一：Sonar是什么？  
Sonar 是一个用于代码质量管理的开放平台，通过插件机制，Sonar 可以集成不同的测试工具，代码分析工具，以及持续集成工具。与持续集成工具（例如 Hudson/Jenkins 等）不同，Sonar 并不是简单地把不同的代码检查工具结果（例如FindBugs，PMD等）直接显示在Web页面上，而是通过不同的插件对这些结果进行再加工处理，通过量化的方式度量代码质量的变化，从而可以方便地对不同规模和种类的工程进行代码质量管理。在对其他工具的支持方面，Sonar 不仅提供了对 IDE 的支持，可以在 Eclipse 和 IntelliJ IDEA 这些工具里联机查看结果；同时Sonar还对大量的持续集成工具提供了接口支持，可以很方便地在持续集成中使用 Sonar,此外，Sonar 的插件还可以对 Java 以外的其他编程语言提供支持，对国际化以及报告文档化也有良好的支持。

#官方网站：http://www.sonarqube.org/

Sonar部署  
Sonar的相关下载和文档可以在下面的链接中找到：http://www.sonarqube.org/downloads/。需要注意最新版的Sonar需要至少JDK 1.8及以上版本

cd /usr/local/src

cd /usr/local/src/wget https://sonarsource.bintray.com/Distribution/sonarqube//sonarqube-5.6.6.zipunzip sonarqube-5.6.6.zip mv sonarqube-5.6.6 /usr/local/ln -s /usr/local/sonarqube-5.6.6/ /usr/local/sonarqube

准备Sonar数据库(mysql版本要等于5.6或者5.6以上，否则sonar无法启动)

Cat /etc/yum.repos.d/mysql-community.repo

[mysql-connectors-community]

name=MySQL Connectors Community

baseurl=http://repo.mysql.com/yum/mysql-connectors-community/el/6/$basearch/

enabled=1

gpgcheck=1

gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

[mysql-tools-community]

name=MySQL Tools Community

baseurl=http://repo.mysql.com/yum/mysql-tools-community/el/6/$basearch/

enabled=1

gpgcheck=1

gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

# Enable to use MySQL 5.5

[mysql55-community]

name=MySQL 5.5 Community Server

baseurl=http://repo.mysql.com/yum/mysql-5.5-community/el/6/$basearch/

enabled=0

gpgcheck=1

gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

# Enable to use MySQL 5.6

[mysql56-community]

name=MySQL 5.6 Community Server

baseurl=http://repo.mysql.com/yum/mysql-5.6-community/el/6/$basearch/

enabled=1

gpgcheck=1

gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

# Note: MySQL 5.7 is currently in development. For use at your own risk.

# Please read with sub pages: https://dev.mysql.com/doc/relnotes/mysql/5.7/en/

[mysql57-community-dmr]

name=MySQL 5.7 Community Server Development Milestone Release

baseurl=http://repo.mysql.com/yum/mysql-5.7-community/el/6/$basearch/

enabled=0

gpgcheck=1

gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

[root@lanmaoly-3 yum.repos.d]# cat mysql-community-source.repo

[mysql-connectors-community-source]

name=MySQL Connectors Community - Source

baseurl=http://repo.mysql.com/yum/mysql-connectors-community/el/6/SRPMS

enabled=0

gpgcheck=1

gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

[mysql-tools-community-source]

name=MySQL Tools Community - Source

baseurl=http://repo.mysql.com/yum/mysql-tools-community/el/6/SRPMS

enabled=0

gpgcheck=1

gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

[mysql55-community-source]

name=MySQL 5.5 Community Server - Source

baseurl=http://repo.mysql.com/yum/mysql-5.5-community/el/6/SRPMS

enabled=0

gpgcheck=1

gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

[mysql56-community-source]

name=MySQL 5.6 Community Server - Source

baseurl=http://repo.mysql.com/yum/mysql-5.6-community/el/6/SRPMS

enabled=0

gpgcheck=1

gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

[mysql57-community-dmr-source]

name=MySQL 5.7 Community Server Development Milestone Release - Source

baseurl=http://repo.mysql.com/yum/mysql-5.7-community/el/6/SRPMS

enabled=0

gpgcheck=1

gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

mysql> CREATE DATABASE sonar CHARACTER SET utf8 COLLATE utf8\_general\_ci;

mysql> GRANT ALL ON sonar.\* TO 'sonar'@'%' IDENTIFIED BY 'sonar@341Jpw';

mysql> FLUSH PRIVILEGES;

配置Sonar

# cd /usr/local/sonarqube/conf/

修改配置文件的数据库配置

# egrep '^[a-Z]' sonar.properties

sonar.jdbc.username=sonar

sonar.jdbc.password=sonar@pw

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

启动Sonar  
你可以在Sonar的配置文件来配置Sonar Web监听的IP地址和端口，默认是9000端口。

# vim sonar.properties

sonar.web.host=0.0.0.0

sonar.web.port=9000

# 启动有创建表和其他操作，速度会有点慢

IMG_256

[root@node1 conf]# /usr/local/sonarqube/bin/linux-x86-64/sonar.sh start

Starting SonarQube...

Started SonarQube.

报错：

org.sonar.api.utils.MessageException: Unsupported mysql version: 5.5. Minimal supported version is 5.6.2017.07.01 11:16:27 ERROR web[o.a.c.c.StandardContext] One or more listeners failed to start. Full details will be found in the appropriate container log file2017.07.01 11:17:09 INFO web[o.a.c.u.SessionIdGeneratorBase] Creation of SecureRandom instance for session ID generation using [SHA1PRNG] took [41,747] milliseconds.2017.07.01 11:17:09 ERROR web[o.a.c.c.StandardContext] Context [] startup failed due to previous errors2017.07.01 11:17:09 WARN web[o.a.c.l.WebappClassLoaderBase] The web application [ROOT] appears to have started a thread named [Abandoned connection cleanup thread] but has failed to stop it. This is very likely to create a memory leak. Stack trace of thread:

java.lang.Object.wait(Native Method)

java.lang.ref.ReferenceQueue.remove(ReferenceQueue.java:143)

com.mysql.jdbc.AbandonedConnectionCleanupThread.run(AbandonedConnectionCleanupThread.java:43)2017.07.01 11:17:09 WARN web[o.a.c.l.WebappClassLoaderBase] The web application [ROOT] appears to have started a thread named [Timer-0] but has failed to stop it. This is very likely to create a memory leak. Stack trace of thread:

java.lang.Object.wait(Native Method)

java.util.TimerThread.mainLoop(Timer.java:552)

java.util.TimerThread.run(Timer.java:505)2017.07.01 11:17:09 INFO web[o.a.c.h.Http11NioProtocol] Starting ProtocolHandler ["http-nio-0.0.0.0-9000"]2017.07.01 11:17:09 INFO web[o.s.s.a.TomcatAccessLog] Web server is started2017.07.01 11:17:09 INFO web[o.s.s.a.EmbeddedTomcat] HTTP connector enabled on port 90002017.07.01 11:17:09 WARN web[o.s.p.ProcessEntryPoint] Fail to start web

java.lang.IllegalStateException: Webapp did not start

at org.sonar.server.app.EmbeddedTomcat.isUp(EmbeddedTomcat.java:84) ~[sonar-server-5.6.6.jar:na]

at org.sonar.server.app.WebServer.isUp(WebServer.java:47) [sonar-server-5.6.6.jar:na]

at org.sonar.process.ProcessEntryPoint.launch(ProcessEntryPoint.java:105) ~[sonar-process-5.6.6.jar:na]

