



SonarQube

Code Review with SonarQube

Divya Satpute

Dreams Source

SONARQUBE

Divya vasasatputesatpute

 SonarQube Tutorial: Improve Your Code Quality 

How to Use SonarQube for Better Code Quality

Table of Contents

- What is SonarQube??
- what is Code Review??
- How To Setup Sonar Server??
- Sonar Admin logins??
- Sonar Rules??
- Sonar Quality Profile and Making Custom Quality Profile as default??
- Using Sonar Quality Profile and Custom Quality Profile as a particular Project??
- Sonar Quality Gate??
- Integrated Sonar Server with Java Maven app??
- Sonar Token Generation

SONARQUBE

Divya vasasatputesatpute



What is SonarQube?



- SonarQube is an open-source platform that helps developers manage code quality and security. It provides insightful metrics and continuous inspection for your codebase. Here's a quick overview:
- SonarQube is Continuous Code Quality Checking Tool
- We can do code Review using SonarQube Tool
- SonarQube is an open-source Software Quality Management Tool
- It will continuously analyse and measures quality of the source code
- It will be generate code review report in html format /pdf format
- It is web based tool and it support 29 programming languages
- It will be support multi OS platform
- It will be support multiple databases (MySQL, Oracle , SQL server)
- It supports multiple Browsers

Divya_satpute

@learnwithdivya.hashnode.dev

SONARQUBE

Divya vasasatputesatpute

- Initially SonarQube was developed only for java projects
- Today SonarQube is supports 29 programming languages
 - *SonarQube will identify below category of issue in project source code*
 - Duplicate code
 - Coding Standards
 - Unit Tests
 - Code Coverage
 - Complex Code
 - Commented Code
 - Potential Bugs





What is Code Coverage & Code Review

Code Coverage: How many lines of source code is tested by Unit test Cases

NOTE: Industry standard code coverage is 80 %

Code Review: Checking Coding Conventions

Key Features:

- **Code Quality** : Automatically checks your code for bugs, vulnerabilities, and code smells.
- **Continuous Integration** : Integrates seamlessly with CI/CD pipelines, ensuring quality checks with every commit.
- **Multi-Language Support** : Supports over 25 programming languages, making it versatile for any project.
- **Real-time Feedback** : Developers receive immediate feedback on code quality to encourage best practices.

SONARQUBE

Divya vasasatputesatpute

Environment Setup

Pre-requisites:

Java is installed on your Machine

if SonarQube 7.6 ----> Java 1.8 version is installed

if SonarQube 7.6 -----> java 11 version is Installed

Note: We can Check this Compatability in official SonarQube Website

Hardware Requirements

Minimum RAM: 2 GB

SONARQUBE

Divya vasasatputesatpute

Step for deploy SonarQube

Create EC2 Instance with 4 GB Ram (t2.medium)

Connect with EC2 instance using Git-Bash

```
$ sudo -i
```

Install wget package using following command

```
$sudo apt install wget -y
```

change directory

```
$cd /opt
```

Install java openjdk

```
$sudo apt install openjdk-11-jdk -y
```

```
$java -version
```

Install java openjdk

unzip package install

```
$sudo apt install unzip -y
```

Divya_satpute

@learnwithdivya.hashnode.dev

SONARQUBE

Divya vasasatputesatpute

install SonarQube zip file change version according to your version

```
$wget
```

<https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-8.4.0.35506.zip>

unzip file

```
$unzip sonarqube-8.4.0.35506.zip
```

Note: SonarQube Server Will Not Run With Root User
we have to create new user called Sonar

create new user

```
$useradd sonar
```

```
$visudo
```

File be look like below image we have to configure user n passwd

sonar ALL=(ALL) NOPASSWD: ALL

Divya_satpute
@learnwithdivya.hashnode.dev

SONARQUBE

Divya vasasatputesatpute

```
GNU nano 7.2 /etc/sudoers.tmp

# Ditto for GPG agent
#Defaults:%sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
sonar    ALL=(ALL) NOPASSWD: ALL
# Members of the admin group may gain root privileges
%admin   ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo    ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "@include" directives:

@include /etc/sudoers.d
^G Help      ^O Write Out  ^W Where Is  ^K Cut       ^T Execute   ^C Location  M-U Undo
^X Exit      ^R Read File  ^_ Replace   ^U Paste     ^J Justify   ^_ Go To Line M-E Redo
? main* 0 0 0 0 0
```

