**SonarQube**

| Attribute | Value |
| --- | --- |
| Owner | Gaurav Jethawa (430707) |
| Contact Information | [GJ00437070@techmahindra.com](mailto:GJ00437070@techmahindra.com) |

## 

Revision History

| Author | Date | Version # | Revision Description |
| --- | --- | --- | --- |
| Gaurav Jethawa | 29/03/2016 | 1.0 | Initial Draft Version  SonarQube Continuous Inspection tool |

Table of Contents

[SonarQube 3](#_Toc447624151)

[SonarQube Installation 3](#_Toc447624152)

[SonarQube with Maven 5](#_Toc447624153)

[SonarQube with Jenkins 7](#_Toc447624154)

[SonarQube with Sonar Scanner 9](#_Toc447624155)

[SonarQube local integration 9](#_Toc447624156)

[SonarQube with Eclipse (SonarLint) 10](#_Toc447624157)

[Analyzing Projects 10](#_Toc447624158)

[Troubleshooting 13](#_Toc447624159)

[References 13](#_Toc447624160)

# [SonarQube](http://en.wikipedia.org/wiki/SonarQube)

SonarQube (previously known as Sonar) is an open source platform for Continuous Inspection of code quality. It is written in java and supported for 25+ languages such as Java, C/C++, C#, PHP, Flex, Groovy, JavaScript, Python, PL/SQL, COBOL, etc., it is also used for Android Development.

It helps for various tasks and provides reports on duplicated code, coding standards, unit tests, code coverage, complex code, potential bugs, comments and design and architecture.

SonarQube is internally using PMD, Findbugs, CheckStyle etc. You can add additionally plugins according to your requirement

Advantages:

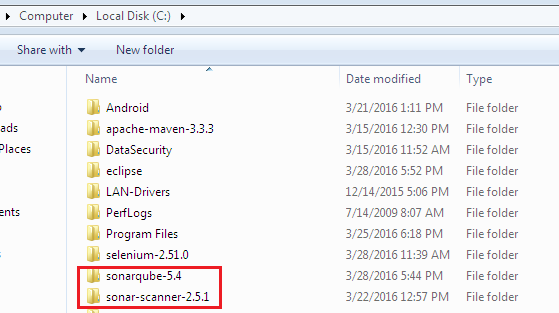
* SonarQube covers: Duplications, Coding standards, Lack of coverage, Potential bugs, Complexity, Documentation and Design, which are known as Developer’s Seven Deadly Sins
* SonarQube Dashboard offer quick insight to code, module, project or portfolio.
* Supports more than 25 languages including Java, C#, C/C++, Cobol, PL/SQL, ABAP, Javascript, PHP, Web, XML
* Automation : Quality analyses can be easily integrated into any continuous integration server to fully automate the process
* Easy to integrate.

# SonarQube Installation

1. Download latest version of SonarQube server and Sonar Scanner from :

<http://www.sonarqube.org/downloads/>

1. Unzip it in C:\



1. Start the SonarQube server

# On Windows, execute:

C:\ sonarqube-5.4\bin\windows-x86-xx\StartSonar.bat

# On other operating system, execute:

/etc/sonarqube-5.4/bin/[OS]/sonar.sh console

Following steps 4 and 5 are optional and required only if using standalone sonar runner to analyze the project.

1. Update sonar server url in properties file located at

C:\sonar-scanner-2.5.1\conf\sonar-runner.properties

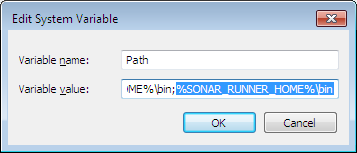
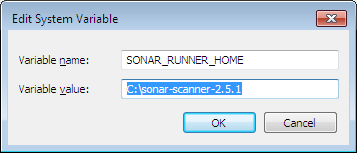
sonar.host.url=<http://INPUHJPC08769:9000>

by default it will be [http://localhost:9000](http://localhost:9000/)

1. Update environment variables.

Create a new SONAR\_RUNNER\_HOME environment variable set to <install\_directory>.

