# How to install Sonar for your Continuous

# Integration

**Version: draft**



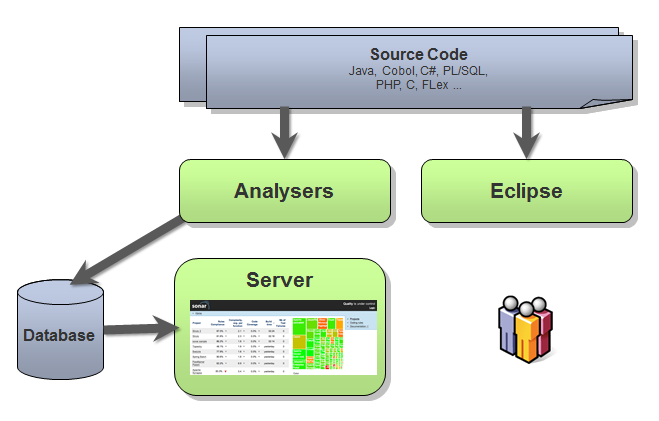
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date**  **(dd/mm/yyyy)** | **Prepared By** | **Reviewed by** | **Changes Done** |
| **draft** | **02/05/2015** | M Subramani |  | Initial Draft |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**objective**

Continuous Integration (**CI**) is a system which involves the tools that monitor your version control system for changes, build and test your application whenever a change is detected. What’s more it helps monitoring code quality and code coverage metrics for making sure that your code is healthy and strong. CI is an efficient tool for team members of a project, which it can simplify and accelerate delivery of production by helping you automate the deployment process.  
  
Here’s how to integrate with Sonar on your Continuous Integration platform for monitoring your code quality.

**Sonar Instalaltion**

To install Sonar, we should install 3 components of Sonar (a database, a sonar server and a sonar client). Here is an image of architecture of Sonar from Sonar doc

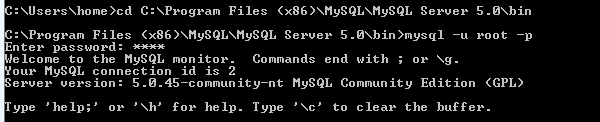


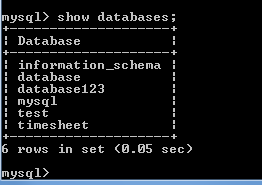
## Database

Sonar uses a Derby or H2 as default database. When running Sonar, it says that these databases may only be used for evaluation. You can use any other database you prefer.

I use MySQL as the database. It needs to create a new schema and a sonar user and give the user permissions to create, update and delete objects in the schema.

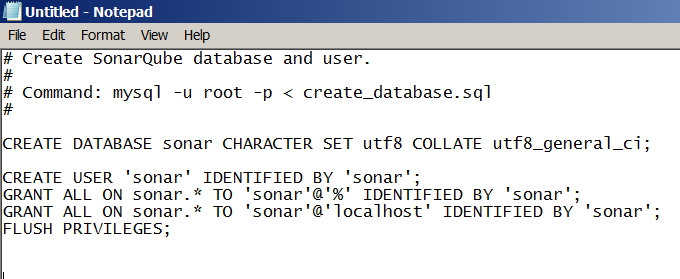
\*before create a schema make sure you’ve started MySQL server, if not use /etc/init.d/mysql start command to start the server)





