Configuring Apache with Tomcat:

Installation and Apache Configuration with Tomcat:

Step 1:

```
yum install wget httpd java-11-openjdk-devel -y
service httpd start
service httpd status
```

Step2:

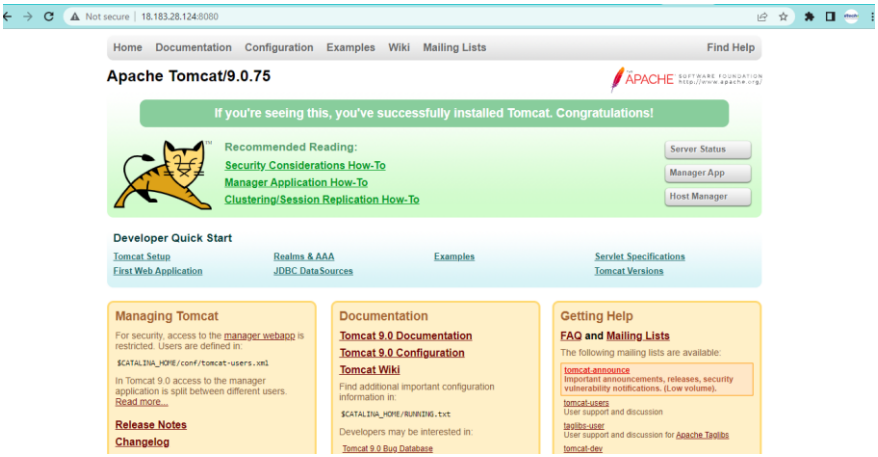
```
wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.75/bin/apache-tomcat-9.0.75.tar.gz
tar -xvf apache-tomcat-9.0.75.tar.gz
cp -pr apache-tomcat-9.0.75 tomcat1
cp -pr apache-tomcat-9.0.75 tomcat2
ref : https://crunchify.com/how-to-run-multiple-tomcat-instances-on-one-server/
```

Step3:

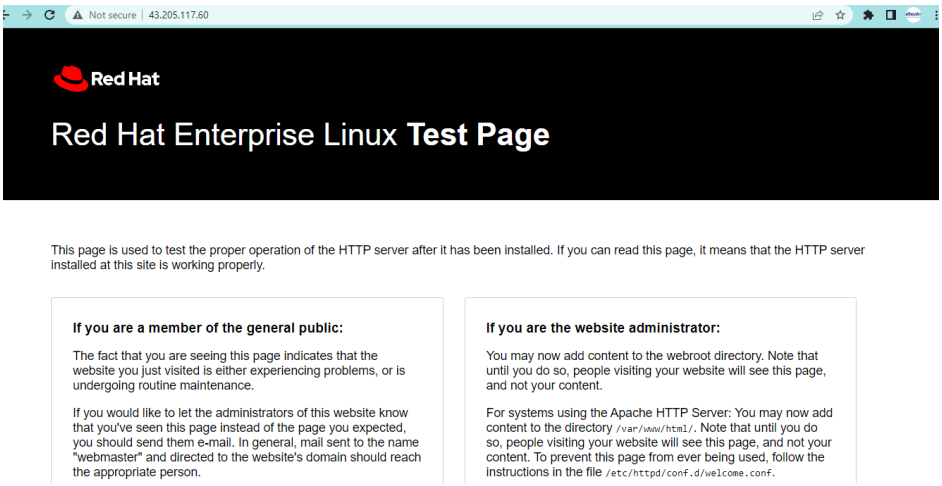
```
sudo vi /etc/httpd/conf.d/proxy.conf
<VirtualHost *:80>
<Proxy balancer://mycluster>
   BalancerMember http://13.233.80.182:9090/
   BalancerMember http://13.233.80.182:8080/
</Proxy>
   ProxyPreserveHost On
   ProxyPass / balancer://mycluster/
   ProxyPassReverse / balancer://mycluster/
</VirtualHost>

service httpd restart
```

