# 整体说明

配置环境变量只配置当前用户的环境变量 vi ~/.bash\_profile 然后使用source 命令使修改生效 source ~/.bash\_profile

# JDK安装

Java环境变量配置

export JAVA\_HOME=/home/jenkins/soft/java/jdk1.8.0\_11

export JRE\_HOME=${JAVA\_HOME}/jre

export CLASSPATH=.:${JAVA\_HOME}/lib:${JRE\_HOME}/lib

export PATH=${JAVA\_HOME}/bin:$PATH

# maven安装

Maven环境变量配置

export MAVEN\_HOME=/home/jenkins/soft/maven/apache-maven-3.5.3

export PATH=${MAVEN\_HOME}/bin:${PATH}

# Ant安装

Ant环境变量配置

export ANT\_HOME=/home/jenkins/soft/ant/apache-ant-1.10.2

/home/jenkins/soft/ant/apache-ant-1.10.2

# 综合配置

export JAVA\_HOME=/home/jenkins/soft/java/jdk1.8.0\_11

export JRE\_HOME=${JAVA\_HOME}/jre

export CLASSPATH=.:${JAVA\_HOME}/lib:${JRE\_HOME}/lib

#export PATH=${JAVA\_HOME}/bin:$PATH

export MAVEN\_HOME=/home/jenkins/soft/maven/apache-maven-3.5.3

export ANT\_HOME=/home/jenkins/soft/ant/apache-ant-1.10.2

export PATH=${JAVA\_HOME}/bin:${MAVEN\_HOME}/bin:${ANT\_HOME}/bin:$PATH

# Linux Jenkins安装

## Master安装说明

### 下载repo文件

sudo wget -O /etc/yum.repos.d/jenkins.repo <http://pkg.jenkins-ci.org/redhat/jenkins.repo>

### 导入密钥

sudo rpm --import <http://pkg.jenkins-ci.org/redhat/jenkins-ci.org.key>

### 安装jenkins

sudo yum install Jenkins

### 配置端口

Jenkins默认端口为8080 如需修改

sudo vi /etc/sysconfig/jenkins

查找到JENKINS\_PORT修改对应的端口值

### 启动jenkins

/sbin/service jenkins start|stop|restart

### 配置jenkins

复制密码下一步

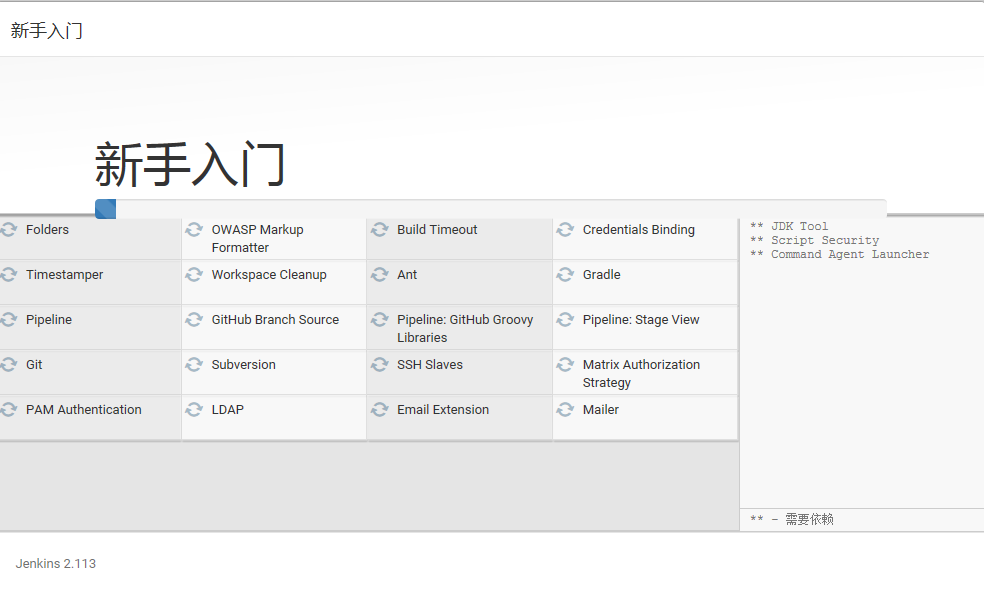
vi /var/lib/jenkins/secrets/initialAdminPassword 

### 安装插件

选择安装推荐的插件（联网情况下）



插件安装中



### 创建第一个用户

创建用户 jenkins/aspire+888 全名称为 Jenkins-robot

电子邮箱: [Jenkins-robot@aspirecn.com](mailto:Jenkins-robot@aspirecn.com)

