**Pipeline Project**

sudo lsof -nP -iTCP -sTCP:LISTEN

PATH=$PATH:/usr/bin export PATH

**Jdk Installtion**

Step1: Download the JDK tar file which i have shared through one drive.

Step2: Transfer the file from local to remote cloud instance through WinSCP.

Step3: Execute the below commands for installation and configuratio

$ sudo su -

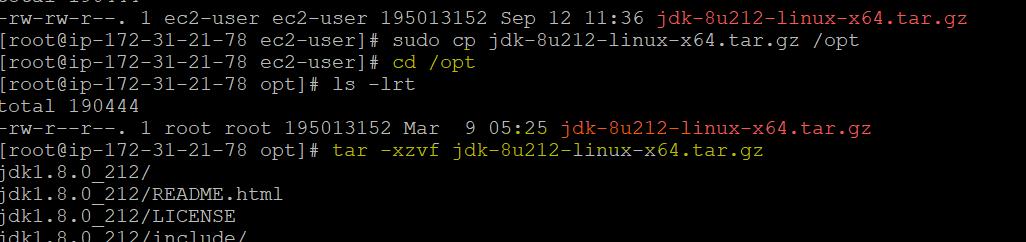
# cd /opt

cd /home/ec2-user

#sudo cp jdk-8u212-linux-x64.tar.gz /opt

cd /opt

# tar -xzvf jdk-8u212-linux-x64.tar.gz



# cd jdk1.8.0\_212/

# alternatives --install /usr/bin/java java /opt/jdk1.8.0\_212/bin/java 2

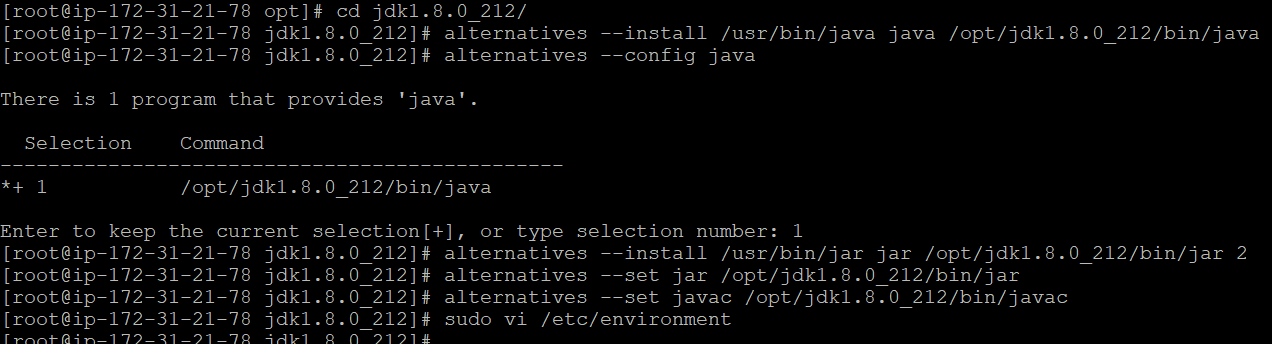
#alternatives --config java (give 1 number)

# alternatives --install /usr/bin/jar jar /opt/jdk1.8.0\_212/bin/jar 2

# alternatives --install /usr/bin/javac javac /opt/jdk1.8.0\_212/bin/javac 2

# alternatives --set jar /opt/jdk1.8.0\_212/bin/jar

# alternatives --set javac /opt/jdk1.8.0\_212/bin/javac



**Setting PATH variable in /etc/environment:**

# sudo vi /etc/environment

export JAVA\_HOME=/opt/jdk1.8.0\_212

export JRE\_HOME=/opt/jdk1.8.0\_212/jre

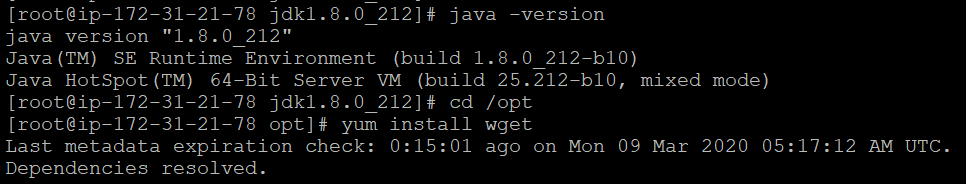
export PATH=$PATH:/opt/jdk1.8.0\_212/bin:/opt/jdk1.8.0\_212/jre/bin

Step4: The Java command below would work as a validation that the Alternatives have been setup correctly.

# source /etc/environment

# java –version

echo $JAVA \_HOME



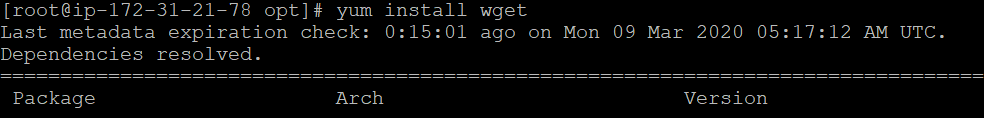
===================================================================================

**Maven Installation**

**1.Check for latest version (**[**https://maven.apache.org/download.cgi**](https://maven.apache.org/download.cgi)**)**

**2.cd /opt**

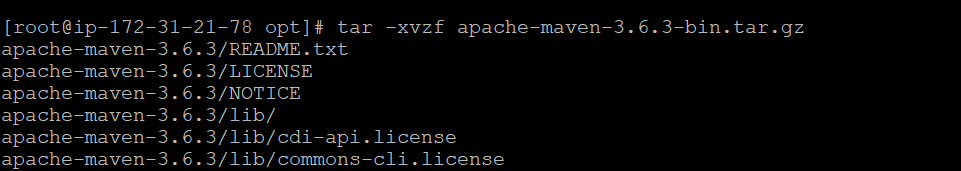
**yum install wget**



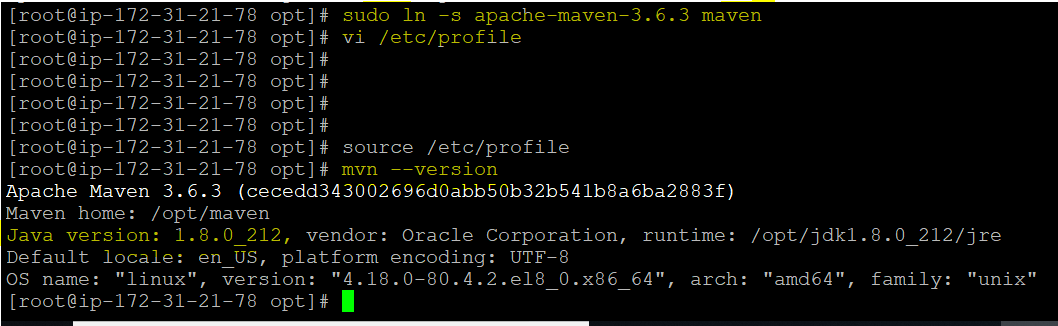
**3.sudo wget** [**http://mirrors.estointernet.in/apache/maven/maven-3/3.6.3/binaries/apache-maven-3.6.3-bin.tar.gz**](http://mirrors.estointernet.in/apache/maven/maven-3/3.6.3/binaries/apache-maven-3.6.3-bin.tar.gz)



**4.tar -xvzf apache-maven-3.6.3-bin.tar.gz**



**5.sudo ln -s apache-maven-3.6.3 maven**



**6.vi /etc/profile**

**export M2\_HOME=/opt/maven**

**export PATH=${M2\_HOME}/bin:${PATH}**

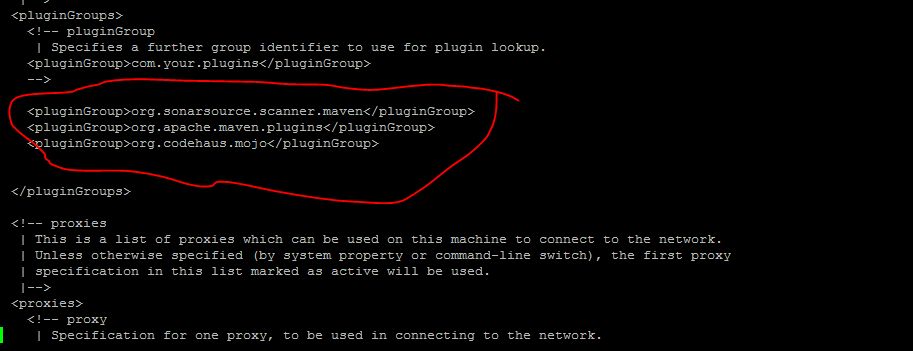
**7.source /etc/profile**

**Cd /opt/maven**

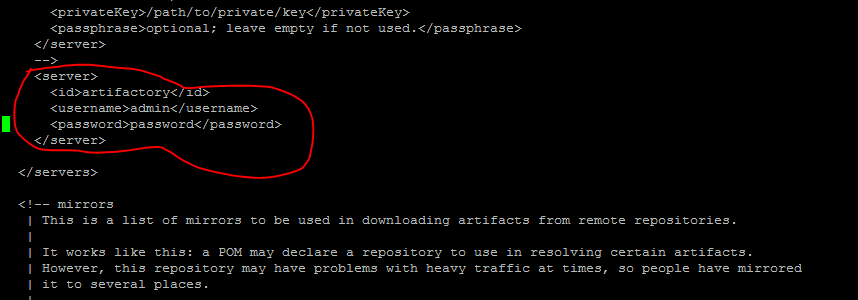
**mvn –version**

**add below plugin to maven settings.xml**

<pluginGroup>org.sonarsource.scanner.maven</pluginGroup>  
<pluginGroup>org.apache.maven.plugins</pluginGroup>  
 <pluginGroup>org.codehaus.mojo</pluginGroup>



