Devops project **Name:Suresh .P /9047817516**

Step:1 1)create EC2-amazon linux –t2.medium

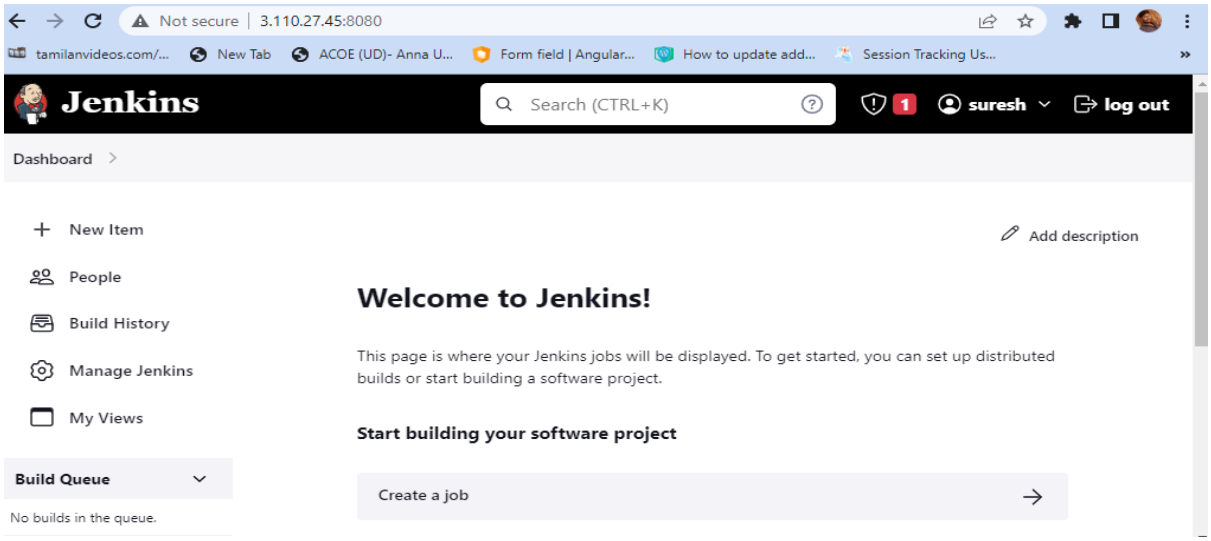
Step:2 2)git,docker , java,Jenkins installed ec2

Step:3 3) sudo cat /var/lib/jenkins/secrets/initialAdminPassword

Step:4 4)open browser public ip:8080 Enter

```
e.
[root@ip-172-31-13-225 lib]# sudo cat /var/lib/jenkins/secrets/initialAdminPassword
26fddb1500a3402687e06cf5b20aeeea
[root@ip-172-31-13-225 lib]# mkdir test1
[root@ip-172-31-13-225 lib]# cd test1
[root@ip-172-31-13-225 test1]# vi file1
[root@ip-172-31-13-225 test1]# cd
[root@ip-172-31-13-225 ~]# git version
git version 2.39.2
[root@ip-172-31-13-225 ~]# docker version
Client:
 Version:           20.10.17
 API version:       1.41
 Go version:        go1.18.6
 Git commit:        100c701
 Built:             Sat Dec  3 04:13:49 2022
 OS/Arch:           linux/amd64
 Context:           default
 Experimental:      true
Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?
```

Step:5 login Jenkins and create pipeline job



Step:6 6) cd opt wget maven

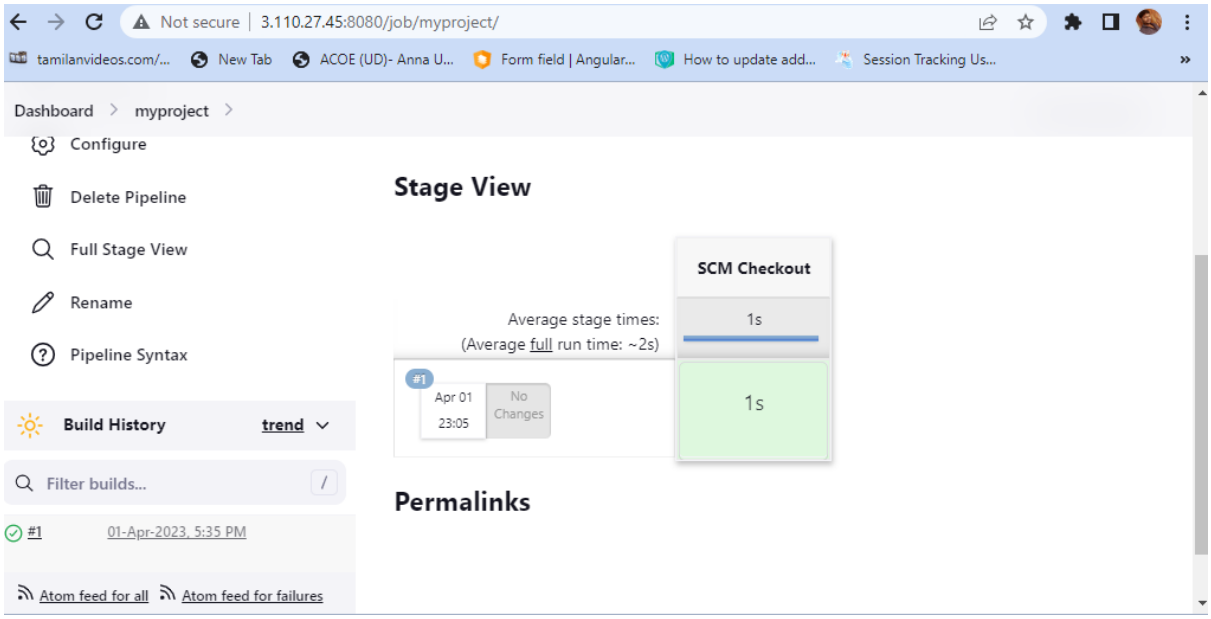
```
[root@ip-172-31-13-225 ~]# cd opt
[root@ip-172-31-13-225 opt]# ls -lrt
total 8828
-rw-r--r-- 1 root root 9039409 Mar 15 10:00 apache-maven-3.9.1-bin.tar.gz
drwxr-xr-x 6 root root      99 Apr  1 17:44 apache-maven-3.9.1
[root@ip-172-31-13-225 opt]# pwd
/root/opt
[root@ip-172-31-13-225 opt]#
```

Step:7

Open git hub repo – select to open – Jenkins file – Then copy the code Then past the pipeline then build now.

