

# 代码质量管理平台SonarQube

自动化工具系列

# 大纲

- **Sonar是什么**
- **Sonar安装配置**
- **Sonar Scanner使用**
- **Sonar集成插件**

# Sonar简介

Sonar是一个用于代码质量管理的开源平台，用于管理源代码的质量，可以从七个维度检测代码质量

通过插件形式，可以支持包括java,C#,C/C++,PL/SQL,Cobol,JavaScript,Groovy等等二十几种编程语言的代码质量管理与检测

Sonar是从七个维度检测代码质量，而作为开发人员至少需要处理前5中代码质量问题。

# sonarQube能带来什么?

1 没有代码标准,不遵循代码标准

sonar可以通过PMD,CheckStyle,Findbugs等等代码规则检测工具规范代码编写

2 潜在的bug,潜在的缺陷

sonar可以通过PMD,CheckStyle,Findbugs等等代码规则检测工具检测出潜在的bug

3 糟糕的复杂度分布

文件、类、方法等，如果复杂度过高将难以改变，这会使得开发人员难以理解它们，且如果没有自动化的单元测试，对于程序中的任何组件的改变都将可能导致需要全面的回归测试

4 重复

显然程序中包含大量复制粘贴的代码是质量低下的,sonar可以展示源码中重复严重的地方

5 没有足够的或者过多的注释

没有注释将使代码可读性变差，特别是当不可避免地出现人员变动时，程序的可读性将大幅下降,而过多的注释又会使得开发人员将精力过多地花费在阅读注释上，亦违背初衷

# sonarQube能带来什么？

6 缺乏单元测试

sonar可以很方便地统计并展示单元测试覆盖率

7 糟糕的设计（原文Spaghetti Design，意大利面式设计）

通过sonar可以找出循环，展示包与包、类与类之间的相互依赖关系

可以检测自定义的架构规则

通过sonar可以管理第三方的jar包

可以利用LCOM4检测单个任务规则的应用情况

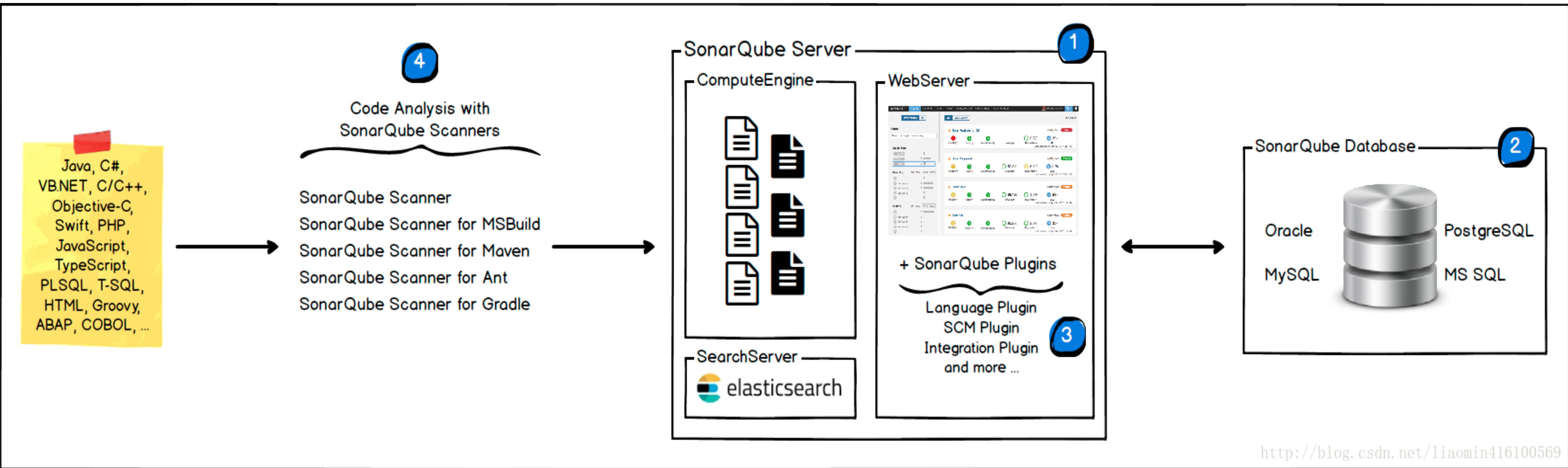
检测耦合

tips：PMD，CheckStyle，Findbugs这些工具都叫静态代码分析工具。什么是静态代码分析？静态代码分析是指无需运行被测代码，仅通过分析或检查源程序的语法、结构、接口等来检查程序的正确性，找出代码隐藏的错误或缺陷，如参数不匹配，有歧义的嵌套语句，错误的递归，非法计算，空指针引用等。