Add the <install\_directory>/bin directory to your path



# SonarQube with Maven

1. INITIAL SETUP : GLOBAL SETTINGS

Edit the settings.xml file, located in users/username/.m2 or $MAVEN\_HOME/conf, to set the plugin prefix and optionally the SonarQube server URL



<settings>

    <pluginGroups>

        <pluginGroup>**org.sonarsource.scanner.maven**</pluginGroup>

    </pluginGroups>

    <profiles>

        <profile>

            <id>sonar</id>

            <activation>

                <activeByDefault>true</activeByDefault>

            </activation>

            <properties>

<!-- sonar server url -->

**<sonar.host.url>**

[**http://INPUHJPC08769:9000**](http://INPUHJPC08769:9000)

**</sonar.host.url>**

            </properties>

        </profile>

     </profiles>

</settings>

1. ADD SONAR PLUGIN IN POM.xml

<plugin>

<groupId>org.sonarsource.scanner.maven</groupId>

<artifactId>sonar-maven-plugin</artifactId>

<version>3.0.1</version>

</plugin>

Add these sonar properties :

<properties>

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

<sonar.language>java</sonar.language>

</properties>

To analyze test classes which are in mavens test directory:

SonarQube follows Mavens folder structure to analyze source classes. Hence it doesn't analyze classes in test folder. SonarQube will analyze main classes and do code coverage for test classes through a separate Plugin. Hence we need to create a separate Profile and change it’s source directory to analyze test classes

<profile>

<id>analyze-test-classes</id>

<properties>

<sonar.sources>src/test/java</sonar.sources>

<sonar.tests></sonar.tests>

<sonar.projectName>${project.name}-tests</sonar.projectName>

<sonar.projectKey>${project.artifactId}-anyKey</sonar.projectKey>

</properties>

</profile>



1. ANALYZING A MAVEN PROJECT

Analyzing a Maven project consists of running a Maven goal: **sonar:sonar** in the directory where the pom.xml file sits.

Cmd: **mvn clean verify sonar:sonar**

In some situation you may want to run sonar:sonar goal as a dedicated step. Be sure to use install as first step for multi-module projects

Cmd: **mvn clean install**

Cmd: **mvn sonar:sonar**

Analyzing classes in test directory using maven profile

Cmd: **mvn sonar:sonar -P analyze-test-classes**

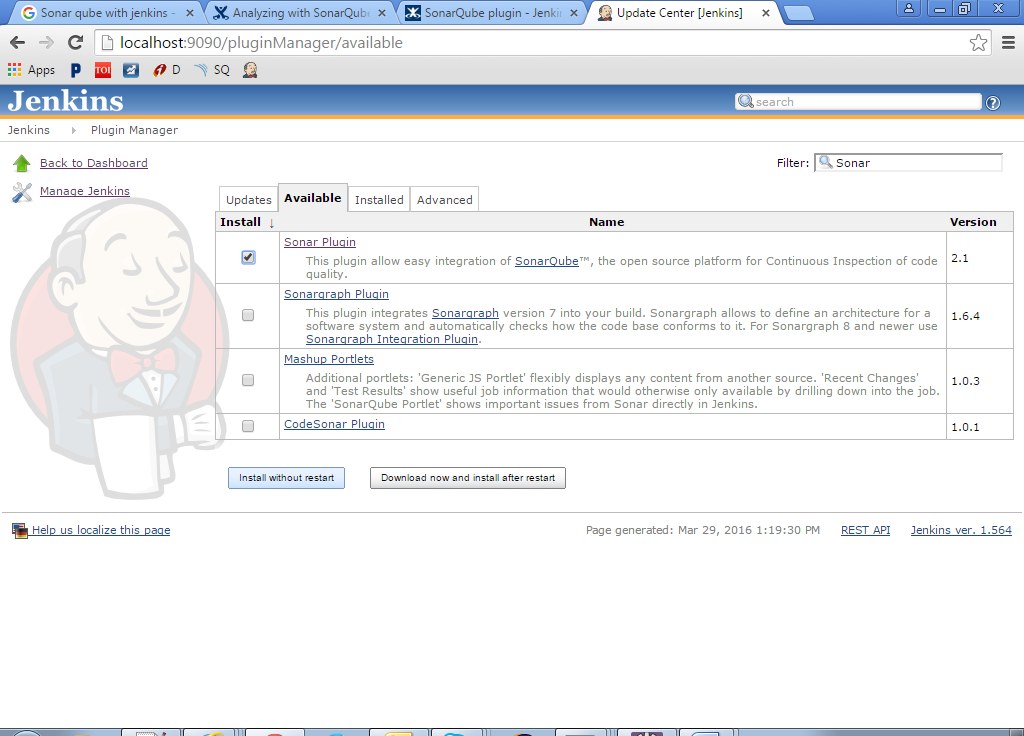
analyze-test-classes is the profile which we created in step 2.

