

## INSTALACION SONARQUBE – KALI LINUX

**Paso 1:** Actualizar el sistema

```
(kali@kali)-[~]  
$ sudo apt-get update && apt-get upgrade
```

**Paso 2:** Verificamos la versión de java instalada, para esto necesitamos la versión 11

```
(kali@kali)-[~/Downloads]  
$ java -version  
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true  
openjdk version "21.0.2" 2024-01-16  
OpenJDK Runtime Environment (build 21.0.2+13-Debian-2)  
OpenJDK 64-Bit Server VM (build 21.0.2+13-Debian-2, mixed mode, sharing)
```

En este caso instalamos la versión 11:

```
(kali@kali)-[~/Downloads]  
$ sudo apt-get install openjdk-11-jre -y
```

Para seleccionar la versión que necesitamos escribimos el siguiente comando y seleccionamos la versión correcta en este caso 1 y damos enter:

```
(kali@kali)-[~/Downloads]  
$ sudo update-alternatives --config java  
There are 3 choices for the alternative java (providing /usr/bin/java).  


| Selection | Path                                        | Priority | Status      |
|-----------|---------------------------------------------|----------|-------------|
| * 0       | /usr/lib/jvm/java-21-openjdk-amd64/bin/java | 2111     | auto mode   |
| 1         | /usr/lib/jvm/java-11-openjdk-amd64/bin/java | 1111     | manual mode |
| 2         | /usr/lib/jvm/java-17-openjdk-amd64/bin/java | 1711     | manual mode |
| 3         | /usr/lib/jvm/java-21-openjdk-amd64/bin/java | 2111     | manual mode |

  
Press <enter> to keep the current choice[*], or type selection number: 1  
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/java to provide /usr/bin/java (java) in manual mode
```

**Paso 3:** Habilitamos el servicio PostgreSQL para iniciar con el arranque del sistema.

```
(kali@kali)-[~]  
$ sudo systemctl enable postgresql  
[sudo] password for kali:  
Synchronizing state of postgresql.service with SysV service script with /usr/lib/systemd/systemd-sysv-instal  
tall.  
Executing: /usr/lib/systemd/systemd-sysv-install enable postgresql  
Created symlink /etc/systemd/system/multi-user.target.wants/postgresql.service → /usr/lib/systemd/system/  
postgresql.service.
```

**Paso 4:** Iniciamos el servicio de postgresql

```
(kali㉿kali)-[~]  
$ sudo systemctl start postgresql
```

**Paso 5:** Asignamos una contraseña a postgres, que se una contraseña fuerte pero que no se les olvide

```
(kali㉿kali)-[~]  
$ sudo passwd postgres  
New password:  
Retype new password:  
passwd: password updated successfully
```

**Paso 6:** iniciamos sesión con el usuario postgres.

```
(kali㉿kali)-[~]  
$ su - postgres  
Password:  
postgres@kali:~$
```

**Paso 7:** Ahora vamos a crear un usuario, el usuario será sonarqube guion bajo y sus iniciales por ejemplo en mi caso sería: sonarqube\_shvb

```
postgres@kali:~$ createuser sonarqube_shvb  
postgres@kali:~$
```

**Paso 8:** Escribimos el siguiente comando en la terminal para que nos genere una shell interactiva PostgreSQL.

```
postgres@kali:~$ psql  
psql (16.2 (Debian 16.2-1))  
Type "help" for help.  
  
postgres=#
```

**Paso 9:** Le asignamos una contraseña al usuario que creamos:

```
postgres=# ALTER USER sonarqube_shvb WITH ENCRYPTED password 'sergio24';  
ALTER ROLE  
postgres=#
```

**Paso 10:** Creamos la base de datos:

```
postgres=# CREATE DATABASE sonarqube OWNER sonarqube_shvb;  
CREATE DATABASE  
postgres=#
```

**Paso 11:** Le asignamos permisos al usuario de la base de datos:

```
postgres=# GRANT ALL PRIVILEGES ON DATABASE sonarqube TO sonarqube_shvb;  
GRANT  
postgres=#
```