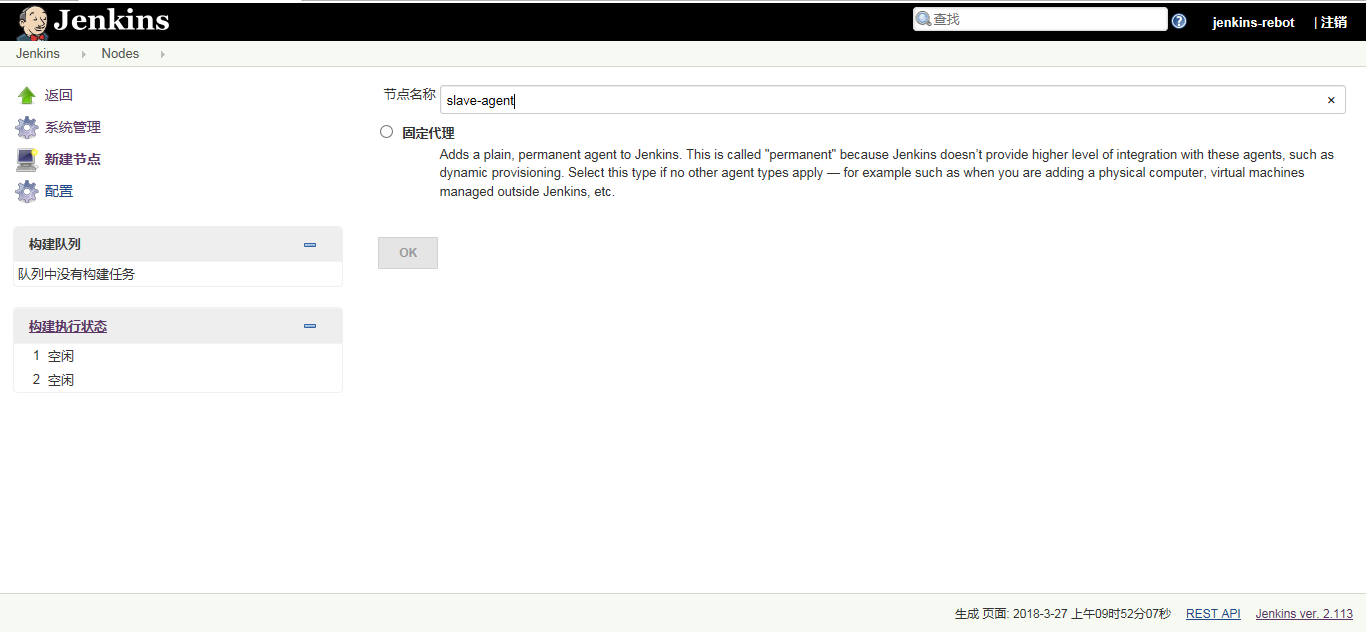
### 进入jenkins



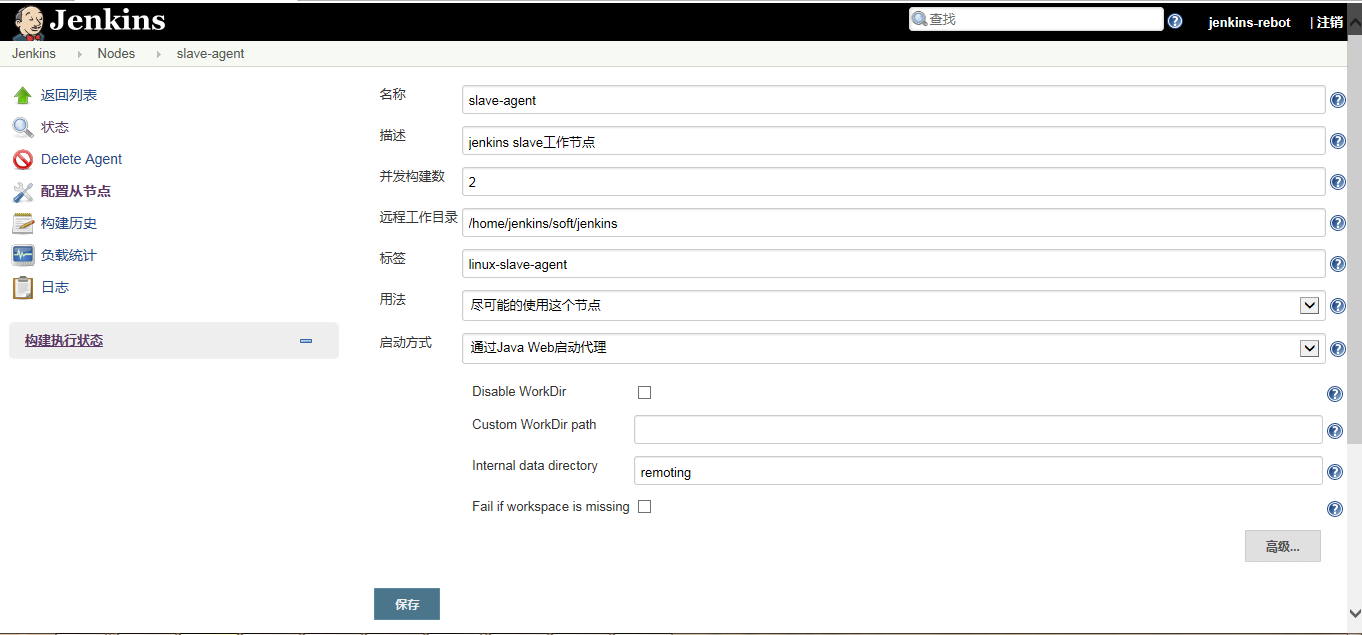
## Slave 安装说明

### 新增slave节点

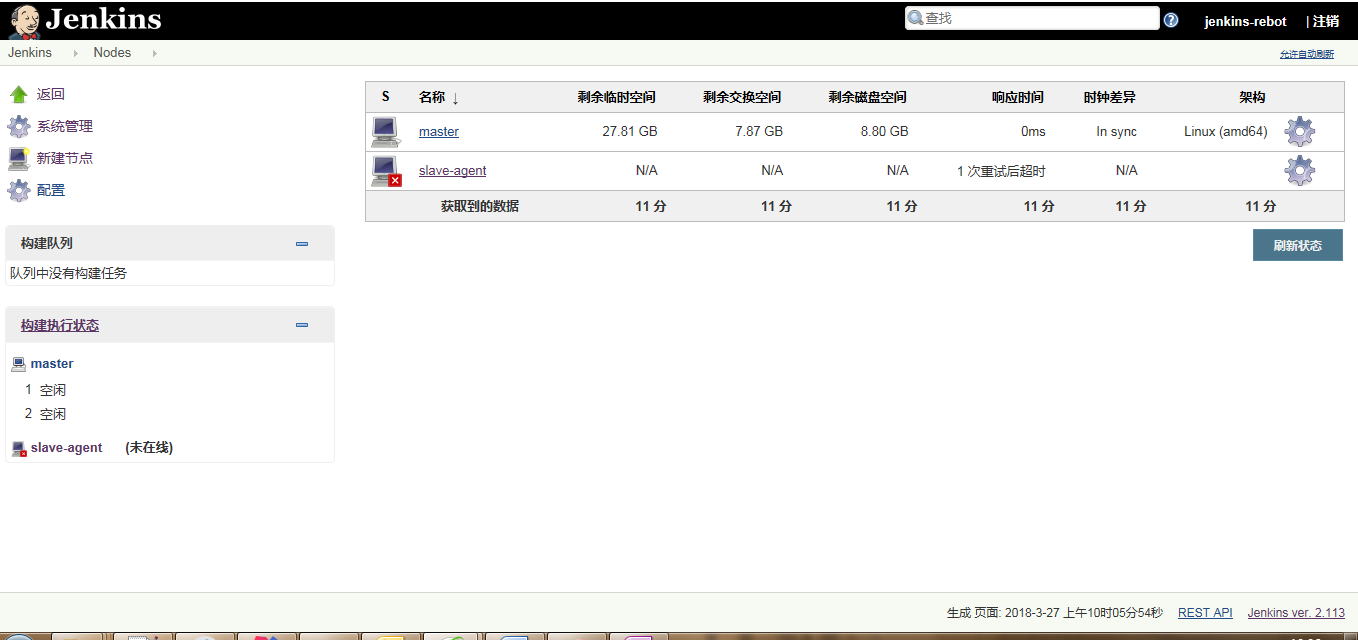
系统管理—〉管理节点—〉新建节点



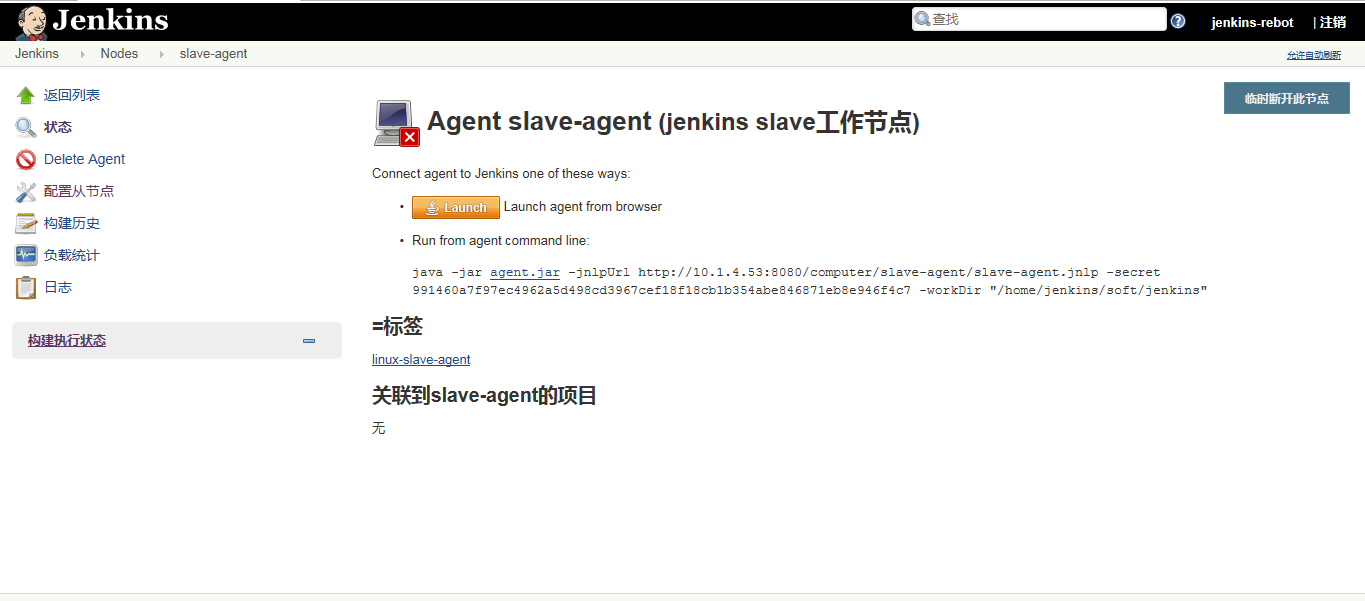
### 填写slave配置信息



配置完成后在nodes菜单下面有一个master和slave2个节点



点击进入到slave-agent节点



### 运行slave节点

Jenkins slave运行方式有2种，windows环境下建议使用方法一，linux环境下建议使用方法二