change Ownership for Sonar Folder

```
$chown -R sonar:sonar /opt/sonarqube-8.4.0.35506
```

change file permission

```
$chmod -R 775 /opt/sonarqube-8.4.0.35506
```

Divya_satpute
@learnwithdivya.hashnode.dev

SONARQUBE

Divya vasasatputesatpute

Now switch to Sonar user

```
$su – sonar
```

Go to SonarQube folder

```
$cd sonarqube-8.4.0.35506
```

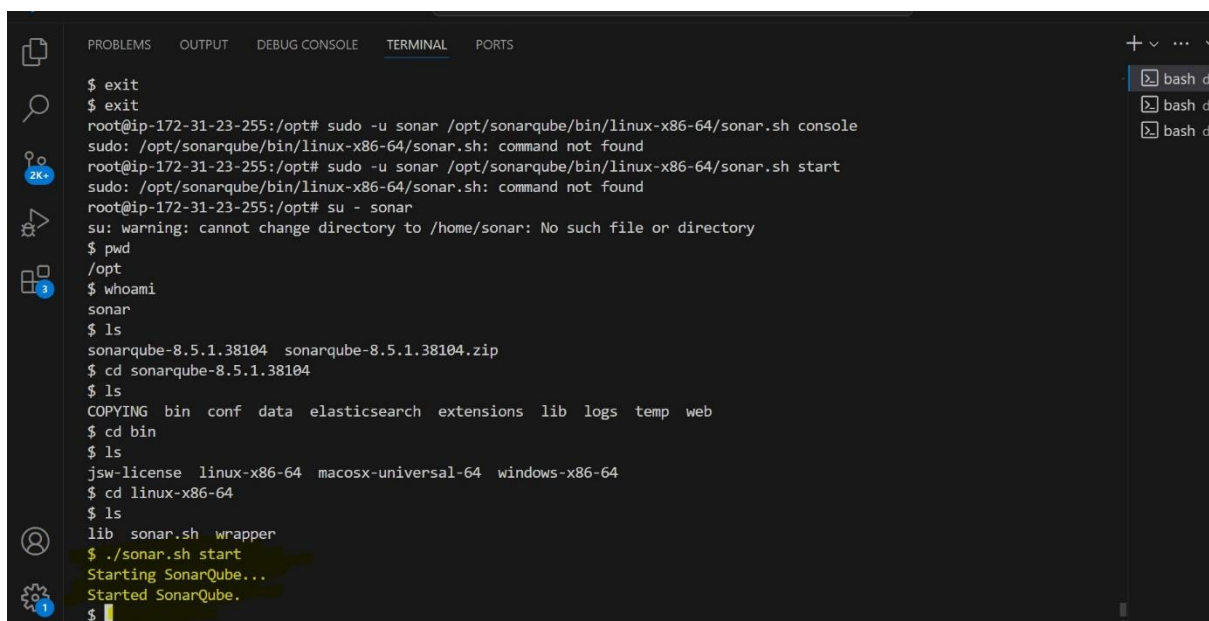
```
$cd bin
```

```
$cd linux-x86-64
```

```
$./sonar.sh start
```

Congratulations to yourself! Celebrate your achievements and continue to shine."

