SonarQube Linux(CentOS 8.x) 설치하기

이 문서는 SonarQube를 Linux에 설치하기 위한 가이드를 공유하기 위해 작성되었다.

도구명	SonarQube Community
버전	9.4
OS	Rocky Linux 8.5
비고	

사전 환경 구성

1. Linux 환경 설정

Linux에서 root 권한으로 SonarQube 실행이 불가능하여, OS계정이 필요하다.

```
# OS
adduser --system --no-create-home sonarqube sonarqube
```

Linux에서 SonarQube를 설치하기 위해서는 다음 사항이 충족되어야 한다.

- vm.max_map_count 값 >= 524288
- fs.file-max 값 >= 131072
- SonarQube 실행 유저는 최소 131072 이상의 File descriptors를 열 수 있어야함
- SonarQube 실행 유저는 최소 8192 이상의 Threads를 열 수 있어야함

다음 명령어를 통해 현재 설정된 값 확인이 가능하다.

```
sysctl vm.max_map_count
sysctl fs.file-max
ulimit -n
ulimit -u
```

```
[sonarqube@jhyun-guide-rockey8 ~]$ sysctl vm.max_map_count
vm.max_map_count = 65530
[sonarqube@jhyun-guide-rockey8 ~]$ sysctl fs.file-max
fs.file-max = 1612738
[sonarqube@jhyun-guide-rockey8 ~]$ ulimit -n
1024
[sonarqube@jhyun-guide-rockey8 ~]$ ulimit -u
63175
[sonarqube@jhyun-guide-rockey8 ~]$
```

다음 명령어를 수행하여 SonarQube를 실행할 OS계정에 위 사항들을 영구적으로 설정할 수 있다.

```
# vm.max_map_count / fs.file-max
#
sudo vim /etc/sysctl.d/99-OS.conf

vm.max_map_count=524288
fs.file-max=131072

#
sudo sysctl --system

#
sysctl vm.max_map_count
sysctl fs.file-max


# Open File descriptors / Threads
#
sudo vim /etc/security/limits.d/99-OS.conf

OS - nofile 131072
OS - nproc 8192

#
ulimit -n
ulimit -u
```

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

```
[root@jhyun-guide-rockey8 conf]# su - sonarqube
Last login: Fri Apr 22 14:06:52 KST 2022 on pts/1
[sonarqube@jhyun-guide-rockey8 ~]$ ulimit -n
131072
[sonarqube@jhyun-guide-rockey8 ~]$ ulimit -u
8192
[sonarqube@jhyun-guide-rockey8 ~]$
```

(옵션) SonarQube 설치 후 SystemD 서비스로 등록하여 실행하는 경우, 다음과 같이 Service 항목에서 설정 할 수 있다.

```
[Service]
...
LimitNOFILE=131072
LimitNPROC=8192
...
```

2.DB 생성

다음 쿼리를 통해 SonarQube에서 사용할 DB 계정 및 DB를 생성한다.

- PostgreSQL 기준
- 참고: PostgreSQL 13 Linux 설치하기

```
# DB
CREATE ROLE sonarqubeuser WITH LOGIN PASSWORD '<PASSWORD>' VALID UNTIL 'infinity';

# DB
CREATE DATABASE sonarqubedb WITH OWNER=sonarqubeuser ENCODING 'UTF8' LC_COLLATE 'en_US.UTF-8' LC_CTYPE 'en_US.
UTF-8' TEMPLATE template0 CONNECTION LIMIT=-1;

#  DB
GRANT ALL PRIVILEGES ON DATABASE sonarqubedb TO sonarqubeuser;


# ()   (public)  SonarQube

ALTER USER sonarqubeuser SET search_path to SonarQubeSchema
```

3. JDK 설치

현 LTS 기준으로 Oracle JRE 또는 OpenJDK 11 버전이 설치되어 있어야 한다.

여기서는 OpenJDK 기준으로 가이드를 진행한다.