Sample Jenkins govey code below:

```
node{
  stage('SCM Checkout'){
    git 'https://github.com/SureshKirshna/my-app.git'
  }}
```



Step:7

#Cd opt

Opt# sudo wget https://dlcdn.apache.org/maven/maven-3/3.9.1/binaries/apache-maven-3.9.1-bin.tar.gz

Opt# tar -xvzf apache-maven-3.9.1-bin.tar.gz

Opt#ls –lrt

[root@ip-172-31-41-142 opt]# cd apache-maven-3.9.1

[root@ip-172-31-41-142 apache-maven-3.9.1]# pwd

/opt/apache-maven-3.9.1

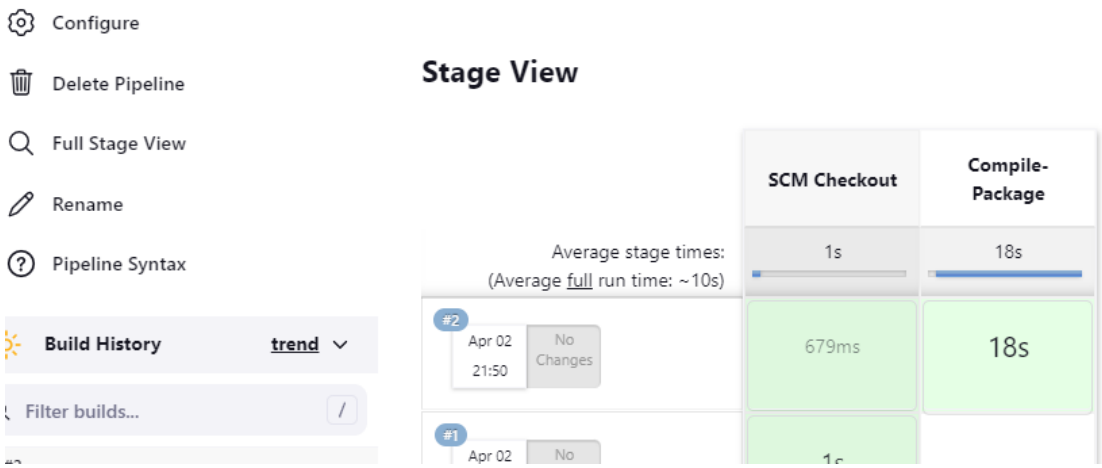
(maven is a third party tool. So go to install plugin)

1)Go to Jenkins – manage Jenkins – manage plugin – type :maven integration plugin – Install without restart.

2) Go to Jenkins – manage Jenkins – global tool configuration – name : maven(maven allays name) –untik install – give maven url: /opt/apache-maven-3.9.1

Then go to git hub repo. select the Jenkins file. Copy the maven code.

```
stage('Compile-Package'){

    def mvnHome =  tool name: 'maven3', type: 'maven'
    sh "${mvnHome}/bin/mvn clean package"
        sh 'mv target/myweb*.war target/newapp.war'
}
```



Step:8

To verify docker version and status.

#sudo systemctl status docker ->inactive stage

#sudo systemctl start docker ->active stage

Error :

```
+ docker build -t sureshdurga/myweb:0.0.2 .
Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the
docker daemon running?
```

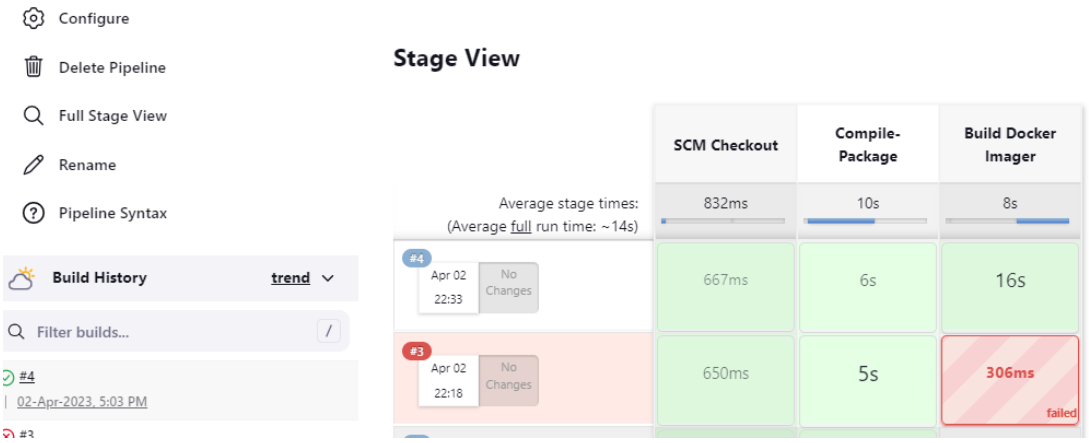
#chmod 777 /var/run/docker.sock

 # docker build –t dockerhubname/imagename

Ex: docker build –t sureshdurga/ myweb:0.0.2

Then goto paste the code in pipeline.

```
stage('Build
Docker
Imager'){
                sh 'docker build -t sureshdurga /myweb:0.0.2 .'
                }
```