For artifactory:



**<server>**

**<id>artifactory</id>**

**<username>admin</username>**

**<password>password</password>**

**</server>**

**================================================================**

**Installtion of Jenkins:**

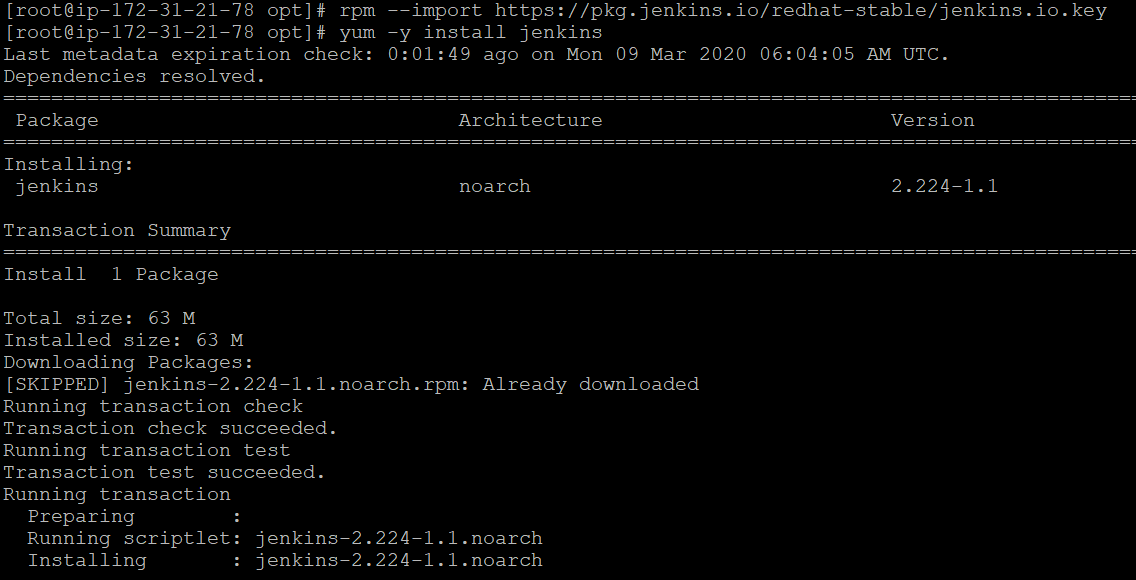
Always check for new pkg.

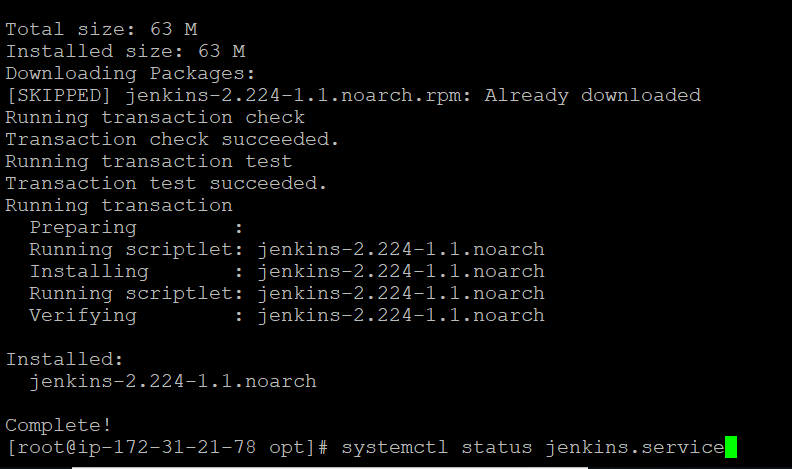
<https://pkg.jenkins.io/>

**yum update**

* sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
* sudo rpm --import <https://pkg.jenkins.io/redhat-stable/jenkins.io.key>
* yum install jenkins
* systemctl status Jenkins
* systemctl start Jenkins

yum install git





**Or**

**Use below steps**

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

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

sudo yum install jenkins

**Creating Swap memory:**

**Create temporary RAM (Memory) – root user**

1. Creating a swap file size of 4GB using dd command as shown below:

**dd if=/dev/zero of=/mnt/swapfile bs=1024 count=4097152**

1. Set permission on the file by using below command, make it readable only by root user

**chmod 600 /mnt/swapfile**

1. Using **mkswap** command and set up file for swap space:

**mkswap /mnt/swapfile**

1. Enable the swap file and add it to the system as a swap file by using below commands

**swapon /mnt/swapfile**

1. Edit the **/etc/fstab**file**,**this will enable the swap file to be mounted at boot time:

**vi /etc/fstab**

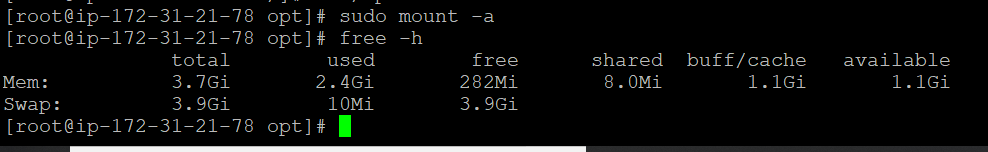
Add below lines in /etc/fstab:

**/swapfile           swap     swap    default  0 0**

**/mnt/swapfile   swap    swap    defaults 0 0**

**sudo mount –a**

Check using **free –h** command, we will be able see the spin up 4GB



**SonarQube Installation**

1. Go to /opt folder:

**cd /opt**

**with root user install the unzip utility.**

**yum install unzip**

1. Install sonarqube by using below commands:

**sudo wget**[**https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-7.6.zip**](https://apc01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fbinaries.sonarsource.com%2FDistribution%2Fsonarqube%2Fsonarqube-7.6.zip&data=02%7C01%7Cshrutiramesh.gurav%40hcl.com%7C1d49ed2cb21b4376cbb708d7bc3d6b84%7C189de737c93a4f5a8b686f4ca9941912%7C0%7C0%7C637184844292839617&sdata=AXIjlNoG3sqDcW%2FJjlexFSKz%2BPtGpITBSfz8h7%2FS%2BQA%3D&reserved=0)

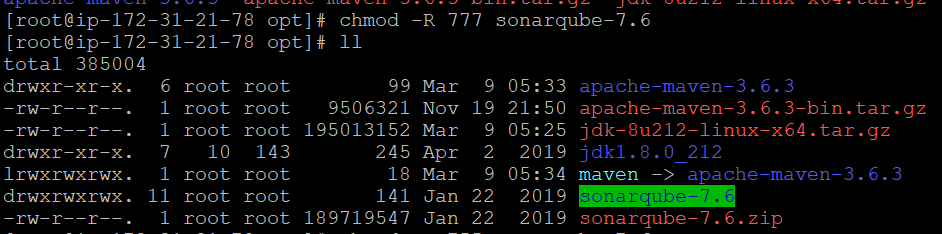
Unzip the software:

**sudo unzip sonarqube-7.6.zip**

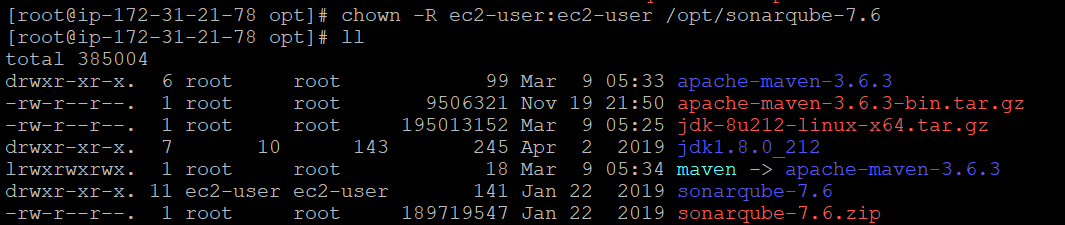
1. Provide full permissions to sonarqube.7.6. to run as non-root user.