패키지 매니저를 통해 설치

다음 명령어를 수행하여 설치 한다.

```
# OpenJDK 11
sudo dnf install -y java-11

#
java --version
```

```
[root@jhyun-guide-rockey8 conf]# dnf install java-11
Last metadata expiration check: 0:01:06 ago on Fri 22 Apr 2022 01:38:21 PM KST.
Dependencies resolved.
================================================================================
 Package                 Architecture Version                              Repository       Size
================================================================================
Installing:
 java-11-openjdk         x86_64       1:11.0.15.0.9-2.el8_5                appstream       267 k
Installing dependencies:
 adwaita-cursor-theme    noarch       3.28.0-2.el8                         appstream       646 k
 adwaita-icon-theme      noarch       3.28.0-2.el8                         appstream        11 M
 alsa-lib                x86_64       1.2.5-4.el8                          appstream       488 k
 at-spi2-atk             x86_64       2.26.2-1.el8                         appstream        88 k
 at-spi2-core            x86_64       2.28.0-1.el8                         appstream       168 k
 atk                     x86_64       2.28.1-1.el8                         appstream       270 k
 colord-libs             x86_64       1.4.2-1.el8                          appstream       234 k
 copy-jdk-configs        noarch       4.0-2.el8                            appstream        29 k
 fribidi                 x86_64       1.0.4-8.el8                          appstream        88 k
 gdk-pixbuf2-modules     x86_64       2.36.12-5.el8                        appstream       108 k
 giflib                  x86_64       5.1.4-3.el8                          appstream        50 k
 graphite2               x86_64       1.3.10-10.el8                        appstream       120 k
 gtk-update-icon-cache   x86_64       3.22.30-8.el8                        appstream        31 k
 harfbuzz                x86_64       1.7.5-3.el8                          appstream       295 k
 hicolor-icon-theme      noarch       0.17-2.el8                           appstream        47 k
 jasper-libs             x86_64       2.0.14-5.el8                         appstream       166 k
 java-11-openjdk-headless x86_64      1:11.0.15.0.9-2.el8_5                appstream        40 M
 javapackages-filesystem noarch       5.3.0-2.module+el8.3.0+125+5da1ae29  appstream        29 k
 jbigkit-libs            x86_64       2.1-14.el8                           appstream        54 k
 lcms2                   x86_64       2.9-2.el8                            appstream       163 k
 libXcomposite           x86_64       0.4.4-14.el8                         appstream        27 k
 libXcursor              x86_64       1.1.15-3.el8                         appstream        35 k
 libXdamage              x86_64       1.1.4-14.el8                         appstream        26 k
 libXfixes               x86_64       5.0.3-7.el8                          appstream        24 k
 libXft                  x86_64       2.3.3-1.el8                          appstream        66 k
 libXi                   x86_64       1.7.10-1.el8                         appstream        48 k
```

```
Complete!
[root@jhyun-guide-rockey8 conf]# java --version
openjdk 11.0.15 2022-04-19 LTS
OpenJDK Runtime Environment 18.9 (build 11.0.15+9-LTS)
OpenJDK 64-Bit Server VM 18.9 (build 11.0.15+9-LTS, mixed mode, sharing)
[root@jhyun-guide-rockey8 conf]#
```

업로드하여 수동 설치

다음 URL에서 Linux용 OpenJDK 11 최신버전을 tar.gz 파일을 다운로드하여 Linux 서버에 업로드 한다.

- Archived OpenJDK GA Releases (java.net)

	Source	Tags are jdk-12+33, jdk-12-ga
11.0.2 (build 11.0.2+9)		
Windows	**64-bit**	zip (sha256) 179 MB
Mac	**64-bit**	tar.gz (sha256) 174 MB
Linux	**64-bit**	tar.gz (sha256) 179 MB
	Source	Tags are jdk-11.0.2+9, jdk-11.0.2-ga

다음 명령어를 순서대로 수행하여 압축을 풀고, JAVA_HOME을 설정 한다.

```
#
tar xvzf openjdk-11.0.2_linux-x64_bin.tar.gz

#
mkdir /app/java11
mv jdk-11.0.2 /app/

# ()
cd /app
ln -s jdk-11.0.2 java

# SonarQube   profile JAVA_HOME
vim ~/.bash_profile

#
JAVA_HOME=/app/java11

PATH=$PATH:$JAVA_HOME/bin
export JAVA_HOME
export PATH

#
source ~/.bash_profile

echo $JAVA_HOME
java --version
```

4. 방화벽 허용 설정 (방법 생략)

SonarQube는 Web 기본 Port로 9000 을 사용한다.

따라서 방화벽을 사용할 경우 해당 포트를 허용해줘야 한다.

SonarQube 설치

여기서는 SonarQube Community 버전으로 가이드 하지만, 상용버전도 설치 방법은 동일하다.