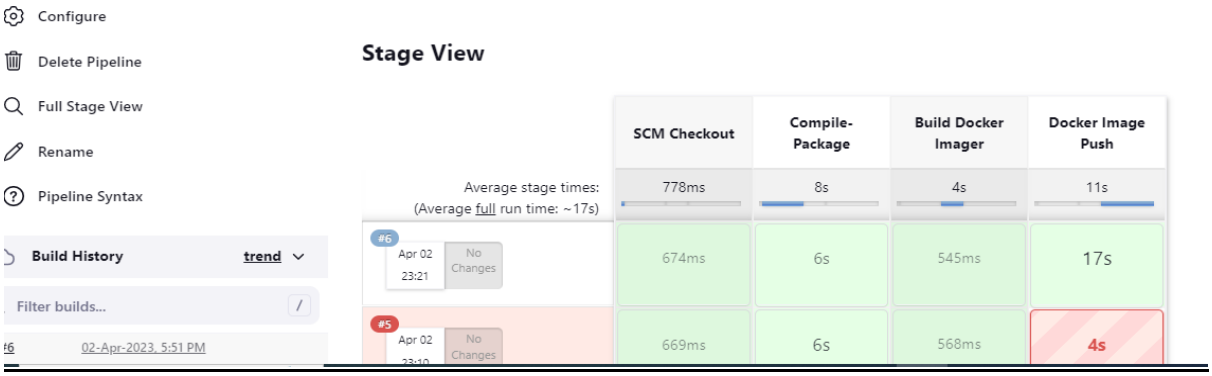


Step:9

1)Go to Jenkins – manage Jenkins – manage credentials – credentials – system – global credentials – add credentials - kind : secret text – Secret **– docker hub password - ID:dockerPass – descption : docker hub password.**

#docker push

```
stage('Docker
Image Push'){
                withCredentials([string(credentialsId: 'dockerPass',
            variable: 'dockerPassword')]) {
                sh "docker login -u sureshdurga -p ${dockerPassword}"
                 }
                sh 'docker push sureshdurga/myweb:0.0.2'
                }
```



Step:10

1) **Ceate EC2-amazon linux - T2 midium -aws**

for sonar qube and my sql

sudo yum install java-1.8.0

 # sudo update-alternatives --config java

#java –version

Install sonarqube step:

 # cd /opt

 # wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-6.7.6.zip

#sudo unzip sonarqube-6.7.6.zip

#sudo mv sonarqube-6.7.6 sonar

```
[ec2-user@ip-172-31-0-32 sonar]$ ls -lrt
total 12
drwxr-xr-x 2 root root   24 Nov 20  2018 temp
drwxr-xr-x 2 root root    6 Nov 20  2018 logs
drwxr-xr-x 4 root root   40 Nov 20  2018 extensions
drwxr-xr-x 2 root root   24 Nov 20  2018 data
-rw-r--r-- 1 root root 7651 Nov 20  2018 COPYING
drwxr-xr-x 2 root root   50 Nov 20  2018 conf
drwxr-xr-x 9 root root 4096 Nov 20  2018 web
drwxr-xr-x 9 root root  140 Nov 20  2018 lib
drwxr-xr-x 7 root root  150 Nov 20  2018 elasticsearch
drwxr-xr-x 8 root root  136 Nov 20  2018 bin
[ec2-user@ip-172-31-0-32 sonar]$
```

```
[ec2-user@ip-172-31-0-32 sonar]$ cd conf
[ec2-user@ip-172-31-0-32 conf]$ ls -lrt
total 24
-rw-r--r-- 1 root root  3311 Nov 20  2018 wrapper.conf
-rw-r--r-- 1 root root 17786 Nov 20  2018 sonar.properties
[ec2-user@ip-172-31-0-32 conf]$ vi sonar.properties
```

Root user only can be modify the sonar .properties file. so you open to root user or specify set root permission then modified and saved. Otherwise con't be change the sonar. properties file. So compulsory give linux file permission

[ec2-user@ip-172-31-0-32 conf]$ vi sonar. Properties

Go to vi editor in sonar .properties file . Then do to some change below lines

sonar.jdbc.username=sonar

sonar.jdbc.password=sonar

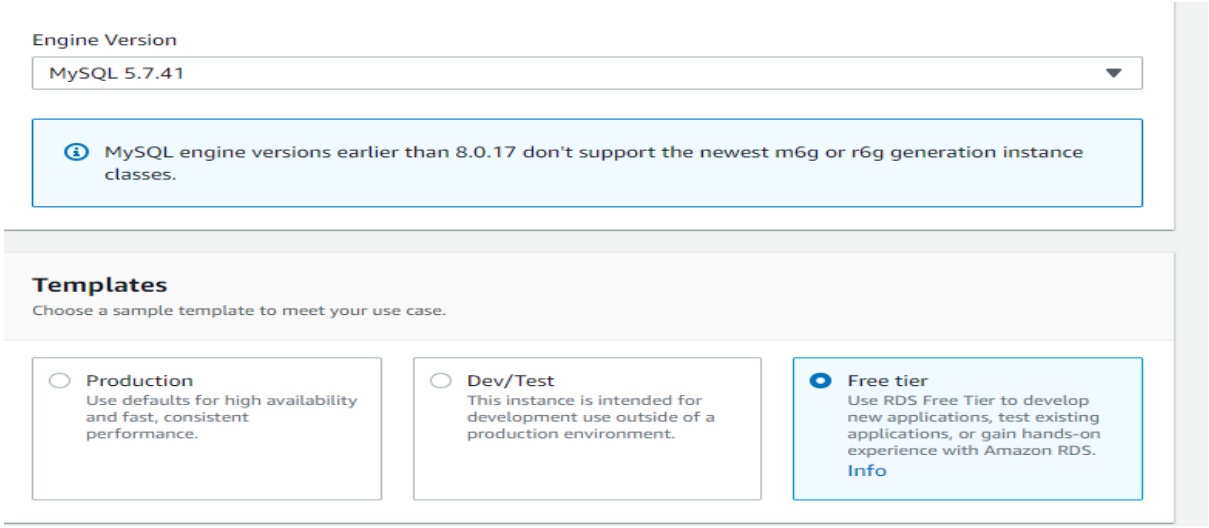
sonar.jdbc.url=jdbc:mysql://sonardatabase.cwovsgxyibkj.ap-south-1.rds.amazonaws.com:3306/sonar?useUnicode=true&characterEncoding=utf8&rewriteBatchedStatements=true&useConfigs=maxPerformance&useSSL=false