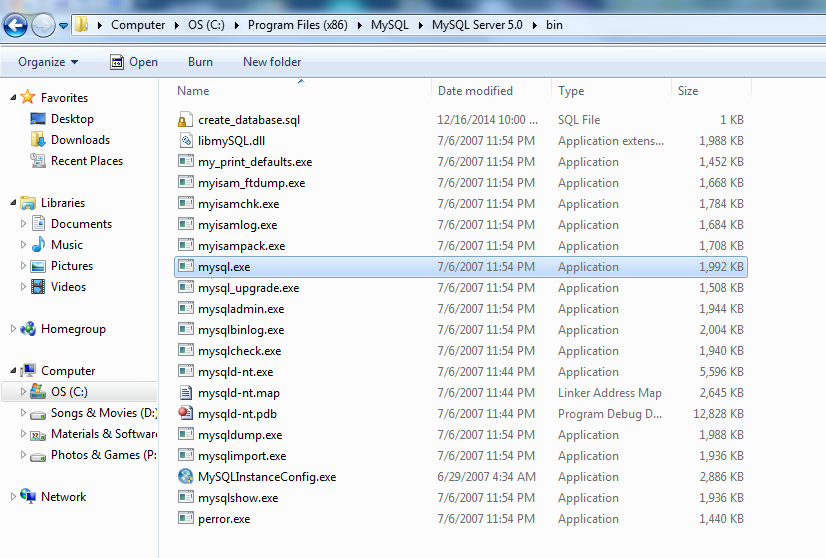
Use below script to create schema

|  |
| --- |
| # |
| # Command: mysql -u root -p < create\_database.sql |
| # |
|  |
| CREATE DATABASE sonar CHARACTER SET utf8 COLLATE utf8\_general\_ci; |
|  |
| CREATE USER 'sonar' IDENTIFIED BY 'sonar'; |
| GRANT ALL ON sonar.\* TO 'sonar'@'%' IDENTIFIED BY 'sonar'; |
| GRANT ALL ON sonar.\* TO 'sonar'@'localhost' IDENTIFIED BY 'sonar'; |
| FLUSH PRIVILEGES; |



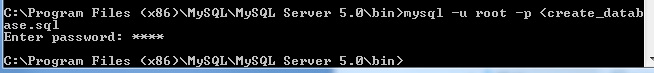
Save sample script under bin folder of C:\Program Files (x86)\MySQL\MySQL Server 5.0\bin. Remember to mention the filename inside the double quotes so that it will not take .txt extension

Now this file is added into directory

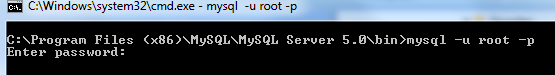


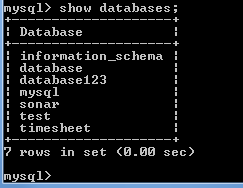
Go to command prompt and navigate to the directory where your MySQL file is installed and run the below command

mysql -u root -p <create\_database.sql



login to mysql and check whether the database is created





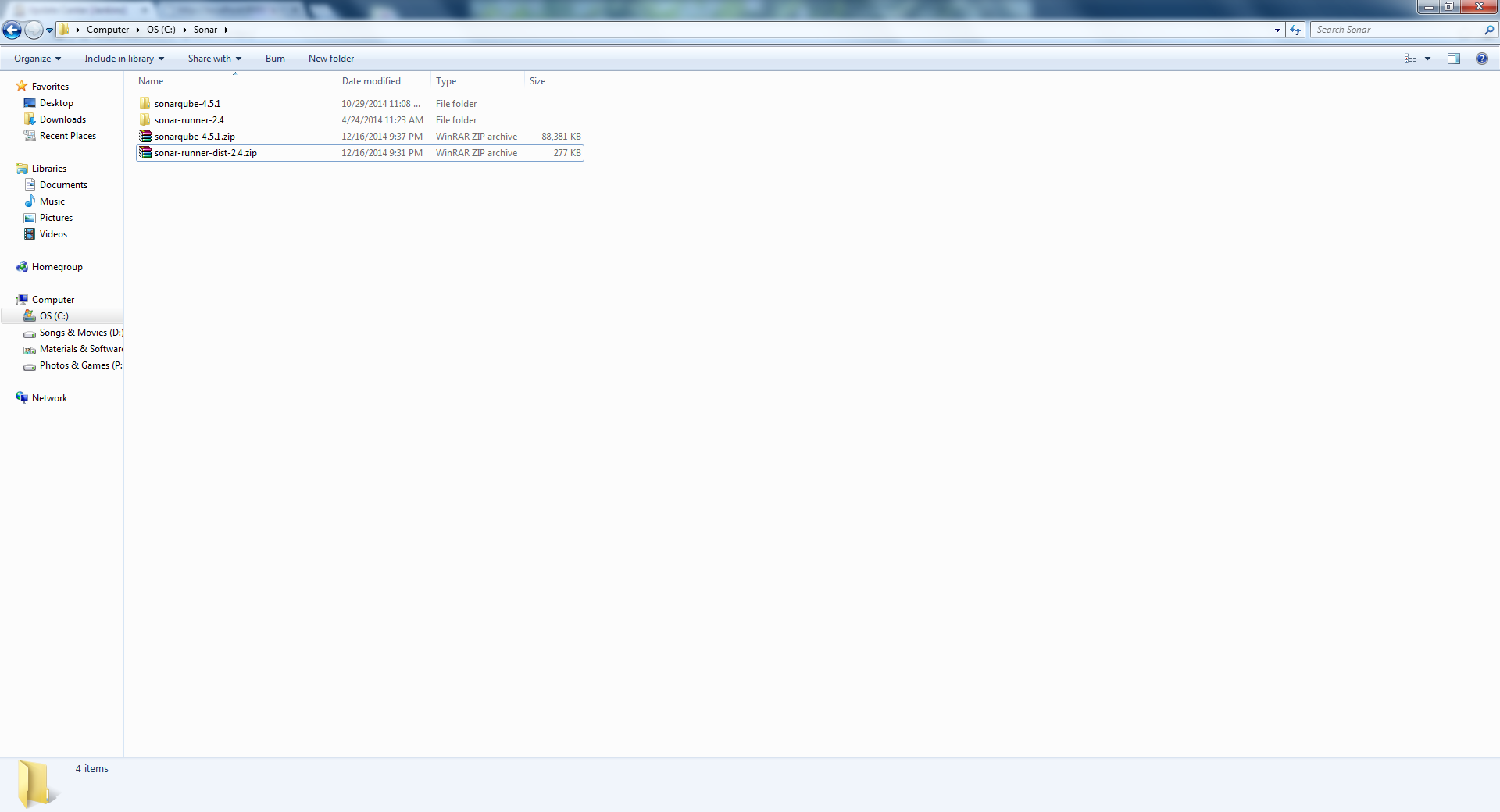
You can use below script to drop the database