**Run as ec2-user (non root)**

**sudo su ec2-user**



**sudo chown -R ec2-user:ec2-user /opt/sonarqube-7.6   [change the ownership]**



**cd /opt**

**sudo chmod 777 -R sonarqube-7.6          [provide access permission]**

**cd /opt/sonarqube-7.6/bin/linux-x86-64**

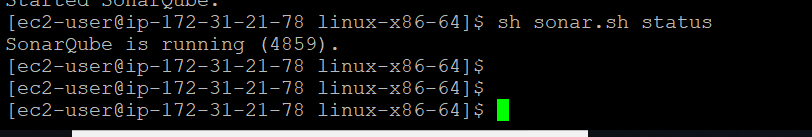
run **ls** command and check **sonar.sh** should be there

Execute Sonar using below command:

**sh sonar.sh start               or            ./sonar.sh start**

check the status using:

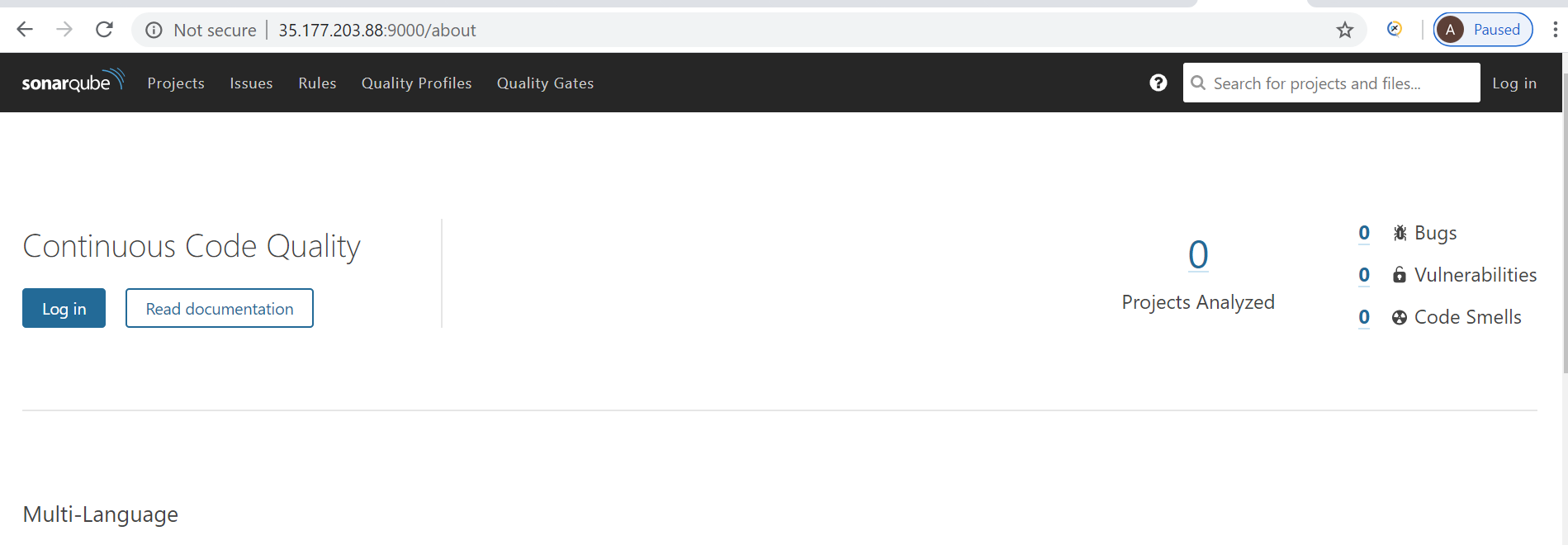
**sh sonar.sh status**



Once sonarqube is running successfully then go to

[http://3.134.106.95:9000](https://apc01.safelinks.protection.outlook.com/?url=http%3A%2F%2F3.134.106.95%3A9000%2F&data=02%7C01%7Cshrutiramesh.gurav%40hcl.com%7C1d49ed2cb21b4376cbb708d7bc3d6b84%7C189de737c93a4f5a8b686f4ca9941912%7C0%7C0%7C637184844292849560&sdata=krn%2BQol%2BAinee3VMBdjsC6Laz0v3M5dHyPNRpWa4Mlg%3D&reserved=0)              [ip is our aws running instance and 9000 is the port]

we will be able to see the below dashboard:



* **Webhook configuration: go to sonar**

**A----my account ----security---🡪abc generate token.**

**Copy that token and paste it in Jenkins --🡪configure system quality gate.**

* **Go to Administrator --**🡪**configuration---webhook**

[**http://100.25.221.152:8080/sonarqube-webhook**](http://100.25.221.152:8080/sonarqube-webhook)

* **go to manage Jenkins--**🡪**configure system**

|  |
| --- |
|  |
| **SonarQube servers** | | | |
|  | Environment variables | Enable injection of SonarQube server configuration as build environment variables |  |
|  | |  |  |
|  | | If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build. |  |
|  | SonarQube installations | |  |  |  |  | | --- | --- | --- | --- | |  | Name |  |  | |  | |  |  | |  | Server URL |  |  | |  | |  |  | |  |

|  |
| --- |
|  |
|  |  |
| **Quality Gates – Sonarqube** | | | |
|  |  | |  | | --- | |  | |  |  |  | |  | Name |  | [Help for feature: Name](http://100.26.169.53:8080/configure) | |  | |  |  | |  | | Make sure the name is unique value |  | |  |  | |  | |  | SonarQube Server URL |  | [Help for feature: SonarQube Server URL](http://100.26.169.53:8080/configure) | |  | |  |  | |  | | Default value is 'http://localhost:9000' |  | |  |  | |  | |  | SonarQube account token |  | [Help for feature: SonarQube account token](http://100.26.169.53:8080/configure) | |  | |  |  | |  | | Use token instead of user and password |  | |  |  | |  | |  | SonarQube account login |  | [Help for feature: SonarQube account login](http://100.26.169.53:8080/configure) | |  | |  |  | |  | | Default value is 'admin' |  | |  |  | |  | |  | SonarQube account password | Concealed |  | |  | |  |  | |  | | Default value is 'admin' |  | |  | Maximum waiting time (milliseconds) |  |  | |  | |  |  | |  | | Default value is '300000' or 5 minutes |  | |  | Time to wait next check (milliseconds) |  |  | |  | |  |  | |  | | Default value is '10000' or 10 seconds |  | |  |  |  |  | |  | |  |  | | Delete | | |  |   Add Sonar instance |  |

**\*if sonar qube is not running try to clear temp folder and restart it.**

**Or go to**

**cd /opt/sonarqube-7.6/lib**

**Run**

**java –jar sonar-application-7.6.jar**

**========================================================================**

**SonarQube Scanner installation Steps:**

<https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-3.3.0.1492-linux.zip>

# sudo wget <https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-3.3.0.1492-linux.zip>

# sudo unzip sonar-scanner-cli-3.3.0.1492-linux.zip

# vi /etc/environment

export SONAR\_SCANNER=/opt/sonar-scanner-3.3.0.1492-linux

export PATH=${SONAR\_SCANNER}/bin:${PATH}

sonar scanner path in Jenkins--🡪global tool configuration:

/opt/sonar-scanner-3.3.0.1492-linux

**==========================================================================**

**# Tomcat Installation Steps**

<https://tomcat.apache.org/download-90.cgi?Preferred=http%3A%2F%2Fapachemirror.wuchna.com%2F> find the latest release

**cd /opt**

**wget https://mirrors.estointernet.in/apache/tomcat/tomcat-9/v9.0.36/bin/apache-tomcat-9.0.36.tar.gz**

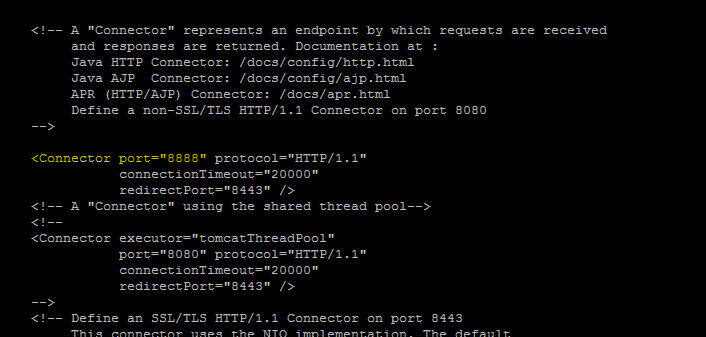
**# tar -xvf apache-tomcat-9.0.36.tar.gz**

