第1章 安装mysql

1.下载mysql 5.7

下载地址

```
https://downloads.mysql.com/archives/community/
https://downloads.mysql.com/archives/get/p/23/file/mysql-5.7.28-linux-glibc2.12-x86_64.tar.gz
```

2.创建目录

```
mkdir -p /data/soft
mkdir -p /data/mysql_3306/
```

3.下载并解压软件

```
cd /data/soft
wget https://downloads.mysql.com/archives/get/p/23/file/mysql-5.7.28-linux-glibc2.12-x86_64.tar.gz
tar zxf mysql-5.7.28-linux-glibc2.12-x86_64.tar.gz -C /opt/
mv /opt/mysql-5.7.28-linux-glibc2.12-x86_64 /opt/mysql-5.7.28
ln -s /opt/mysql-5.7.28 /opt/mysql
```

4.设置环境变量

```
echo "export PATH=$PATH:/opt/mysql/bin" >>/etc/profile
source /etc/profile
mysql -V
```

5.清除遗留环境

```
rpm -qa|grep mariadb
yum remove mariadb-libs -y
rm -rf /etc/my.cnf
```

6.安装mysql依赖包

```
yum install -y libaio-devel
```

7.创建mysql普通用户并授权

```
useradd -s /sbin/nologin -M mysql
chown -R mysql.mysql /data/
chown -R mysql.mysql /opt/mysql*
```

8.初始化数据库

```
mysqld --initialize-insecure --user=mysql --basedir=/opt/mysql --
datadir=/data/mysql_3306/
```

9.编辑mysql配置文件

```
cat> /etc/my.cnf <<EOF
[mysqld]
user=mysql
basedir=/opt/mysql
datadir=/data/mysql_3306
socket=/tmp/mysql.sock
[mysql]
socket=/tmp/mysql.sock
EOF
```

10.准备启动脚本并启动数据库

```
cp /opt/mysql/support-files/mysql.server  /etc/init.d/mysqld
chkconfig --add mysqld
systemctl start mysqld
netstat -lntup|grep 3306
```

11.前台启动

```
/opt/mysql/bin/mysqld --basedir=/opt/mysql --datadir=/data/mysql_3306 --plugin-
dir=/opt/mysql/lib/plugin --user=mysql --log-error=/data/mysql_3306/log/mysql.log --
pid-file=/data/mysql_3306/db01.pid --socket=/tmp/mysql.sock
```

12.修改root密码

```
mysqladmin password
```

13.登陆mysql

```
mysql -uroot -p123456
```

第2章 安装SonarQube

1.安装java环境

```
yum install java -y
```

2.解压并创建软链接

```
unzip sonarqube-7.0.zip -d /opt/
ln -s /opt/sonarqube-7.0/ /opt/sonarqube
```

3.创建普通用户并更改授权

```
useradd sonar -M -s /sbin/nologin
chown -R sonar.sonar /opt/sonarqube*
```

4.配置sonarqube数据库连接信息

```
[root@sonar ~]# vim /opt/sonarqube/conf/sonar.properties
sonar.jdbc.username=root
sonar.jdbc.password=123456
sonar.jdbc.url=jdbc:mysql://localhost:3306/sonar?
useUnicode=true&characterEncoding=utf8&rewriteBatchedStatements=true&useConfigs=maxP
erformance&useSSL=false
```

5.指定sonarqube启动用户

```
vim /opt/sonarqube/bin/linux-x86-64/sonar.sh
RUN_AS_USER=sonar
```

6.创建sonarqube数据库

```
mysql -uroot -p123456 -e 'create database sonar default character set utf8;'
mysql -uroot -p123456 -e 'show databases;'
```

7.编写systemd启动文件

```
cat >/usr/lib/systemd/system/sonar.service<<'EOF'
[Unit]
Description=sonar

[Service]
```

```
ExecStart=/opt/sonarqube/bin/linux-x86-64/sonar.sh start
ExecStop=/opt/sonarqube/bin/linux-x86-64/sonar.sh stop
Type=forking
User=sonar
Group=sonar

[Install]
WantedBy=multi-user.target
EOF
systemctl daemon-reload
```