#### 方法一：Launch agent from browser(下载包)

点击launch按钮，会出现一个java frame，连接上jenkins后在frame左上角会出现一个file菜单

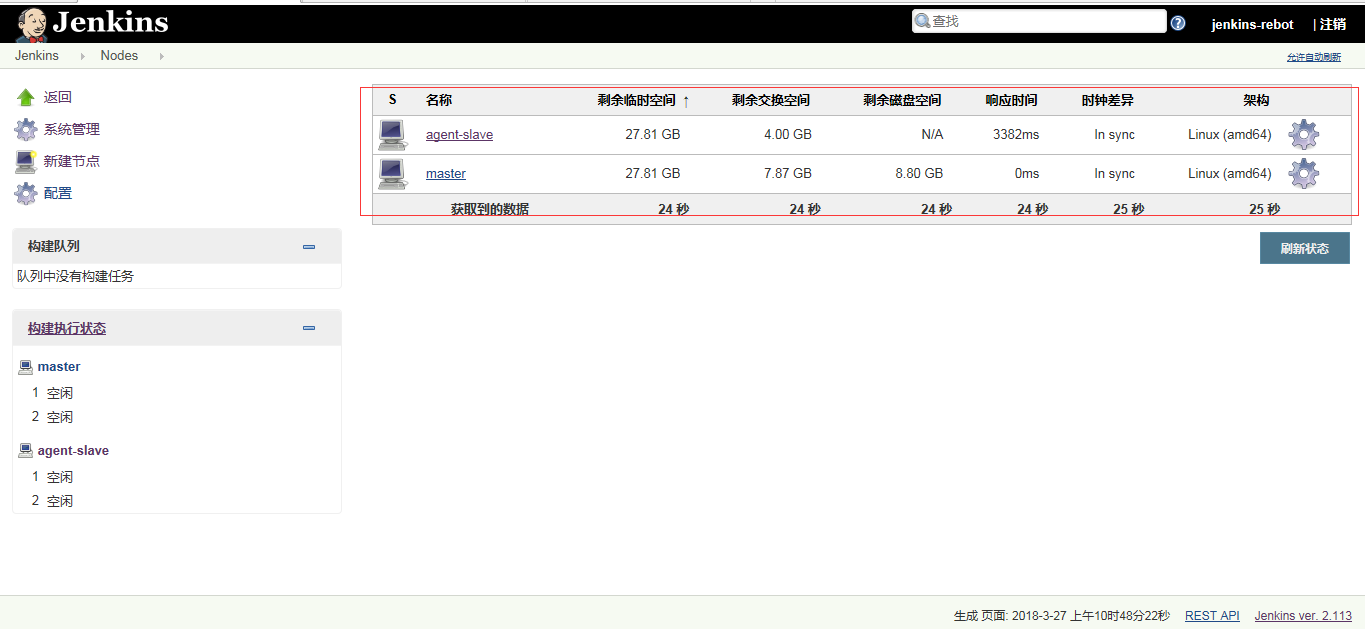


#### 方法二：cmd 方式 Run from agent command line

1. 点击页面的[agent.jar](http://10.1.4.53:8080/jnlpJars/agent.jar) 现在slave jar包
2. 在slave目标机器上创建一个工作目录 eg：/home/jenkins/soft/Jenkins
3. 将agent.jar 上传到slave机器jenkins工作目录中 /home/jenkins/soft/Jenkins
4. 新建slave启动脚本start\_jenkins\_slave.sh，脚本内容如下