Y nos salimos:

```
postgres=# \q  
postgres@kali:~$ exit  
logout
```

**Paso 12:** Vamos a instalar SonarQube, nos vamos a la carpeta de descargas, lo descargamos de esta dirección: <https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-9.2.2.50622.zip>:

```
(kali@kali)-[~/Downloads]  
$ cd Downloads
```

```
(kali@kali)-[~/Downloads]  
$ wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-9.2.2.50622.zip
```

Ahora vamos a descomprimir el archivo:

```
(kali@kali)-[~/Downloads]  
$ sudo unzip sonarqube-9.2.2.50622.zip
```

**Paso 13:** Creamos el directorio de instalación:

```
(kali㉿kali)-[~]  
$ sudo mkdir /opt/sonarqube  
[sudo] password for kali:
```

**Paso 14:** vamos mover el contenido de la carpeta:

```
(kali㉿kali)-[~/Downloads]  
$ sudo mv sonarqube-*/*/ /opt/sonarqube/
```

**Paso 15:** agregamos un usuario del sistema para sonarqube y le damos los permisos:

```
(kali㉿kali)-[~/Downloads]  
$ sudo useradd -M -d /opt/sonarqube/ -r -s /bin/bash sonarqube
```

```
(kali㉿kali)-[~/Downloads]  
$ sudo chown -R sonarqube:sonarqube -R /opt/sonarqube/
```

**Paso 16:** Vamos al archivo de configuración de sonarqube para cambiar los parámetros de la base de datos:

```
(kali㉿kali)-[~/Downloads]  
$ sudo nano /opt/sonarqube/conf/sonar.properties
```

**Paso 17:** Realizamos los siguientes ajustes en el archivo y al terminar, guardamos y salimos del archivo:

```
#  
# 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=sonarqube_shvb  
sonar.jdbc.password=sergio24  
sonar.jdbc.url=jdbc:postgresql://localhost:5432/sonarqube
```

**Paso 18:** Creamos un servicio para sonarqube:

```
(kali㉿kali)-[~/Downloads]  
$ sudo nano /etc/systemd/system/sonarqube.service
```

**Paso 19:** Se agrega lo siguiente en el documento, al terminar guardamos control + o y salimos control + x:

```
GNU nano 7.2 /etc/systemd/system/sonarqube.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=sonarqube  
Group=sonarqube  
Restart=always  
  
LimitNOFILE=65536  
LimitNPROC=4096  
  
[Install]  
WantedBy=multi-user.target
```

```
[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
```

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

**Paso 20:** Vamos editar el archivo sonar.sh

```
(kali㉿kali)-[~/Downloads]
$ sudo nano /opt/sonarqube/bin/linux-x86-64/sonar.sh
```

Vamos a la línea 48 y realizamos el ajuste, guardamos control + o y salimos control + x:

```
# 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
```

después hay que reiniciar el demonio del sistema, y habilitamos a sonarqube para que inicie al cargar el sistema y lo iniciamos:

```
(kali㉿kali)-[~/Downloads]
$ sudo systemctl daemon-reload

(kali㉿kali)-[~/Downloads]
$ sudo systemctl enable sonarqube

(kali㉿kali)-[~/Downloads]
$ sudo systemctl start sonarqube
```

**Paso 21:** Vamos a instalar nginx:

```
(kali㉿kali)-[~/Downloads]
$ sudo apt-get install nginx -y
```

Lo habilitamos y quitamos la configuración por default:

```
(kali㉿kali)-[~/Downloads]
$ sudo systemctl enable nginx
Synchronizing state of nginx.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable nginx
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.

(kali㉿kali)-[~/Downloads]
$ sudo unlink /etc/nginx/sites-enabled/default
```

Vamos configurar el sitio:

```
(kali㉿kali)-[~/Downloads]
$ sudo nano /etc/nginx/sites-available/sonarqube.conf
```

