Append these entries to bottom of the “sysctl.conf” file.

sudo nano /etc/sysctl.conf

vm.max\_map\_count=262144

fs.file-max=65536

ulimit -n 65536

ulimit -u 4096

And, also append these entries at the end of the “limits.conf” file.

sudo nano /etc/security/limits.conf

sonarqube - nofile 65536

sonarqube - nproc 4096

**STEP 02: INSTALL OPENJDK 11**

**Download & Install JDK 11 APT Repositories**

Now, It’s time to install Java on your system. Don’t forget to install compatible Java version with you SonarQube version.

First perform a system update.

sudo apt-get update -y

Then, Install OpenJDK 11

sudo apt-get install openjdk-11-jdk -y

**Set Default JDK Version**

Then, You need to set newly installed Java version as your default Java version

sudo update-alternatives --config java

**Verify Install Java Version**

java –version

### STEP 02: INSTALL & CONFIGURE POSTGRESQL DATABASE FOR SONARQUBE

In this tutorial I’m using PostgreSQL as my database engine. You also can use other compatible DB such as MySQL or Oracle.

It’s always better to check version compatibility matrix, which recommends by SonarQube developers.

REF: <https://docs.sonarqube.org/latest/requirements/requirements/>

Let’s do a system update again.

sudo apt update

**Import Trusted PGP Key and PostgreSQL APT Repo**

Then, Install trusted GPG key on your system. And create a repository file for PostgreSQL.

wget -q https://www.postgresql.org/media/keys/ACCC4CF8.asc -O - | sudo apt-key add -

sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt/ `lsb\_release -cs`-pgdg main" >> /etc/apt/sources.list.d/pgdg.list'

**Install PostgreSQL**

Let’s install PostgreSQL on your system.

sudo apt install postgresql postgresql-contrib

**Check PostgreSQL Version**

sudo -u postgres psql -c "SELECT version();"

**Enable & Start PostgreSQL Service**

Enable & start service to be able to start at the system boots up.

sudo systemctl enable postgresql.service

sudo systemctl start postgresql.service

**Change PostgreSQL default user password**

Change default PostgreSQL password and set new password.

sudo passwd postgres

**Switch to PostgreSQL User**

Now, Switch into “postgres” user.

su - postgres

**Create New User “sonar”**

Create a new database user which named with “sonar”.

createuser sonar

**Log Into PostgreSQL Shell**

Now, Login to postgresql database shell.

psql

**Set Password for SonarQube Database User “sonar”**

And, Then set a password for the database user “sonar”

ALTER USER sonar WITH ENCRYPTED PASSWORD 'p@ssw0rd';

**Create New Database “sonarqube”**

Create a new database which named with “sonarqube”

CREATE DATABASE sonarqube OWNER sonar;

**Grant Privileges to “sonar” User on “sonarqube” Database**

Now, Grant all privileges to that user and database.

GRANT ALL PRIVILEGES ON DATABASE sonarqube to sonar;

Exit From PostgreSQL Shell

\q

**Exit From “postgres” User**

exit

**Restart & Check PostgreSQL DB Service Status again**

Enable PostgreSQL service to be able to start automatically at systems boots-up.

Sudo systemctl restart postgresql

Sudo systemctl status -l postgresql

Now Check wether PostgreSQL is listing on default port “5432”

netstat -tulpena | grep postgres

### STEP 03: DOWNLOAD & INSTALL SONARQUBE

Now, It’s time to download SonarQube binary archive file and extract on out installation directory.

**Download SonarQube Archive File**

REF: <https://binaries.sonarsource.com/Distribution/sonarqube/>

Now, Let’s create temporary directory and download SonarQube archive file.

sudo mkdir /sonarqube/

cd /sonarqube/

sudo curl -O https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-8.3.0.34182.zip

Additionally, you may need to install “zip” apt package if not available your system.

sudo apt-get install zip -y

Extract your downloaded archive into /opt/ directory.

sudo unzip sonarqube-8.3.0.34182.zip -d /opt/

Move Extracted setup into /opt/sonarqube/ directory

sudo mv /opt/sonarqube-8.3.0.34182/ /opt/sonarqube

### STEP 04: CREATE GROUP & USER FOR SONARQUBE

Now, We need to create a system user and group for SonarQube service.

**Create a group named “sonar”**

First create a system group which named with “sonar”

sudo groupadd sonar

**Create a user named “sonar” and into “sonar” group with directory access**

Then, Create an user and the add user into the group with directory permission to the /opt/ directory.

sudo useradd -c "SonarQube - User" -d /opt/sonarqube/ -g sonar sonar

Provide user & group directory ownership to “/opt/sonarqube/”\*\*

sudo chown sonar:sonar /opt/sonarqube/ -R

### STEP 05: CONFIGURE SONARQUBE

Now, Let’s head-over to “**sonar.properties**” configuration file and do the following changes

sudo nano /opt/sonarqube/conf/sonar.properties

**UnComment and type PostgreSQL database username and password that we’ve created at privous step.**

Now, We need to point our PostgreSQL database to SonarQube service. We are using “localhost” as db host since we’ve installed postgreSQl on same server.

