Log in or Sign up



Search			

Tutorials Tags Forums Linux Commands Subscribe

**Q** Tutorial search

Home

How to Install SonarQube on Ubuntu 22.04

## How to Install SonarQube on Ubuntu 22.04

SonarQube or formerly Sonar is an open-source platform for static code analysis and code security. It allows you to perform static code analysis and code quality to detect bugs and enhance application security. It also provides reports such as duplicate code, coding standards, code complexity, and security recommendation.

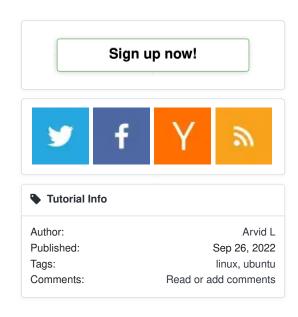
#### This tutorial exists for these OS versions

- Ubuntu 22.04 (Jammy Jellyfish)
- Ubuntu 20.04 (Focal Fossa)
- Ubuntu 18.04 (Bionic Beaver)
- Ubuntu 16.04 (Xenial Xerus)

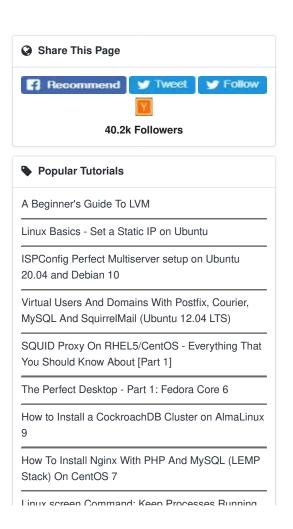
#### On this page

- Prerequisites
- Installing Java OpenJDK
- Installing PostgreSQL Database System
- Setting up System
- <u>Downloading SonarQube Package</u>
- Configuring SonarQube
- Running SonarQube with Reverse Proxy
- SonarQube Installatioon
- Conclusion

With sonarQube, you can automate static code analysis for 29 programming languages. You can easily integrate SonarQube with your existing CI/CD tools such as Jenkins, Azure DevOps, or IDE such as IntelliJ and Visual Code Studio.



- - -





In this guide, you will learn how to install SonarQube static code analysis on Ubuntu 22.04 server. You will also learn how to install PostgreSQL which will be used as the database for SonarQube and the Nginx web server that will be used as the reverse proxy.

### **Prerequisites**

Before starting this guide, you should have the following:

- An Ubuntu server 22.04 with UFW firewall enabled.
- A non-root user with sudo/administrator privileges.
- A domain name pointed to the Ubuntu server IP address.

## **Installing Java OpenJDK**

Your first step here is to install the Java OpenJDK on your Ubuntu system. The SonarQube server required Java OpenJDK v11 to be installed on your Linux machine.

Before start installing packages, run the following apt command to update and refresh your Ubuntu package index repository.

sudo apt update

Then, install the Java OpenJDK v11 using the following apt command. The default OpenJDK version on the latest Ubuntu 22.04 is Java OpenJDK v11.

