SONAR

Check your code quality

ABSTRACT

This document will guide you in step by step approach from setting up of your sonar server to check the code quality.

Jyoti Prakash Behera
M1027358

Version 1.0

Note:

If you choose serial number 4 over serial number 3 then you can skip the 3rd and directly jump to step 4.

This document is done on initial knowledge sharing by **Shashi Kumar (Campus mind Feb 2018)**.

If you find some other way then please share with me so that I can update this document.

STEPS TO USE SONARQUBE SERVER:

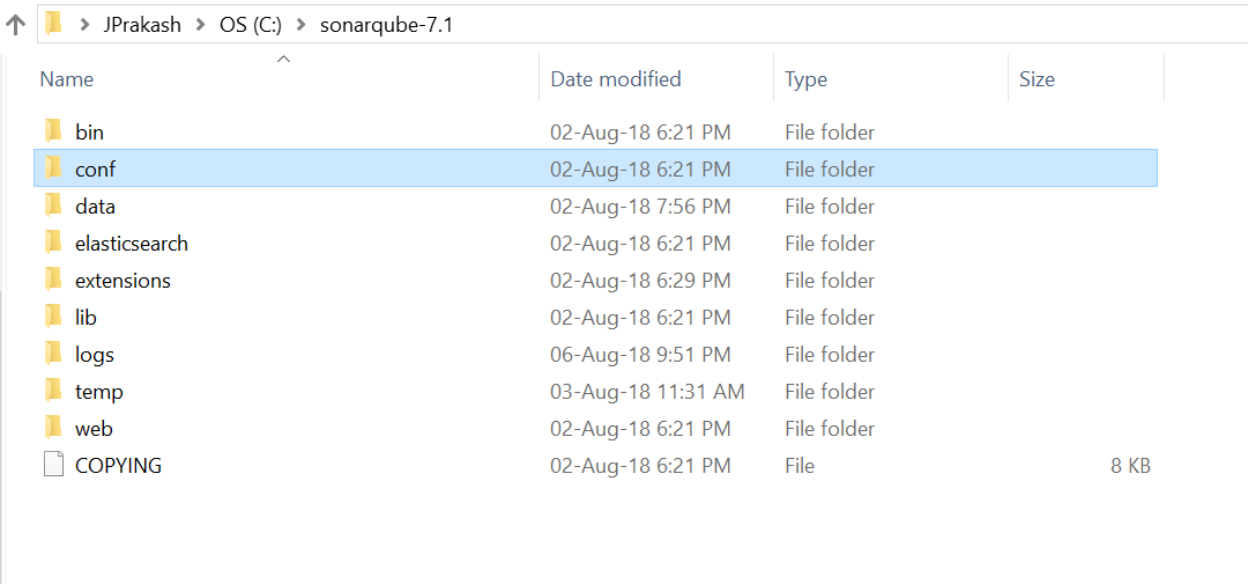
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Step1:

Get sonarqube-x.x zip from whitelist and extract it to your local.

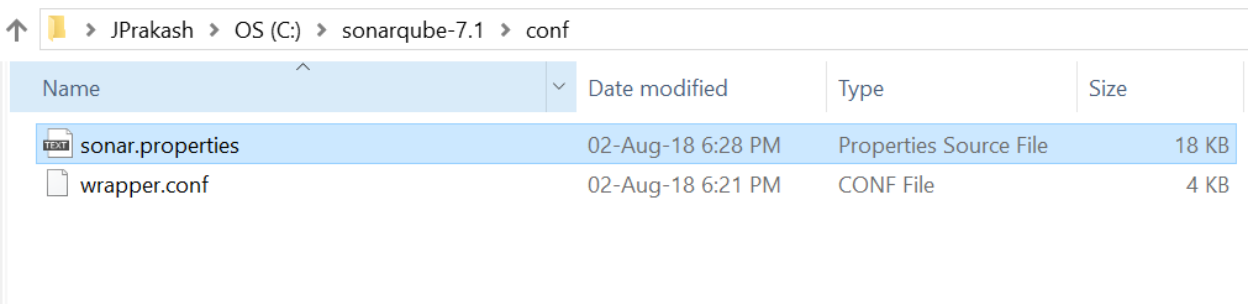
Step 2:

Go inside “conf” folder.



Step 3:

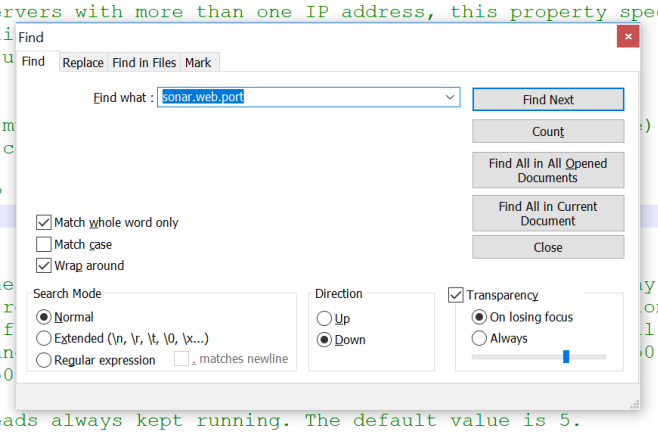
Select and open “sonar.properties” file with notepad++


~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Step 4:

Search for the string “sonar.web.port” in the document. This line would be somewhere around line number 111. ☺

```
98
99  # Same as previous property, but allows to not repeat all other settings like -Xmx
100 #sonar.web.javaAdditionalOpts=
101
102 # Binding IP address. For servers with more than one IP address, this property specifies which
103 # address will be used for li
104 # By default, ports will be u
105 #sonar.web.host=0.0.0.0
106
107 # Web context. When set, it m                                                    ).
108 # The default value is root c
109 #sonar.web.context=
110 # TCP port for incoming HTTP
111 #sonar.web.port=9000
112
113
114 # The maximum number of conne                                                  y given time.
115 # When this number has been r                                                  ons until
116 # the number of connections f                                                  l accept connectic
117 # based on the sonar.web.conn                                                  0.
118 #sonar.web.http.maxThreads=50
119
120 # The minimum number of threads always kept running. The default value is 5.
121 #sonar.web.http.minThreads=5
122
123 # The maximum queue length for incoming connection requests when all possible request processing
124 # threads are in use. Any requests received when the queue is full will be refused.
```



Uncomment that line by removing # from the starting of the line. By default the port assigned to sonar.web.port is 9000. This port is blocked by MindTree network. So change it to any available port.

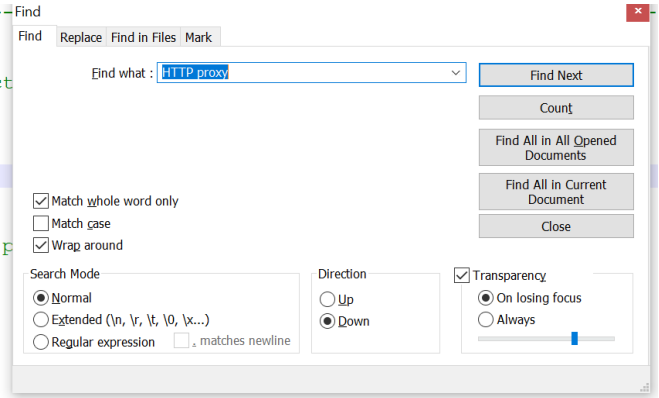
I have used port no 8081 for the purpose.

```
106
107 # Web context. When set, it must start with forward slash (for example /sonarqube).
108 # The default value is root context (empty value).
109 #sonar.web.context=
110 # TCP port for incoming HTTP connections. Default value is 9000.
111 sonar.web.port=8081
112
113
114 # The maximum number of connections that the server will accept and process at any given time.
```

Step 5:

Search for the text “HTTP proxy” in the document. After searching for “HTTP proxy” we can find “HTTPS proxy” just below “HTTP proxy”.

```
#--------------------------------------------------
# UPDATE CENTER

# Update Center requires an internet connect
# It is enabled by default.
#sonar.updatecenter.activate=true

# HTTP proxy (default none)
#http.proxyHost=
#http.proxyPort=
# HTTPS proxy (defaults are values of http.p
#https.proxyHost=
#https.proxyPort=

# NT domain name if NTLM proxy is used
#http.auth.ntlm.domain=

# SOCKS proxy (default none)
#socksProxyHost=
#socksProxyPort=
```



Step 6:

Uncomment and assign “172.17.2.218” to both “http.proxyHost” and “https.proxyHost”.

Uncomment and assign “8085” to both “http.proxyPort” and “https.proxyPort”.

```
#--------------------------------------------------
# UPDATE CENTER

# Update Center requires an internet connection to request https://update.sonarsource.org
# It is enabled by default.
#sonar.updatecenter.activate=true

# HTTP proxy (default none)
http.proxyHost=172.17.2.218
http.proxyPort=8085
# HTTPS proxy (defaults are values of http.proxyHost and http.proxyPort)
https.proxyHost=172.17.2.218
https.proxyPort=8085

# NT domain name if NTLM proxy is used
#http.auth.ntlm.domain=

# SOCKS proxy (default none)
#socksProxyHost=
#socksProxyPort=
```