1.SonarQube 다운로드

아래 URL에서 SonarQube zip파일을 다운로드하여 Linux 서버에 업로드 한다.

- Download | SonarQube

2.SonarQube 설치

다음 명령어를 순서대로 수행하여 zip 파일을 압축 풀고 SonarQube 디렉토리를 설치할 경로에 위치한다.

- 가이드에서는 /app/sonarqube(심볼릭 링크로 실제 디렉토리는 sonarqube-9.4.0.54424) 경로를 SONARQUBE_HOME 으로 사용한다.

```
#
unzip sonarqube-9.4.0.54424.zip

#
mkdir -p /app
mv sonarqube-9.4.0.54424 /app/

# ()
cd /app
ln -s sonarqube-9.4.0.54424 sonarqube
```

```
[root@jhyun-guide-rockey8 app]# pwd
/app
[root@jhyun-guide-rockey8 app]# ls -al
total 0
drwxr-xr-x.  3 root root  52 Apr 22 13:22 .
dr-xr-xr-x. 18 root root 235 Apr 22 13:22 ..
lrwxrwxrwx.  1 root root  22 Apr 22 13:22 sonarqube -> sonarqube-9.4.0.54424/
drwxr-xr-x. 11 root root 172 Apr  1 09:15 sonarqube-9.4.0.54424
[root@jhyun-guide-rockey8 app]#
```

$SONARQUBE-HOME 소유자 권한을 실행할 OS계정으로 변경 한다.

```
# OS

sudo chown OS:OS -R sonarqube-9.4.0.54424
sudo chown OS:OS -R sonarqube
```

```
[root@jhyun-guide-rockey8 app]# chown sonarqube:sonarqube -R sonarqube-9.4.0.54424/
[root@jhyun-guide-rockey8 app]# chown sonarqube:sonarqube -R sonarqube
[root@jhyun-guide-rockey8 app]# ls -al
total 0
drwxr-xr-x.  4 root      root       84 Apr 22 13:51 .
dr-xr-xr-x. 18 root      root      235 Apr 22 13:22 ..
lrwxrwxrwx.  1 root      root       11 Apr 22 13:51 java11 -> jdk-11.0.2/
drwxr-xr-x.  8 root      root       96 Apr 22 13:41 jdk-11.0.2
lrwxrwxrwx.  1 sonarqube sonarqube  22 Apr 22 13:22 sonarqube -> sonarqube-9.4.0.54424/
drwxr-xr-x. 11 sonarqube sonarqube 172 Apr  1 09:15 sonarqube-9.4.0.54424
[root@jhyun-guide-rockey8 app]#
```

3. DB 및 기본 설정

sonar.properties 수정

$SONARQUBE-HOME/conf/sonar.properties 파일을 열어 다음 항목들을 수정한다.