you did it!!!! Now your SonarQube started it look like this:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
$ exit
$ exit
root@ip-172-31-23-255:/opt# sudo -u sonar /opt/sonarqube/bin/linux-x86-64/sonar.sh console
sudo: /opt/sonarqube/bin/linux-x86-64/sonar.sh: command not found
root@ip-172-31-23-255:/opt# sudo -u sonar /opt/sonarqube/bin/linux-x86-64/sonar.sh start
sudo: /opt/sonarqube/bin/linux-x86-64/sonar.sh: command not found
root@ip-172-31-23-255:/opt# su - sonar
su: warning: cannot change directory to /home/sonar: No such file or directory
$ pwd
/opt
$ whoami
sonar
$ ls
sonarqube-8.5.1.38104  sonarqube-8.5.1.38104.zip
$ cd sonarqube-8.5.1.38104
$ ls
COPYING  bin  conf  data  elasticsearch  extensions  lib  logs  temp  web
$ cd bin
$ ls
jsw-license  linux-x86-64  macosx-universal-64  windows-x86-64
$ cd linux-x86-64
$ ls
lib  sonar.sh  wrapper
$ ./sonar.sh start
Starting SonarQube...
Started SonarQube.
$
```

Divya_satpute

@learnwithdivya.hashnode.dev

SONARQUBE

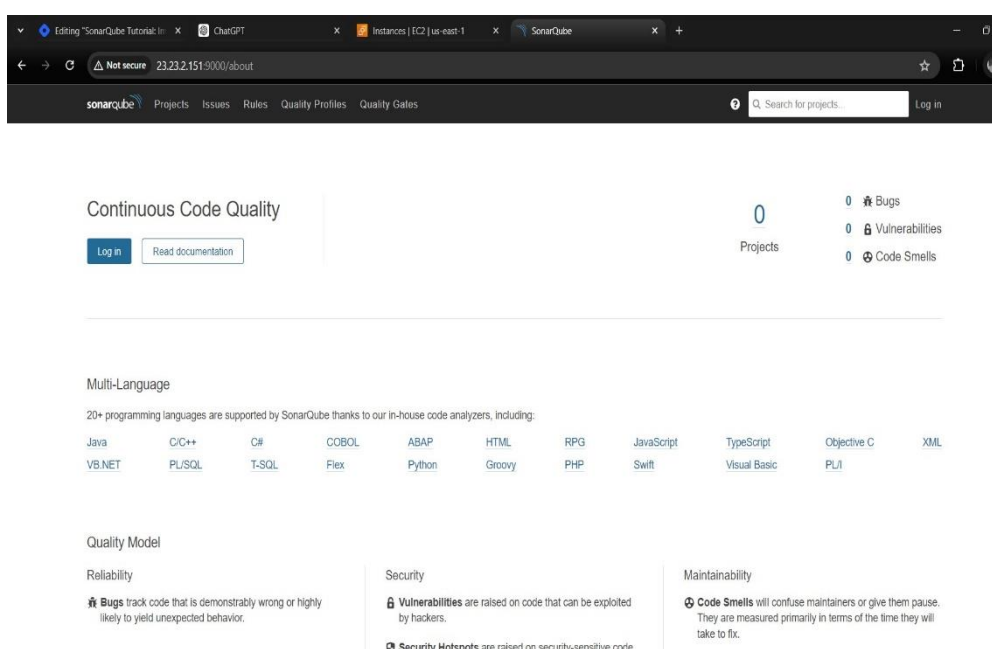
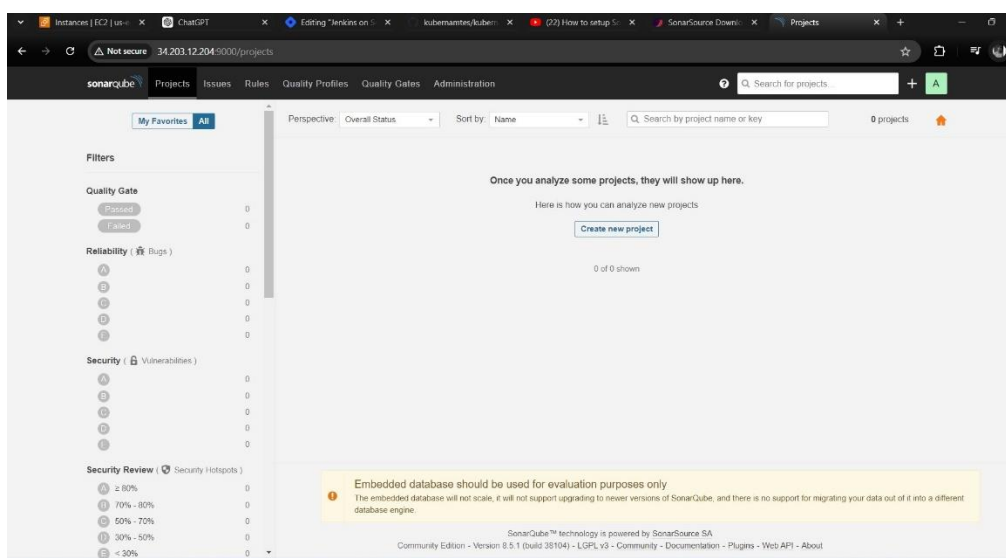
Divya vasasatputesatpute

Access your SonarQube:

URL: `http:// EC2-VM-Public-IP :9000/`

Note: SonarQube runs on 9000 port by Default. Enable this port in security group as custom TCP

SonarQube Dashboard look like this:



Divya_satpute

@learnwithdivya.hashnode.dev

