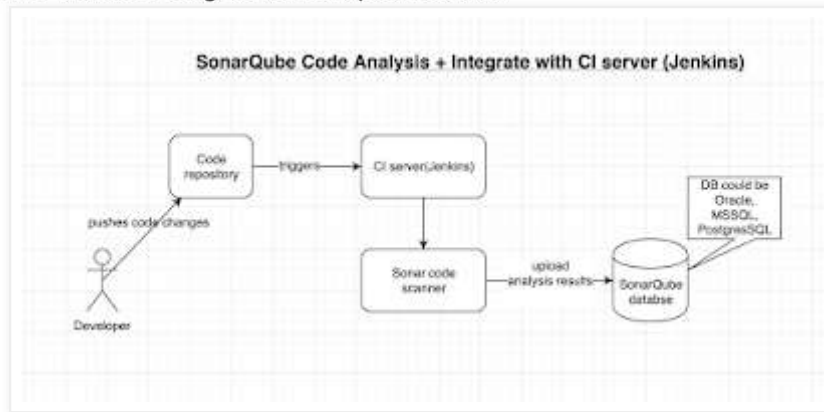


Install SonarQube on Ubuntu - How to install SonarQube on Ubuntu 16.0.4?

Please find steps for installing SonarQube on Ubuntu EC2. Make sure port 9000 is opened in security group(firewall rule).

SonarQube is java based tool along with back end - back end can be MySQL, Oracle or PostgreSQL. We will use Postgres for set up on Ubuntu.



Let us start with java install (skip java install if you already have it installed)

1. Java steps

```
sudo apt-get update
```

```
sudo apt-get install default-jdk -y
```

Verify Java Version

```
java -version
```

```
openjdk version "1.8.0_191"
```

```
OpenJDK Runtime Environment (build 1.8.0_191-8u191-b12-2ubuntu0.16.04.1-b12)
```

```
OpenJDK 64-Bit Server VM (build 25.191-b12, mixed mode)
```

2. Postgres Installation

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

```
ubuntu@ip-172-31-46-114:~$ sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt/ `lsb_release -cs`-pgdg main" >> /etc/apt/sources.list.d/pgdg.list'
```

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

```
ubuntu@ip-172-31-46-114:~$ sudo wget -q https://www.postgresql.org/media/keys/ACCC4CF8.asc -O - | sudo apt-key add -
```

```
3. sudo apt-get -y install postgresql postgresql-contrib
```

```
update-alternatives: using /usr/share/postgresql/16/bin/postmaster.1.gz to provide /usr/share/postgresql/postmaster
Setting up postgresql (16+200.pgdg.04.1) ...
Setting up postgresql-contrib (16+200.pgdg.04.1) ...
Setting up postgresql (16+200.pgdg.04.1) ...
Creating config file /etc/default/postgresql with new version
update-alternatives: using /usr/bin/postgresql to provide /usr/bin/postgresql
Processing triggers for libc-bin (2.27-0ubuntu2) ...
Processing triggers for systemd (235-0ubuntu2) ...
Processing triggers for Unattended-Upgrade (0.100.0-19) ...
E: Target Packages (main/binary-all/packages) is configured multiple times in /etc/apt/sources.list.d/pgdg.list:1
E: Target Packages (main/binary-all/packages) is configured multiple times in /etc/apt/sources.list.d/pgdg.list:1
E: Target Translations (main/lib/translation-en-US) is configured multiple times in /etc/apt/sources.list.d/pgdg.list:2
E: Target Translations (main/lib/translation-en) is configured multiple times in /etc/apt/sources.list.d/pgdg.list:2
ubuntu@ip-172-31-46-114:~$
```

4. sudo systemctl start postgresql
5. sudo systemctl enable postgresql

Change the password for postgres user by entering below command

6. sudo passwd postgres

enter as admin (it wont display in screen though, thats fine)

enter admin again for r-typing password

```
ubuntu@ip-172-31-40-114:~$ sudo passwd postgres
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
ubuntu@ip-172-31-40-114:~$
```

Login as postgres user now

7. su - postgres

enter admin as password

```
ubuntu@ip-172-31-40-114:~$ su - postgres
Password:
postgres@ip-172-31-40-114:~$
```

8. Now create a user below

createuser sonar

9. Switch to sql shell by entering

psql

```
postgres@ip-172-31-40-114:~$ psql
psql (10.5 (Ubuntu 10.5-1.pgdg16.04+1))
Type "help" for help.

postgres=#
```