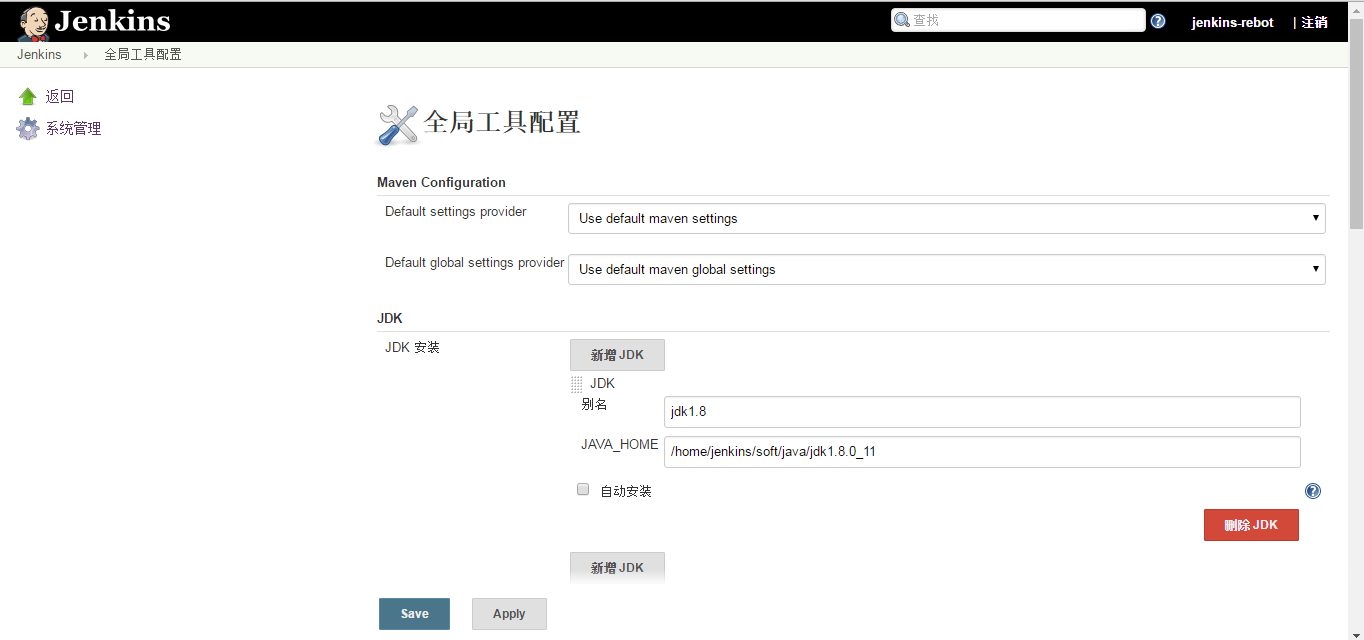
nohup java -jar ./agent.jar -jnlpUrl http://10.1.4.53:8080/computer/agent-slave/slave-agent.jnlp -secret f7206632afb8d7f73a34e8419b9777bc74eb546433c3e038e6df7200a057f105 -workDir "/home/jenkins/soft/jenkins" &