|  |
| --- |
| # |
| # Command: mysql -u root -p < drop\_database.sql |
| # |
|  |
| DROP DATABASE IF EXISTS sonar; |
| DROP USER 'sonar'@'localhost'; |
| DROP USER 'sonar'@'%'; |

### Installation

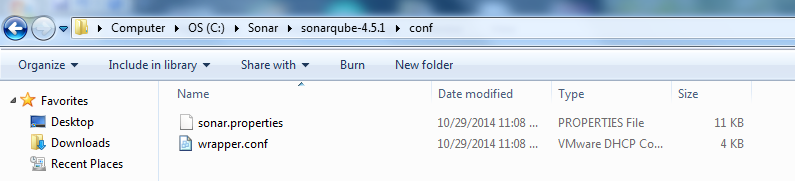
## Sonar server

First, download Sonar from its website at <http://www.sonarqube.org/downloads/>. Then, extract the zip file and the we can start configuring (Set SONAR\_HOME in environment variable and append the bin directory in path variable)

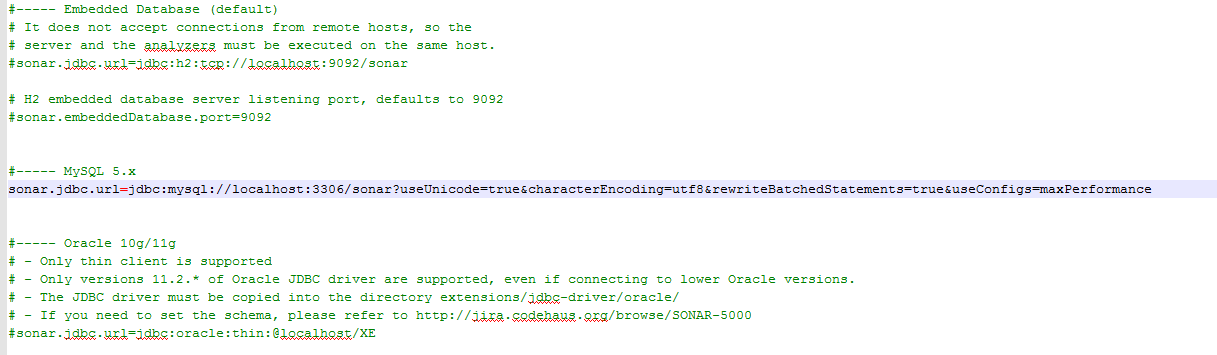


### Configuration

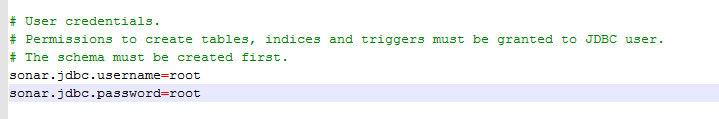
In the sonar folder, **go to the conf-folder and open sonar.properties**.