SONARQUBE

Divya vasasatputesatpute

SonarQube Reports

The screenshot displays the SonarQube web interface in a browser. The address bar shows the URL `23.23.2.151:9000/projects`. The interface includes a top navigation bar with tabs for Projects, Issues, Rules, Quality Profiles, Quality Gates, and Administration. A search bar is present on the right. The main content area shows a list of projects. Two projects are visible:

- 38_SB_REST_H2_DB_APP**: Status **Passed**. Last analysis: July 19, 2024 at 11:12 PM. Metrics: 0 Bugs, 0 Vulnerabilities, 0 Hotspots Reviewed, 0 Code Smells, 0.0% Coverage, 0.0% Duplications. 169 files (Java, XML).
- my-webapp Maven Webapp**: Status **Passed**. Last analysis: July 19, 2024 at 8:59 PM. Metrics: 0 Bugs, 0 Vulnerabilities, 0 Hotspots Reviewed, 0 Code Smells, 0.0% Coverage, 0.0% Duplications. 111 files (JSP, XML).

On the left, there are filters for Quality Gate (Passed: 2, Failed: 0), Reliability (0 Bugs), Security (0 Vulnerabilities), and Security Review (2 Security Hotspots). A yellow warning banner at the bottom states: "Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database."

Divya_satpute
@learnwithdivya.hashnode.dev

SONARQUBE

Divya vasasatputesatpute



Integrated Sonar Server With Java Maven app

Configure Sonar Properties Under <Properties/> tag in pom.xml