# Sonar组成

sonarqube系统是一个代码质量检测工具 由以下四个组件组成

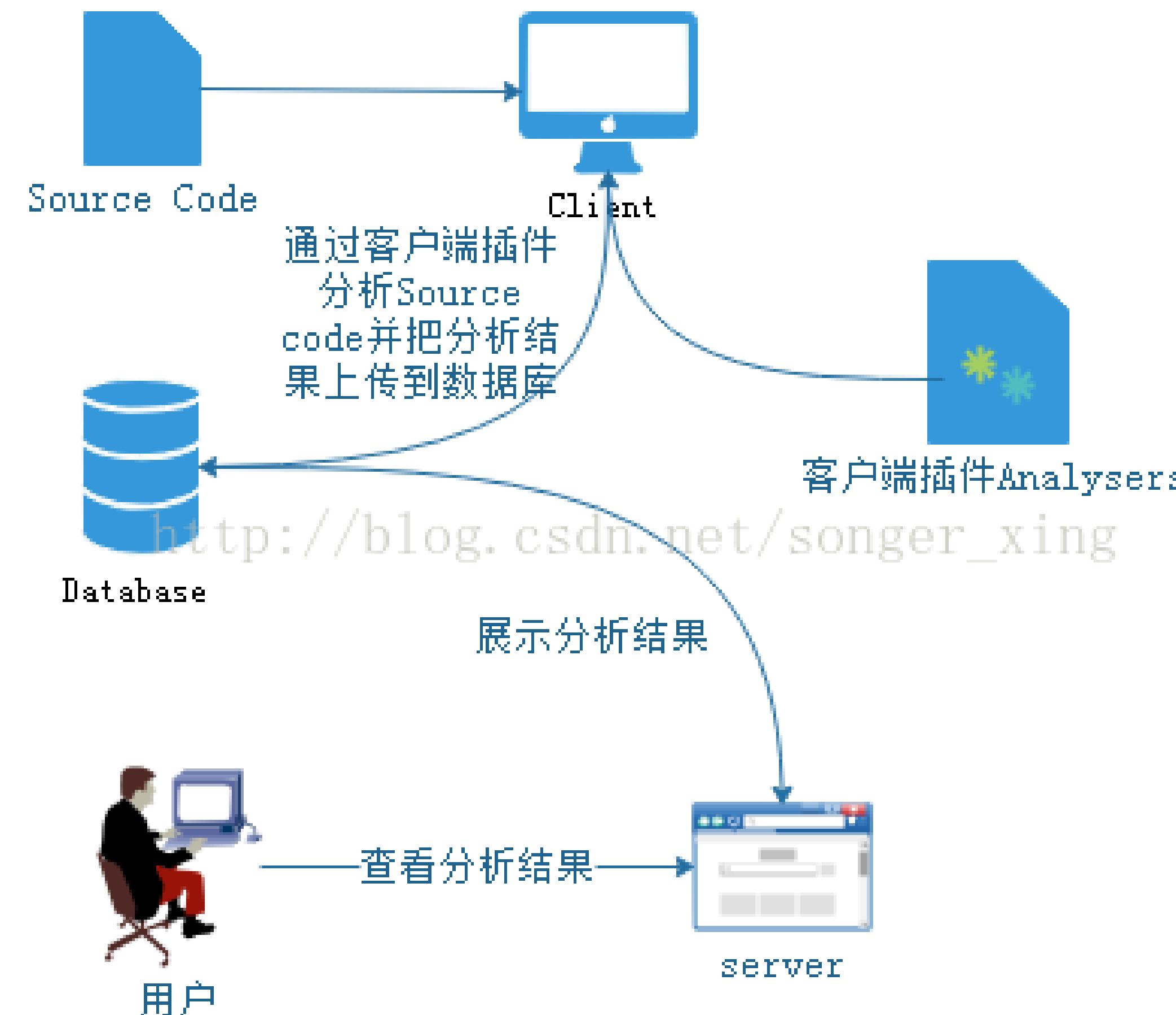
- 1 一个sonarqube服务器，包含三个子进程（web服务（界面管理），搜索服务，计算引擎服务（写入数据库））
- 2 一个sonarqube数据库，配置sonarqube服务
- 3 多个sonarqube插件，位于解压目录extensions\plugins目录
- 4 一个或者多个sonarqube scanners用于分析特定的项目，相当于客户端