1. 启动完成之后在master的页面上可以看到slave运行状态



# 全局工具配置

配置jdk，ant，maven，git，svn,docker变量



# 全局环境变量配置

如果某些环境变量，每个项目在构建的时候都需要的，可以在工作节点上添加一个环境变量，通过系统管理\节点管理，选择需要配置工作节点进行配置。

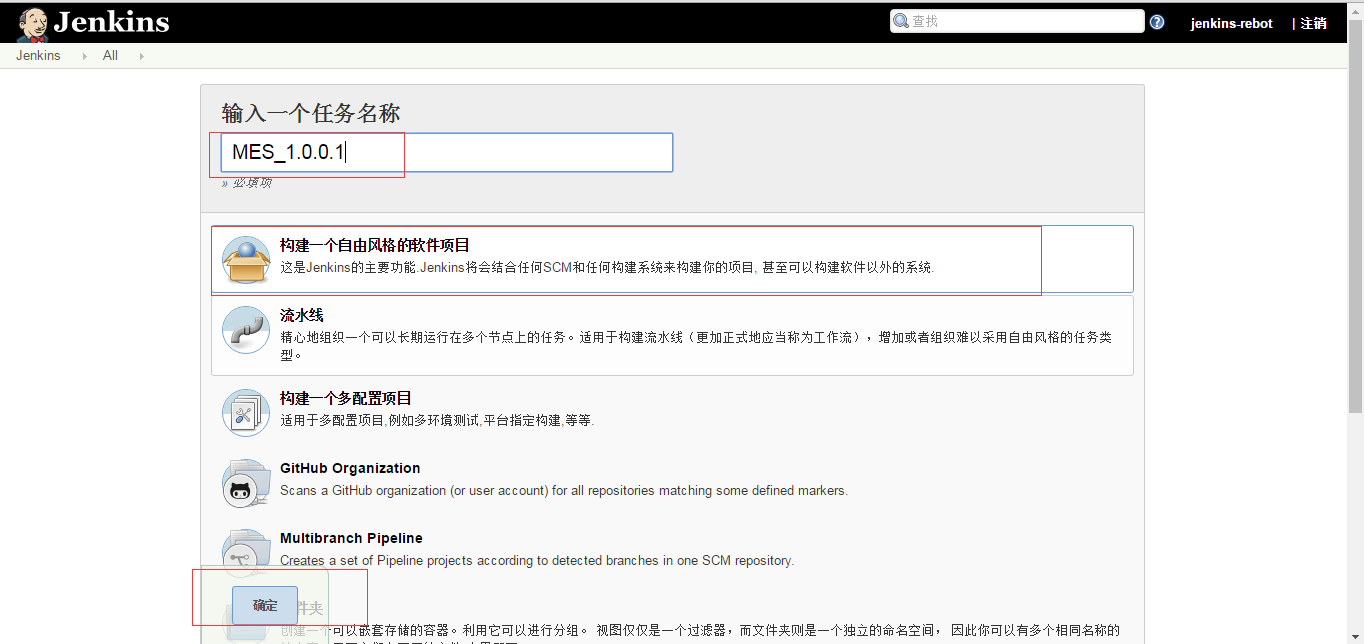


# 配置项目

## Svn +Ant 项目配置

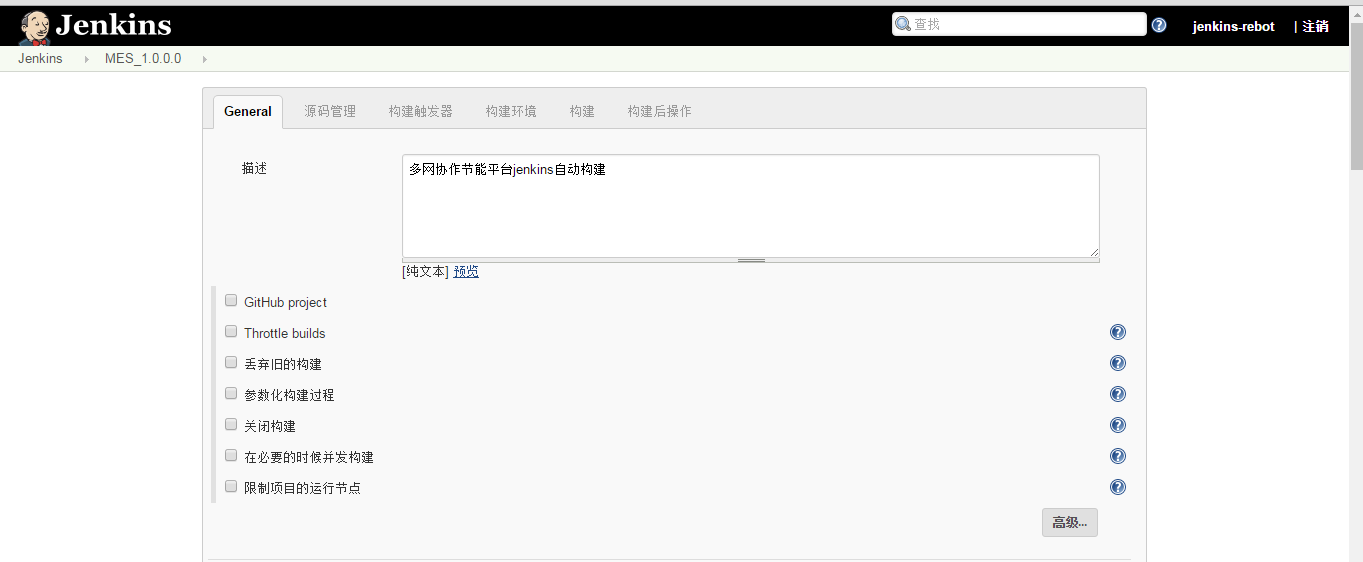
### 创建一个项目

创建任务，选择自由风格的项目即可。



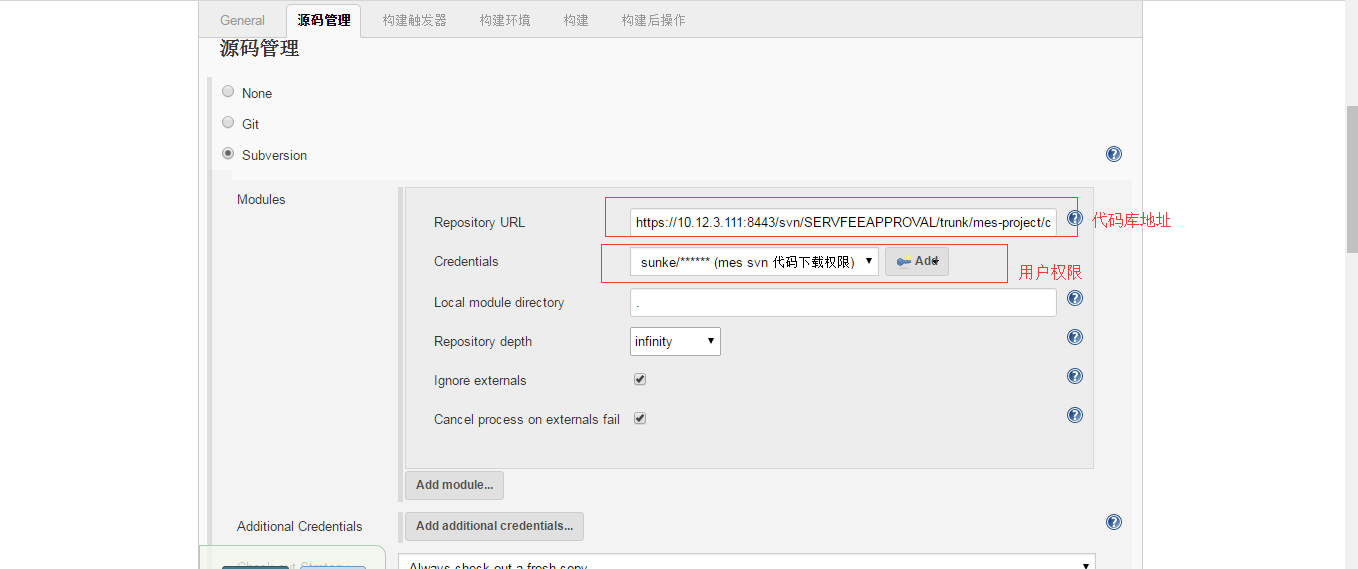
### 填写项目基本描述信息

填写项目描述。



### 源码库配置

填写svn的地址和可用的svn账号密码即可。



### 构建触发器配置

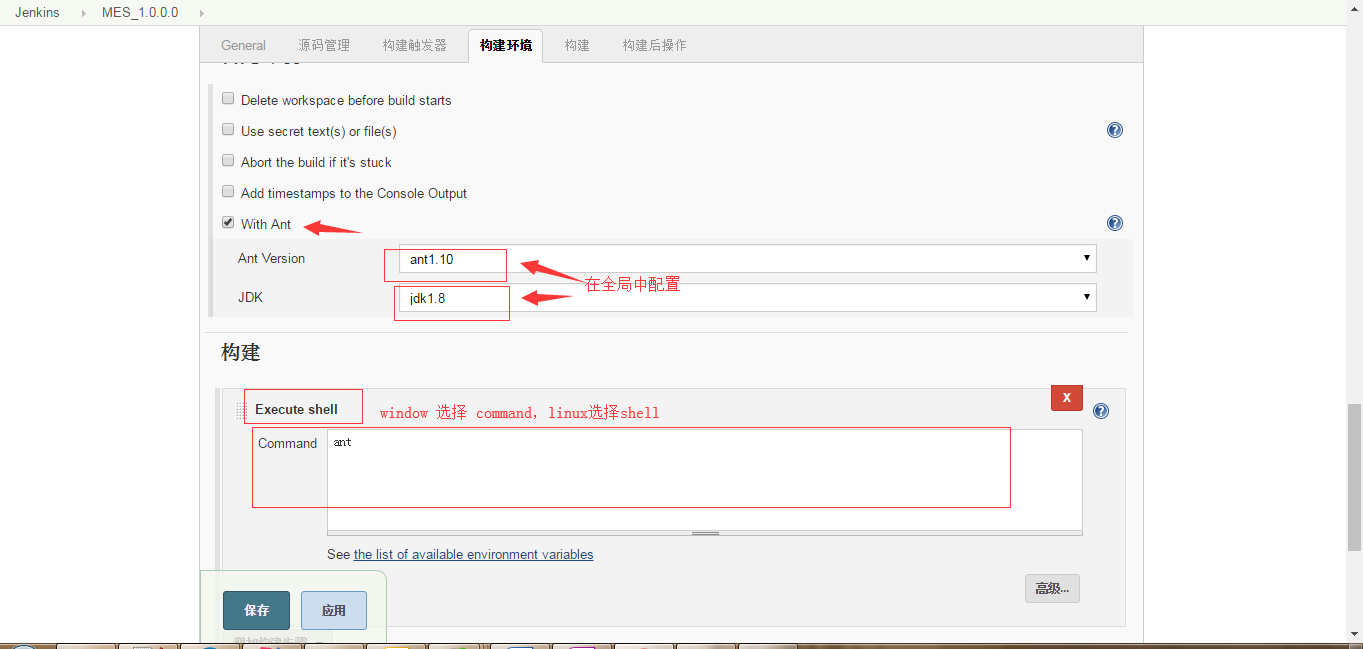
触发器选在定时触发，日程表共5位，相邻位之间使用空格分开，从右往左单位分别为年，月，日，时，分。写法类似cron表达式，H/5 \* \* \* \*表示没5分钟执行一次。



