



Hãy nói theo cách của bạn

**TRUNG TÂM GIẢI PHÁP CÔNG NGHỆ THÔNG TIN VÀ VIỄN
THÔNG VIETTEL - VTICT**

**TÀI LIỆU HƯỚNG DẪN CÀI ĐẶT
VÀ PHÂN TÍCH MÃ NGUỒN SỬ DỤNG
SONARQUBE**

Hà Nội, 10/2105

BẢNG GHI NHẬN THAY ĐỔI

*A – Tạo mới, M – Sửa đổi, D – Xóa bỏ

MỤC LỤC

1. GIỚI THIỆU	4
1.1. Mục đích và ý nghĩa của Tài liệu	4
1.2. Phạm vi tài liệu	4
1.3. Các thuật ngữ và từ viết tắt	4
2. HƯỚNG DẪN CÀI ĐẶT SONARQUBE	5
3. QUÉT MÃ NGUỒN JAVA PROJECT VỚI SONAR-RUNNER	7
4. QUÉT MÃ NGUỒN MAVEN PROJECT	8
5. SỬ DỤNG SONAR ĐỂ QUÉT VỚI .NET PROJECT	9
5.1. Install C# plugin	9
5.2. Cấu hình MSBuild SonarQube Runner và thực hiện quét với .NET project	9
6. HƯỚNG DẪN SỬ DỤNG VÀ TÙY CHỈNH CÁC RULES	12
6.1. Một số màn hình chung	12
6.2. Add thêm bộ rule	13
6.3. Thêm bớt/xóa bớt một số rule không cần thiết	15

1. GIỚI THIỆU

1.1. Mục đích và ý nghĩa của Tài liệu

Tài liệu hướng dẫn cài đặt và sử dụng SonarQube để quét mã nguồn với các dự án Java và .NET.

1.2. Phạm vi tài liệu

Tài liệu phục vụ các đối tượng sau:

Cán bộ phát triển: Người thực hiện trực tiếp dự án sử dụng để quét mã nguồn.

1.3. Các thuật ngữ và từ viết tắt

N/A

2. CÀI ĐẶT SERVER SONARQUBE

2.1. Kiến trúc SonarQube

SonarQube là một nền tảng mã nguồn mở để quản lý chất lượng mã nguồn. SonarQube hỗ trợ khá nhiều ngôn ngữ: Java, C#, C/C++, PL/SQL, Cobol, ABAP...

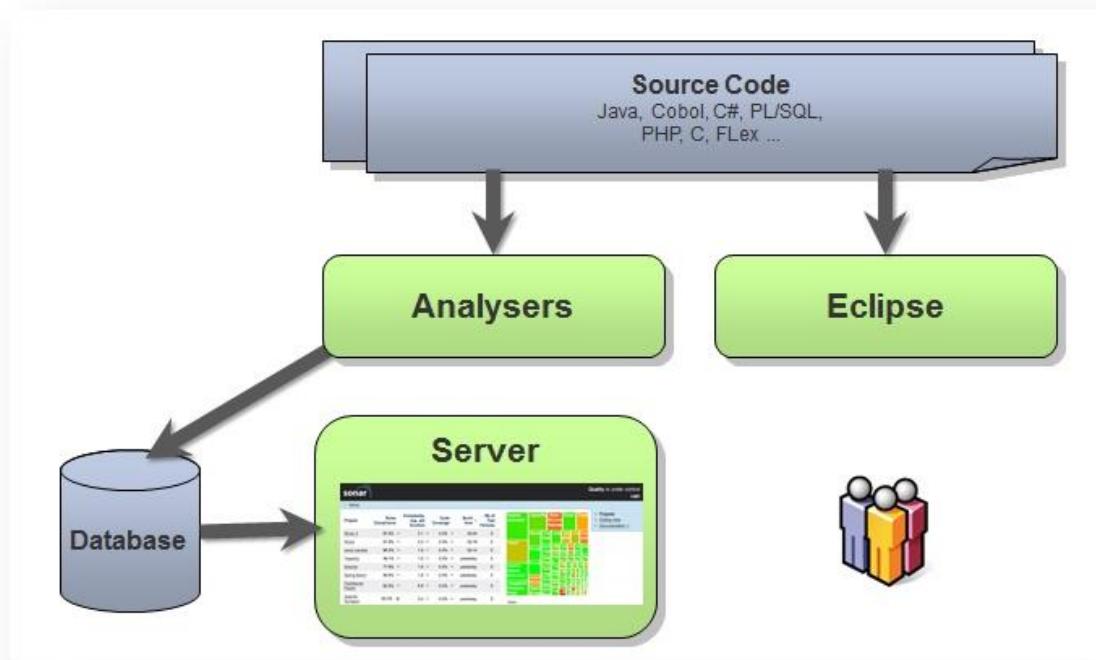
SonarQube platform gồm 3 thành phần:

1. **Database** lưu trữ:

- o Cấu hình của SonarQube (security, plugins settings, etc.)
- o Quality snapshots của các dự án

2. **Web Server** giúp người dùng xem xét trạng thái chất lượng của các dự án một cách trực quan và giúp người dùng cấu hình Sonar

3. Một hoặc vài **Analyzers** dùng để phân tích chất lượng.



2.2. Cài đặt Database

Sonar có thể làm việc được với nhiều loại DBMS khác nhau, để đơn giản hóa và thống nhất thì chúng ta sẽ chỉ dùng MySQL bản opensource.

Việc cài đặt MySQL không được đề cập đến trong tài liệu này, chỉ có lưu ý về việc sử dụng engine InnoDB cho tất cả các table của database.

Để đạt được hiệu năng cao cho DB của Sonar, làm 2 bước sau:

1. Set giá trị Maximum của RAM cho parameter: innodb_buffer_pool_size
2. Set giá trị ít nhất 15Mb cho parameter: query_cache_size

2.3. Cài đặt Web Server

Download Web Server tại địa chỉ: <http://www.sonarqube.org/downloads/>

unzip tại địa chỉ: <install_directory>

<install_directory> được hiểu là nơi giải nén gói Web Server, ví dụ:
D:\Sonar\sonarqube-4.5.1

Cấu hình kết nối với database:

Sửa file <install_directory>/conf/sonar.properties để cấu hình các tùy chọn kết nối tới database. Trong file này sẽ có đầy đủ templates cho tất cả các database mà Sonar hỗ trợ, tuy nhiên sẽ chỉ có một kết nối duy nhất được sử dụng, với MySQL thì phần cấu hình sẽ như sau:

<install_directory>/conf/sonar.properties

```
sonar.jdbc.username=sonar
sonar.jdbc.password=sonar
#---- MySQL 5.x
sonar.jdbc.url=jdbc:mysql://localhost:3306/sonar?useUnicode=true&characterEncoding=
utf8&rewriteBatchedStatements=true&useConfigs=maxPerformance
```

Chú ý tạo trước một database sonar ở trên cơ sở dữ liệu MySql!

Đối với Windows Server 64 bits (tốt nhất là Windows Server 2012 R2), phải cấu hình Sonar hỗ trợ server mode. Tìm tham số sonar.web.javaOpts và xóa comment của nó đi:

```
sonar.web.javaOpts=.....
```

Khởi động Web Server:

Port mặc định sẽ là "9000" và context path mặc định là "/". Các giá trị này có thể được thay đổi bằng cách thay đổi các giá trị trong file <install_directory>/conf/sonar.properties. Ví dụ, muốn đổi thành port 80 và context part thành sonar thì cần cấu hình như sau:

<install_directory>/conf/sonar.properties

```
sonar.web.host=192.0.0.1
sonar.web.port=80
sonar.web.context=/sonar
```

Chạy file bat <install_directory>/bin/windows-x86-XX/StartSonar.bat để khởi động Web Server. Có thể truy cập địa chỉ sau để xem server dùng chung cho VTICT:
<http://10.61.68.127:9000/>

3. CÀI ĐẶT VÀ SỬ DỤNG SCANER

3.1. Quét mã nguồn JAVA project với Sonar-runner.

Cài đặt Sonar-runner:

- Download Sonar-runner từ website. [Click here](#)
- Giải nén vào thư mục cài đặt <install_directory>
- Trong file <install_directory>/conf/sonar-runner.properties, cấu hình cho Sonar-runner như sau:

```
#---- MySQL
```

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

```
#---- Oracle
```

```
#sonar.jdbc.url=jdbc:oracle:thin:@localhost/XE
```

```
#---- Microsoft SQLServer
```

```
#sonar.jdbc.url=jdbc:jtds:sqlserver://localhost/sonar;SelectMethod=Cursor
```

```
#---- Global database settings
```

```
sonar.jdbc.username=sonar
```

```
sonar.jdbc.password=sonar
```

- Tạo biến môi trường SONAR_RUNNER_HOME tới thư mục <install_directory>.
- Thêm <install_directory>/bin vào biến môi trường path.
- Sử dụng command sau để phân tích code:

```
sonar-runner [options]
```

Options:

<i>-D,--define <arg></i>	Define property
<i>-e,--errors</i>	Produce execution error messages
<i>-h,--help</i>	Display help information
<i>-v,--version</i>	Display version information
<i>-X,--debug</i>	Produce execution debug output

3.2. Quét mã nguồn maven project

Điều kiện cần: Đã download và cài đặt Maven

Cấu hình maven:

- Chỉnh sửa file *setting.xml* trong *\$MAVEN_HOME/conf* hoặc trong thư mục *~/.m2* như sau:

```
<settings>
  <profiles>
    <profile>
      <id>sonar</id>
      <activation>
        <activeByDefault>true</activeByDefault>
      </activation>
      <properties>
        <!-- Example for MySQL-->
        <sonar.jdbc.url>
          jdbc:mysql://localhost:3306/sonar?useUnicode=true&characterEncoding=utf8
        </sonar.jdbc.url>
        <sonar.jdbc.username>sonar</sonar.jdbc.username>
        <sonar.jdbc.password>sonar</sonar.jdbc.password>
        <!-- Optional URL to server. Default value is http://localhost:9000 -->
        <sonar.host.url>
          http://myserver:9000
        </sonar.host.url>
      </properties>
    </profile>
  </profiles>
</settings>
```

Chú ý:

Trong trường hợp gặp lỗi *java.lang.OutOfMemoryError*, đặt lại giá trị biến môi trường MAVEN_OPTS như sau:

Với linux:

```
export MAVEN_OPTS="-Xmx512m -XX:MaxPermSize=128m"
```

Với windows:

```
set MAVEN_OPTS=-Xmx512m -XX:MaxPermSize=128m
```

Thực hiện quét với maven project:

```
# The sonar:sonar goal must be executed in a dedicated mvn command
mvn clean install
mvn sonar:sonar
# The following command may lead to unexpected issues
mvn clean install sonar:sonar
```

3.3. Sử dụng Sonar để quét với .NET project

3.3.1. Install C# plugin

The screenshot shows the SonarQube Update Center interface. At the top, there are two tabs: "Installed Plugins" (selected) and "Available Plugins". Under "Available Plugins", there is a table with columns: PLUGIN, VERSION, and DESCRIPTION. The table contains the following data:

PLUGIN	VERSION	DESCRIPTION
C# [csharp]	4.1	Enable analysis and reporting on C# projects.
CSS [css]	1.3	Enables analysis of CSS files.
Java [java]	3.0	SonarQube rule engine.
JavaScript [javascript]	2.7	Enables analysis of JavaScript projects.
PHP [php]	2.6	Enables analysis of PHP projects.
ReSharper [resharper]	2.0	Enables the use of ReSharper rules on C# and VB.