<http://blog.csdn.net/liaomin416100569>

# Sonar工作流程

通过客户端插件分析源代码，sonar客户端可以采用IDE插件、Sonar-Scanner插件、Ant插件和Maven插件方式，并通过各种不同的分析机制对项目源代码进行分析和扫描，并把分析扫描后的结果上传到sonar的数据库，通过sonar web界面对分析结果进行管理



# 为什么要选择sonarQube

到目前为止

没有哪个CI工具可以提供良好的钻取功能。

没有CI插件可以将所有的软件质量的度量数据整合到一起。

没有CI插件提供管理视角。

没有设计/架构问题相关的CI插件

没有CI工具或插件提供整体项目质量的仪表盘。

但是sonarQube都有，而且相比于阿里编码规约这种市面上常见类似软件，能提供但不限如下优点

1 更加优秀的图形化界面，基本上通过界面就可以对自己项目的代码状况一目了然

2 可以查询出其它软件难以定位到的问题

a 可能导致空指针异常的问题(对象在进行使用前没有加空的判断)

b 可能导致内存泄漏的问题，在try catch 块里面,直接使用e.printStackTrace()将堆栈信息打印到内存的

c 可能导致的漏洞，成员变量使用public定义的

d 流等未关闭或者是非正常关闭都能够检测出来

.....

# Sonar安装前准备

要求：

Centos7, jdk8, mysql 5.6-5.7, sonarqube-6.7.6(LTS)

请先确保linux已经安装了jdk8和mysql5.7，并且已经配置好了环境

tips: Sonar是基于Java开发的，因此运行Sonar自然需要JDK

# Sonar安装前准备

Sonar要求mysql必须是InnoDB存储引擎

1 查看mysql目前是什么存储引擎

```
# mysql -u root -p  
show engines;      先登录mysql  
                   查看mysql目前提供的存储引擎
```

2 如果不是InnoDB，设置为InnoDB，修改配置文件/etc/my.cnf

```
# vi /etc/my.cnf  
[mysqld]  
default-storage-engine=INNODB
```

重启mysql数据库，再次登录查看默认存储引擎设置是否生效

```
# service mysqld restart
```

# Sonar安装前准备

## 1 设置mysql缓存参数

a 设置innodb\_buffer\_pool\_size,参数值设置尽可能大一些，这个参数主要是缓存InnoDB表的索引，数据，插入数据时的缓冲

默认值：128M，我们这里设置为256M

b 设置mysql的查询缓存query\_cache\_size的开关为1，然后设置15M，重启mysql数据库

```
# vi /etc/my.cnf
[mysqld]
innodb_buffer_pool_size = 256M
query_cache_type=1
query_cache_size=32M
# service mysqld restart
```

## 2 查看缓存设置是否生效

```
show variables like '%query_cache%';
```

```
mysql> show variables like '%query_cache%';
+-----+-----+
| variable_name | value |
+-----+-----+
| have_query_cache | YES |
| query_cache_limit | 1048576 |
| query_cache_min_res_unit | 4096 |
| query_cache_size | 33554432 |
| query_cache_type | ON |
| query_cache_wlock_invalidate | OFF |
+-----+-----+
```

# Sonar安装

1 在mysql中新建一个sonarQube数据库（UTF-8编码）

2 从官网下载最新LTS版本的sonarQuba安装包（sonarqube-6.7.6.zip）

a linux命令行下执行下载：

```
wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-6.7.6.zip
```

b 解压安装并更名

```
unzip sonarqube-6.7.6.zip  
mv sonarqube-6.7.6 sonarQube
```

```
[esuser@sonarQube]$ ls  
bin  conf  COPYING  data  elasticsearch  extensions  lib  logs  temp  web
```

