

→ NEXUS ARTIFACT UPLOADER VERSION 2.14

#SETTING UP SONARQUBE SERVER

```
# yum install -y docker
```

```
# systemctl start docker
```

```
# docker pull sonarqube
```

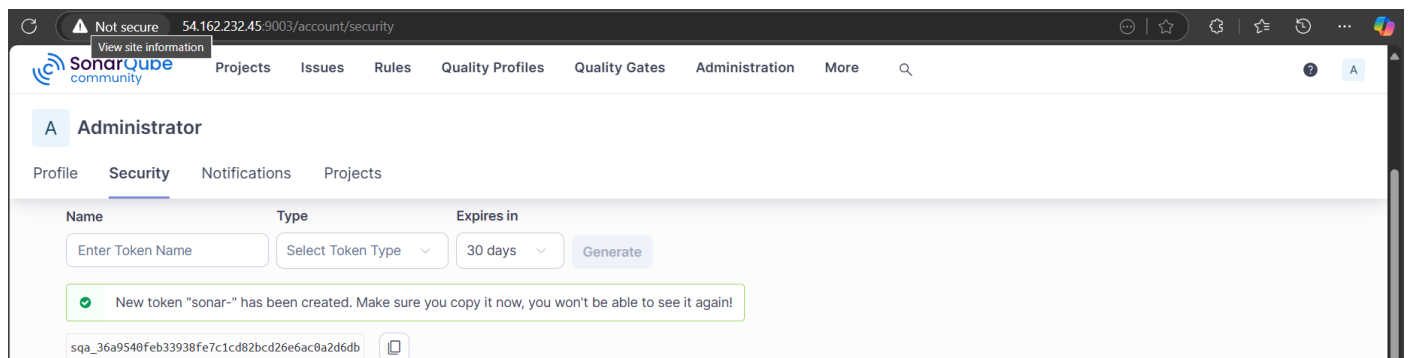
```
# docker images
```

```
# docker run -itd -p 9003:9000 sonarqube
```

```
# docker ps
```

```
[root@ip-10-0-0-43 ec2-user]# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
2a8509961b90   sonarqube     "/opt/sonarqube/dock..." About an hour ago Up About an hour 0.0.0.0:9003->9000/tcp, :::9003->9000/tcp flamboyant_northcutt
```

<LOGGING IN TO SONARQUBE SERVER THROUGH GUI AND CREATING A SECRET TEXT FOR INTEGRATING IT IN JENKINS>



#SETTING UP MAVEN IN JENKINS SERVER

```
# cd /opt/
```

```
# wget https://dlcdn.apache.org/maven/maven-3/3.9.9/binaries/apache-maven-3.9.9-bin.tar.gz
```

```
# tar xvf apache-maven-3.9.9-bin.tar.gz
```

```
# cd ~
```

```
# vi .bash_profile
```

```
MAVEN=/opt/apache-maven-3.9.9/bin
```

```
PATH=$PATH:$HOME/bin:$MAVEN
```

```
# chown -R jenkins:jenkins apache-maven-3.9.9
```

#SETTING UP NEXUS SERVER

```
# yum install java-1.8*
```

```
# 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
```

```
# sudo adduser nexus
```

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

```
# sudo chown -R nexus:nexus /opt/nexus
```

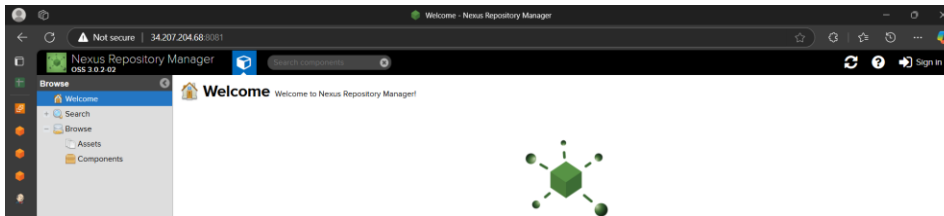
```
# vi /opt/nexus/bin/nexus.rc
```

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

```
# su - nexus
```

```
# service nexus start
```