Execute the below three lines (one by one)

ALTER USER sonar WITH ENCRYPTED password 'password';

CREATE DATABASE sonar OWNER sonar;

\q

```
postgres=# ALTER USER sonar WITH ENCRYPTED password 'password';
ALTER ROLE
postgres=# CREATE DATABASE sonar OWNER sonar;
CREATE DATABASE
postgres=#
postgres=# \q
postgres@ip-172-31-40-114:~$
```

type exit to come out of postgres user.

```
postgres@ip-172-31-40-114:~$ exit
Logout
ubuntu@ip-172-31-40-114:~$
```

3. Now install SonarQube Web App

sudo wget <https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-6.4.zip>

```

HTTP Request sent, awaiting response... 200 OK
Length: 139755847 (133M) [application/zip]
Saving to: 'sonarqube-6.4.zip'

sonarqube-6.4.zip           100%[_____
2018-09-16 04:03:41 (58.1 MB/s) - 'sonarqube-6.4.zip' saved [139755847/139755847]

```

sudo apt-get -y install unzip

sudo unzip sonarqube-6.4.zip -d /opt

```

creating: /opt/sonarqube-6.4/lib/jdbc/postgresql/postgres
creating: /opt/sonarqube-6.4/lib/jdbc/h2/
inflating: /opt/sonarqube-6.4/lib/jdbc/h2/h2-1.3.176.jar
ubuntu@ip-172-31-40-114:~$

```

sudo mv /opt/sonarqube-6.4 /opt/sonarqube -v

```

ubuntu@ip-172-31-40-114:~$ sudo mv /opt/sonarqube-6.4 /opt/sonarqube -v
'/opt/sonarqube-6.4' -> '/opt/sonarqube'
ubuntu@ip-172-31-40-114:~$

```

Modify sonar.properties file

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

uncomment the below lines by removing # and add values highlighted yellow

sonar.jdbc.username=sonar

sonar.jdbc.password=password

```

sonar.jdbc.username=sonar
sonar.jdbc.password=password
#----- Embedded Database (default)

```

Next, uncomment the below line, removing #

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

```

#----- PostgreSQL 8.x/9.x
# If you don't use the schema named "public", pl
sonar.jdbc.url=jdbc:postgresql://localhost/sonar

```

Press escape, and enter :wq! to come out of the above screen.

Create Sonar as a service

Execute the below command:

sudo vi /etc/systemd/system/sonar.service

```

[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=root
Group=root
Restart=always

[Install]
WantedBy=multi-user.target

```

add the below code in green color:

[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=root
Group=root
Restart=always

[Install]
WantedBy=multi-user.target

sudo systemctl enable sonar
sudo systemctl start sonar
sudo systemctl status sonar

```
ubuntu@ip-172-31-48-114:~$ sudo systemctl status sonar
● sonar.service - SonarQube service
   Loaded: loaded (/etc/systemd/system/sonar.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2018-09-16 04:10:23 UTC; 4s ago
     Process: 5709 ExecStart=/opt/sonarqube/bin/linux-x86-64/sonar.sh start (code=exited, status=0/SUCCESS)
    Main PID: 5754 (wrapper)
      CGroup: /systemd/system/sonar.service
              └─5754 /opt/sonarqube/bin/linux-x86-64/sonar.sh start
```

type q now to come out of this mode.

Now execute the below command to see if Sonarqube is up and running. This may take a few minutes.

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

Make sure you get the below message that says sonarqube is up..

```
2018-09-20 01:44:54 INFO app[o.s.a.p.java.ProcessLauncherImpl] Launch process[web]: /usr/lib/jvm/java-8-oracle/bin/java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Xms128m -Xmx128m -XX:+HeapDumpOnOutOfMemoryError -Djtemp -cp ./lib/common/*:./lib/server/*:/opt/sonarqube/lib/jdbc/postgresql/postgresql-9.4.1209.jar bServer /opt/sonarqube/temp/sq-process8141698489790978211properties
2018-09-20 01:45:07 INFO app[o.s.a.scheduler.impl] Process[web] is up
2018-09-20 01:45:07 INFO app[o.s.a.p.java.ProcessLauncherImpl] Launch process[ce]: /usr/lib/jvm/java-8-oracle/bin/java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Xms128m -Xmx128m -XX:+HeapDumpOnOutOfMemoryError -Djtemp -cp ./lib/common/*:./lib/server/*:./lib/ce/*:/opt/sonarqube/lib/jdbc/postgresql/postgresql-9.4.1209.jar app.CeServer /opt/sonarqube/temp/sq-process3589015779696091342properties
2018-09-20 01:45:11 INFO app[o.s.a.scheduler.impl] Process[ce] is up
2018-09-20 01:45:11 INFO app[o.s.a.scheduler.impl] SonarQube is up
```