- DB PostgreSQL 연동으로 가이드하며 PostgreSQL은 JDBC Driver 라이브러리 설치 불필요함

```
## DB
# DB
sonar.jdbc.username=DB
sonar.jdbc.password=DB

# JDBC URL
# PostgreSQL     ,  5432
sonar.jdbc.url=jdbc:postgresql://IP:Port/DB


## () SonarQube Web Port
SonarQube  9000  ,
sonar.web.port=9000


## () JVM Heap
Web / CE() /  Elasticsearch  JVM Heap
#sonar.web.javaOpts=-Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError

#sonar.ce.javaOpts=-Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError

#sonar.search.javaOpts=-Xmx512m -Xms512m -XX:MaxDirectMemorySize=256m -XX:+HeapDumpOnOutOfMemoryError
```

```
#--------------------------------------------------------------------------------------------------
# DATABASE
#
# IMPORTANT:
# - The embedded H2 database is used by default. It is recommended for tests but not for
#   production use. Supported databases are Oracle, PostgreSQL and Microsoft SQLServer.
# - Changes to database connection URL (sonar.jdbc.url) can affect SonarSource licensed products.

# User credentials.
# Permissions to create tables, indices and triggers must be granted to JDBC user.
# The schema must be created first.
sonar.jdbc.username=sonarqubeuser
sonar.jdbc.password=guide123!

#----- Embedded Database (default)
# H2 embedded database server listening port, defaults to 9092
#sonar.embeddedDatabase.port=9092


#----- Oracle 12c/18c/19c
# The Oracle JDBC driver must be copied into the directory extensions/jdbc-driver/oracle/.
# Only the thin client is supported, and we recommend using the latest Oracle JDBC driver. See
# https://jira.sonarsource.com/browse/SONAR-9758 for more details.
# If you need to set the schema, please refer to http://jira.sonarsource.com/browse/SONAR-5000
#sonar.jdbc.url=jdbc:oracle:thin:@localhost:1521/XE


#----- PostgreSQL 9.6 or greater
# By default the schema named "public" is used. It can be overridden with the parameter "currentSchema".
sonar.jdbc.url=jdbc:postgresql://localhost/sonarqubedb


#----- Microsoft SQLServer 2014/2016/2017/2019 and SQL Azure
# A database named sonar must exist and its collation must be case-sensitive (CS) and accent-sensitive (AS)
# Use the following connection string if you want to use integrated security with Microsoft Sql Server
```

```
# Binding IP address. For servers with more than one IP address, this property specifies which
# address will be used for listening on the specified ports.
# By default, ports will be used on all IP addresses associated with the server.
#sonar.web.host=0.0.0.0

# Web context. When set, it must start with forward slash (for example /sonarqube).
# The default value is root context (empty value).
#sonar.web.context=
# TCP port for incoming HTTP connections. Default value is 9000.
#sonar.web.port=9000


# The maximum number of connections that the server will accept and process at any given time.
# When this number has been reached, the server will not accept any more connections until
# the number of connections falls below this value. The operating system may still accept connections
# based on the sonar.web.connections.acceptCount property. The default value is 50.
#sonar.web.http.maxThreads=50

# The minimum number of threads always kept running. The default value is 5.
#sonar.web.http.minThreads=5

# The maximum queue length for incoming connection requests when all possible request processing
# threads are in use. Any requests received when the queue is full will be refused.
# The default value is 25.
#sonar.web.http.acceptCount=25

# The number of milliseconds this Connector will wait for another HTTP request before closing the
# connection. The default value is to use the value that has been set for the connectionTimeout
# attribute. Use a value of -1 to indicate no (i.e. infinite) timeout.
# The default value is 60000 (ms).
#sonar.web.http.keepAliveTimeout=60000

# By default users are logged out and sessions closed when server is restarted.
# If you prefer keeping user sessions open, a secret should be defined. Value is
# HS256 key encoded with base64. It must be unique for each installation of SonarQube.
```

```
#--------------------------------------------------------------------------------------------------
# WEB SERVER
# Web server is executed in a dedicated Java process. By default heap size is 512MB.
# Use the following property to customize JVM options.
#    Recommendations:
#
#    The HotSpot Server VM is recommended. The property -server should be added if server mode
#    is not enabled by default on your environment:
#    http://docs.oracle.com/javase/8/docs/technotes/guides/vm/server-class.html
#
#    Startup can be long if entropy source is short of entropy. Adding
#    -Djava.security.egd=file:/dev/./urandom is an option to resolve the problem.
#    See https://wiki.apache.org/tomcat/HowTo/FasterStartUp#Entropy_Source
#
#sonar.web.javaOpts=-Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError

# Same as previous property, but allows to not repeat all other settings like -Xmx
#sonar.web.javaAdditionalOpts=
```

```
#--------------------------------------------------------------------------------------------------
# COMPUTE ENGINE
# The Compute Engine is responsible for processing background tasks.
# Compute Engine is executed in a dedicated Java process. Default heap size is 512MB.
# Use the following property to customize JVM options.
#    Recommendations:
#
#    The HotSpot Server VM is recommended. The property -server should be added if server mode
#    is not enabled by default on your environment:
#    http://docs.oracle.com/javase/8/docs/technotes/guides/vm/server-class.html
#
#sonar.ce.javaOpts=-Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError

# Same as previous property, but allows to not repeat all other settings like -Xmx
#sonar.ce.javaAdditionalOpts=
```

```
#--------------------------------------------------------------------------------------------------
# ELASTICSEARCH
# Elasticsearch is used to facilitate fast and accurate information retrieval.
# It is executed in a dedicated Java process. Default maximum heap size is 512MB.
# It is recommended to also set MaxDirectMemorySize (-XX:MaxDirectMemorySize) and set it to half the maximum he
ap size.
#
# ------------------------------------------------
# Word of caution for Linux users on 64bits systems
# ------------------------------------------------
# Please ensure Virtual Memory on your system is correctly configured for Elasticsearch to run properly
# (see https://www.elastic.co/guide/en/elasticsearch/reference/5.5/vm-max-map-count.html for details).
#
# When SonarQube runs standalone, a warning such as the following may appear in logs/es.log:
#      "max virtual memory areas vm.max_map_count [65530] is too low, increase to at least [262144]"
# When SonarQube runs as a cluster, however, Elasticsearch will refuse to start.
#

# JVM options of Elasticsearch process
#sonar.search.javaOpts=-Xmx512m -Xms512m -XX:MaxDirectMemorySize=256m -XX:+HeapDumpOnOutOfMemoryError

# Same as previous property, but allows to not repeat all other settings like -Xmx
#sonar.search.javaAdditionalOpts=

# Elasticsearch port for incoming HTTP connections. Default is 9001. Use 0 to get a free port.
# As a security precaution, should be blocked by a firewall and not exposed to the Internet.
#sonar.search.port=9001

# Elasticsearch TCP transport port that is bound to loopback address. When nothing is set, a random port will b
e chosen.
# As a security precaution, your OS configuration should not expose this port for external access.
#sonar.es.port=
```