**cd /opt/apache-tomcat-9.0.36/conf**

**cp server.xml server.xml\_bkp**

**vi server.xml**

**Change connector port 8080 to 8888**



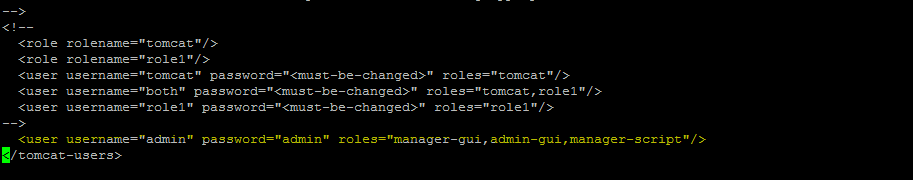
**cd /opt/apache-tomcat-9.0.36/conf**

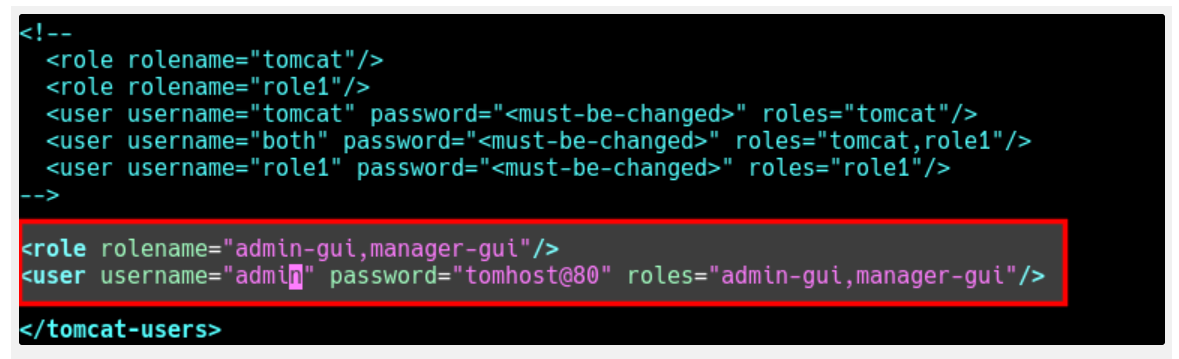
**cp tomcat-users.xml tomcat-users.xml\_bkp**