El contenido del documento debe quedar así:

```
GNU nano 7.2 /etc/nginx/sites-available/sonarqube.conf
server {
    listen 80;
    server_name localhost;

    location / {
        proxy_pass http://127.0.0.1:9000;
    }
}
```

Habilitamos la nueva configuración, y restauramos el servicio:

```
(kali@kali)-[~/Downloads]
$ sudo ln -s /etc/nginx/sites-available/sonarqube.conf /etc/nginx/sites-enabled/sonarqube.conf

(kali@kali)-[~/Downloads]
$ sudo service nginx configtest
Testing nginx configuration:..

(kali@kali)-[~/Downloads]
$ sudo systemctl restart nginx

(kali@kali)-[~/Downloads]
$
```

## Paso 22: vamos a cambiar los límites del kernel

```
(kali@kali)-[~]
$ sudo nano /etc/sysctl.conf
```

Nos vamos al final del archivo y agregamos lo siguiente, al terminar guardamos con control +o y salimos con control + c:

```
#####
# Magic system request Key
# 0=disable, 1=enable all, >1 bitmask of sysrq functions
# See https://www.kernel.org/doc/html/latest/admin-guide/sysrq.html
# for what other values do
#kernel.sysrq=438

vm.max_map_count=262144
fs.file-max=65536
ulimit -n 65536
ulimit -u 4096

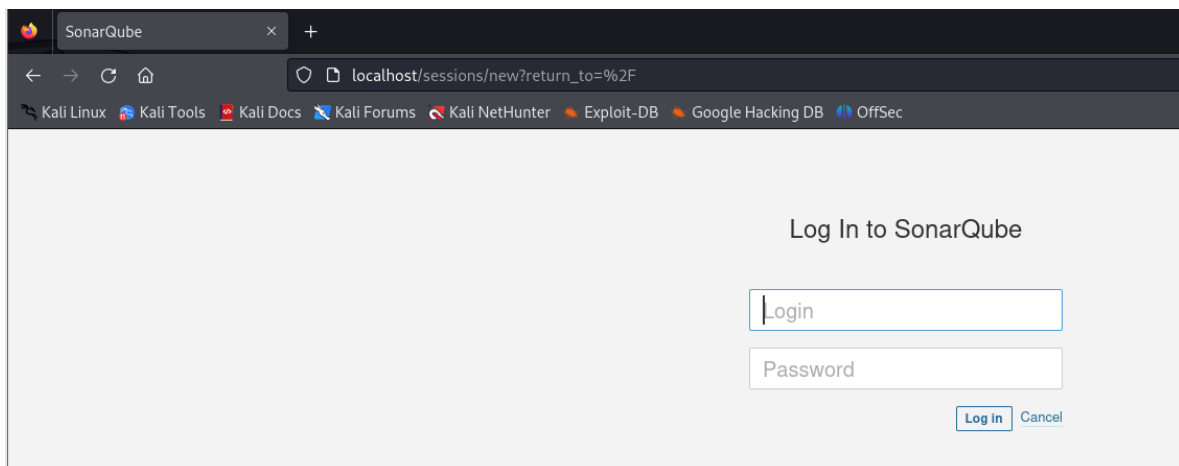
```

<b>^G</b> Help	<b>^O</b> Write Out	<b>^W</b> Where Is	<b>^K</b> Cut	<b>^T</b> Execute	<b>^C</b> Location	<b>M-U</b> Undo
<b>^X</b> Exit	<b>^R</b> Read File	<b>^_\</b> Replace	<b>^U</b> Paste	<b>^J</b> Justify	<b>^/_</b> Go To Line	<b>M-E</b> Redo

