

第一章 安装准备

1.1 规划

OS : Ubuntu 14.04.4 x64
IP : 172.16.6.31
JDK : 1.8.0_171
Tomcat : 8.5.33.0
Jenkins : 2.121.3
目录: /data/tomcat-jenkins-8080/webapps/Jenkins
web: <http://172.16.6.31:8080/jenkins/>
账单密码: admin/admin
SonarQube: 6.7.5
目录: /data/sonarqube-6.7.5
web: <http://172.16.6.31:9000>
账号密码: admin/admin
Sonar-scanner: 2.8
目录: /data/sonar-scanner-2.8
MySQL: 5.7.11
jdbc:mysql://10.1.5.15:3306
用户密码: sonar/123456

1.2 创建程序用户

useradd lx2
sonarqube 不能以 root 用户运行, 否则无法启动起来

1.3 open files 参数

保证使用 ulimit -n 输出的是不小于 65536 的数, 否则 sonarqube 起不来

```
[lx2@yunwei data]$ ulimit -n  
65536  
[lx2@yunwei data]$
```

1.4 MySQL 准备

mysql 按照步骤略。

保证 mysql 版本不低于 5.6, 否则 sonarqube 起不来

创建 mysql 用户及 sonar 数据库:

```
CREATE DATABASE sonar CHARACTER SET utf8 COLLATE utf8_general_ci;  
GRANT ALL ON sonar.* TO 'sonar'@'%' IDENTIFIED BY '123456';
```

```
FLUSH PRIVILEGES;
```

1.5 jdk 安装

保证能看到如下效果

```
[lx2@yunwei ~]$ java -version  
  
java version "1.8.0_171"  
Java(TM) SE Runtime Environment (build 1.8.0_171-b11)  
Java HotSpot(TM) 64-Bit Server VM (build 25.171-b11, mixed mode)  
[lx2@yunwei ~]$  
[lx2@yunwei ~]$  
[lx2@yunwei ~]$ which java  
/usr/local/jdk/bin/java  
[lx2@yunwei ~]$
```

步骤这里略过。

1.6 tomcat 安装

用来运行 jenkins，步骤略

第二章 SonarQube 安装配置

2.1 sonar 下载

到这里

<https://www.sonarqube.org/downloads/>

选择合适版本，我选了 6.7.5 版本

并解压到规划的路径

2.2 修改配置文件

cat /data/sonarqube-6.7.5/conf/sonar.properties

保证有以下三行：

```
sonar.jdbc.username=sonar
sonar.jdbc.password=123456
sonar.jdbc.url=jdbc:mysql://10.1.5.15:3306/sonar?useUnicode=true&characterEncoding=utf8&rewriteBatchedStatements=true&useConfigs=maxPerformance&useSSL=false
```

```
# The schema must be created first.
sonar.jdbc.username=sonar
sonar.jdbc.password=123456

#----- Embedded Database (default)
# H2 embedded database server listening port, defaults to 9092
#sonar.embeddedDatabase.port=9092

#----- MySQL 5.6 or greater
# Only InnoDB storage engine is supported (not myISAM).
# Only the bundled driver is supported. It can not be changed.
sonar.jdbc.url=jdbc:mysql://10.1.5.15:3306/sonar?useUnicode=true&characterEncoding=utf8&rewriteBatchedStatements=true&useConfigs=maxPerformance&useSSL=false
```

2.3 启动

/data/sonarqube-6.7.5/bin/linux-x86-64/sonar.sh start

另外，该命令也可使用 stop|restart|status 等参数

2.4 汉化

到如下

<https://github.com/SonarQubeCommunity/sonar-l10n-zh>

地址，选择合适版本



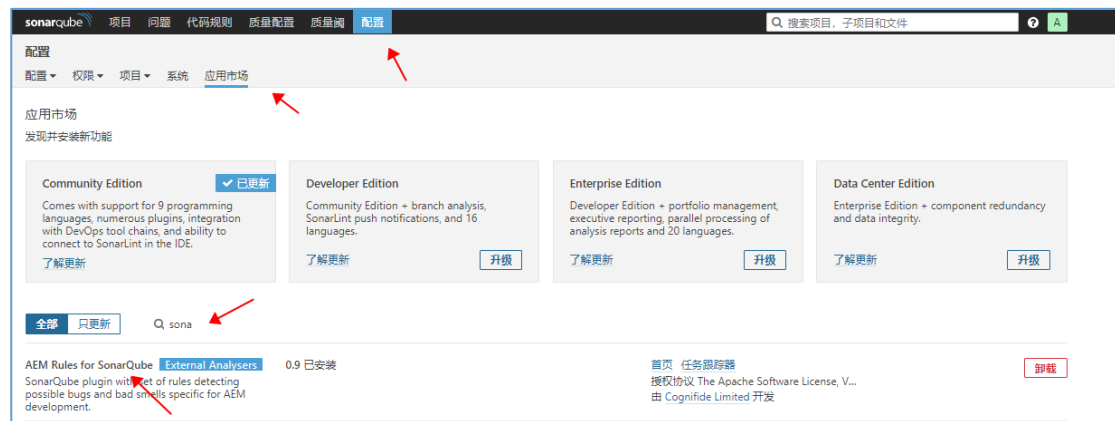
下载 jar 包，放到
/data/sonarqube-6.7.5/extensions/plugins
目录

```
[lx2@yunwei plugins]$ pwd
/data/sonarqube-6.7.5/extensions/plugins
[lx2@yunwei plugins]$
[lx2@yunwei plugins]$ ls sonar-l10n-zh-plugin-1.19.jar
sonar-l10n-zh-plugin-1.19.jar
[lx2@yunwei plugins]$
```