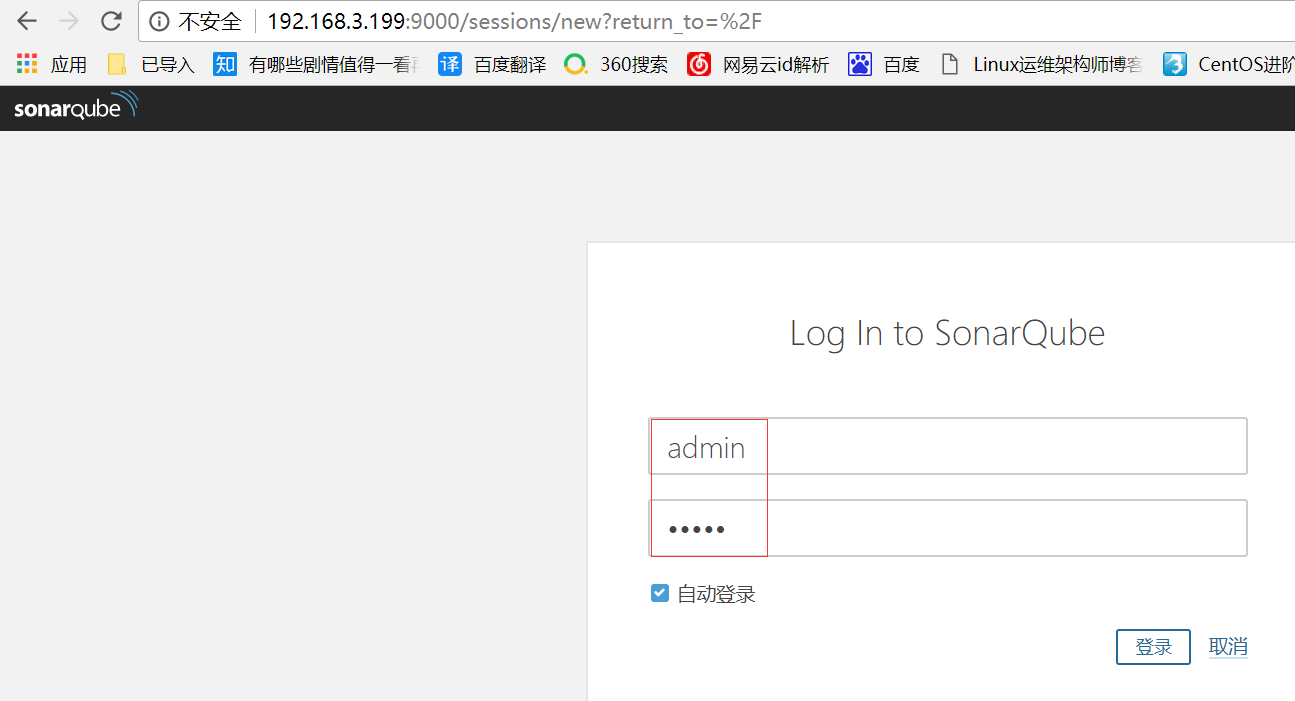
at org.sonar.server.app.WebServer.main(WebServer.java:68) [sonar-server-5.6.6.jar:na]2017.07.01 11:17:09 INFO web[o.a.c.h.Http11NioProtocol] Pausing ProtocolHandler ["http-nio-0.0.0.0-9000"]2017.07.01 11:17:16 INFO web[o.a.c.h.Http11NioProtocol] Stopping ProtocolHandler ["http-nio-0.0.0.0-9000"]2017.07.01 11:17:30 INFO web[o.a.c.h.Http11NioProtocol] Destroying ProtocolHandler ["http-nio-0.0.0.0-9000"]