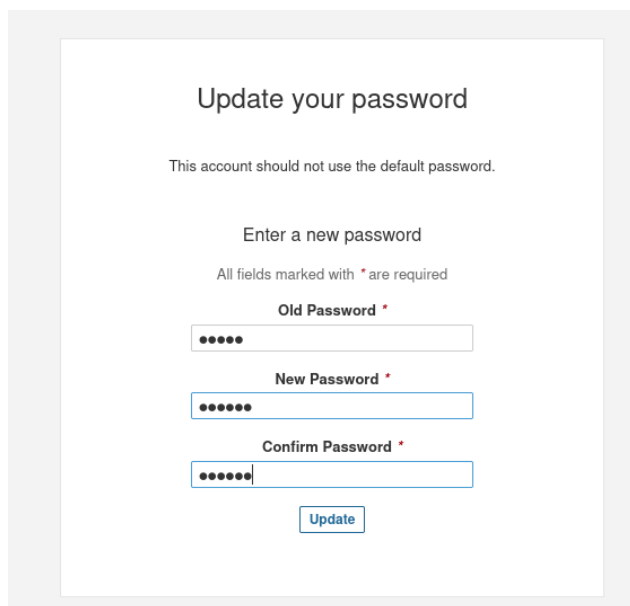
Para que la configuración entre en vigor para a escribir el siguiente comando:

```
(kali@kali)-[~]  
$ sudo systemctl --system
```

**Paso 23:** Vamos al navegador y colocamos localhost, entramos usuario admin, contraseña: admin

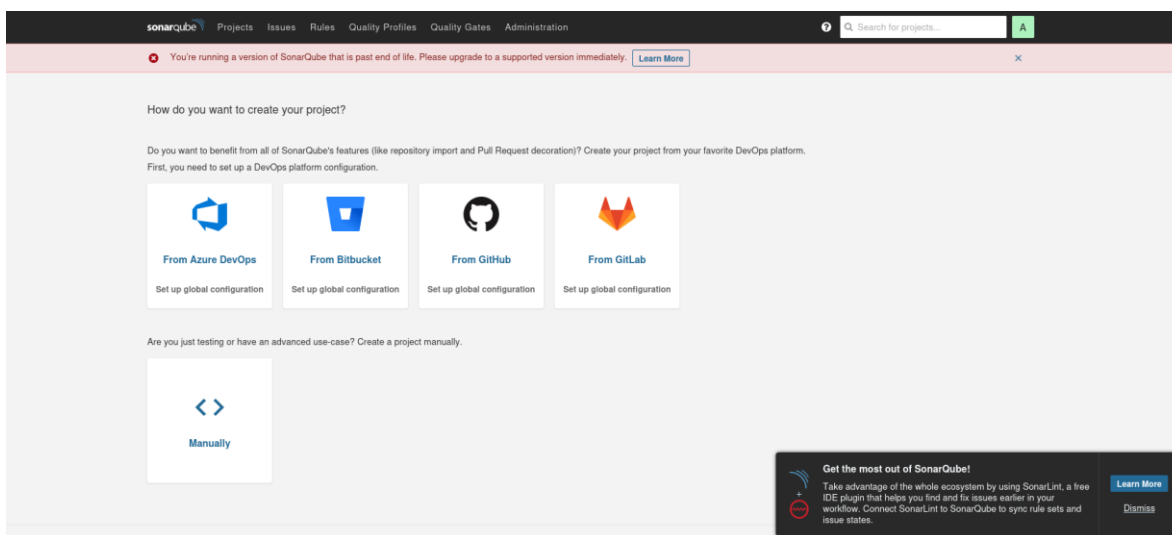


Una vez que entremos nos pedirá que cambiemos la contraseña:





Y con esto nos permitirá entrar al sitio:



**Paso 24:** Vamos a instalar el scanner, para esto crearemos la carpeta de instalación:

```
(kali@kali)-[~]  
$ sudo mkdir /opt/sonarscanner
```

Ahora nos movemos a la carpeta:

```
(kali@kali)-[~]  
$ cd /opt/sonarscanner
```

Descargamos el scanner de <https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-5.0.1.3006-linux.zip>:

```
(kali@kali)-[/opt/sonarscanner]  
$ sudo wget https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-5.0.1.3006-linux.zip
```

Descomprimos el archivo:

```
(kali@kali)-[/opt/sonarscanner]  
$ sudo unzip sonar-scanner-cli-5.0.1.3006-linux.zip
```