重启服务，汉化就生效了

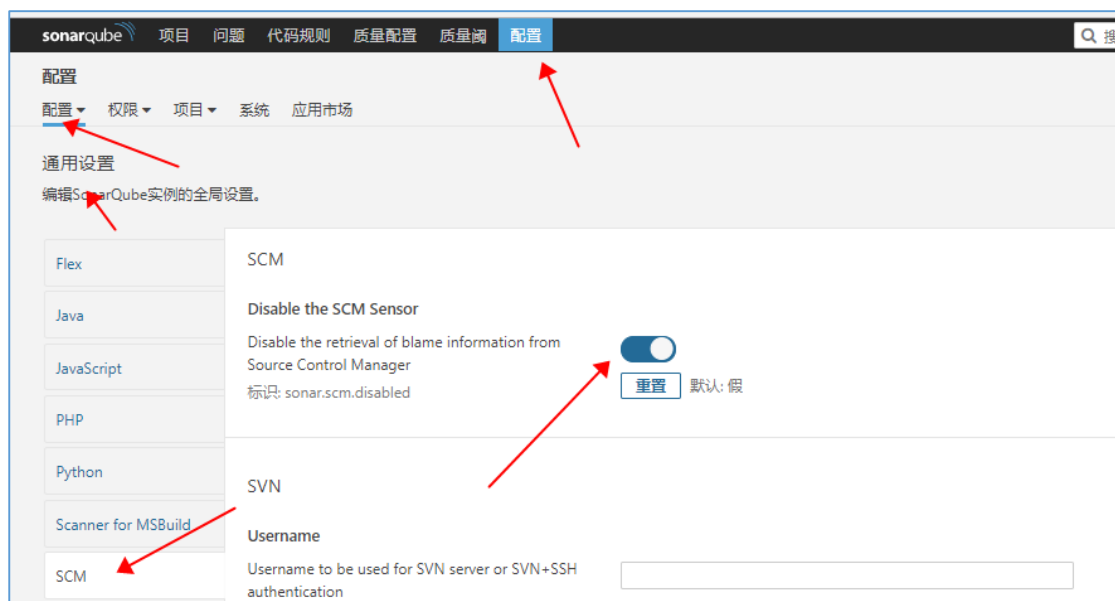
2.5 安装插件

如下图所示，安装插件
AEM Rules for SonarQube



2.6 禁用 scm

如下图所示，禁用 scm 功能



2.7 生成 token

如下图，点击即可生成 token



只显示一次，注意记录，比如我的 token 是

7a4ee23f4dc52b16cb625021efd151d1a8e15fdb

第三章 SonarQube Scanner

3.1 下载和解压

到

<https://sonarsource.bintray.com/Distribution/sonar-scanner-cli/>

选择合适版本，下载，比如我的

wget <https://sonarsource.bintray.com/Distribution/sonar-scanner-cli/sonar-scanner-2.8.zip>

然后解压到规划的目录

3.2 修改配置文件（可选）

```
cat /data/sonar-scanner-2.8/conf/sonar-scanner.properties
```

保证有以下两行即可：

```
sonar.host.url=http://172.16.6.31:9000
```

默认是

```
sonar.host.url=http://localhost:9000
```

```
[lx2@yunwei conf]$ pwd
/data/sonar-scanner-2.8/conf
[lx2@yunwei conf]$
[lx2@yunwei conf]$ grep -v \# sonar-scanner.properties
sonar.host.url=http://172.16.6.31:9000
```

如果 sonar 和 sonar-scanner 在同一台机器，可不修改。

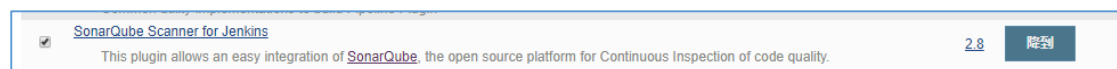
第四章 Jenkins 安装配置

4.1 下载和安装

到 jenkins 官网下载 war 包，放到 tomcat/webapps 下面，启动即可，无其他特殊配置，比较简单，这里略过

4.2 插件安装

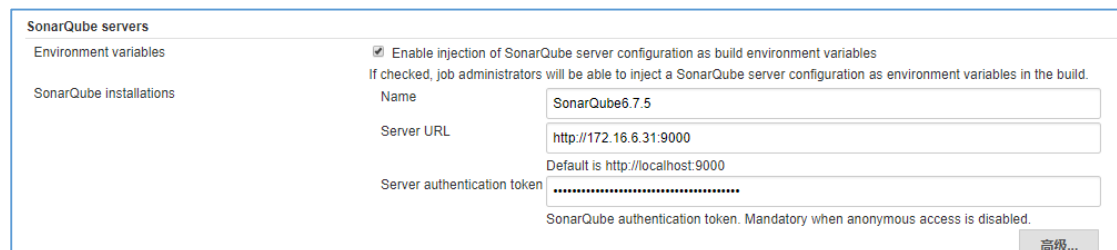
安装插件 SonarQube Scanner for Jenkins



4.3 SonarQube 配置

系统配置—>系统配置，找到
SonarQube installations
段，按照下面进行配置

Name : SonarQube6.7.5 #随意写
Server URL : http://172.16.6.31:9000 #准确填写
Server authentication token : **** #2.7 生成的 token



4.4 SonarQube Scanner 配置（可选）

系统配置—>全局工具设置，找到 SonarQube Scanner 部分，进行配置

Name : sonar-scanner-2.8 #随意写
SONAR_RUNNER_HOME : /data/sonar-scanner-2.8 #这个一定要准确填写

SonarQube Scanner

SonarQube Scanner 安装

新增 SonarQube Scanner

SonarQube Scanner

Name sonar-scanner-2.8

SONAR_RUNNER_HOME /data/sonar-scanner-2.8

☐ 自动安装

新增 SonarQube Scanner

系统下SonarQube Scanner 安装列表

其实这一步也可以配置到操作系统环境变量里面，添加之后就不需要进行这一步了。

```
export SONAR_SCANNER_HOME=/data/sonar-scanner-2.8
export PATH=${SONAR_SCANNER_HOME}/bin:${PATH}
```

4.5 构建任务时添加步骤

添加构建步骤

Execute SonarQube Scanner

动作，内容如下

#项目 key 和名称，在 sonar 里面要唯一

sonar.projectKey=openapi

sonar.projectName=openapi111

#项目版本

sonar.projectVersion=1.0

#源码位置

sonar.sources=.

#编译后的 class 位置

sonar.java.binaries=target

Execute SonarQube Scanner

X

Task to run

?

JDK

(Inherit From Job)

?

JDK to be used for this SonarQube analysis

Path to project properties

?

Analysis properties

#项目key和名称，在sonar里面要唯一
sonar.projectKey=openapi
sonar.projectName=openapi111
#项目版本
sonar.projectVersion=1.0
#源码位置
sonar.sources=.
#编译后的class位置
sonar.java.binaries=target

?

Additional arguments

?

JVM Options

?

第五章 观察和排错

5.1 web 观察效果

登陆 sonar 的 web, <http://172.16.6.31:9000/projects>



可进入具体项目查看 bugs 等详细内容



5.2 异常处理

构建日志中出现如下错误:

```
INFO: -----
INFO: Total time: 2:22.742s
INFO: Final Memory: 61M/771M
INFO: -----
ERROR: Error during SonarQube Scanner execution
ERROR: Failed to upload report - 500: An error has occurred. Please contact your administrator
ERROR:
ERROR: Re-run SonarQube Scanner using the -X switch to enable full debug logging.
ERROR: SonarQube scanner exited with non-zero code: 1
Finished: FAILURE
```

查看 sonarqube 日志/data/sonarqube-6.7.5/logs/web.log:
出现如下错误

```
at org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:61)
at java.lang.Thread.run(Thread.java:745)
Caused by: com.mysql.jdbc.PacketTooBigException: Packet for query is too large (4452942 > 4194304). You can change this value on the server by setting the max_allowed_packet' variable.
at com.mysql.jdbc.MysqlIO.send(MysqlIO.java:2678)
at com.mysql.jdbc.MysqlIO.sendCommand(MysqlIO.java:2599)
at com.mysql.jdbc.MysqlIO.sqlQueryDirect(MysqlIO.java:2680)
at com.mysql.jdbc.ConnectionImpl.execSQL(ConnectionImpl.java:2490)
at com.mysql.jdbc.PreparedStatement.executeInternal(PreparedStatement.java:1858)
at com.mysql.jdbc.PreparedStatement.executeUpdateInternal(PreparedStatement.java:2079)
at com.mysql.jdbc.PreparedStatement.executeUpdateInternal(PreparedStatement.java:2013)
at com.mysql.jdbc.PreparedStatement.executeUpdate(PreparedStatement.java:5104)
```

Caused by: com.mysql.jdbc.PacketTooBigException: Packet for query is too large (4452942 > 4194304). You can change this value on the server by setting the max_allowed_packet' variable.

此错误表示，发到 mysql 的包太大，超过了显示，此时需要检查
第一处：mysql 的 max_allowed_packet 参数，如下

```
mysql> show variables like 'max_allowed_packet';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| max_allowed_packet | 33554432 |
+-----+-----+
1 row in set
```

mysql 配置为 32M，这个配置没问题，如果有问题，请解决这个问题

第二处：修改/data/sonarqube-6.7.5/conf/sonar.properties 内容如下
sonar.jdbc.url=jdbc:mysql://10.1.5.15:3306/sonar?useUnicode=true&characterEncoding=utf
8&rewriteBatchedStatements=true&useConfigs=maxPerformance&maxAllowedPacket=200
000000&useSSL=false

其实就是在 2.2 步基础上添加&maxAllowedPacket=200000000 参数

第六章 添加单元测试内容

经过以上几步, 已经实现了基本的代码检测功能, 但还有一项就是单元测试覆盖率显示为 0, 如下图



下面开始处理这部分:

6.1 添加构建步骤

在 Execute SonarQube Scanner 步骤前,

在 ansible 部署这一步的后面

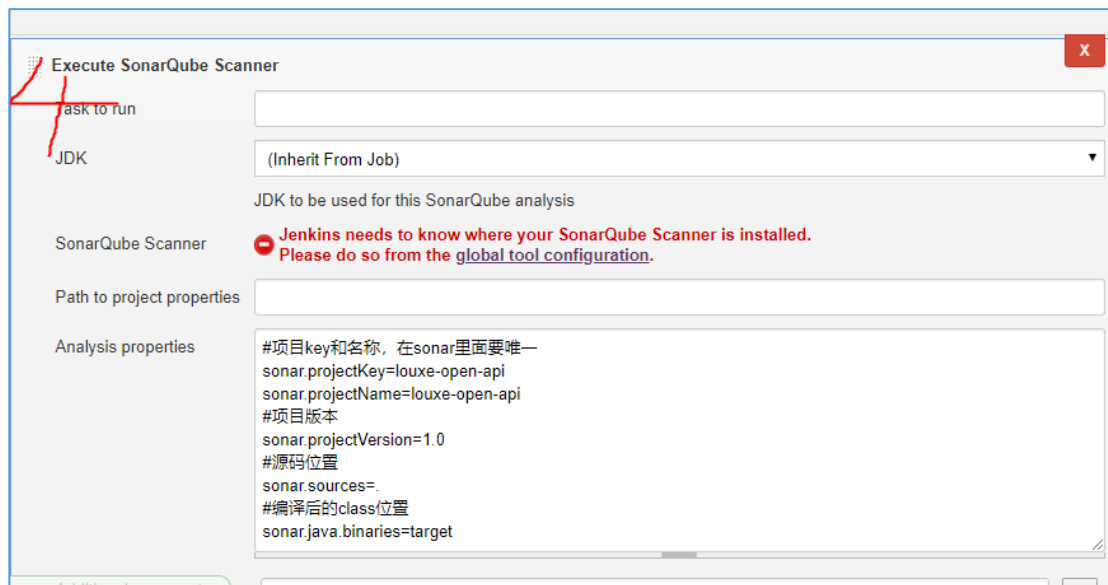
添加如下两行

```
mvn clean org.jacoco:jacoco-maven-plugin:prepare-agent install -Dmaven.test.failure.ignore=true
```

```
mvn sonar:sonar
```

如图





一定要注意前后顺序,

- 1) 放到部署这一步的后面, 因为这一步会导致编译出来的 jar 包被修改
- 2) 放到上报这一步前面, 是为了保证上报时把这一步的执行结果也一并上报

6.2 pom.xml 配置

保证未跳过单元测试, 设置为 false, 或者不配置这一段内容。

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-surefire-plugin</artifactId>
  <configuration>
    <skip>false</skip>
  </configuration>
</plugin>
```

如下图:

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-surefire-plugin</artifactId>
  <configuration>
    <skip>false</skip>
  </configuration>
</plugin>
```

6.3 观察效果