IMG_257

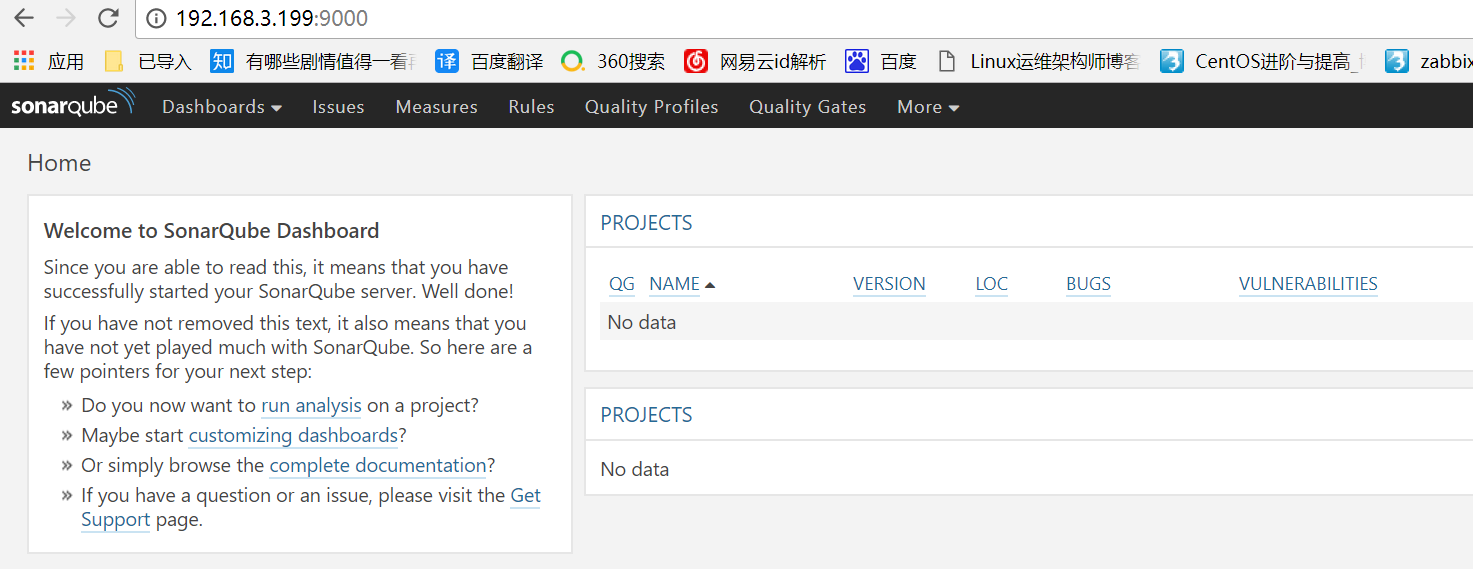
解决办法：  
升级mysql版本到5.6或者5.6以上

登陆：http://192.168.3.199:9000/

账号密码默认都是admin



#到此sonar就安装完成了，下一步将进入配置使用阶段。

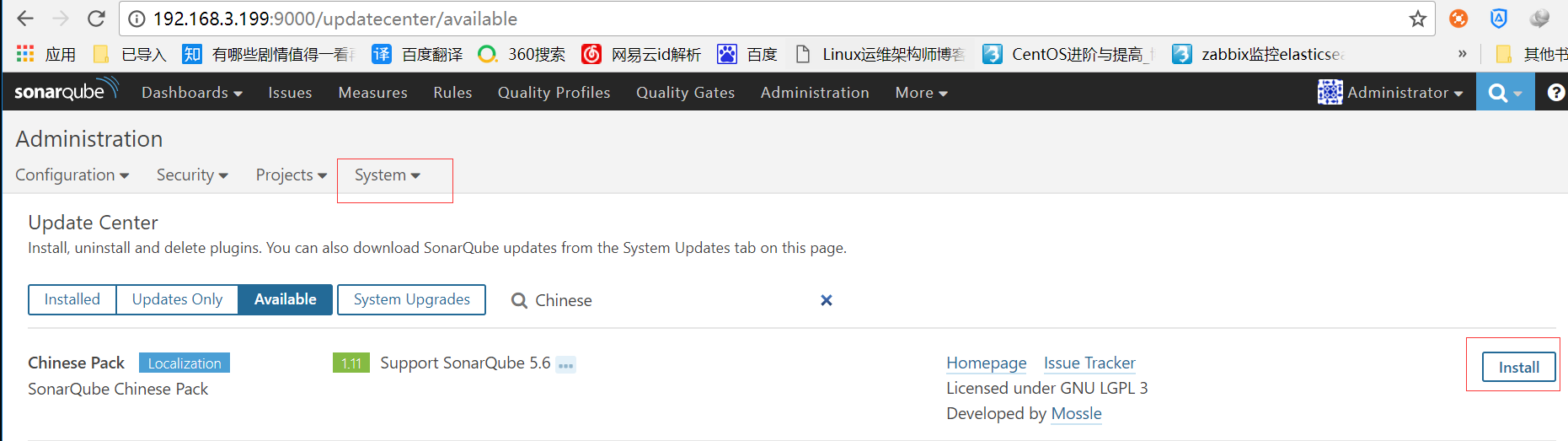


二：配置并使用sonar  
2.1：安装插件部分：  
2.1.1：默认的插件目录：

# ll /usr/local/sonarqube-5.6.6/extensions/plugins/

#如果在线安装插件不成功，可以把插件下载后放在此目录在重启sonar服务也可以实现安装插件，jenkins也可以通过此方式安装，另外jenkins还支持上传插件

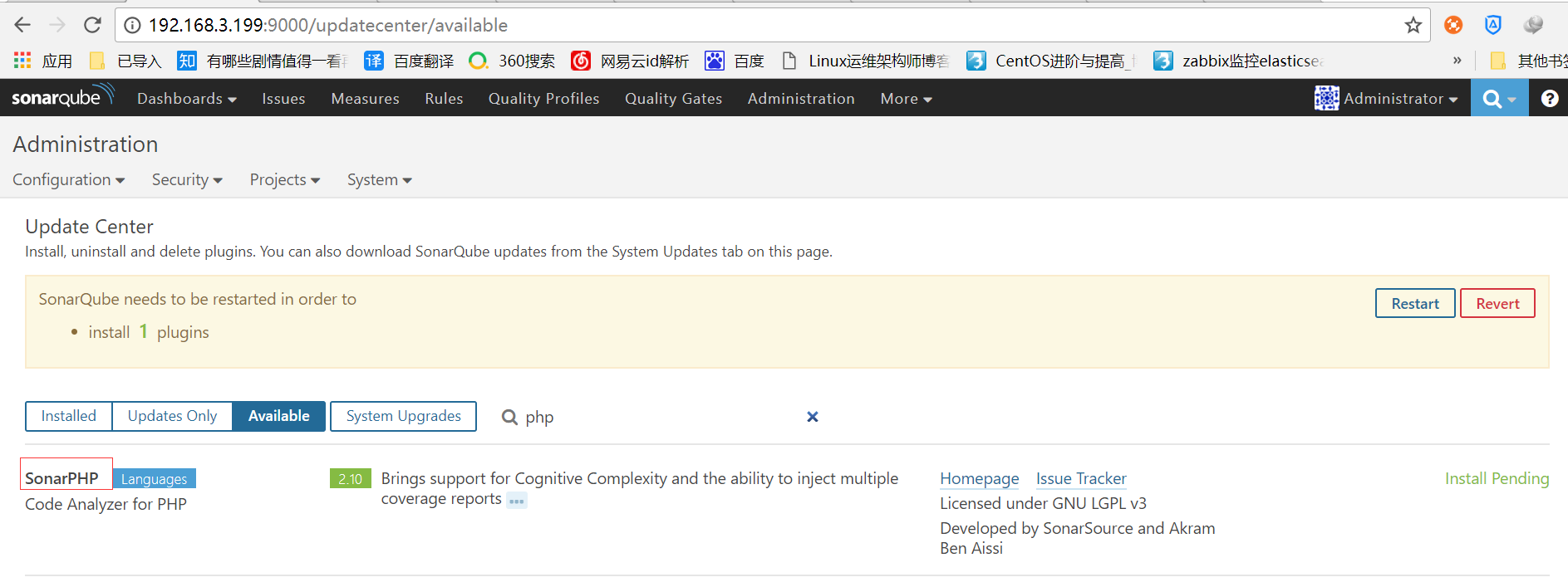
2.1.2：安装插件：  
administration-system-update center-available，在后面的搜索框搜索插件名称，然后点install安装：

  
或在插件目录/usr/local/sonarqube/extensions/plugins执行  
  
wget https://github.com/SonarQubeCommunity/sonar-l10n-zh/releases/download/sonar-l10n-zh-plugin-1.11/sonar-l10n-zh-plugin-1.11.jar(中文插件：)然后重启服务：

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

Sonar插件安装包下载地址：  
https://sonarsource.bintray.com/Distribution/

主要的是sonar对代码的分析是通过插件完成的，即分析java代码要安装java插件，分析php代码要安装php插件，分析什么语言就安装什么语言的插件  
本次我们安装了php、python、java语法检测插件



root@node1 plugins]# pwd

IMG_262

/usr/local/sonarqube/extensions/plugins

[root@node1 plugins]# ll

total 12312

-rw-r--r-- 1 root root 128 Feb 16 18:19 README.txt-rw-r--r-- 1 root root 4840602 Jul 4 17:05 sonar-java-plugin-4.11.0.10660.jar-rw-r--r-- 1 root root 3733262 Jul 4 17:05 sonar-php-plugin-2.10.0.2087.jar-rw-r--r-- 1 root root 4024311 Jul 4 17:05 sonar-python-plugin-1.8.0.1496.jar