Now access sonarQube UI by going to browser and enter public dns name with port 9000
Now to go to browser --> http://your_SonarQube_publicdns_name:9000/

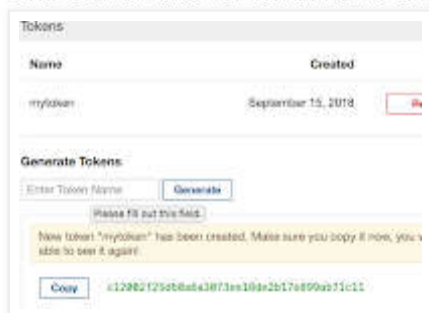
Here below are the steps for integrating SonarQube with Jenkins:

Pre-requisites: Make sure SonarQube is up and running and do the below steps:

Make sure Sonarqube plug-in installed in Jenkins

1. You need to login to SonarQube using admin/admin and click on administration, security, users, click on Tokens, under generate token.

Give some value for token name and click on generate. Copy the token.



2. After installing SonarQube successfully, login to Jenkins. Manage Jenkins --> Configure

System --> SonarQube installation



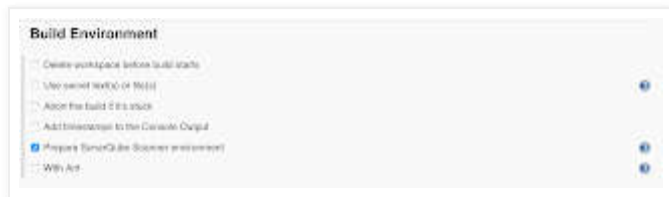
The screenshot shows the 'SonarQube servers' configuration page. Under the 'Environment variables' tab, the 'Enable injection of SonarQube server configuration as build environment variables' checkbox is checked. Below this, there are fields for 'Name' (set to 'SonarQube'), 'Server URL' (set to 'http://localhost:9000'), and 'Server authentication token' (with a 'Generate a new authentication token' link). A note at the bottom states: 'SonarQube authentication token. Mandatory when authenticating against a remote SonarQube instance.'

Enter name, URL as `http://localhost:9000`, paste the token you copied from step #1

3. Click on Enable injection of Sonarqube server configuration.

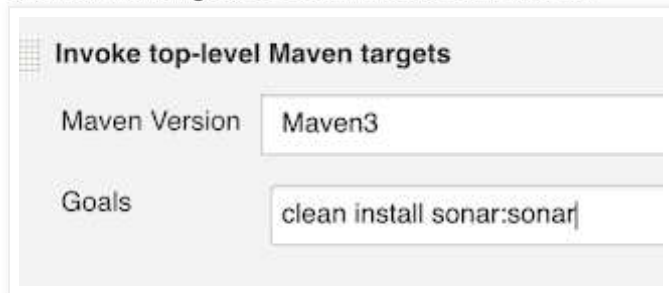
4. Save.

5. Click on your existing free style job, click on configure. click on prepare Sonarqube scanner environment.



The screenshot shows the 'Build Environment' configuration page. Under the 'Prepare SonarQube Scanner environment' checkbox, which is checked, there is a sub-section for 'SonarQube Scanner environment' with a 'Configure' link. Other options like 'Create workspace before build starts', 'Use secret text(s) or file(s)', 'Add the build file's stack', 'Add build output to the Console Output', and 'With Art' are also visible.

6. enter maven goal as **clean install sonar:sonar**



The screenshot shows the 'Invoke top-level Maven targets' configuration page. The 'Maven Version' is set to 'Maven3'. The 'Goals' field contains the text 'clean install sonar:sonar'.

7. click on save and build the job.

You will see that Jenkins will integrate with Sonarqube which does code analysis of your project. Login to SonarQube, click on Projects to see the project dash board.