- Chọn Available Plugins, lựa chọn cài đặt C# plugin để cài đặt cho Plugin này.
- Sau khi cài đặt thành công, khởi động lại SonarQube Server để active C# Plugin

3.3.2. Cấu hình MSBuild SonarQube Runner và thực hiện quét với .NET project

Cấu hình MSBuild SonarQube Runner trên Build Agent Machine (Cài đặt trên máy muốn phân tích code, ví dụ máy phát triển hoặc build agent)

Extract:

- Download MSBuild SonarQube Runner từ website của Sonar
- Giải nén MSBuild.SonarQube.Runner-[version] trên thư mục bạn muốn. Ví dụ: C:\SonarQube\bin

Cấu hình:

- Chỉnh sửa file C:\SonarQube\bin\SonarQube.Analysis.xml để có các tham số sau(đối với SonarQube 5.1.x):
 - sonar.jdbc.url
 - sonar.jdbc.username

- o sonar.jdbc.password

```

<?xml version="1.0" encoding="utf-8" ?>
<!--
This file defines properties which would be understood by the MSBuild.SonarQube.Runner, if not overridden (see below)
By default the MSBuild.SonarQube.Runner picks-up a file named SonarQube.Analysis.xml in the folder it
is located (if it exists). It is possible to use another properties file by using the /s:filePath.xml flag

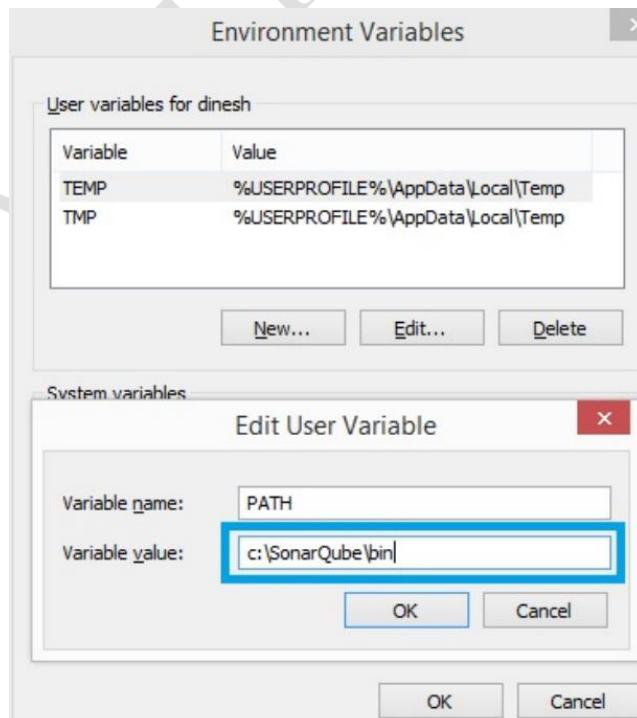
The overriding strategy of property values is the following:
- A project-specific property defined in the MSBuild *.proj file (corresponding to a SonarQube module) can override:
- A property defined in the command line (/d:propertyName=value) has which can override:
- A property defined in the SonarQube.Analysis.xml configuration file [this file] which can override:
- A property defined in the SonarQube User Interface at project level which can override:
- A property defined in the SonarQube User Interface at global level which can't override anything.

Note that the following properties cannot be set through an MSBuild project file or an SonarQube.Analysis.xml file:
sonar.projectName, sonar.projectKey, sonar.projectVersion
The following flags need to be used to set their value: /n:[SonarQube Project Name] /k:[SonarQube Project Key] /v:[SonarQube Pro

-->
<SonarQubeAnalysisProperties xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" x
<Property Name="sonar.host.url">http://localhost:9000</Property>
<!--
<Property Name="sonar.login"></Property>
<Property Name="sonar.password"></Property>
-->
<!-- Required only for versions of SonarQube prior to 5.2 -->
<!--
<Property Name="sonar.jdbc.url">jdbc:jtds:sqlserver://mySqlServer/sonar;instance=SQLEXPRESS;SelectMethod=Cursor</Property>
<Property Name="sonar.jdbc.username">sonar</Property>
<Property Name="sonar.jdbc.password">sonar</Property>
-->
</SonarQubeAnalysisProperties>

```

- Cập nhật đường dẫn %PATH% cho file chạy MSBuild SonarQube Runner. Nếu thư mục giải nén là C:\SonarQube\bin thì cấu hình như sau:



- Thực hiện quét:
 - o Chạy *MSBuild.SonarQube.Runner.exe begin* với cú pháp:

```
MSBuild.SonarQube.Runner.exe begin /v:1.0 /n:"BUGD UNIONE"  
/k:BUGD.UNI.ONE
```

Trong đó:

- /v:"Version project"
- /n:"Name project"
- /k:"Key project"

o Build project:

```
msbuild
```

Trường hợp sử dụng nuget:

```
nuget restore
```

```
msbuild
```

Chú ý: Chạy MSBuild.SonarQube.Runner trong "MSBuild console", hoặc "VS Developer Command Prompt" để có thể chạy được MSBuild command

o Kết thúc

```
MSBuild.SonarQube.Runner.exe end
```

4. HƯỚNG DẪN SỬ DỤNG VÀ TÙY CHỈNH CÁC RULES

4.1. Một số màn hình chung

- Truy cập vào server dùng chung của VTICT theo địa chỉ sau: <http://10.61.68.127:9000/> để xem thông tin của các dự án đã tiến hành phân tích code. Màn hình dashboard hiển thị như sau:

The screenshot shows the SonarQube dashboard for two projects: BUYT_TIEMCHUNG and VOFFICE_VPCP. For BUYT_TIEMCHUNG, Complexity is 4,468 with 4.1 Functions, 10.9 Classes, and 11.6 Files. For VOFFICE_VPCP, Complexity is 40,157 with 6.2 Functions, 50.7 Classes, and 52.4 Files. A red arrow points from the text "Các dự án đã tiến hành quét source code" to the complexity section of the dashboard. To the right, a table titled "PROJECTS" lists various projects with their names, versions, LOC, Technical Debt, and last analysis date. A red arrow points from the text "Trạng thái xử lý lỗi" to the "Technical Debt" column.

NAME	VERSION	LOC	TECHNICAL DEBT	LAST ANALYSIS
BUGD_UNI_ONE_SYNC	1.0	16,484	33d	Sep 28 2015
BUGD_VA_14016_QLTQG_QuanLy	1.0	97,116	305d	Sep 28 2015
BUGD_VA_14016_QLTQG_ThiSinh	1.0	9,896	9d	Sep 19 2015
BUGD_VA_14016_QLTQG_ToolHoTroCham	1.0	6,671	3d 7h	Sep 19 2015
BUYT_HOSPITAL_ONE	1.0	21,076	388d	Sep 21 2015
BUYT_Shi.One_GiamDinh	1.0	180,424	540d	Sep 28 2015
BUYT_Shi.One_TiepNhan	1.0	105,835	1,467d	Sep 21 2015
BUYT_TIEMCHUNG	1.0	29,477	46d	Sep 25 2015
BUYT_YTEXAPHUONG	1.0	532,759	1,388d	Sep 15 2015
CSDLQG_VA_15018_CSDLTKTHDS	1.0	86,336	60d	Sep 19 2015
VOffice_VPCP	1.0	245,290	525d	Sep 16 2015
VTICT:HISONE.BILLINGS	1.0	14,659	11d	Oct 09 2015
VTICT:HISONE.PATIENT	1.0	14,261	10d	Oct 09 2015

- Chi tiết thông tin dự án sau khi quét như sau:

The screenshot shows the project dashboard for BUGD UNI ONE SYNC. It includes a main dashboard with metrics like Lines Of Code (16,484), Files (400), Functions (709), and a complexity section. A red arrow points from the text "Chi tiết các lỗi phát hiện ra trong một dự án" to the "Issues" section, which shows 3,605 issues categorized by severity: Blocker (0), Critical (13), Major (3,269), Minor (323), and Info (0). The dashboard also displays SQALE Rating (A), Technical Debt Ratio (3.3%), and other metrics like Directory Tangle Index (0.0%) and Dependencies To Cut (0).

- Độ nghiêm trọng của các lỗi được phân chia như sau:

Severity	Mô tả
Blocker	Những rủi ro (risk) về hoạt động hoặc an toàn thông tin có thể gây đến sự mất ổn định cho toàn hệ thống.
Critical	Những rủi ro (risk) về hoạt động hoặc an toàn thông tin trong một số trường hợp cụ thể, không ảnh hưởng tới toàn bộ hệ thống. Ví dụ: NullPointerException, badly caught exceptions...
Major	Những vấn đề có thể gây ra tác động đáng kể về hiệu năng. Ví dụ: too complex methods, package cycles, etc.