```
#<sonar.host.url>sonar-server-url</sonar.host.url>
```

```
#<sonar.login>username</sonar.login>
```

```
#<sonar.password>pwd</sonar.password>
```

#paste your value here:

```
<properties>
```

```
<sonar.host.url>http://23.23.2.151:9000</sonar.host.u  
rl>
```

```
<sonar.login>admin</sonar.login>
```

```
<sonar.password>admin</sonar.password>
```

```
</properties>
```

SONARQUBE

Divya vasasatputesatpute

Go to Project pom.xml file location and execute below goal in command Prompt

```
$mvn sonar:sonar
```

After Build Success go to Sonar Dashboard and verify that

Note: Instead of Username And Password We can Configure Token Also

Because it is not Recommended way to configure login id and password

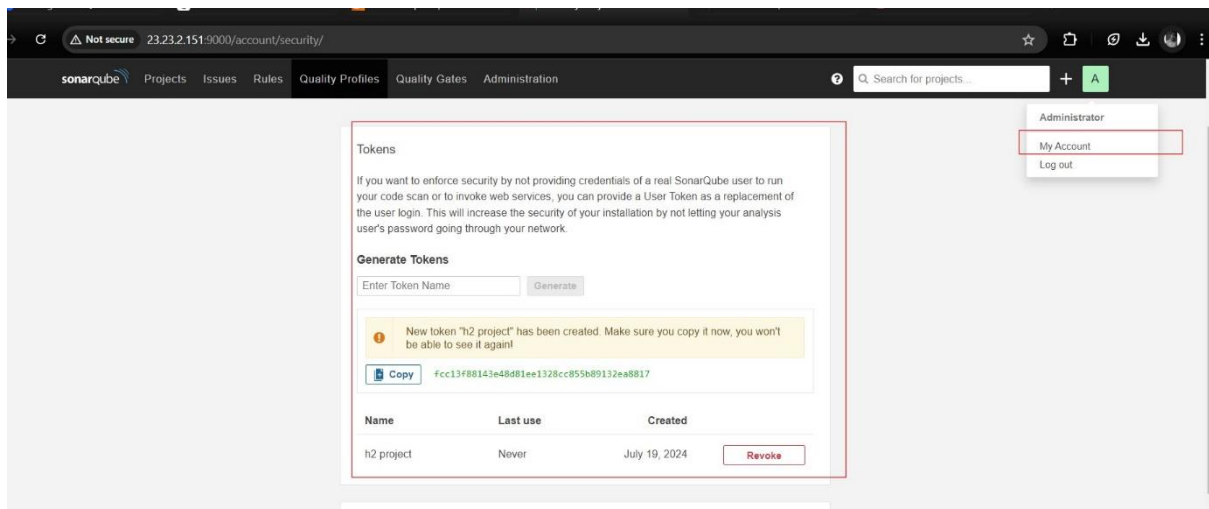
for Security Purpose you can use token instead of password

SONARQUBE

Divya vasasatputesatpute

Steps configure Token

Go to Sonar --->Login ---> Click On profile ----> My Account ---> Security ----->Generate Token



Copy the Token and Configure that in pom.xml like below

```
#<sonar.host.url>sonar-server-url</sonar.host.url>
```

```
#<sonar.login>token</sonar.login>
```

#paste your value here:

```
<properties>
```

```
<sonar.host.url>http://23.23.2.151:9000</sonar.host.url>
```

```
<sonar.login>fcc13f88143e48d81ee1328cc855b89132ea8817</sonar.login>
```

```
</properties>
```

Divya_satpute

@learnwithdivya.hashnode.dev

SONARQUBE

Divya vasasatputesatpute

Then build the Project using "**mvn sonar:sonar**" goal
you got result like this :

