

|  |
| --- |
|  |

**Sonar Installation and Configuration**

**Guide**

# Table of Contents

[Table of Contents 2](#_Toc420428206)

[1. Introduction 4](#_Toc420428207)

[2. prerequiste 4](#_Toc420428208)

[3. installation and configuration 4](#_Toc420428209)

[3.1 **Installation** and **Sonar.properties File:** 5](#_Toc420428210)

[3.2 **SonarQube runner:** 5](#_Toc420428211)

[3.2.1 sonar-runner.properties File: 6](#_Toc420428212)

[**3.3 Sonar-project.properties File:** 6](#_Toc420428213)

[**3.4 Analyzer Execution:** 7](#_Toc420428214)

[4.Download and install the plugin SonarQube Jenkins 9](#_Toc420428215)

[**4.1 Plugin installation** 9](#_Toc420428216)

[**4.2 Configuring the SonarQube Runner** 10](#_Toc420428217)

[**4.3 Configuring SonarQube** 11](#_Toc420428218)

[5.SONAR TASK IN GRUNT 14](#_Toc420428219)

[**5.1 Installation of task** 14](#_Toc420428220)

[**5.2 SonarRunner task in gruntfile** 14](#_Toc420428221)

[**5.3 Running sonarRunner task** 16](#_Toc420428223)

# 1.Introduction

[**Sonar**](http://www.sonarsource.org/) is an open-source product which is used to gather several metrics about code quality, put them all in a single dashboard, and provide some tips to help you making your code better, more sustainable, more reliable, less bugged.

# 2.prerequiste

**Java**

The SonarQube Java analyzer is able to analyze any kind of java source files whatever is the version of Java they comply to. But SonarQube analysis and SonarQube server require some specific versions of the JVM to be executed.EG-7u5+,8.

**Database**

The charset of the database has to be set to “UTF-8” and the language to “English”.eg: mysql,oracle,Microsoft sql server,PostgreSQL.

**Web Browser**

To get the full experience SonarQube has to offer, you must enable JavaScript in your browser.

Eg.safari,opera,chrome,Mozilla

# 3.installation and configuration

* Download latest version of Sonar. e.g 5.0 Stable Release.
* wget http://dist.sonar.codehaus.org/sonarqube-5.0.1.zip
* Unzip the package using command: unzip sonarPakageName.zip
* unzip sonarqube-5.0.1.zip
* mv sonarqube-5.0.1 /opt/sonar
* Open to the file which is in folder: /conf/sonar.properties

## **3.1 sonar.properties File:**

In that file you can configure sonar settings related to Database and HTTP.  
Just to make it easy and time saving we just set the database settings and rest of it will be on default settings.

1.# Permissions to create tables, indices and triggers must be granted to JDBC user.

2.# The schema must be created first.

3.sonar.jdbc.username=<databse username>

4.sonar.jdbc.password=<database password>

1. #sonar.web.context=/sonar

# TCP port for incoming HTTP connections. Disabled when value is -1.

#sonar.web.port=9000

1.#----- MySQL 5.x

2.# Comment the embedded database and uncomment the following line to use MySQL

3.sonar.jdbc.url=jdbc:<mysql://localhost>:3306/sonar?useUnicode=true&characterEncoding=utf8&rewriteBatchedStatements=true

In case if you find any issue while connecting to database then change the localhost to ipAddress of your system.

Create a DB which you have mentioned in “sonar.jdbc.url”. For example above we have used “sonar” as our database name.

## **3.2 SonarQube runner:**

* Download SonarQube runner latest version. It is responsible for analyzing the project and it automatically submit the results in database. Which sonar use for reporting purposes.
* Unzip the package using command: unzip sonar-runner
* Open the file which is in folder: /conf/sonar-runner.properties

## **3.2.1 sonar-runner.properties File:**

In this file you will find Database and security related settings of sonar qube.Uncomment any driver of your choice.

1.#----- MySQL

2.sonar.jdbc.url=jdbc:<mysql://localhost>:3306/sonar?useUnicode=true&characterEncoding=utf8

1.#----- Global database settings

2.sonar.jdbc.username={userName}

3.sonar.jdbc.password={password}

## **3.3 Sonar-project.properties File:**

sonar-project.properties file contains information about project and source folders. This file persists at root of your project directory. Below is the content which we can define in that file.

01.# Required metadata

02.sonar.projectKey=com.javapitshopt

03.sonar.projectName=javapitshop

04.sonar.projectVersion=1.1.6

05.

06.# Paths to source directories.

07.# Paths are relative to the sonar-project.properties file. Replace "\" by "/" on Windows.

08.# Do not put the "sonar-project.properties" file in the same directory with the source code.

09.# (i.e. never set the "sonar.sources" property to ".")

10.sonar.sources=src,WebContent,build/classes

11.

12.# The value of the property must be the key of the language.

13.sonar.language=js

14.

15.# Encoding of the source code

16.sonar.sourceEncoding=UTF-8

17.

18.# Additional parameters

19.#sonar.my.property=value

## **3.4 Analyzer Execution:**

All the configurations are completed. Now we need to run sonar and analyze our project with sonar-qube runner.

To execute sonar move to the bin directory in sonar base folder and execute command.

* **./sonar.sh start**      [in case of any issue try doing that with root permission or with sudo.]

Hit URL localhost:9000 in browser and you will find sonar web page in front of you.Now move to your project root directory where sonar-project.properties file is being saved and execute command.

* **/pathToSonarRunnerDirectory/bin/sonar-runner**

It will start analyzing the project and populate the database so that sonar can use it to display the statistics.After its completion we can check the results at sonar portal by refreshing the portal.

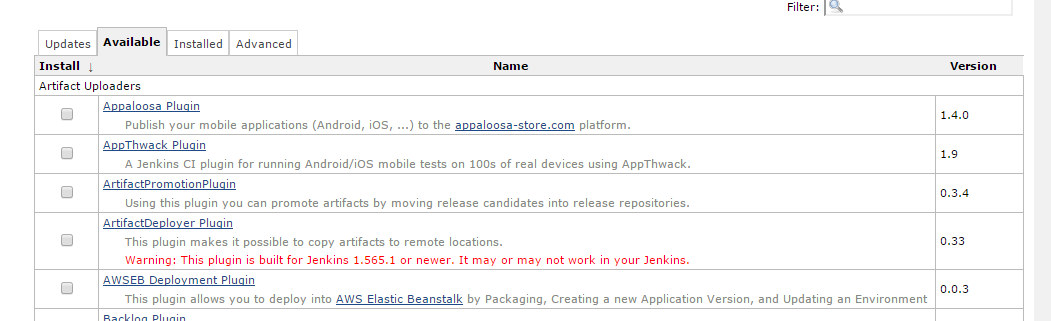
# 4.Download and install the plugin SonarQube Jenkins

## **4.1 Plugin installation**

In the Administration page of Jenkins, activate the menu to manage plugins:

[](http://qualilogy.com/fr/wp-content/uploads/sites/2/2013/05/JenkinsAdminMenu.jpg)

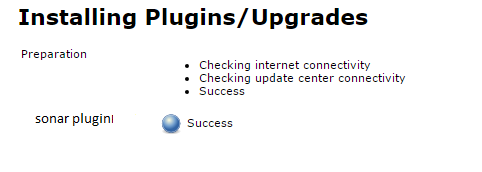
On the next page, select the tab for all the available plugins:



Search and select the SonarQube plugin. I suggest you do a search on the string ‘Sonar’ to quickly find the plugin in this particularly long list.



At the bottom of the page, click the button ‘Install without restart': Jenkins starts the installation and once it is finished…



tells us that the SonarQube plugin has been updated. We can return to the Administration page of Jenkins to select the setup menu, which will allow us to set our SonarQube installation.

## **4.2 Configuring the SonarQube Runner**

Jenkins will use the SonarQube Runner to run analysis of code.In order to configure it, we

return to the Administration page of Jenkins to activate the menu of configuration.

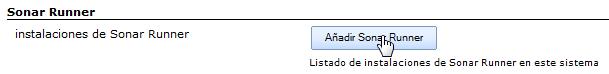


Then, Jenkins proposes us to configure or even install a JDK.

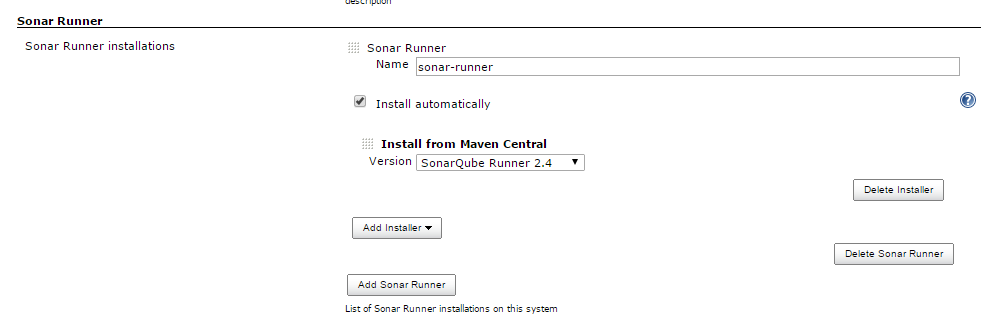
[](http://qualilogy.com/fr/wp-content/uploads/sites/2/2013/05/JenkinsConfigJDK.jpg)

This is not necessary if you have proceeded as indicated during the [installation of a Java JDK](http://qualilogy.com/en/install-sonar-jdk/), creating a JAVA\_HOME variable and declaring in the path of your PC, the directory containing the java executables (%JAVA\_HOME%\bin).

Below, a section devoted to the SonarQube Runner will allow us to configure it. First, click the button to add an instance of SonarQube Runner.

[](http://qualilogy.com/fr/wp-content/uploads/sites/2/2013/05/JenkinsAddSonarRunner.jpg)

This will open a section to enter its configuration:



In this section:

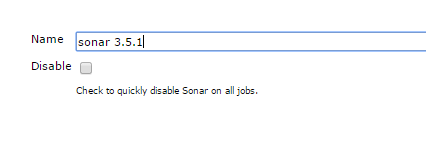
* Uncheck the automatic installation option.
* Enter a name for you SonarQube Runner instance.
* Enter its localisation.

And finally, do not forget to save your settings. It’s done.

## **4.3 Configuring SonarQube**

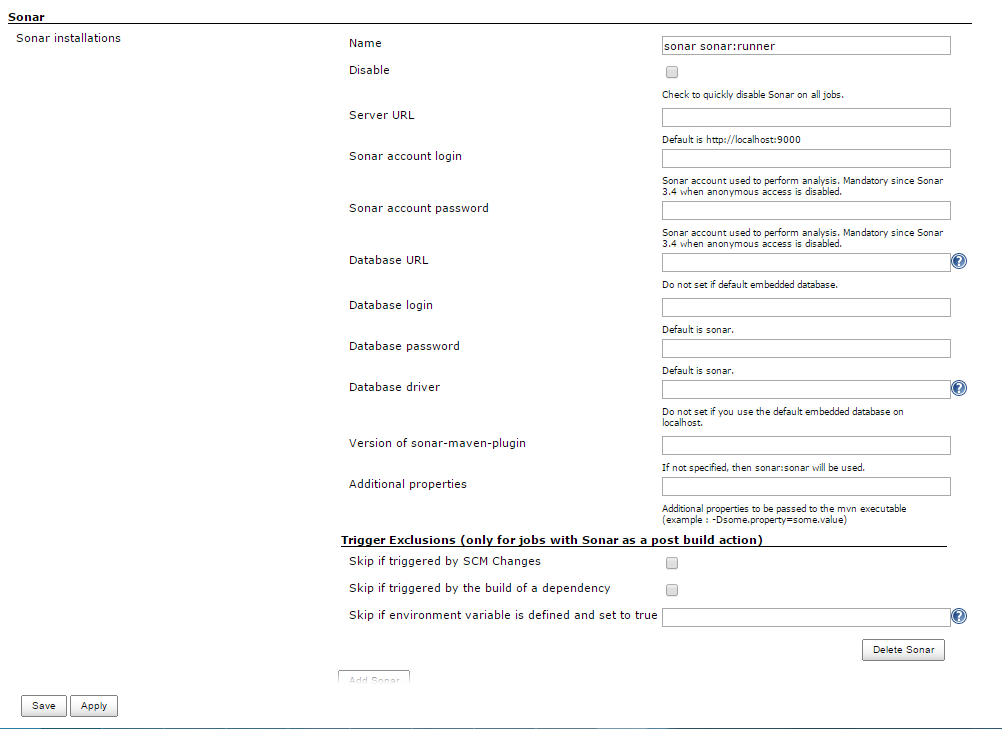
The configuration of SonarQube into Jenkins will follow the same logic. In this same configuration page of the ‘Jenkins system’, locate the section dedicated to SonarQube:

[](http://qualilogy.com/fr/wp-content/uploads/sites/2/2013/05/JenkinsAddSonar.jpg)

Again, click the button to create a new SonarQube instance

Jenkins then asks to enter a name to identify this SonarQube installation.

These are the same as those listed in our ‘sonar-properties’ file, to configure SonarQube. So just open this file to use the same parameters:



We indicate:

* The url of our SonarQube web application.
* The access to our database.
* The JDBC driver.

Don’t forget to save your settings with the button located at the end of the page.

Now we can use Jenkins to create and run SonarQube analysis.

# 5.SONAR TASK IN GRUNT

## **5.1 Installation of task**

Sonar Analysis Runner from grunt. The major version is kept in sync with sonar runner distribution version.

**npm install grunt-sonar-runner --save-dev**

Once the plugin has been installed, it may be enabled inside your Gruntfile with this line of JavaScript:

**grunt.loadNpmTasks('grunt-sonar-runner');**

## [**5.2 sonarRunner task**](https://www.npmjs.com/package/grunt-sonar-runner#the-sonarrunner-task) **in gruntfile**

In your project's Gruntfile, add a section named sonarRunner to the data object passed into

grunt.initConfig().

grunt.initConfig({

sonarRunner**:** {

        analysis**:** {

            options**:** {

                debug**:** true,

                separator**:** '\n',

                sonar**:** {

                    host**:** {

                        url**:** 'http://localhost:9000'

                    },

                    jdbc**:** {

                        url**:** 'jdbc:mysql://localhost:3306/sonar',

                        username**:** 'sonar',

                        password**:** 'sonar'

                    },

                    projectKey**:** 'sonar:grunt-sonar-runner:0.1.0',

                    projectName**:** 'Grunt Sonar Runner',

                    projectVersion**:** '0.10',

                    sources**:** ['test'].join(','),

                    language**:** 'js',

                    sourceEncoding**:** 'UTF-8'

                }

            }

        }

    }

});

### [**Options**](https://www.npmjs.com/package/grunt-sonar-runner#options)

#### [**options.debug**](https://www.npmjs.com/package/grunt-sonar-runner#options-debug)

Type: Boolean Default value: 'false'

A Boolean value to display debug information when preparing sonar analysis.

#### [**options.separator**](https://www.npmjs.com/package/grunt-sonar-runner#options-separator)

Type: String Default value: os.EOL

A string value representing a new line character to separate individual sonar properties

#### [**options.sonar.host.url**](https://www.npmjs.com/package/grunt-sonar-runner#options-sonar-host-url)

Type: String Default value: 'http://localhost:9000'

Sonar Dashboard URL

#### [**options.sonar.jdbc.url**](https://www.npmjs.com/package/grunt-sonar-runner#options-sonar-jdbc-url)

Type: String Default value: ''

JDBC connection url

#### [**options.sonar.jdbc.username**](https://www.npmjs.com/package/grunt-sonar-runner#options-sonar-jdbc-username)

Type: String Default value: ''

JDBC connection username

#### [**options.sonar.jdbc.password**](https://www.npmjs.com/package/grunt-sonar-runner#options-sonar-jdbc-password)

Type: String Default value: ''

JDBC connection password

#### [**options.sonar.sources**](https://www.npmjs.com/package/grunt-sonar-runner#options-sonar-sources)

Type: String Default value: ''

comma separated list of directories to analyse. All js files in the provided directories will be included.

#### [**options.sonar.projectKey**](https://www.npmjs.com/package/grunt-sonar-runner#options-sonar-projectkey)

Type: String Default value: ''

project key usually of form group:artifactId:version

#### [**options.sonar.projectName**](https://www.npmjs.com/package/grunt-sonar-runner#options-sonar-projectname)

Type: String Default value: ''

Project Summary

#### [**options.sonar.projectVersion**](https://www.npmjs.com/package/grunt-sonar-runner#options-sonar-projectversion)

Type: String Default value: ''

Current project build version

#### [**options.sonar.language**](https://www.npmjs.com/package/grunt-sonar-runner#options-sonar-language)

Type: String Default value: null

Sets the project language. If this property is undefined,.

## [**5.3 running sonarRunner task**](https://www.npmjs.com/package/grunt-sonar-runner#the-sonarrunner-task)

**grunt sonarRunner:analysis**