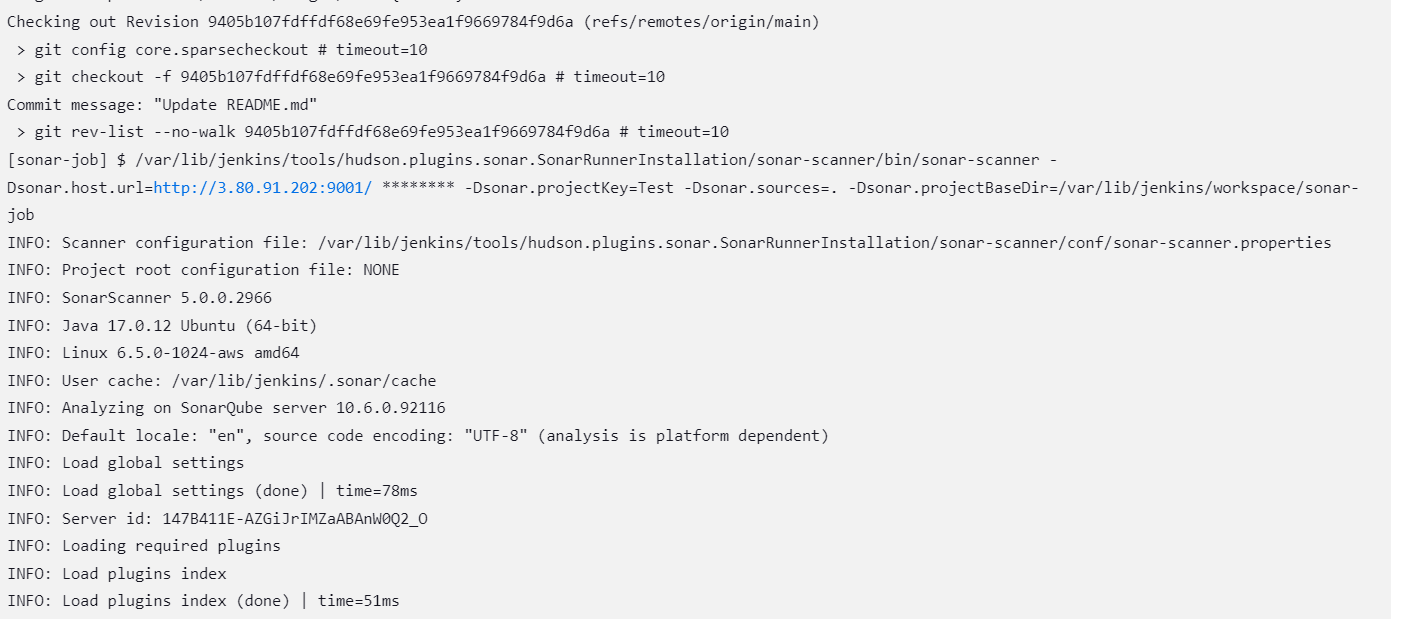
**5)Slack Notification**



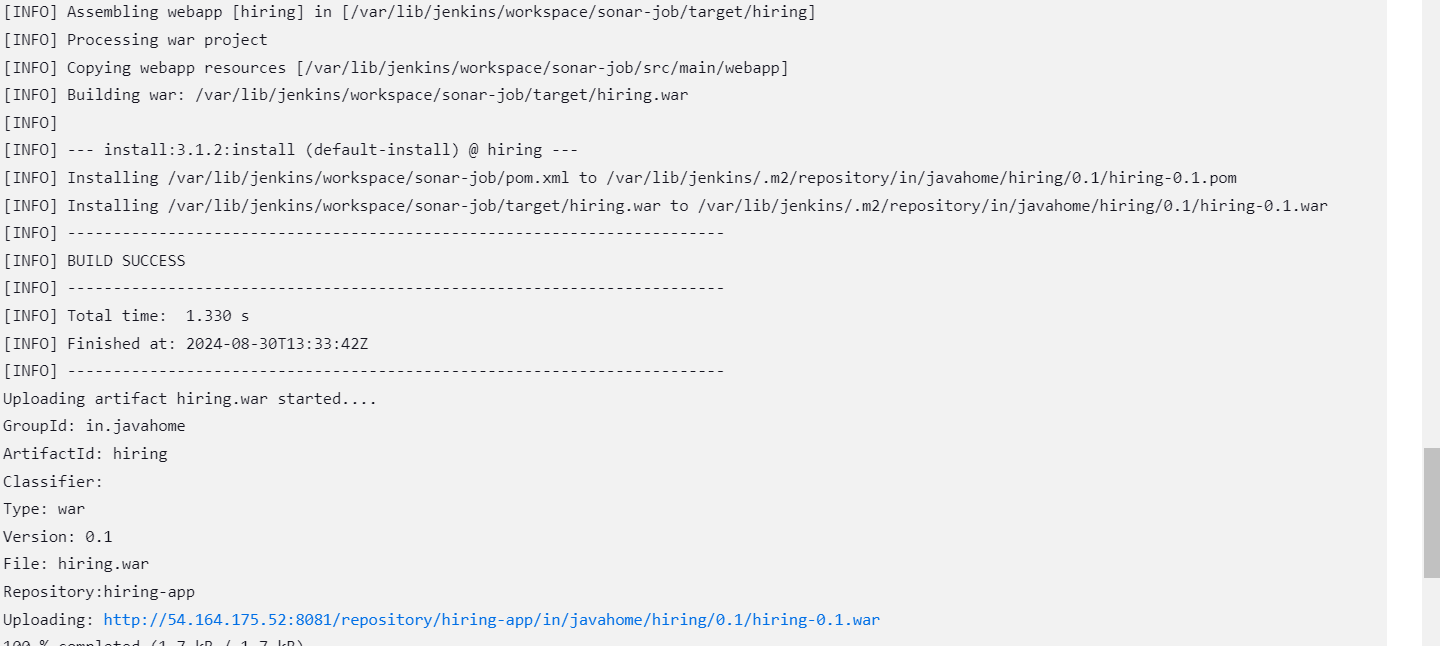