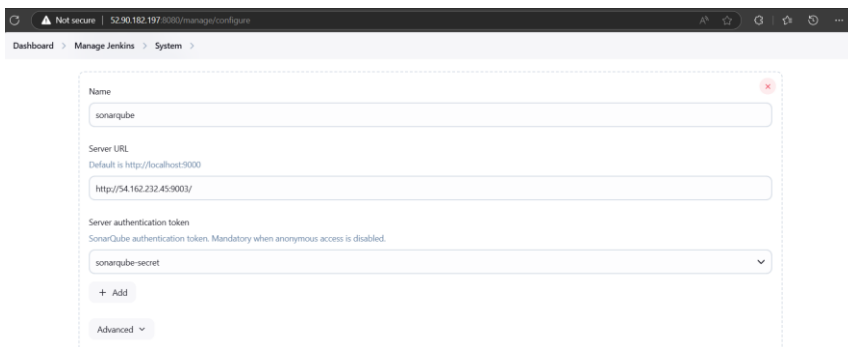
```
[nexus@ip-10-0-0-151 ~]$ service nexus status  
nexus is running.
```



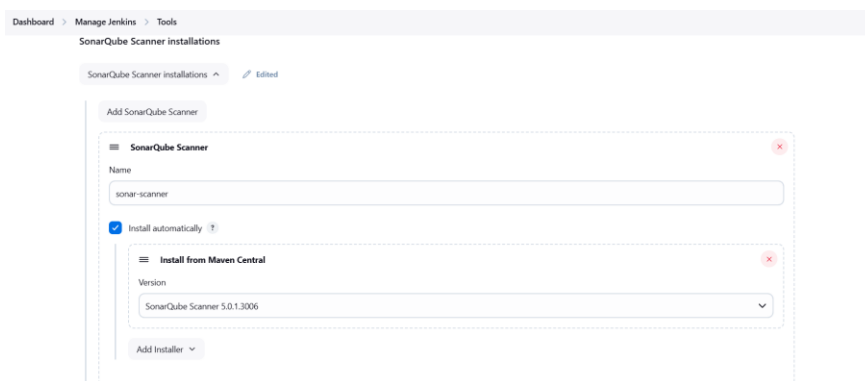
<default username is admin and password is admin123>

#CONFIGURATION IN JENKINS GUI

ADDING THE SONAR URL AND SECRET TEXT THAT WE CREATED



ADDING THE SONAR SCANNER FOR INTEGRATION OF THE CODE TO SONARCUBE FOR QUALITY CHECK



ADDING SLACK NOTIFICATIONS

OPEN SLACK>APPS>JENKINS>CONFIGURE>SELECT CHANNEL>

FOLLOW INSTRUCTIONS TO SETUP IN JENKINS GUI

>ADD WORKSPACE

>ADD SECRET TEXT

>ADD CHANNEL NAME AND TEST

Dashboard > Manage Jenkins > System

Slack

Workspace [?](#)
techshorizon

Credential [?](#)
slack-jenkins

+ Add

Default channel / member id [?](#)
jenkins-integration

☐ Custom slack app bot user [?](#)

Advanced [?](#)

Success Test Connection

ADD MAVEN CONFIGURATION

Dashboard > Manage Jenkins > Tools

Maven

Name
MAVEN

MAVEN_HOME
/opt/poche/maven-3.8.6/

☐ Install automatically [?](#)

STEP:2) CREATING A JOB WITH GIT-REPO , SONAR INTEGRATION , SLACK , MAVEN , NEXUS

→ADD GIT REPO

Dashboard > hiring-app > Configuration

Configure

General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions

Source Code Management

☐ None
☒ Git [?](#)

Repositories [?](#)

Repository URL [?](#)
https://github.com/betawins/hiring-app.git

Credentials [?](#)
- none -

↓ x.4.4

→ADDING SONAR

Dashboard > hiring-app > Configuration

Configure

General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions

Execute SonarQube Scanner

JDK [?](#)
JDK to be used for this SonarQube analysis
(Inherit From Job)

Path to project properties [?](#)

Analysis properties [?](#)
sonar.projectKey=hiring-app
sonar.sources=src/main/java

Additional arguments [?](#)

→ADDING SLACK

Dashboard > hiring-app > Configuration

Configure

General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions

Post-build Actions

Slack Notifications

☐ Notify Build Start
☒ Notify Success
☐ Notify Aborted
☐ Notify Not Built
☐ Notify Unstable
☐ Notify Regression
☐ Notify Every Failure
☐ Notify Back To Normal