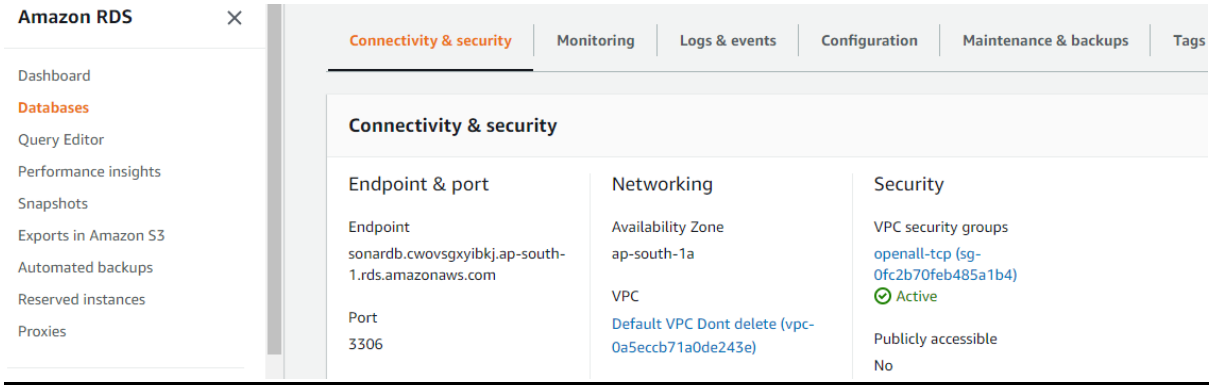
sonar.web.host=0.0.0.0

sonar.web.port=9000

next step : Then go to bin :

Step:12 create AWS – RDS – DB-MSQL - 5.7.41 - Free tier - private DB





Then go to sonarqube EC2-server Follow the commands

 # yum install mysql –y

 # mysql –version

EC2 to mysql connect: copy: Endpoint

 # mysql –h sonardatabase.cwovsgxyibkj.ap-south-1.rds.amazonaws.com -P 3306 –u admin –p

mysql> CREATE DATABASE sonar CHARACTER SET utf8 COLLATE utf8_general_ci;

Create a local and a remote user

mysql> CREATE USER sonar@localhost IDENTIFIED BY 'sonar';

mysql> CREATE USER sonar@'%' IDENTIFIED BY 'sonar';

Grant database access permissions to users.

mysql> GRANT ALL ON sonar.* TO sonar@localhost;

mysql> GRANT ALL ON sonar.* TO sonar@'%';

check users and databases

mysql> show databases;

mysql> SELECT User FROM mysql.user;

mysql> FLUSH PRIVILEGES;

mysql> QUIT

```
MySQL [(none)]> CREATE DATABASE sonar CHARACTER SET utf8 COLLATE utf8_general_ci;
Query OK, 1 row affected (0.00 sec)

MySQL [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| innodb             |
| mysql              |
| performance_schema |
| sonar              |
| sys                |
+--------------------+
6 rows in set (0.00 sec)


MySQL [(none)]> CREATE USER sonar@localhost IDENTIFIED BY 'sonar';
Query OK, 0 rows affected (0.00 sec)

MySQL [(none)]> CREATE USER sonar@'%' IDENTIFIED BY 'sonar';
Query OK, 0 rows affected (0.00 sec)

MySQL [(none)]> GRANT ALL ON sonar.* TO sonar@localhost;
Query OK, 0 rows affected (0.00 sec)

MySQL [(none)]> GRANT ALL ON sonar.* TO sonar@'%';
Query OK, 0 rows affected (0.00 sec)


MySQL [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| innodb             |
| mysql              |
| performance_schema |
| sonar              |
| sys                |
+--------------------+
6 rows in set (0.00 sec)

MySQL [(none)]> SELECT User FROM mysql.user;
+----------------+
| User           |
+----------------+
| admin          |
| sonar          |
| mysql.session  |
| mysql.sys      |
| rdsadmin       |
| sonar          |
+----------------+
6 rows in set (0.00 sec)

MySQL [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

MySQL [(none)]>
```

Step:13 **ec2-user#: chown –R ec2-user ***

Sonarqube default not started in root user. So go to ec2-user give file permission command : sudo chown –R ec2-user * Then start the sonarqube.(./sonar.sh start)

```
[ec2-user@ip-172-31-38-62 linux-x86-64]$ sudo chown -R ec2-user *
[ec2-user@ip-172-31-38-62 linux-x86-64]$
```

```
[ec2-user@ip-172-31-38-62 ~]$ cd /opt
[ec2-user@ip-172-31-38-62 opt]$ cd sonar
[ec2-user@ip-172-31-38-62 sonar]$ cd bin
[ec2-user@ip-172-31-38-62 bin]$ ls
jsw-license  linux-x86-32  linux-x86-64  macosx-universal-64  windows-x86-32  windows
[ec2-user@ip-172-31-38-62 bin]$ cd linux-x86-64
[ec2-user@ip-172-31-38-62 linux-x86-64]$ ./sonar.sh start
Starting SonarQube...
Started SonarQube.
[ec2-user@ip-172-31-38-62 linux-x86-64]$ ./sonar.sh status
SonarQube is running (1778).
[ec2-user@ip-172-31-38-62 linux-x86-64]$
```

Step:14

Goto browser for sonarqube login: public ip:port(13.127.162.156:9000).