1. See [Analyzing Projects](#_Analyzing_Projects) section to browse the results.

# SonarQube with Jenkins

1. Install SonarQube plugin:

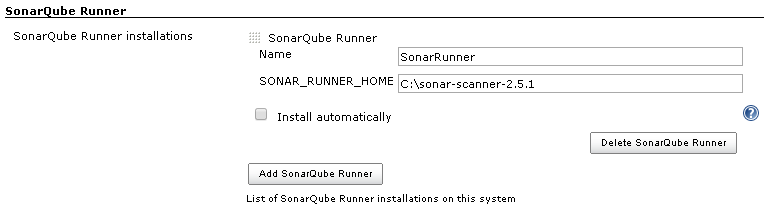
Go to Jenkins dashboard 🡪Manage Jenkins 🡪 Manage Plugins 🡪 Available tab. Search for Sonar Plugin and click on install button.

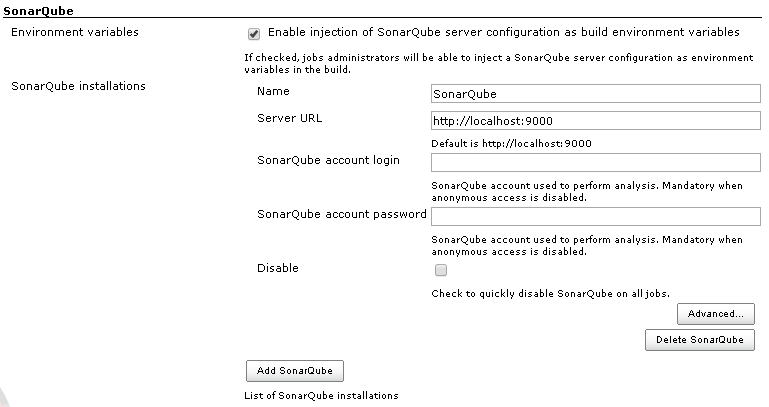


If above step doesn’t work due to lack of full internet access, go to Advanced tab, Upload Plugin option and upload the below sonar2.3.hpi file.

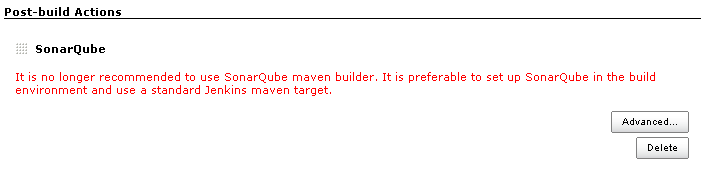


1. Go to Jenkins dashboard 🡪Manage Jenkins 🡪 Configure System and add SonarQube and Runner



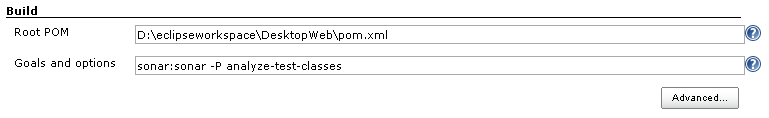


1. Go to the job configuration and under Post build Actions, select SonarQube.



Alternate way:

If you have configured SonarQube with Maven, you can trigger it from maven command using sonar:sonar



# SonarQube with Sonar Scanner

Analyzing with sonar scanner (The SonarQube Scanner is recommended as the default launcher to analyze a project with SonarQube)

1. Create a configuration file in the root directory of the project: ***sonar-project.properties***

****

1. Navigate to directory where ***sonar-project.properties*** is present and run the following command to launch the analysis:

D:\workspace\projectname>**C:\sonar-scanner-2.5.1\bin\*sonar-runner***

1. See [Analyzing Projects](#_Analyzing_Projects) section to browse the results.

# SonarQube local integration

1. Copy sonar-scanner-2.5.1 from shared folder [\\INPUHJPC08769\shared\sonar-scanner-2.5.1](file:///\\INPUHJPC08769\shared\sonar-scanner-2.5.1)
2. Paste this folder in D: drive. (D:\sonar-scanner-2.5.1\bin)
3. Refer [SonarQube with Sonar Scanner](#_SonarQube_with_Sonar) for next steps.
4. Mention appropriate values in sonar-project.properties like :

sonar.projectKey=**RogersPOC-{username}**

sonar.projectName=**RogersPOC-{username}-{purpose}**

sonar.language=js,java

sonar.sources=js/controllers, js/derivatives

**Sources**: is a comma-separated relative path to directories (after root folder )containing source files. E.g. src/main/java, js/controllers, etc