**vi tomcat-users.xml**

**Add the below line**

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





**cd /opt/apache-tomcat-9.0.36/webapps/manager/META-INF**

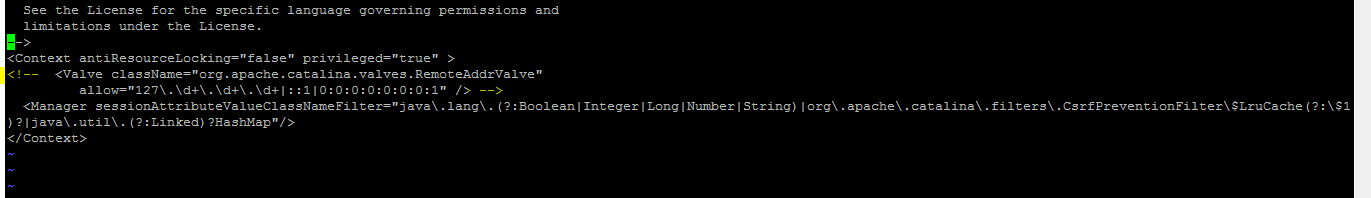
**vi context.xml**

**Comment below line**

**<Context antiResourceLocking="false" privileged="true" >**

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

**allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" /> -->**



**cd /opt/apache-tomcat-9.0.36/bin**

**Run below script**

**./** **startup.sh**

**./** **shutdown.sh**

**# systemctl daemon-reload**

**# systemctl start tomcat.service**

**# systemctl enable tomcat.service**

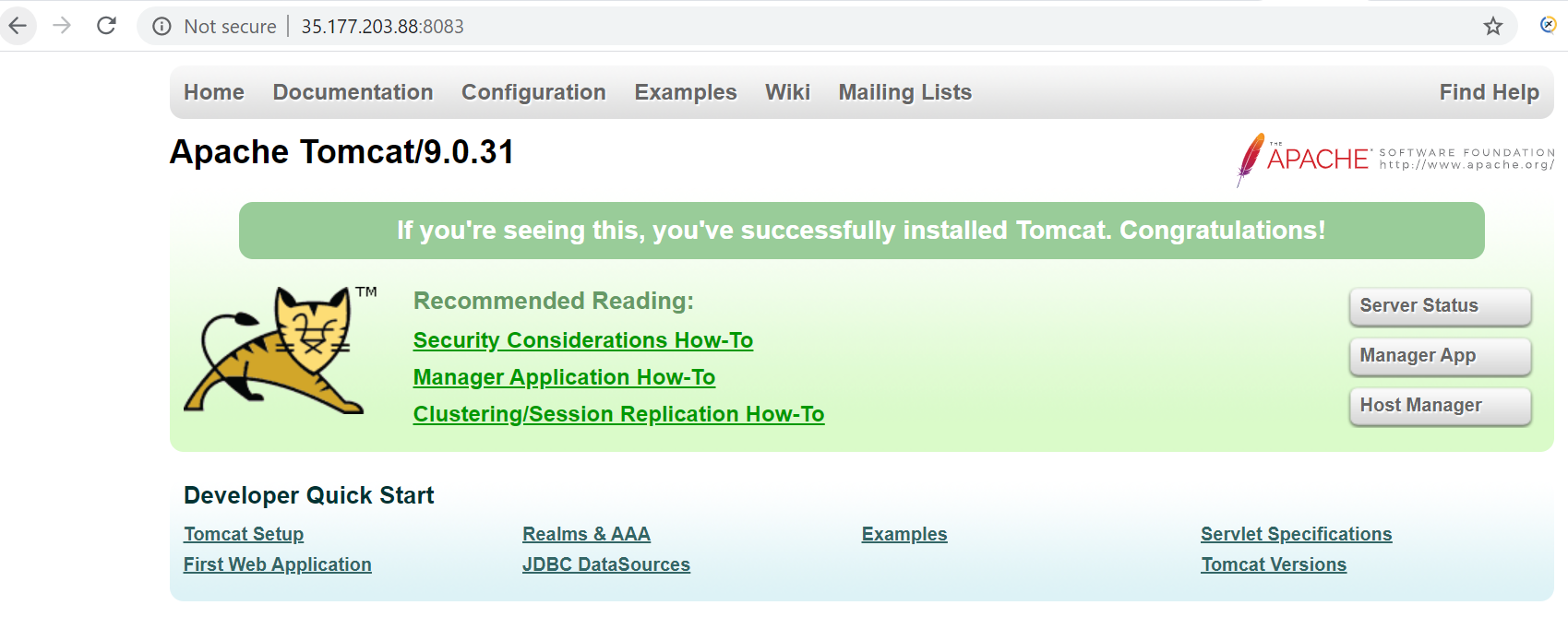
**# systemctl status tomcat.service**

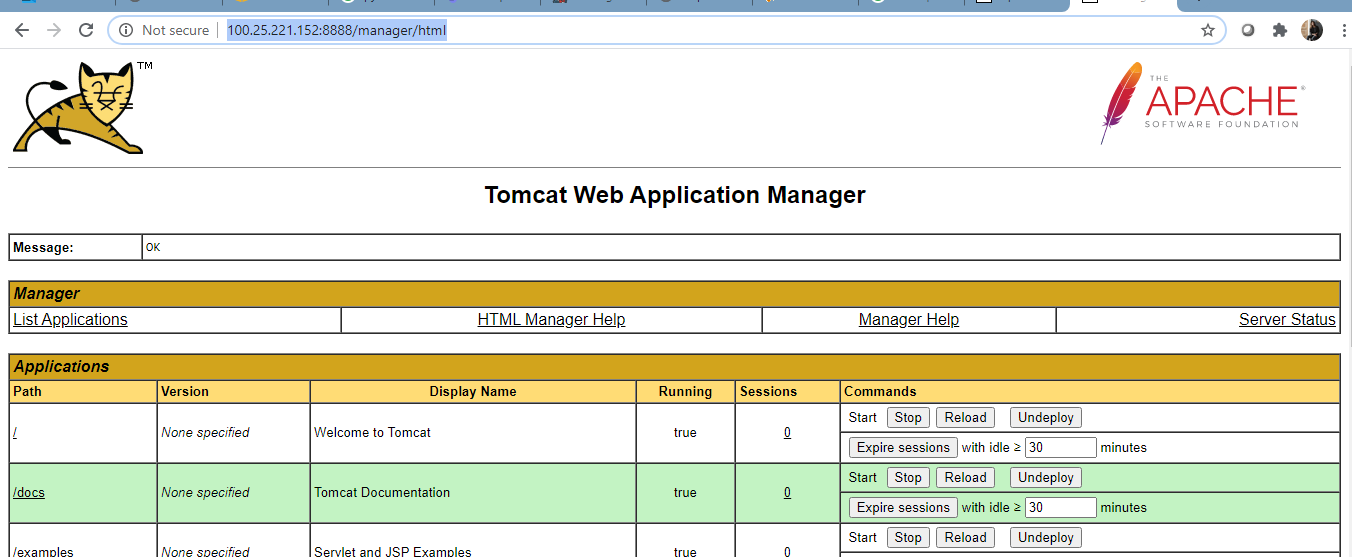
**# netstat –tlpn**

sudo lsof -iTCP -P -n | grep 8888

**systemctl restart tomcat.service**

<http://100.25.221.152:8888/manager/html>

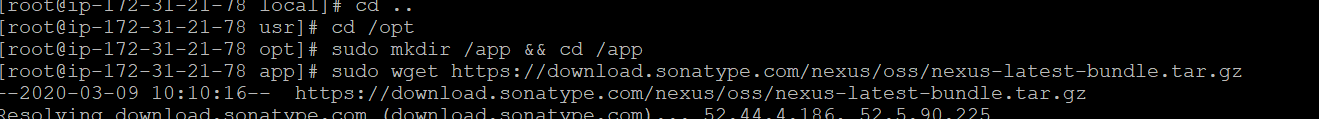




ln -s /opt/apache-tomcat-9.0.35/bin/startup.sh /usr/bin/tomcatup  
ln -s /opt/apache-tomcat-9.0.35/bin/shutdown.sh /usr/bin/tomcatdown

**Nexus Installtion:**

1. sudo mkdir /app && cd /app
2. sudo wget <https://download.sonatype.com/nexus/oss/nexus-latest-bundle.tar.gz>



**tar -xvf nexus-latest-bundle.tar.gz**

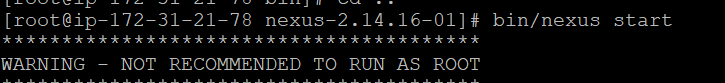


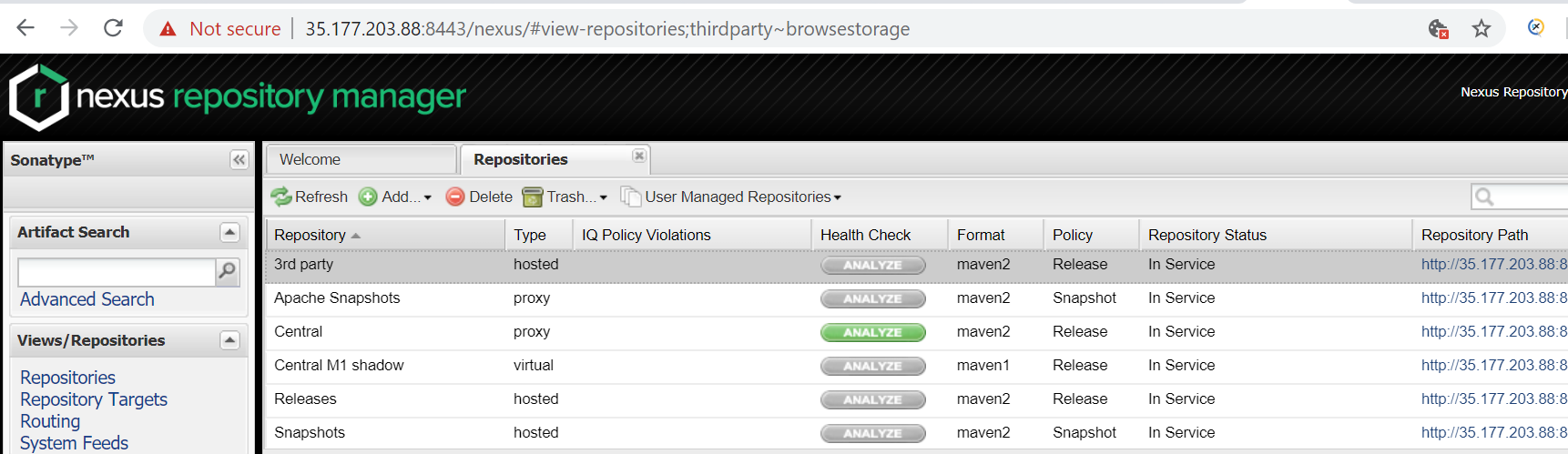
**cd /app/nexus-2.14.18-01/**

**4 . bin/nexus start**

**5. export RUN\_AS\_USER=root**

**bin/nexus start**





**Nexus:8081**

**Sonar:9000**

**Tomacat:8888**

**Jenkins8080**

NODEJS INSTALLATION

1.cd /opt

2.yum install -y gcc-c++ make

           After complete, install npm:

3.yum install nodejs -y

4.npm install yarn -g

                  To Configure with Jenkins

                            Download NodeJS plugin in Jenkins:

                             Jenkins-->Manage Jenkins-->Manage Plugins

5.Configure Node Js:

                      Jenkins-->Manage Jenkins-->Global Tools Configuration

                          NODEJS

                           /usr/bin

6.vi /etc/environment

                     export NODE\_HOME=/usr/bin/node

                      export PATH=${NODE\_HOME}/bin:${PATH}

7.node --version

8.npm -version

---------------------------------------------------------------------------------------------------------------------------

ANGULAR INSTALLATION

npm install -g polymer-cli --unsafe-perm

npm install -g @angular/cli (enable angular configuration)

**DB \_Installation**

**1.Refer link for latest version of MySQL : https://www.tecmint.com/install-latest-mysql-on-rhel-centos-and-fedora/**

**2.sudo su -**

**3.cd /opt**

**4.wget https://repo.mysql.com/mysql80-community-release-el8-1.noarch.rpm**

**5.yum localinstall mysql80-community-release-el8-1.noarch.rpm**

**6.yum repolist enabled**

**7.yum install mysql-commuinty-server (don't run this command)**

**8.yum install mysql-server mysql**

**9.systemctl start mysqld**

**10.systemctl status mysqld**

**11.mysql\_secure\_installation (Y,Low,Yes,then all NO)**

**12.mysql -u root –p (press enter)**

**13.pwd : H@ckatha0n/HAckath0n**

**14.mysql> create database ingdb;**

**15.mysql> show databases;**

**16.mysql> use ingdb;**

**17.mysql> CREATE USER 'inguser1'@'%' IDENTIFIED BY 'H@ckath0n';**

**Query OK, 0 rows affected (0.01 sec)**

**18.mysql> GRANT ALL PRIVILEGES ON \*.\* to 'inguser1'@'%';**

**Query OK, 0 rows affected (0.00 sec)**

**19.mysql> flush privileges;**

**Query OK, 0 rows affected (0.00 sec)**

**20.mysql> exit**

**Bye**

**21. systemctl start mysqld**

**=================================================**

1. **Install python**

# sudo yum -y update

# sudo yum install -y python36

1. **Install virtual environment**

# python3.6 -m venv myvirtual\_venv (here myvirtual\_venv is the name of the directory)

# source myvirtual\_venv/bin/activate

It will enter into the virtual environment

# python –version

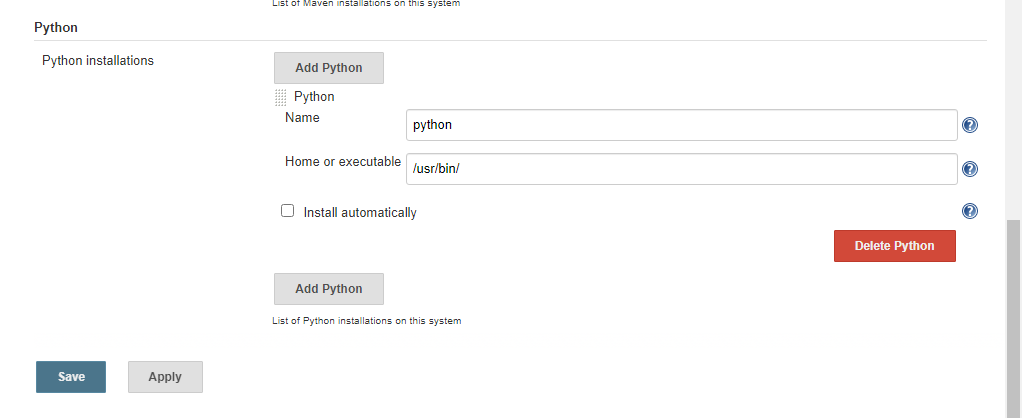
# pip install --upgrade pip

# pip list (to see the list of packages available)

Go To manage jenkins🡪manageplugin-🡪install

* Python plugin
* Shiningpanda plugin

Go to jenkins global tool configuration-🡪set the path of python



**Install nginx reverse Proxy and access tomcat only through ip:**

**Steps:**

1. sudo yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
2. sudo yum install -y epel-release
3. sudo yum update -y
4. sudo yum install nginx -y
5. sudo nginx -v
6. sudo systemctl start nginx
7. sudo systemctl enable nginx
8. sudo systemctl status nginx
9. http://server-ip
10. **cd /etc/nginx**