8.启动SonarQube

```
systemctl start sonar.service
```

9.检查服务

```
[root@sonar ~]# netstat -lntup|grep java
tcp        0      0 127.0.0.1:32000         0.0.0.0:*               LISTEN
18202/java
tcp6       0      0 :::9000                 :::*                    LISTEN
18305/java
tcp6       0      0 127.0.0.1:9001          :::*                    LISTEN
18227/java
tcp6       0      0 127.0.0.1:36949         :::*                    LISTEN
18475/java
```

10.启动报错

使用systemd启动后失败，查看es日志发现提示max file descriptors太低：

```
[root@sonar /opt/sonarqube/logs]# tail -f /opt/sonarqube/logs/es.log
2020.05.14 09:51:19 INFO  es[][o.e.t.TransportService] publish_address
{127.0.0.1:9001}, bound_addresses {127.0.0.1:9001}
2020.05.14 09:51:19 WARN  es[][o.e.b.BootstrapChecks] max file descriptors [4096]
for elasticsearch process is too low, increase to at least [65536]
2020.05.14 09:51:19 WARN  es[][o.e.b.BootstrapChecks] max virtual memory areas
vm.max_map_count [65530] is too low, increase to at least [262144]
```

解决方法：

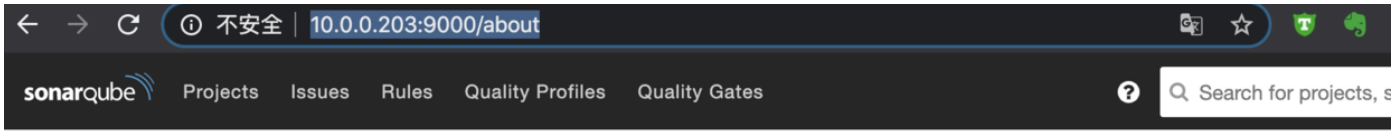
```
echo "vm.max_map_count=262144" >> /etc/sysctl.conf
echo "root             -       nofile          65536"  >> /etc/security/limits.conf
echo "sonar            -       nofile          65536"  >> /etc/security/limits.conf
sysctl -p
```