Input Y when prompted to confirm the installation and press ENTER to proceed.

```
oot@sonarqube-server:-#
 root@sonarqube-server:~# sudo apt install default-jdk
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
 The following additional packages will be installed:
  alsa-topology-conf alsa-ucm-conf at-spi2-core ca-certificates-java dconf-gsettings-backend dconf-servic
  fontconfig-config fonts-dejavu-core fonts-dejavu-extra gsettings-desktop-schemas java-common libasound
  libatk-wrapper-java-jni libatk1.0-0 libatk1.0-data libatspi2.0-0 libavahi-client3 libavahi-common-data
  libdrm-nouveau2 libdrm-radeon1 libfontconfig1 libfontenc1 libgif7 libgl1 libgl1-amber-dri libgl1-mesa-d
  libharfbuzz0b libice-dev libice6 libjpeg-turb08 libjpeg8 liblcms2-2 libllvm13 libpciaccess0 libpcsclite
  libwayland-client0 libx11-dev libx11-xcb1 libxau-dev libxaw7 libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0 l
  libxcb-sync1 libxcb-xfixes0 libxcb1-dev libxcomposite1 libxdmcp-dev libxfixes3 libxft2 libxi6 libxinera
  libxshmfence1 libxt-dev libxt6 libxtst6 libxv1 libxxf86dga1 libxxf86vm1 mesa-vulkan-drivers openjdk-11-
  openjdk-11-jre-headless session-migration x11-common x11-utils x11proto-dev xorg-sgml-doctools xtrans-d
  libasound2-plugins alsa-utils cups-common libice-doc liblcms2-utils pcscd libsm-doc libx11-doc libxcb-do
  libnss-mdns fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei | fonts-wqy-zenhei fonts-indic
 The following NEW packages will be installed:
  alsa-topology-conf alsa-ucm-conf at-spi2-core ca-certificates-java dconf-gsettings-backend dconf-servic
  default-jre-headless fontconfig-config fonts-dejavu-core fonts-dejavu-extra gsettings-desktop-schemas
  libatk-wrapper-java libatk-wrapper-java-jni libatk1.0-0 libatk1.0-data libatspi2.0-0 libavahi-client3
  libdrm-amdgpu1 libdrm-intel1 libdrm-nouveau2 libdrm-radeon1 libfontconfig1 libfontenc1 libgif7 libgl1
  libglx-mesa0 libglx0 libgraphite2-3 libharfbuzz0b libice-dev libice6 libjpeg-turbo8 libjpeg8 liblcms2-:
  libsm-dev libsm6 libvulkan1 libwayland-client0 libx11-dev libx11-xcb1 libxau-dev libxaw7 libxcb-dri2-0
  libxcb-shape0 libxcb-shm0 libxcb-sync1 libxcb-xfixes0 libxcb1-dev libxcomposite1 libxdmcp-dev libxfixes
  libxrandr2 libxrender1 libxshmfence1 libxt-dev libxt6 libxtst6 libxv1 libxxf86dga1 libxxf86vm1 mesa-vul
  openjdk-11-jre-headless session-migration x11-common x11-utils x11proto-dev xorg-sgml-doctools xtrans-de
0 upgraded, 100 newly installed, 0 to remove and 19 not upgraded.
Need to get 305 MB of archives.
After this operation, 596 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 https://mirrors.edge.kernel.org/ubuntu jammy/main amd64 alsa-topology-conf all 1.2.5.1-2 [15.5 kB]
```

Once Java OpenJDK is installed, verify the Java version using the following java command. You will see the output of the Java version that is installed on your system.

```
root@sonarqube-server:~#
root@sonarqube-server:~#
java -version
openjdk version "11.0.16" 2022-07-19
OpenJDK Runtime Environment (build 11.0.16+8-post-Ubuntu-0ubuntu122.04)
OpenJDK 64-Bit Server VM (build 11.0.16+8-post-Ubuntu-0ubuntu122.04, mixed mode, sharing)
root@sonarqube-server:~#
root@sonarqube-server:~#
```

### **Installing PostgreSQL Database System**

SonarQube supports multiple database systems such as PostgreSQL, Microsoft SQL Server, and oracle database. For this example, you will use PostgreSQL as the database for your SonarQube installation.

At the time of this writing, the latest version of SonarQube required at least PostgreSQL v9.6. In this guide, you will install PostgreSQL v13 from the official PostgreSQL repository.

First, add the GPG key of the PostgreSQL repository using the following command.

```
wget -q https://www.postgresql.org/media/keys/ACCC4CF8.asc -0 - | sudo apt-key add -
```

Now, add the PostgreSQL repository for the Ubuntu system using the below command.

0

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

#### sudo apt update

```
root@sonarqube-server:-#
root@sonarqube-server:-#
wget -q https://www.postgresql.org/media/keys/ACCC4CF8.asc -0 - | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).

OK
root@sonarqube-server:-#
sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt/ `lsb_release -cs`-pgdg main" >> /etc/apt/sources.list.d/pgdg.list'
root@sonarqube-server:-#
root@sonarqube-server:-# sudo apt update
Hit:1 https://mirrors.edge.kernel.org/ubuntu jammy InRelease
Hit:2 https://mirrors.edge.kernel.org/ubuntu jammy-updates InRelease
Hit:3 https://mirrors.edge.kernel.org/ubuntu jammy-backports InRelease
Hit:4 https://mirrors.edge.kernel.org/ubuntu jammy-backports InRelease
Get:5 http://apt.postgresql.org/pub/repos/apt jammy-pgdg InRelease [91.7 k8]
Get:6 http://apt.postgresql.org/pub/repos/apt jammy-pgdg/main and64 Packages [242 k8]
Fetched 334 kB in 4s (86.8 kB/s)
```