Vamos a editar el siguiente archivo:

```
(kali@kali)-[/opt/sonarscanner]  
$ sudo nano sonar-scanner-5.0.1.3006-linux/conf/sonar-scanner.properties
```

Quedaría de la siguiente manera:

```
GNU nano 7.2 sonar-scanner-5.0.1.3006-linux/conf/sonar-scanner.properties
#Configure here general information about the environment, such as SonarQube server c
#No information about specific project should appear here

#—— Default SonarQube server
sonar.host.url=http://localhost:9000

#—— Default source code encoding
#sonar.sourceEncoding=UTF-8
```

Lo hacemos ejecutable:

```
(kali㉿kali)-[/opt/sonarscanner]
$ sudo chmod +x sonar-scanner-5.0.1.3006-linux/bin/sonar-scanner
```

Le creamos un enlace simbólico:

```
(kali㉿kali)-[/opt/sonarscanner]
$ sudo ln -s /opt/sonarscanner/sonar-scanner-5.0.1.3006-linux/bin/sonar-scanner /usr/local/bin/sonar-scanner
```

Para utilizarlo nos vamos a Documentos:

```
(kali㉿kali)-[/opt/sonarscanner]
$ cd /home/kali/Documents
```

Creamos una nueva carpeta para realizar una prueba:

```
(kali㉿kali)-[~/Documents]
$ sudo mkdir sonar-test
```

Entramos a la carpeta:

```
(kali㉿kali)-[~/Documents]
$ cd sonar-test
```

Descargamos un proyecto de pruebas de <https://github.com/SonarSource/sonar-scanning-examples/archive/refs/heads/master.zip>:

```
(kali㉿kali)-[~/Documents/sonar-test]
$ sudo wget https://github.com/SonarSource/sonar-scanning-examples/archive/refs/heads/master.zip
```

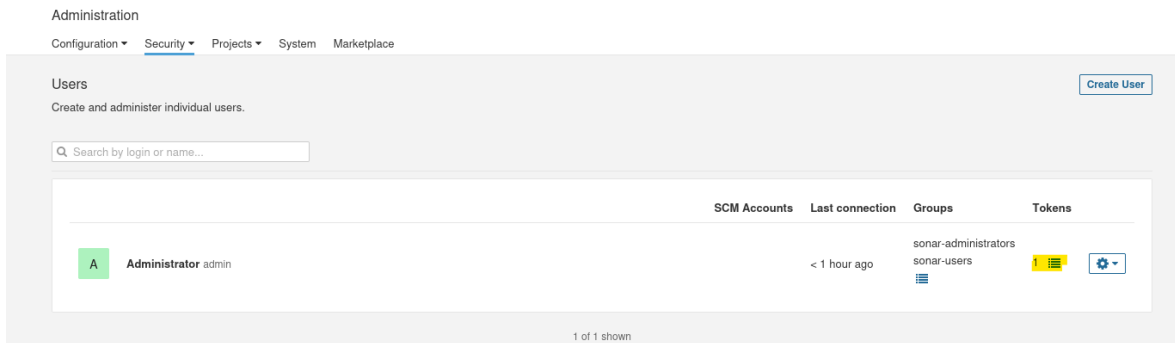
Lo descomprimos:

```
(kali㉿kali)-[~/Documents/sonar-test]
$ sudo unzip master.zip
```

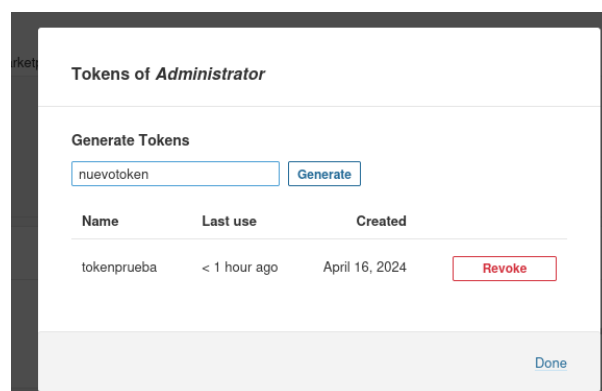
Entramos a la carpeta:

```
(kali@kali)-[~/Documents/sonar-test]
$ cd sonar-scanning-examples-master/sonar-scanner
```