第3章 初始化SonarQube

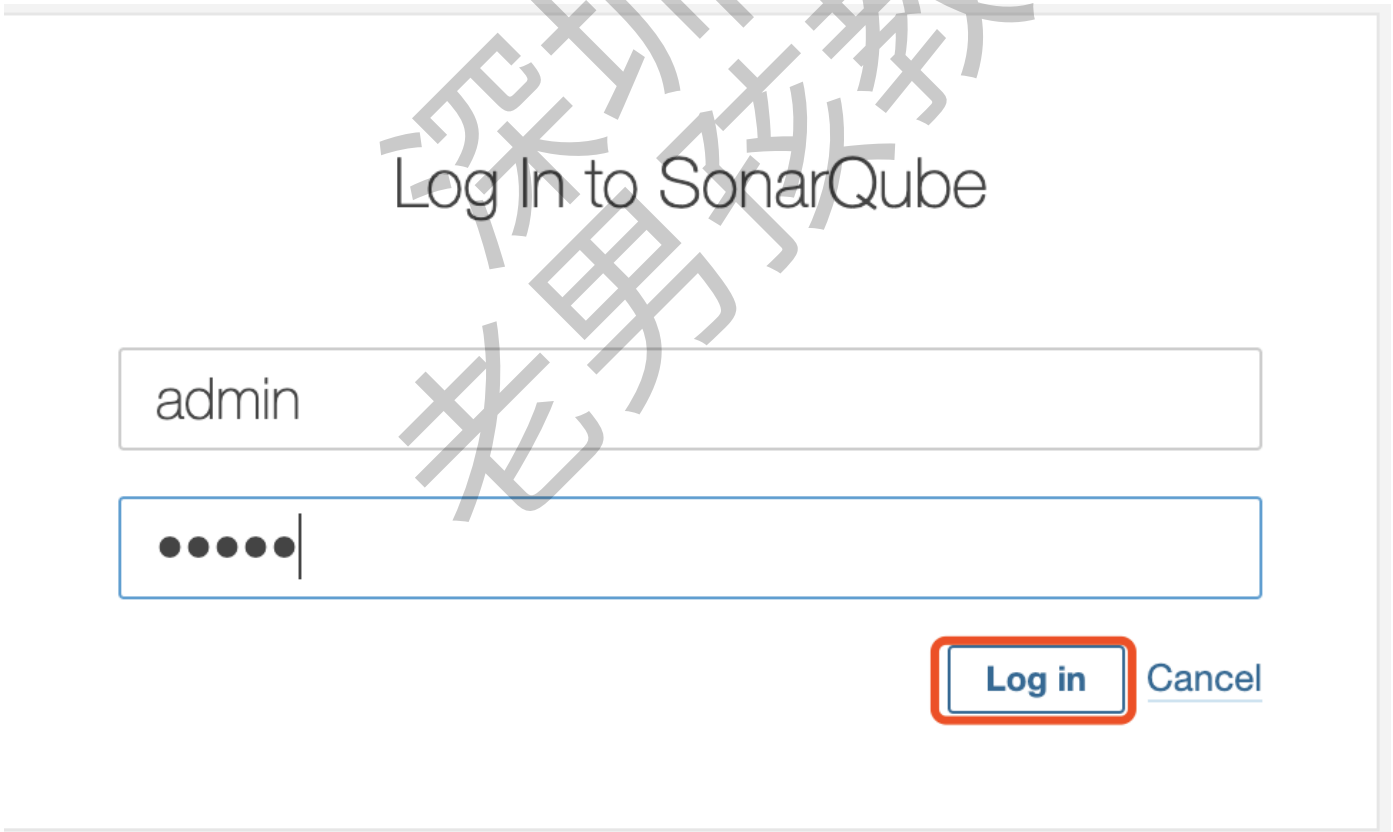
1.使用admin登陆

登陆地址为:

```
http://10.0.0.203:9000/about
```

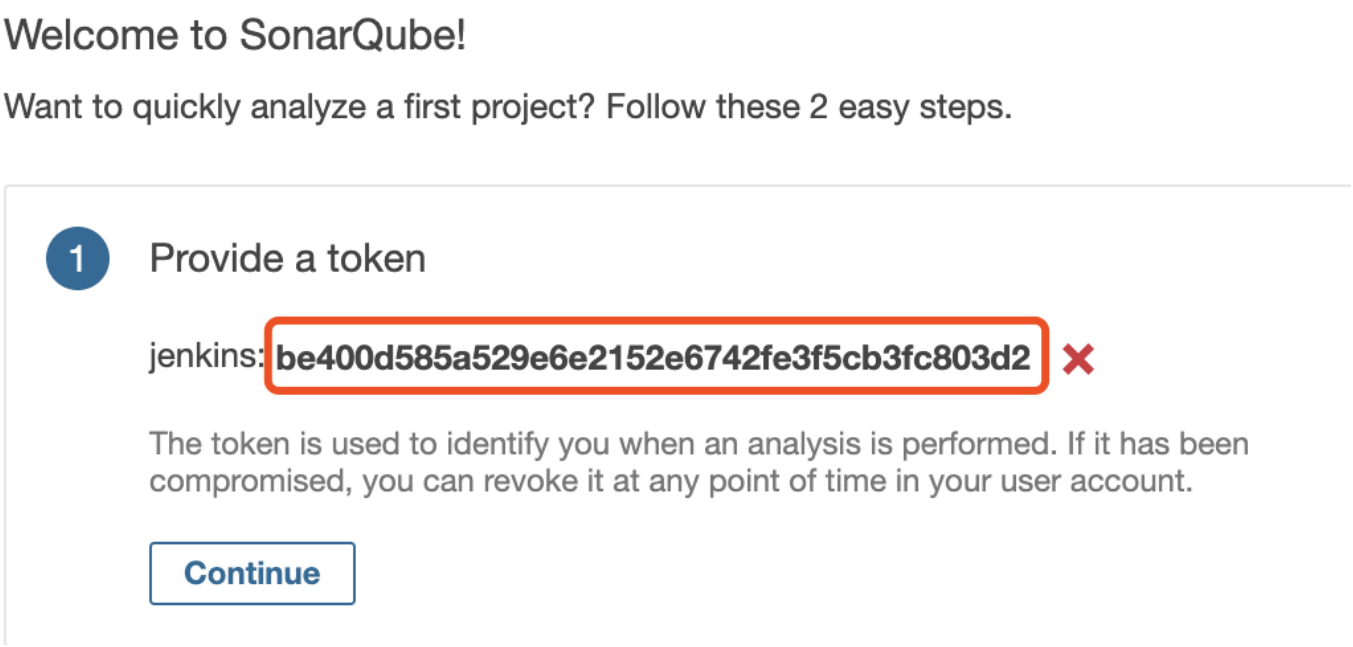


账号密码均为admin:



2.生成token

在输入框内输入jenkins，然后点击Generate生成token，需要保存好这个tocken，后面会用到



3.选择项目类型

点击Done之后会给我们生成提示信息：

第4章 安装插件

1.在线安装中文插件方法

Administration-->Marketplace-->chinese-->install

2.离线安装插件方法

离线安装的话只需要将插件压缩包解压到指定目录然后重启服务即可，解压之前可以先备份插件目录

```
mv /opt/sonarqube/extensions/plugins/ /opt/sonarqube/extensions/plugins_bak
tar xf sonar_plugins.tar.gz -C /opt/sonarqube/extensions/
```

3.重启服务

```
systemctl restart sonar.service
```

4.浏览器访问查看

第5章 安装客户端

1.jenkins主机安装客户端

我们需要将jenkins拉取的代码推送到SonarQube,所以需要在jenkins主机上安装sonar客户端：

```
unzip sonar-scanner-cli-4.0.0.1744-linux.zip -d /opt/
cd /opt/
ln -s sonar-scanner-4.0.0.1744-linux sonar-scanner
```

写入环境变量：

```
echo 'export PATH=$PATH:/opt/sonar-scanner/bin' >> /etc/profile
source /etc/profile
```

2.推送代码到SonarQube

进入代码目录执行推送命令

注意：这里的推送命令是初始化的时候生成的，Dsonar.login的值也是初始化时候生成的token

```
cd /var/lib/jenkins/workspace/h5game/
/opt/sonar-scanner/bin/sonar-scanner  \
  -Dsonar.projectKey=html \
  -Dsonar.sources=. \
  -Dsonar.host.url=http://10.0.0.203:9000 \
  -Dsonar.login=4f57dfb332463fa8220be49856a0f1d27c88a142
```

我们也可以将服务器相关的命令写入配置文件里，这样推送的命令可以精简一些：

```
vim /opt/sonar-scanner/conf/sonar-scanner.properties
sonar.host.url=http://10.0.0.203:9000
sonar.login=be400d585a529e6e2152e6742fe3f5cb3fc803d2
sonar.sourceEncoding=UTF-8
```

然后推送命令只需要指定两个选项即可：

```
cd /var/lib/jenkins/workspace/my-freestyle-job/
sonar-scanner \
  -Dsonar.projectKey=html \
  -Dsonar.sources=.
```