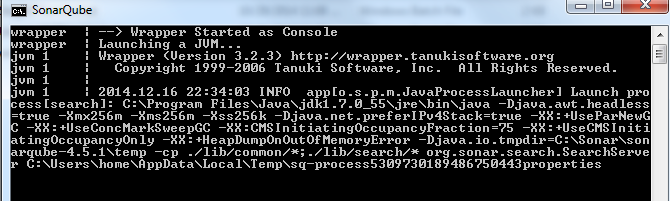
You will see many commented lines. Since we are using MySQL, we need to uncomment below portion



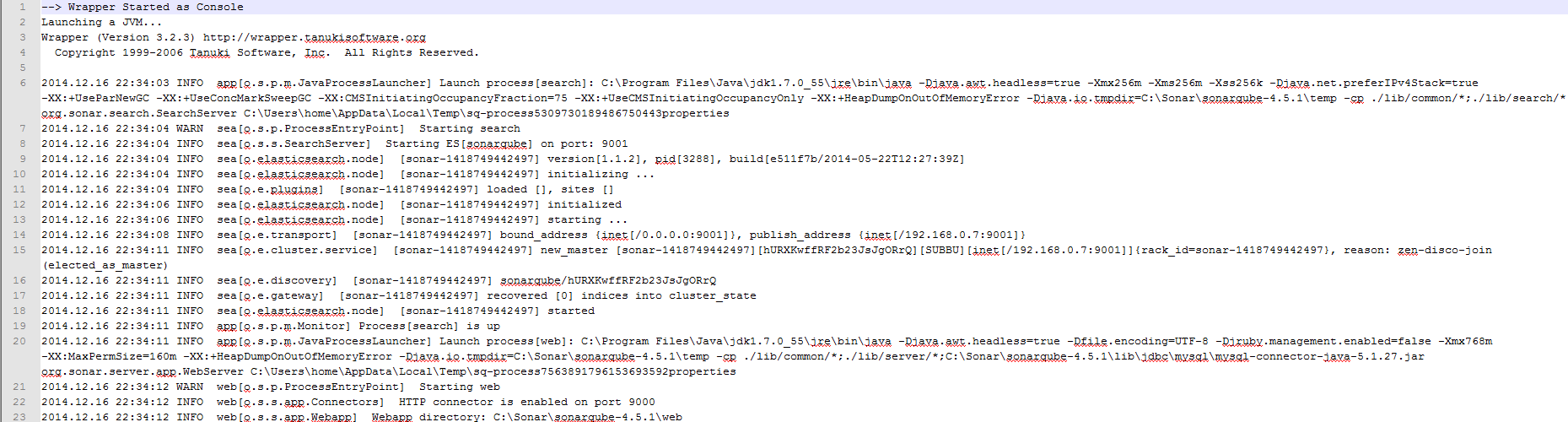
These are the login credentials to my database. For MySQL, it will be:



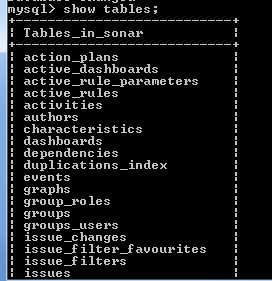
When that’s finished, we can start Sonar, but **before we do that, make sure your database is running and that the Sonar database is created!**. To start sonar, go to the bin-folder, select your operating system folder and click on StartSonar.xxx. For me, it will be windows-x86-64 and StartSonar.bat. After starting up, a command box will appear, create the needed tales in your database and start up. To see if Sonar is up and running, got to <http://localhost:9000>

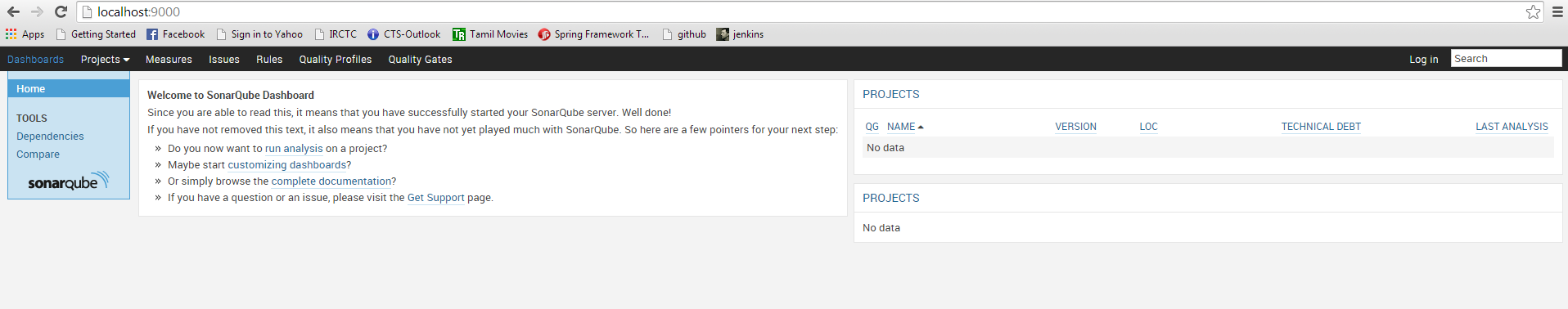


You can check the logs under C:\Sonar\sonarqube-4.5.1\logs



You can check the list of tables created in MYSQL database





## Sonar client

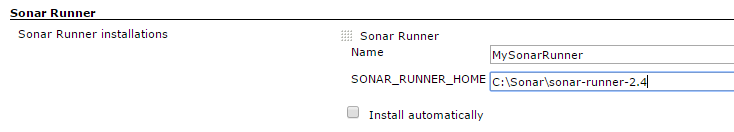
There are different clients available like CI Engine, Sonar runner, Maven, etc. We use CI Engine (Jenkins) to launch analyze of code quality with Jenkins sonar plugin which enables to trigger Sonar analysis from Jenkins by two ways:

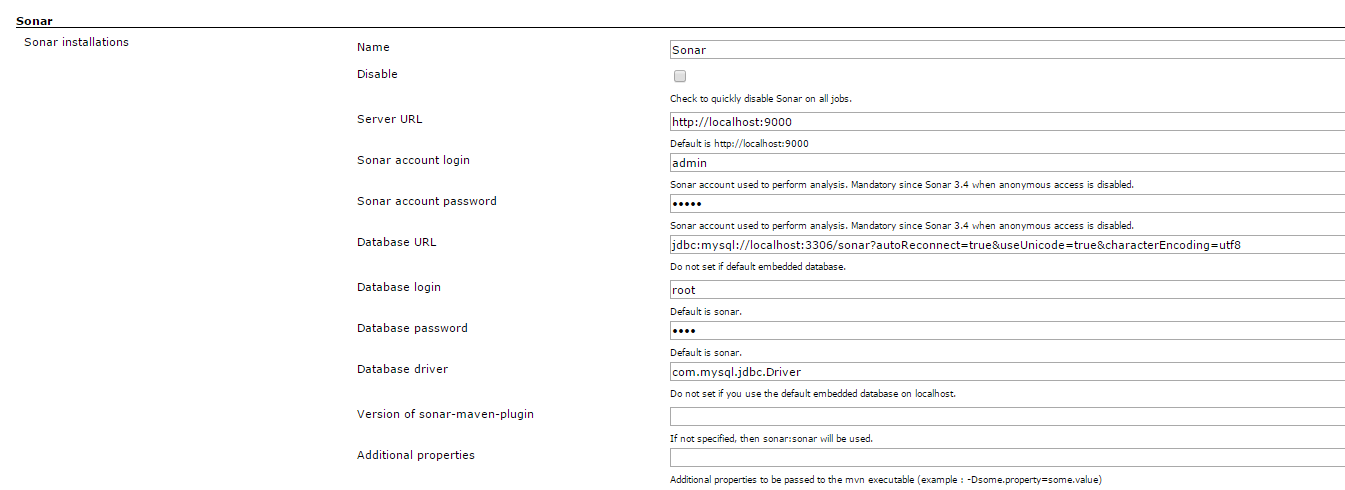
* Build step to trigger the analysis with the Sonar Runner (which I use)
* Post-build action to trigger the analysis with Maven

So now it’s time to install Sonar Runner.

# Install Sonar plugin for Jenkins and configure Jenkins

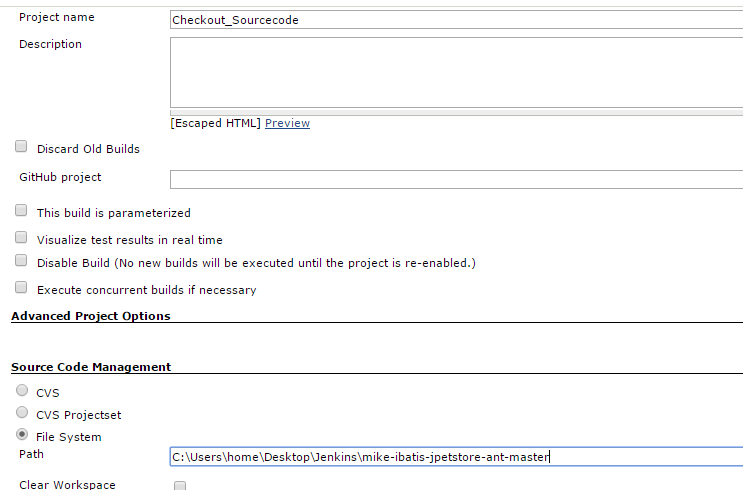
You can download Sonar plugin directly from your CI website and install it. Now go to Jenkins Admin -> System configuration, configure Sonar and Sonar Runner:

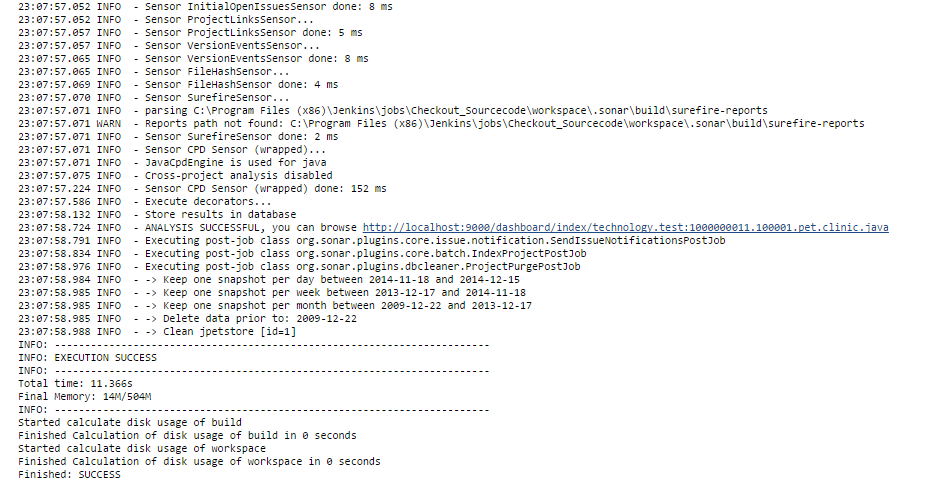




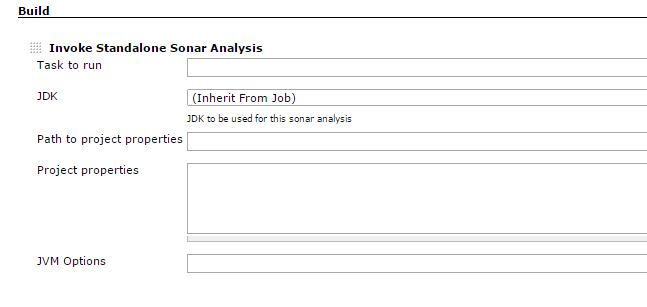
# Create a project (job) for analyzing code quality

Before analyze of code quality we need to checkout our code from repository. For simplicity reason am talking code from local file system

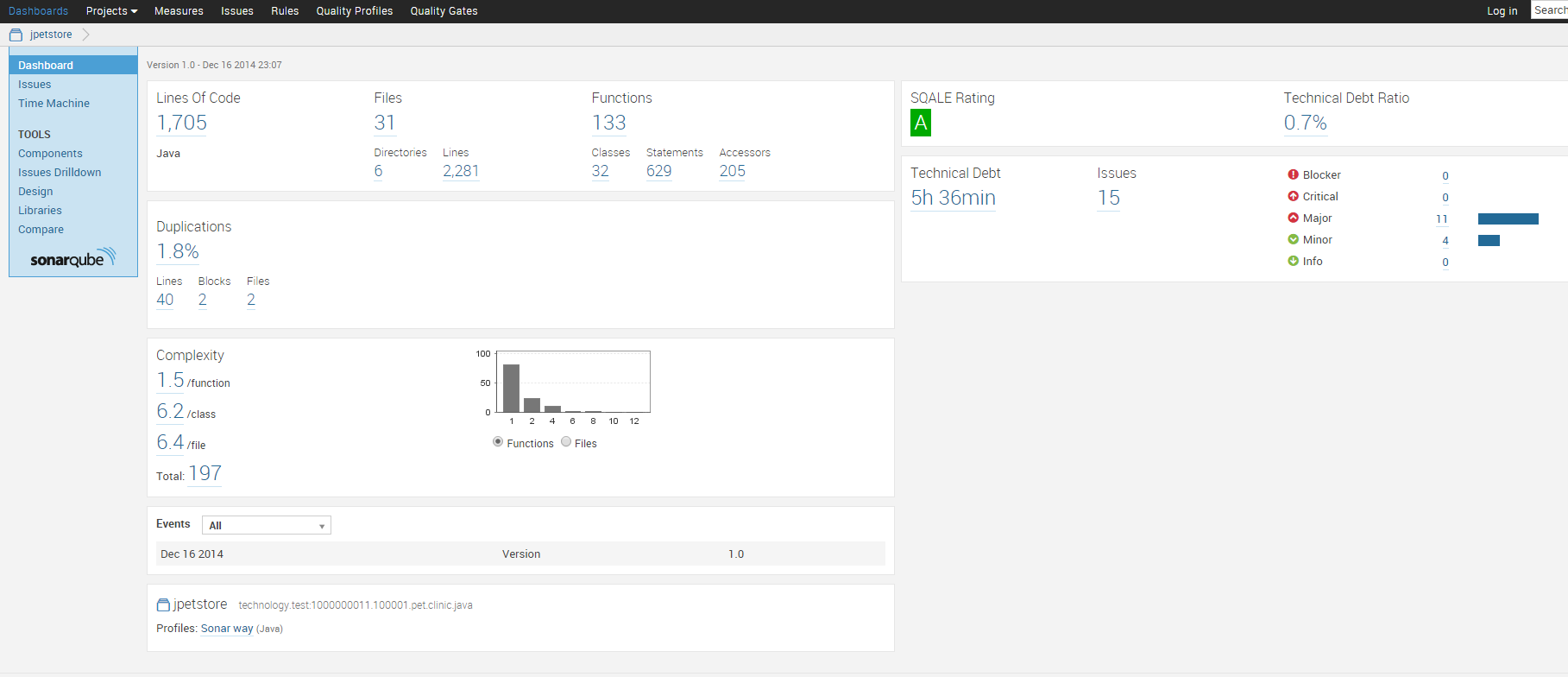




Here’s the configure of build step to launch Sonar:



Sonar Analysis Result



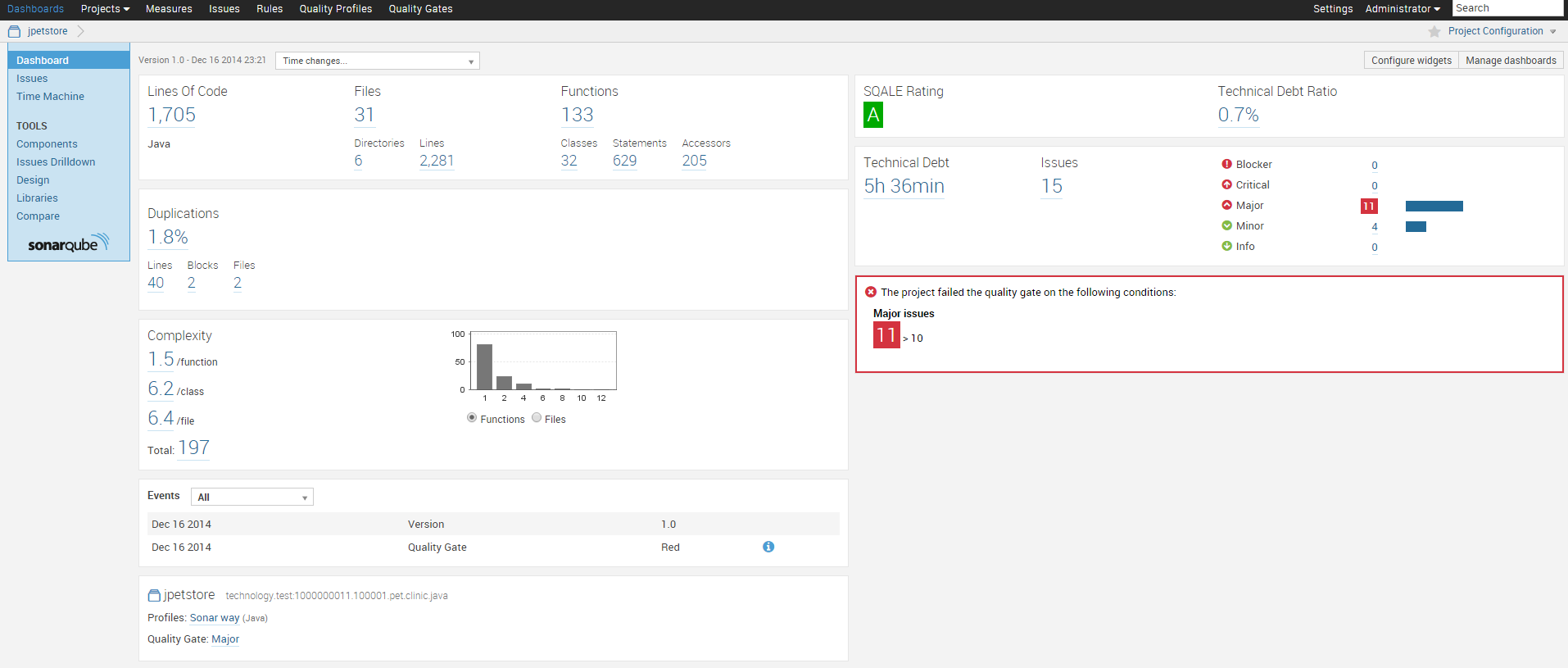
From now on you can launch a build for your project which will first checkout code from repository and then monitor the code quality.

# [Build Breaker Plugin](http://docs.codehaus.org/display/SONAR/Build+Breaker+Plugin)

This plugin will mark the build failed if the project fails its [quality gate](http://docs.codehaus.org/display/SONAR/Build+Breaker+Plugin).

Installation

1. Install the plugin through the [Update Center](http://docs.codehaus.org/display/SONAR/Update+Center) or download it into the SONARQUBE\_HOME/extensions/plugins directory
2. Restart the SonarQube server



### Eclipse plugin

Now sonar is up and running, we need to configure Eclipse and Maven to use sonar. First, install the Eclipse plugin for Sonar. You can find it at <http://dist.sonar-ide.codehaus.org/eclipse/>. After installing the plugin, we need a little configuration in Maven to use the plugin.

### Maven configuration

When you run Sonar in Maven without a configuration, Maven will still use the default database. We need to add a profile in the Maven settings. In the .m2 folder, open the settings.xml. Then add the code below. You also need to do that in the settings.xml in your Maven conf folder. You can find other database profiles at <http://mojo.codehaus.org/sonar-maven-plugin/examples/use-enterprise-database.html>

### Update Maven in Eclipse

Now sonar and Maven are configured, we need to update your project in Eclipse. Right click on your project in Eclipse, go to Maven and click on **Update Project**. Then the Maven settings will become active.

### Running Sonar in Eclipse

Now we are ready to run Sonar in Eclipse. Right click on your root project, select **run as** and click on **run configurations**. Create a new Maven build and insert into your goal: **clean install sonar: sonar**. Then click on run and your project will show up in Sonar. Enjoy!