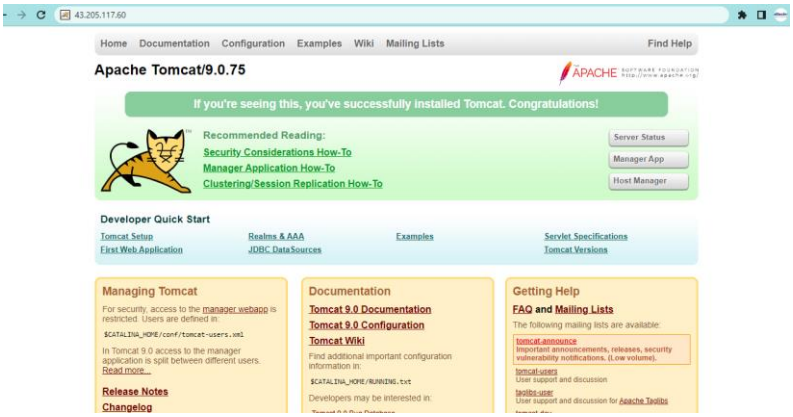
Accessing with 8080 checking it working or not:



Configure Apache with 80 port:



Whenever u enter Ip then it should go automatically hit Tomcat:



Configuring the SSL Certificate on Tomcat:

- keytool -genkey -keyalg RSA -alias <local hostname> -keystore tomcat.jks -validity 90 -keysize 2048

Example:

- keytool -genkey -keyalg RSA -alias ip-172-31-10-159.ap-northeast-1.compute.internal -keystore tomcat.jks -validity 90 - keysize 2048

Set a password and remember it will be used in the next steps:

- after that setup first name and last name one should be <local hostname which was given in alias above>
- then go to /home/ec2-user/conf/
- vi server.xml

- remove the 8080 port then paste the below content in the same format :

<Connector

port="8080" maxHttpHeaderSize="8192" maxThreads="150" minSpareThreads="25"

maxSpareThreads="75" enableLookups="false" disableUploadTimeout="true" acceptCount="100"

scheme="https" secure="true" SSLEnabled="true" clientAuth="false" sslProtocol="TLS" keyAlias="local-hostname"

keystoreFile="/root/tomcat.jks" keystorePass="Password which u had set above"

/>

Example:

<Connector

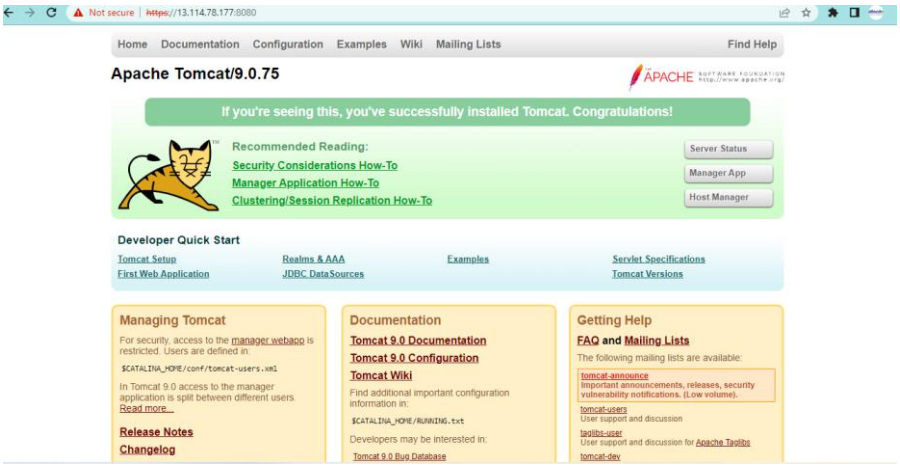
port="8080" maxHttpHeaderSize="8192" maxThreads="150" minSpareThreads="25"

maxSpareThreads="75" enableLookups="false" disableUploadTimeout="true" acceptCount="100"

scheme="https" secure="true" SSLEnabled="true" clientAuth="false" sslProtocol="TLS" keyAlias="ip-172-31-10-159.ap-northeast- 1.compute.internal"

keystoreFile="/root/tomcat.jks" keystorePass="anand@123"