### 构建环境&&构建配置

根据项目的实际情况选择构建的环境，本案例使用的是ant+jdk1.8进行项目的构建。案例中可选的jdk和ant可以先在全局工具中配置好（请参考全局工具配置章节）。

构建部分，需要根据jenkins工作节点的环境情况来选择需要执行的脚本，如工作节点为windows则选择执行cmd，linux则选择执行shell，在command中可以写如何在工作节点环境中可执行的命令。Command当前的路径为$JENKINS\_HOME/workspace/$PROJECT\_NAME,可以根据实际的需要使用工作环境支持的命令进行目录的切换（eg：linux可使用cd命令进行目录的切换）。

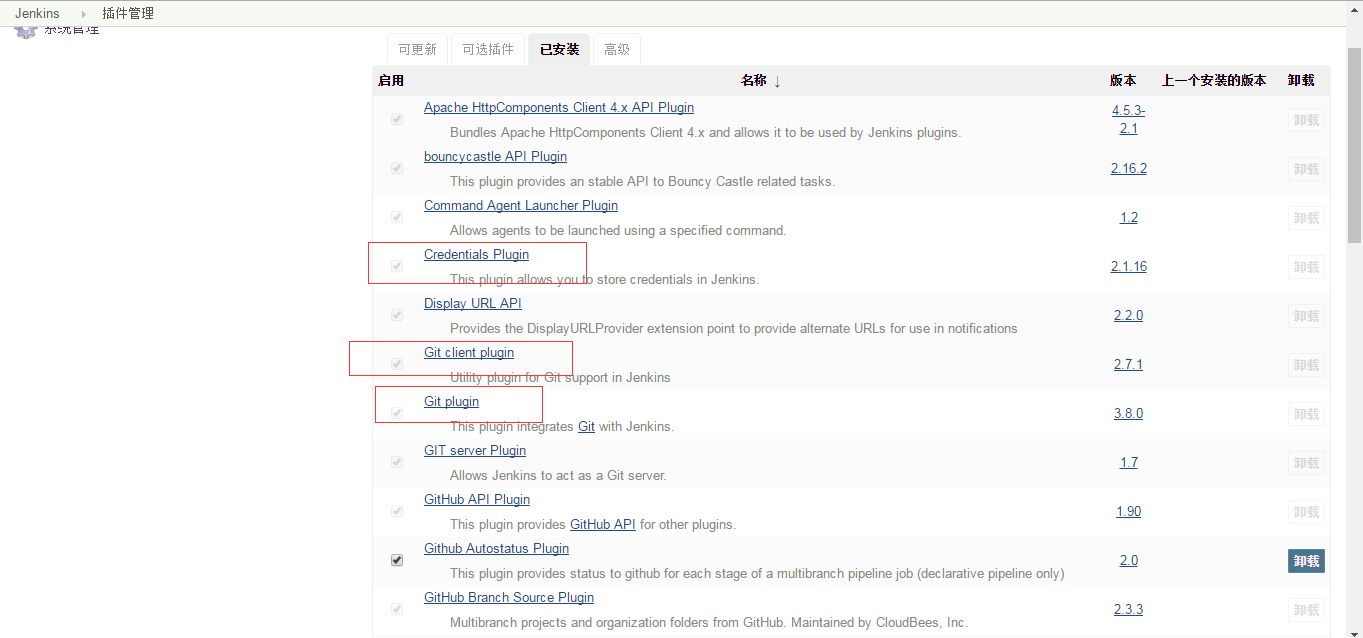


## GIT + Ant项目配置

Jenkins 集成 git 代码库和svn类是，只要在jenkins中先安装git client ，git plugin相关的插件，如果使用的是github则需要安装github相关的插件。

