#### 1. 安装、配置和启动SonarQube

1.1. 启动MySQL服务器并创建[数据库](https://www.2cto.com/database/" \t "https://www.2cto.com/kf/201709/_blank)sonarqube

**SonarQube服务依赖于数据库MySQL5.6及其以上的版本，用于代码分析结果的数据存储**

启动MySQL

# service [mysql](https://www.2cto.com/database/MySQL/" \t "https://www.2cto.com/kf/201709/_blank)d start

# service mysqld status

创建数据库：sonarqube

# mysql -uroot -p

mysql> CREATE DATABASE `sonarqube` /\*!40100 DEFAULT CHARACTER SET utf8 \*/;

mysql> quit;

1.2. 安装、配置和启动SonarQube服务器

安装

# cd /usr/local

# unzip sonarqube-5.6.6.zip

# mv sonarqube-5.6.6 sonarqube

配置

# vi sonarqube/conf/sonar.properties

sonar.jdbc.url=jdbc:mysql://192.168.2.132:3306/sonarqube?useUnicode=true&characterEncoding=utf8&rewriteBatchedStatements=true&useConfigs=maxPerformance

sonar.jdbc.username=\*\*\*\*\*\*\*\*\*

sonar.jdbc.password=\*\*\*\*\*\*\*\*\*

启动

# /usr/local/sonarqube/bin/linux-x86-64/sonar.sh start

# /usr/local/sonarqube/bin/linux-x86-64/sonar.sh status

#### 2.访问SonarQube服务并配置代码规则

2.1.访问SonarQube服务

[http://192.168.2.132:9000](http://192.168.2.132:9000/" \t "https://www.2cto.com/kf/201709/_blank) ，访问之前须在防火墙中开启9000端口 默认管理员账户和密码：admin/admin

2.2.配置代码规则

进入更新中心（配置>[系统](https://www.2cto.com/os/" \t "https://www.2cto.com/kf/201709/_blank)>更新中心），安装插件

Chinese Pack：汉化包，虽然汉化的非常不彻底，廖剩无几吧 Checkstyle ：Analyze [Java](https://www.2cto.com/kf/ware/Java/" \t "https://www.2cto.com/kf/201709/_blank) code with Checkstyle. PMD ：Analyze Java code with PMD. SonarJava ：Code Analyzer for Java Sonar[Python](https://www.2cto.com/kf/web/Python/" \t "https://www.2cto.com/kf/201709/_blank) ：Code Analyzer for Python

#### 源码">3. 使用SonarQube分析maven项目源码

3.1. 修改maven的配置文件setting.xml，配置SonarQube插件

# vi setting.xml

org.sonarsource.scanner.maven

sonar

true

jdbc:mysql://192.168.2.132:3306/sonarqube?useUnicode=true&characterEncoding=utf8

\*\*\*\*\*\*

\*\*\*\*\*\*

http://192.168.2.132:9000

3.2. 基于命令行进行代码审查

**切换到项目根路径（pom.xml文件所在的路径），使用下面的命令进行代码审查**

# mvn clean install sonar:sonar

3.3. 基于IDE：eclipse进行代码审查

#----- 切换到执行自定义maven命令的界面

project>right key>Run As>maven build...

#----- 执行代码审查命令

Goals:clean install sonar:sonar

#### 4. 使用SonarQube插件sonar-runner分析非maven项目源码

4.1. 配置sonar-runner

配置sonar-runner服务到系统变量

增加变量SONAR\_RUNNER\_HOME;

追加%SONAR\_RUNNER\_HOME%\bin到path尾部

配置sonar-runner服务

# vi config sonar-runner.properties

#----- SonarQube server

sonar.host.url=http://192.168.2.132:9000

#----- MySQL

sonar.jdbc.url=jdbc:mysql://192.168.2.132:3306/sonar?useUnicode=true&characterEncoding=utf8

#----- Global database settings

sonar.jdbc.username=root

sonar.jdbc.password=qiqi7179

#----- Security (when 'sonar.forceAuthentication' is set to 'true')

sonar.login=admin

sonar.password=admin

4.2. 使用sonar-runner分析java代码

项目根路径创建配置文件sonar.properties

#----- projectKey和projectValue用作唯一标识项目

sonar.projectKey=

sonar.projectName=

sonar.projectVersion=

#----- 源码路径，如果在多个路径，用英文逗号‘,’隔开

sonar.sources=src

#----- 编译后的class文件路径

sonar.java.binaries=WebContent/WEB-INF/classes

sonar.language=java

sonar.sourceEncoding=UTF-8

#----- 此项设置不需要动

sonar.my.property=value

\*切换项目根路径执行代码审查命令

# sonar-runner

4.3. 使用sonar-runner分析python代码

项目根路径创建配置文件sonar.properties

sonar.projectKey=Py

sonar.projectName=Py

sonar.projectVersion=1.0

sonar.sources=./

#----- 此项设置不要动

sonar.language=py

sonar.sourceEncoding=UTF-8

sonar.my.property=value

\*切换项目根路径执行代码审查命令

# sonar-runner

#### 5. 使用Eclipse插件SonarLint进行代码审查

Eclipse MarketPlace搜索SonarLint安装后重启即可自动分析代码（java、python均可）

该插件的好处主要是在编写代码的过程中就会有提示可能的坏味道、[漏洞](https://www.2cto.com/" \t "https://www.2cto.com/kf/201709/_blank)和bug，不用到代码审查阶段就可以大大提高代码的质量

该插件也可以结合SonarQube使用，将分析的结果发送给SonarQube