wrapper.conf 수정

(옵션) Java 실행 경로를 특별히 설정해야 할 경우, 다음 $SONARQUBE-HOME/conf/wrapper.conf 파일을 열어 다음 항목을 수정한다.

```
# Java     ex) /app/java11/bin/java
wrapper.java.command=java
```

```
# Path to JVM executable. By default it must be available in PATH.
# Can be an absolute path, for example:
#wrapper.java.command=/path/to/my/jdk/bin/java
wrapper.java.command=java


#
# DO NOT EDIT THE FOLLOWING SECTIONS
#


#*****************************************************************
# Wrapper Java
#*****************************************************************
wrapper.java.additional.1=-Dsonar.wrapped=true
wrapper.java.additional.2=-Djava.awt.headless=true
# extra args needed by hazelcast
wrapper.java.additional.3=--add-exports=java.base/jdk.internal.ref=ALL-UNNAMED
wrapper.java.additional.4=--add-opens=java.base/java.lang=ALL-UNNAMED
wrapper.java.additional.5=--add-opens=java.base/java.nio=ALL-UNNAMED
```

sonar.sh 수정

$SONARQUBE-HOME/bin/linux-x86-64/sonar.sh 파일을 열어 다음 항목을 수정한다.

```
#      SonarQube   OS
RUN_AS_USER=OS
```

```
# If specified, the Wrapper will be run as the specified user.
# IMPORTANT - Make sure that the user has the required privileges to write
#  the PID file and wrapper.log files.  Failure to be able to write the log
#  file will cause the Wrapper to exit without any way to write out an error
#  message.
# NOTE - This will set the user which is used to run the Wrapper as well as
#  the JVM and is not useful in situations where a privileged resource or
#  port needs to be allocated prior to the user being changed.
RUN_AS_USER=sonarqube

# The following two lines are used by the chkconfig command. Change as is
#  appropriate for your application.  They should remain commented.
# chkconfig: 2345 20 80
# description: Test Wrapper Sample Application
```

4. SonarQube 실행

다음 명령어를 사용하여 SonarQube를 실행한다.

```
# SonarQube
$SONARQUBE-HOME/bin/linux-x86-64/sonar.sh start

# SonarQube
$SONARQUBE-HOME/bin/linux-x86-64/sonar.sh stop


# ()        $SONARQUBE-HOME/bin/linux-x86-64/sonar.shUsage: ./sonar.sh { console | start | stop | force-stop |
restart | status | dump }
```

5. (옵션) Linux SystemD 서비스로 등록하여 사용

다음 명령어를 통해 SonarQube를 Linux SystemD 서비스로 등록하여 사용 할 수 있다.

```
#
sudo vim /etc/systemd/system/sonarqube.service


# sonarqube.service
-----------------------------------------------------
[Unit]
Description=SonarQube service
After=syslog.target network.target

[Service]
Type=simple
User=OS
Group=OS
PermissionsStartOnly=true
ExecStart=/bin/nohup (  Java /)java -Xms32m -Xmx32m -Djava.net.preferIPv4Stack=true -jar $SONARQUBE-HOME/lib
/sonar-application-.jar
StandardOutput=syslog
LimitNOFILE=131072
LimitNPROC=8192
TimeoutStartSec=5
Restart=always
SuccessExitStatus=143

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

# SonarQube
sudo systemctl enable sonarqube.service

# SystemD SonarQube
sudo systemctl start sonarqube.service

# SystemD SonarQube
sudo systemctl stop sonarqube.service
```