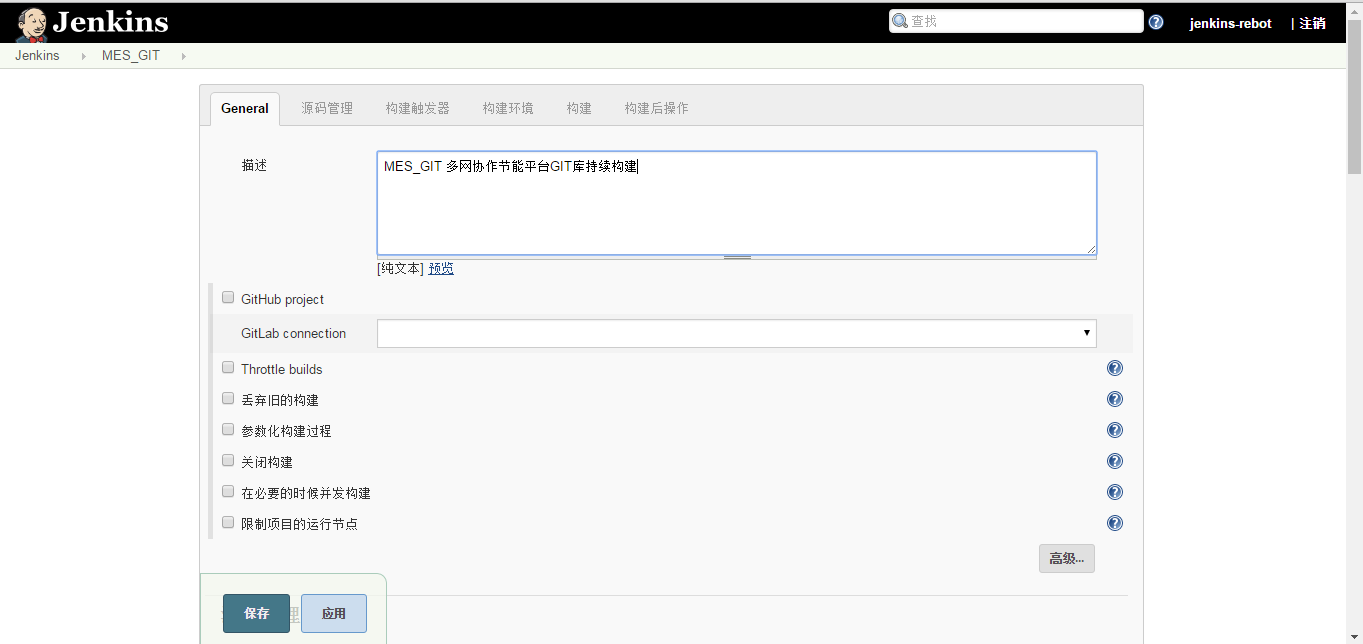
### 前提条件

需要先安装Git client,git plugin插件，建议把git相关的几个插件都安装一下。



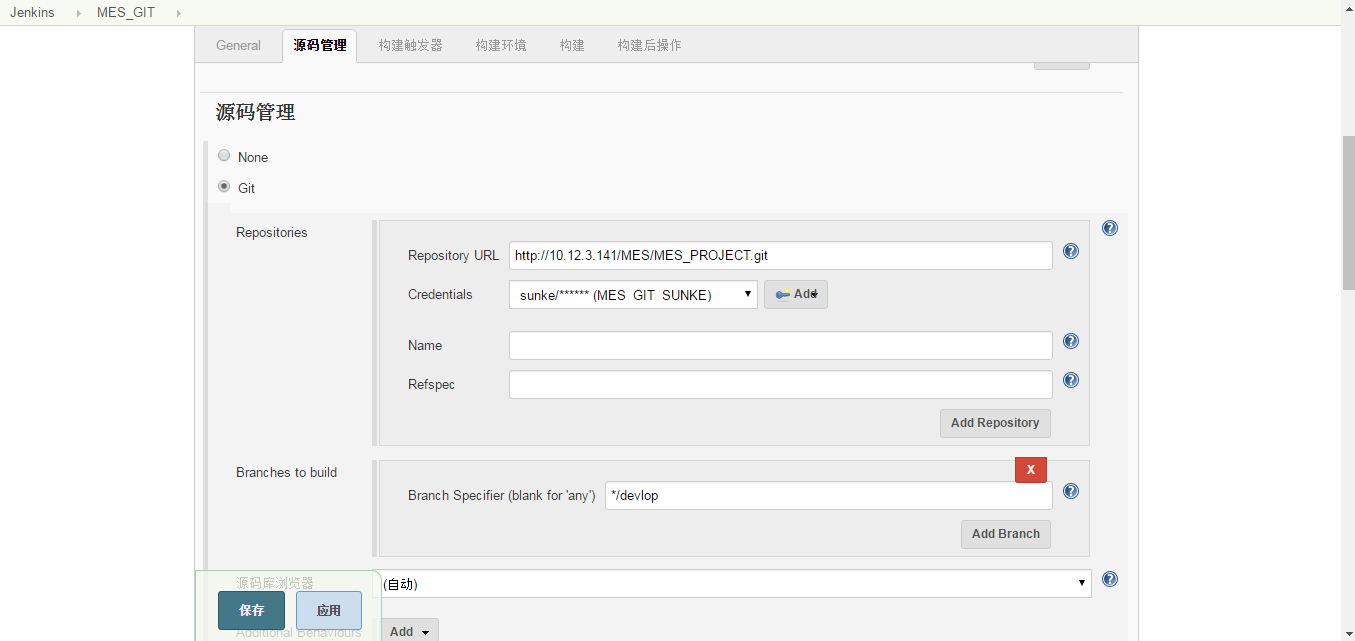
### 项目基本配置

创建一个简单类型的项目，填写描述。



### 代码库配置

填写git仓库地址（填入的地址使用git clone可用），填入鉴权信息，选择需要下的分支（样例图中选择的是devlop分支）。



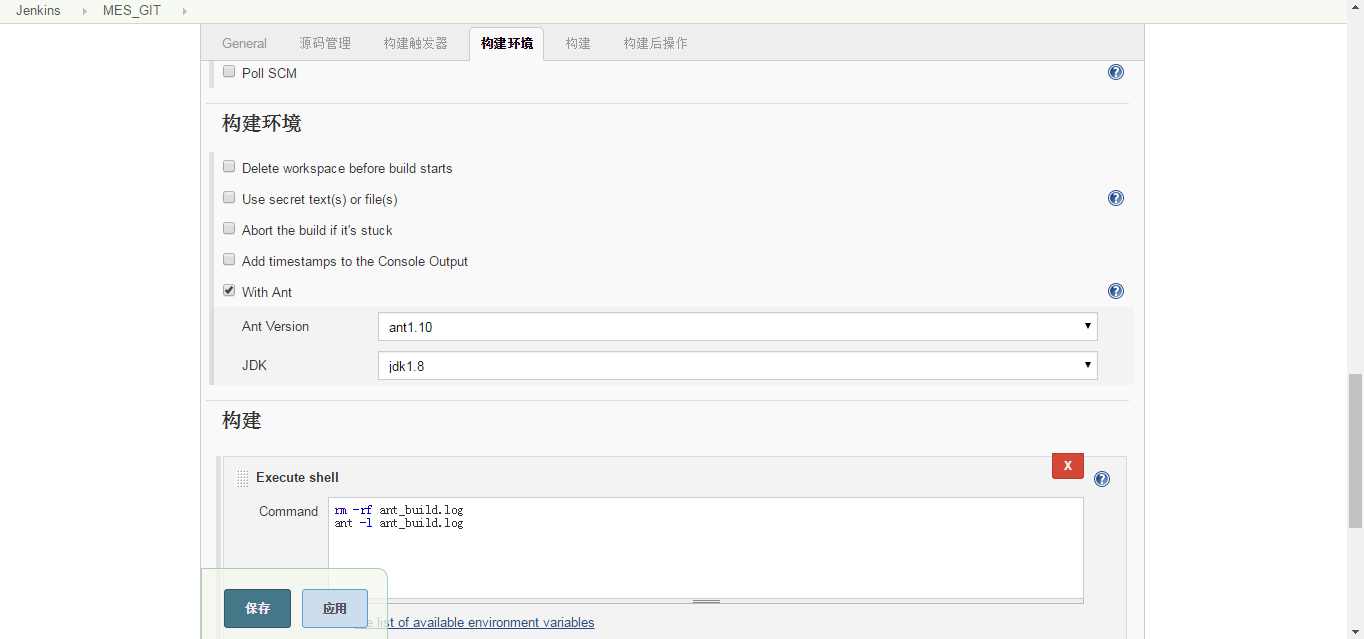
### 配置构建触发器

配置构建触发器，这里选择的是Job定时器(build periodically),配置每5分钟执行一次，日程表表达式类似于cron表达式写法，从做到右依次代表分，时，日，月，年。H表示范围。