Severity	Mô tả
Minor	Các vấn đề về coding conventions....
Info	Những rủi ro chưa xác định rõ.

Những lỗi Blocker và Critical là lỗi quan trọng → cần được ưu tiên xử lý triệt để!

- Xem lỗi xảy ra trong từng file và gợi ý sửa lỗi.

The screenshot shows the SonarQube Issues page for the component 'BUGD UNI ONE SYNC'. The left sidebar has 'Issues' selected. The main area displays a list of issues categorized by severity: Blocker (0), Critical (13), Major (3269), Minor (323), and Info (0). A red box highlights a critical error: 'A static field in a generic type is not shared among instances of different close constructed types.' with a link to 'DataSync/GenericSync.cs' at line 12. Another red box highlights another critical error: 'Make the enclosing method "static" or remove this set.' with a link to 'MainService.cs' at line 37. Both errors are marked as 'Critical', 'Open', 'Not assigned', 'Not planned', and have a debt of '10min'.

The bottom part of the screenshot shows a detailed view of a critical issue: 'Static fields should not be used in generic types' (chsharpquid:52743). It includes a note about shared static fields between closed generic types, a 'Noncompliant Code Example' with Java code, and a 'Permalink' button.

4.2. Add thêm bộ rule

Login vào hệ thống với tài khoản mặc định là **admin/admin**.

Sonar cung cấp một số bộ rule khác nhau tương ứng với từng ngôn ngữ.

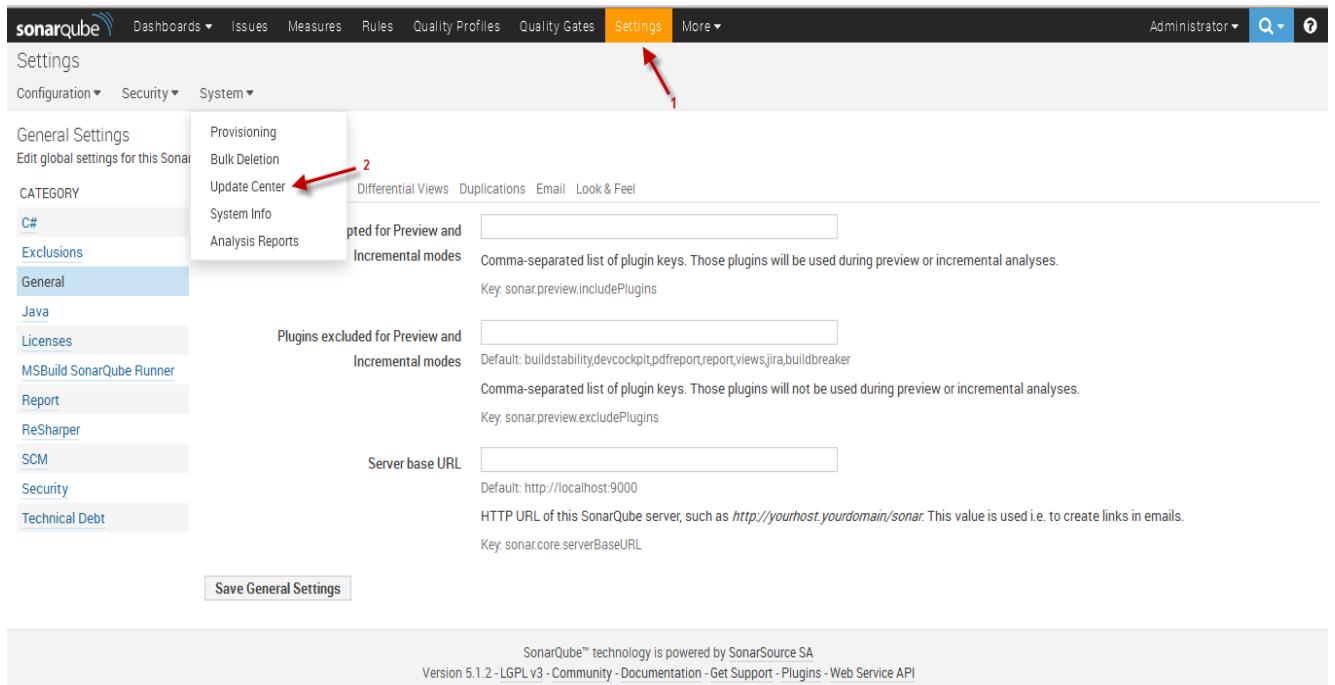
Java:

- SonarQube for java
- PMD
- Findbug
- Checkstyle
- Find security bugs

- ...
.NET

- SonarQube for .NET
- Fxcop/ Code analytic C#
- StyleCop
- ...