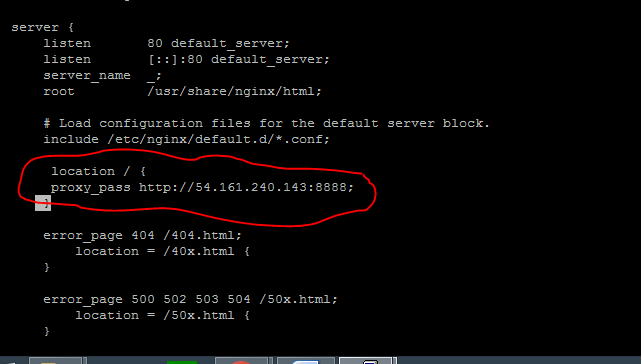
**vi nginx\_conf**

**change the server tag as below:**

**location / {**

**proxy\_pass** [**http://54.161.240.143:8888**](http://54.161.240.143:8888)**;**

**}**



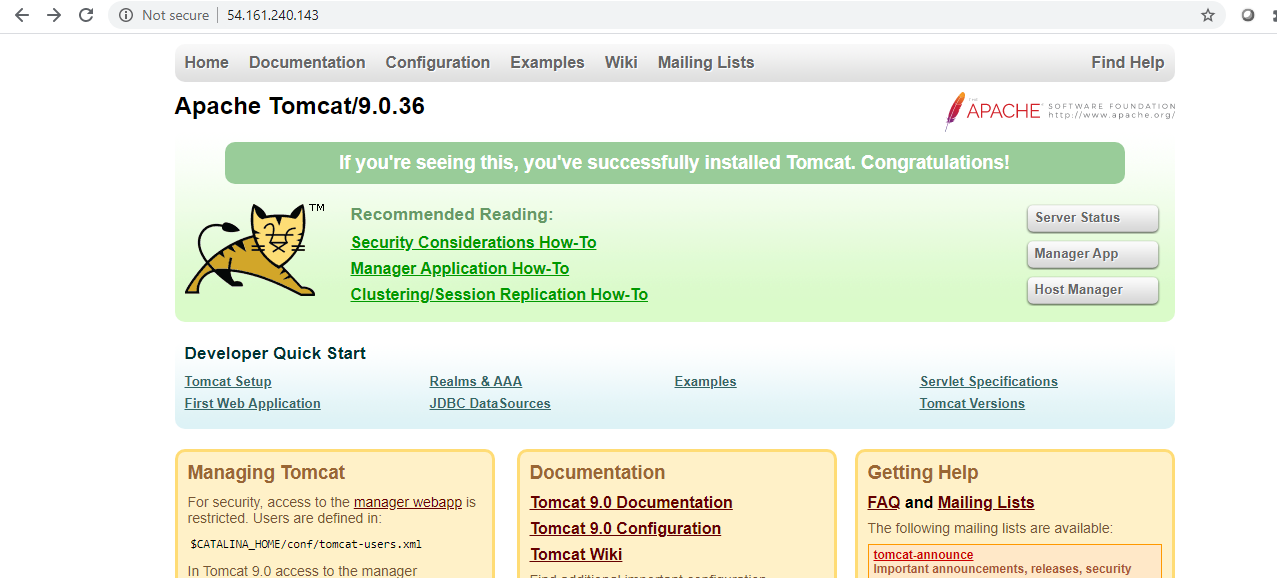
**Above is the http:ip:tomcat port.**

**11.** systemctl restart nginx

setsebool -P httpd\_can\_network\_connect=1

access the tomcat by providing only ip.

54.161.240.143



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

Some extra command for nginx: no need to execute

**sudo yum install firewalld**

**systemctl start firewalld**

**sudo firewall-cmd --zone=public --add-port=8888/tcp –permanent**

**sudo firewall-cmd --zone=public --add-port=80/tcp --permanent**

**sudo firewall-cmd --reload**

**setsebool -P httpd\_can\_network\_connect 1**

**Install apache reverse Proxy and access tomcat only through ip**

<https://cwiki.apache.org/confluence/display/HTTPD/TomcatReverseProxy>

**STEPS.**

**1) rpm -ivh https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm**

**2) yum repolist all**

**3) yum list httpd**

**4) yum install httpd**

**5) httpd –v**

**6) systemctl start|stop|status httpd.service**

**7) systemctl enable|disable httpd.service**

**Reverse Proxy setup:**

**Cd /etc/httpd/conf.d**

**touch tomcat.conf**

**vi tomcat.conf**

**copy paste below tag**

**<VirtualHost \*:80>**

**ProxyRequests Off**

**ProxyPass / http://54.161.240.143:8888/**

**ProxyPassReverse / http://54.161.240.143:8888/**

**</VirtualHost>**

**Execute the below command**

**httpd -k graceful**

**httpd –t**

**setsebool -P httpd\_can\_network\_connect 1**

**Execute the below command if firewall is installed**

**Note:if it is installed you have to enable each port.**

**sudo yum install firewalld**

**systemctl start firewalld**

**sudo firewall-cmd --zone=public --add-port=8888/tcp –permanent**

**sudo firewall-cmd --zone=public --add-port=80/tcp --permanent**

**sudo firewall-cmd --reload**

**setsebool -P httpd\_can\_network\_connect 1**

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

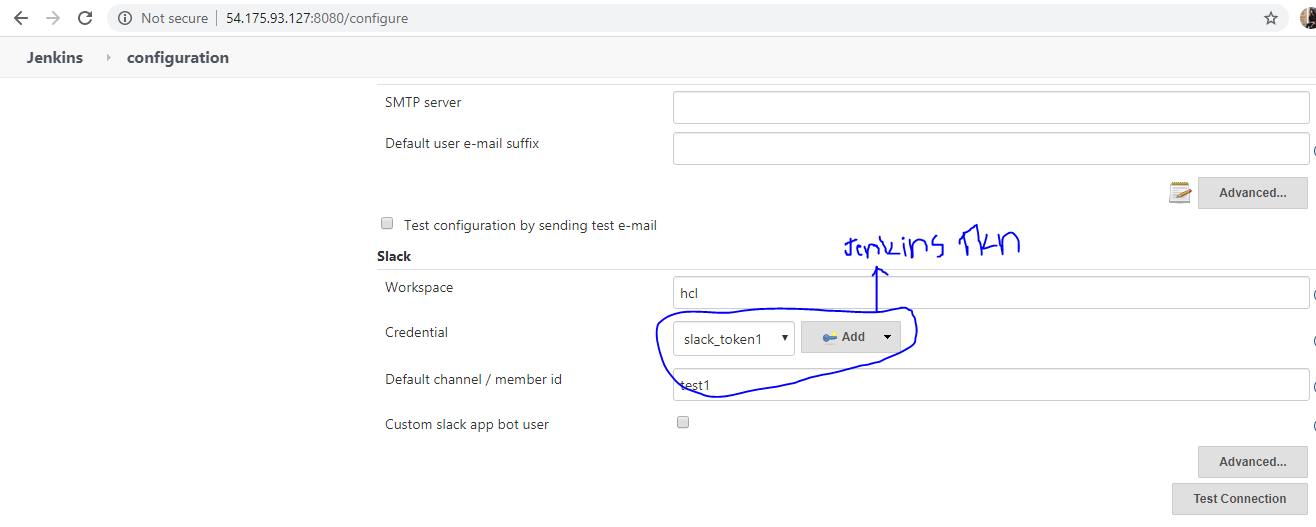
cp /var/lib/jenkins/workspace/Nginx\_Tomcat/target/demo-0.0.1-SNAPSHOT.war /opt/apache-tomcat-9.0.36/webapps/demo-0.0.1-SNAPSHOT.war'

**c8CYpVz8oFcYW7Ha3zAcQ7i9-------------slack**

**slack Configuration**

**1.Create a channel.**

**2.add the CI integration as a Jenkins--------------🡪app🡪search Jenkins CI. Select the respective created channel to add it. It will generate the token copy it to configuration----🡪slack configuration---🡪as a secret text.**