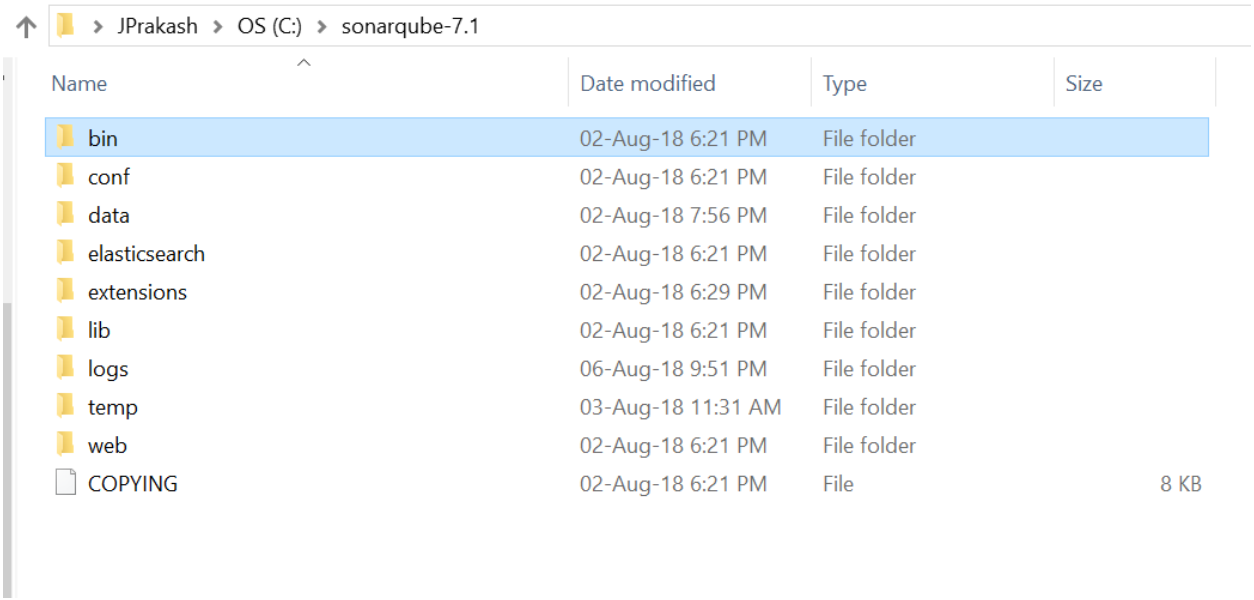
Do not get confused “HTTP proxy” with “SOCKs proxy”. Remember, we need to have only HTTP proxy for sonar server to work. ☺

Step 7:

Save and close the file.

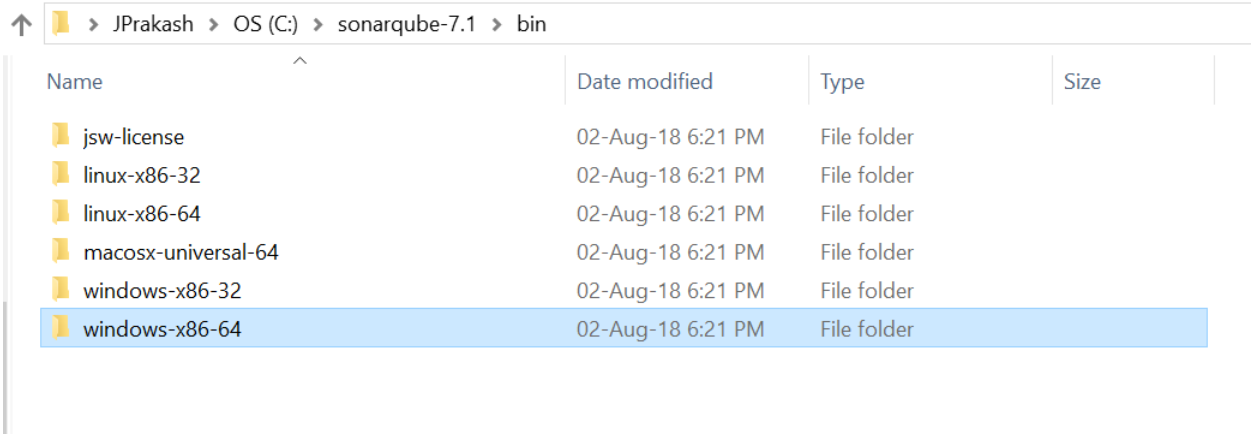
Step 8:

Go back and open “bin” folder from sonarqube root directory.



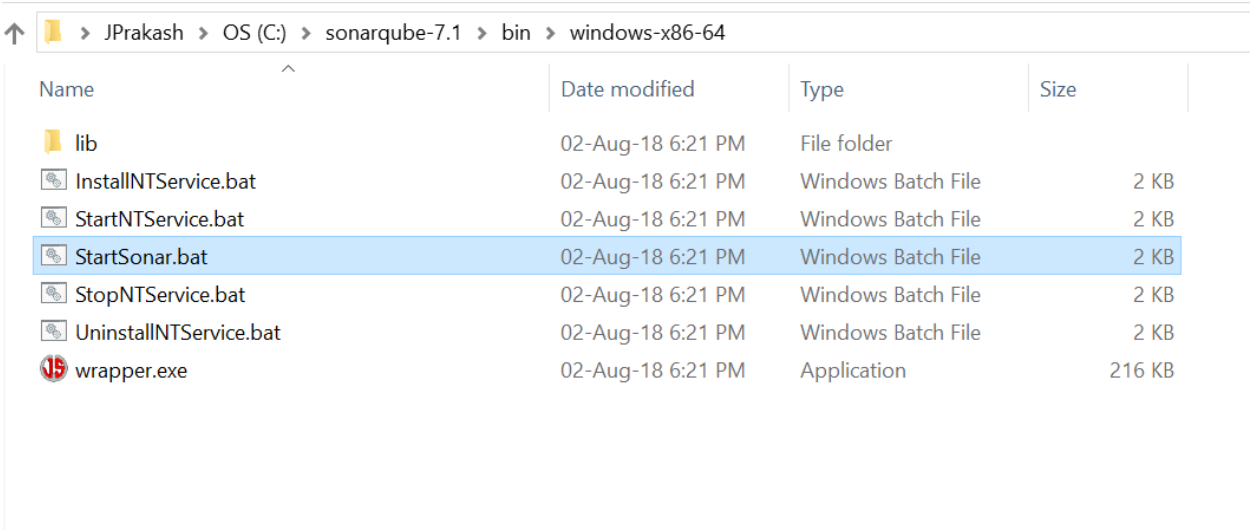
Step 9:

Select and open “windows-x86-64” directory (as I am using a 64 bit windows system. You should choose accordingly)

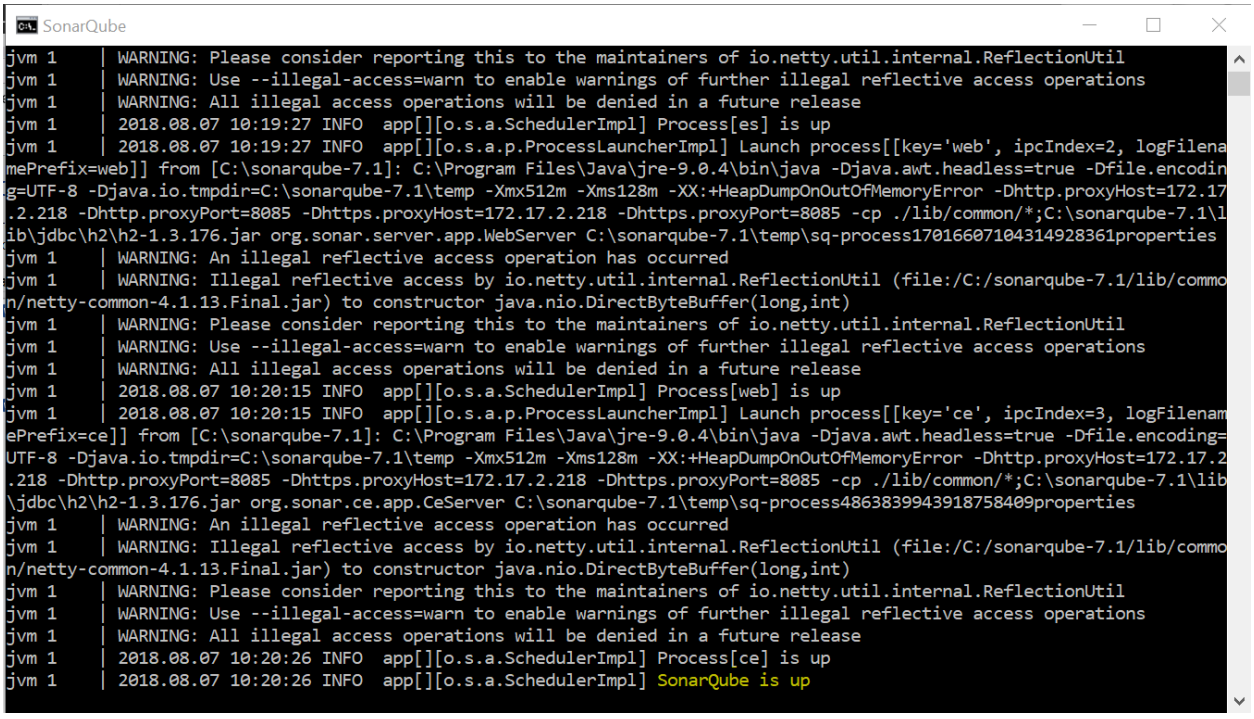


Step 10:

Select and double click on the batch file named “StartSonar.bat”.



This will open sonarqube command prompt window and if above steps are followed correctly we will be able to see a message at the end as “SonarQube is up”



NOTE:

If still you get any error message then it may be any reason from this 2:

1. SonarQube need JVM in the background. Makesure java is installed properly.
2. Port on which you are trying to run SonarQube is already in use. If this is the case free the port or close the services related to java which will close this port or restart your system which will close the port.☺
   Then follow this document again from step 1 to step 10.