之前介绍组成的时候说sonarqube是sonar的服务端，相当于一个web服务器，用来发布应用，在线浏览、配置分析等。怎么样？有没有很面熟的感觉？是不是有点像tomcat呢？

bin: sonarqube运行命令文件夹

conf: sonarqube配置文件夹

data: 嵌入式数据库的数据（H2数据库引擎），建议只用于测试和演示

elasticsearch: 搜索引擎

extensions: sonarqube的插件等存放文件夹

lib: sonarqube存放的运行库文件夹（jar）

logs: sonarqube日志文件夹

temp: sonarqube临时文件夹

web: sonarqube系统UI界面文件夹

# Sonar安装

## 3 编辑sonar配置

```
# cd sonarQube/conf/  
# vi sonar.properties
```

Mysql数据用户名

```
sonar.jdbc.username=root  
sonar.jdbc.password=5462837zhu
```

配置mysql数据库

```
sonar.jdbc.url=jdbc:mysql://localhost:3306/sonarQube?useUnicode=true&characterEncodingxxxx
```

设置sonar服务

```
sonar.web.host=0.0.0.0  
sonar.web.context=/sonarQube  
sonar.web.port=9000
```

# Sonar安装

4 启动sonarQube web service，第一次启动会自动在数据库生成所需的表，可进入数据库查看下

```
[root@localhost linux-x86-64]# pwd  
/usr/sonar6.7.6/sonarqube/bin/linux-x86-64  
[root@localhost linux-x86-64]# ls  
lib sonar.sh wrapper  
[root@localhost linux-x86-64]# sh sonar.sh start  
starting SonarQube...  
Started SonarQube.  
-----
```

```
sh sonar.sh start
```

5 浏览器中输入：<http://192.168.1.149:9000/sonarQube>



Continuous Code Quality

Log in

Read documentation

# Sonar安装注意的坑

由于6.6版本加入了elasticsearch，不能以root用户启动，因为安全问题elasticsearch不让用root用户直接运行，所以要创建新用户，用新用户启动

```
//创建esuser用户
//目录组和用户都是esuser
//sonarqube文件设置777
//编写配置文件
# useradd esuser
# chown -R esuser.esuser sonar6.7.6
# chmod 777 -R sonarqube-6.7.5
# vi sonar6.7.6/sonarQube/elasticsearch/config/elasticsearch.yml
```

```
//开启端口和指定服务
network.host: 192.168.1.149 (请填写自己服务器的ip)
http.port: 9200
```

# Sonar安装注意的坑

如果启动中出现错误：如：

max file descriptors [4096] for elasticsearch process likely too low, increase to at least [65536]

max virtual memory areas vm.max\_map\_count [65530] likely too low, increase to at least [262144]

解决65536：

切换到root用户，进入limits.d目录下修改配置文件。

`vi /etc/security/limits.conf`

添加如下内容：

```
* soft nofile 65536  
* hard nofile 131072  
* soft nproc 2048  
* hard nproc 4096
```

解决262144：

切换到root用户修改配置sysctl.conf

`vi /etc/sysctl.conf`

添加下面配置：

`vm.max_map_count=655360`

并执行命令：

`sysctl -p`

# Sonar web下载汉化包

帐号密码默认都为： admin， 登录之后下载中文汉化包， 然后重启sonarQube： sh sonar.sh restart

The screenshot shows the SonarQube administration interface with the 'Administration' tab selected. Below it, the 'Marketplace' tab is highlighted with a red box. The main content area displays two editions: 'Community Edition' (Installed) and 'Developer Edition'. A search bar at the bottom has 'chinese pack' typed into it, also highlighted with a red box. A red box highlights the 'Chinese Pack' entry in the search results, which includes 'Localization' and 'SonarQube Chinese Pack'.

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration

Administration Configuration Security Projects System Marketplace

Marketplace

Discover and install new features

Community Edition ✓ Installed

Comes with support for 9 programming languages, numerous plugins, integration with DevOps tool chains, and ability to connect to SonarLint in the IDE.

[Learn more](#)

Developer Edition

Community Edition + branch analysis, SonarLint push notifications, and 16 languages.