IMG_263

2.1.3:代码检测测试,把sonar-scanner和sonarqube关联起来

sonar-scanner下载地址：https://sonarsource.bintray.com/Distribution/sonar-scanner-cli/

cd /usr/local/srcwget https://sonarsource.bintray.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-3.0.3.778-linux.zipunzip sonar-scanner-2.6.1.zip mv sonar-scanner-2.6.1 /usr/local/

cd /usr/local/ln -s sonar-scanner-2.6.1 sonar-scanner

配置sonar-scanner:

[root@node1 conf]# vim /usr/local/sonar-scanner/conf/sonar-scanner.properties

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

sonar.sourceEncoding=UTF-8

sonar.jdbc.username=sonar

sonar.jdbc.password=sonar@341Jpw

sonar.jdbc.url=jdbc:mysql://192.168.3.12:3306/sonar?useUnicode=true&amp;characterEncoding=utf8

检测PHP项目，需要在sonar中先安装SonarPHP插件，github搜索php-sonar-runner项目，对这个项目进行检测  
https://github.com/hasanyousuf/php-sonar-runner-unit-tests

unzip php-sonar-runner-unit-tests-master.zip

cd php-sonar-runner-unit-tests-master

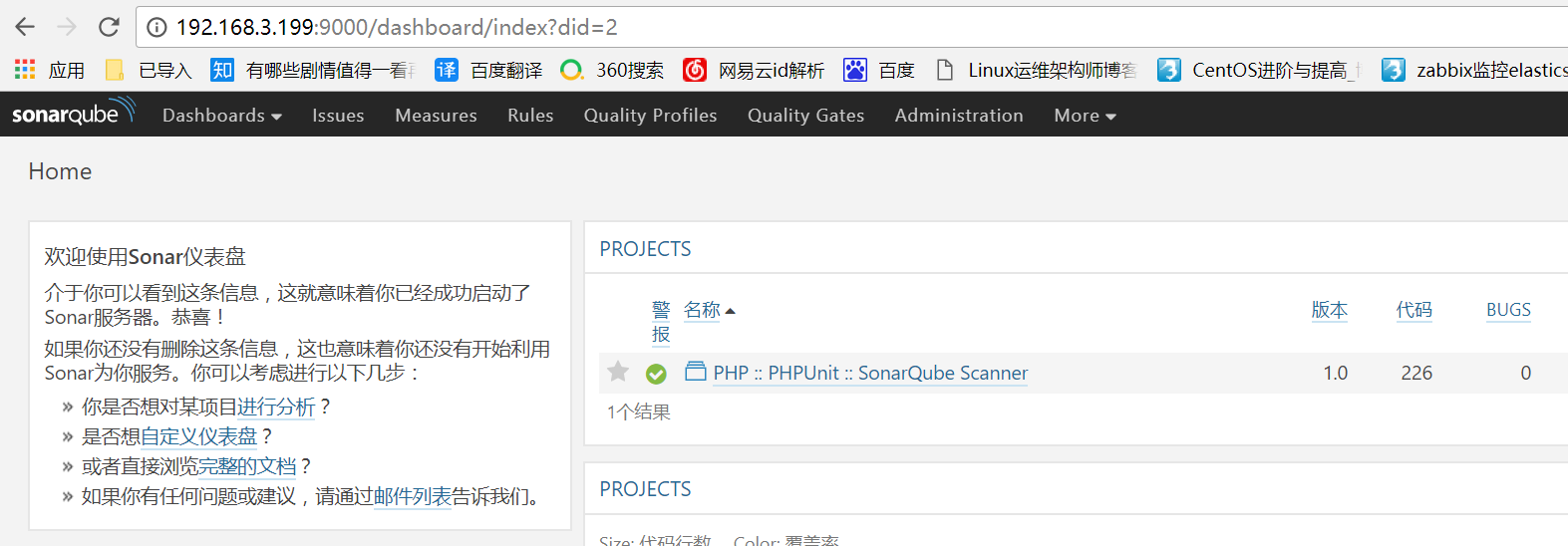
[root@node1 php-sonar-runner-unit-tests-master]# pwd

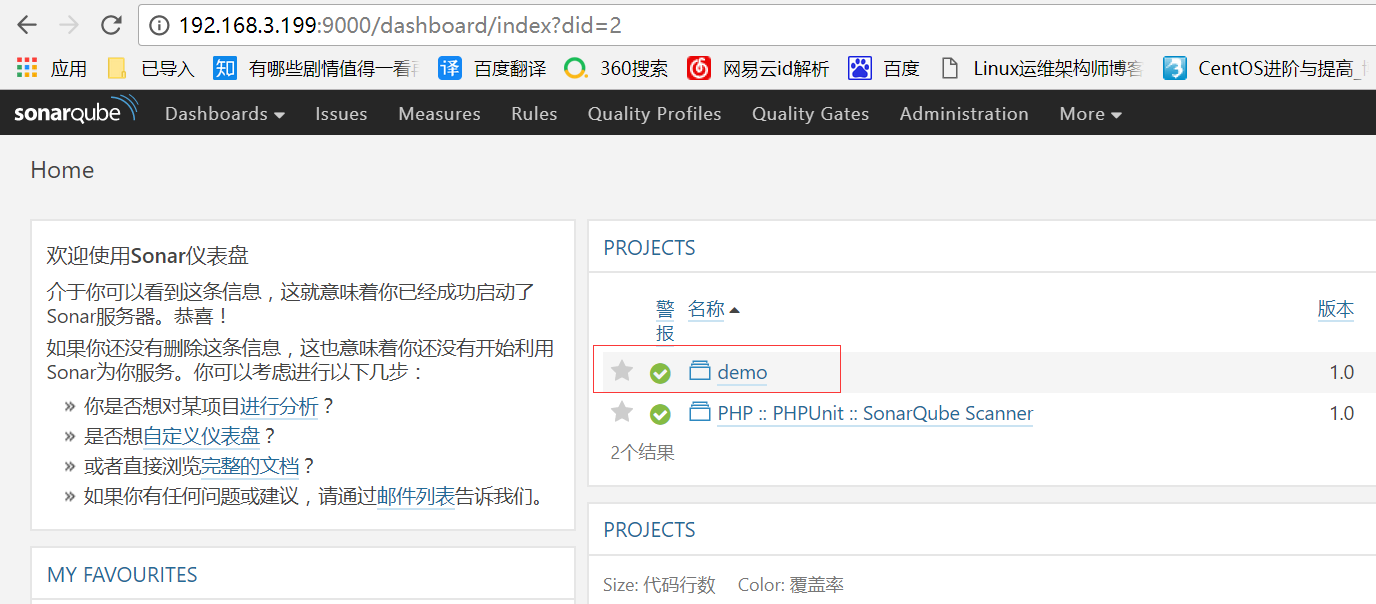
/root/php-sonar-runner-unit-tests-master