STEPS TO USE SONARRUNNER:

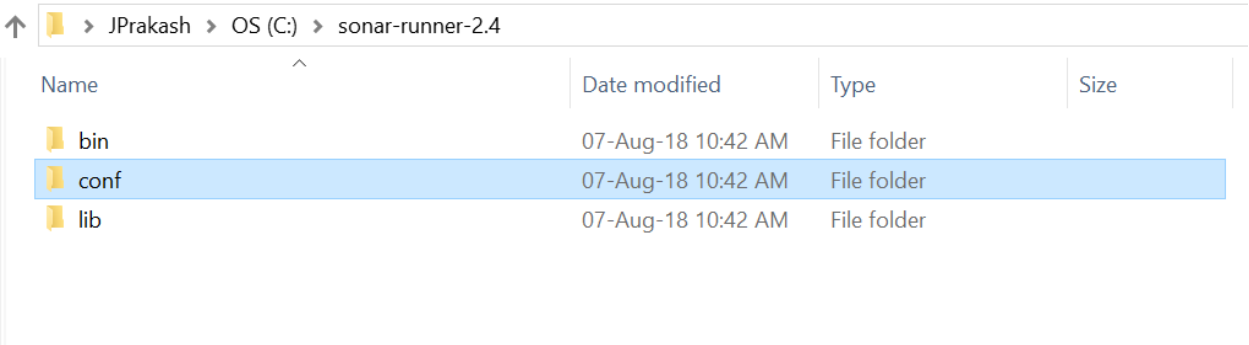
~~~~~~~~~~~~~~~~~~~~~~~~~~

Step1:

Get sonar-runner-x.x zip from whitelist and extract it to your local.

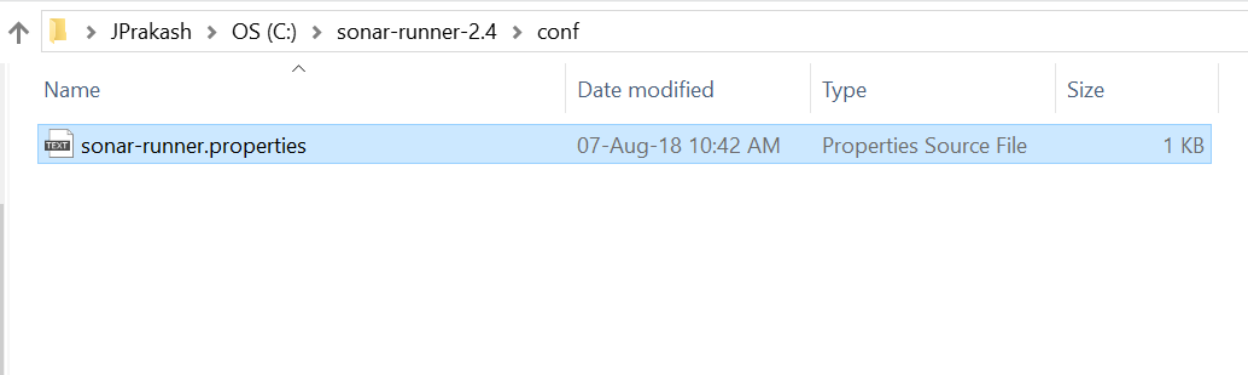
Step 2:

Go inside “conf” folder.



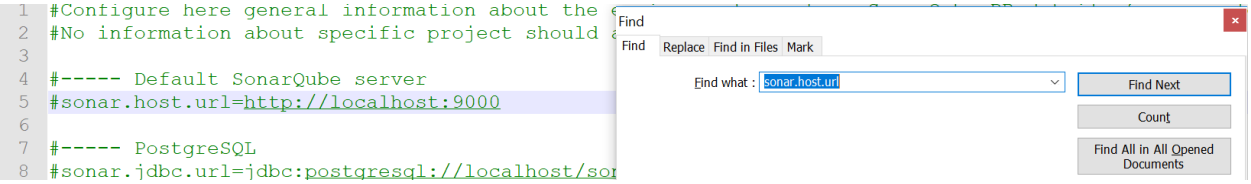
Step 3:

Select and open “sonar.properties” file with notepad++



Step 4:

Search for the string “sonar.host.url” in the document. This line would be somewhere around line number 5. ☺


~~~~~~~~~~~~~~~~~~~~~~~~~~

Uncomment that line by removing # from the starting of the line. By default the port assigned to sonar is 9000. This port is blocked by MindTree network. So change it to any available port.

I have used port no 8081 for the purpose.

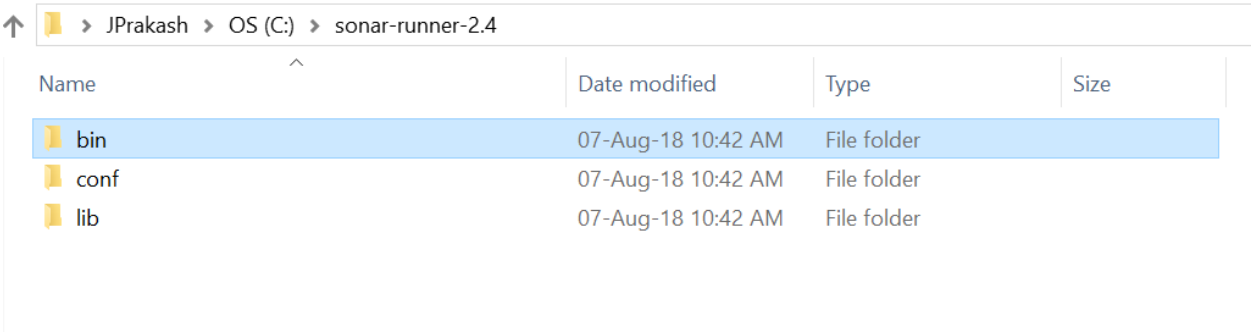
```
#Configure here general information about the environment, such as SonarQube DB details for example
#No information about specific project should appear here

#----- Default SonarQube server
sonar.host.url=http://localhost:8081

#----- PostgreSQL
#sonar.jdbc.url=jdbc:postgresql://localhost/sonar

#----- MySQL
#sonar.jdbc.url=jdbc:mysql://localhost:3306/sonar?useUnicode=true&amp;characterEncoding=utf8
```

Step 5:

Save and close the file.

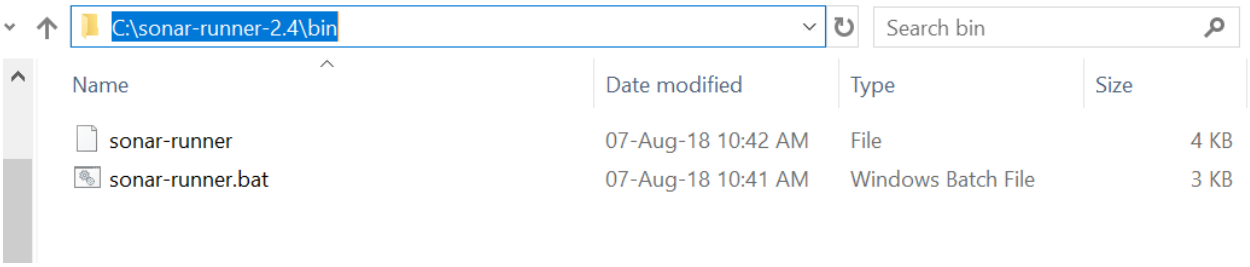
Step 6:

Go back and open “bin” folder from sonar-runner root directory.



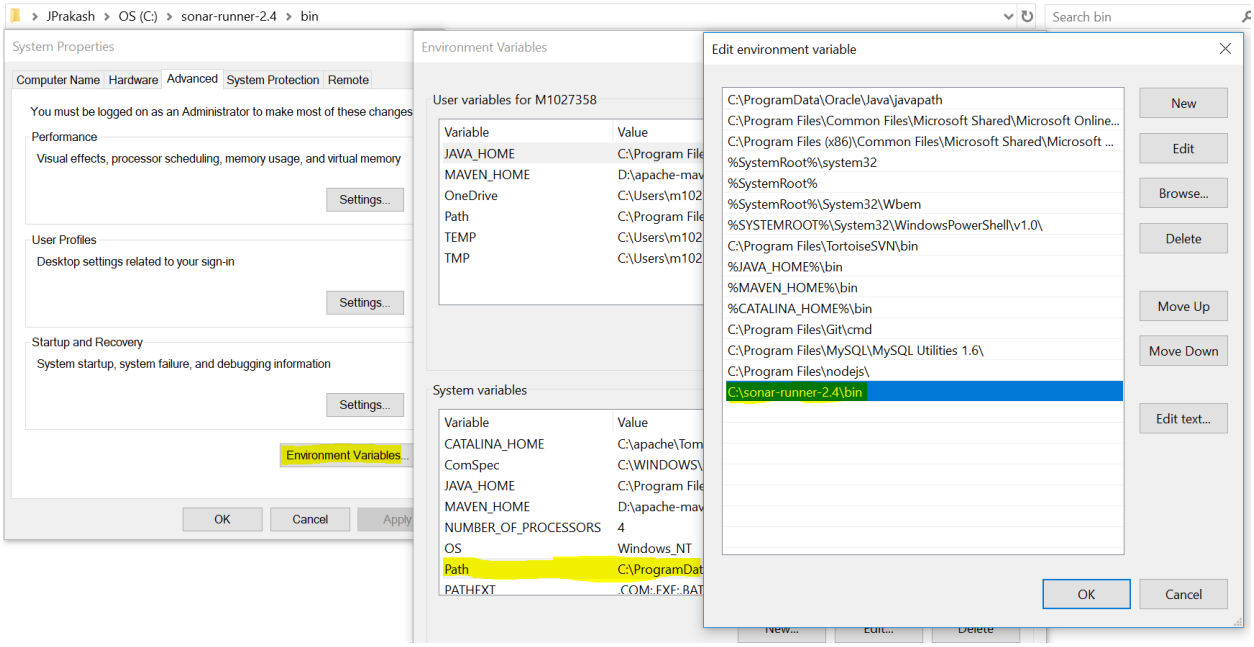
Step 7:

Go inside bin and copy the path which contains “sonar-runner” in its directory.



Step 8:

Set the copied path in the environment variables. If you are not able to set the path then jump to “Note” in this page.



Step 8:

Save and close the windows.