Các bước để thêm các bộ rule như sau:



SonarQube Settings - General Settings

General Settings
Edit global settings for this SonarQube instance.

CATEGORY

- C#**
- Exclusions**
- General** (selected)
- Java**
- Licenses**
- MSBuild SonarQube Runner**
- Report**
- ReSharper**
- SCM**
- Security**
- Technical Debt**

Update Center (Step 1)

Differential Views **Duplications** **Email** **Look & Feel**

Plugins for Preview and Incremental modes
Comma-separated list of plugin keys. Those plugins will be used during preview or incremental analyses.
Key: sonar.preview.includePlugins

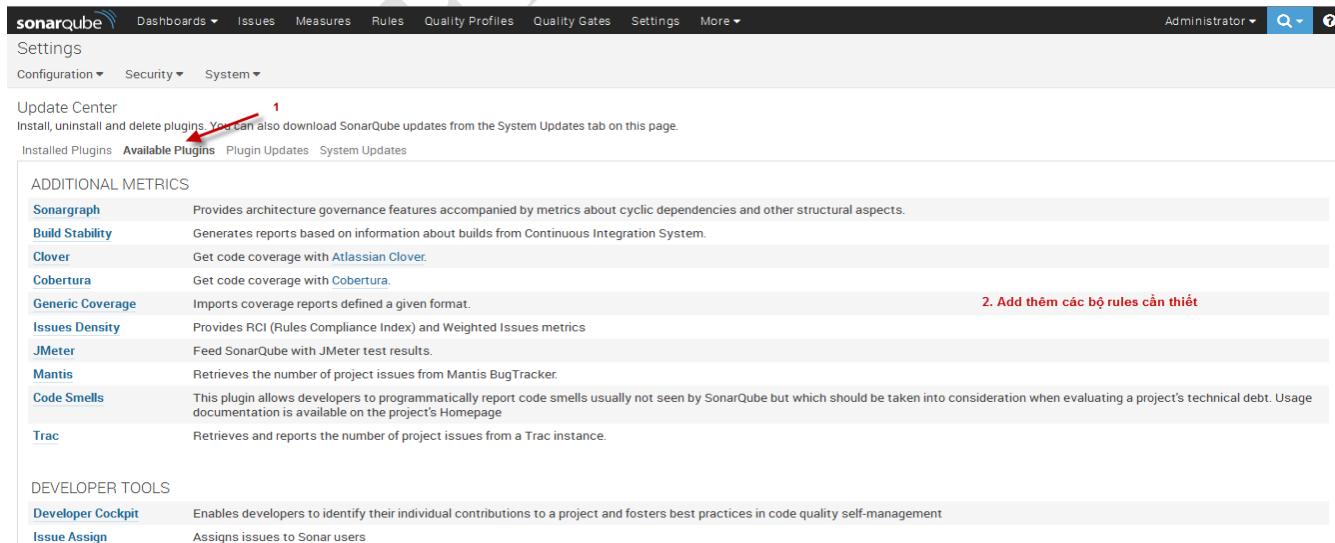
Plugins excluded for Preview and Incremental modes
Comma-separated list of plugin keys. Those plugins will not be used during preview or incremental analyses.
Key: sonar.preview.excludePlugins

Server base URL
Default: http://localhost:9000
HTTP URL of this SonarQube server, such as <http://yourhost.yourdomain/sonar>. This value is used i.e. to create links in emails.
Key: sonar.core.serverBaseUrl

Save General Settings

SonarQube™ technology is powered by SonarSource SA
Version 5.1.2 - LGPL v3 - Community - Documentation - Get Support - Plugins - Web Service API

- Cài đặt thêm các rules muốn sử dụng



SonarQube Settings - Update Center

Update Center (Step 1)
Install, uninstall and delete plugins. You can also download SonarQube updates from the System Updates tab on this page.