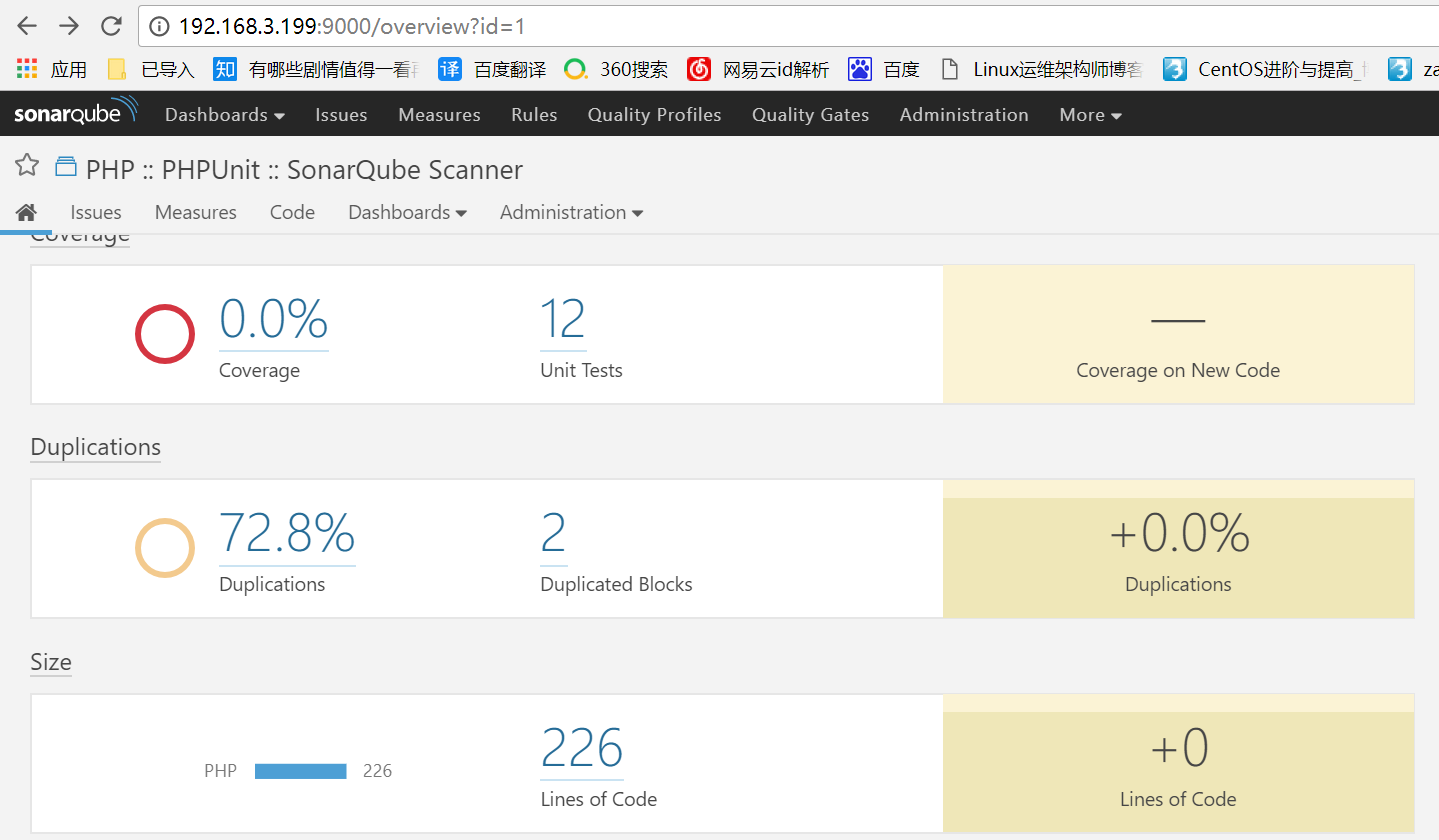
# 直接在php目录运行sonar-scanner,用于实现代码质量分析

[root@node1 php-sonar-runner-unit-tests-master]# /usr/local/sonar-scanner/bin/sonar-scanner

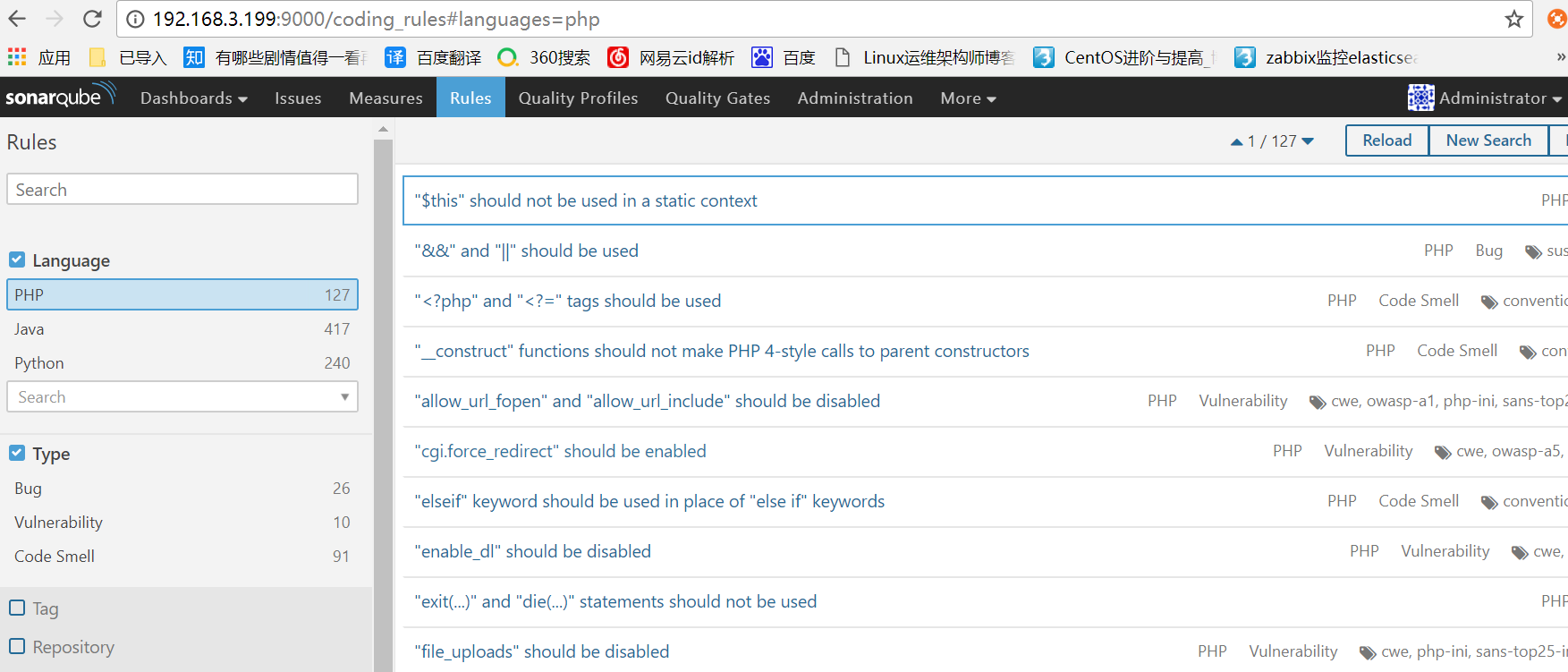
在sonar管理界面查看扫描结果：  
dashboard --> home 点项目名称可以查看更具体的信息