NOTE:

Now our sonar runner is ready to work.☺ If you were not able to set the environment variable then there is a work around. If you have set the path then only “sonar-runner” will work in the command prompt. If you have not set the path then you have to take help of full path.

i.e. “C:\sonar-runner-2.4\bin\sonar-runner”

But make sure that name of the folder\directory which contain sonar runner do not have gaps.

e.g: path “C:\Jyoti Prakash\sonar-runner-2.4\bin\ sonar-runner” will not work as a space is there in “Jyoti Prakash”

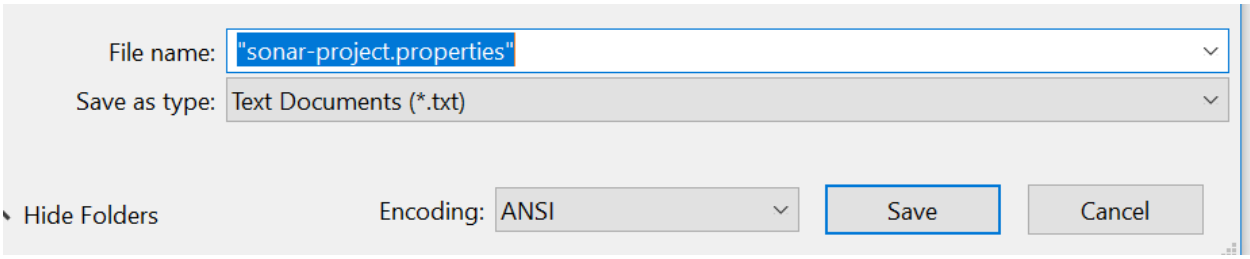
Whereas “C:\JyotiPrakash\sonar-runner-2.4\bin\ sonar-runner” will work fine in the future.

STEPS TO USE SONARRUNNER AND SONARQUBE TO CHECK A PROJECT:

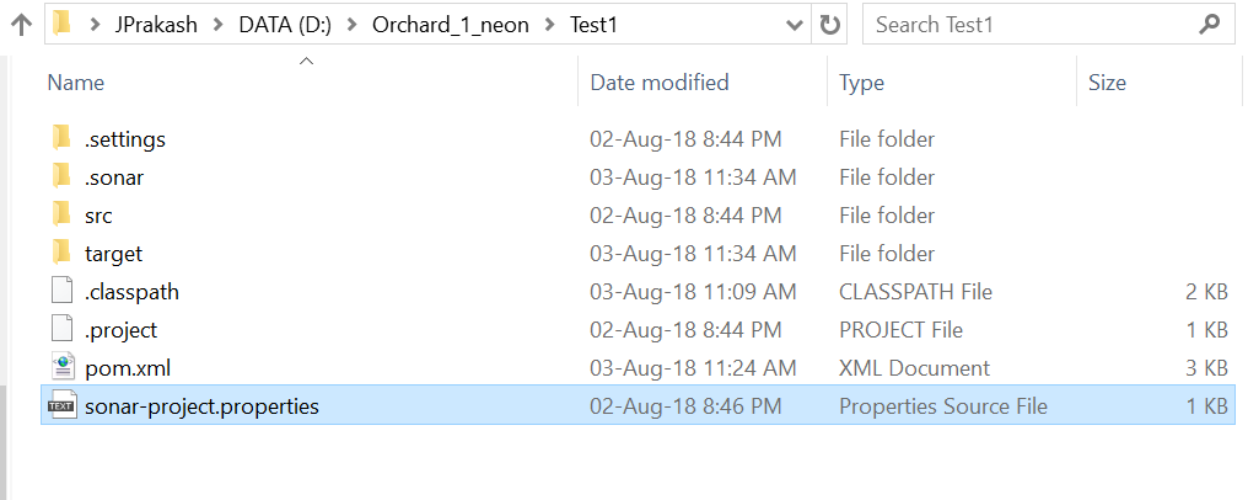
~~~~~~

Step 1:

Go to root directory of the project which you want to test and create “sonar-project.properties” properties file in the same location. (Create text file and while saving give file name in double quotes and with .properties extension)



Once “.txt” file got converted to “.properties” file then no need to have that text file in that location.



Open that file in notepad++ and add below lines in that document:

sonar.projectName=Test1

sonar.projectKey=Test1

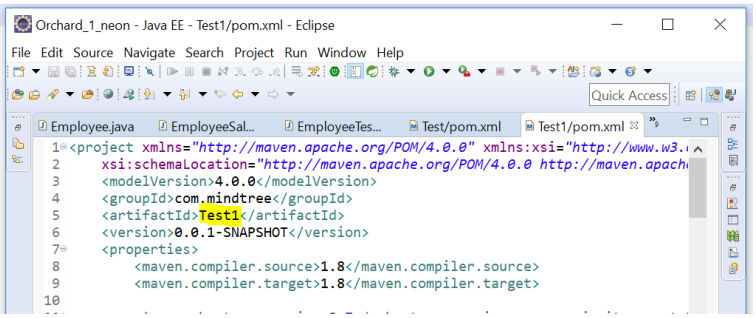
sonar.sources=.\\src\\main\\java

sonar.java.binaries=.\\target\\classes
~~~~~~

Step 2:

Open that file using notepad++ and set “sonar.projectName” and “sonar.projectKey” as your artefactId from your pom.xml of that project.

```
sonar.projectName=Test1
sonar.projectKey=Test1
sonar.sources=.\\src\\main\\java
sonar.java.binaries=.\\target\\classes
```



Step 3:

Have a “property” tag in your pom.xml as :

```
<properties>
	<maven.compiler.source>1.8</maven.compiler.source>
	<maven.compiler.target>1.8</maven.compiler.target>
</properties>
```

So that when we will update the project, it will not go back to other versions.

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.mindtree</groupId>
    <artifactId>Test1</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <properties>
        <maven.compiler.source>1.8</maven.compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
    </properties>
        <!-- <cobertura.version>2.7</cobertura.version> <sonar.junit.reportsPath>target/surefire-reports</sonar.junit.reportsPath>
            <sonar.cobertura.reportPath>target/site/cobertura/coverage.xml</sonar.cobertura.reportPath> -->

    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>4.12</version>
```

NOTE:

Make sure to paste properties tag before <dependencies>. And if you already have a properties tag then add only these lines in that properties tag:

```
<maven.compiler.source>1.8</maven.compiler.source>
<maven.compiler.target>1.8</maven.compiler.target>
```

Remember, a pom cannot have multiple <properties> tag. ☺

OR:

Have this <plugin> in your <build> -> <plugins>