Goto sonar qube default login :

Username: admin

Password: admin

Then goto next step sonarqube token generated.

sonarqubetokensd: **db4daefd1463124e829f788b63f095d15692891**
The token is used to identify you when an analysis is performed. If it has been compromised, you can revoke it at any point of time in your user account.

EXAMPLE:
Can you confirm that you installed SonarQube according to the documentation 86? The initial login information should be

username: admin
password: admin

Step:15 Then sonar to Jenkins intergration:

Then goto Jenkins login page install the sona plugin

1)Go to Jenkins – manage Jenkins – manage plugin – type :sonarqubescaner plugin – Install without restart.

2)Go to Jenkins – manage Jenkins – manage credentials – credentials – system – global credentials – add credentials - kind : secret text – Secret **– sonarqube token past password - ID: sonar – descption : docker hub password – created.**

EXAMPLE:

```
stage('SonarQube
Analysis') {
                    def mvnHome =  tool name: 'maven3', type: 'maven'
                                   withSonarQubeEnv('sonar') {
                                     sh "${mvnHome}/bin/mvn sonar:sonar"
                                   }
                                }
```

STEP:16

 # Ceate EC2-amazon linux - T2 midium –aws Then install java 1.8 /nexus / docker

 # sudo –i

 # sudo yum update

 # sudo yum install java-1.8.0-openjdk.x86_64 –y

Then install nexus follow the command

 # cd /opt

Goto authorized nexus webpage Then select and copy the nexus tar file and goto install .

opt# sudo sudo wget -O nexus.tar.gz https://download.sonatype.com/nexus/3/latest-unix.tar.gz

opt# sudo tar -xvf nexus.tar.gz ->untar the file

opt# sudo mv nexus-3* nexus -> renamed

opt# cd bin

bin# vi nexus.rc

nexus ALL=(ALL) NOPASSWD: ALL ->:wq!save & exit

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

As a good security practice, it is not advised to run nexus service with root privileges. So create a new user named nexus to run the nexus service

```
opt#   sudo adduser nexus
```

```
opt#  cd
```

```
root# sudo su nexus  -> for  go to nexus  user
```

Change the ownership of nexus files and nexus data directory to nexus user.

```
Nexus# nexusudo chown -R nexus:nexus /app/nexus
```

```
Nexus# sudo chown -R nexus:nexus /app/sonatype-work
```

```
chmod -R 775 /opt/nexus
```

```
chmod -R 775 /opt/sonatype-work
```

```
#vi  /opt/nexus/bin/nexus.rc
```
->open file

Uncomment run_as_user parameter and set it as following.

```
run_as_user="nexus"
```
Then :wq! Save the file exit

```
# sudo vi /app/nexus/bin/nexus.vmoptions  -> just view verify & save then exit
```

Create a nexus systemd unit file.

```
# sudo vi /etc/systemd/system/nexus.service
```

Add the following contents to the unit fille. ExecStart=/opt/nexus/bin/nexus start ,

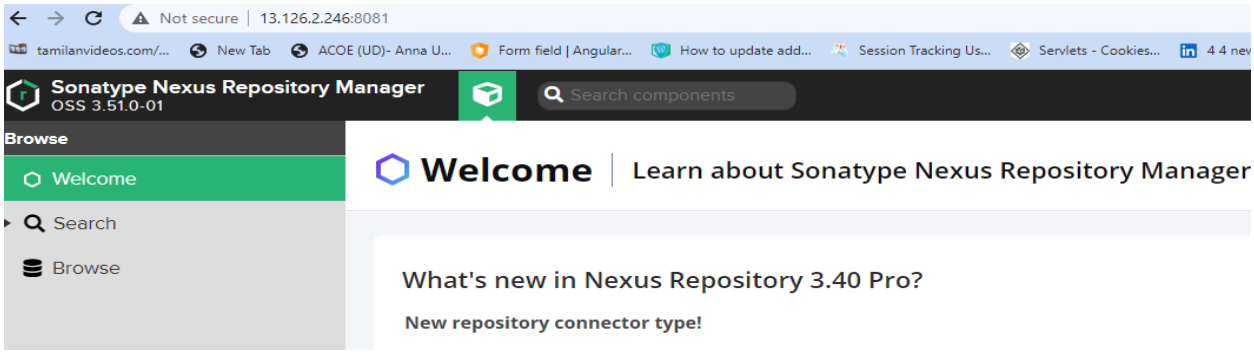
ExecStart=/opt/nexus/bin/nexus stop

```
[Unit]
Description=nexus service
After=network.target

[Service]
Type=forking
LimitNOFILE=65536
User=nexus
Group=nexus
ExecStart=/opt/nexus/bin/nexus start
ExecStop=/opt/nexus/bin/nexus stop
User=nexus
Restart=on-abort

[Install]
WantedBy=multi-user.target
```

```
# systemctl daemon-reload
# systemctl enable nexus
# systemctl start nexus
# systemctl status nexus
```

#Access the Nexus server from Laptop/Desktop browser.