**Language**: If not set, a multi-language analysis will be triggered. Language Support:



1. Refer [Analyzing Projects](#_Analyzing_Projects) section to browse the results.

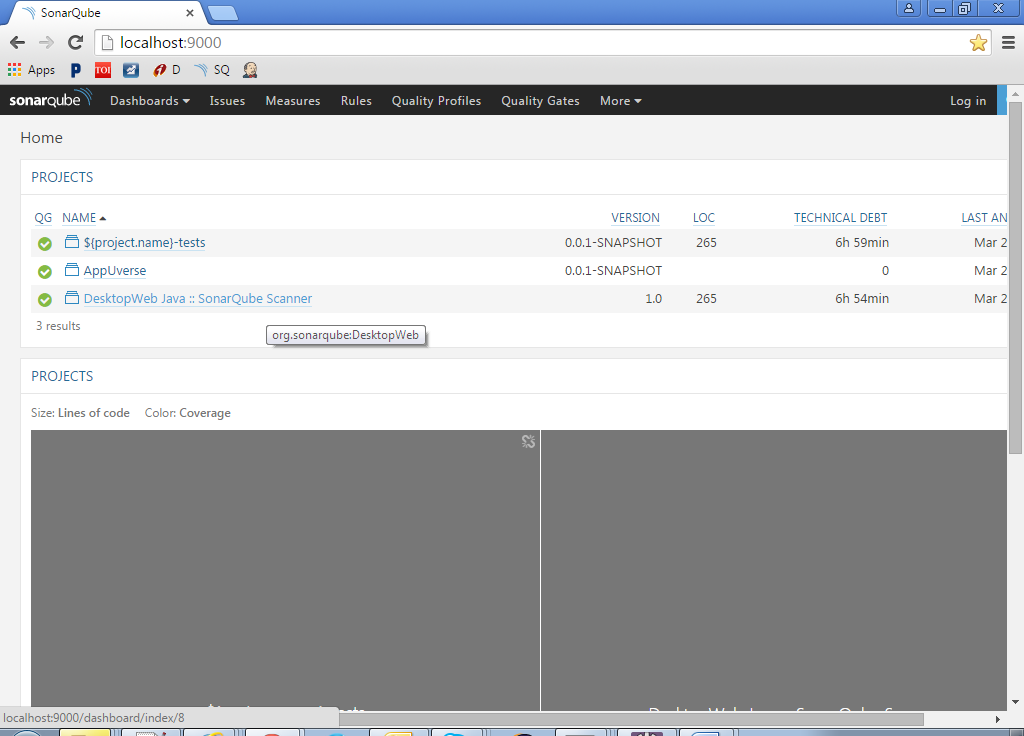
# SonarQube with Eclipse (SonarLint)