Now, install the PostgreSQL database v13 via the apt command below. Input Y when prompted to confirm the installation, then press ENTER to proceed.

sudo apt install postgresql-13

```
root@sonarqube-server:~#
root@sonarqube-server:~# sudo apt install postgresql-13
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
 libcommon-sense-perl libjson-perl libjson-xs-perl libllvm14 libpq5 libtypes-seri
  ssl-cert
Suggested packages:
 postgresql-doc-13
The following NEW packages will be installed:
 libcommon-sense-perl libjson-perl libjson-xs-perl libllvm14 libpq5 libtypes-seri
  postgresql-common ssl-cert
0 upgraded, 12 newly installed, 0 to remove and 19 not upgraded.
Need to get 41.5 MB/42.0 MB of archives.
After this operation, 168 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://apt.postgresql.org/pub/repos/apt jammy-pgdg/main amd64 postgresql-cli
```

Once PostgreSQL is installed, run the following command to verify the 'postgresql' service and to make sure the service is running.

```
sudo systemctl is-enabled postgresql
sudo systemctl status postgresql
```

You will see the output of the 'postgresql' service is enabled, which means will be run automatically at system boot. And the current status of the 'postgresql' service is running.

```
root@sonarqube-server:~# sudo systemctl is-enabled postgresql
```

```
Process: 8641 Execstart=/pin/true (code=exited, status=0/SUCCESS)

Main PID: 8641 (code=exited, status=0/SUCCESS)

CPU: 3ms
```

With the PostgreSQL installed on your server, you are ready to set up a new database and user for the SonarQube via the PostgreSQL shell.

Run the following command to log in to the PostgreSQL shell.

```
sudo -u postgres psql
```

Now, run the following PostgreSQL queries to create a new database and user for SnonarQube. In this example, you will create the PostgreSQL database and user 'sonarqube'. And be sure to change the password with a strong password.

```
CREATE USER sonarqube WITH PASSWORD 'Password';
CREATE DATABASE sonarqube OWNER sonarqube;
GRANT ALL PRIVILEGES ON DATABASE sonarqube TO sonarqube;
```

```
root@sonarqube-server:~#
root@sonarqube-server:~# sudo -u postgres psql
could not change directory to "/root": Permission denied
psql (13.8 (Ubuntu 13.8-1.pgdg22.04+1))
Type "help" for help.

postgres=# CREATE USER sonarqube WITH PASSWORD 'Password';
CREATE ROLE
postgres=# CREATE DATABASE sonarqube OWNER sonarqube;
CREATE DATABASE
postgres=# GRANT ALL PRIVILEGES ON DATABASE sonarqube TO sonarqube;
GRANT
postgres=#
```

Next, run the following queries to check the list of databases and users on the PostgreSQL server.

```
\1
\du
```

If your database and user is created, you will see the database 'sonarqube' on the list of databases, and the user 'sonarqube' on the list of users.

```
| List of databases | Name | Owner | Encoding | Collate | Ctype | Access privileges | Destarce | Destarce | LITES | Lap US LITES | Lap US LITES |
```

Lastly, log out from PostgreSQL using the query below.

\*q* 

### **Setting up System**

To install SonarQube on a Linux system, you must have a dedicated user that will be running SonarQube and some additional configurations such as ulimit and kernel parameters.

Now, you will create a new user for SonarQube, and set up custom kernel parameters via sysctl.conf file, and set up ulimit.

Run the following command to create a new user 'sonarqube' on your system.

sudo useradd -b /opt/sonarqube -s /bin/bash sonarqube

Next, open the file /etc/sysctl.conf using nano editor.

sudo nano /etc/sysctl.conf

Add the following configuration to the bottom of the line. The SonarQube required the kernel parameter vm.max\_map\_count to be greater than '524288' and the fx.file-max to be greater than '131072'.

vm.max\_map\_count=524288
fs.file-max=131072

Save the file and exit the editor when you are finished.

Now, run the sysctl command below to apply new changes on the '/etc/sysctl.conf' file.