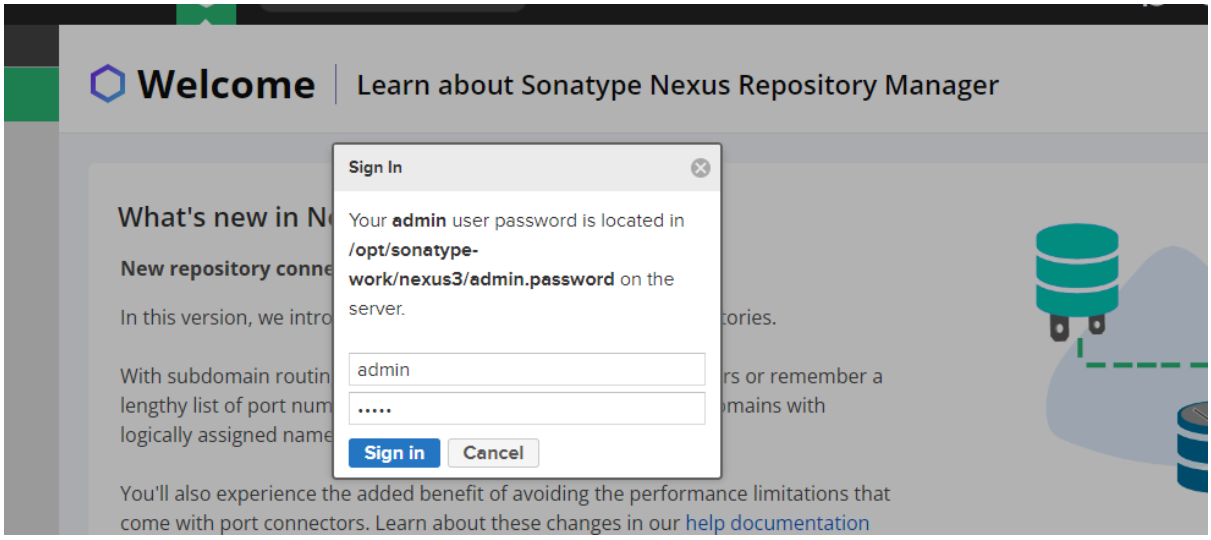
http://IPAddess/Hostname:8081/



#Default Credentials

User Name:

Password:



Then next step:

New password :admin123

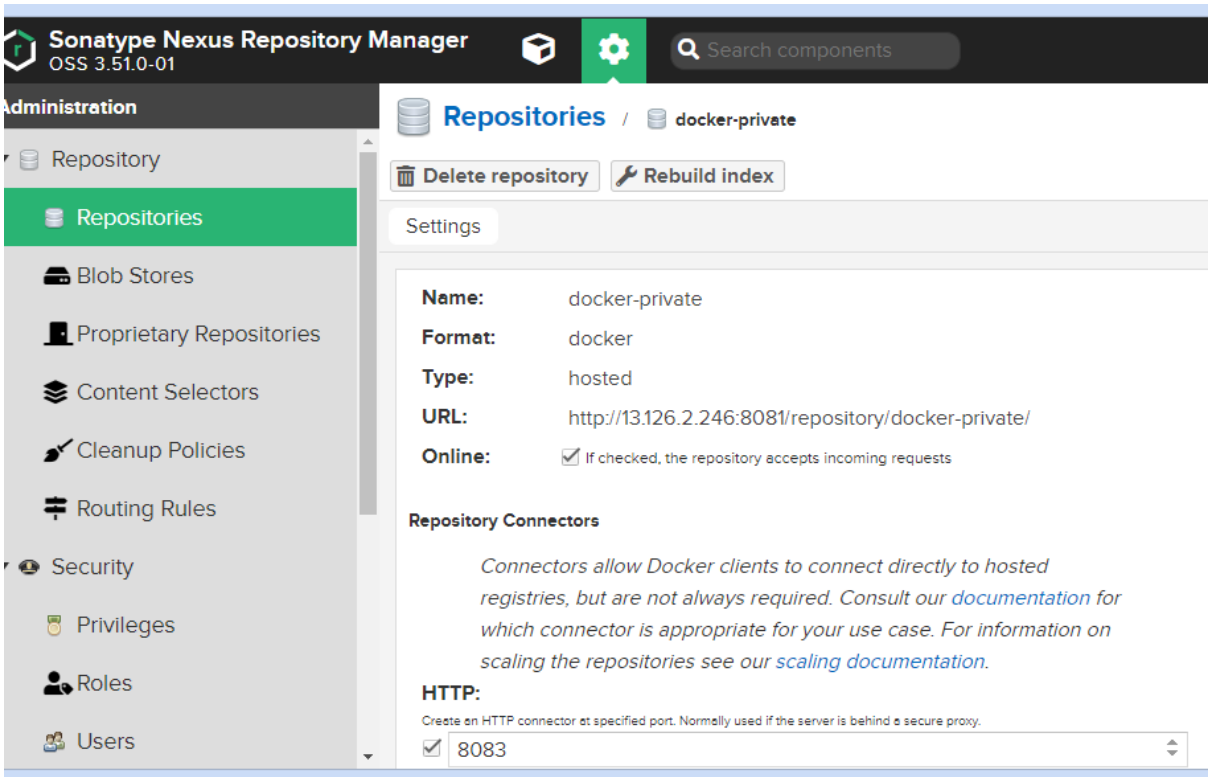
Conform password:admin123

Next step:

To choose Enable anonymous mode -> Next

Finesh.