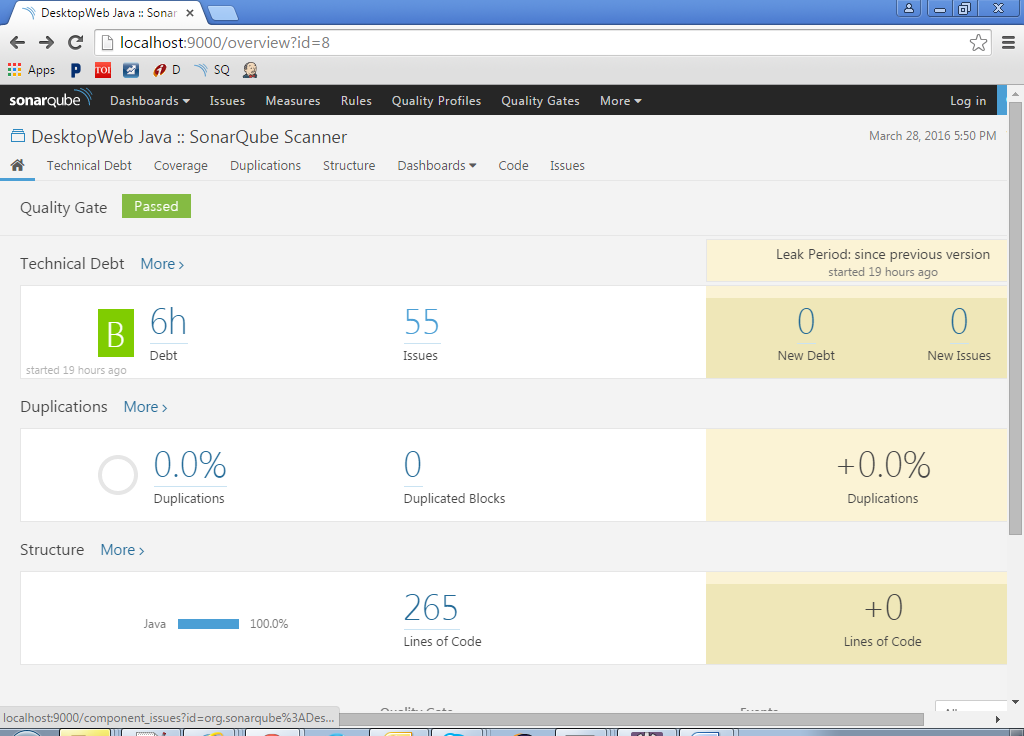
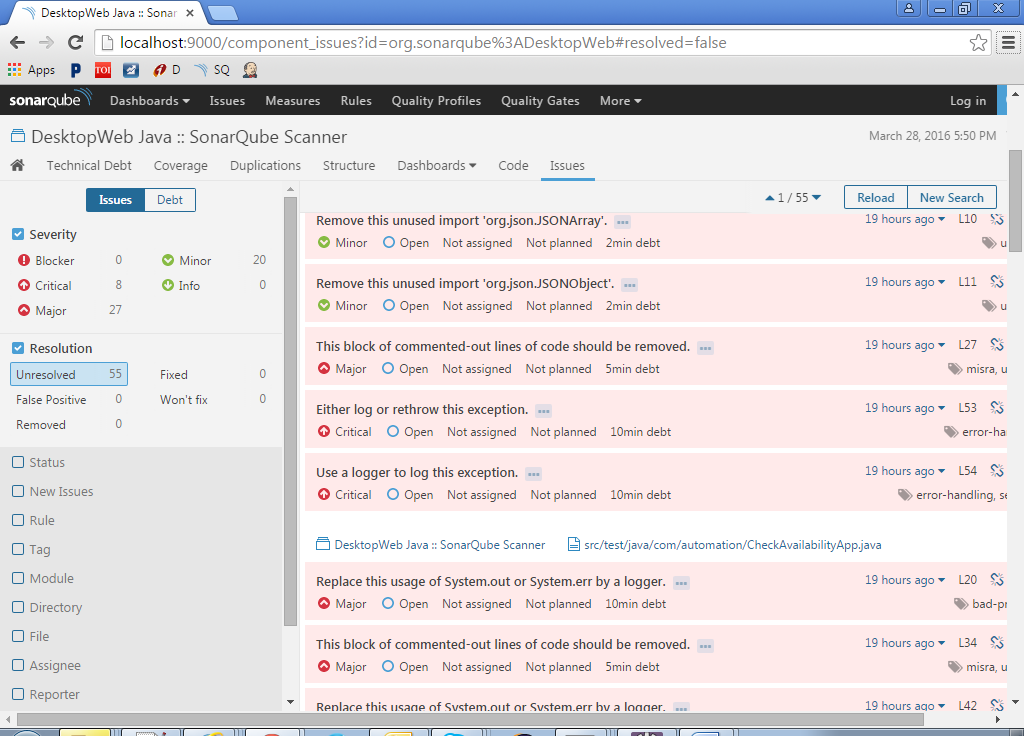
ToDo

# Analyzing Projects

1. Access SonarQube : <http://localhost:9000/>

Remove server: <http://10.10.21.172:9000>



1. Click Desired project 
2. Click on issues or technical debt.

# Troubleshooting

1. Build fails while using SonarQuber Runner for Jenkins with error

FATAL: Couldn’t find any executable in C:\apache-maven-3.3.3

[SonarQube analysis completed](http://stacktrace.jenkins-ci.org/search?query=SonarQube%20analysis%20completed): FAILURE

Build step 'SonarQube' changed build result to FAILURE

Build step 'SonarQube' marked build as failure

Finished: FAILURE

-OPEN

# References

<http://docs.sonarqube.org/display/SONAR/Documentation/>

http://www.javatips.net/blog/2013/10/sonarqube-tutorial

<http://docs.sonarqube.org/display/SONAR/Get+Started+in+Two+Minutes>

<http://docs.sonarqube.org/display/SCAN/Analyzing+with+SonarQube+Scanner>

<http://docs.sonarqube.org/display/SCAN/Analyzing+with+SonarQube+Scanner+for+Maven>

http://docs.sonarqube.org/display/SCAN/Analyzing+with+SonarQube+Scanner+for+Jenkins