In the following output, you can see the new kernel parameters is applied.

```
* Applying /etc/sysctl.conf ...
net.ipv6.conf.all.disable_ipv6 = 1
vm.max_map_count = 524288
fs.file-max = 131072
root@sonarqube-server:~#
```

Next, run the following command to set up ulimit for the SonarQube. This will take temporary effects on your system, when the system is rebooted, the ulimits will revert to default.

```
ulimit -n 131072
ulimit -u 8192
```

To make ulimit configuration permanently, create a new config file '/etc/security/limits.d/99-sonarqube.conf' using the following command.

```
sudo nano /etc/security/limits.d/99-sonarqube.conf
```

Add the following configuration to the file.

```
sonarqube - nofile 131072
sonarqube - nproc 8192
```

Save the file and close the editor when you are finished.

Now that you have completed the configuration of your Ubuntu system for SnonarQube installation. You will be downloading the SonarQube package and setting up SonarQube installation in the next step.

# **Downloading SonarQube Package**

The SonarQube can be installed in two different ways, via zip file and Docker image. In this example, you will install the SonarQube via the zip file package that you will download from the official SonarQube download page.

At the time of this writing, the SonarQube latest version v9.6.1, which you will be installing in the following steps.

Before downloading the SonarQube package, run the following apt command to install a basic package such as unzip and wget.

sudo apt install unzip software-properties-common wget

```
wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-9.6.1.59531.zip
```

After the SonarQube is downloaded, you will see the zip file 'sonarqube-9.6.1.59531.zip' on your working directory.

Extract the SonarQube package using the unzip command below. You should get a new directory 'sonarqube-9.6.1.59531' where the SonarQube package is stored.

```
unzip sonarqube-9.6.1.59531.zip
```

Move the directory 'sonarqube-9.6.1.59531' to the '/opt/sonarqube' using the below command.

```
mv sonarqube-9.6.1.59531 /opt/sonarqube
```

Lastly, change the ownership of the SonarQube installation directory '/opt/sonarqube' to the user 'sonarquba' via the chown command as below.

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

Now you have downloaded the SonarQube package to the installation directory '/opt/sonarqube'. Next, you will configure your SonarQube installation, then set up a systemd service file for SonarQube.

```
root@sonarqube-server:~#
root@sonarqube-server:~# mv sonarqube-9.6.1.59531 /opt/sonarqube
root@sonarqube-server:~#
root@sonarqube-server:~# sudo chown ~R sonarqube:sonarqube /opt/sonarqube
root@sonarqube-server:~#
root@sonarqube-server:~# ls /opt/sonarqube/
bin conf COPYING data dependency-license.json elasticsearch extensions lib logs temp web
root@sonarqube-server:~#
root@sonarqube-server:~#
root@sonarqube-server:~#
```

## **Configuring SonarQube**

After downloading the SonarQube package, you will set up the SonarQUbe installation by editing the default config file '/opt/sonarqube/conf/sonar.properties'.

You will add the PostgreSQL database details, set up the max memory heap for the Elasticsearch process, and set up the web host and port for the SonarQube service via the file '/opt/sonarqube/conf/sonar.properties'. And lastly, you will set up SonarQube as a systemd service.

Now, open the SonarQube configuration file '/opt/sonarqube/conf/sonar.properties' using nano editor.

11 of 18 10/6/23, 18:56

For the database configuration, uncomment some of the following options and change the default value using your database details.

```
sonar.jdbc.username=sonarqube
sonar.jdbc.password=Password
sonar.jdbc.url=jdbc:postgresql://localhost:5432/sonarqube
```

Now, uncomment the following configuration to set up the max heap memory size for the elasticsearch process. In his example, the max heap will be 512 MB.

```
sonar.search.java 0 pts = -Xmx 512m - XMs 512m - XX: Max Direct Memory Size = 256m - XX: + Heap Dump On Out Of Memory Error \\
```

Lastly, uncomment and change the following configurations to set up the IP address and port of the SonarQube will be running. Also, the log level will be 'INFO" and stored in the 'logs' directory of the SonarQube installation directory.

```
sonar.web.host=127.0.0.1
sonar.web.port=9000
sonar.web.javaAdditionalOpts=-server
sonar.log.level=INFO
sonar.path.logs=logs
```

Save the file and exit the editor when you are finished.

After you have finished the SonarQube configuration. Now, you will set up the systemd service file for SonarQube. This allows you easily to control the SonarQube process by using the systematic command.

Run the following command to create a new systemd service file '/etc/systemd/system/sonarqube.service'.

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

Add the following configuration to the 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=sonarqube
Group=sonarqube
Restart=always
LimitNOFILE=65536
LimitNPROC=4096
```

Save the file and exit the editor when you are done.

Now, reload the systemd manager by using the following command.