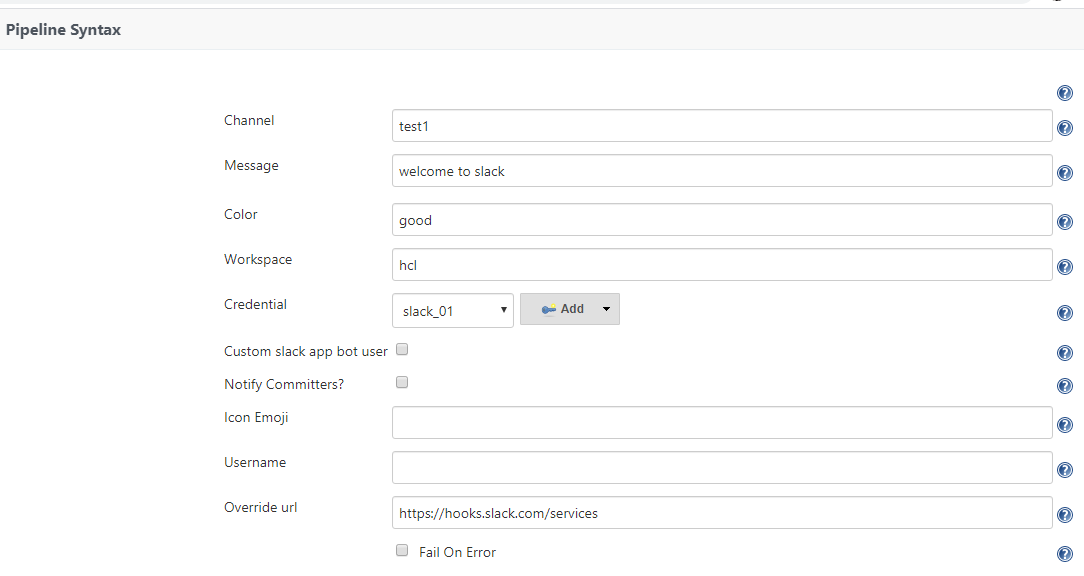


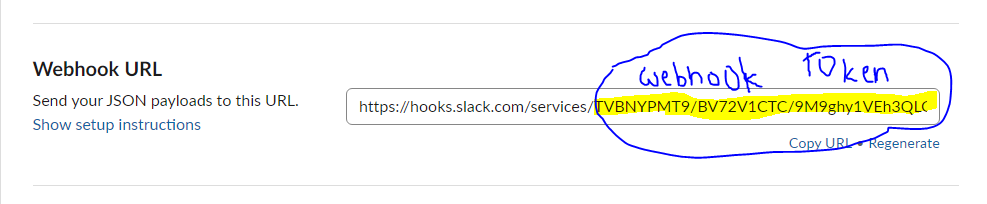
**Example:**

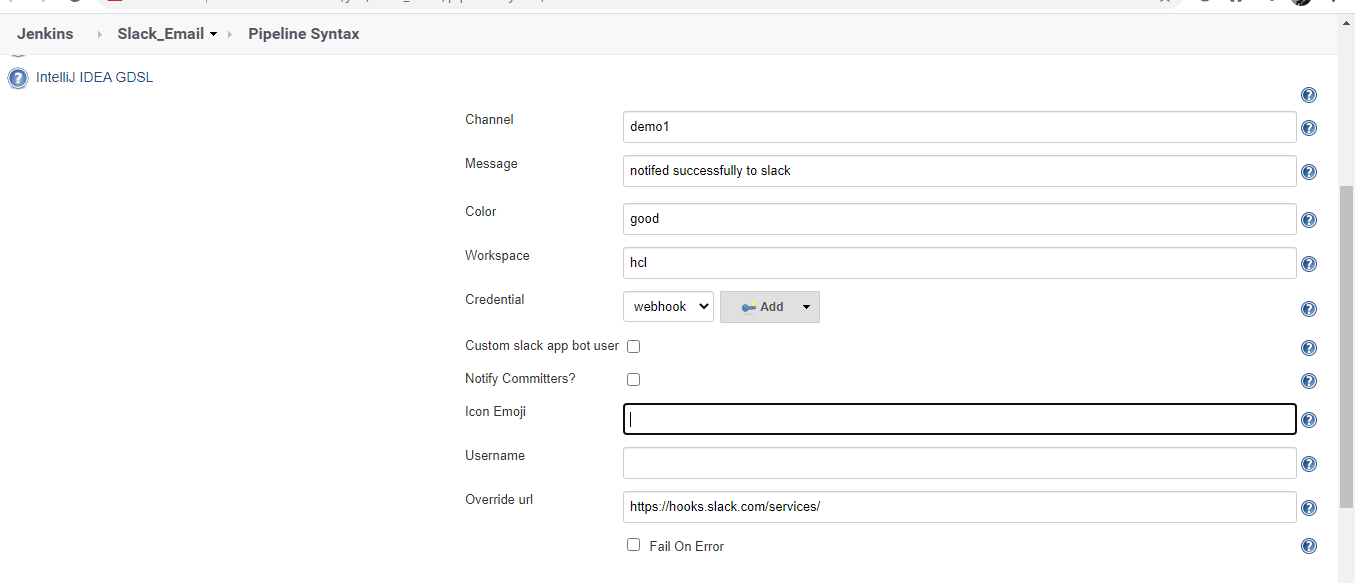
**Webhook configuration:**

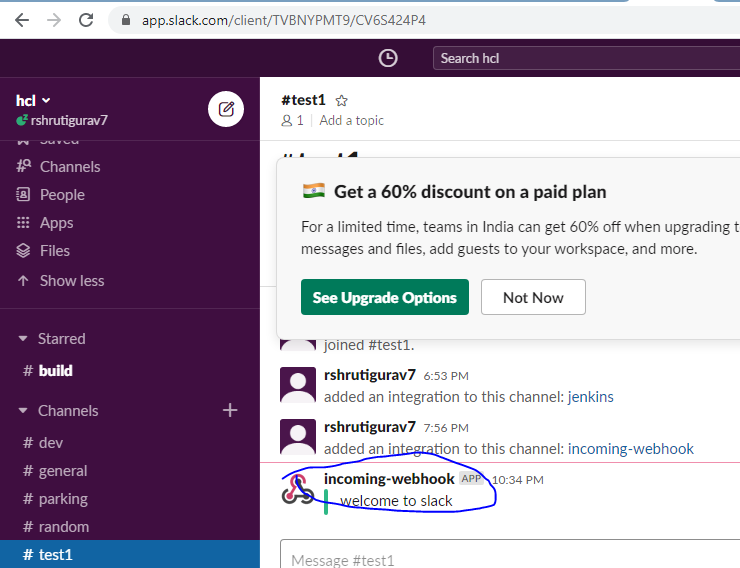
**Add the webhook inbound integrator to created channel.It will generate the webhook url.Copy the url and add it to url tab while generating the script.and add the the next generated token to secret text ,click on generate script.**