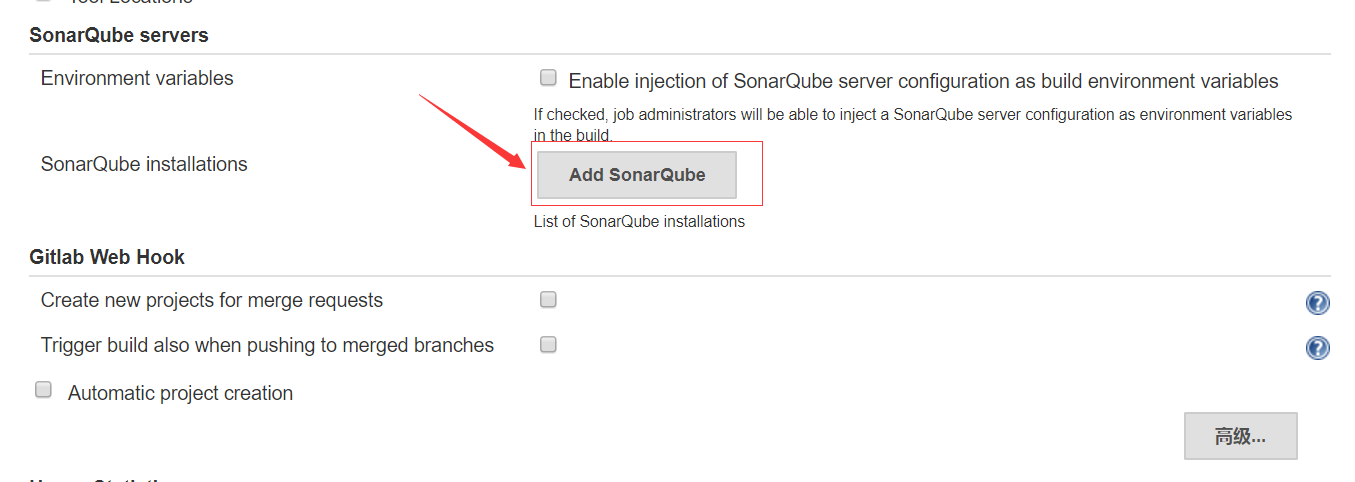
2.1.4：代码规则：



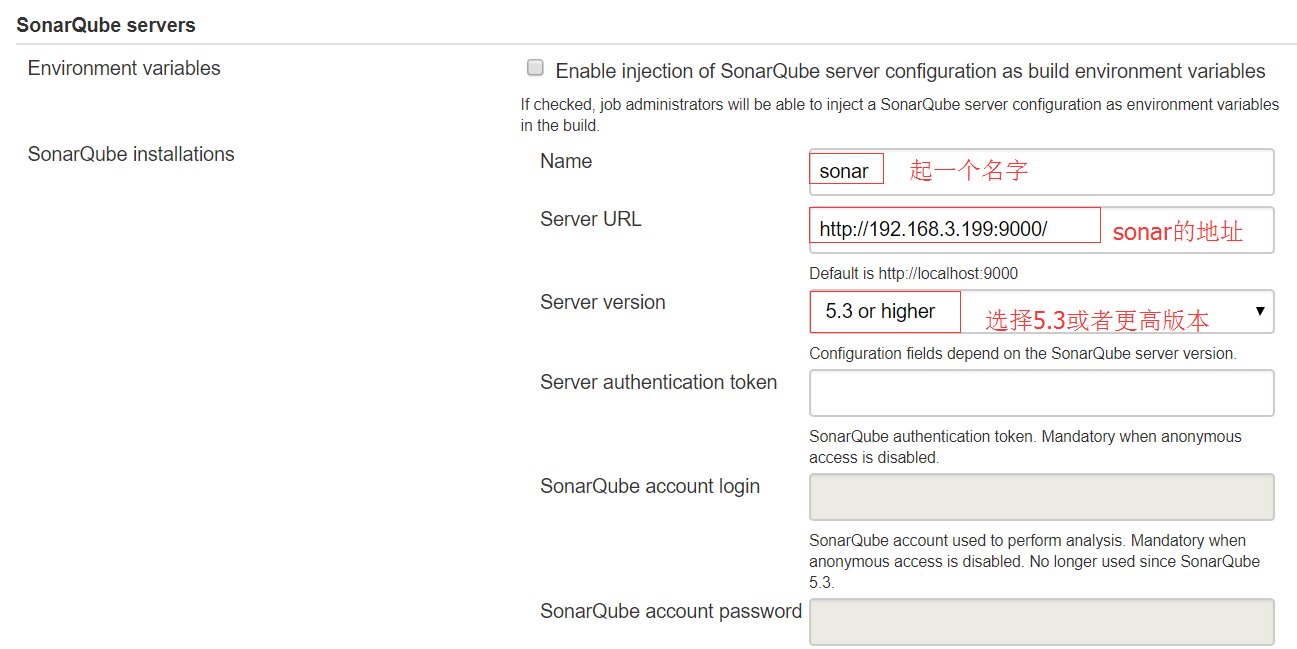
2.4：如何让jenkins关联到sonar scanner？  
有两种方式保存配置文件，一是保存在项目里面，二是在jenkins管理界面进行配置：  
2.4.1：在jenkins插件安装界面安装SonarQuebe Scanner for Jenkins插件：



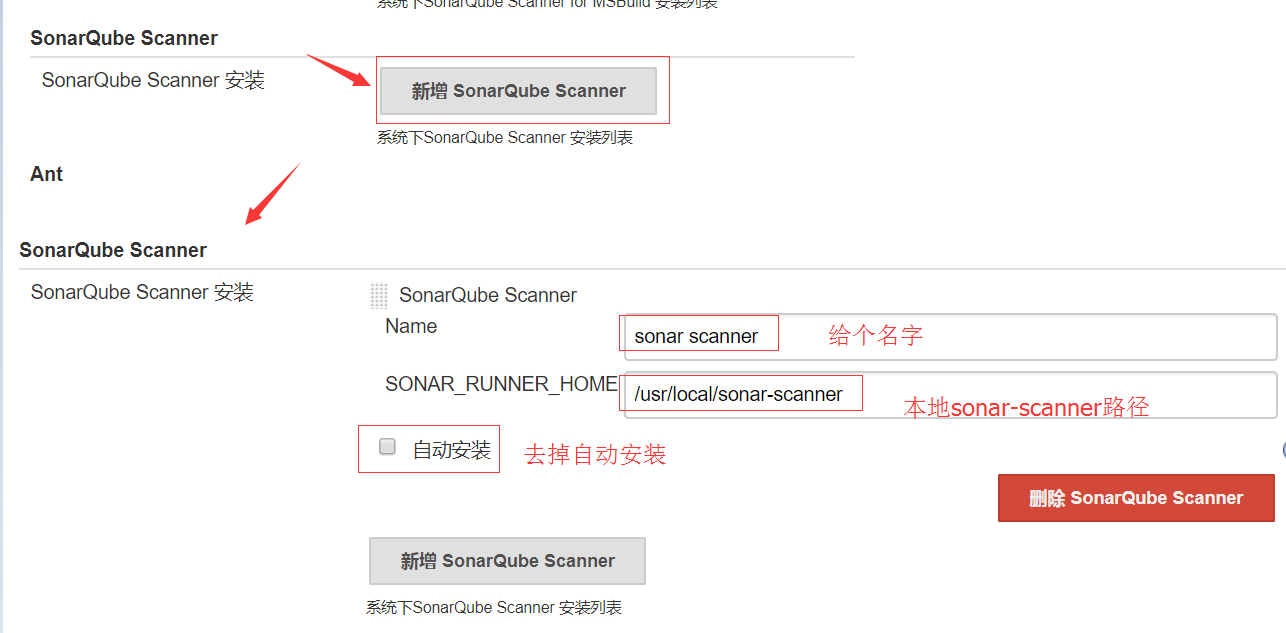
2.4.2将jenkins关联sonar：  
jenkins中操作：系统管理-系统设置，找到 SonarQube servers 部分