[Learn more](#) Upgrade

All Updates Only  chinese pack

Chinese Pack Localization  
SonarQube Chinese Pack

1.19 Support SonarQube 6.7 ...

1 show

# 使用SonarQube Scanner分析代码

1 从官网下载scanner对应的版本：

<https://docs.sonarqube.org/display/SCAN/Analyzing+with+SonarQube+Scanner>

2 通过编辑xxxx/conf/sonar-scanner.properties更新全局设置以指向SonarQube服务器：

```
sonar.host.url = http://192.168.1.149:9000/sonarQube
```

3 将xxxx/bin设置到环境变量PATH中。

4 在命令行输入sonar-scanner -h， 验证安装结果

```
INFO:  
INFO: usage: sonar-scanner [options]  
INFO:  
INFO: Options:  
INFO: -D,--define <arg> Define property  
INFO: -h,--help Display help information  
INFO: -v,--version Display version  
information  
INFO: -X,--debug Produce execution  
debug output
```

# 使用SonarQube Scanner分析代码

5 在需要分析的项目根目录中创建配置文件：sonar-project.properties

```
# 当前项目实例的唯一表示  
sonar.projectKey=kafka  
# 显示在sonarqube 界面上的项目名称  
sonar.projectName=kafka  
sonar.projectVersion=1.0  
  
# 相对于当前配置文件目录 下的源代码目录 不管什么平台路径分隔符只能有 / 不能使用\  
sonar.sources=src/main/java  
  
# 源代码的字符集  
#sonar.sourceEncoding=UTF-8
```

6 从需要分析的项目根目录运行以下命令以启动分析，出现执行EXECUTION SUCCESS表示成功

sonar-scanner

```
.....  
INFO: Task total time: 9.250  
INFO: -----  
INFO: EXECUTION SUCCESS  
INFO: -----  
INFO: Total time: 15.389s  
INFO: Final Memory: 17M/326M
```