→ ADDING MAVEN

Dashboard > hiring-app > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

Invoke top-level Maven targets

Maven Version: MAVEN

Goals: clean install

Advanced

→ ADDING NEXUS

>>first creating a repo

Nexus Repository Manager

Administration

Repositories / hiring-app

Settings

Name: hiring-app

Format: maven2

Type: default

URL: http://34.207.204.68:8081/repository/hiring-app/

Online: ☒ If checked, the repository accepts incoming requests

Maven 2

Version policy: Mixed

Layout policy: Permissive

Storage

Blob store: default

>>adding to jenkins job **NOTE:** DETAILS THAT HAS TO FILLED WILL BE FOUND IN ***pom.xml***

Dashboard > hiring-app > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

Nexus artifact uploader

Nexus Details

Nexus Version: NEXUS3

Protocol: HTTP

Nexus URL: 34.207.204.68:8081/

Credentials: admin/***** (nexus)

Dashboard > hiring-app > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

0.1

Repository: hiring-app

Artifacts

Artifact

ArtifactId: hiring

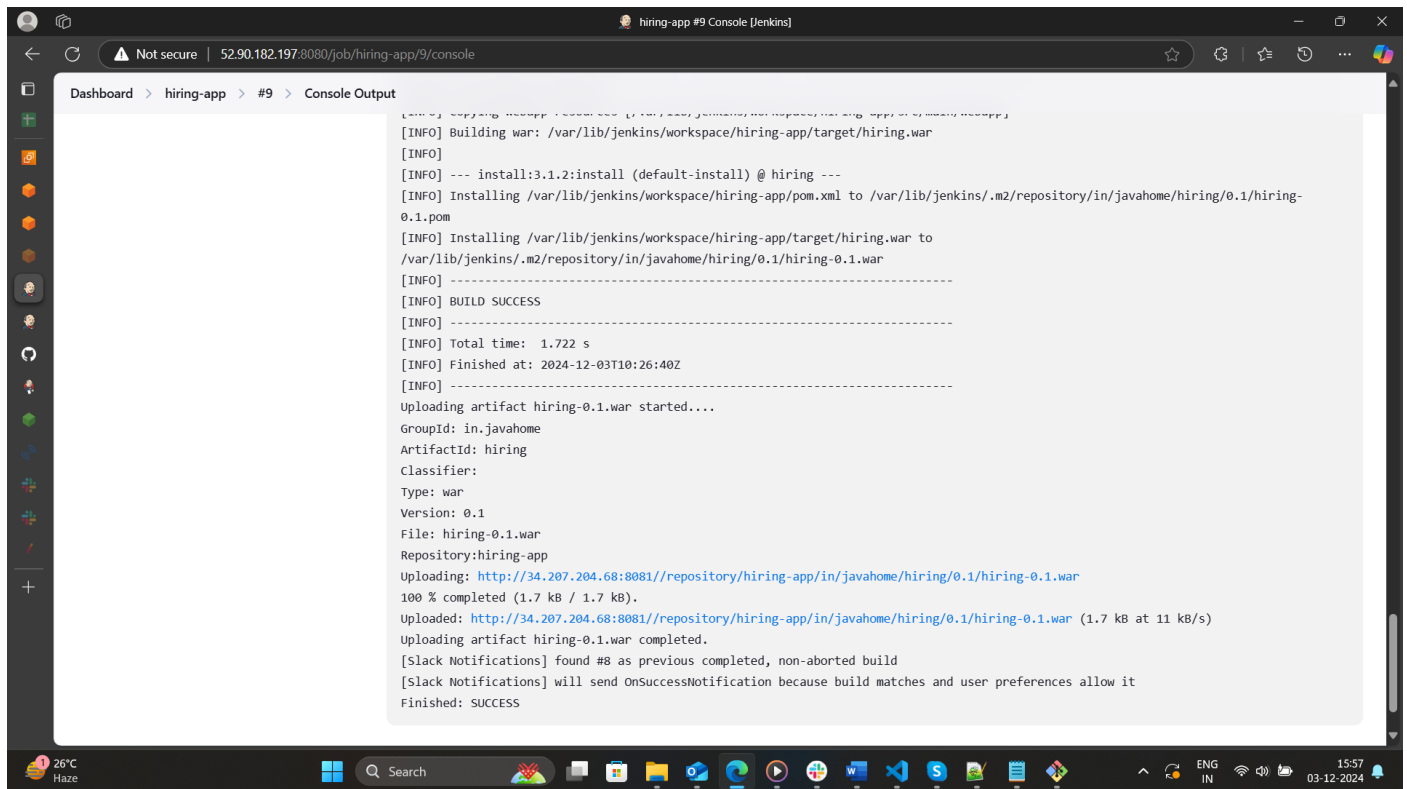
Type: war

Classifier:

File: /var/lib/jenkins/m2/repository/in/avahome/hiring/0.1/hiring-0.1.war

The file location is the location where the maven is saving the package

→SAVING IT AND CLICKING BUILD←



STEP:3) DEPLOYING THE HIRING-APP ON APACHE TOMCAT

→LAUNCH ONE INSTANCE AND INSTALL JAVA AND APACHE TOMCAT IN IT

```
# yum install java-17*
```

```
# cd /opt
```

```
# wget https://dlcdn.apache.org/tomcat/tomcat-10/v10.1.33/bin/apache-tomcat-10.1.33.tar.gz
```

```
# tar xvf apache-tomcat-10.1.33.tar.gz
```

```
# find / -name context.xml
```

→EDIT ALL THE CONTEXT.XML FILE AND COMMENT THE VALUE WITH <!-- -->

FOR EXAMPLE

```
<!-- <Valve
className="org.apache.catalina.valves.RemoteAddrValve"
```

```
# vi /opt/apache-tomcat-10.1.33/conf/tomcat-users.xml
```

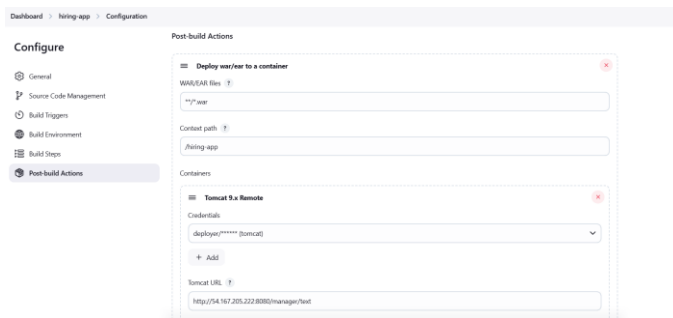
→ADD THIS VALUE IN THE FILE

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

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

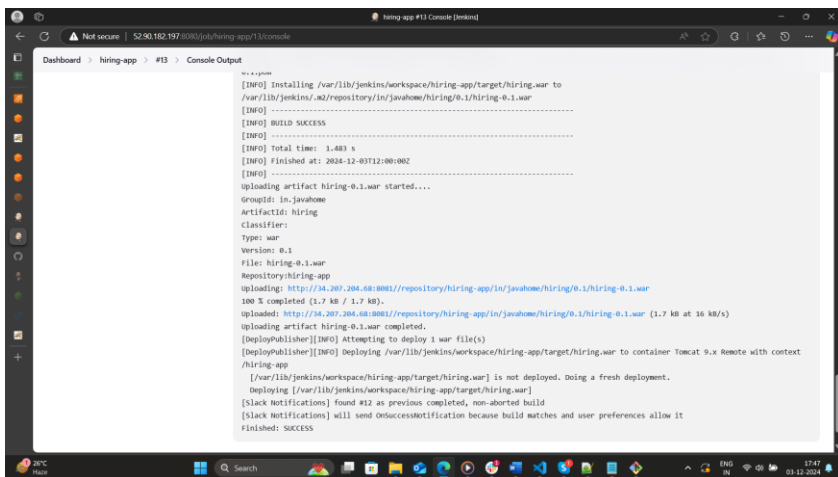
```
<user username="deployer" password="deployer" roles="manager-gui,manager-script"/>
```

>>ADDING THE POST BUILD ACTION IN JENKINS



CREDENTIALS ARE USERNAME AND PASSWORD THAT WE SAVED IN TOMCAT-USERS.XML FILE

STEP:4) BUILD THE JOB AND CHECK THE OUTPUT



<CHECKING THE SLACK , SONARQUBE , NEXUS AND THE URL OF OUR DEPLOYED PATH>