Then create repository



Then goto Jenkins machin

Create deamon.file -> vi /etc/docker/deamon.json

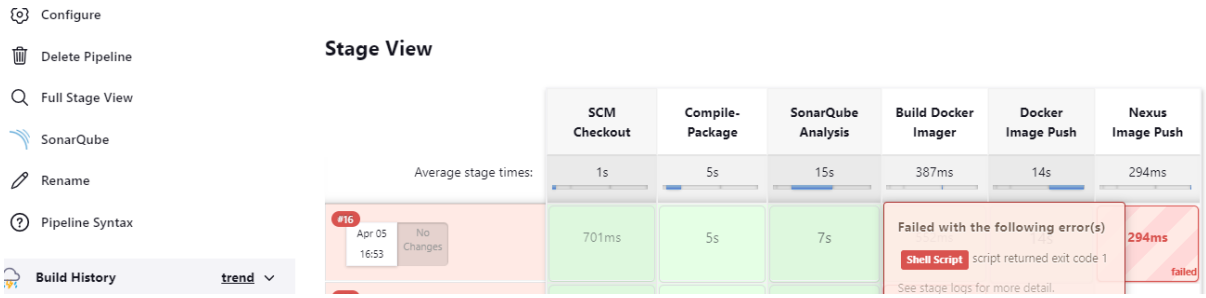
{

"insecure-registries": ["13.126.2.246:8083"]

}

:wq! ->save & exit

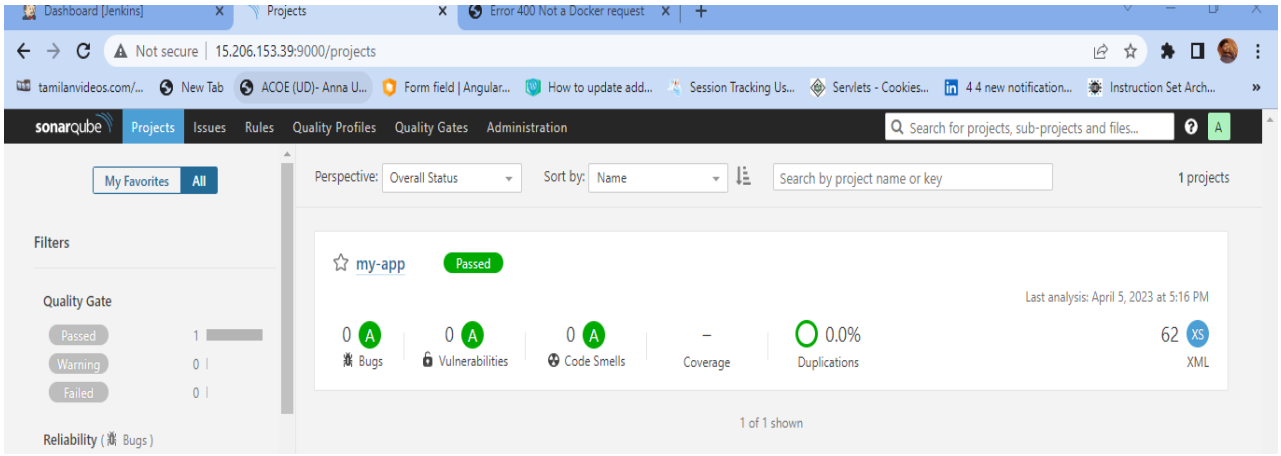
Then build



Error:

```
+ docker login -u admin -p admin123 13.126.2.246:8083
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
Error response from daemon: Get "https://13.126.2.246:8083/v2/": http: server gave HTTP response to HTTPS client
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
ERROR: script returned exit code 1
Finished: FAILURE
```

Sonarqube source code quality / bugs: output



JENKINS FIL

```
node{
  stage('SCM Checkout'){
    git 'https://github.com/SureshKirshna/my-app.git'
// update github  repository URL
  }
  stage('Compile-Package'){

     def mvnHome =  tool name: 'maven3', type: 'maven'
    sh "${mvnHome}/bin/mvn clean package"
          sh 'mv target/myweb*.war target/newapp.war'
  }
  stage('SonarQube Analysis') {
                def mvnHome =  tool name: 'maven3', type: 'maven'
                withSonarQubeEnv('sonar') {
                  sh "${mvnHome}/bin/mvn sonar:sonar"
                }
            }
  stage('Build Docker Imager'){
  sh 'docker build -t sureshdurga/myweb:0.0.2 .'
// update docker hub username for docker image build & push
  }
  stage('Docker Image Push'){
  withCredentials([string(credentialsId: 'dockerPass', variable: 'dockerPassword')]) {
  sh "docker login -u sureshdurga -p ${dockerPassword}"
// update docker hub username for docker image build & push
   }
  sh 'docker push sureshdurga/myweb:0.0.2'
```

```
  }
  stage('Nexus Image Push'){
  sh "docker login -u admin -p admin123 13.126.2.246:8083"
//update nexus server public ip with nexus docker repository http port number
  sh "docker tag sureshdurga /myweb:0.0.2 13.126.2.246:8083/damo:1.0.0"
// update docker hub username for docker image build & push
  sh 'docker push 13.126.2.246:8083/damo:1.0.0'
//update nexus server public ip with nexus docker repository http port number
  }
  stage('Remove Previous Container'){
        try{
              sh 'docker rm -f tomcattest'
        }catch(error){
              //  do nothing if there is an exception
        }
  stage('Docker deployment'){
  sh 'docker run -d -p 8090:8080 --name tomcattest sureshdurga/myweb:0.0.2'
// update docker hub username for docker image build & push
  }
}
}
```

Docker file

```
FROM
tomcat:8
          # Take the war and copy to webapps of tomcat
          COPY target/newapp.war /usr/local/tomcat/webapps/
```

