

=====

Sonar Qube

=====

- => It is used for code quality check.
- => Using sonarqube we can perform code review (static analysis).
- => It will identify developers mistakes in code.
- => SonarQube developed using by java language.
- => It supports for 30+ programming languages code review.
- => SonarQube available in 2 flavours

- 1) Enterprise Edition (licensed)
- 2) Communit Edition (free)

=====

Sonar Issues

=====

- 1) Bugs (Ex: NPE)
- 2) Vulnerabilities (Ex: security issues, data leak)
- 3) Code smells (unused imports, unused method params etc..)
- 4) Duplicate Code Blocks
- 5) Code Coverage (related to unit testing)

=====

Sonar Server Setup

=====

=> We can setup Sonar Server in 3 ways...

Approach-1) Take one Linux vm in aws cloud and install sonar server

Approach-2) Setup Sonar Server as docker container

URL : <https://github.com/ashokitschool/DevOps-Documents/blob/main/06-Sonar-Setup-Docker.md>

Approach-3) Sonar Cloud

=====

Integrate Sonar server with Java Maven App

=====

-> Configure Sonar Properties under <properties/> tag in "pom.xml"

```
<properties>
    <sonar.host.url>http://13.201.30.174:9000/</sonar.host.url>
    <sonar.login>admin</sonar.login>
    <sonar.password>abc@123</sonar.password>
</properties>
```

-> Right click on project -> Run As -> Maven Build -> Execute below goal

```
sonar:sonar
```

-> After build success, go to sonar dashboard and verify the results.

Instead of username and pwd we can configure sonar token in pom.xml

```
=====
Working with Sonar Token
=====
```

-> Go to Sonar Server Dashboard -> Click on profile -> My Account -> Security -> Generate Token

-> Copy the token and configure that token in pom.xml file like below

```
<properties>
    <sonar.host.url>http://3.111.219.207:9000/</sonar.host.url>
    <sonar.login>sqa_fabdab18ac24e342de9eec5bc6f94d2c24223370</sonar.login>
</properties>
```

-> Right click on project -> Run As -> Maven Build -> Execute below goal

```
sonar:sonar
```

If we re-start EC2 vm, then to start sonar server execute below commands

```
sudo service docker start
docker start sonarqube
```

Note: When we re-start EC2 vm then public ip gets changed. (update sonar server public ip in pom.xml)

```
=====
Lessons learnt
=====
```

- 1) Don't use StringBuffer, instead of that use StringBuilder class which is not thread safe.
- 2) Don't create Random class object inside method, create at class level and re-use that.
- 3) Declare private constructor for AppConstants class
- 4) Remove unwanted curly braces in lambda expressions
- 5) Remove commented code
- 6) Remove unused import statements and unused method parameters
- 7) Declare constants for repeated String literals
- 8) Remove Duplicate code
- 9) Handle NullPointerExceptions properly

```
=====
How code review will happen in realtime ?
=====
```

=> JENKINS CI CD pipelines will be used to perform project build and deployment.

=> As part of project build process we need to perform code review.

=> Sonar Server will be integrated with Jenkins pipeline to automate code review process.

Note : SonarQube server setup and jenkins pipeline creation will be taken care by DevOps team in project.

=> As a developer we will run jenkins pipeline and we will check sonar dashboard for code review report.

=> Based on sonar report, we have to fix sonar issues in code.

=====

For new joinees below tasks will be assigned in first 3 months of time

- 1) Sonar Fixes
- 2) Implement Unit test cases using Junits
- 3) Improve code coverage of the project (min 80%)
- 4) Bug fixing