Para ejecutarlo primero debemos ir al navegador entramos al sitio, en administration >> Security >> users , una vez aquí generemos un token dando clic el botón de barras que esta en la columna de tokens



Nos abrirá la siguiente ventana, le damos un nombre y después clic en el botón de Generate:



Una vez que lo genere lo copiamos:

### Tokens of Administrator

#### Generate Tokens

New token "nuevotoken" has been created. Make sure you copy it now, you won't be able to see it again!

Copy

831ed9507c2a64a7f3fbb98512947f9f6b2c31bb

Name	Last use	Created	
tokenprueba	< 1 hour ago	April 16, 2024	<input type="button" value="Revoke"/>
nuevotoken	Never	April 16, 2024	<input type="button" value="Revoke"/>

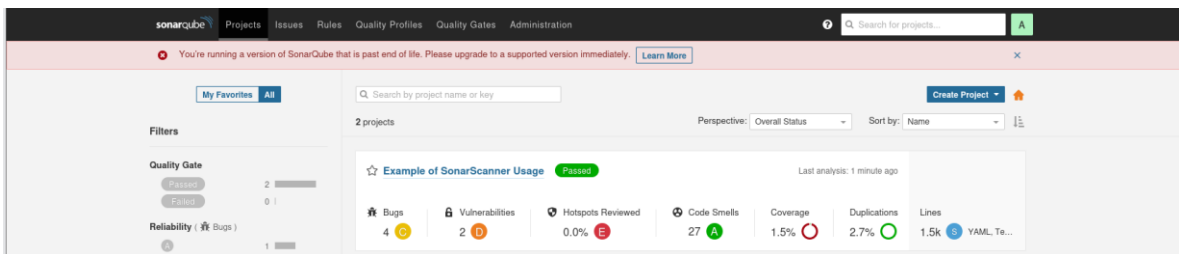
Done

Ahora nos regresamos a la terminal y vamos a ejecutar el escaneo y en la parte de login pegamos el token y damos enter :

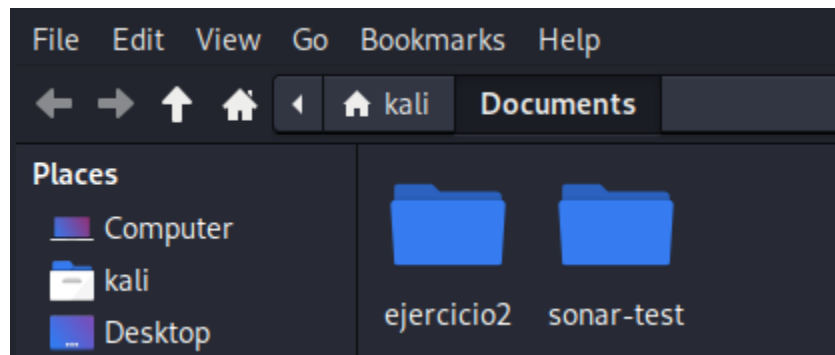
```
(kali㉿kali)-[~/Documents/sonar-test/sonar-scanning-examples-master/sonar-scanner]
$ sudo sonar-scanner -D sonar.login=831ed9507c2a64a7f3fbb98512947f9f6b2c31bb
```

```
INFO: Sensor VB.NET Properties [vbnet] (done) | time=0ms
INFO: _____ Run sensors on project
INFO: Sensor Zero Coverage Sensor
INFO: Sensor Zero Coverage Sensor (done) | time=8ms
INFO: SCM Publisher No SCM system was detected. You can use the 'sonar.scm.provider' property to explicitly specify it.
INFO: CPD Executor 3 files had no CPD blocks
INFO: CPD Executor Calculating CPD for 6 files
INFO: CPD Executor CPD calculation finished (done) | time=37ms
INFO: Analysis report generated in 86ms, dir size=328.1 kB
INFO: Analysis report compressed in 48ms, zip size=78.0 kB
INFO: Analysis report uploaded in 67ms
INFO: ANALYSIS SUCCESSFUL, you can browse http://localhost:9000/dashboard?id=org.sonarqube%3Asonar-scanner
INFO: Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
INFO: More about the report processing at http://localhost:9000/api/ce/task?id=AY7ojTobKJ2kXJ-1gJAP
INFO: Analysis total time: 10.371 s
INFO: _____
INFO: EXECUTION SUCCESS
INFO: _____
INFO: Total time: 11.389s
INFO: Final Memory: 47M/164M
INFO: _____
```