Installed Plugins **Available Plugins** (selected) **Plugin Updates** **System Updates**

ADDITIONAL METRICS

Sonargraph	Provides architecture governance features accompanied by metrics about cyclic dependencies and other structural aspects.
Build Stability	Generates reports based on information about builds from Continuous Integration System.
Clover	Get code coverage with Atlassian Clover.
Cobertura	Get code coverage with Cobertura.
Generic Coverage	Imports coverage reports defined a given format.
Issues Density	Provides RCI (Rules Compliance Index) and Weighted Issues metrics
JMeter	Feed SonarQube with JMeter test results.
Mantis	Retrieves the number of project issues from Mantis BugTracker.
Code Smells	This plugin allows developers to programmatically report code smells usually not seen by SonarQube but which should be taken into consideration when evaluating a project's technical debt. Usage documentation is available on the project's Homepage.
Trac	Retrieves and reports the number of project issues from a Trac instance.

DEVELOPER TOOLS

Developer Cockpit	Enables developers to identify their individual contributions to a project and fosters best practices in code quality self-management
Issue Assign	Assigns issues to Sonar users

2. Add them (Step 2) **các bộ rules cần thiết**

Update Center
Install, uninstall and delete plugins. You can also download SonarQube updates from the System Updates tab on this page.

Installed Plugins Available Plugins Plugin Updates System Updates

ADDITIONAL METRICS

- Sonargraph** Provides architecture governance features accompanied by metrics about cyclic dependencies and other structural aspects.
License: Apache License 2
Author: hello2morrow
Links: Homepage Issue Tracker
Version: 3.4.2 (Apr 13, 2015)
- Build Stability** Generates reports based on information about builds from Continuous Integration System.
- Clover** Get code coverage with Atlassian Clover.
- Cobertura** Get code coverage with Cobertura.
- Generic Coverage** Imports coverage reports defined in a given format.
- Issues Density** Provides RCI (Rules Compliance Index) and Weighted Issues metrics
- JMeter** Feed SonarQube with JMeter test results.
- Mantis** Retrieves the number of project issues from Mantis BugTracker.
- Code Smells** This plugin allows developers to programmatically report code smells usually not seen by SonarQube but which should be taken into consideration when evaluating a project's technical debt. Usage documentation is available on the project's Homepage
- Trac** Retrieves and reports the number of project issues from a Trac instance.

- Gom nhóm các bộ rules vừa cài đặt để sử dụng.

Administrator ▾

Rules

1. ? expression can be re-written as ?? expression
2. Repository
3. ? expression has identical true and false branches
4. Bulk Change Activate In... Deactivate In...

Lựa chọn bộ rule gom nhóm để sử dụng

Activate In Quality Profile (675 rules)

FindBugs Java 442
SonarQube Java 318
PMD Java 263
FxCop / Code Analysis C# 233
FindBugs Contrib Java 215
StyleCop C# 170
Checkstyle Java 145
SonarQube C# 86
Find Security Bugs Java 63

Apply Close

4.3. Thêm bớt/xóa bớt một số rule không cần thiết

Đối với từng bộ rules, sẽ có một số rules khi sử dụng chúng ta thấy không cần thiết. Để tiến hành loại bỏ chúng ra khỏi tập rules sử dụng để quét, tiến hành các bước như sau:

- Xem thông tin các bộ rules đang áp dụng.

NAME	RULES	PROJECTS	DEFAULT	OPERATIONS
FindBugs	378	0	Set as Default	Back up Rename Copy Delete
FindBugs Security Audit	72	0	Set as Default	Back up Rename Copy Delete
FindBugs Security Minimal	48	0	Set as Default	Back up Rename Copy Delete
Sonar way	565			✓ Back up Rename Copy

- Lựa chọn một bộ rules, xem xét các mức độ phân loại nghiêm trọng.

Mức Độ	Số lượng
Blocker	26
Critical	259
Major	228
Minor	48
Info	4

- Active/deactive các rules không cần thiết.