3.web页面查看扫描结果

4.执行报错解决

报错现象：推送的时候提示我们找不到node环境

```
INFO: Sensor SonarCSS Metrics [cssfamily]
INFO: Sensor SonarCSS Metrics [cssfamily] (done) | time=788ms
INFO: Sensor SonarCSS Rules [cssfamily]
ERROR: Error when running: 'node -v'. Is Node.js available during analysis? No CSS files will be analyzed.
org.sonarsource.nodejs.NodeCommandException: Error when running: 'node -v'. Is Node.js available during analysis?
        at org.sonarsource.nodejs.NodeCommand.start(NodeCommand.java:77)
        at org.sonarsource.nodejs.NodeCommandBuilderImpl.getVersion(NodeCommandBuilderImpl.java:171)
        at org.sonarsource.nodejs.NodeCommandBuilderImpl.checkNodeCompatibility(NodeCommandBuilderImpl.java:144)
        at org.sonarsource.nodejs.NodeCommandBuilderImpl.build(NodeCommandBuilderImpl.java:120)
        at org.sonar.css.plugin.StylelintCommandProvider.nodeCommand(StylelintCommandProvider.java:64)
        at org.sonar.css.plugin.CssRuleSensor.execute(CssRuleSensor.java:111)
```

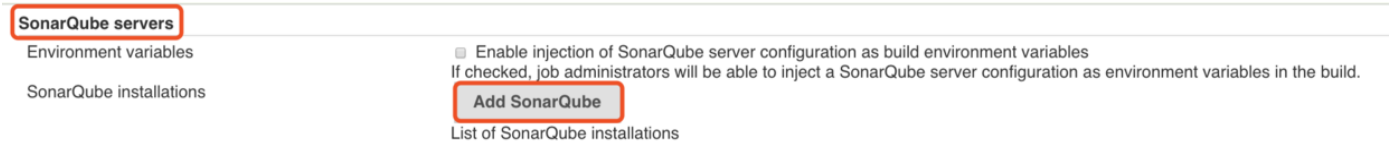
解决方法：在jenkins服务器上安装nodejs环境,然后重新推送就不会再报错了：

```
cd /opt/
wget https://nodejs.org/dist/v12.13.0/node-v12.13.0-linux-x64.tar.xz
tar xf node-v12.13.0-linux-x64.tar.xz
mv node-v12.13.0-linux-x64 node
echo 'export PATH=$PATH:/opt/node/bin' >> /etc/profile
source /etc/profile
npm -v
node -v
```