/>



- Pre-install httpd and mod_ssl:
- Installing SSL on Apache:
- yum install httpd mod_ssl -y

- go to the cd /etc/httpd and create directory name certs and execute below commands

Go to the certs directory

- openssl genrsa -out server.key 2048

- openssl req -new -key server.key -out server.csr ||||||---> openssl req -in server.csr -text --- this command to see the file in human readle format

The above command will ask for information about our SSL certificate

- openssl x509 -req -in server.csr -signkey server.key -days 365 -out server.crt |||||--> openssl x509 -in server.crt -text --- this command to see the file in human readle format

- Go to the <VirtualHost _default_:443> this line and change like this <VirtualHost *:443>

- And check below lines has to be same on this file

- SSLEngine on
- SSLcertificatefile "/etc/httpd/certs/server.crt"
- SSLcertificatekeyfile "/etc/httpd/certs/server.key"

save and exit

and restart httpd

And after doing this to make work load balance work in Apache go to :

- /etc/httpd/conf.d/ssl_conf
- And paste the below code:

```
<Proxy balancer://mycluster>
  BalancerMember http://13.233.80.182:9090/
```

```
        BalancerMember http://13.233.80.182:8080/
    </Proxy>
        ProxyPreserveHost On
        ProxyPass / balancer://mycluster/
        ProxyPassReverse / balancer://mycluster/
```

 # Installing and setup of SONARQUBE:

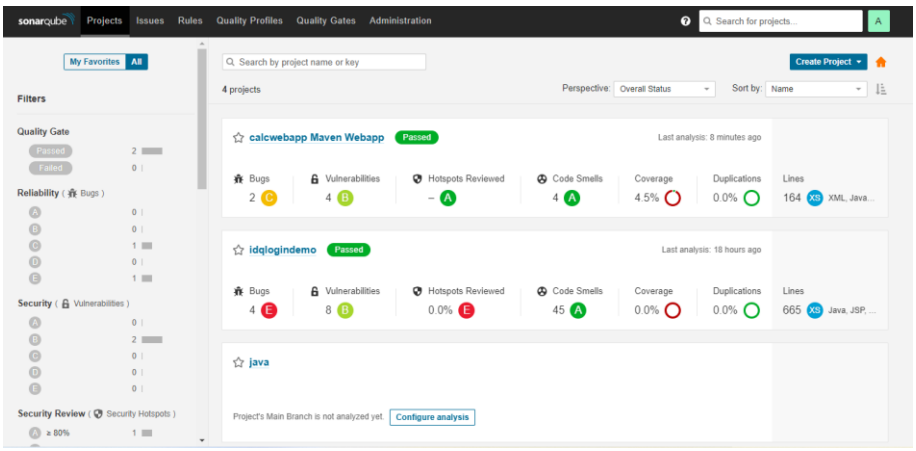
```
Pre-Requirements of Installation:
  It must have 4 CPUs or 4 GB of RAM workspace required to work on SonarQube.

  step1:
      Install java
      sudo java-11-openjdk-devel wget unzip -y
   step2:
      Go to SonarQube Downloads:
          wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-
9.1.0.47736.zip

    step3:
        Install unzip command.
        Yum install unzip
    step4:
        sudo unzip https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-
9.1.0.47736.zip
```

Then:

```
Use the T2 Medium instant type :
      1. Add user to perform in Sonar cube (Don't run as Root for the best practices).
          useradd sonar
      2. passwd Shashi
          set password
      3. Set the permissions using Chown:
          chown -R sonar:sonar sonar/  (it creates all same permission through out the
sonarqube directory)
      4. Then go to bin folder:
         cd bin
          cd linux.
         (before starting the sonar shift to the user and start otherwise we will face the errors)
       5. Then start the sonar.
          sh sonar.sh start
       6.To check the status of the sonar:
          sh sonar.sh status and check the port (ip:9090).
          User id & passwd:
          admin & admin
```