**6) Deploy On tomcat**

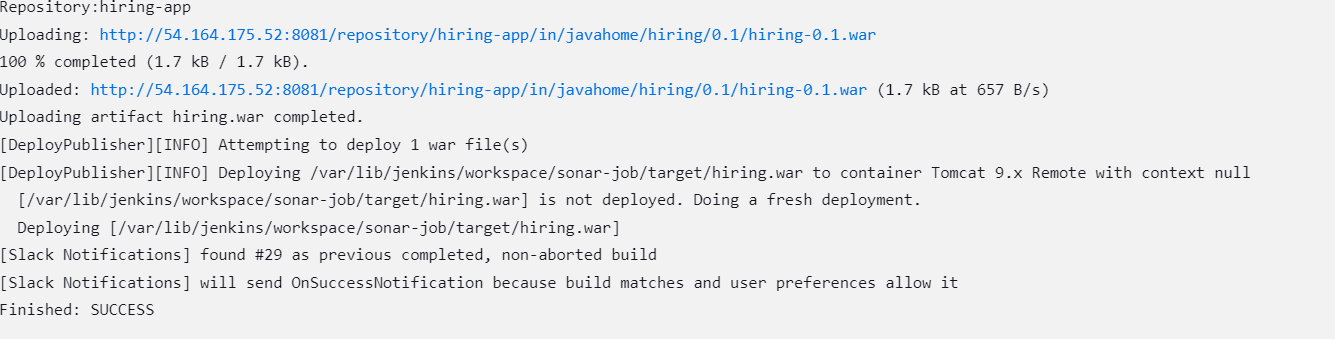












Deployed Successfully.