```
[INFO] 20:59:31.225 2 source files to be analyzed
[INFO] 20:59:31.361 Sensor XML Sensor [xml] (done) | time=152ms
[INFO] 20:59:31.361 2/2 source files have been analyzed
[INFO] 20:59:31.366 ----- Run sensors on project
[INFO] 20:59:31.366 Sensor Zero Coverage Sensor
[INFO] 20:59:31.366 Sensor Zero Coverage Sensor (done) | time=0ms
[INFO] 20:59:31.366 SCM Publisher SCM provider for this project is: git
[INFO] 20:59:31.366 SCM Publisher 3 source files to be analyzed
[INFO] 20:59:31.491 SCM Publisher 2/3 source files have been analyzed | time=125ms
[WARNING] 20:59:31.491 Missing blame information for the following files:
[WARNING] 20:59:31.491 * pom.xml
[WARNING] 20:59:31.491 This may lead to missing/broken features in SonarQube
[INFO] 20:59:31.507 CPD Executor Calculating CPD for 1 file
[INFO] 20:59:31.507 CPD Executor CPD calculation finished (done) | time=0ms
[INFO] 20:59:31.585 Analysis report generated in 78ms, dir size=85 KB
[INFO] 20:59:31.725 Analysis report compressed in 140ms, zip size=14 KB
[INFO] 20:59:32.682 Analysis report uploaded in 957ms
[INFO] 20:59:32.692 ANALYSIS SUCCESSFUL, you can browse http://23.23.2.151:9000/dashboard?id=com.example%3Amy-webapp
[INFO] 20:59:32.692 Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
[INFO] 20:59:32.692 More about the report processing at http://23.23.2.151:9000/api/ce/task?id=AZDLnNm35ALQy0h0INf4
[INFO] 20:59:32.698 Analysis total time: 20.757 s
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 01:37 min
[INFO] Finished at: 2024-07-19T20:59:32+05:30
[INFO] -----
D:\divya\Maven-web-app>
```

And you can see your Dashboard and report will be generated

The screenshot displays the SonarQube web interface. The top navigation bar includes links for Projects, Issues, Rules, Quality Profiles, Quality Gates, and Administration. A search bar is present on the right. The main content area shows the project dashboard for 'my-webapp Maven Webapp', which is marked as 'Passed'. The dashboard includes a 'Last analysis' timestamp of 'July 19, 2024 at 8:59 PM'. Below this, there are several metrics: 0 Bugs, 0 Vulnerabilities, 0 Hotspots Reviewed, 0 Code Smells, and 0.0% Duplications. A '111 JSP, XML' count is also visible. The left sidebar contains filters for Quality Gate, Reliability, Security, and Security Review. A yellow warning banner at the bottom states: 'Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.'

Divya_satpute
@learnwithdivya.hashnode.dev

SONARQUBE

Divya vasasatputesatpute

Quality Profile in SonarQube

Quality profile means set of rules to perform code review

For each language SonarQube provided one Quality Profile

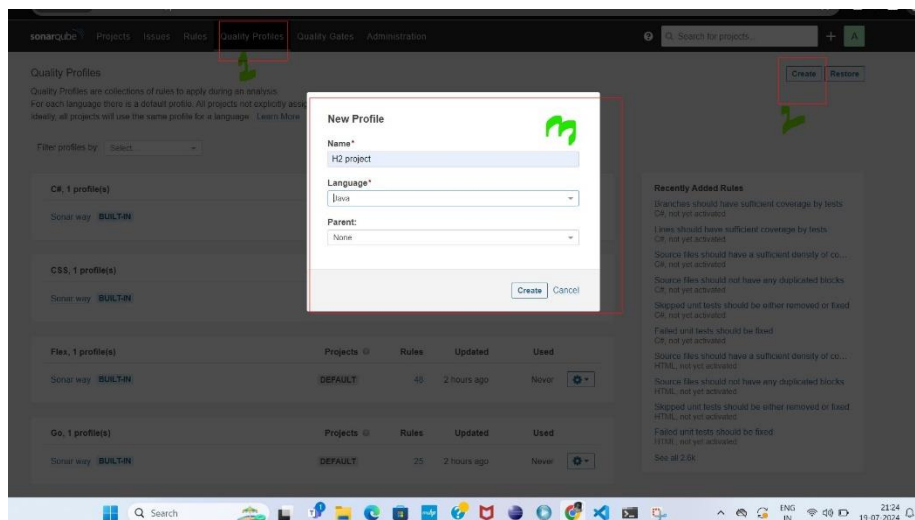
We can create our own quality profile based on Requirement

create one own Quality profile

\$ Name: SBI project

\$ language: java

\$ parent: none

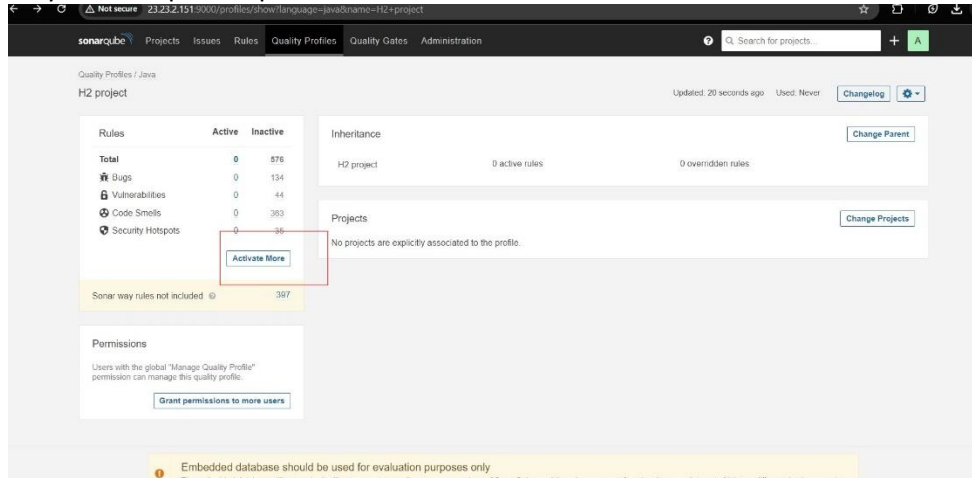


Divya_satpute

@learnwithdivya.hashnode.dev

SONARQUBE

Divya vasasatputesatpute



by clicking Activate more, you can add custom rules according to your requirement

Note: If we have any common ruleset for all projects then we can create one own quality profile and we can use that as parent quality profile for other profiles

we can also configure quality profile to specific project only

- click on project name
- go to administration
- click on quality profile
- select profile required

SONARQUBE

Divya vasasatputesatpute