```
sudo systemctl daemon-reload
```

After that, start and enable the 'sonarqube.service' via the systematl command below.

```
sudo systemctl start sonarqube.service sudo systemctl enable sonarqube.service
```

Lastly, verify the 'sonarqube.service' status using the following command and make sure its status is running.

```
sudo systemctl status sonarqube.service
```

You will see the output in the following screenshot. The 'sonarqube.service' status is running, and it's also enabled, which means it will automatically run at system boot.

```
root@sonarqube-server:~#
root@sonarqube-server:~# sudo systemctl status sonarqube.service

• sonarqube.service - SonarQube service

Loaded: loaded (/etc/systemd/system/sonarqube.service; enabled; vendor preset: enabled)

Active: active (running) since Tue 2022-09-13 17:38:48 CEST; 1min 1s ago

Main PID: 9100 (java)

Tasks: 116 (limit: 4575)

Memory: 1.26

CPU: 1min 41.552s

CGroup: /system.slice/sonarqube.service

-9100 java -Xms8m -Xmx32m --add-exports=java.base/jdk.internal.ref=ALL-UNNAMED --
9124 /usr/lib/jvm/java-11-openjdk-amd64/bin/java -XX:+UseG1GC -Djava.io.tmpdir=
9238 /usr/lib/jvm/java-11-openjdk-amd64/bin/java -Djava.awt.headless=true -Dfile
```

Now that you have the SonarQube is running as a systemd service, you will install and set up a reverse proxy for the SonarQube that runs on localhost with the default port TCP '9000'.

Your SonarQube installation is now running, you can now install the Nginx web server and set up server blocks that will be used as the reverse proxy for SonarQube.

Run the following apt command to install the Nginx web server to your Ubuntu system. Input Y when prompted to confirm the installation and press ENTER to proceed.

sudo apt install nginx

```
root@sonarqube-server:~#
root@sonarqube-server:~# sudo apt install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
 libdeflate0 libgd3 libjbig0 libnginx-mod-http-geoip2 libnginx-mod-http-image-filter libng
 libnginx-mod-stream-geoip2 libtiff5 libwebp7 nginx-common nginx-core
Suggested packages:
 libgd-tools fcgiwrap nginx-doc
The following NEW packages will be installed:
 libdeflate0 libgd3 libjbig0 libnginx-mod-http-geoip2 libnginx-mod-http-image-filter libng
 libnginx-mod-stream-geoip2 libtiff5 libwebp7 nginx nginx-common nginx-core
0 upgraded, 14 newly installed, 0 to remove and 19 not upgraded.
Need to get 1,314 kB of archives.
After this operation, 4,129 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 https://mirrors.edge.kernel.org/ubuntu jammy/main amd64 libdeflate0 amd64 1.10-2 [70
```

Once nginx is installed, verify the nginx service and make sure the service status is running via the systematl command as below.

```
sudo systemctl is-enabled nginx
sudo systemctl status nginx
```

You will see the output of the nginx service is enabled, which means it will run automatically at system boot. And the current status of the nginx service is running.

```
root@sonarqube-server:~#
root@sonarqube-server:~# sudo systemctl is-enabled nginx
enabled
root@sonarqube-server:~# sudo systemctl status nginx
• nginx.service - A high performance web server and a reverse proxy server
    Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
    Active: active (running) since Tue 2022-09-13 18:57:22 CEST; 4h 52min ago
    Docs: man:nginx(8)
    Process: 4247 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, st
    Process: 4248 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SU
    Main PID: 4249 (nginx)
    Tasks: 3 (limit: 4575)
```

Create a new server blocks configuration '/etc/nginx/sites-available/sonarqube.conf' using the following command.

```
sudo nano /etc/nginx/sites-available/sonarqube.conf
```

Add the following configuration to your file and be sure to change the domain name.

```
server {
    listen 80;
    server_name sonar.hwdomain.io;
    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;
}
```

Save the file and exit the editor when you are finished.

Next, activate the server block configuration 'sonarqube.conf' by creating a symlink of that file to the '/etc/nginx/sites-enabled' directory. Then, verify your Nginx configuration files.

```
sudo ln -s /etc/nginx/sites-available/sonarqube.conf /etc/nginx/sites-enabled/
sudo nginx -t
```

If you have proper and correct Nginx configuration files, you should see the output message such as "test is successfull".

Lastly, run the following systematl command below to restart the nginx service and apply the new server block configuration.