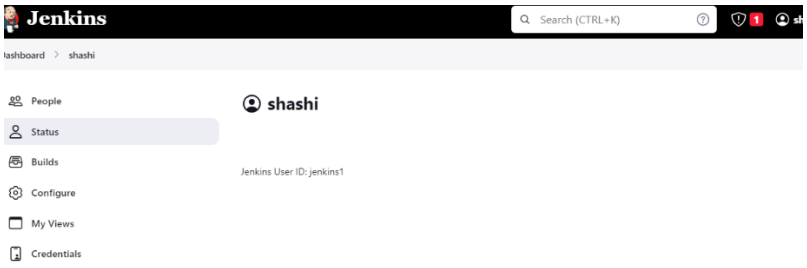
After logging in:

Go to administration and create a token which later used in Jenkins Integration:

- Jenkins Installation on Red hat:

- Install from this link: https://pkg.jenkins.io/redhat-stable/

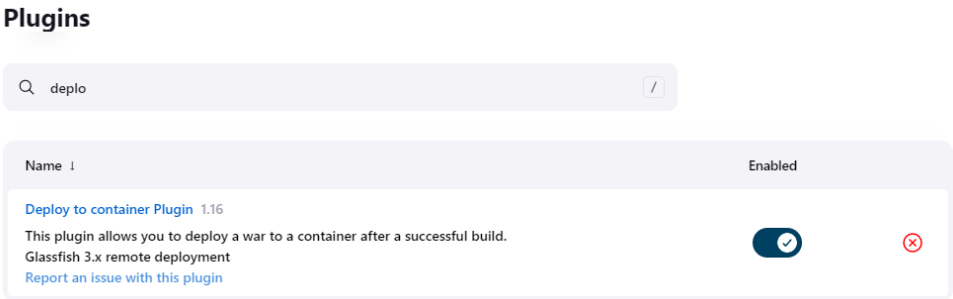
Integrating & Deploying the Java Project on Tomcat Using Jenkins:



- Pre-Setup for Integrating and Deploying:

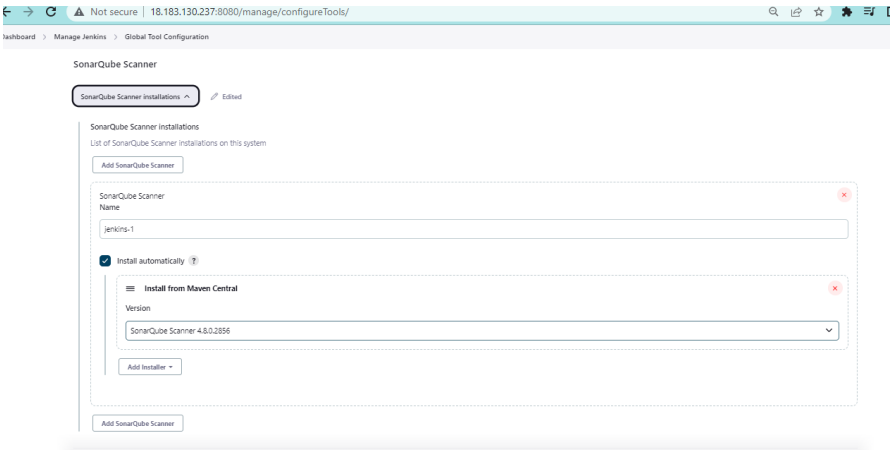
Step1:

- Install the Plugins which you can see below:



Step2:

- Go to Global Configuration Tools and try to write same name of the token name in sonar :



Step 3:

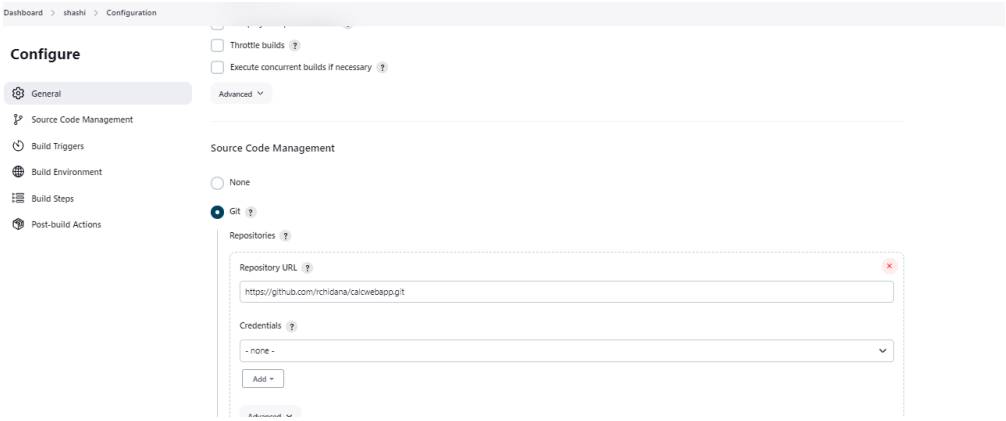
- Add your Tomcat user details which u edited in tomcat-users.xml file.

The example I have added:

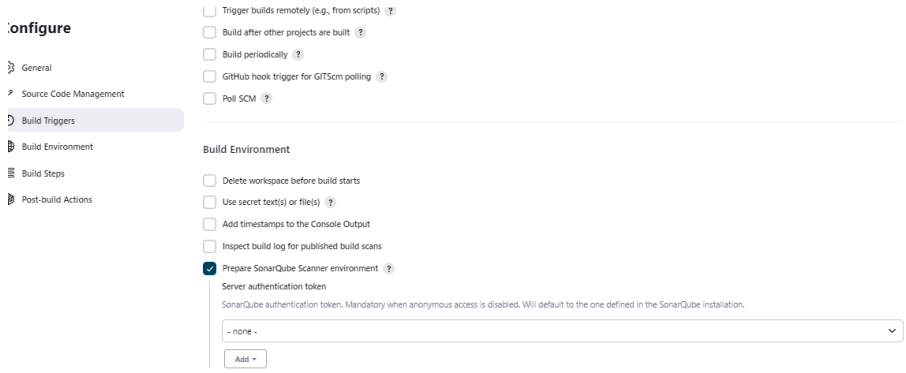
- Username: admin & password: s3cet
- Then add: http://18.183.28.124:8080/
- Sonar one u already added in the above step so need not be required.

Step 4:

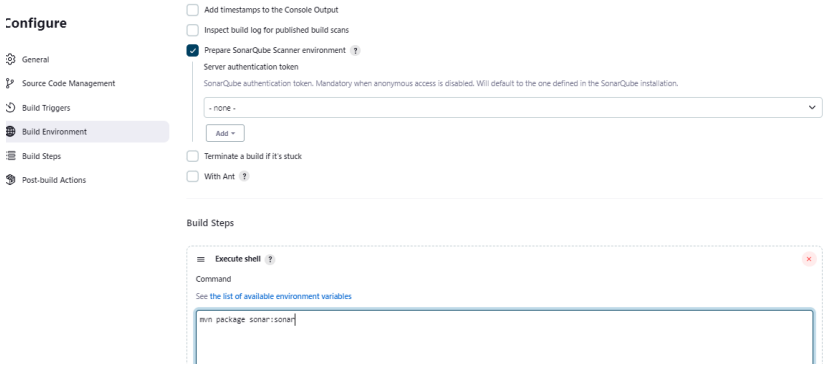
- Add a new project and create a name:

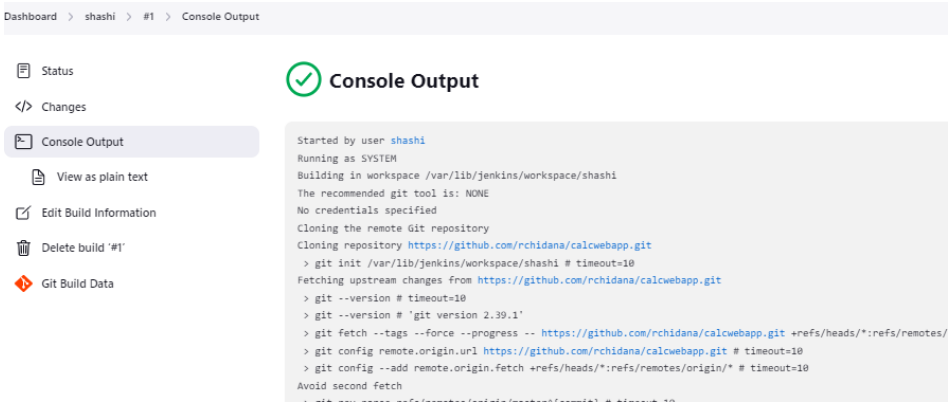


Then:



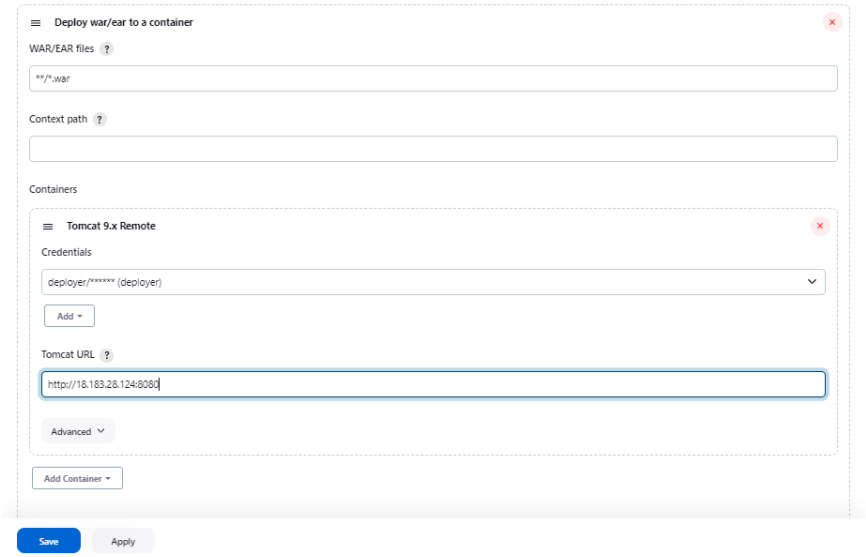
And next step:

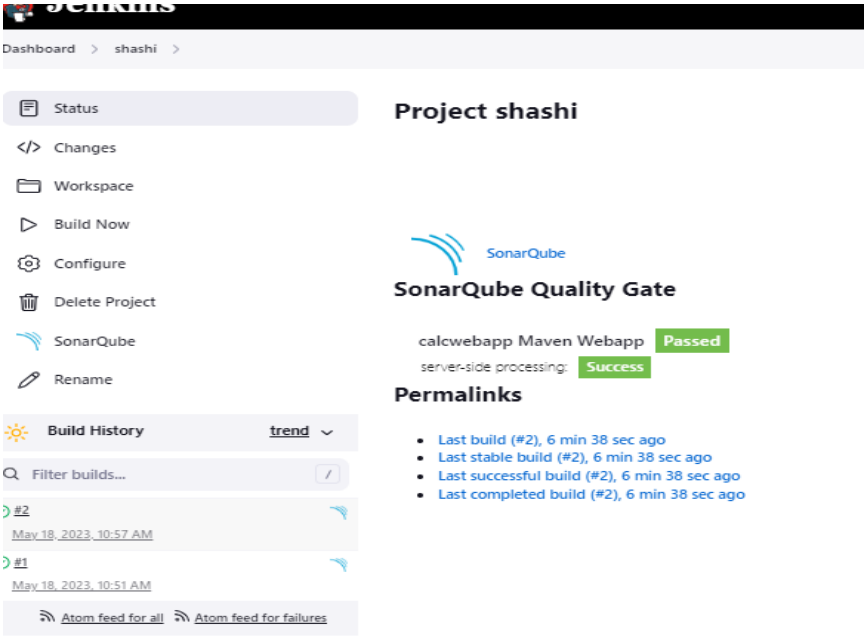




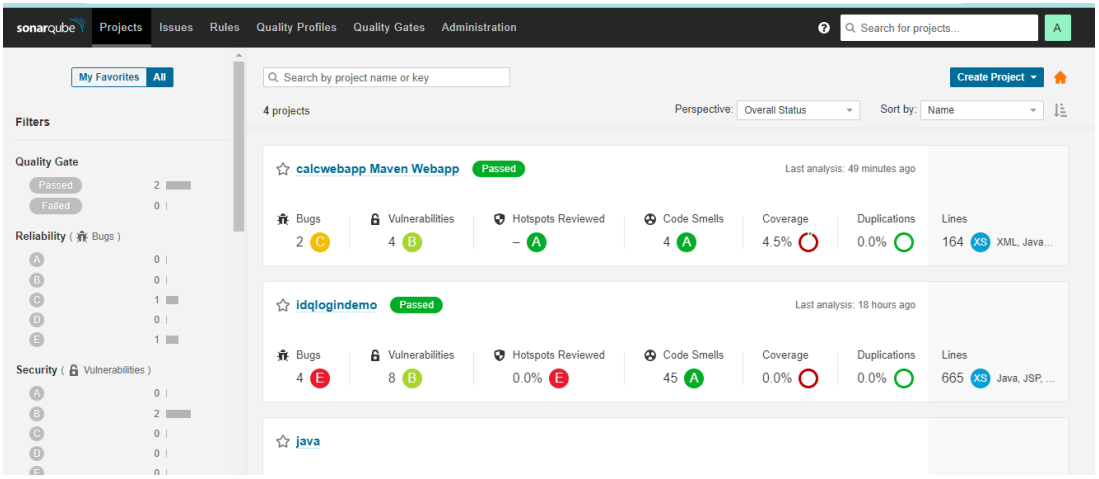
Follow the above steps and make the Integration complete:

- Then we will go into Deployment:
- Now go into the Project and click configure and go to the last step Post Build Actions:



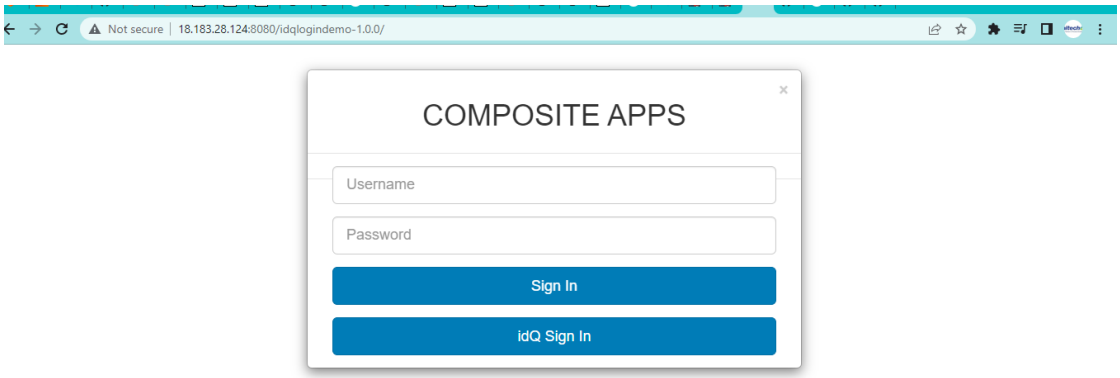


- **In SonarQube the bugs, Quality of code, Code smells will be available.**



- The below two Projects I have deployed on Tomcat:

Project1:



Project2: