**Integration of SonarQube with Jenkins for PHP Projects**

**What is SonarQube?**

* SonarQube is an Open source quality management platform, dedicated to continuously analyze and measure source code quality.
* Supports [25+ languages](http://docs.codehaus.org/display/SONAR/Plugin+Library): [Java](http://en.wikipedia.org/wiki/Java_(programming_language)), [C](http://en.wikipedia.org/wiki/C_(programming_language))/[C++](http://en.wikipedia.org/wiki/C%2B%2B), [Objective-C](http://en.wikipedia.org/wiki/Objective-C), [C#](http://en.wikipedia.org/wiki/C_Sharp_(programming_language)), [PHP](http://en.wikipedia.org/wiki/PHP), [Flex](http://en.wikipedia.org/wiki/Flex_(language)), [Groovy](http://en.wikipedia.org/wiki/Groovy_(programming_language)), [JavaScript](http://en.wikipedia.org/wiki/JavaScript), [Python](http://en.wikipedia.org/wiki/Python_(programming_language)), [PL/SQL](http://en.wikipedia.org/wiki/PL/SQL),[COBOL](http://en.wikipedia.org/wiki/COBOL), etc.
* Offers reports on [duplicated code](http://en.wikipedia.org/wiki/Duplicate_code), [coding standards](http://en.wikipedia.org/wiki/Programming_style), [unit tests](http://en.wikipedia.org/wiki/Unit_testing), [code coverage](http://en.wikipedia.org/wiki/Code_coverage), [complex code](http://en.wikipedia.org/wiki/Cyclomatic_complexity), bugs, comments and [design and architecture](http://en.wikipedia.org/wiki/Software_design).
* Integrates with external tools: [JIRA](http://en.wikipedia.org/wiki/JIRA), [Mantis](http://en.wikipedia.org/wiki/Mantis_Bug_Tracker), [LDAP](http://en.wikipedia.org/wiki/LDAP), [Fortify](http://en.wikipedia.org/wiki/Fortify_Software), etc.

**Why do we need Sonar if Jenkins is already setup with all the static code analysis reports?**

* Jenkins is great for running builds and it can be setup in a way that it generates useful reports about the source code, but Sonar has been specifically designed for this. It has a great user interface and gives so much better statistics.
* Easily see what direction the project is going and how much technical debt you are accumulating.
* Drill down to different parts of the project and compare them, you can see how the code base has changed over time and compare different versions etc.
* Useful with large projects that have a long lifespan so you can better keep track of code quality.
* Take action on the issues Sonar has found by assigning them to people.

After Integration of **Jenkins-PHP** implementation, to Integrated Jenkins with SonarQube we have to follow the following steps.

**Installing SonarQube**

First, install SonarQube using yum by running these commands:

*$ sudo wget -O /etc/yum.repos.d/sonar.repo* [*http://downloads.sourceforge.net/project/sonar-pkg/rpm/sonar.repo*](http://downloads.sourceforge.net/project/sonar-pkg/rpm/sonar.repo)

*$ yum install sonar*

This documentation focuses on CentOS installation; you can find [packages for most popular Linux distributions](http://sonar-pkg.sourceforge.net/).

**Edit the Sonar Config. file**

Location: /opt/sonar/conf/sonar.properties

Change DB to MySQL. You can of course use whatever database you like, but I had MySQL already installed on the machine so it was an obvious choice. Just comment out the line about the embedded H2 database (which is not intended for production use) and uncomment the MySQL line, save and exit.

**Create Database for Sonar**

Create an empty database called sonar and create a user named sonar that has access to the database.

*CREATE DATABASE sonar CHARACTER SET utf8 COLLATE utf8\_general\_ci;*

*CREATE USER 'sonar' IDENTIFIED BY 'sonar';*

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

*flush privileges;*

**Note**: If you have a machine with some domain name then you have to add that too in the grant privileges on sonar database.

**Change Default Port of Sonar if in Use**

Change the default port of Sonar since I have other services already using port 9000.

Change **Sonar.web.port = 9080** in /opt/sonar/conf/sonar.properties.

Change Sonar.web.host =your ip address.

Uncomment the database credentials.

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

# The schema must be created first.

**sonar.jdbc.username=sonar**

**sonar.jdbc.password=sonar**

After all the settings go ahead and restart sonar server:

Sudo Service sonar restart

If everything went fine, SonarQube should be running and we can point a browser to: http://serverAddress:9000 (or http://serverAddress:9090 in my case).

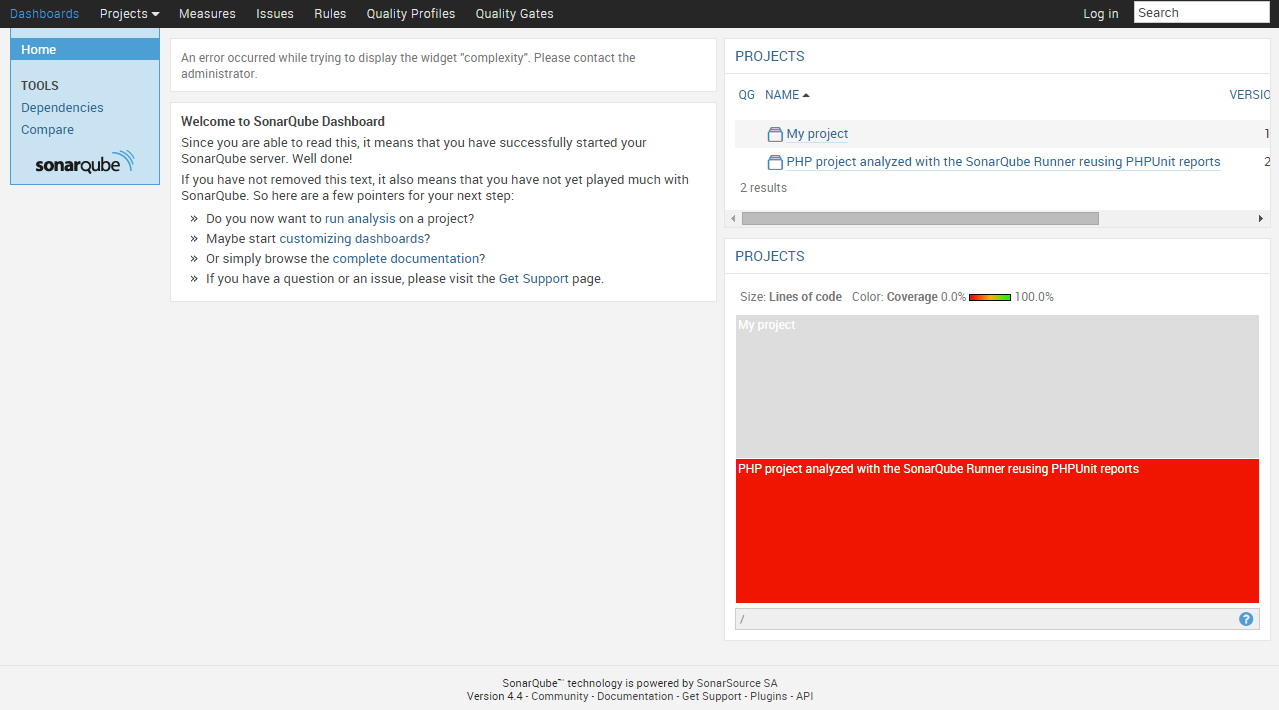


Fig: showing dashboard of SonarQube.

**Installing SonarQube Runner**

First download and uncompressed the file (check the above link for the latest version) and create a symbolic link to this specific version by executing the following commands:

cd /opt

sudo wget <http://repo1.maven.org/maven2/org/codehaus/sonar/runner/sonar-runner-dist/2.4/sonar-runner-dist-2.4.zip>

sudo unzip sonar-runner-dist-2.4.zip

sudo ln -s sonar-runner-2.4 sonar-runner

Next we need to edit the configuration file and update database information and server URL.

Location:  /opt/sonar-runner/conf/sonar-runner.properties

Enter following details

1. Sonar.host.url
2. Uncomment the line for MySQL as the database.
3. Uncomment database username and password.
4. Also you have to uncomment and update the login details if you selected the “Force user authentication” setting in the previous section.

**Create environment variable**

Next we need to create a new environment variable called SONAR\_RUNNER\_HOME and add the sonar runner bin directory to the PATH environment variable

sudo echo -e '#!/bin/bash\nexport SONAR\_RUNNER\_HOME=/opt/sonar-runner\nexport PATH=$PATH:$SONAR\_RUNNER\_HOME/bin' > /etc/profile.d/sonar-runner.sh

And if you don’t want to log out and back in again for this to take effect, you can run these commands to add the environment variables for the current session:

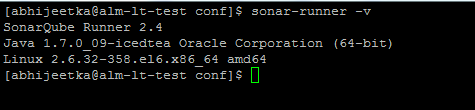
*export SONAR\_RUNNER\_HOME=/opt/sonar-runner*

*export PATH=$PATH:$SONAR\_RUNNER\_HOME/bin*

Now the runner has been installed, and you can test it by executing this command:

*Sonar-runner -v*

**Output :**



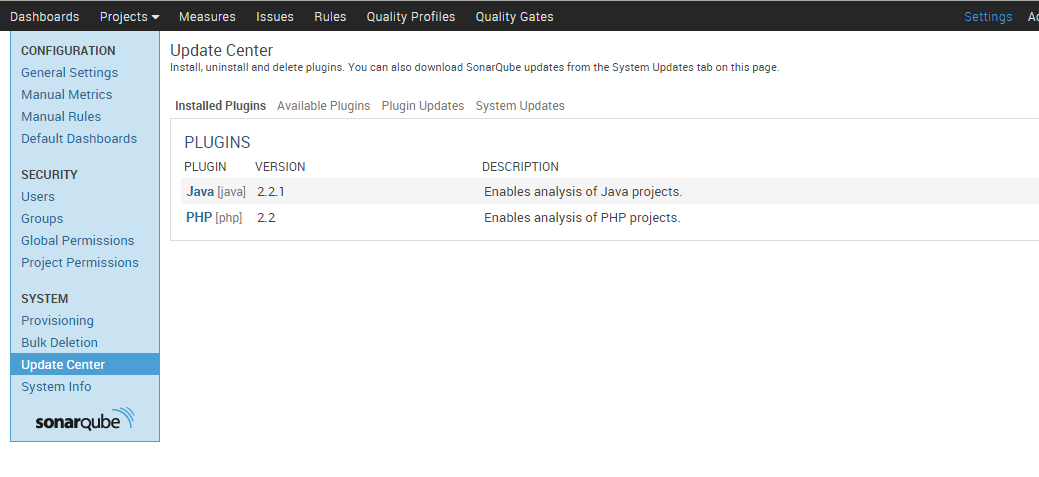
## Installing PHP environment for SonarQube

SonarQube relies on some external tools for PHP analysis, so we need to make sure that the following software’s should be installed on our system:

* PHP Depend,
* PHPMD,
* PHP\_CodeSniffer
* PHPUnit
* xDebug

Next log into SonarQube and head to “Settings” > “System” > “Update Center”. Click “Available Plugins” and click “PHP” in the list of available languages and then click Install.

Restart sonar server.



## Integrating SonarQube with Jenkins

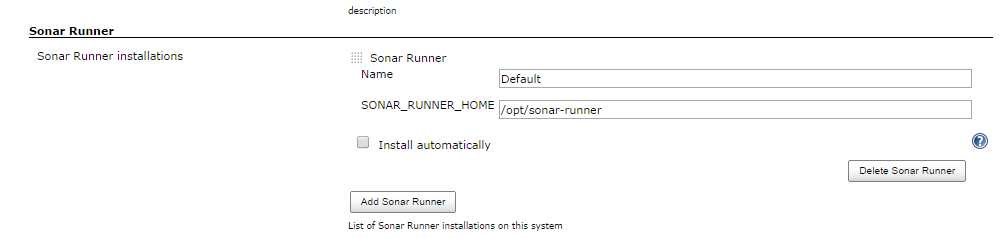
**Install the SonarQube Jenkins Plugin.**

Log into Jenkins as administrator and go to “Manage Jenkins” > “Manage Plugins”, click the “Available” tab and type “sonar” to the filter input in the top right, select Jenkins Sonar Plugin and install it and restart Jenkins.

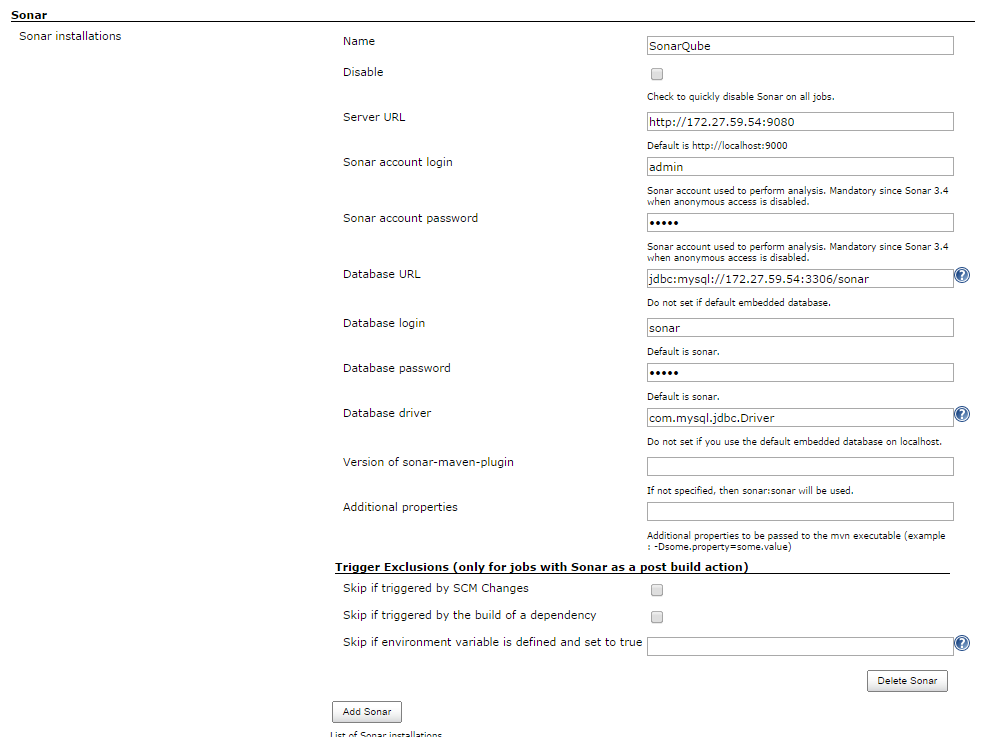
**Tell Jenkins where our Sonar installation is**

* “Manage Jenkins” > “Configure System” and find the “Sonar Runner” section and click “Add Sonar Runner”.
* Specify sonar runner name( I called mine “Default”).
* Point it to sonar runner home directory(/opt/sonar-runner).
* Find the “Sonar” section and click “Add Sonar”.
* Give your Sonar installation a name (I called mine “SonarQube”), click “Advanced” and fill in your server URL, login account and database details. Finally hit Save.

**Sonar runner**



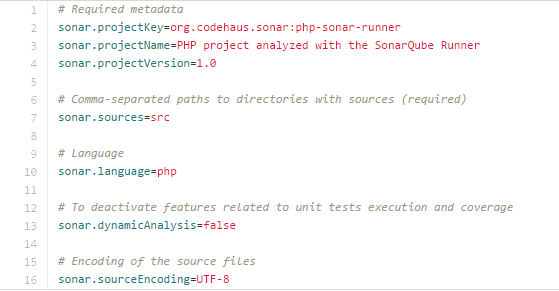
**Sonar**



**Create Sonar project properties**

Next create a file called sonar-project.properties in the root directory the project you want to analyze with SonarQube.