```
INFO] 20:59:12.280 Execute project builders
INFO] 20:59:12.280 Execute project builders (done) | time=0ms
INFO] 20:59:12.297 Project key: com.example:my-webapp
INFO] 20:59:12.297 Base dir: D:\divya\Maven-web-app
INFO] 20:59:12.297 Working dir: D:\divya\Maven-web-app\target\sonar
INFO] 20:59:12.328 Load project settings for component key: 'com.example:my-webapp'
INFO] 20:59:12.987 Load quality profiles
INFO] 20:59:14.278 Load quality profiles (done) | time=1291ms
INFO] 20:59:14.294 Load active rules
INFO] 20:59:27.512 Load active rules (done) | time=13218ms
INFO] 20:59:27.560 Indexing files...
INFO] 20:59:27.560 Project configuration:
INFO] 20:59:27.766 3 files indexed
INFO] 20:59:27.770 0 files ignored because of scm ignore settings
INFO] 20:59:27.770 Quality profile for jsp: Sonar way
INFO] 20:59:27.770 Quality profile for xml: Sonar way
INFO] 20:59:27.770 ----- Run sensors on module my-webapp Maven Webapp
INFO] 20:59:27.860 Load metrics repository
INFO] 20:59:28.605 Load metrics repository (done) | time=745ms
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by net.sf.cglib.core.ReflectUtils$1 (file:/C:/Users/Rinku/.sonar/cache/52f5340dd05620cd16
plugin.jar) to method java.lang.ClassLoader.defineClass(java.lang.String,byte[],int,int,java.security.ProtectionDomain)
WARNING: Please consider reporting this to the maintainers of net.sf.cglib.core.ReflectUtils$1
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
INFO] 20:59:30.145 Sensor SonarCSS Rules [cssfamily]
INFO] 20:59:30.145 No CSS, PHP, HTML or VueJS files are found in the project. CSS analysis is skipped.
INFO] 20:59:30.160 Sensor SonarCSS Rules [cssfamily] (done) | time=15ms
INFO] 20:59:30.160 Sensor JaCoCo XML Report Importer [jacoco]
```

above image they took default Quality profile

Java, 2 profile(s)	Projects	Rules	Updated	Used
H2 project	0	0	13 minutes ago	Never
Sonar way BUILT-IN	DEFAULT	397	2 hours ago	
JavaScript, 2 profile(s)	Projects	Rules	Updated	
Sonar way BUILT-IN	DEFAULT	100	2 hours ago	
Sonar way Recommended BUILT-IN	0	148	2 hours ago	

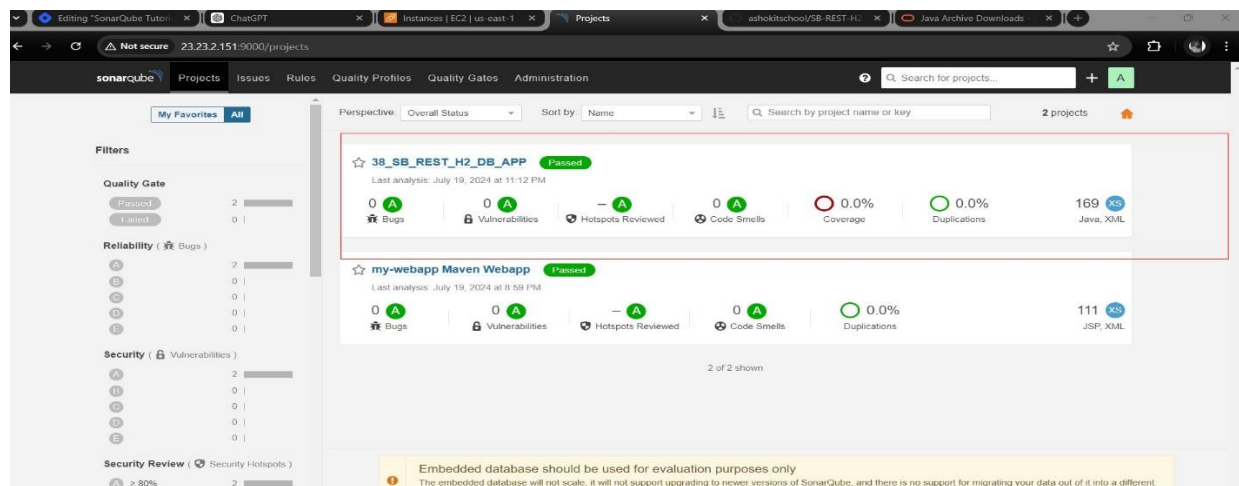
Note: you can see above image our custom quality profile created go to settings and set it default

Divya_satpute
@learnwithdivya.hashnode.dev

SONARQUBE

Divya vasasatputesatpute