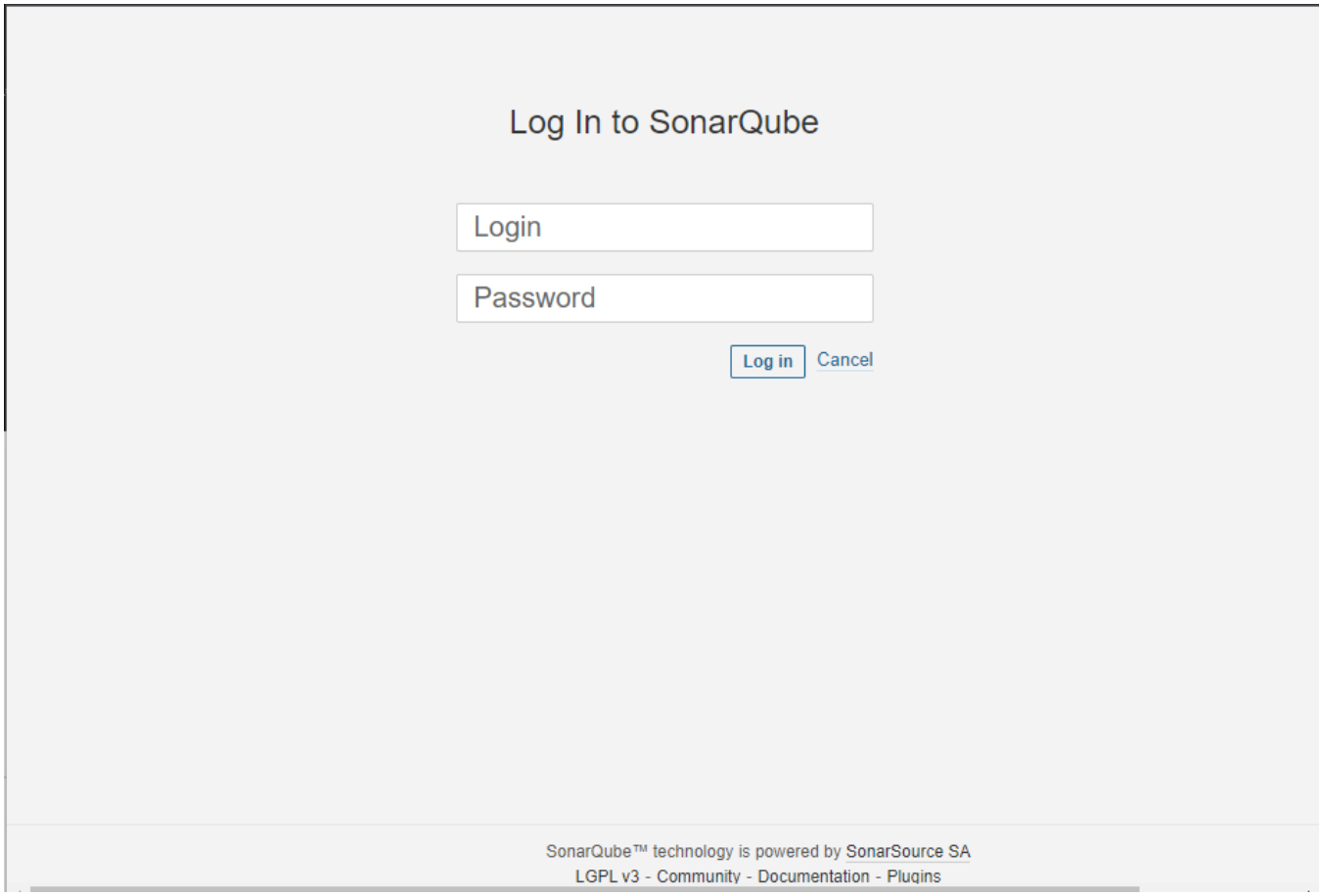
```
[Unit]
Description=SonarQube service
After=syslog.target network.target

[Service]
Type=simple
User=sonarqube
Group=sonarqube
PermissionsStartOnly=true
ExecStart=/bin/nohup java -Xms32m -Xmx32m -Djava.net.preferIPv4Stack=true -jar /app/sonarqube/lib/sonar-applica
tion-9.4.0.54424.jar