This file configures sonar-runner.



**Add build step to run Sonar analysis**

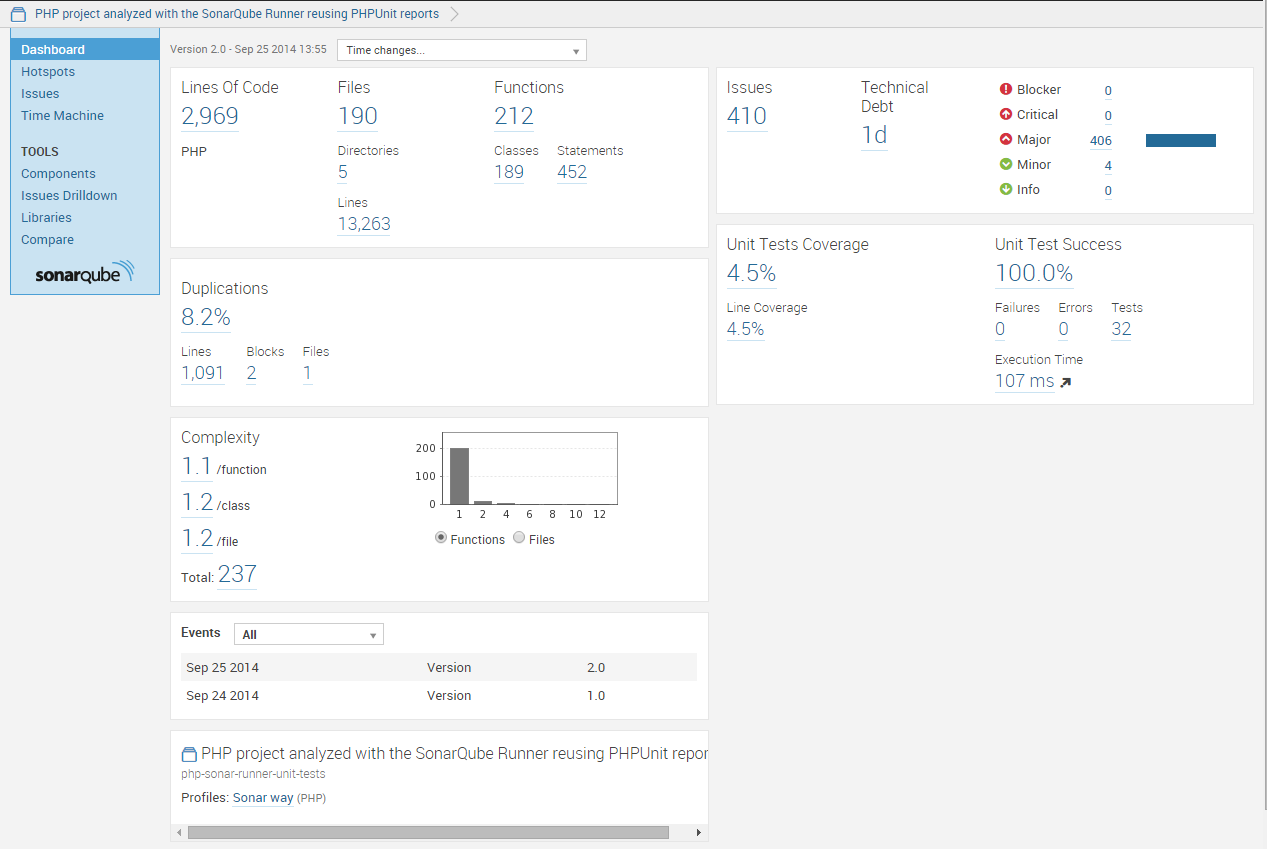
Open the project in Jenkins and click “Configure” on the left sidebar.

Scroll down to the “Build” section and click “Add build step” and select “Invoke Standalone Sonar Analysis”.



**Note**: If you used the default name and location for the sonar-project.properties file, you will not have to specify it’s path.

Now everything is set up and you can try it out by running a build.



Jenkins will have a link for SonarQube dashboard for full report.