```
NFO] 23:12:08.624 Load project settings for component key: 'in.ashokit:38_SB_REST_H2_DB_APP'
NFO] 23:12:09.032 Load quality profiles
NFO] 23:12:09.585 Load quality profiles (done) | time=553ms
NFO] 23:12:09.616 Load active rules
NFO] 23:12:14.520 Load active rules (done) | time=4904ms
NFO] 23:12:14.567 Indexing files...
NFO] 23:12:14.567 Project configuration:
NFO] 23:12:14.772 7 files indexed
NFO] 23:12:14.772 0 files ignored because of scm ignore settings
NFO] 23:12:14.772 Quality profile for java: H2 project
NFO] 23:12:14.772 Quality profile for xml: Sonar way
NFO] 23:12:14.772 ----- Run sensors on module 38_SB_REST_H2_DB_APP
NFO] 23:12:14.867 Load metrics repository
NFO] 23:12:15.102 Load metrics repository (done) | time=235ms
RNING: An illegal reflective access operation has occurred
RNING: Illegal reflective access by net.sf.cglib.core.ReflectUtils$1 (file:/C:/Users/Rinku/.sonar/cache/52f5340
ugin.jar) to method java.lang.ClassLoader.defineClass(java.lang.String,byte[],int,int,java.security.ProtectionD
RNING: Please consider reporting this to the maintainers of net.sf.cglib.core.ReflectUtils$1
RNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
```



Note : The [38_SB_REST_H2_DB_APP project create code review using Token](#) and custom **Quality Profile**

The [my-webapp Maven Webapp project create code review](#) using **login id** and **password** and **default Quality profile**

Quality Gate

Quality Gate represent set of metric to identify project quality is passed or failed

Every Project Quality Gate Should be Passed

Divya_satpute

@learnwithdivya.hashnode.dev

SONARQUBE

Divya vasasatputesatpute

In Sonar we have default Quality Gate

If required, we can create our own Quality Gate also

Note: If the project Quality Gate is **failed** then we should not Accept that code for deployment

Thanks a lot! 🎉 Celebrating achievements, big or small, is important.