

|  |  |
| --- | --- |
| Document name | SonarQube Installation Guide |
| Version no. | 1.1 |
| Release date | 11-Feb-15 |



|  |
| --- |
| Guidelines of Lunch Committee  (HR-GLC) |

|  |
| --- |
| Guidelines of Lunch Committee  (HR-GLC) |

SonarQube Installation Guide

##### Document History

| Version No. | Authored / Modified by | Reviewed by, Date | Approved by, Date | Remark / Change History |
| --- | --- | --- | --- | --- |
| 1.0 | Rucha Kulkarni  06-Feb-15 | Rahul Pandoh  06-Feb-15 | Varghese Mathew  09-Feb-15 | Initial Document |
| 1.1 | Rucha Kulkarni  11-Feb-15 | Vidya Jadhav  11-Feb-15 | Varghese Mathew  11-Feb-15 | Prerequisites and compatibility |

Contents

[Prerequisites and compatibility 3](#_Toc411352005)

[Sonar setup on Linux 5](#_Toc411352006)

1. [Download stable version of Sonar 5](#_Toc411352007)
2. [Create the database and user with proper access credentials 7](#_Toc411352008)
3. [Make required changes in configuration files 8](#_Toc411352009)
4. [Start sonar service (root user and proper permission required) 9](#_Toc411352010)
5. [Verify the success 11](#_Toc411352011)
6. [Trouble shooting and Error logs 13](#_Toc411352012)

[Sonar setup on Windows 14](#_Toc411352013)

1. [Steps to setup Sonar 14](#_Toc411352014)
2. [Steps to start Sonar web server 15](#_Toc411352015)

[Sonar setup with Eclipse 17](#_Toc411352016)

1. [Installing Sonar plugin for Eclipse 17](#_Toc411352017)
2. [Sonar Web Browser View: 23](#_Toc411352018)

[Sonar for Maven 2 and Maven 3 24](#_Toc411352019)

1. [Start the Sonar webserver as follows: 24](#_Toc411352020)
2. [Set up Sonar server and other configurations in Maven's settings.xml file as follows: 24](#_Toc411352021)
3. [Analyzing a Maven Project for Sonar 25](#_Toc411352022)
4. [Run Sonar as part of a build: 25](#_Toc411352023)
5. [Sonar Web Browser View: 30](#_Toc411352024)

[Sonar only of Jenkins 34](#_Toc411352025)

1. [Configuring the Sonar plugin 34](#_Toc411352026)

[References 44](#_Toc411352027)

### Prerequisites and compatibility

#### Hardware Requirements

* The SonarQube server requires at least 1GB of RAM to run efficiently. However, the amount of disk space you need will depend on how much code you analyze with SonarQube.

#### Supported platforms

1. Java

|  |  |
| --- | --- |
| [**Oracle JRE**](http://java.sun.com/javase/downloads) | (error) 6 (supported for all the versions before SonarQube LTS 4.5.1)  (tick) 7 (faster analysis than with version 6 - u55+ is recommended)  (tick) 8 (since SonarQube 4.3) |
| **OpenJDK** | (error) 6  (tick) 7 (u55+ is recommended)  (tick) 8 |

1. Compatibility with Eclipse IDE:

|  |  |
| --- | --- |
| **SonarQube** | 3.6+ |
| [**Eclipse**](http://eclipse.org/) | 3.7.x, 3.8.x, 4.2.x, 4.3.x (Indigo, Juno, Kepler) |

1. Database Server

|  |  |
| --- | --- |
| [**Microsoft SQL Server**](http://www.microsoft.com/sqlserver/) | (tick) 2008 (MSSQL Server 10.0) with bundled jTDS driver. **Microsoft drivers are not supported**. Express Edition is supported. (tick) 2012 (MSSQL Server 11.0) with bundled jTDS driver. **Microsoft drivers are not supported**. Express Edition is supported.  (warning) Collation must be case-sensitive (CS) and accent-sensitive (AS) |
| [**MySQL**](http://www.mysql.com/) | (tick) 5.1 and 5.5 (tick) 5.6 is supported since SonarQube 3.5 (tick) 5.7 is supported since SonarQube 4.4  (warning) Only InnoDB storage engine is supported, but not MyISAM (warning) Only the bundled mysql-connector-java jar is supported |
| [**Oracle**](http://www.oracle.com/database/) | (tick) 10G with [Oracle 11.2.x drivers](http://www.oracle.com/technetwork/database/enterprise-edition/jdbc-112010-090769.html)  (tick) 11G with [Oracle 11.2.x drivers](http://www.oracle.com/technetwork/database/enterprise-edition/jdbc-112010-090769.html)  (tick) XE Editions are supported   (warning) The driver ojdbc14.jar is not supported  (warning) Only the thin mode is supported, not OCI |
| [**PostgreSQL**](http://www.postgresql.org/) | (tick) 8.x (tick) 9.x |

**Note**: The charset of the database has to be set to "UTF-8" and the language to "English".

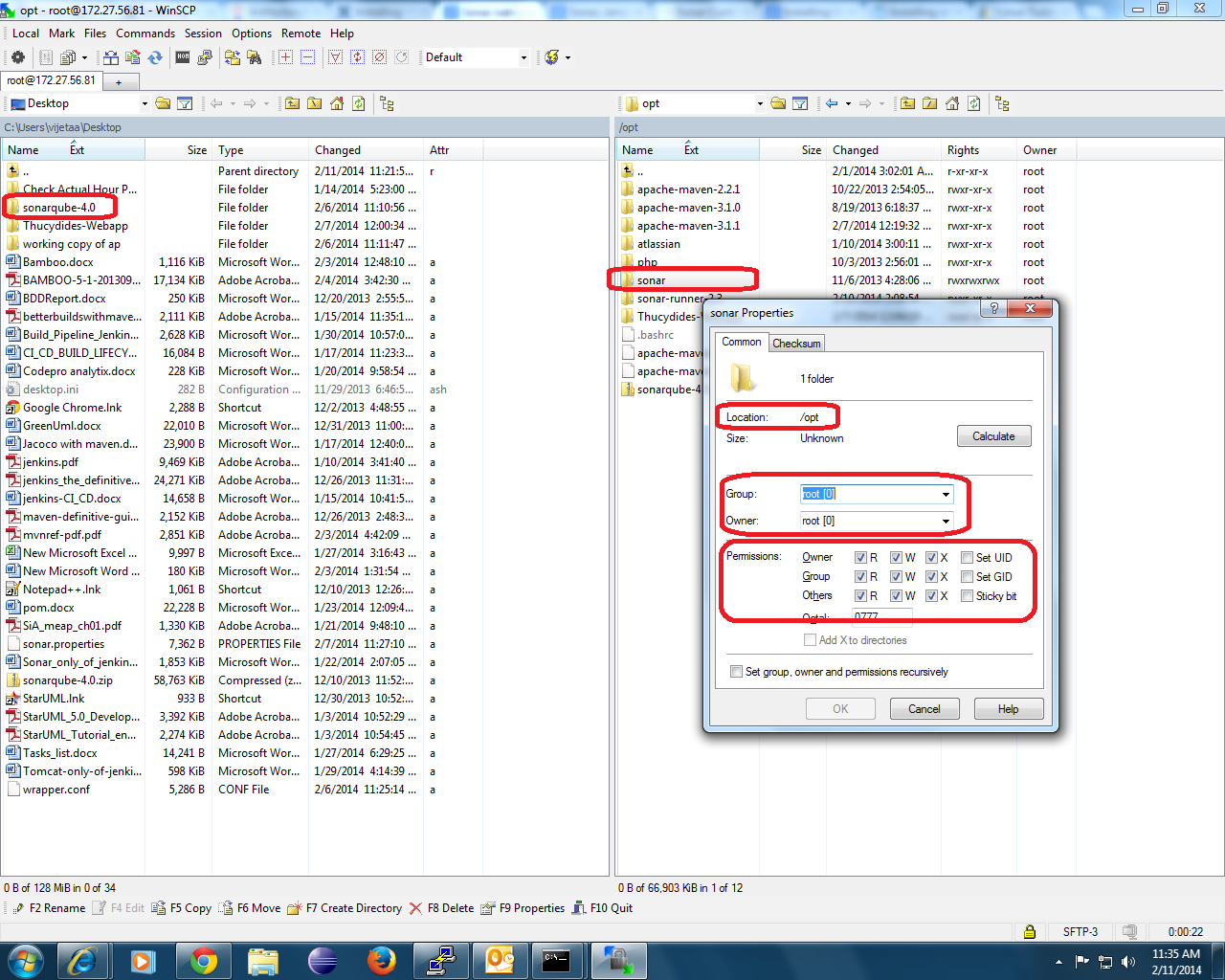
# Sonar setup on Linux

## Download stable version of Sonar

* 1. **Download SonarQube distribution on Windows client and upload on Linux machine using FTP**
     1. Download and unzip the stable SonarQube distribution from following URL

e.g. version 4.0 from <http://www.sonarqube.org/downloads/>

* + 1. Upload the folder on Linux machine using FTP (We have chosen folder /opt)
    2. Rename the folder to ‘sonar’
    3. Make sure it has all the permissions required. Sonar service requires root user to run.



* 1. **Download on Linux machine directly**
     1. Login to Linux server using SSH / Putty
     2. Goto to desired directory e.g. cd /opt
     3. Download and unzip the stable SonarQube distribution using following command which downloads version 4.0 and rename folder to ‘sonar’

1: wget http://dist.sonar.codehaus.org/sonarqube-4.0.zip

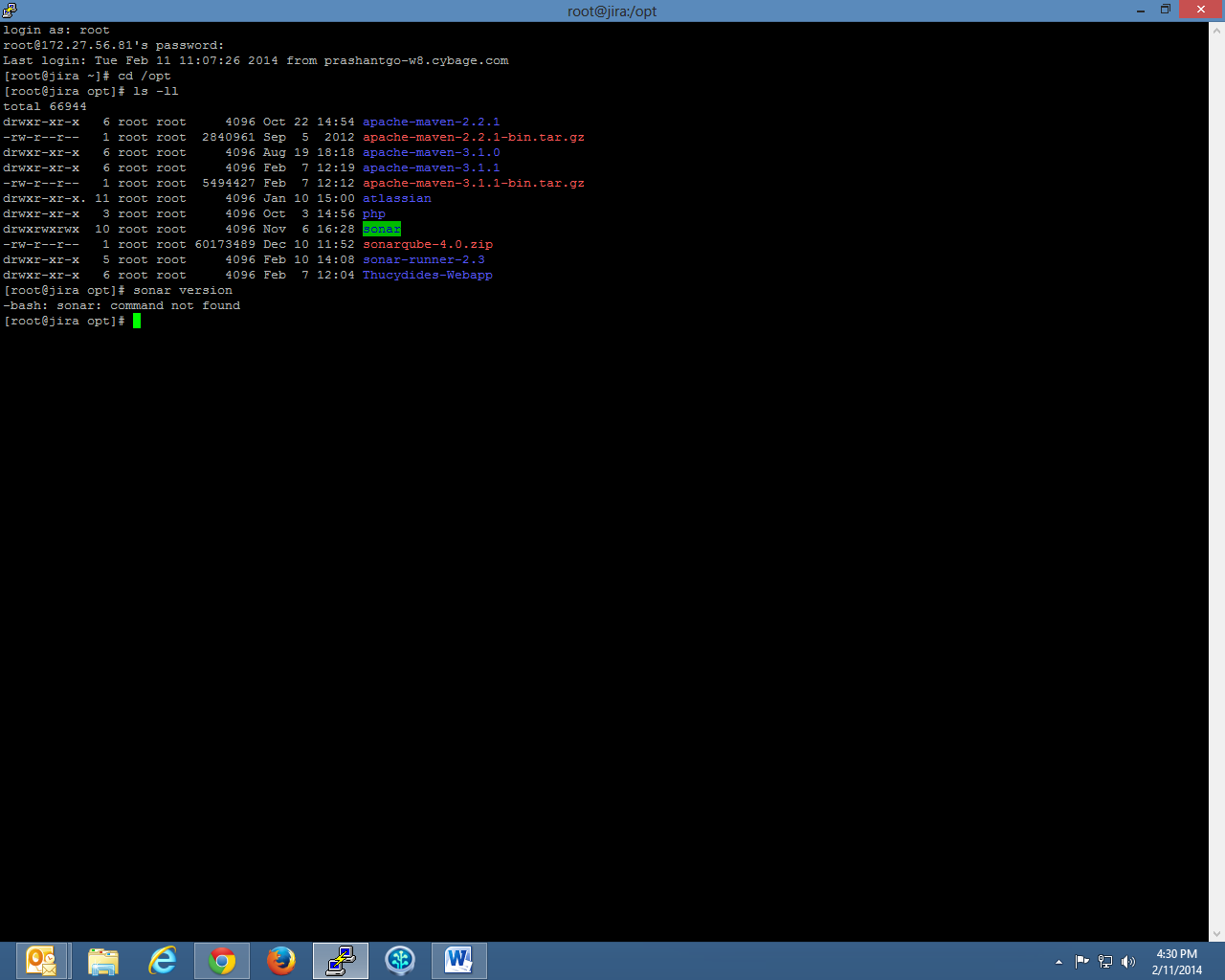
2: unzip sonarqube-4.0.zip

3: mv sonarqube-4.0 sonar

* + 1. Make sure it has all the permissions required. Sonar service requires root user to run. Following command will give all permissions to all users for sonar.

User the permissions 755 / 765 etc as per your need.

* + - 1. chmod –R 777 sonar



## Create the database and user with proper access credentials

We have used MySQL database in this example

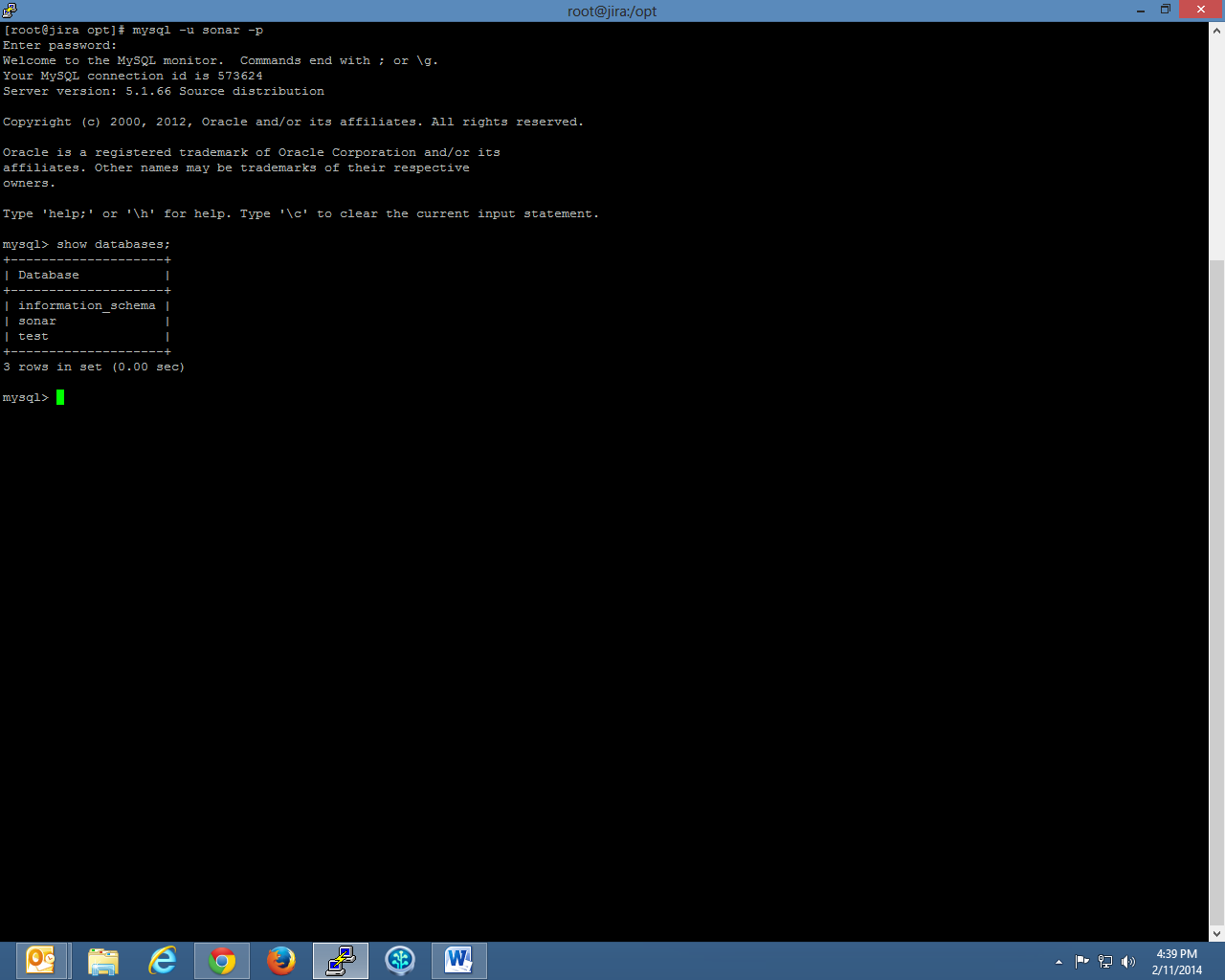
* 1. Create a database called ‘sonar’ (may need root user access)
  2. Create a user ‘sonar’ with all privileges to ‘sonar’ database

Mysql > CREATE DATABASE sonar CHARACTER SET utf8 COLLATE utf8\_general\_ci;

Mysql > grant all privileges on sonar.**\*** to 'sonar'**@**'localhost' identified by 'sonarpassword';

Mysql > flush privileges;

* 1. Verification of this step can be done using steps given in following screenshot



## Make required changes in configuration files

We have used host IP as 172.27.56.81 and port as 9090 for sonar.

Since sonar and mysql is hosted on same server, for database, we have used localhost..

You can replace these IP address, port etc with appropriate values

Add/edit following lines in this file. Comment similar lines for repeat/ duplicate entries if any.

sonar.web.host: 172.27.56.81

sonar.web.port:                           9090

sonar.web.context:                        /sonar

Server: http:// 172.27.56.81:9090

sonar.host.url: http:// 172.27.56.81:9090

sonar.jdbc.username:                       sonar

sonar.jdbc.password:                       sonarpassword

sonar.jdbc.maxActive: 10

sonar.jdbc.maxIdle: 5

sonar.jdbc.minIdle: 2

sonar.jdbc.maxWait: 5000

sonar.jdbc.minEvictableIdleTimeMillis: 600000

sonar.jdbc.timeBetweenEvictionRunsMillis: 30000

sonar.jdbc.url:jdbc:mysql://localhost:3306/sonar

**Note:** **If sonar instance has to be accessed from an Eclipse IDE installed at different IP address, update sonar.jdbc.url: jdbc:mysql://localhost:3306/sonar to sonar.jdbc.url: jdbc:mysql://<Network IP of this machine>:3306/sonar. Also, update etc/mysql/my.cnf entry bind-address to Network IP of this machine from 127.0.0.1.**

## Start sonar service (root user and proper permission required)

Once the configuration files are changed use following commands to start sonar

(We are assuming 64bit Linux OS. If this defers, choose appropriate folder instead of highlighted below)

sudo service sonar start

OR

./opt/sonar/bin/linux-x86-64/sonar.sh start

Verify sonar is running using command as below …

[root@jira /]# sudo service sonar status

sonar is running (13533).

Similarly, you can use following commands for respective purposes…

sudo service sonar stop

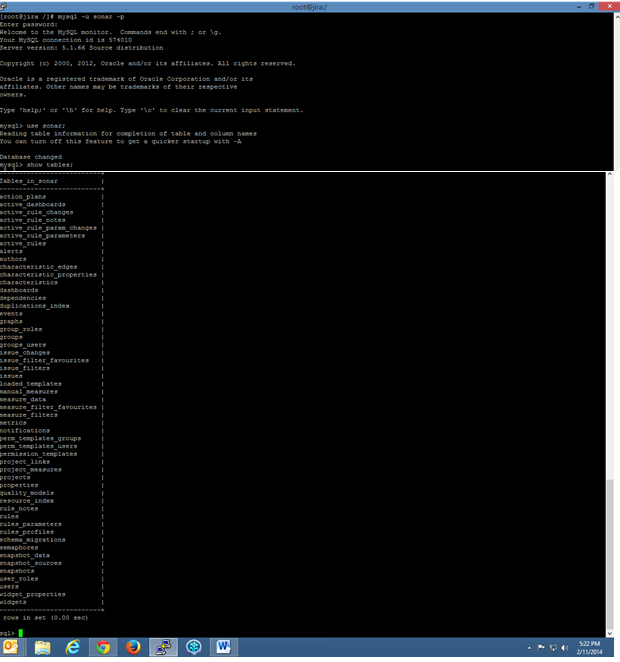
sudo service sonar restart

**NOTE:**

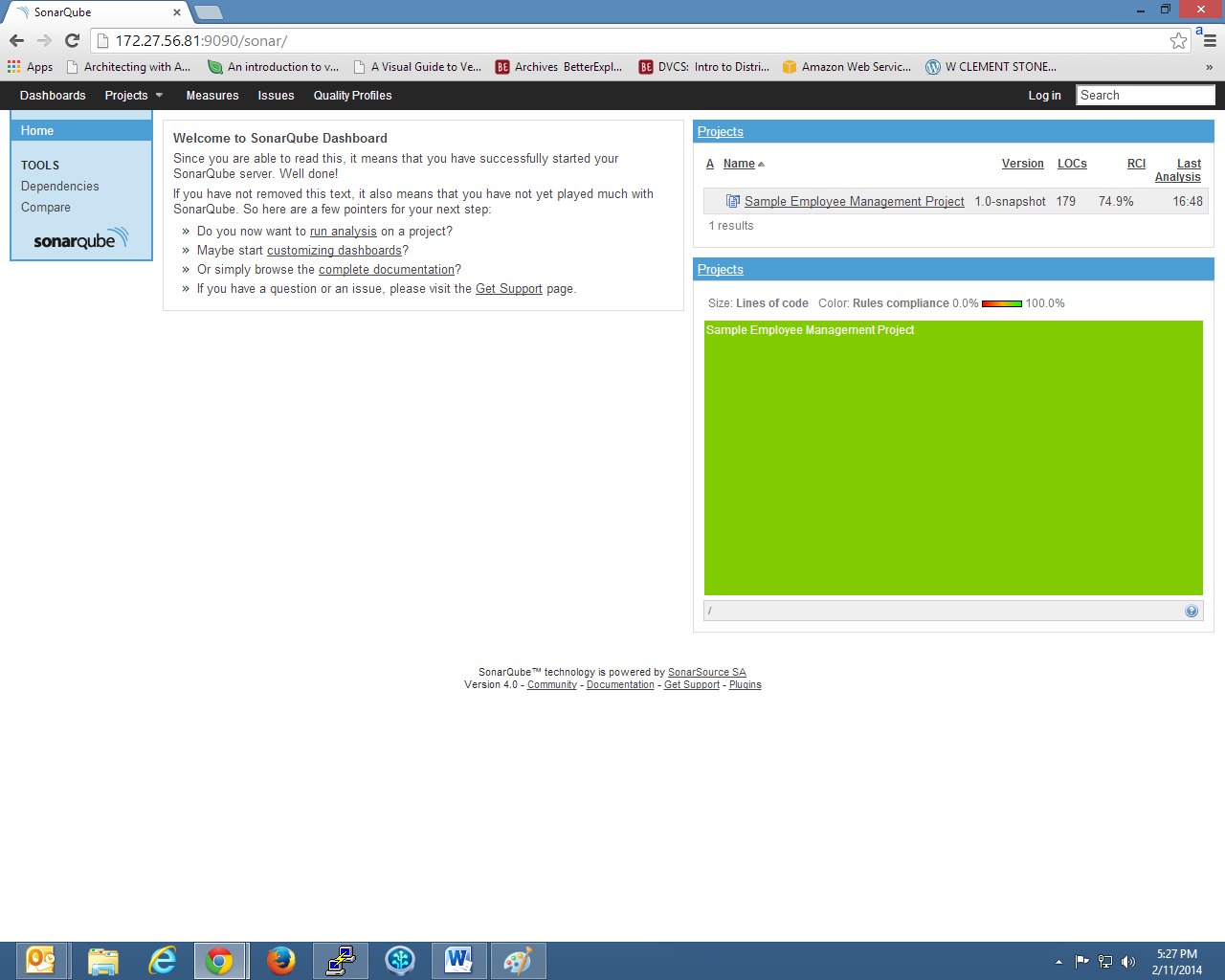
* **For all versions of sonarQube, you need to configure these two files. Only one property from wrapper.conf and six properties from sonar.properties(marked in red) need to be configured. Don’t replace these files while upgrading or while changing version. Configure these properties manually.**
* **Whenever any changes in any configuration files are made, you need to restart sonar service.**

## Verify the success

1: Check if there are tables created in sonar database



2: Check the URL <http://172.27.56.81:9090/sonar/> is working…



## Trouble shooting and Error logs

* 1. Permission related errors you can do either of following
     + 1. Use command ‘chmod’ to set proper permissions
       2. Set proper permissions using FTP client.
  2. Sonar still not working as expected…
     1. Check the error log generated in following folder

cd /opt/sonar/logs/

ls -ll

* 1. Other errors
     1. There is GOOGLE ☺

# Sonar setup on Windows

## Steps to setup Sonar

* 1. [Download](http://www.sonarsource.org/downloads/) and unzip the SonarQube 3.7.2 distribution (let's say in " D:\sonar Qube-3.7.2-server") Refer link <http://www.sonarqube.org/downloads/>
  2. Setup Wrapper Java properties into “D:\sonar Qube-3.7.2-server\conf\**wraper.conf**”

as shown below :

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Wrapper Java Properties

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# JVM

wrapper.java.command=C:/Program Files (x86)/Java/jdk1.6.0\_45/bin

wrapper.java.command=java

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

* 1. Setup Sonar properties into “D:\sonar Qube-3.7.2-server\conf\**sonar.properties**”

as shown below :

#---------------------------------------------------------

# WEB SETTINGS - STANDALONE MODE ONLY

# These settings are ignored when the war file is deployed to a JEE server.

#---------------------------------------------------------

# Listen host/port and context path (for example / or /sonar). Default values are #0.0.0.0:9000/.

sonar.web.host: localhost

sonar.web.port: 9000

sonar.web.context: /sonar

Server: http://localhost:9000

# Permissions to create tables and indexes must be granted to JDBC user.

# The schema must be created first.

sonar.host.url=http://localhost:9000/

sonar.jdbc.username: root

sonar.jdbc.password: vidya

#----- Connection pool settings

sonar.jdbc.maxActive: 10

sonar.jdbc.maxIdle: 5

sonar.jdbc.minIdle: 2

sonar.jdbc.maxWait: 5000

sonar.jdbc.minEvictableIdleTimeMillis: 600000

sonar.jdbc.timeBetweenEvictionRunsMillis: 30000

#----- MySQL 5.x

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

sonar.jdbc.url:jdbc:mysql://localhost:3306/sonar

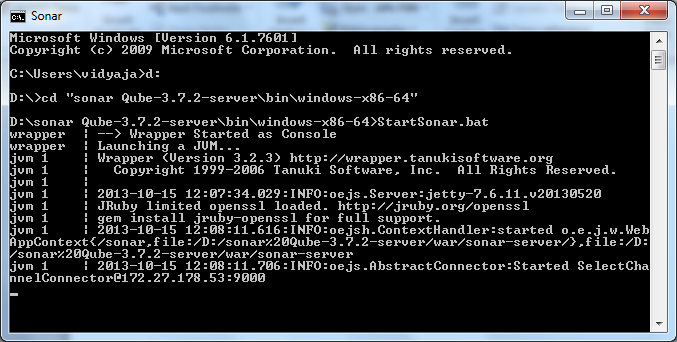
**NOTE:**

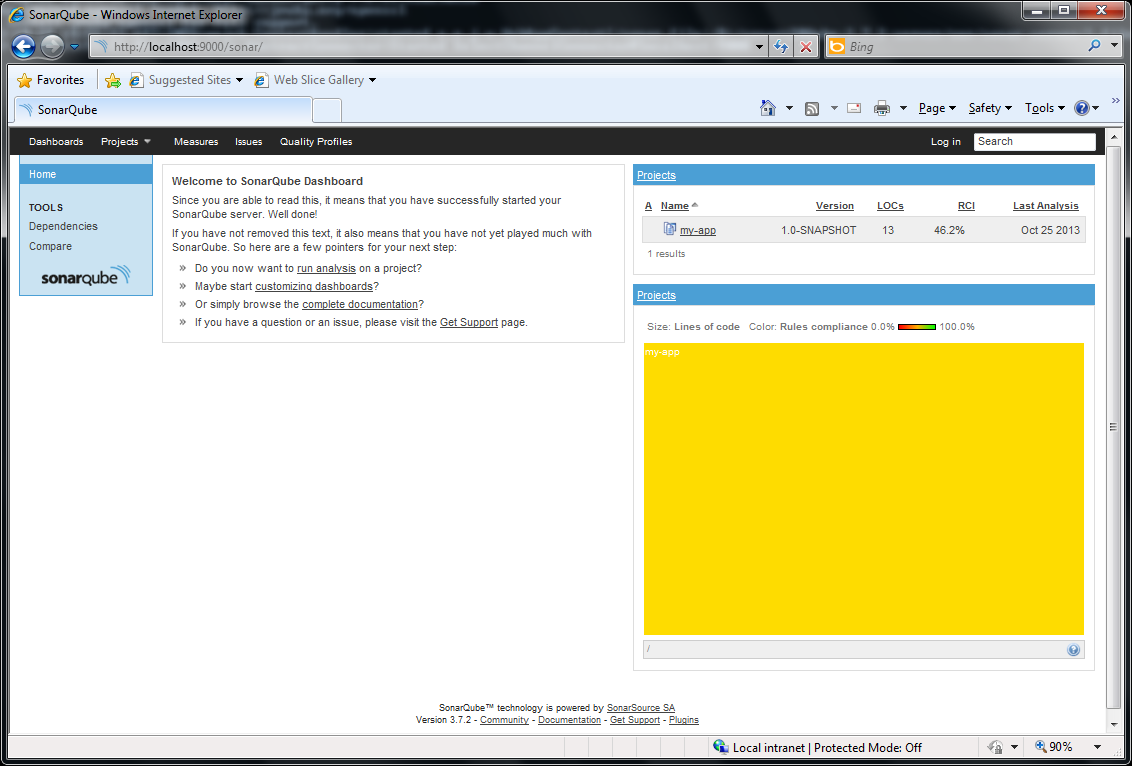
* **For all versions of sonarQube, you need to configure these two files. Only one property from wrapper.conf and six properties from sonar.properties(marked in red) need to be configured. Don’t replace these files while upgrading or while changing version. Configure these properties manually.**
* **Whenever any changes in any configuration files are made, you need to restart sonar service.**

## Steps to start Sonar web server

* 1. Execute the following script to start the server:

On Windows: bin/windows-x86-XX/StartSonar.bat as shown in below screenshot :



* 1. You can now browse SonarQube at <http://localhost:9000> as shown in below screenshot

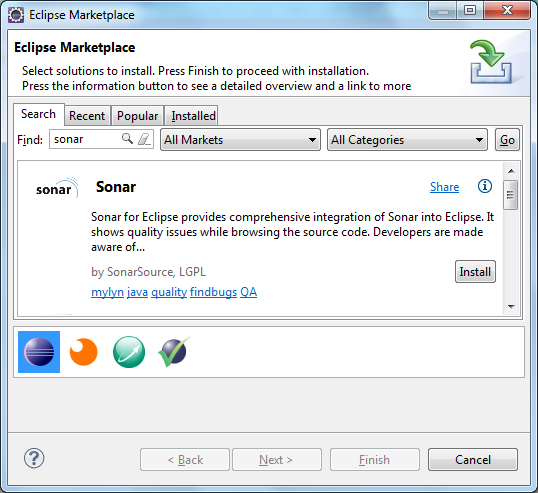
Upgrade database issue then:

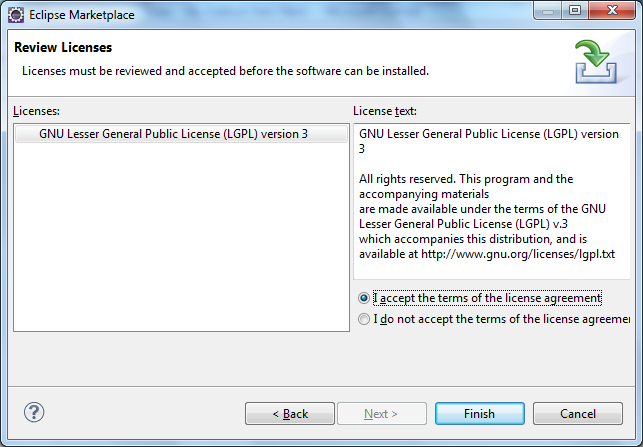
Drop the existing table and create new schema

And //http:localhost:9000/sonar/setup

# Sonar setup with Eclipse

## Installing Sonar plugin for Eclipse

1. Select from the menu "Help | Eclipse Marketplace" to start the installation process. As shown in screenshots below : 

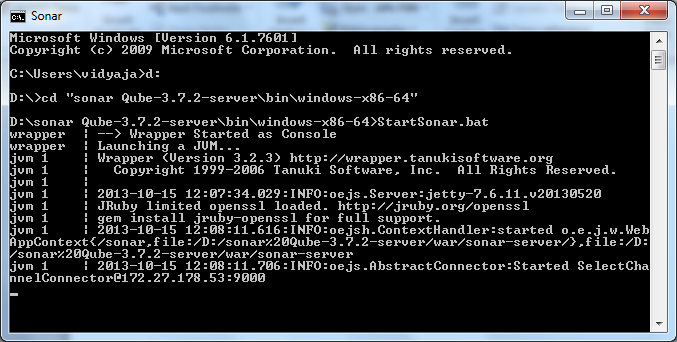


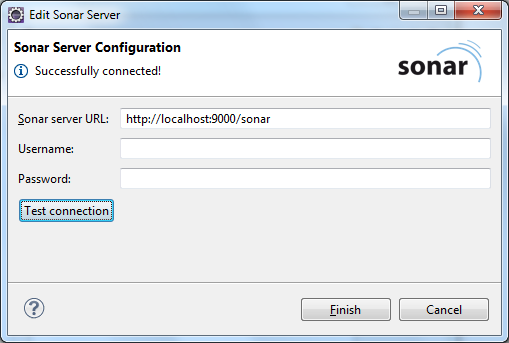
1. Once the installation process is finished, Eclipse will ask if you want to restart the IDE. It is ***strongly recommended that you restart the IDE***.
2. Once the installation is completed as shown above, start the Sonar webserver as follows :

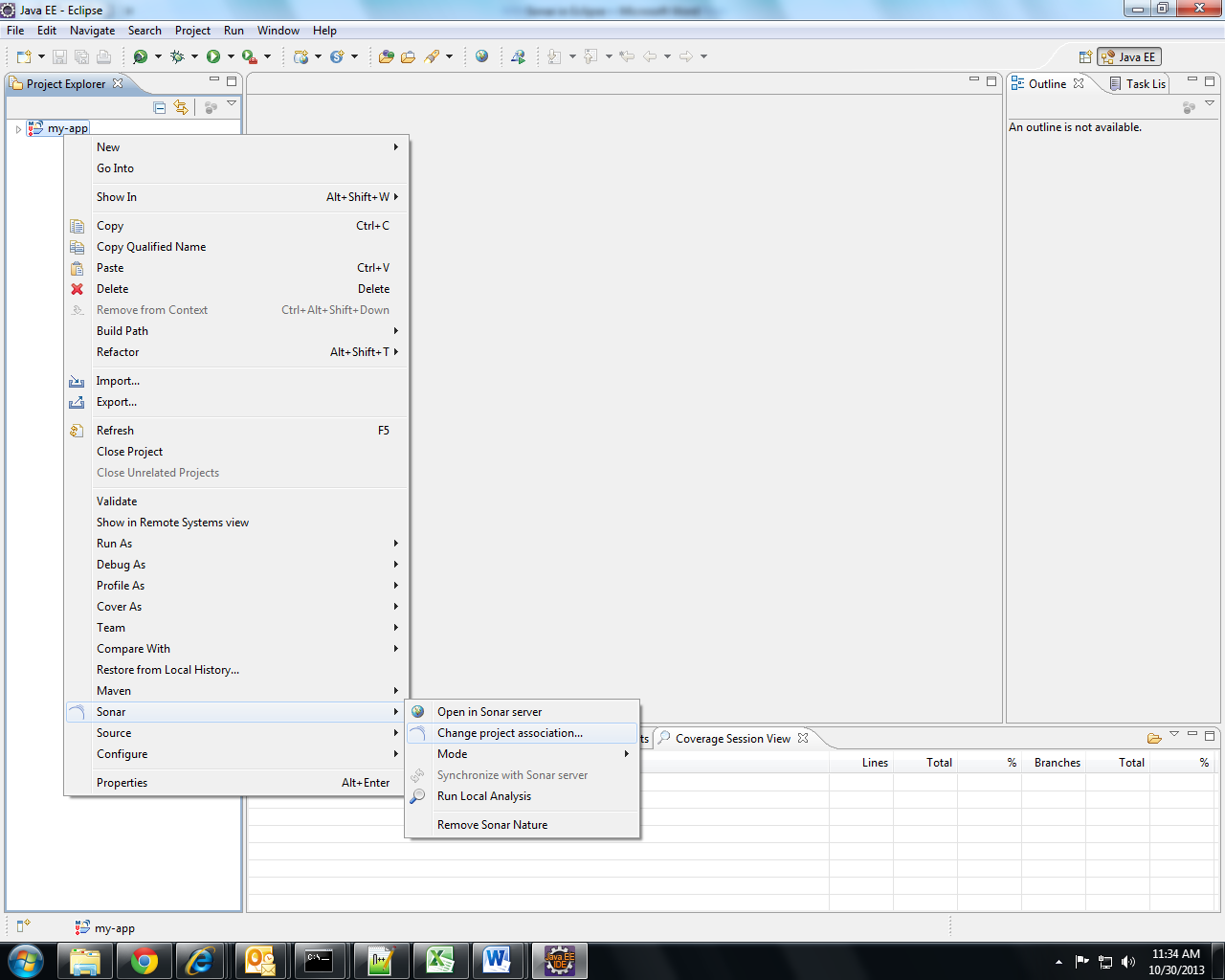
Refer chapter “Sonar setup on windows” to setup Sonar server.

Execute the following script to start the server:

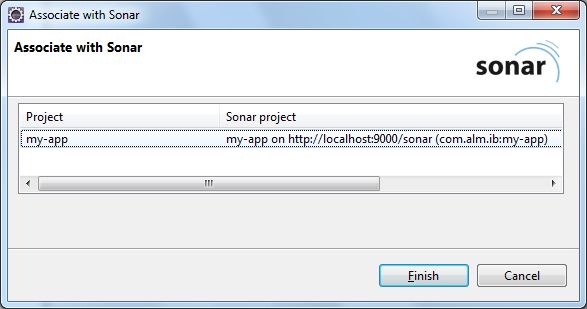
On Windows: bin/windows-x86-XX/StartSonar.bat as shown in below screenshot:



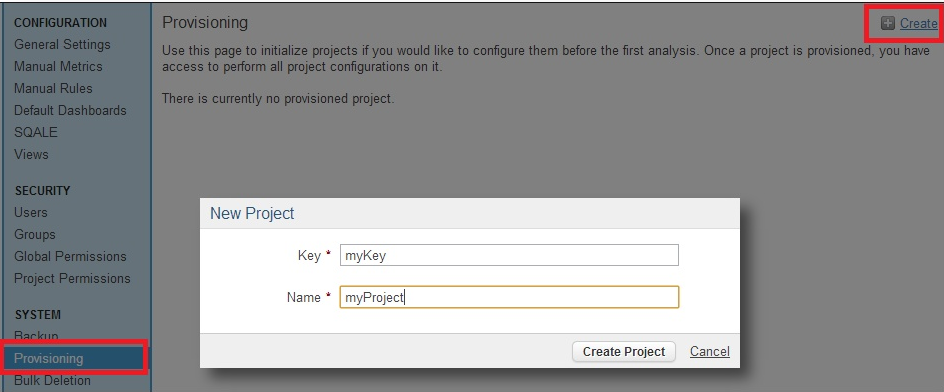
1. Go to Window| Preferences| Sonar| Servers and test the Sonar connection as shown in below screenshot
2. Go to Project Explorer : right click on Project and go to Sonar| Change Project Association as shown in below screenshot:



1. Associate your Eclipse project with a sonar project as shown below. As you start entering name in ‘sonar project’ column you will get list of sonar projects if Eclipse is properly connected to Sonar:

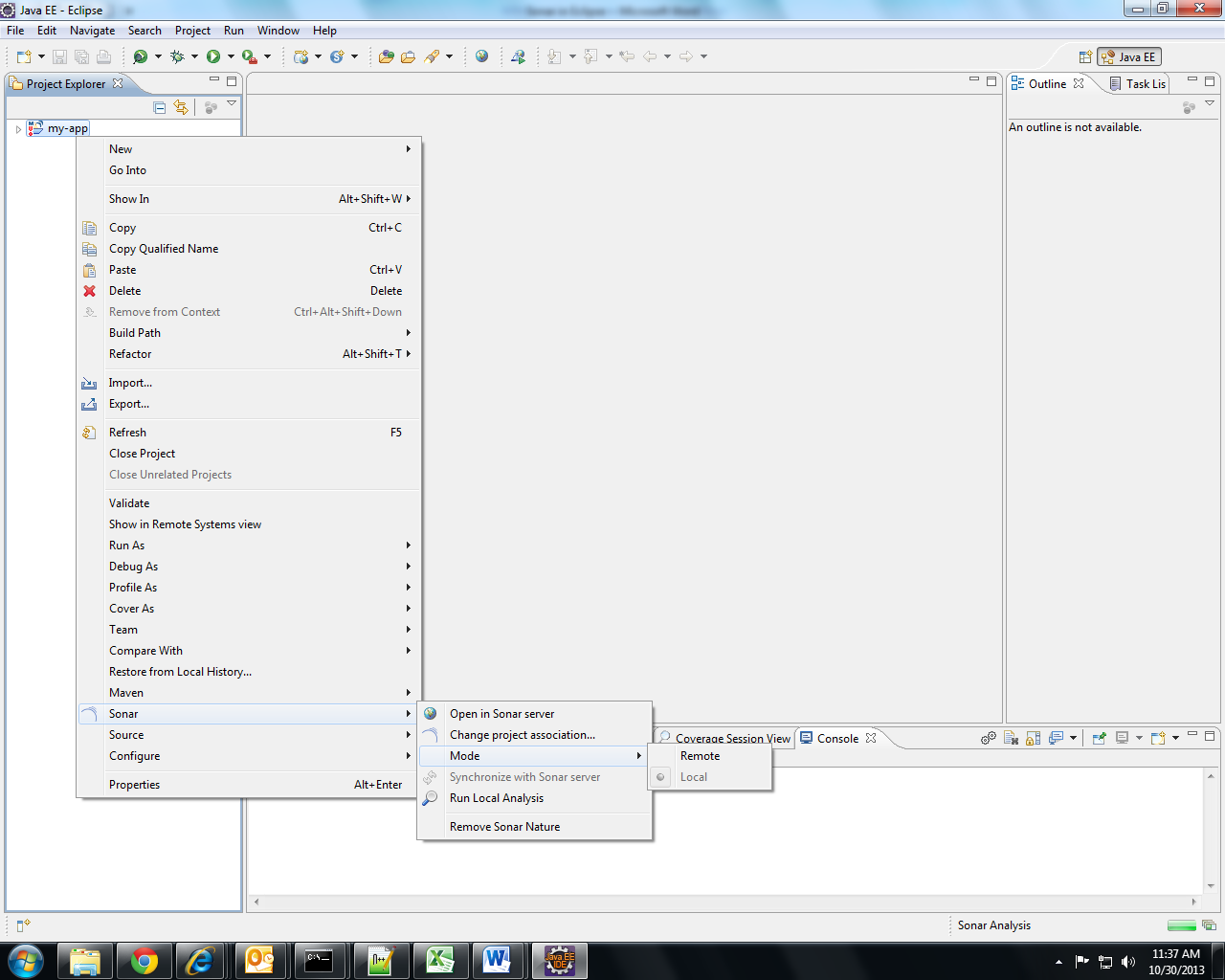


If there is not a single project in Sonar, create one using provisioning option. First login as administrator and in settings, you will get provisioning option. Key can be anything but Name should be ArtifactId of project:

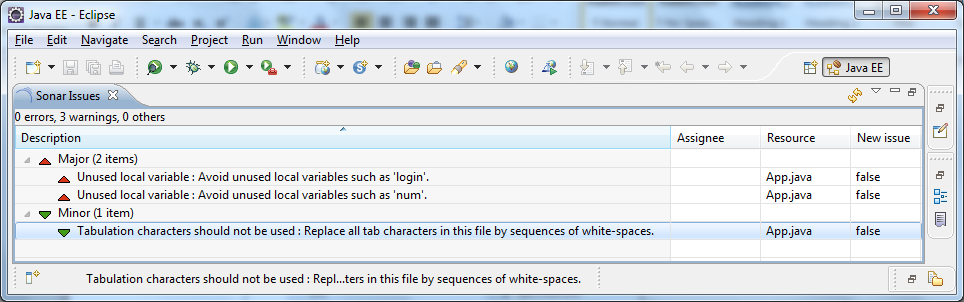


Your project is now associated to be analyzed on your SonarQube server. This means that you'll be able to see existing issues for the project in Eclipse, and that when you run a local analysis in Eclipse, it will use the rules in SonarQube profile to analyze the project rather than the language's default profile.

Select mode as Local as shown below: When switching to the Local mode, a local analysis will run automatically



1. Now Run local analysis and Sonar issues will be displayed on console as shown below :



## Sonar Web Browser View:

The **Web** view displays the main project dashboard (for project) or the component viewer (for class or file).

This web page can also be used to access information and services (Clouds, Hotspots, etc.) that are not available in SonarQube Eclipse.

As shown below Dashboard gives overall report for all issues divided into Blocker, Critical, Major, Minor, Info and also other parameters such as Unit tests Coverage, Rules compliance, Unit test Success, Lines of code, Duplications, Complexity etc



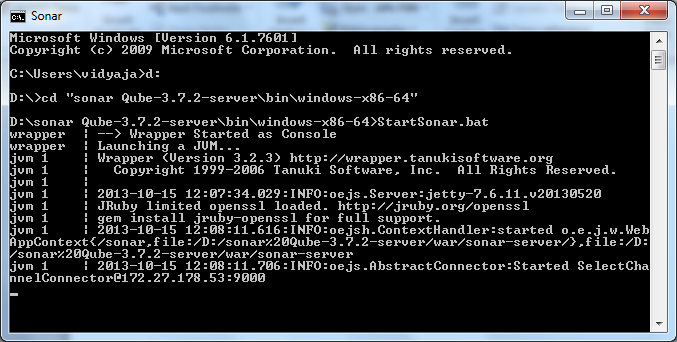
# Sonar for Maven 2 and Maven 3

## Start the Sonar webserver as follows:

Refer document “Sonar setup on windows” to setup Sonar server.

Execute the following script to start the server:

On Windows: bin/windows-x86-XX/StartSonar.bat as shown in below screenshot:



## Set up [Sonar](http://www.sonarsource.org/) server and other configurations in Maven's [settings.xml](http://maven.apache.org/settings.html) file as follows:

<profiles>

<profile>

<id>sonar</id>

<activation>

<activeByDefault>true</activeByDefault>

</activation>

<properties>

<!-- EXAMPLE FOR MYSQL -->

<sonar.jdbc.url>

jdbc:mysql://localhost:3306/sonar

</sonar.jdbc.url>

<sonar.jdbc.driver>com.mysql.jdbc.Driver</sonar.jdbc.driver>

<sonar.jdbc.username>root</sonar.jdbc.username>

<sonar.jdbc.password>vidya</sonar.jdbc.password>

<!-- SERVER ON A REMOTE HOST -->

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

</properties>

</profile>

</profiles>

Note:

If you don’t want to create a new profile, you can directly add this property tag in your pom.xml

<properties>

<!-- EXAMPLE FOR MYSQL -->

<sonar.jdbc.url>

jdbc:mysql://localhost:3306/sonar

</sonar.jdbc.url>

<sonar.jdbc.driver>com.mysql.jdbc.Driver</sonar.jdbc.driver>

<sonar.jdbc.username>root</sonar.jdbc.username>

<sonar.jdbc.password>vidya</sonar.jdbc.password>

<!-- SERVER ON A REMOTE HOST -->

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

</properties>

## Analyzing a Maven Project for Sonar

* 1. Analyzing a Maven project consists of running a Maven goal in the directory where the pom.xml file sits. If possible, an install goal should be performed prior to the sonar:sonar one.
  2. Recommended Way :

|  |
| --- |
| mvn clean install -DskipTests=true  mvn sonar:sonar  Use skipTests=true, not to run unit tests twice: once during the install goal and  during the sonar:sonar goal. |

## Run Sonar as part of a build:

* 1. Sonar goals for maven 2 are added in POM.xml as follows :

<plugin>

<groupId>org.codehaus.mojo</groupId>

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

<version>1.0</version>

<executions>

<execution>

<phase>verify</phase>

<goals>

<goal>sonar</goal>

</goals>

</execution>

</executions>

</plugin>

* 1. Complete POM.xml is as follows for maven 2 :-

<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/maven-v4\_0\_0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.alm.ib</groupId>

<artifactId>my-app</artifactId>

<packaging>jar</packaging>

<version>1.0-SNAPSHOT</version>

<name>my-app</name>

<url>http://maven.apache.org</url>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.9</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.codehaus.mojo</groupId>

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

<version>1.0</version>

<executions>

<execution>

<phase>verify</phase>

<goals>

<goal>sonar</goal>

</goals>

</execution>

</executions>

</plugin>

</plugins>

<pluginManagement>

<plugins>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>2.3.2</version>

<configuration>

<jdk>1.6</jdk>

<source>1.6</source>

<target>1.6</target>

</configuration>

</plugin>

</plugins>

</pluginManagement>

</build>

</project>

* 1. Sonar goals for maven 3 are added in POM.xml as follows :

<plugin>

<groupId>org.codehaus.sonar</groupId>

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

<version>4.0</version>

<executions>

<execution>

<phase>verify</phase>

<goals>

<goal>sonar</goal>

</goals>

</execution>

</executions>

</plugin>

* 1. Complete POM.xml is as follows for maven 3 :-

<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/maven-v4\_0\_0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.alm.ib</groupId>

<artifactId>my-app</artifactId>

<packaging>jar</packaging>

<version>1.0-SNAPSHOT</version>

<name>my-app</name>

<url>http://maven.apache.org</url>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.9</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.codehaus.sonar</groupId>

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

<version>4.0</version>

<executions>

<execution>

<phase>verify</phase>

<goals>

<goal>sonar</goal>

</goals>

</execution>

</executions>

</plugin>

</plugins>

<pluginManagement>

<plugins>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>2.3.2</version>

<configuration>

<jdk>1.6</jdk>

<source>1.6</source>

<target>1.6</target>

</configuration>

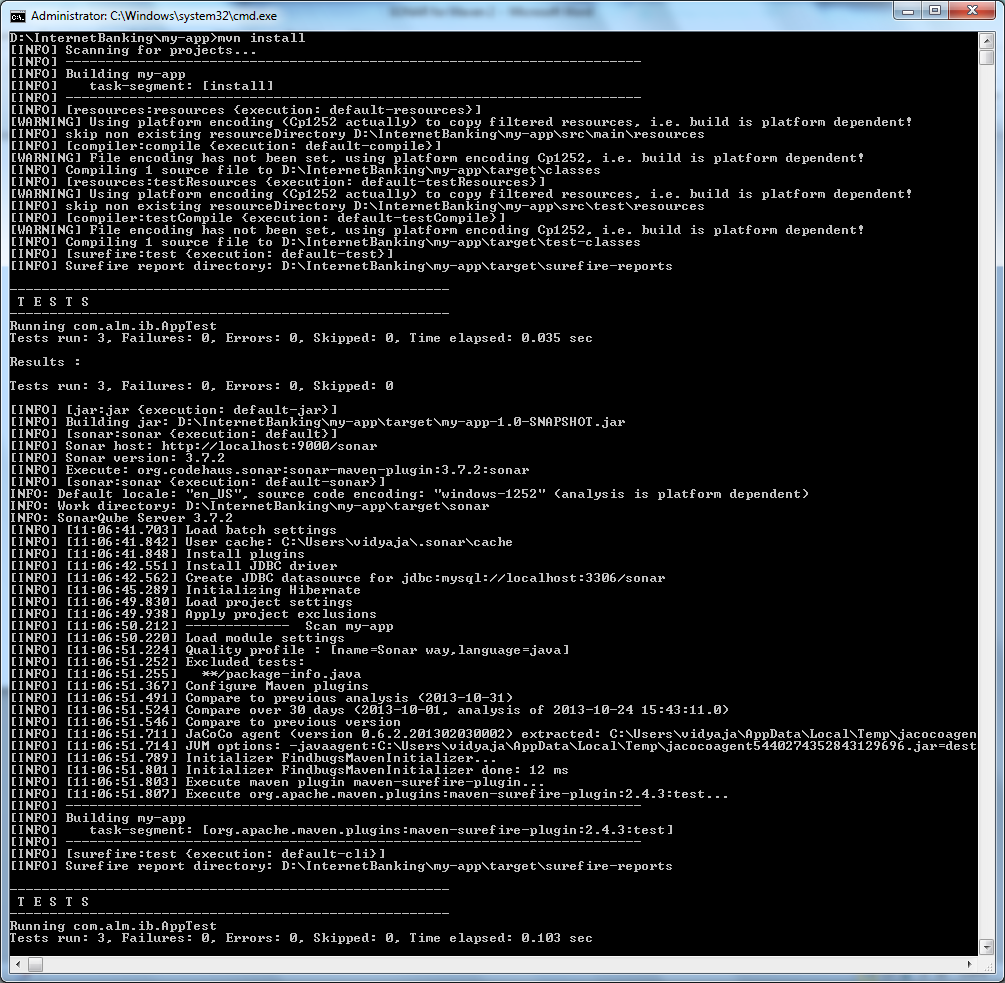
</plugin>

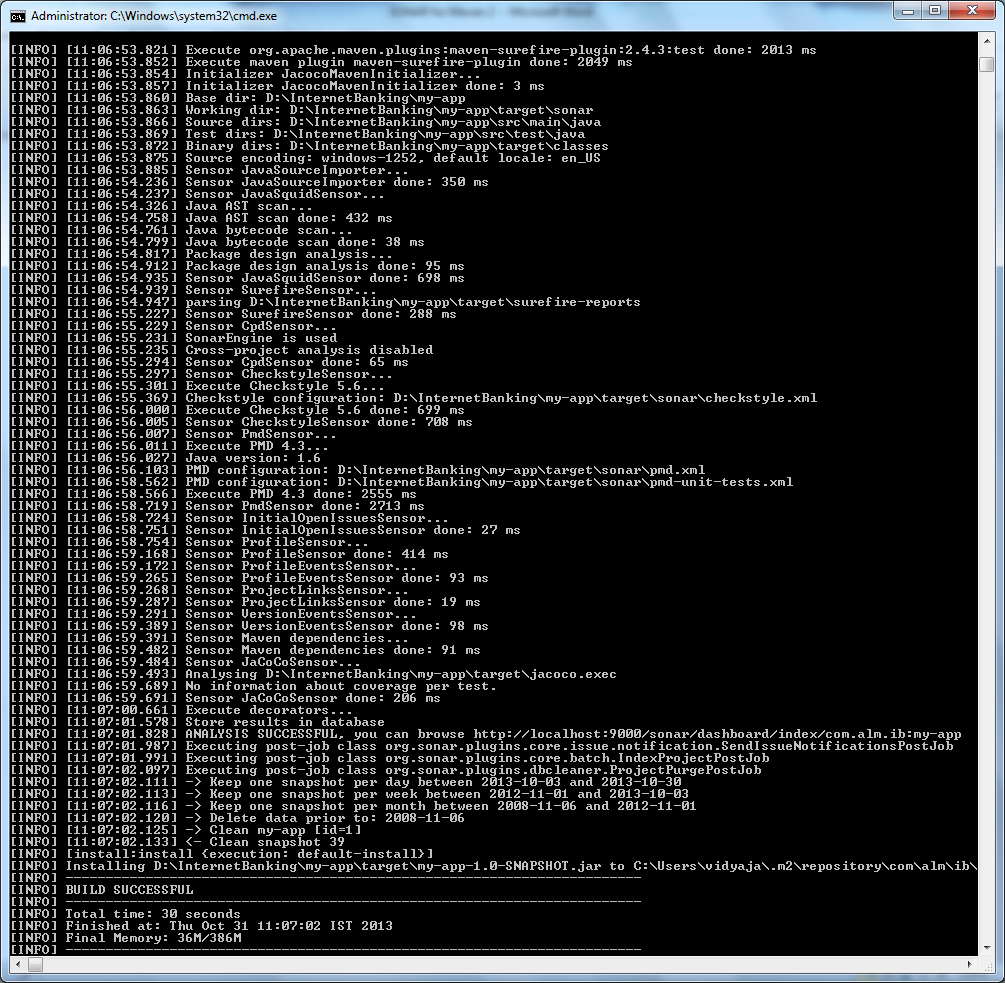
</plugins>

</pluginManagement>

</build>

</project>

1. Run command mvn:install to run Sonar local static code analysis as shown in below screenshot : 



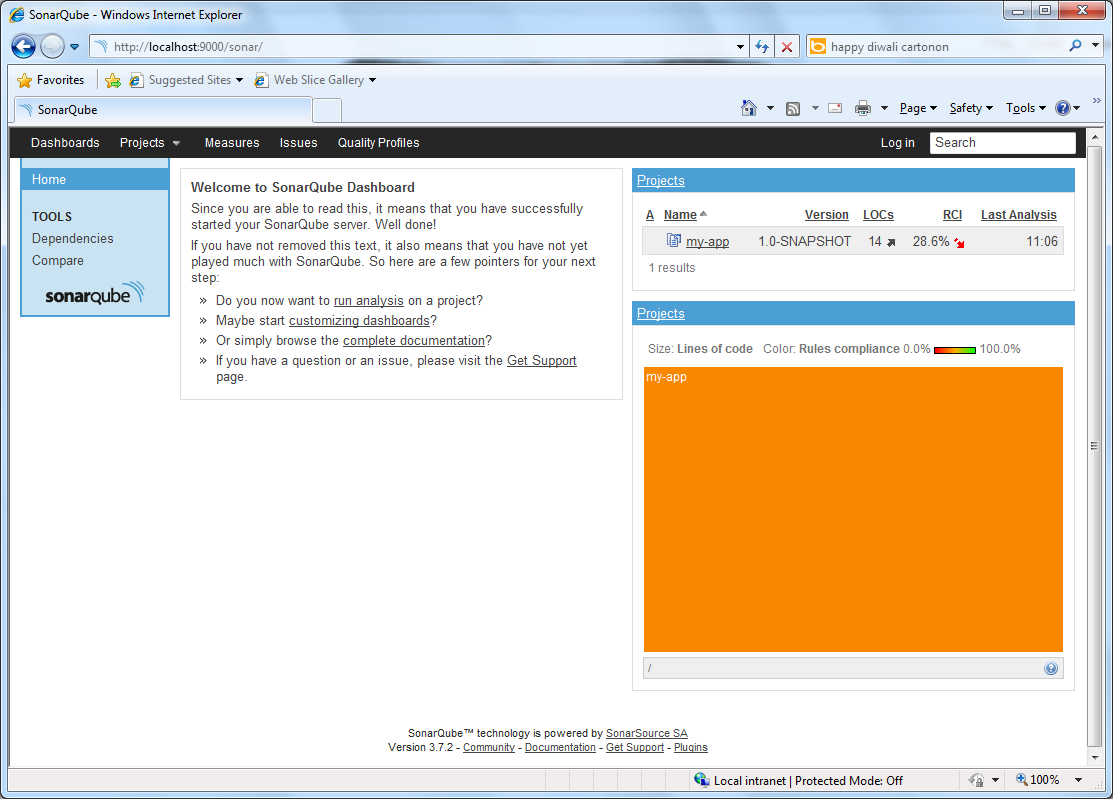
## Sonar Web Browser View:

The **Web** view displays the main project dashboard (for project) or the component viewer (for class or file).

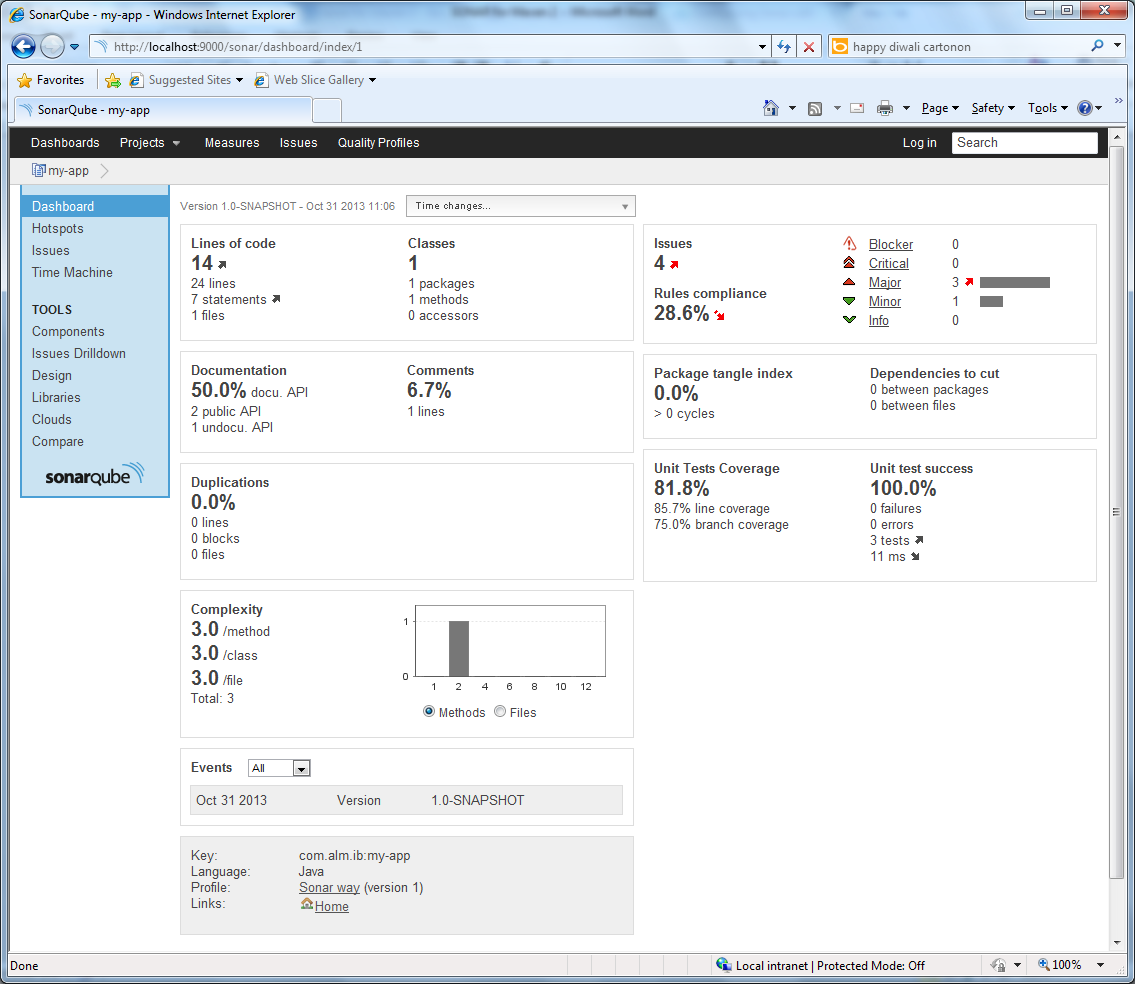
This web page can also be used to access information and services (Clouds, Hotspots, etc.) that are not available in SonarQube Eclipse.

As shown below Dashboard gives overall report for all issues divided into Blocker, Critical, Major, Minor, Info and also other parameters such as Unit tests Coverage, Rules compliance, Unit test Success, Lines of code, Duplications, Complexity etc

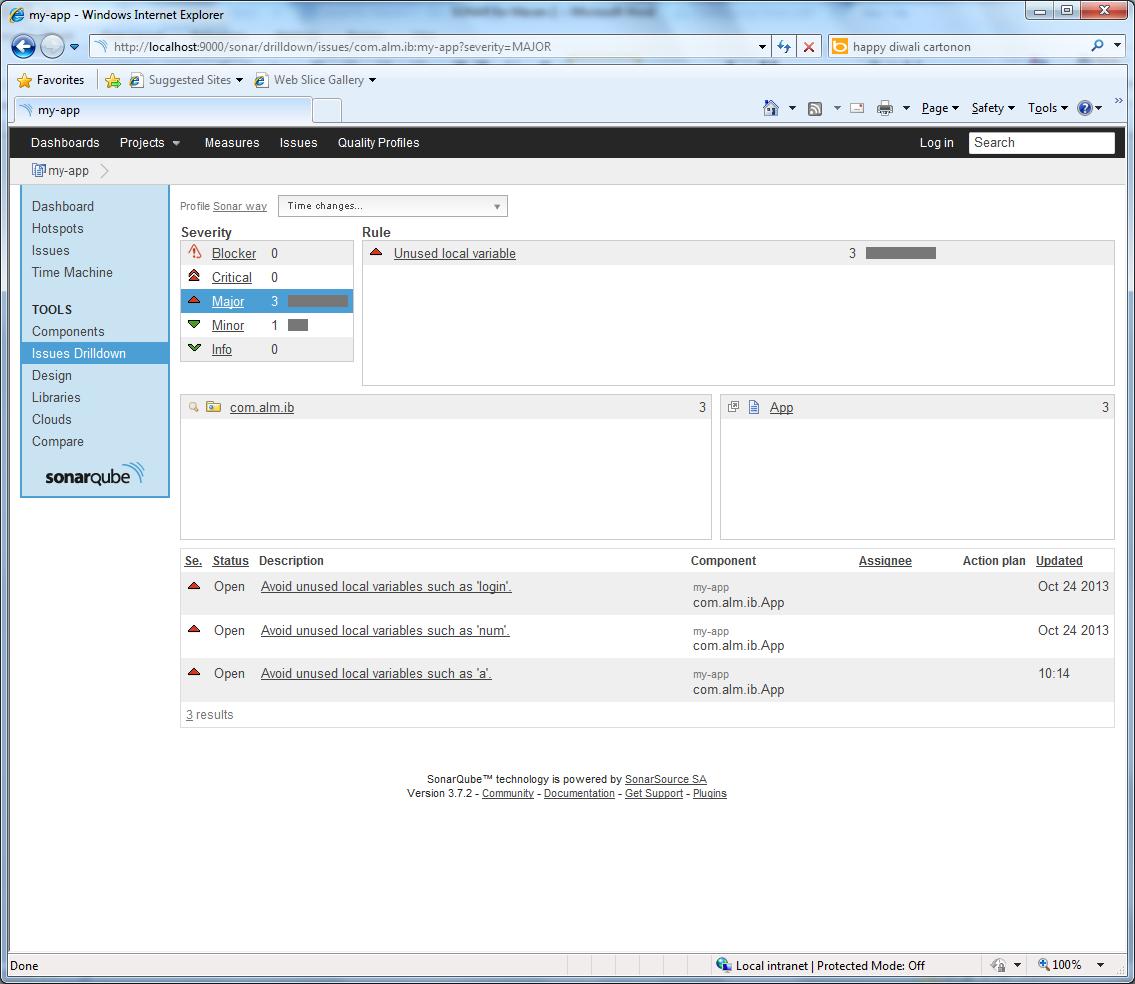
1. SonarQube dashboard



1. Analysis of a project



1. Checking minor issues

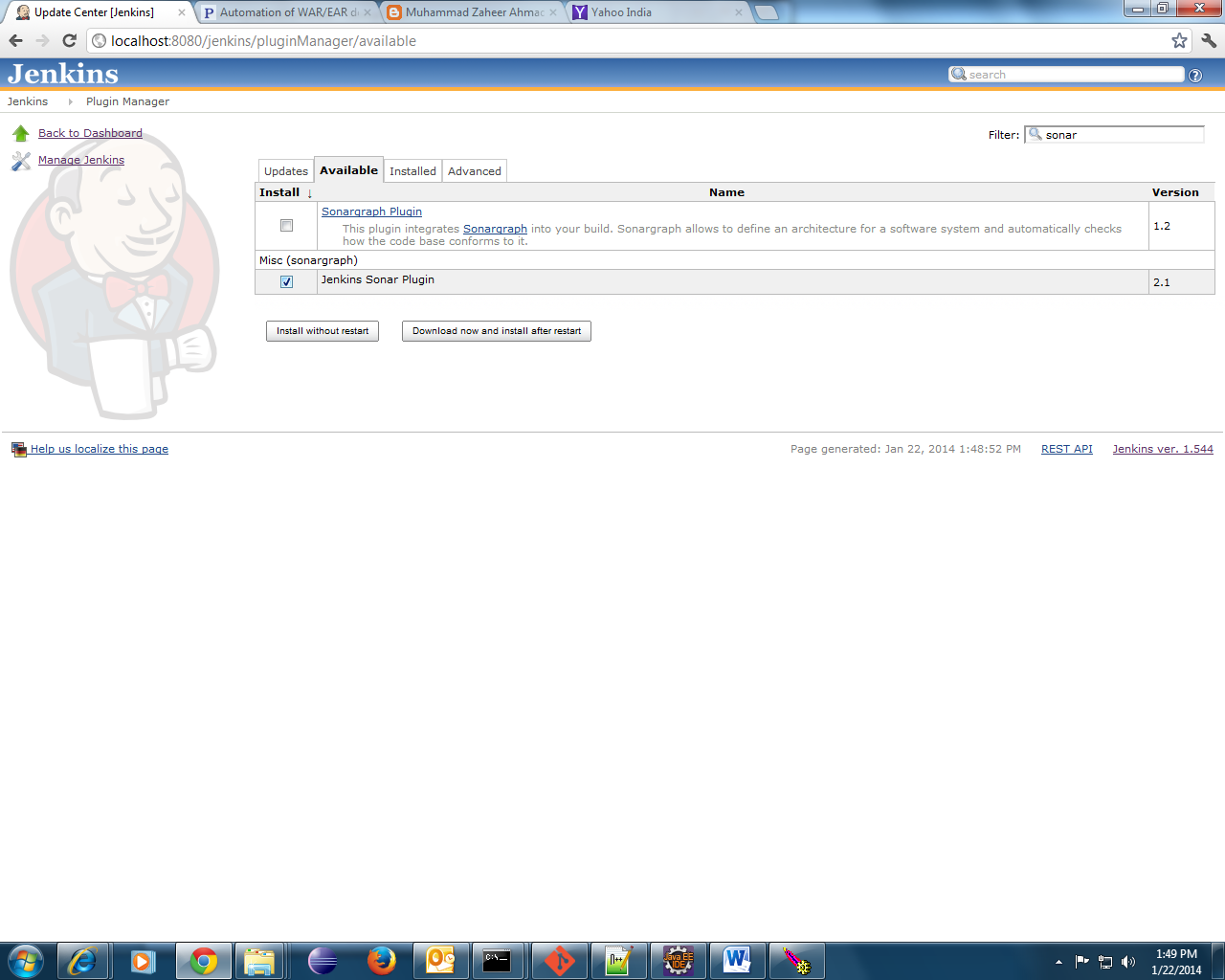


# Sonar only of Jenkins

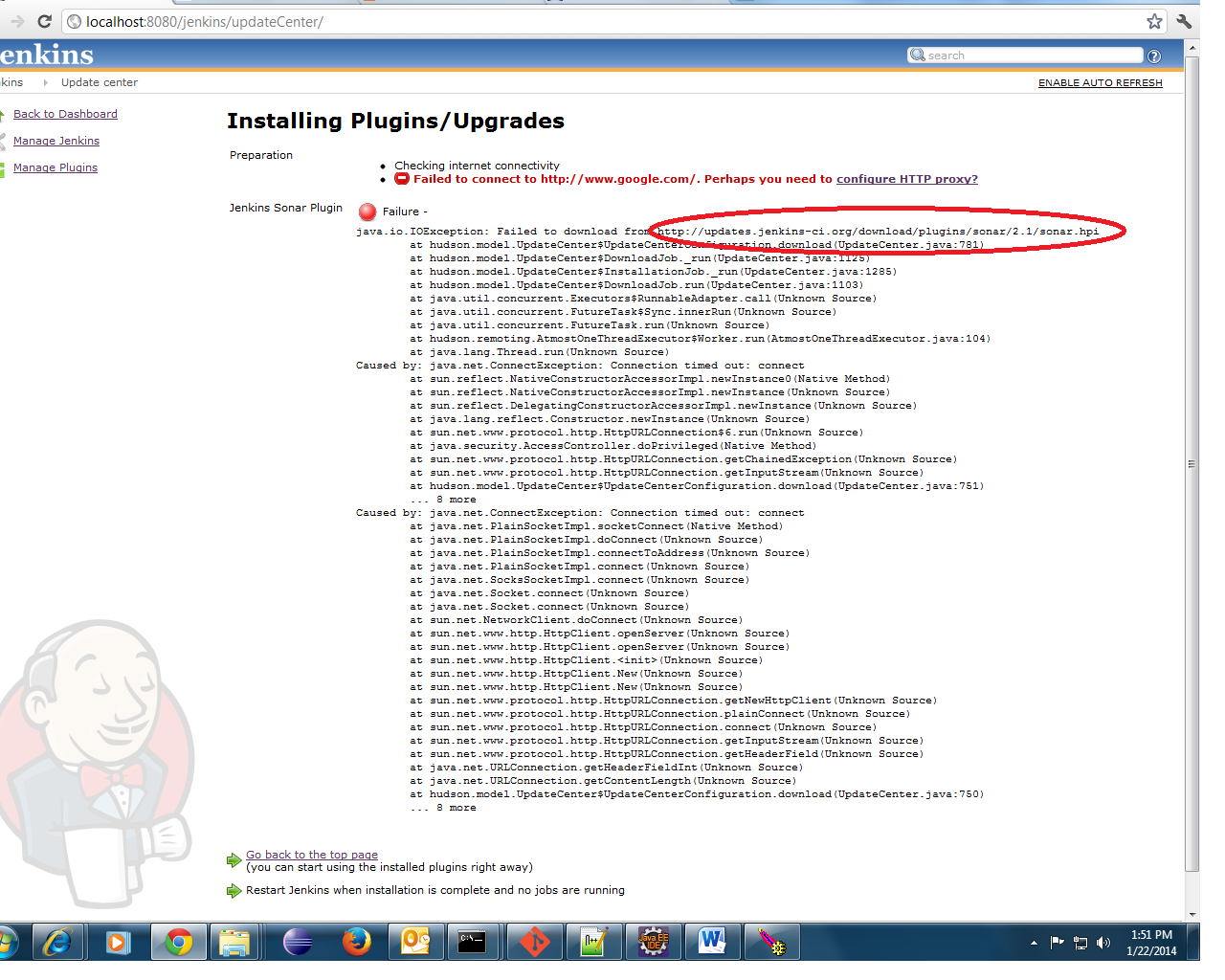
## Configuring the Sonar plugin

1. Install the sonar plugin (via Manage Jenkins -> Manage Plugins)

As shown in below screenshot:

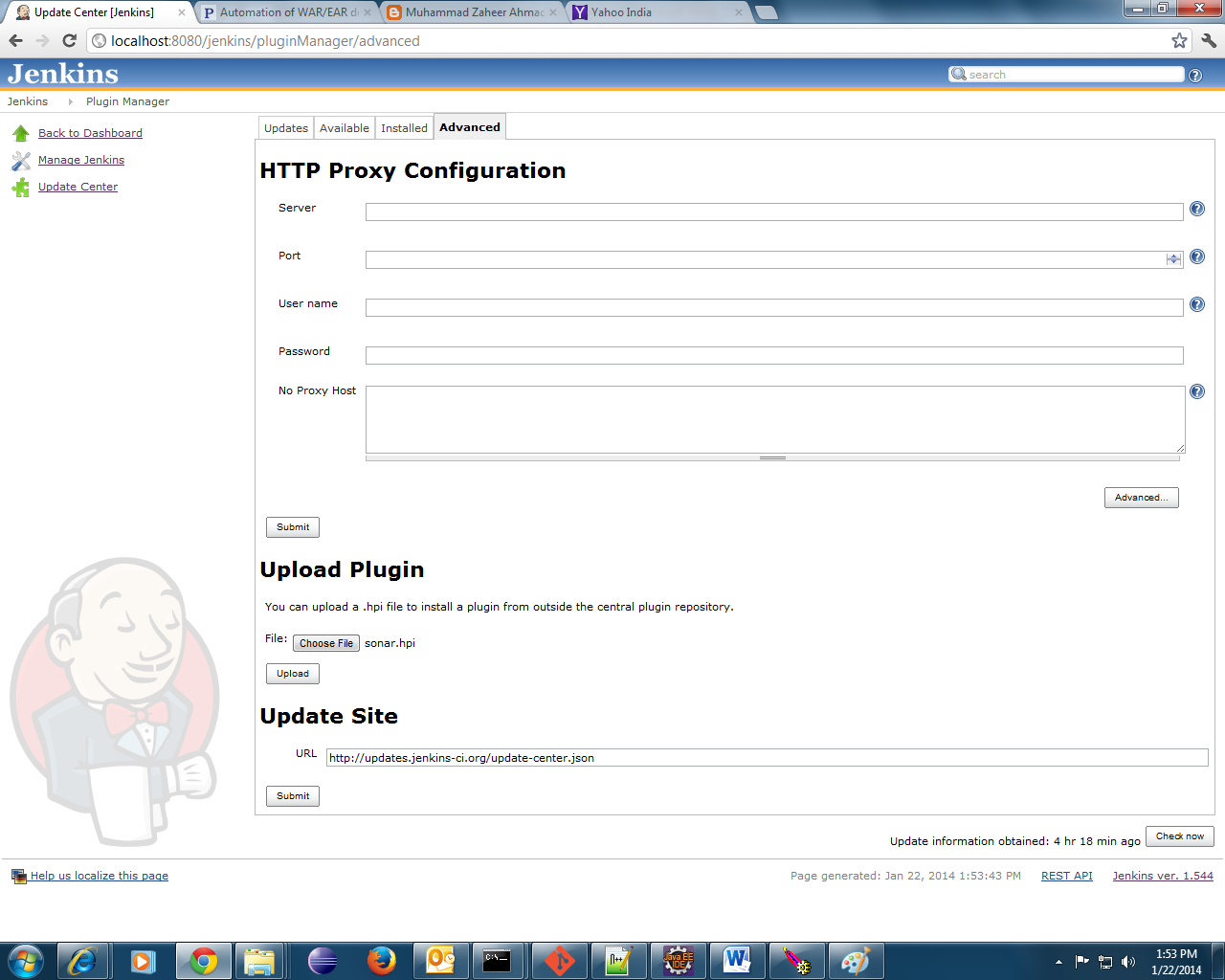


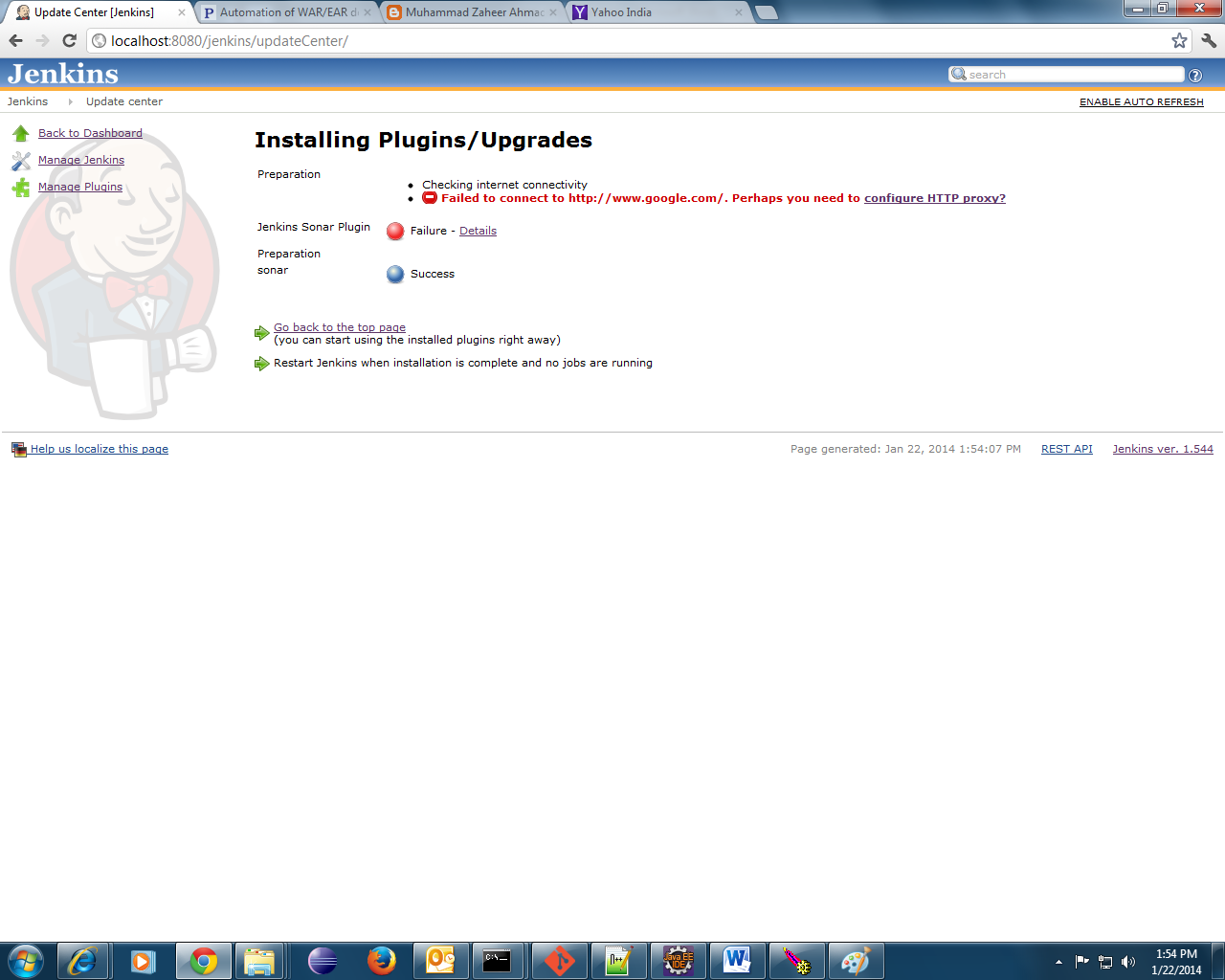
1. If it failed to download, manually copy the link which has been highlighted in below screenshot and download it.



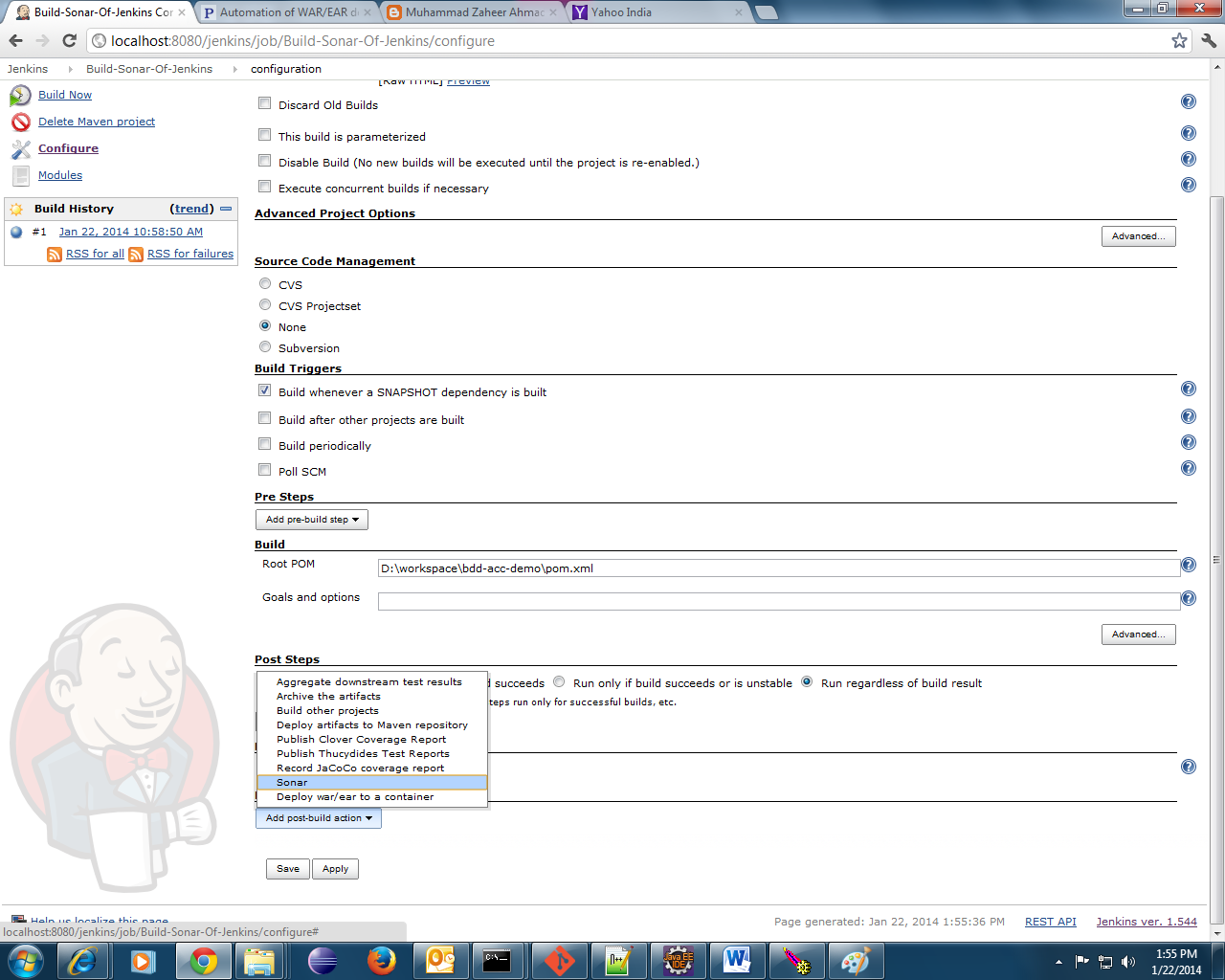
1. After download, click on configure HTTP proxy as show in above screenshot.

And upload the downloaded plugin as show below.