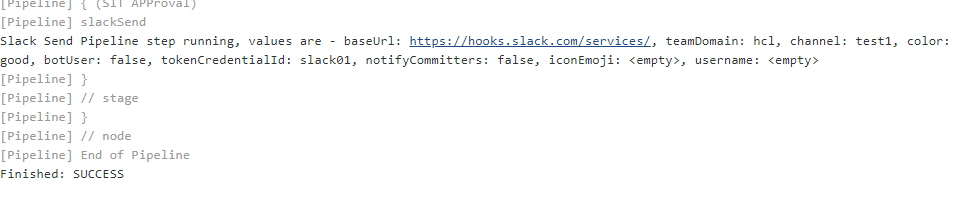
**Example:**



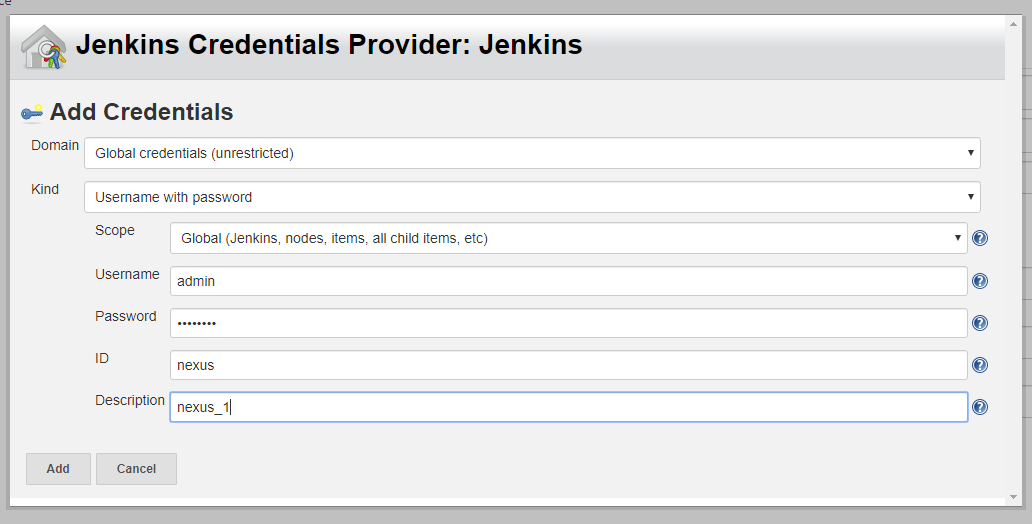


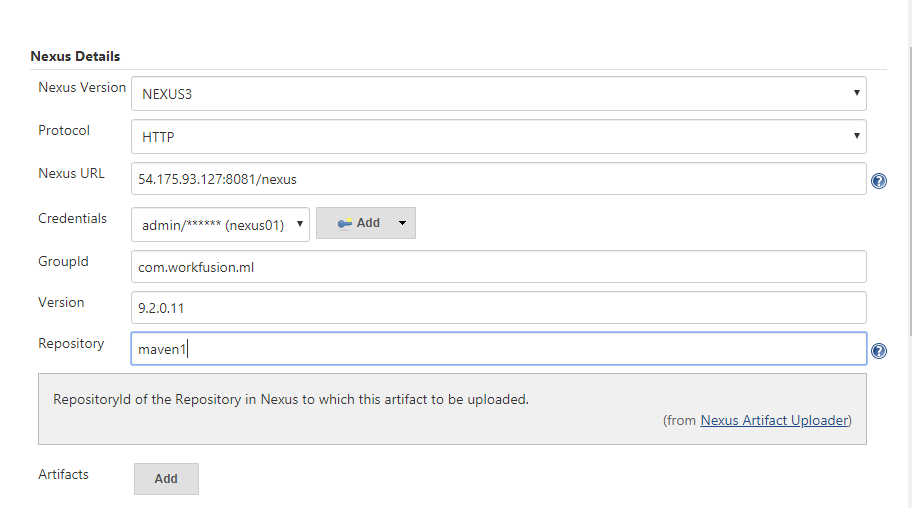


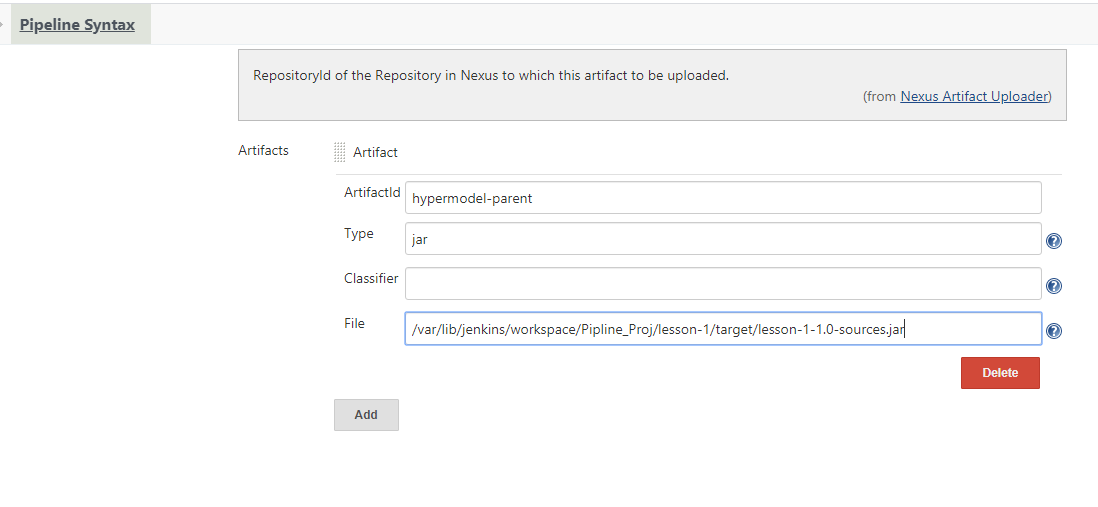


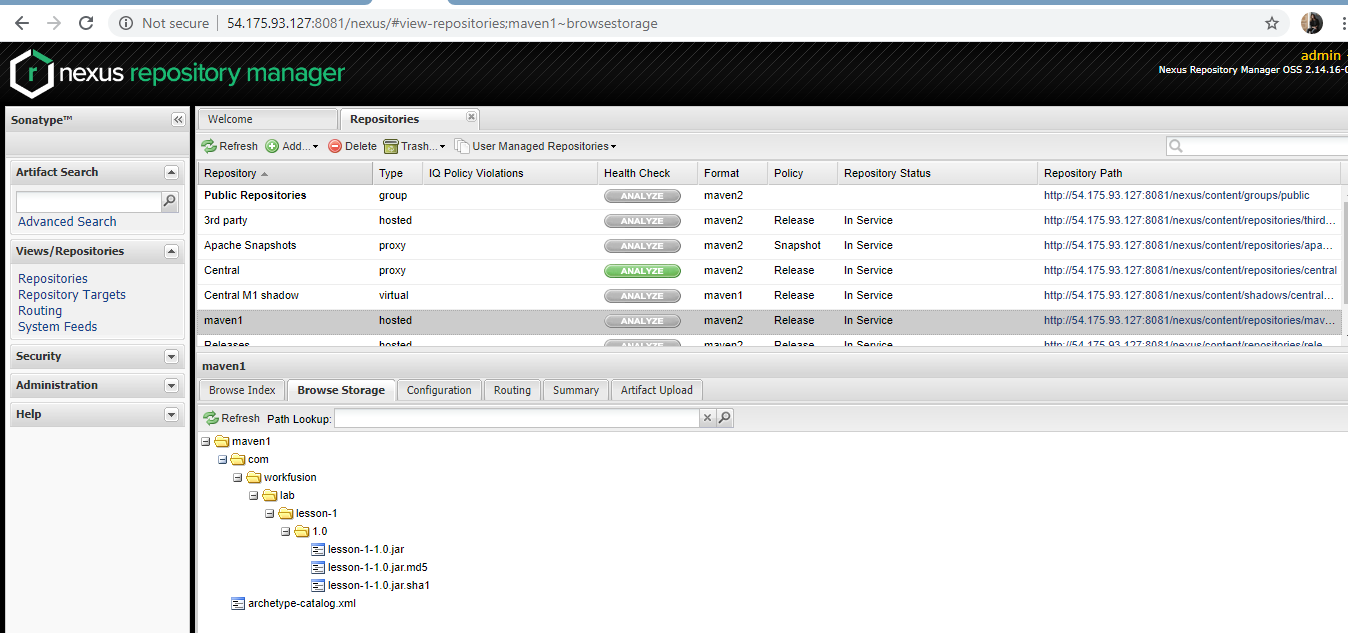


**Nexus :Script generator**









**Below Configuration should do in mange Jenkins--🡪global tool configuration**

|  |  |
| --- | --- |
|  | |
| **JDK** | | | | | | | | | | | | | |
|  | | | | | | | | | | | | |  |
|  | | JDK installations | | | | | | Add JDK  JDK   |  |  |  |  | | --- | --- | --- | --- | |  | Name |  |  | |  | |  |  | |  | JAVA\_HOME |  |  | |  | |  |  | | | | | |  |
|  |
| **Git** | | | | | | | | | | | | | | | |
|  | Git installations | | | | |  |  |  |  | | --- | --- | --- | --- | | **Git** | | |  | |  | Name |  |  | |  | |  |  | |  | Path to Git executable |  | [Help for feature: Path to Git executable](http://100.26.169.53:8080/configureTools/) | |  | |  |  | |  |  | |  | |  |  |  |  | | Install automatically | | | [Help for feature: Install automatically](http://100.26.169.53:8080/configureTools/) | |  |  |  |  | | | | | | | | | | |  |
| **Maven** | | | | | | | | | | | |
|  | | | | | | | | |  | | |
|  | | | | Maven installations | | | Add Maven  Maven   |  |  |  |  | | --- | --- | --- | --- | |  | Name |  |  | |  | |  |  | |  | MAVEN\_HOME |  |  | |  | |  |  | |  |  |  |  | | |  | | |
|  | | |
| **SonarQube Scanner** | | | | | | | | | | |
|  | | | | | | | | | |  |
|  | | | SonarQube Scanner installations | | | Add SonarQube Scanner  SonarQube Scanner   |  |  |  |  | | --- | --- | --- | --- | |  | Name |  |  | |  | |  |  | |  | SONAR\_RUNNER\_HOME |  |  | |  | |  |  | | | | |  |

<http://18.189.22.51:9093/parking/swagger-ui.html>

ip:prt/context-path/swagger-ui.html

~