StandardOutput=syslog
LimitNOFILE=131072
LimitNPROC=8192
TimeoutStartSec=5
Restart=always
SuccessExitStatus=143

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

SonarQube 동작 확인

브라우저 접속

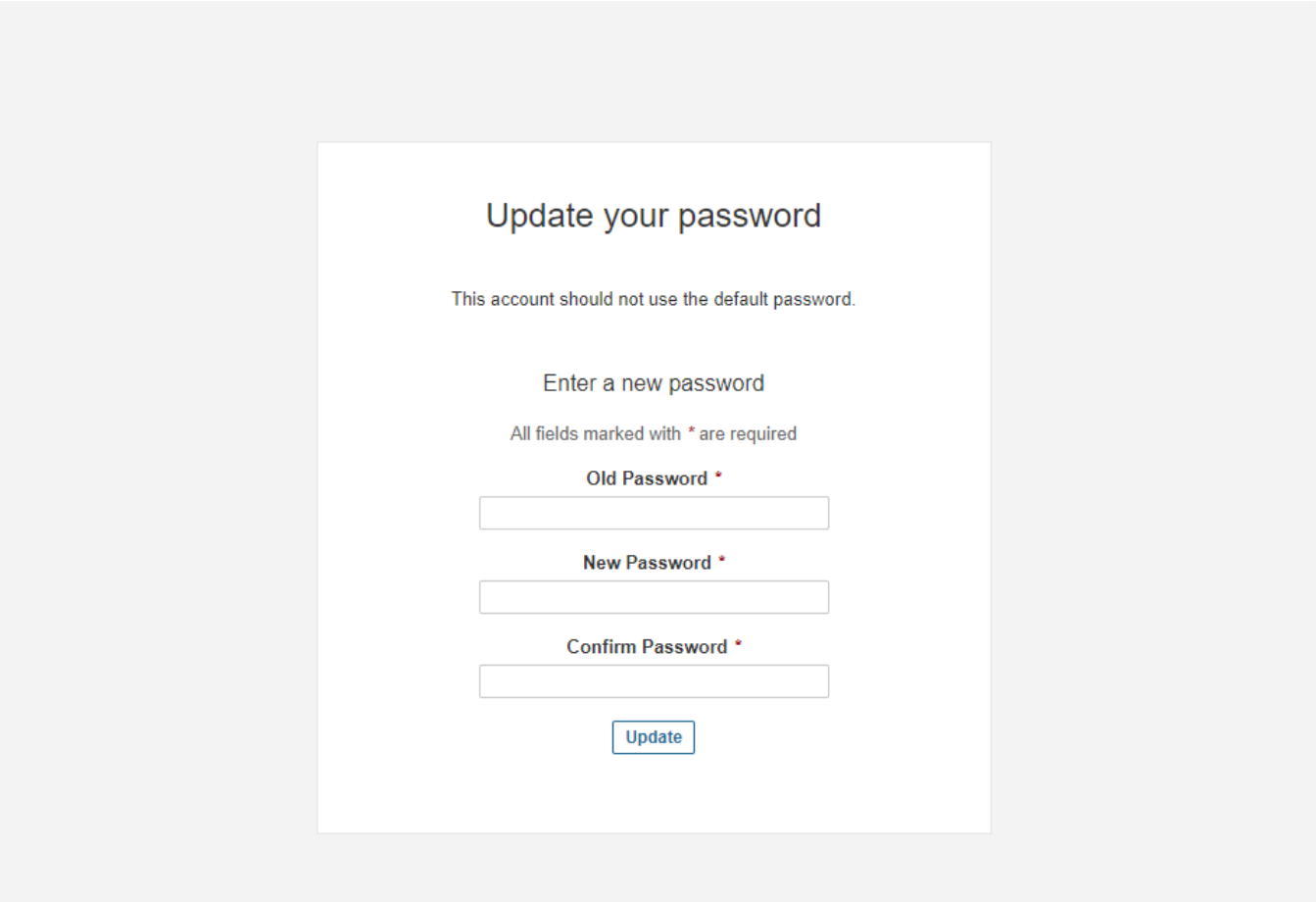
SonarQube 실행 후 구동이 완료되면 브라우저를 통해 접속할 수 있다.

- URL: http://IP:9000