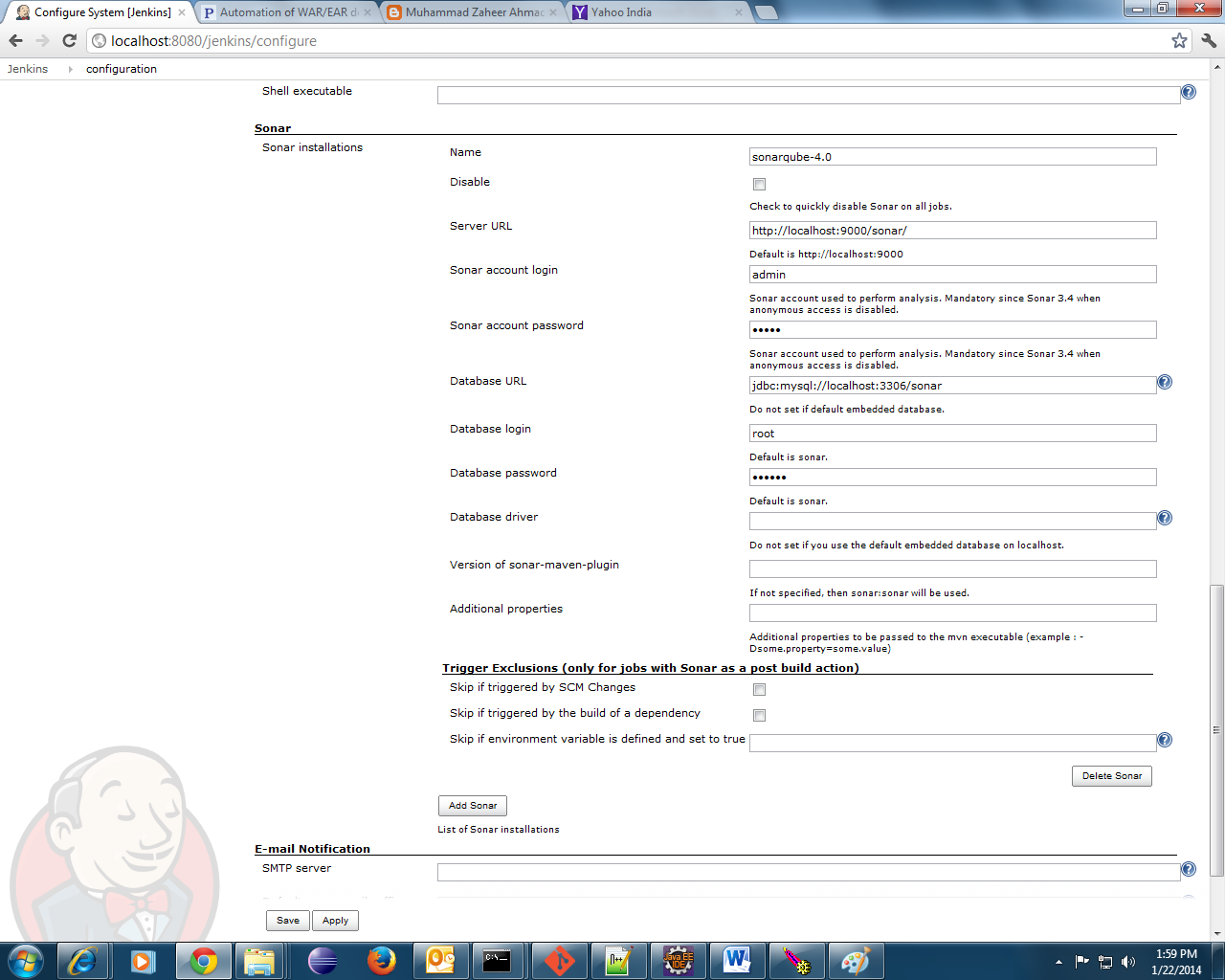




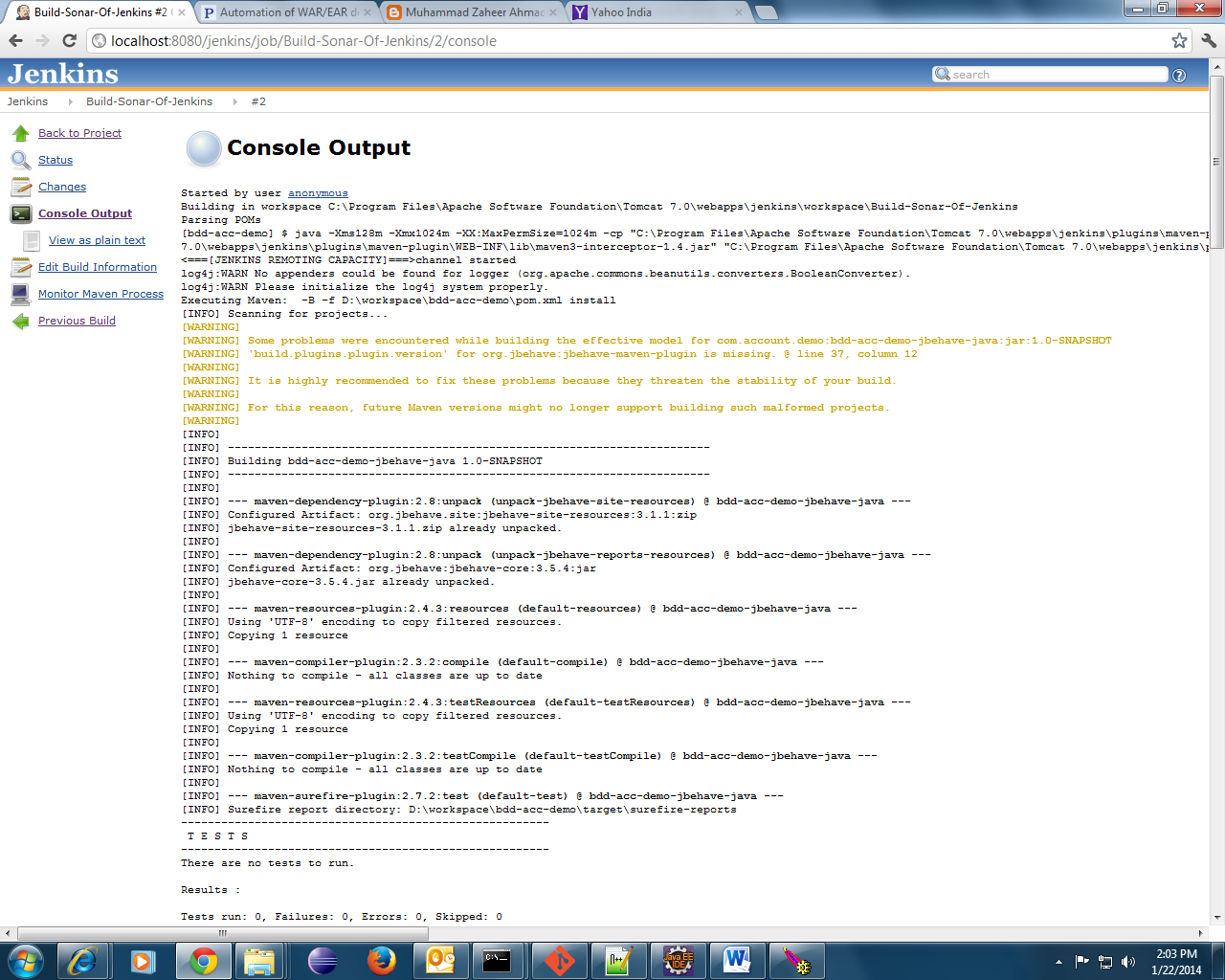
1. Go to Jenkins dashboard: then go to the project say ALM > Configure > Post build Actions and sonar and give pom.xml path and then apply and save changes as follows:

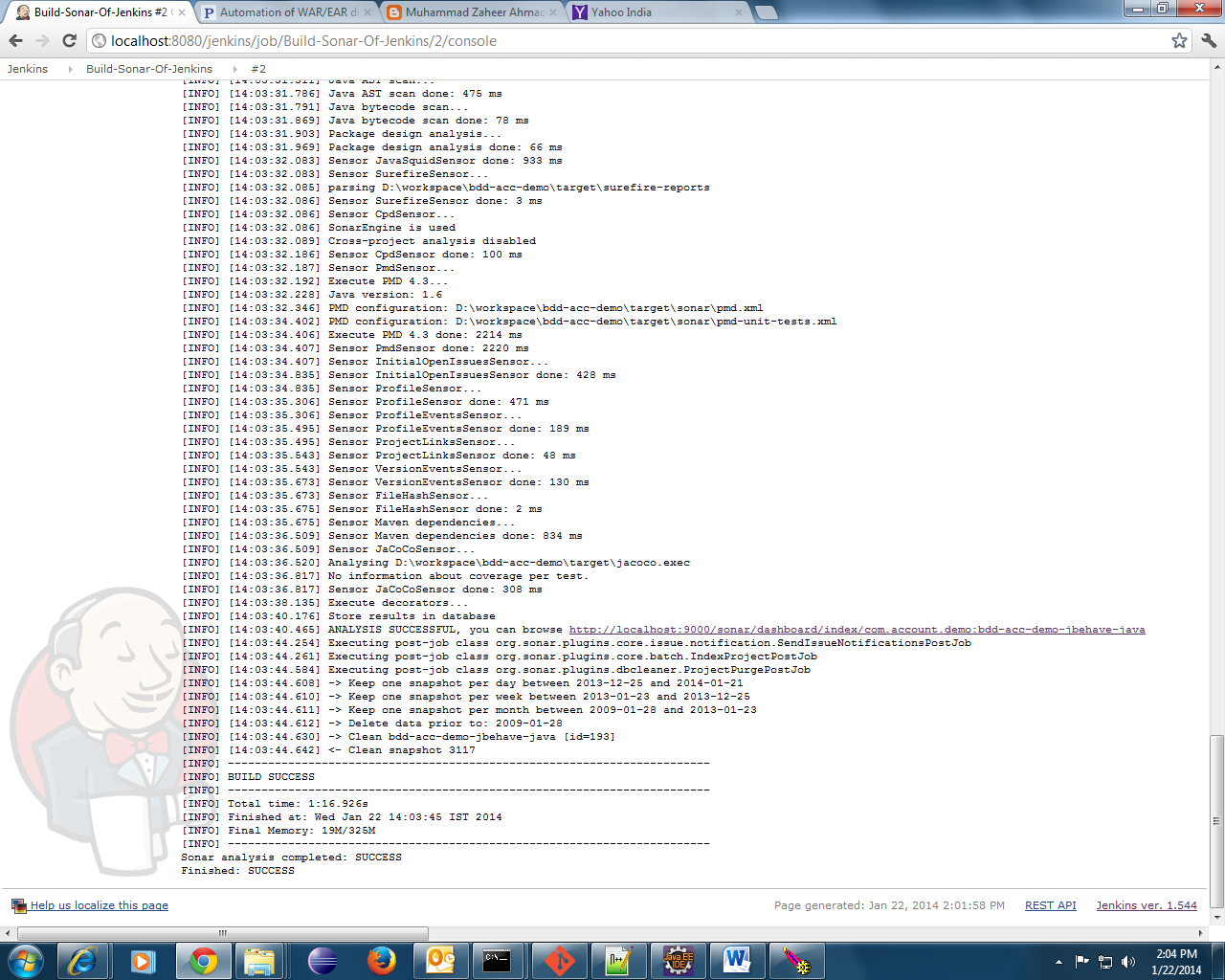


1. Add sonar setting via Jenkins->manage Jenkins->configure system



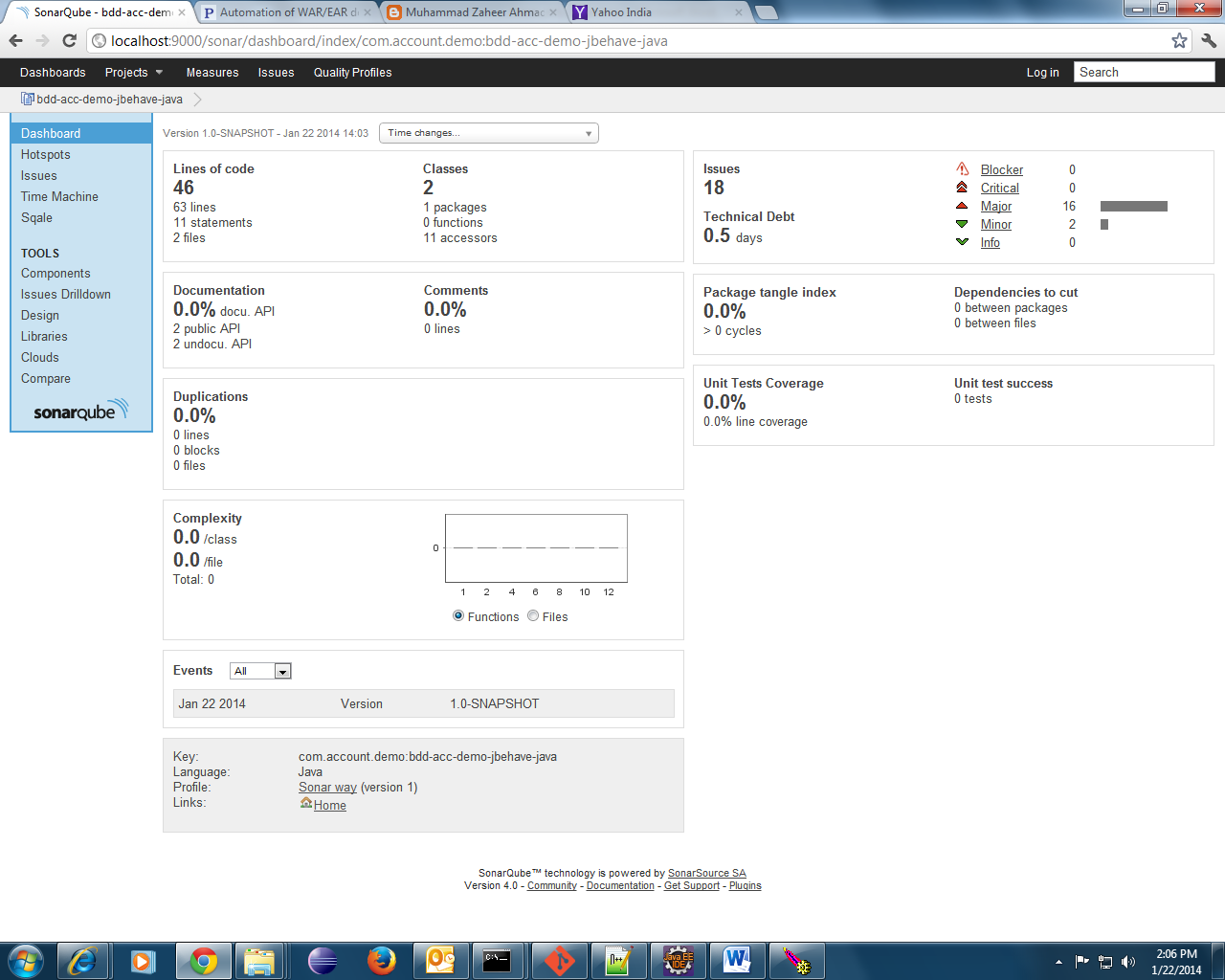
1. Run build by a click on build now and check console output as shown below :





1. Click on sonar and it be redirected to sonar dashboard as follows :





### References

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

[http://docs.sonarqube.org/display/Sonar/Requirements](http://docs.sonarqube.org/display/SONAR/Requirements)

<http://www.thedaytodayjava.com/java/sonar-simple-installation-on-linux/>

<https://weblogs.java.net/blog/johnsmart/archive/2009/06/installing_sona.html>

[http://docs.sonarqube.org/display/Sonar/Installing+SonarQube+in+Eclipse](http://docs.sonarqube.org/display/SONAR/Installing+SonarQube+in+Eclipse)

[http://docs.sonarqube.org/display/Sonar/Installing+SonarQube+in+IntelliJ](http://docs.sonarqube.org/display/SONAR/Installing+SonarQube+in+IntelliJ)

<http://docs.sonarqube.org>