Un-comment these lines and modify them as necessary.

1.sonar.jdbc.username=sonar

2.sonar.jdbc.password=p@ssw0rd

3.sonar.jdbc.url=jdbc:postgresql://localhost/sonarqube

4.sonar.search.javaOpts=-Xmx512m -Xms512m -XX:+HeapDumpOnOutOfMemoryError

*########### OPTIONAL USE ONLY #############*

sonar.jdbc.username=sonar

sonar.jdbc.password=sonar

sonar.jdbc.url=jdbc:postgresql://localhost/sonarqube

sonar.web.host=127.0.0.1

sonar.web.port=9000

sonar.web.javaAdditionalOpts=-server

sonar.search.javaOpts=-Xmx512m -Xms512m -XX:+HeapDumpOnOutOfMemoryError

sonar.log.level=INFO

sonar.path.logs=logs

*###########################################*

### STEP 06: CONFIGURE SYSTEMD SERVICE FOR SONARQUBE

Now, Create a startup script for SonarQube service that start at system boots

Create a systemd service file for SonarQube to be run at system startup.

sudo nano /etc/systemd/system/sonarqube.service

Add these content into the “sonarqube.service” file.

[Unit]

Description=SonarQube service

After=syslog.target network.target

[Service]

Type=forking

ExecStart=/opt/sonarqube/bin/linux-x86-64/sonar.sh start

ExecStop=/opt/sonarqube/bin/linux-x86-64/sonar.sh stop

User=sonar

Group=sonar

Restart=always

LimitNOFILE=65536

LimitNPROC=4096

[Install]

WantedBy=multi-user.target

**Enable & Start SonarQube Service**

sudo systemctl daemon-reload

sudo systemctl enable sonarqube.service

sudo systemctl start sonarqube.service

sudo systemctl status -l sonarqube.service

After sometime later, Check whether the port are listening

sudo netstat -tulpena | grep 9000

### STEP 07: CONFIGURE NGINX REVERSE PROXY FOR SONARQUBE

**Install NGINX Package**

Now we need to expose our SonarQube server into outside as it is listening only on localhost. Therefore we are creating a Nginx reverse proxy to redirect outside traffic into the SonarQube.

sudo apt-get install nginx -y

Goto **/etc/nginx/nginx.conf** and un-comment these two lines

sudo nano /etc/nginx/nginx.conf

include /etc/nginx/conf.d/\*.conf;

include /etc/nginx/sites-enabled/\*;

**Create NGINX Configuration File For SonarQube**

Create a reverse proxy configuration file

sudo nano /etc/nginx/sites-enabled/sonarqube.conf

Copy and paste this vertual-host server block and change “server\_name” entry as you required.

server{

listen 80;

server\_name sonarqube.da.com;

access\_log /var/log/nginx/sonar.access.log;

error\_log /var/log/nginx/sonar.error.log;

proxy\_buffers 16 64k;

proxy\_buffer\_size 128k;

location / {

proxy\_pass http://127.0.0.1:9000;

proxy\_next\_upstream error timeout invalid\_header http\_500 http\_502 http\_503 http\_504;

proxy\_redirect off;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto http;

}

}

**Check ENGINX configurations**

sudo nginx -t

**Enable & Restart Nginx Service**

sudo systemctl enable nginx.service

sudo systemctl restart nginx.service

sudo systemctl status -l nginx.service

**Check whether port 80 listening for connections**

sudo netstat -tulpena | grep 80

### STEP 08: FIREWALL CONFIGURATION

**Allow TCP ports 9000, 9001, 80 through the firewall**

sudo ufw allow 80,9000,9001/tcp

sudo ufw status

\*\* make sure /etc/nginx/sites-enable/ there is no other sites . only sonarqube.conf should be there