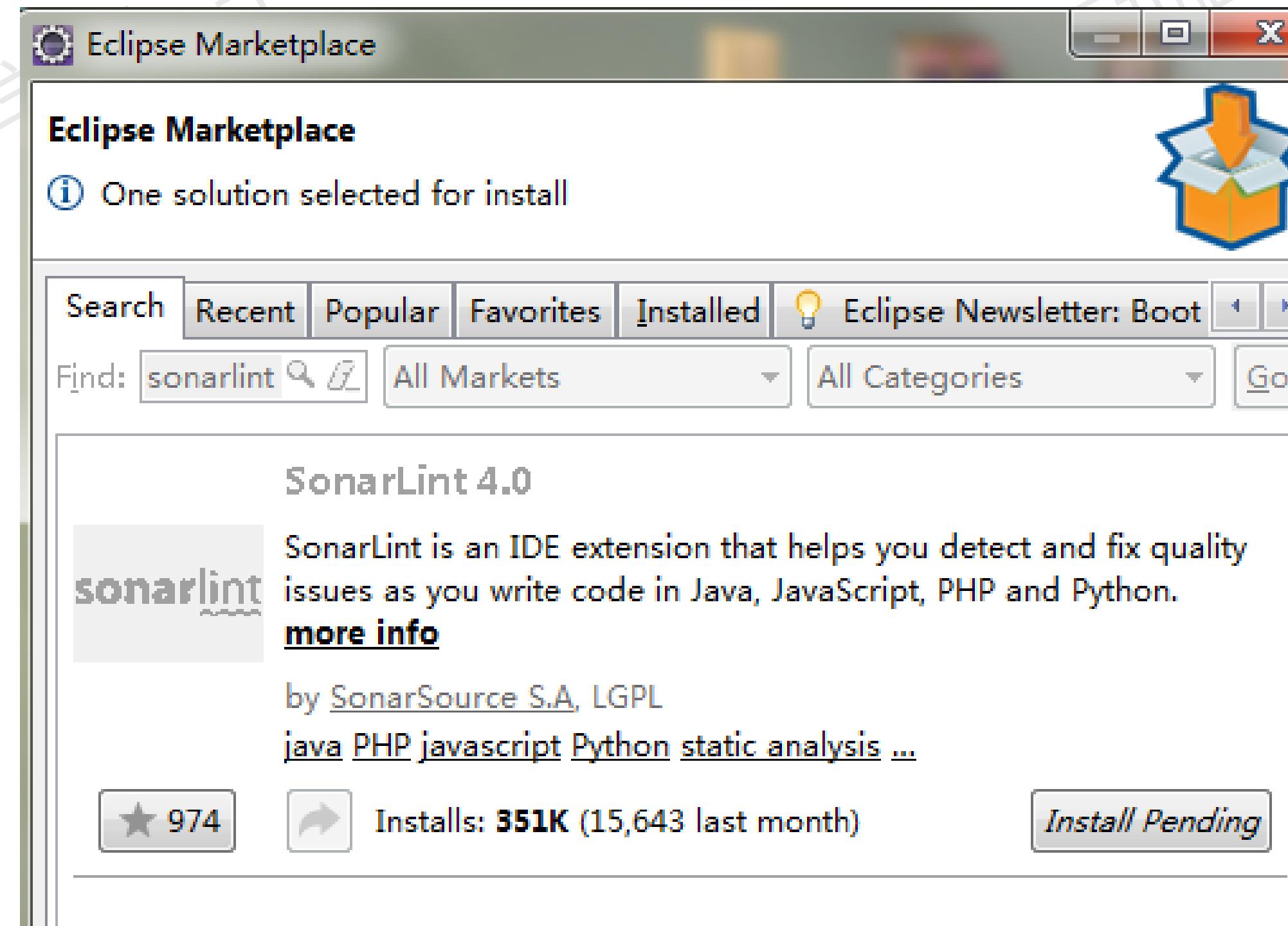
7 登录sonarQube Web查看结果

# 使用SonarQube Eclipse插件分析代码

1 eclipse ide安装sonarlint 实时监测代码质量，插件地址

<https://github.com/SonarSource/sonarlint-eclipse>

也可以使用markerplace搜索安装，例如Eclipse Oxygen版本，可以在markerplace搜索安装



SonarLint相当于sonar的一个插件，它及时反馈给开发人员新的bug和质量问题。  
是常用IDE的一个扩展，如Eclipse、VS、IntelliJIDEA。

# 使用SonarQube Eclipse插件分析代码

2 查找视图sonarLint Bindings点击Connect to a SonarQube server，有两个资源可选，我们选择之前安装的sonarQube。



[Connect to a SonarQube server...](#)

点击完成会eclipse会抛出错误

```
The following plugins do not meet the required minimum versions, please upgrade them on your SonarQube server:  
javascript (installed: 3.2.0.5506, minimum: 4.0)  
java (installed: 4.15.0.12310, minimum: 5.1)
```

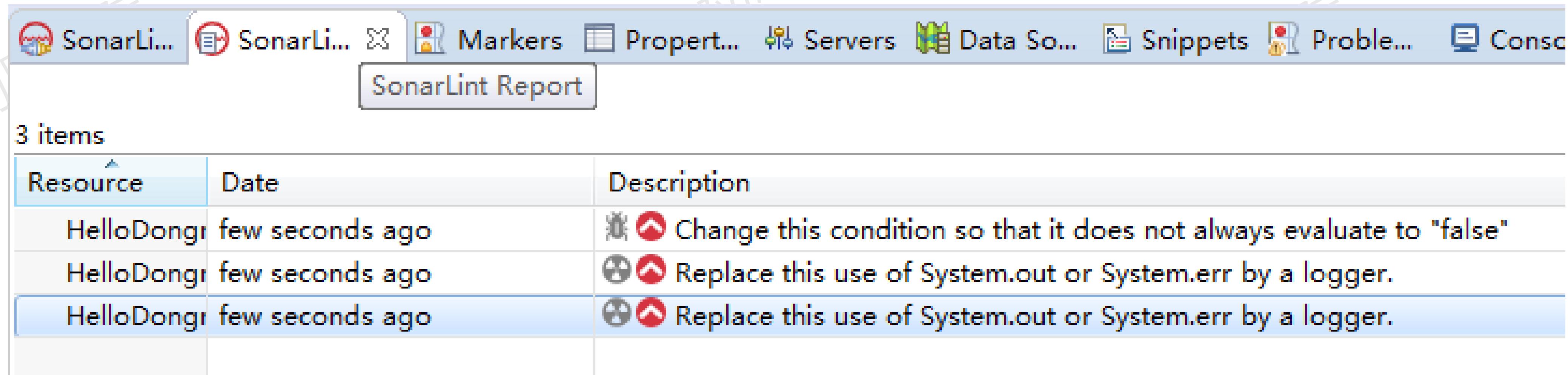
sonarqube使用 sonar-java插件来扫描java代码，使用sonar-javascript插件扫描js代码，现在默认的java插件是4.15，错误提示最小必须使用5.1，js同理。

## 解决方案

- 可以去<https://docs.sonarqube.org/display/PLUG>官网下载对应的插件
- 到sonarqube安装目录下extensions\plugins目录中删除对应的低版本插件，将下载的新版本拷贝进去，然后重启sonarqube
- 然后在eclipse中的sonarLint Bindings视图中右键Update all project setting 即不再报错

# 使用SonarQube Eclipse插件分析代码

3 在项目上右键sonarLint→analyze，在SonarLint Report查看到所有分析结果



The screenshot shows the SonarLint Report view in the Eclipse IDE. The title bar has several tabs: SonarLi..., SonarLi..., Markers, Propert..., Servers, Data So..., Snippets, Proble..., and Conso... The 'SonarLint Report' tab is currently selected, indicated by a blue border. Below the tabs, the text '3 items' is displayed. A table follows, with columns labeled 'Resource', 'Date', and 'Description'. The table contains three rows, each with a resource name 'HelloDongri', a date 'few seconds ago', and a description with a warning icon. The descriptions are: 'Change this condition so that it does not always evaluate to "false"', 'Replace this use of System.out or System.err by a logger.', and 'Replace this use of System.out or System.err by a logger.'.

Resource	Date	Description
HelloDongri	few seconds ago	 Change this condition so that it does not always evaluate to "false"
HelloDongri	few seconds ago	 Replace this use of System.out or System.err by a logger.
HelloDongri	few seconds ago	 Replace this use of System.out or System.err by a logger.

# 使用SonarQube Maven插件分析代码

要求：maven版本3.0以上

Maven的版本	2.X	3.X
兼容性		

1 编辑位于\$MAVEN\_HOME/conf或~/.m2中的settings.xml文件，设置SonarQube服务器URL。

```
<profile>
    <id>sonar</id>
    <activation>
        <activeByDefault>true</activeByDefault>
    </activation>
    <properties>
        <sonar.host.url>http://192.168.1.149:9000/sonarQube</sonar.host.url>
    </properties>
</profile>
```

2 分析Maven项目，运行Maven目标：sonar:sonar在pom.xml文件所在的目录中

```
mvn sonar:sonar
```

3 登录sonarQube Web查看结果

# 简介SonarQube Web使用

The screenshot shows the SonarQube Web interface with two main project cards:

- HelloJava** (正常):
  - Bugs: 1 (C)
  - 漏洞: 0 (A)
  - 坏味道: 1 (A)
  - 覆盖率: 0.0%
  - 重复: 0.0%

最近一次分析: 2018年12月11日 下午3:42  
18 xs Java
- repository** (错误):
  - Bugs: 1 (C)
  - 漏洞: 0 (A)
  - 坏味道: 6 (A)
  - 覆盖率: 0.0%
  - 重复: 0.0%

最近一次分析: 2018年12月12日 上午10:43  
45 xs XML, Java

左侧过滤器和质量阀概览:

- 过滤器: 我的收藏 (selected), 所有
- 质量阀:
  - 正常: 1
  - 警告: 0
  - 错误: 1
- 可靠性 (Bug):
  - A: 0
  - B: 2
  - C: 2
  - D: 0
  - E: 0
- 安全性 (漏洞):
  - A: 2
  - B: 0
  - C: 0
  - D: 0
  - E: 0
- 可维护性 (坏味道):
  - A: 2
  - B: 0
  - C: 0
  - D: 0
  - E: 0

底部信息:

SonarQube™ technology is powered by SonarSource SA  
版本 6.7.6 (build 38781) - LGPL v3 - 社区 - 文档 - 获取支持 - 插件 - Web接口 - 关于

# 简介SonarQube Web使用

## 1 质量阀

SonarQube Web管理者通过配置和设置以下参数值对项目源代码进行：

复杂度、覆盖率、文档、重复、问题、可维护性、可靠性、安全性、大小等约束和规范。

## 2 配置

系统：系统信息

应用市场：下载，更新，卸载插件

项目：创建、编辑、修改、删除(批量)SonarQube项目，查看分析记录

权限：新建用户组，用户，角色，设置角色到项目，以及提供一些默认的角色模版

配置：Java：配置检查的java源文件及静态代码检查规范检查

SCM：配置软件控制器。上文已经提到的配置项。比如：svn、git等等

设置数据库清理，界面等等

## 3 质量配置

查看目前支持的语言插件，查看插件配置的规则，新建规则配置项

## 4 代码规则

查看各个语言提供的规则，分配规则到配置项等

## 5 问题

查看每个项目分析出来的问题，查看问题等

## 6 项目

查看各个项目的问题总数等