```
<plugin>
	<artifactId>maven-compiler-plugin</artifactId>
	<version>3.5.1</version>
	<configuration>
		<source>1.8</source>
		<target>1.8</target>
	</configuration>
</plugin>
```

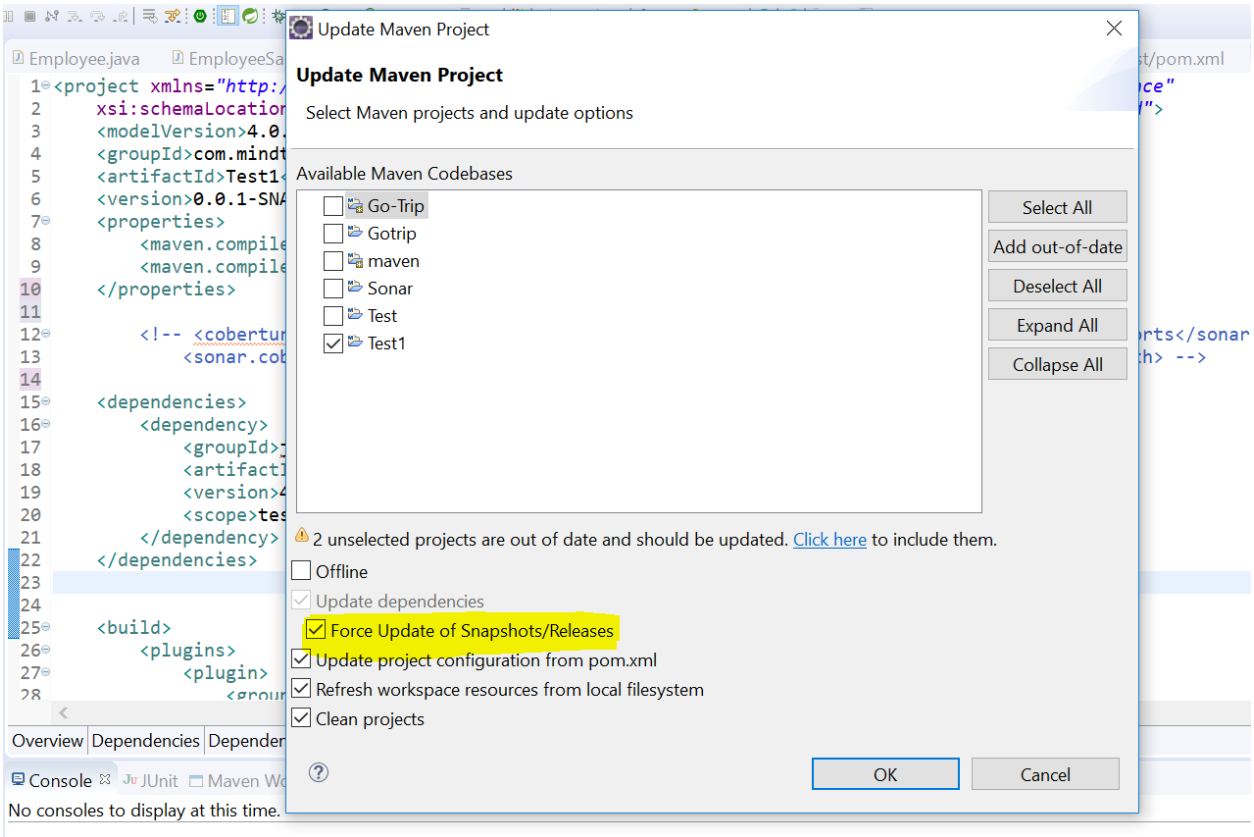
NOTE:

Make sure to paste <plugin> inside <plugins> which is inside<build>. <build> tag must be present after dependancies.

Remember, a pom cannot have multiple <build> tag. ☺

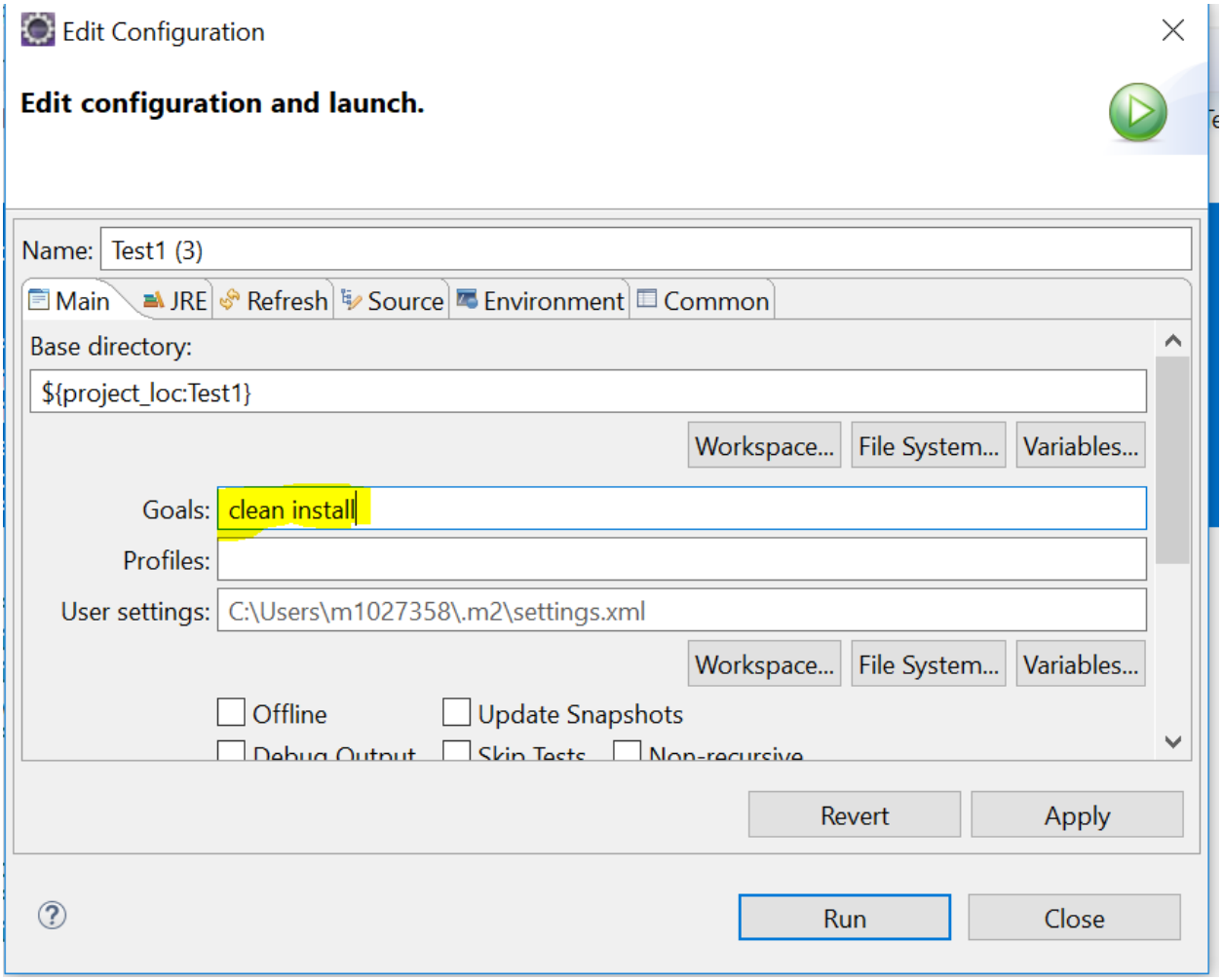
Step 4:

Update the project after checking “Force Update of Snapshots/Releases”. And wait till building of workspace is completed. (Which can be seen in bottom right corner of eclipse)



Step 5:

Do a “clean install” in your project.

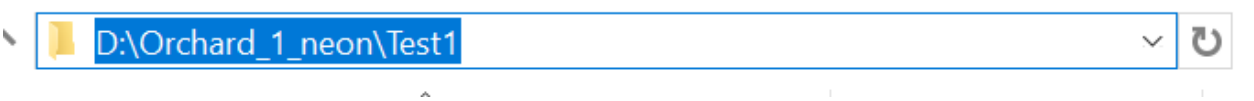


Step 6:

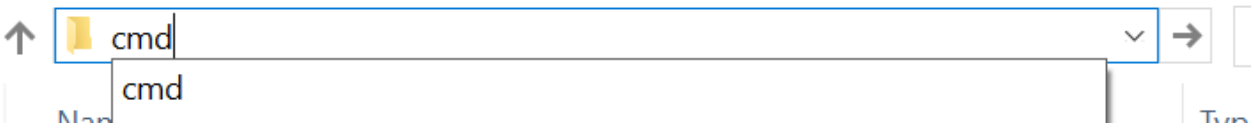
If you are getting “BUILD SUCCESS” then you are ready for next step. Else first fix your project.

Step 7:

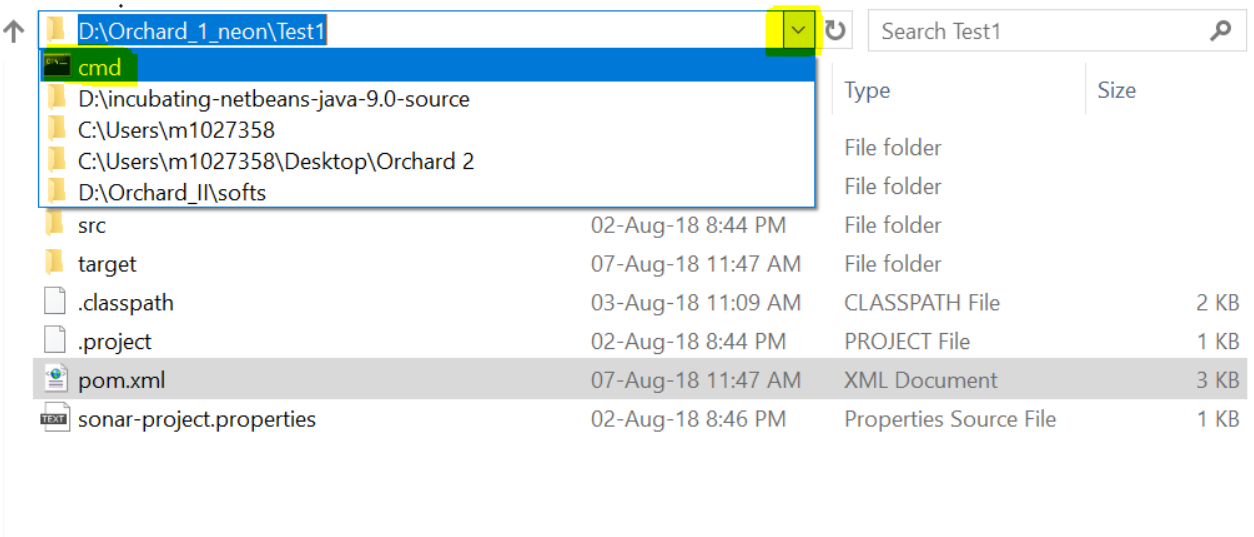
Then go to root directory of the project and open command prompt from there by tying “cmd” in that path directly:



->



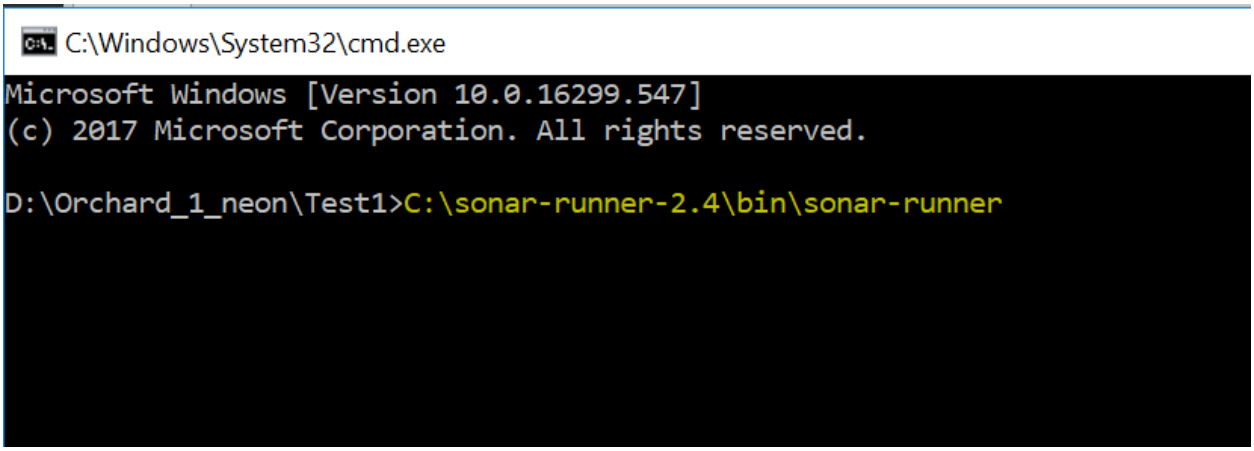
or



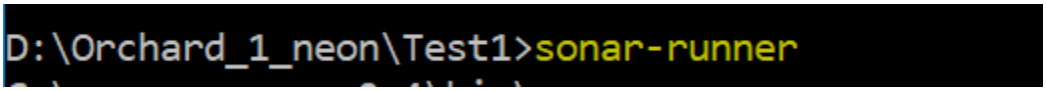
Or by going to that root directory through command prompt.

Step 8:

If you have not set sonar-runner path in environment variable then use full path of sonar runner.



If you have set the path of sonar-runner in the environment variable then only “sonar-runner” will do the same.



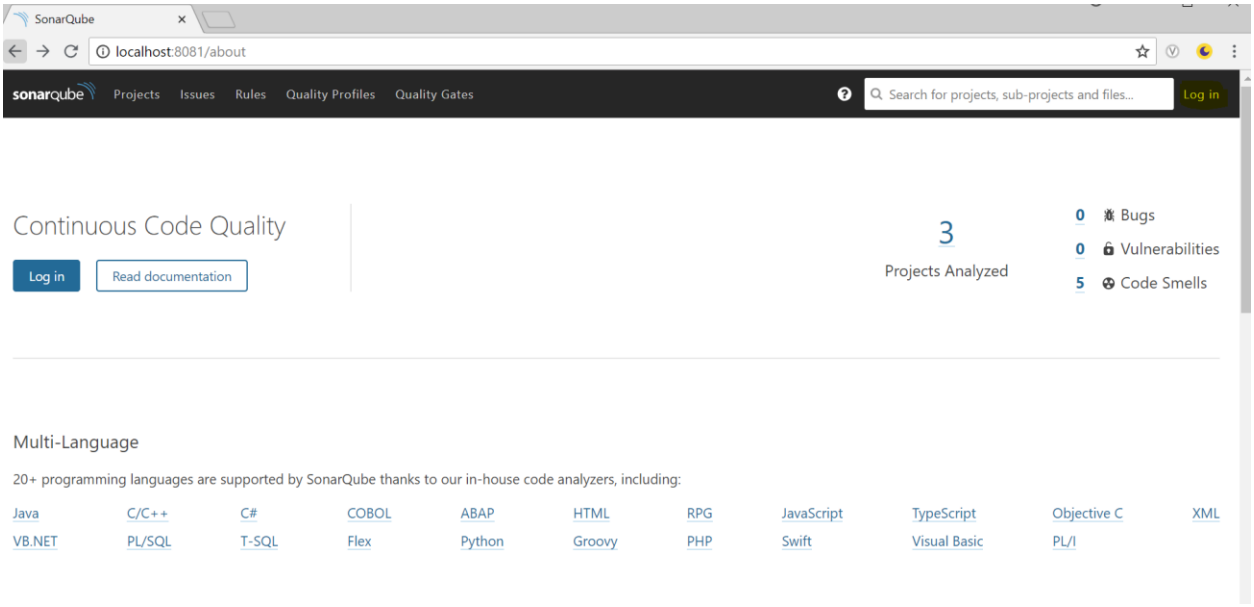
Step 9:

Wait for the message “EXECUTION SUCCESS” at the end. If it fails then you might have skipped some step☺

Step 10:

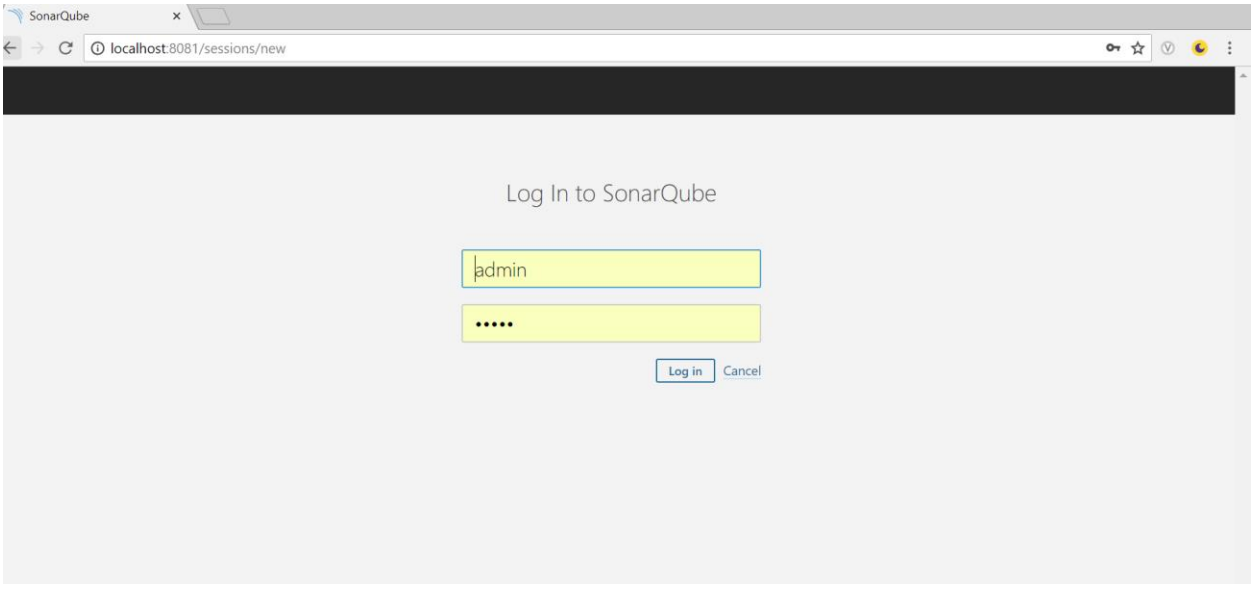
Put URL http://localhost:8081 in the browser. (I used 8081 coz I have set SonarQube server port as 8081)

You should be able to see below page



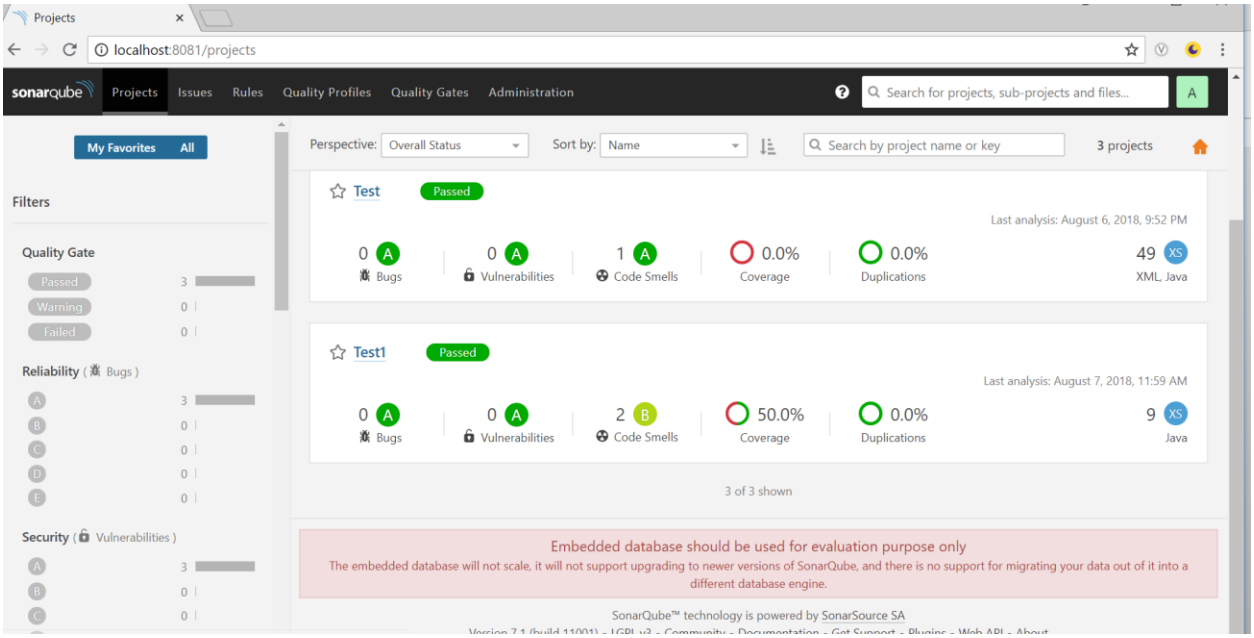
Step 11:

Click on “Log in” button on top right corner. And you should be able to see below page:



Step 12:

Enter username and password both as “admin” and click on “Log in”. You will be able to see your project in the sonar server.



STEPS TO USE SONARQUBE (ALONE) TO CHECK A PROJECT:

~~~~~~

Step 1:

Have a “property” tag in your pom.xml as :

```
<properties>
	<maven.compiler.source>1.8</maven.compiler.source>
	<maven.compiler.target>1.8</maven.compiler.target>
</properties>
```

So that when we will update the project, it will not go back to other versions.

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.mindtree</groupId>
    <artifactId>Test1</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <properties>
        <maven.compiler.source>1.8</maven.compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
    </properties>
        <!-- <cobertura.version>2.7</cobertura.version> <sonar.junit.reportsPath>target/surefire-reports</sonar.junit.reportsPath>
            <sonar.cobertura.reportPath>target/site/cobertura/coverage.xml</sonar.cobertura.reportPath> -->

    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>4.12</version>
```

NOTE:

Make sure to paste properties tag before <dependencies>. And if you already have a properties tag then add only these lines in that properties tag:

```
<maven.compiler.source>1.8</maven.compiler.source>
<maven.compiler.target>1.8</maven.compiler.target>
```

Remember, a pom cannot have multiple <properties> tag. ☺

OR:

Have this <plugin> in your <build> -> <plugins>

```
<plugin>
	<artifactId>maven-compiler-plugin</artifactId>
	<version>3.5.1</version>
	<configuration>
		<source>1.8</source>
		<target>1.8</target>
	</configuration>
</plugin>
```
~~~~~~

NOTE:

Make sure to paste <plugin> inside <plugins> which is inside<build>. <build> tag must be present after dependancies.

