**Sonarqube**

1. How to determine if Quality Software were developed to the client??
   1. Software testing:
      1. unit testing: is written and performed by developers. Testing individual component of the code after Maven build code [mvn test or mvn package/install]
      2. Functional: testing the functionalities of the app
      3. regression testing : testing the new code with the old code

v10 [25k lines of code] | v11 [added/modified 1klines of code]

* + 1. performance
    2. penetration : security

[username/password]

* 1. Code Quality Analysis:
     1. git/GitHub : the code committed to github includes

src Codes -- 10,000 lines of code

JUnit Tests Cases -- -- 3,000 unit test cases

buildScript-pom.xml

javaProjects

1. sonarqube:

how many test cases should be there? 10, 000

how many unit test case are written? 3,000

code coverage = 3000/1000\*100% = 30% = poor

1. maven:
   1. mvn package
   2. validation
   3. compilation
   4. Testing JUnit
   5. packaging[jav/war/ear]
2. Sonarqube Bench-marking: comparing actual results against the benchmark/standards
   1. How many unit test cases were ran? How many test cases should have being ran?
   2. Is our code easily readable?

----> code smells

Is our code portable [environment independent]?

-- if not code smells / hard coding

* 1. Are there vulnerabilities in the code ?

---> if they are then --- password -- any 4 characters

1. Code Quality expectations:
   1. code coverage = 90%
   2. Duplicated code = 5% {Functions}
   3. Coding standards =
   4. Unit tests =
   5. Complex code =
   6. Comments =
   7. Potential Bugs = 0
   8. Architecture & Design
2. To check if sonar is working = sh /opt/sonarqube/bin/linux-x86-64/sonar.sh status | curl -v hostname:9000
3. Sonarqube directory structure :
   1. bin : sonar.sh start|stop|restart ( switch case bash shell script )
   2. COPYING
   3. elasticsearch
   4. lib
   5. Temp
   6. conf = sonar.properties (changing port, configuration)
   7. data
   8. extensions
   9. logs (sonar.log)
   10. web
4. To execute sonarqube report =mvn sonar:sonar

First sonar = maven sonarqube plugin

Second sonar = goal

1. For maven and sonarqube to communicate, we need to include in maven login credentials.
   1. Vim pom.xml in maven
   2. To the properties tag and insert/correct sonarqube server details (private ip4, sonar user login and password)

<properties>

<jdk.version>1.8</jdk.version>

<spring.version>5.1.2.RELEASE</spring.version>

<junit.version>4.11</junit.version>

<log4j.version>1.2.17</log4j.version>

<sonar.host.url>http:172.31.87.34:9000/</sonar.host.url>

<sonar.login>doten</sonar.login>

<sonar.password>admin123</sonar.password>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>

</properties>

1. Sonarqube comes with scanner
   1. compute engine
   2. elastic search
   3. web server
   4. Database
2. To configure sonar url
   1. To create user = administration --- security
   2. Pom.xml in maven
3. The default code quality is sonar way but we can define or set parameters for code quality in code profile and quality gates. We can set for each code language eg. Java, c#, c++, jsp, xml etc.
4. Instead of password, we can use token which is more secured.
   1. Token can be gotten from sonar url - demonstrator (A) -security - user- generate token
   2. go to pom.xml in maven, replace login with token and delete/ comment password as show below

<properties>

<jdk.version>1.8</jdk.version>

<spring.version>5.1.2.RELEASE</spring.version>

<junit.version>4.11</junit.version>

<log4j.version>1.2.17</log4j.version>

<sonar.host.url>http:172.31.87.34:9000/</sonar.host.url>

<sonar.login>ac84c2758d6f7c8264ef8aa3236dfb8ecfe2c545</sonar.login>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>

</properties>