添加sonar访问地址，然后点保存



2.4.3添加扫描器：  
2.4.3.1：#系统管理–global-tool-ocnfigration --> 添加本地sonar scanner，然后点保存



2.5：配置jenkins项目构建操作：

2.5.1:复制之前sonar scanner的代码检测配置文件内容，如：

IMG_272

# cat /root/php-sonar-runner-unit-tests-master/sonar-project.properties

sonar.projectKey=org.sonarqube:php-ut-sq-scanner

sonar.projectName=PHP :: PHPUnit :: SonarQube Scanner

sonar.projectVersion=1.0

sonar.sources=src

sonar.tests=tests

sonar.language=php

sonar.sourceEncoding=UTF-8

IMG_273

2.5.1：选择自己的项目（web-demo）-构建触发器-构建-execute sonarqube scanner，将配置文件的内容修改成如下格式填写完成后点保存：

sonar.projectKey=web-demo

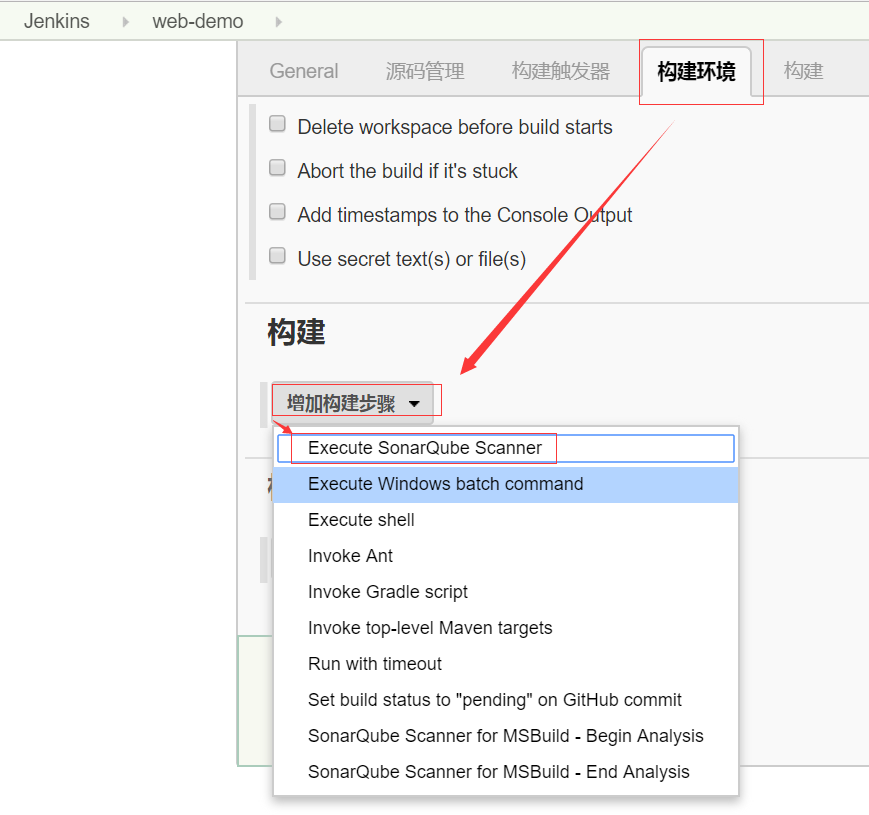
sonar.projectName=web-demo

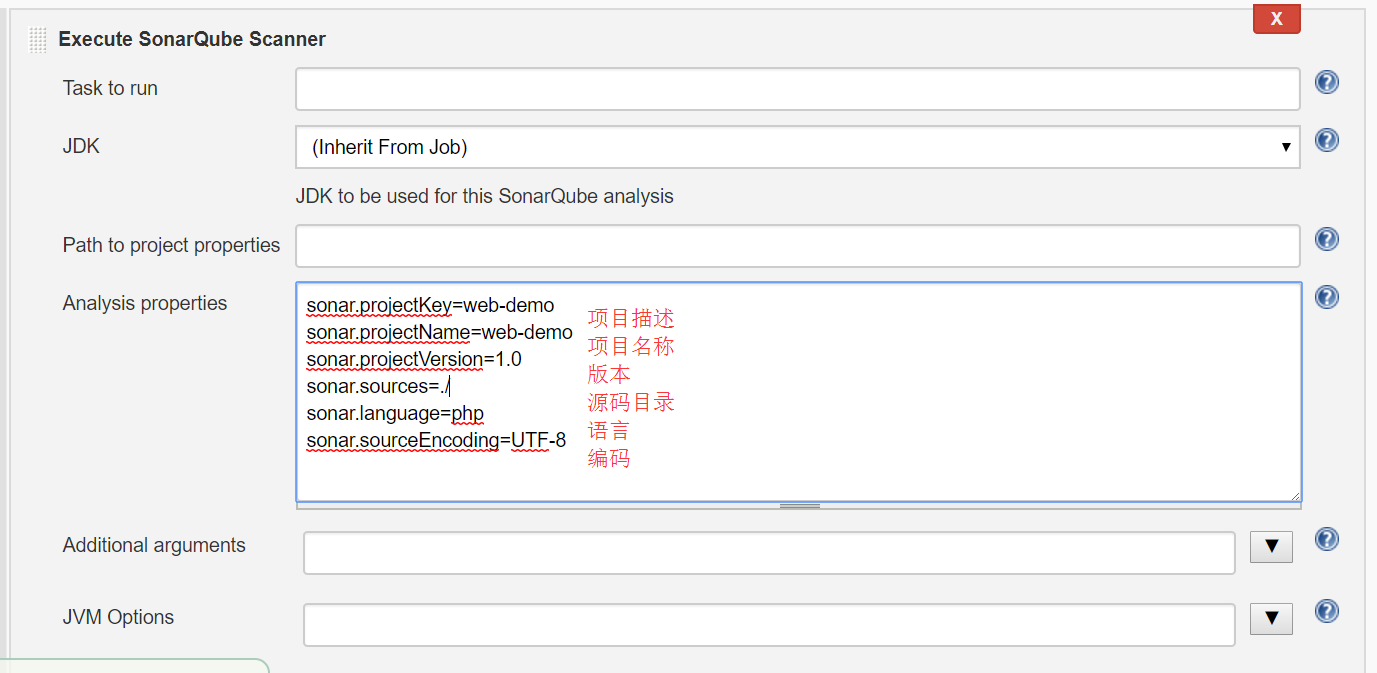
sonar.projectVersion=1.0

sonar.sources=./

sonar.language=php

sonar.sourceEncoding=UTF-8





可以看到，右边多了个快捷方式

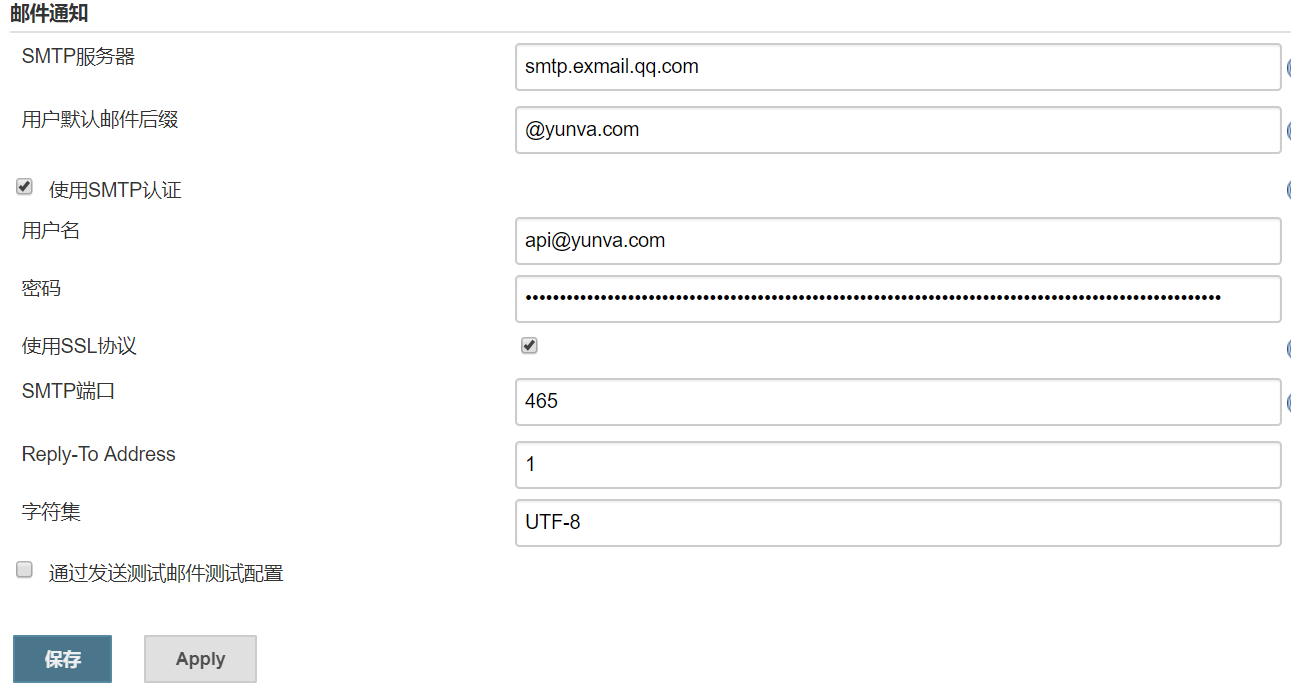


2.6：测试jenkins项目构建：  
2.6.1：在jenkins选择自己的项目点击立即构建，以下是构建成功的界面：



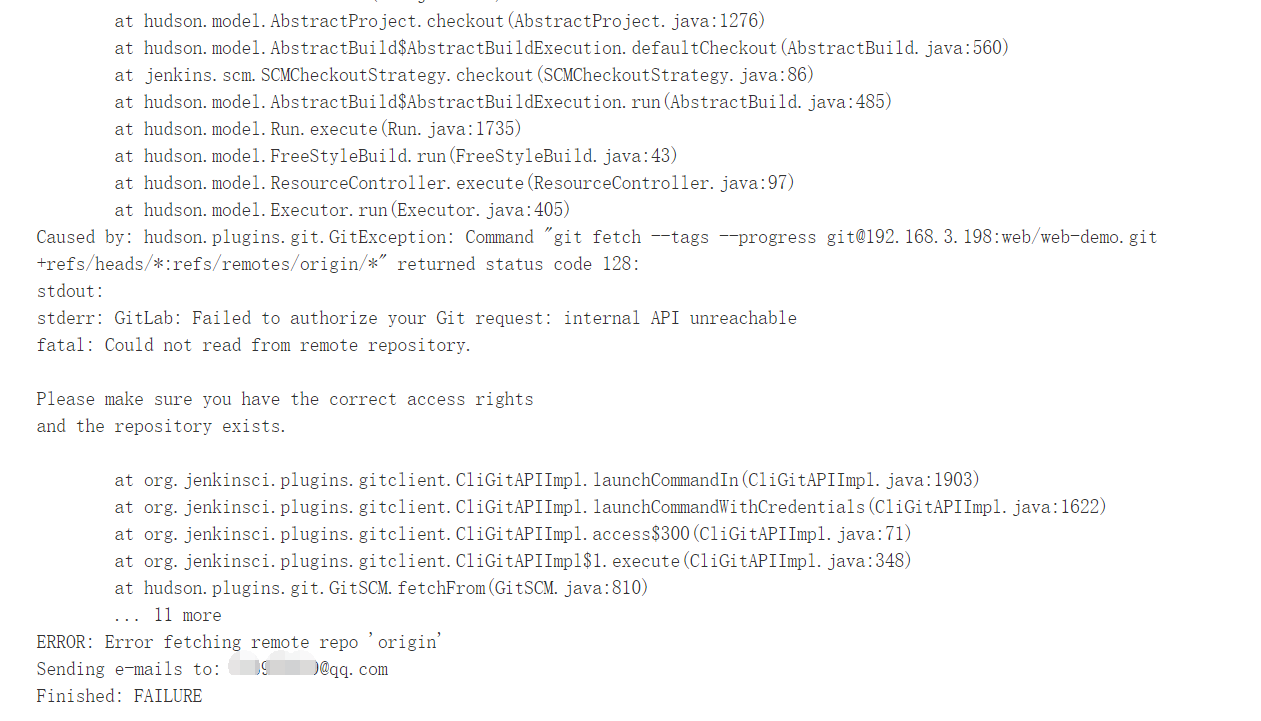
2.6.2：在sonar查看是否有代码扫质量分析结果：

2.7：添加构建后操作  
2.7.1:添加邮件通知，当构建失败后向指定的邮箱通知失败信息：  
#发件箱设置：





2.7.2：将github服务关闭，然后构建项目，由于git服务无法访问所以肯定会导致项目构建失败触发邮件通知：  
2.7.2.1：关闭git服务：  
# gitlab-ctl stop  
2.7.2.2:构建项目，以下是构建失败的控制台输出信息：



2.7.2.3：以下是失败的邮件通知：