```
sudo systemctl restart nginx
```

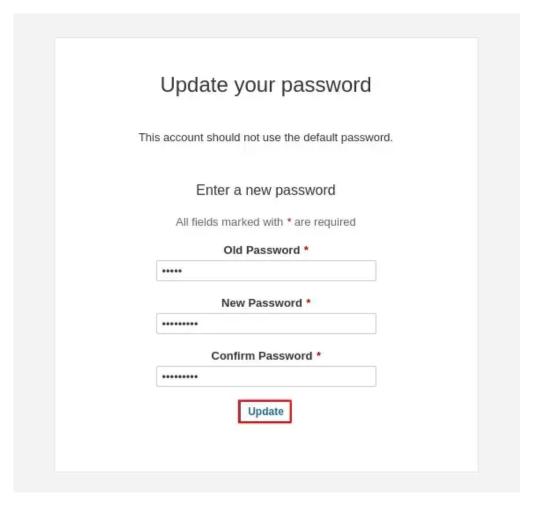
#### SonarQube Installatioon

After you have finished the reverse proxy configuration for SonarQube, now you can access your SonarQube installation via your domain and set up some basic configuration of SonarQube.

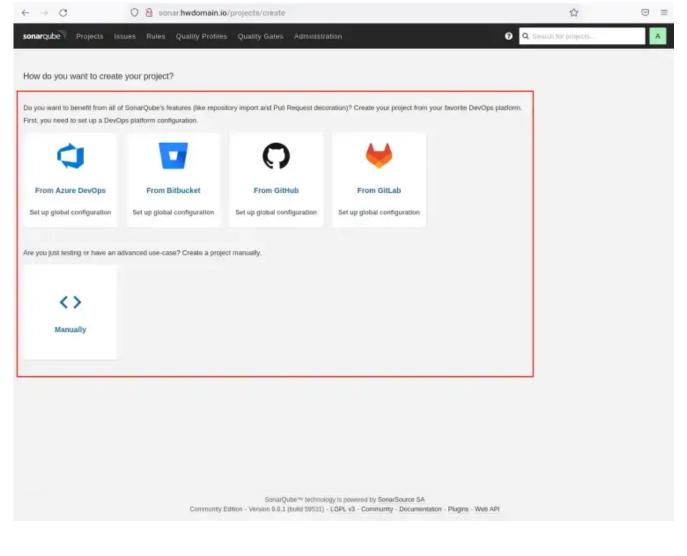
Now you will get the SonarQube login page. Input the default username and password admin/admin and click Login.



Once you logged in, you will be asked to set up a new password for SnonarQube. Input the old password admin, then input the new strong password and repeat, then click Update.



How to Install SonarQube on Ubuntu 22.04



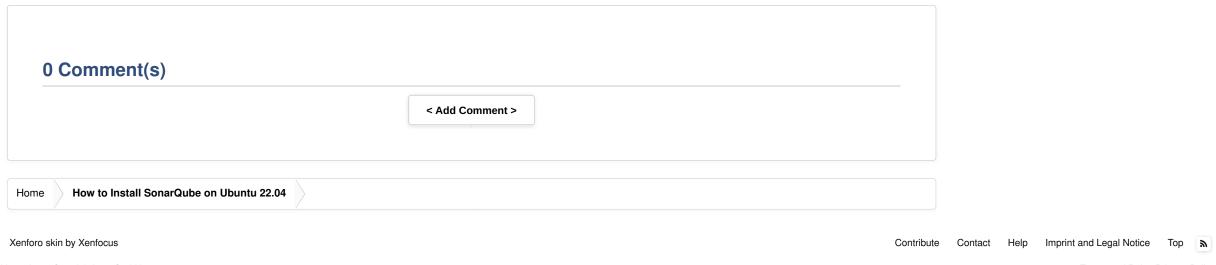
### **Conclusion**

In this guide, you have installed SonarQube on the latest Ubuntu 22.04 server. Also, you have completed the basic installation of the PostgreSQL database via the official PostgreSQL repository and installed the Nginx web server. Afterward, you have also configured the reverse proxy for the SOnarQube with Nginx server blocks.

SonarQube is now installed on your Ubuntu server. You can now add the project that you will be analyzing to the SonarQube via the git repository.

<sup>™</sup> view as pdf | <sup>™</sup> print

Share this page:



Howtoforge © projektfarm GmbH. Terms and Rules Privacy Policy