Remember, a pom cannot have multiple <build> tag. ☺

Step 2:

Add below line in properties tag

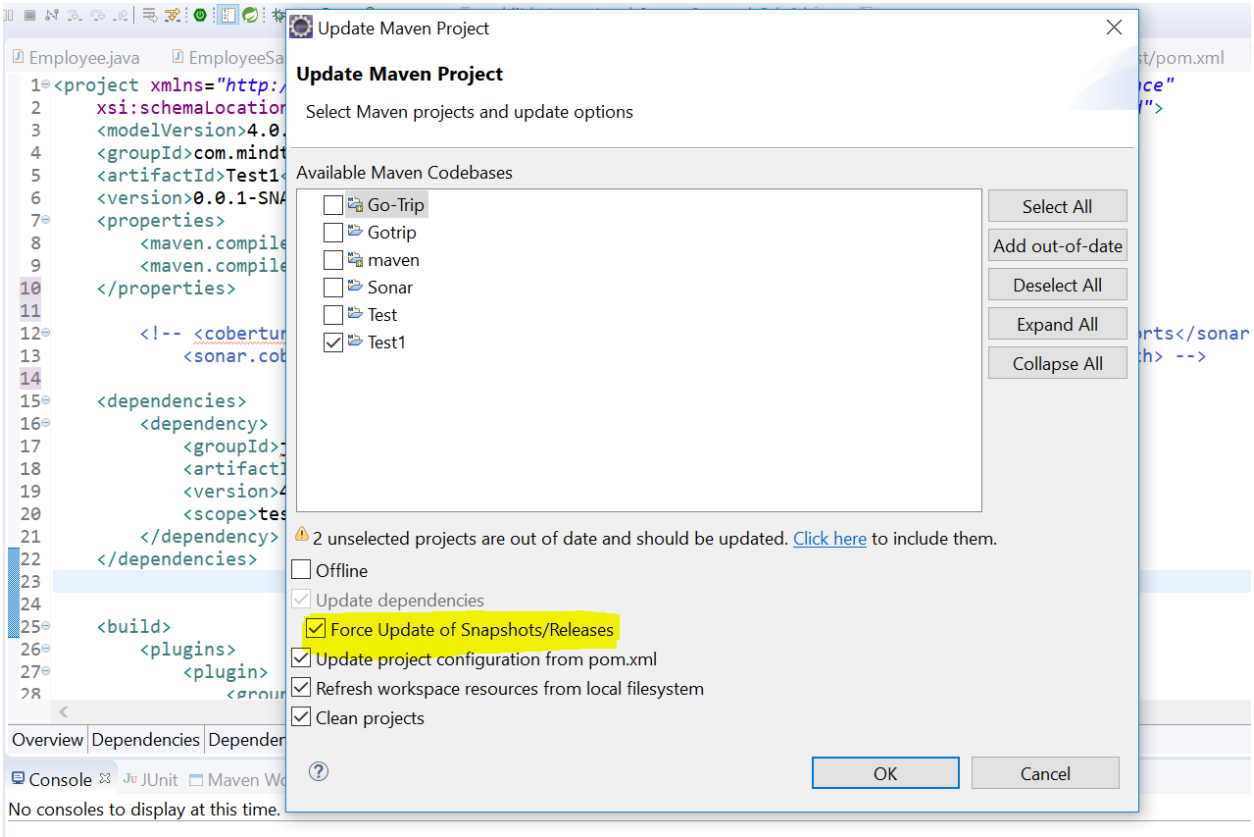
```
<sonar.host.url>http://localhost:8081</sonar.host.url>
```

And add below plugin in <plugins> present in <build> tag

```
<plugin>
	<groupId>org.sonarsource.scanner.maven</groupId>
	<artifactId>sonar-maven-plugin</artifactId>
	<version>3.2</version>
</plugin>
```

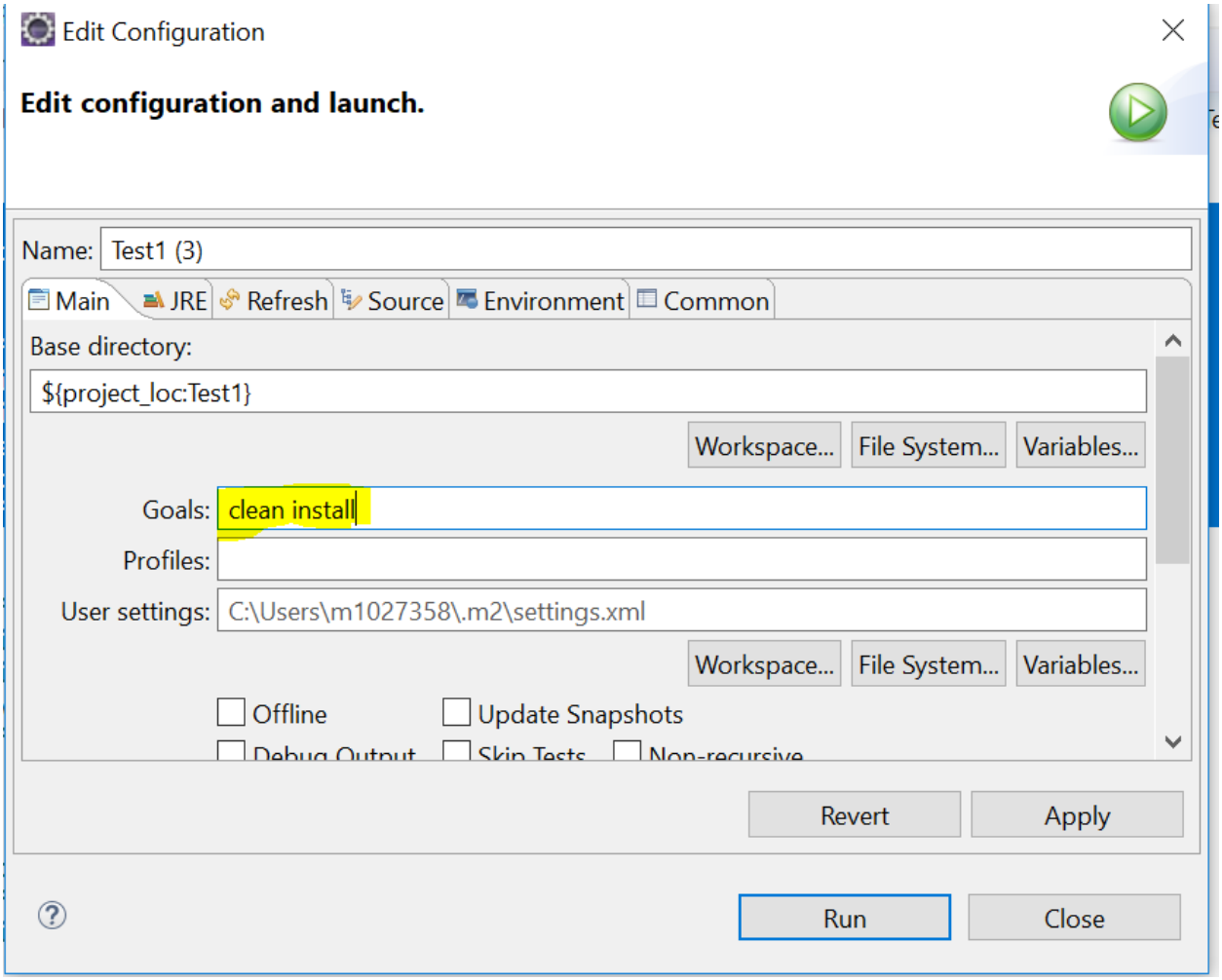
Step 3:

Update the project after checking “Force Update of Snapshots/Releases”. And wait till building of workspace is completed. (Which can be seen in bottom right corner of eclipse)



Step 4:

Do a “clean install” in your project.

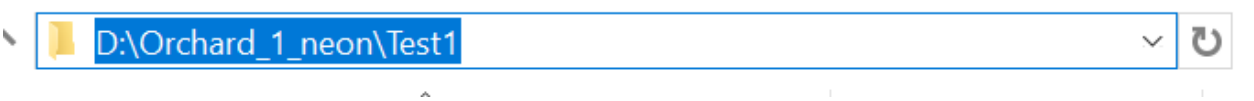


Step 5:

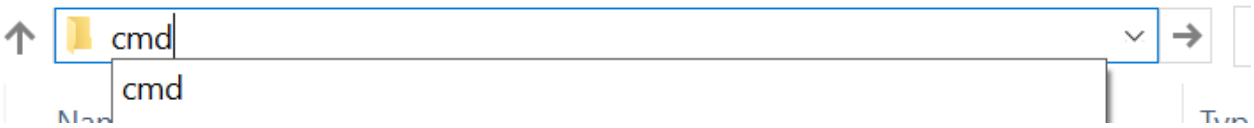
If you are getting “BUILD SUCCESS” then you are ready for next step. Else first fix your project.

Step 6:

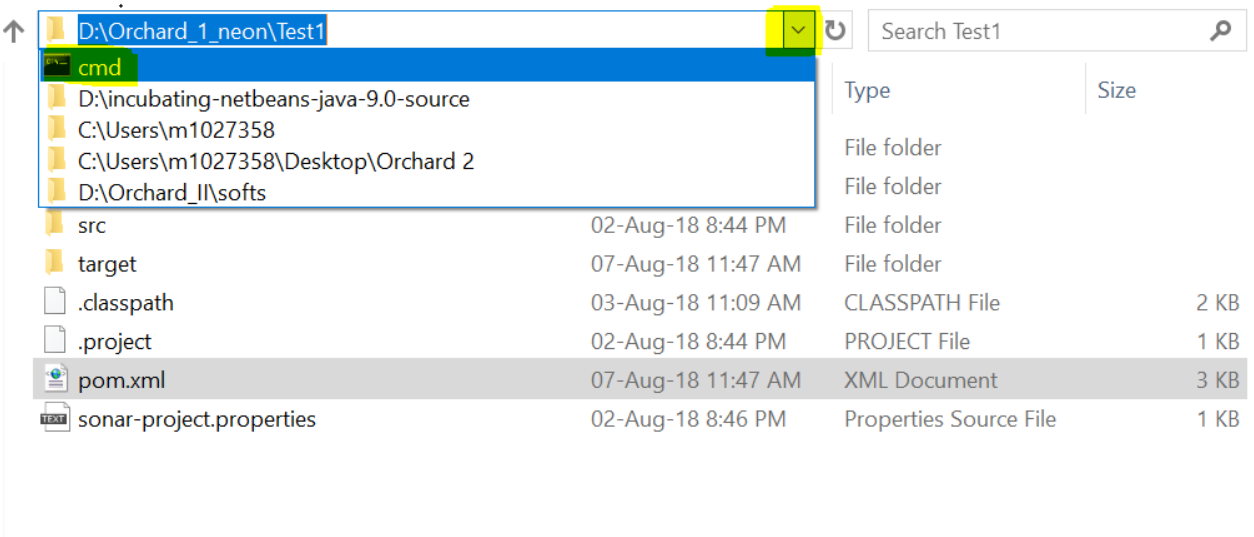
Then go to root directory of the project and open command prompt from there by tying “cmd” in that path directly:



->



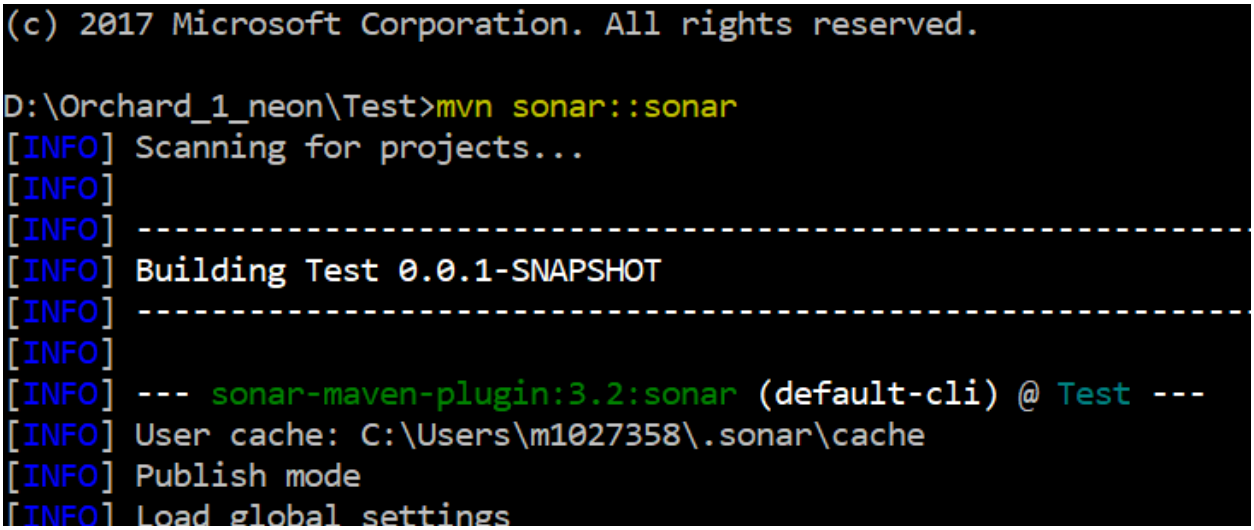
or



Or by going to that root directory through command prompt.

Step 7:

Insert maven command ”**mvn sonar::sonar**” in the command prompt.



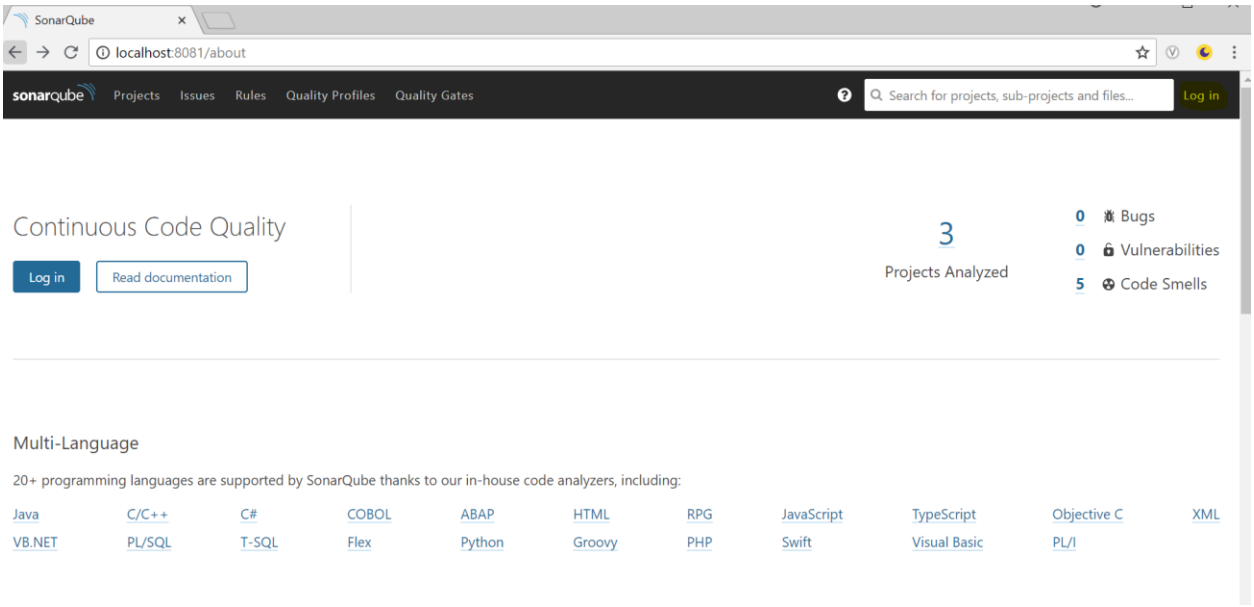
Step 8:

Wait for the message “BUILD SUCCESS” at the end. If it fails then you might have skipped some step☺

Step 9:

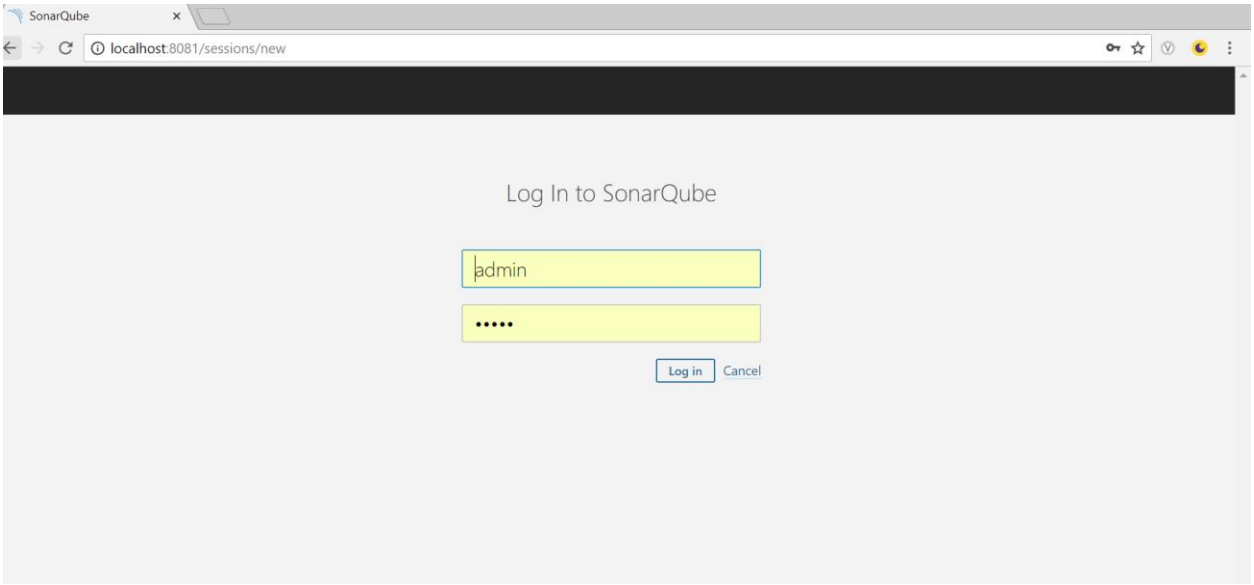
Put URL http://localhost:8081 in the browser. (I used 8081 coz I have set SonarQube server port as 8081)

You should be able to see below page



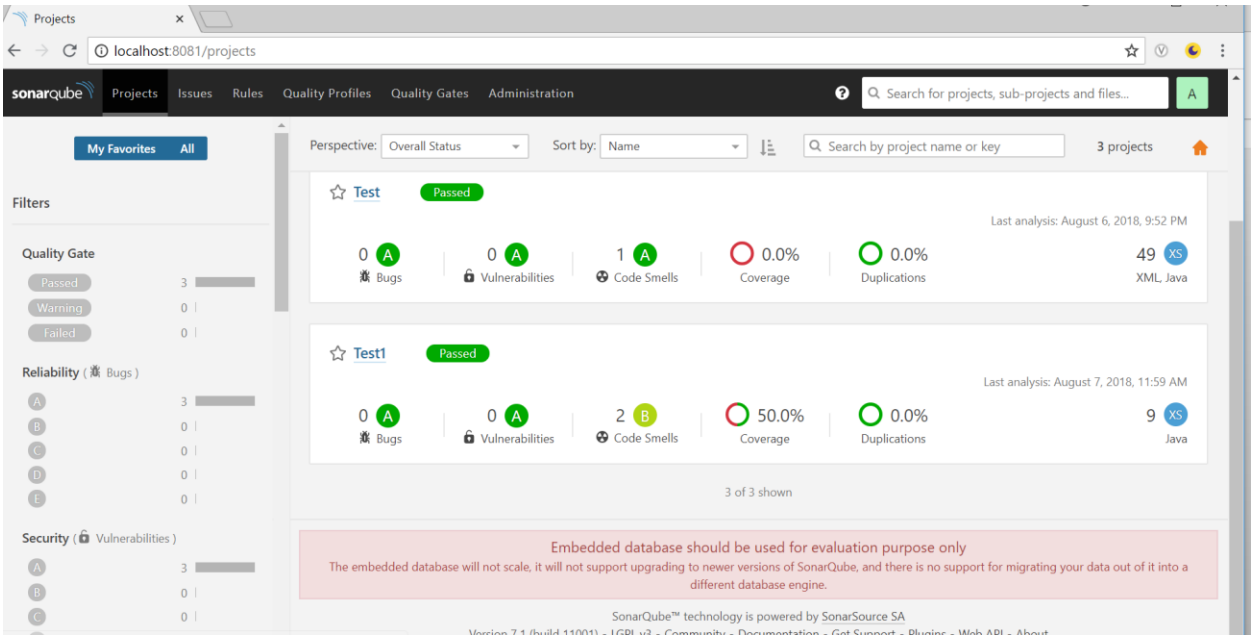
Step 10:

Click on “Log in” button on top right corner. And you should be able to see below page:



Step 11:

Enter username and password both as “admin” and click on “Log in”. You will be able to see your project in the sonar server.



I hope this document has helped you in initial setups. If you find any scope of improving this knowledge then feel free to write back.

Thank you,

Jyoti Prakash Behera

Mindtree Limited

IDCO Plot No-1,

Chandaka Special Economic Zone,

PO-KIIT Campus, bhubaneswar - 751024

M + 91 9040433367

E jyoti.behera@mindtree.com

Welcome to possible