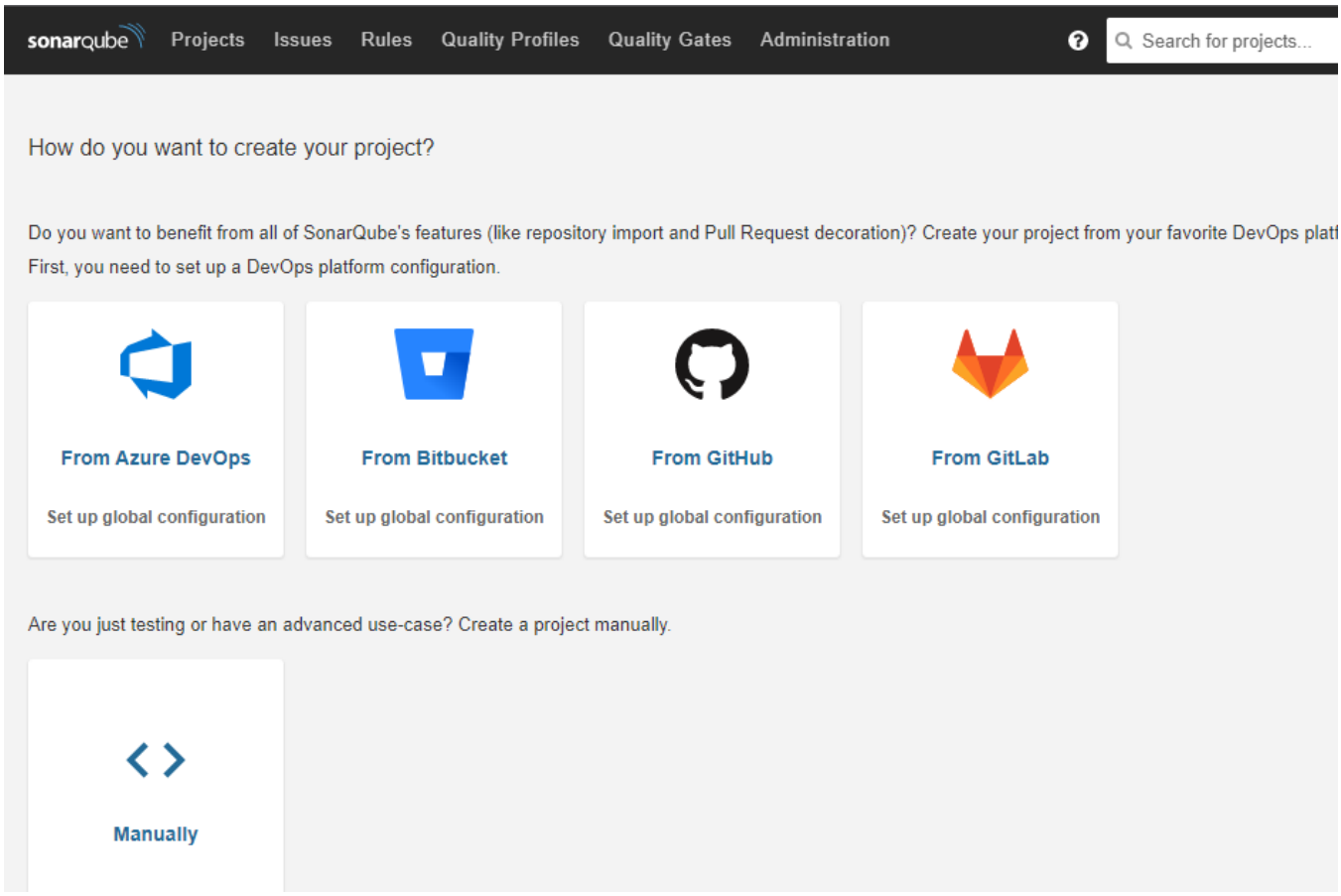


최초 로그인

초기 관리자 계정은 admin / admin 으로, 로그인 할 경우, 암호를 변경할 수 있다.



SonarQube 프로젝트 SetupWizard가 표시되면 SonarQube가 정상적으로 설치된 것을 확인할 수 있다.



참조

- Prerequisites and Overview | SonarQube Docs
- Install the Server | SonarQube Docs
- Operating the Server | SonarQube Docs
- Environment Variables | SonarQube Docs