### STEP 09: ACCESS SONARQUBE THROUGH WEB BROWSER

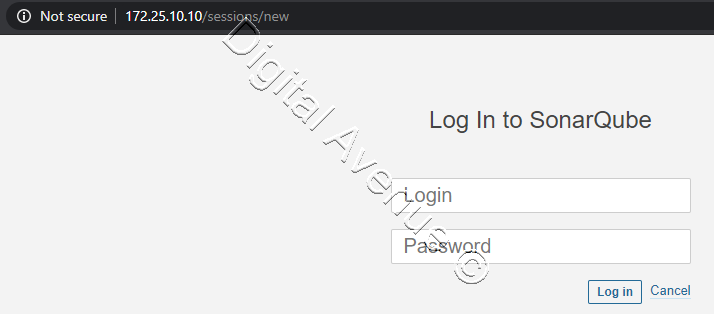
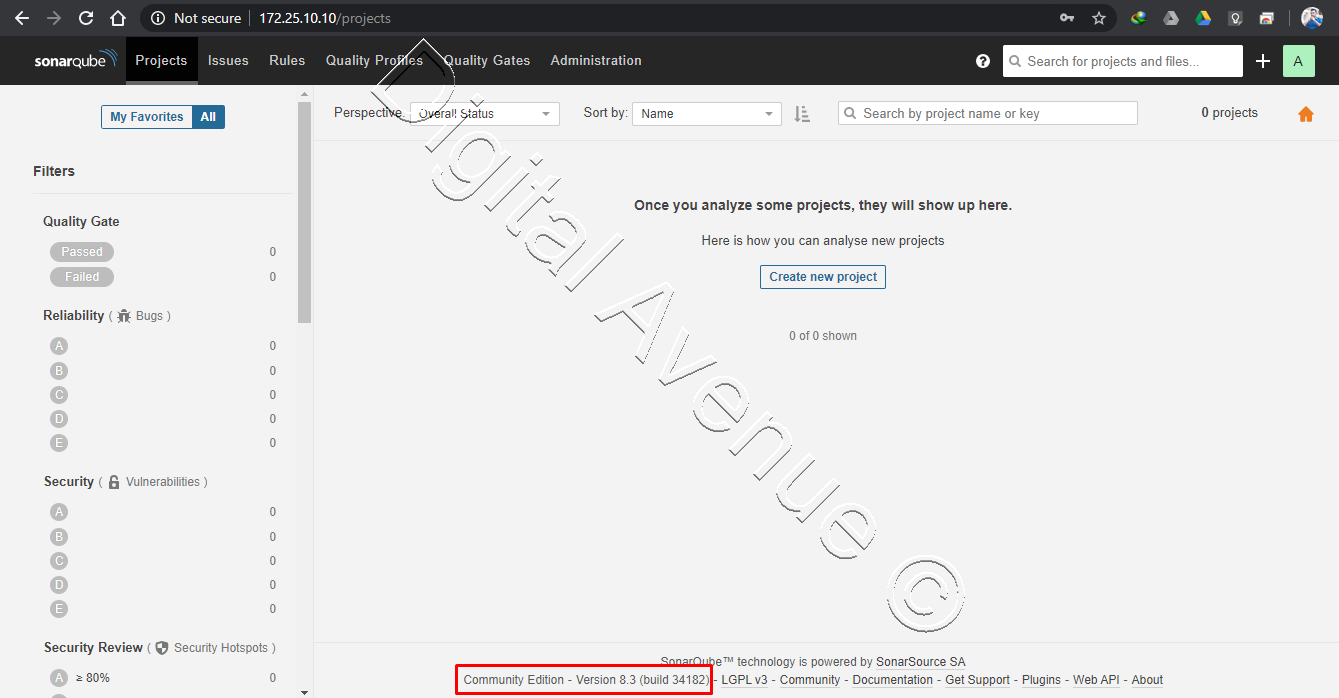
Now, SonarQube installation & configuration has been completed. It’s time to access web console through the web browser.

Provide the default administrator account username and password as admin / admin

**Default Username: admin**

**Default Password: admin**

*http://172.25.10.10/ OR http://YOUR-SERVER-IP*

##### TORUBLESHOOTING TIPS

Sometime SonaqQube will not start as we expected. Most of the time the reason is related to elasticsearch service. SonarQube uses elasticsearch as it’s indexing engine. So, We may need to troubleshoot elasticsearch as well.

Here are some troubleshooting tips:

SonarQube stores their service logs under “/opt/sonarqube/logs” directory. You may need those log files in case of troubleshooting purpose.

**Troubleshooting Tips : Log Paths**

/opt/sonarqube/logs/es.log

/opt/sonarqube/logs/sonar.log

/opt/sonarqube/logs/web.log

**Troubleshooting Tips: JVM OPTION & HEAP MEMORY ISSUES**

Additionally you may required to modify some entries related to elasticsearch and JVM options, Therefore SonarQube using elastciseach and JVM options. The reason is our system’s HEAP MEMORY will not be compatible with the JVM configurations.

If your sonarqube service not starting or keep restarting, check following log file.

tail -f /opt/sonarqube/logs/es.log

tail -f /opt/sonarqube/logs/sonar.log

tail -f /opt/sonarqube/logs/access.log

and check port number 9000 or 9001 listing on locahost.

If not, your JVM.OPTION may not compatible with you physical RAM amount.Then, You need to define matching JAVA HEAP Memory size for you host machine.

vim /opt/sonarqube/elasticsearch/config/jvm.options

*# Xms represents the initial size of total heap space*

*# Xmx represents the maximum size of total heap space*

-Xms1g

-Xmx1g

You may need to adjust your HEAP MEMORY according to you physical usable memory size.

/opt/sonarqube/elasticsearch/config/elasticsearch.yml

/opt/sonarqube/elasticsearch/config/log4j2.properties

**SonarQube initial configuration has been completed. In the next tutorial, I will show you how to integrate and analyze your project code on SonarQube with Jenkins server and GitLab. And analysis of code deployments real-time.**

If you need further clarification, Please ask on YouTube video comment section.