Una vez que termine regresamos al navegador y nos mostrara el escaneo que realizo:



**Paso 25:** Para analizar nuestro código, en la carpeta de documentos pegamos nuestro proyecto



Abrimos una terminal en la carpeta de documento y entramos a nuestro proyecto

```
(kali@kali)-[~/Documents]
$ cd ejercicio2
```

Vamos a crear un documento:

```
(kali㉿kali)-[~/Documents/ejercicio2]  
$ sudo nano sonar-project.properties
```

Con la siguiente información en el projectName escriben su nombre, guardamos y salimos:

```
File Actions Edit View Help  
GNU nano 7.2 sonar-project.properties  
sonar.projectKey=spring-boot  
sonar.projectName=Proyecto de Sergio Humberto Vazquez Barrientos  
sonar.projectVersion=1.0  
sonar.sources=src/main/java/  
sonar.language=java  
sonar.java.binaries=target/classes  
WARNING: Detected SonarLint and disabled part of IDE. Please upgrade to a supported version immediately. [Learn More]
```

Y ahora si podemos ejecutar el escaneo de nuestro código:

```
(kali㉿kali)-[~/Documents/ejercicio2]  
$ sudo sonar-scanner -D sonar.login=831ed9507c2a64a7f3fbb98512947f9f6b2c31bb  
  
INFO: Analysis report uploaded in 60ms  
INFO: ANALYSIS SUCCESSFUL, you can browse http://localhost:9000/dashboard?id=spring-boot  
INFO: Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report  
INFO: More about the report processing at http://localhost:9000/api/ce/task?id=AY7oLCY5KJ2kXJ-1gJAQ  
INFO: Analysis total time: 5.472 s  
INFO: _____  
INFO: EXECUTION SUCCESS  
INFO: _____  
INFO: Total time: 6.512s  
INFO: Final Memory: 8M/37M  
INFO: _____
```

Al terminar vamos al navegador entramos al sitio y revisamos el resultado:

Proyecto de Sergio Humberto Vazquez Barrientos Passed Last analysis: 11 seconds ago

Bugs	Vulnerabilities	Hotspots Reviewed	Code Smells	Coverage	Duplications	Lines
0 <span>A</span>	0 <span>A</span>	- <span>A</span>	9 <span>A</span>	0.0% <span>C</span>	0.0% <span>G</span>	392 <span>XS</span> Java

sonarqube

ProjectsIssuesRulesQuality ProfilesQuality GatesAdministration

Search for projects...

A

You're running a version of SonarQube that is past end of life. Please upgrade to a supported version immediately.

Learn More

x

Proyecto de Sergio Humberto Vazquez Barrientos

master

Last analysis had 2 warnings

April 16, 2024 at 4:27 PM

Version 1.0

OverviewIssuesSecurity HotspotsMeasuresCodeActivity

Project SettingsProject Information

To benefit from more of SonarQube's features, set up analysis in your favorite CI.

x

QUALITY GATE STATUS

Passed

All conditions passed.

MEASURES

New Code

Since April 16, 2024

Started 1 hour ago

Overall Code

0New Bugs

ReliabilityA

0New Vulnerabilities

SecurityA

0New Security Hotspots

Reviewed

Security ReviewA

0Added Debt

0New Code Smells

MaintainabilityA