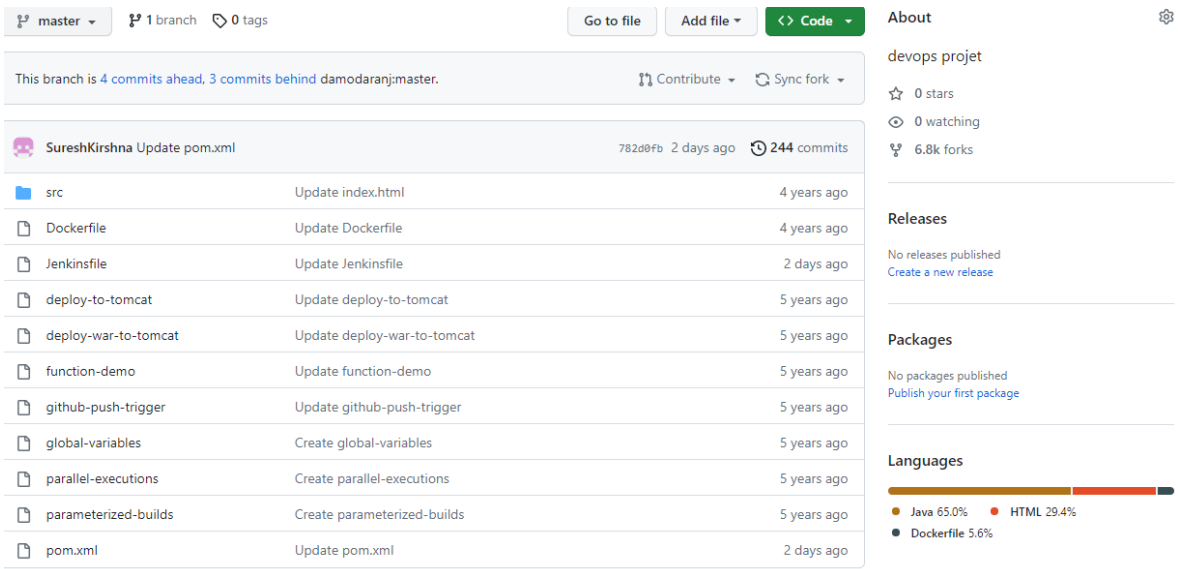
Pom.yml- file

```
<project
xmlns="http://maven.apache.or
g/POM/4.0.0"
xmlns:xsi="http://www.w3.org/
2001/XMLSchema-instance"
      xsi:schemaLocation="http://maven.
apache.org/POM/4.0.0
http://maven.apache.org/maven-
v4_0_0.xsd">
      <modelVersion>4.0.0</modelVersion
>
      <groupId>in.javahome</groupId>
      <artifactId>myweb</artifactId>
      <packaging>war</packaging>
      <version>0.0.5</version>
      <name>my-app</name>

      <!-- SureshKirshna on Github
Webhooks-->


      <url>http://maven.apache.org</url
>


      <dependencies>
            <dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi</artifactId>
                  <version>3.7</version>
            </dependency>


            <dependency>

      <groupId>javax.servlet</groupId>

      <artifactId>javax.servlet-
api</artifactId>

      <version>3.0.1</version>
```

```
                    <!--
<scope>provided</scope> -->
            </dependency>
            <dependency>

      <groupId>junit</groupId>

      <artifactId>junit</artifactId>

      <version>3.8.1</version>
                  <scope>test</scope>
            </dependency>

      </dependencies>
      <distributionManagement>
             <snapshotRepository>
                <id>nexus</id>

<url>http://65.2.131.118:8083/repositor
y/maven-snapshots/</url>
             </snapshotRepository>

            <repository>
                <id>nexus</id>

<url>http://65.2.131.118:8083/repositor
y/maven-releases/</url>
            </repository>
      </distributionManagement>

<build>
    <plugins>
        <plugin>

<groupId>org.apache.maven.plugins</grou
pId>
            <artifactId>maven-compiler-
plugin</artifactId>
            <version>3.6.1</version>
            <configuration>
                <source>1.7</source>
                <target>1.7</target>
            </configuration>
        </plugin>
      <plugin>
```

```
<groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-surefire-plugin</artifactId>
                <version>2.19</version>
            </plugin>
        </plugins>
    </build>

    </project>
```

GITHUB REPO:



Error:

1) **Docker image push error**

```
+ docker build -t sureshdurga/myweb:0.0.2 .
Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
ERROR: script returned exit code 1
Finished: FAILURE
```

2)sonarqube error

```
[INFO] ------------------------------------------------------------------------
[ERROR] Failed to execute goal org.sonarsource.scanner.maven:sonar-maven-plugin:3.9.1.2184:sonar (default-cli) on project
myweb: Unable to execute SonarScanner analysis: Fail to get bootstrap index from server: connect timed out -> [Help 1]
[ERROR]
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging.
[ERROR]
[ERROR] For more information about the errors and possible solutions, please read the following articles:
[ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/MojoExecutionException
[Pipeline] }
WARN: Unable to locate 'report-task.txt' in the workspace. Did the SonarScanner succeed?
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
ERROR: script returned exit code 1
Finished: FAILURE
```

3)docker image buld error

```
+ docker build -t sureshdurga/myweb:0.0.2 .
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post
"http://%2Fvar%2Frun%2Fdocker.sock/v1.24/build?
buildargs=%7B%7D&cachefrom=%5B%5D&groupparent=&cpuperiod=0&cpuquota=0&cpusetcpus=&cpusetmems=&cpushares=0&dockerfile=Docker
file&labels=%7B%7D&memory=0&memswap=0&networkmode=default&rm=1&shmsize=0&t=sureshdurga%2Fmyweb%3A0.0.2&target=&ulimits=null&
version=1": dial unix /var/run/docker.sock: connect: permission denied
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
ERROR: script returned exit code 1
Finished: FAILURE
```

4)docker permission error

```
+ docker-private login -u admin -p admin123 65.2.131.118:8083
/var/lib/jenkins/workspace/myproject@tmp/durable-3b71f097/script.sh: line 1: docker-private: command not found
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
ERROR: script returned exit code 127
Finished: FAILURE
```

5)nexus docker repo connecting error

```
[Pipeline] sh
+ docker login -u admin -p admin123 65.2.131.118:8083
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
Error response from daemon: Get "https://65.2.131.118:8083/v2/": http: server gave HTTP response to HTTPS client
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
ERROR: script returned exit code 1
Finished: FAILURE
```