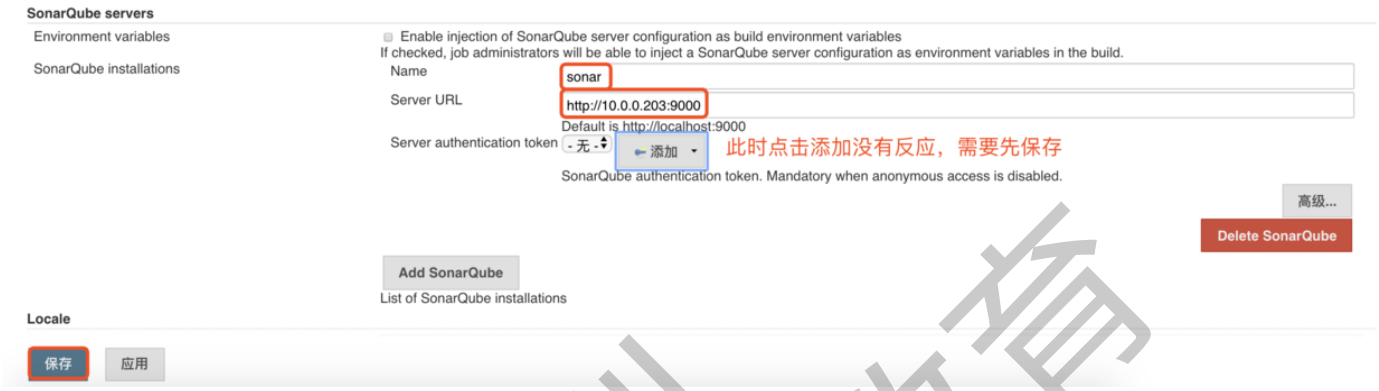
第6章 与Jenkins集成

1.配置SonarQube凭证信息

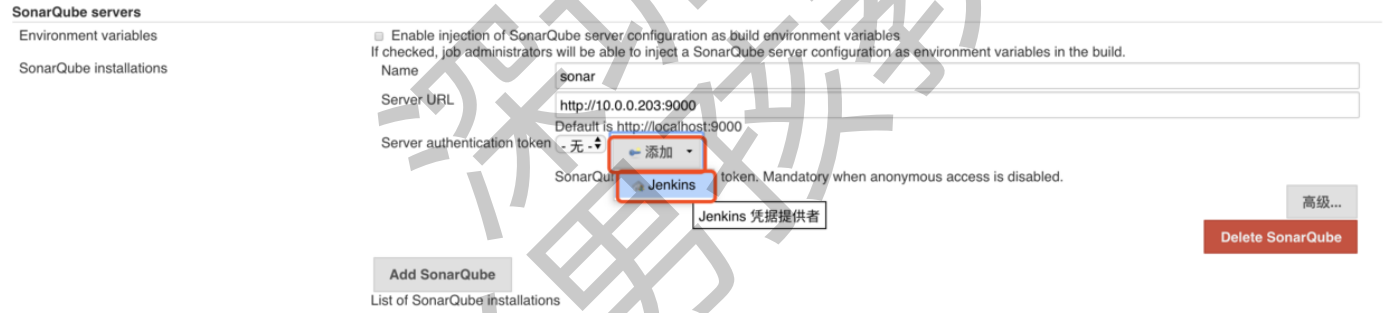
在jenkins页面进入 系统管理-->系统配置-->找到sonar的配置



填写sonar服务器信息：

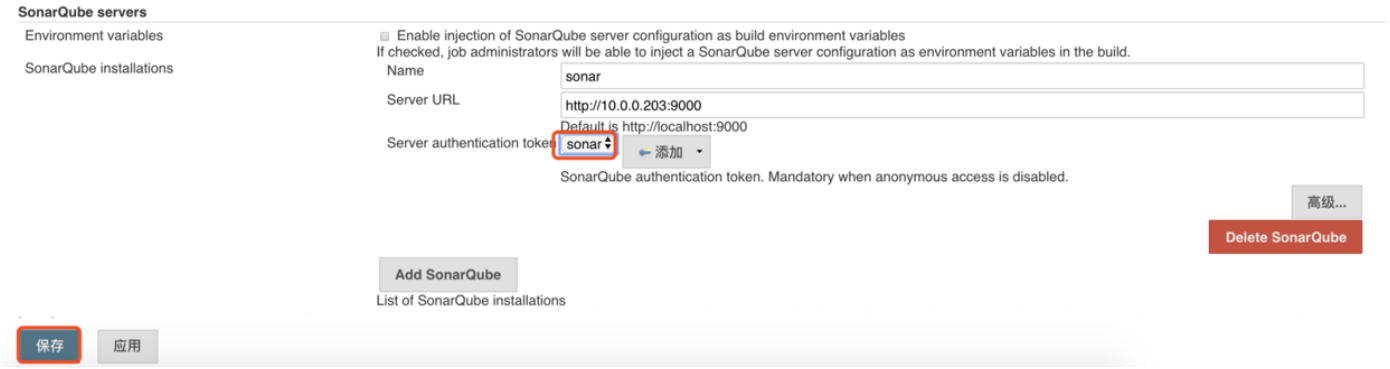


此时点击添加凭证按钮没有反应，没关系，先保存一下，然后回来再添加一次即可：



填写sonar初始化的token信息：

添加完成后就可以选择sonar的凭证了：

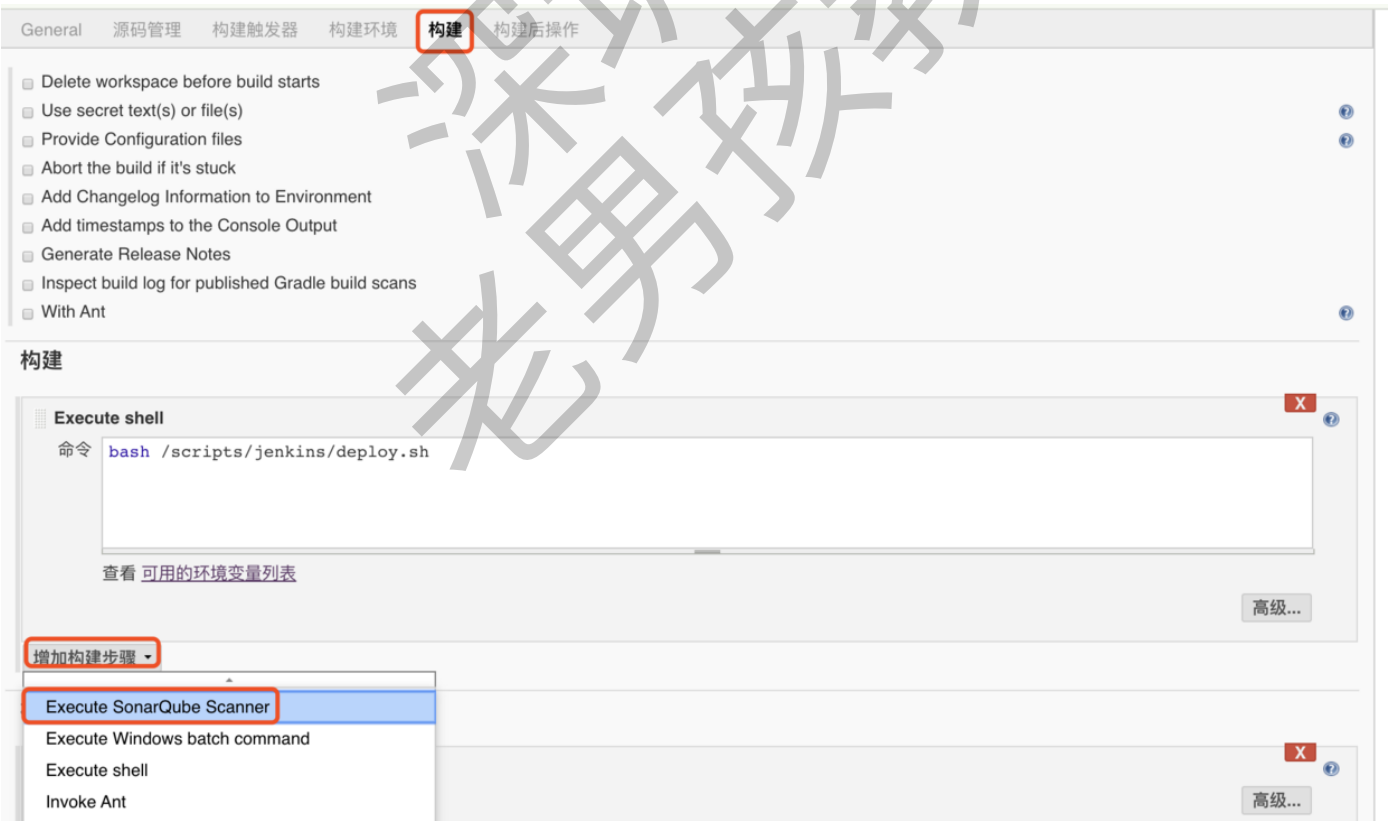


2.配置sonar客户端家目录

点击系统管理-->全局工具配置-->找到SonarQube Scanner选项：

3.工程中配置sonar构建选项

添加构建步骤：



填写详细信息：

填写参数：

```
sonar.projectName=${JOB_NAME}
sonar.projectKey=html
sonar.sources=.
```

4.调整构建执行顺序

注意！！！这里我们还需要将构建顺序调整一下，先执行代码扫描，然后再发布版本

可以直接拖动选项块来调整顺序，最终结果如下：

5.发布测试

我们可以使用git尝试发布代码，然后查看执行是否成功：

```
git branch
git pull
vim index.html
git add .
git commit -m "v5.0 稳定版"
git push -u origin master
```

sonar查看是否发布：