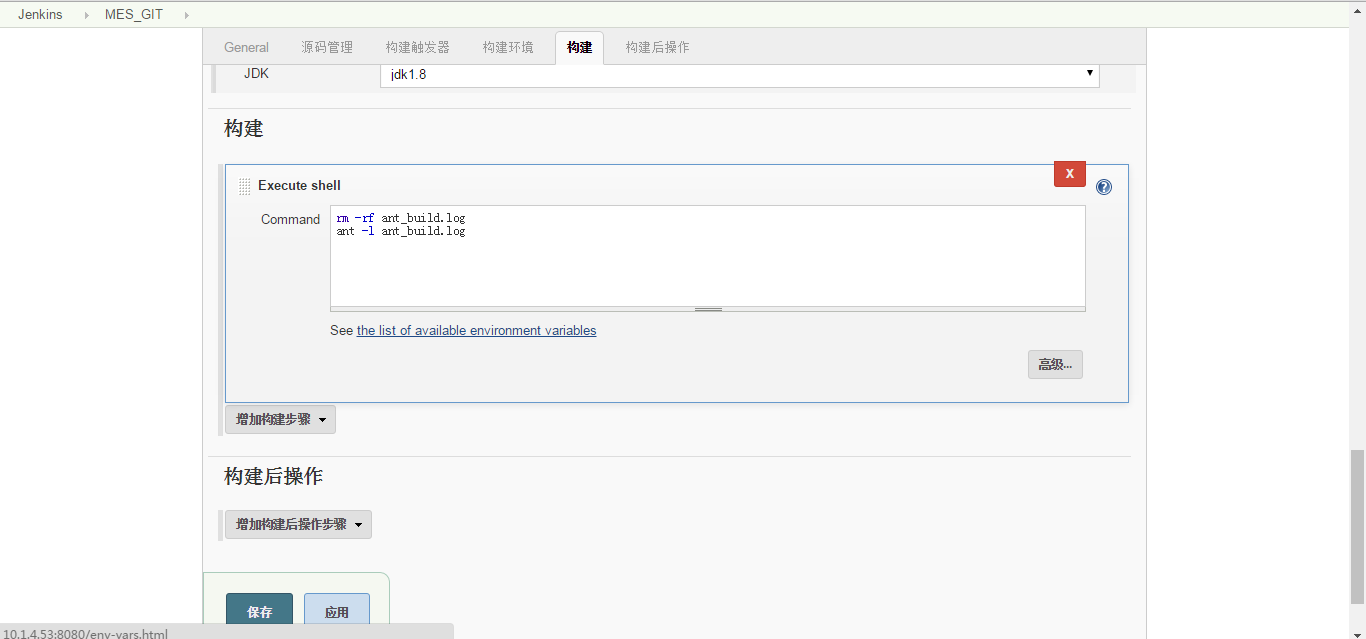
### 构建环境选择

由于项目使用的是ant进行构建，这里选择ant，并指定对应的jdk版本。



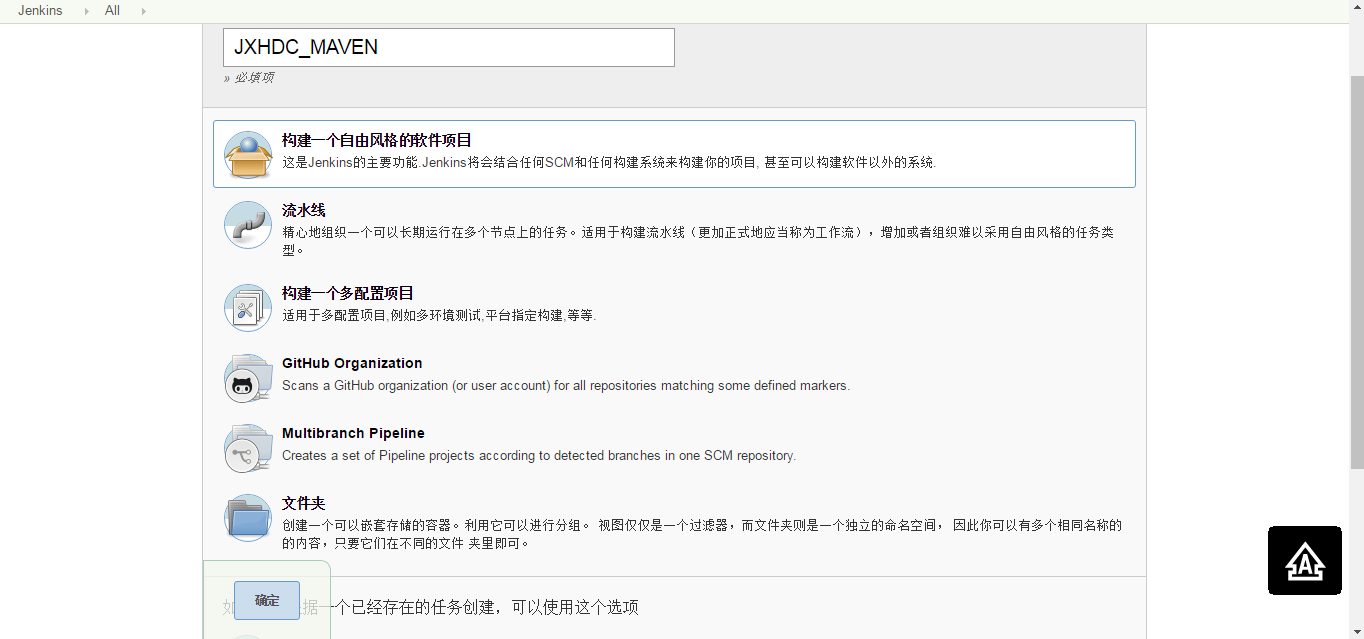
### 构建脚本

构建脚本需要根据jenkins工作节点的环境来编写。如工作节点为windows需要选择excute cmd，如果为linux需要选择excute shell。在command中可以编写环境中需要的所有指令，如linux 的cd ，rm ， export等等。Command初始路径为$JENKINS\_HOME/workspace/$PROJECT\_NAME中，如果需要执行的文件不再当前目录，可以使用cd命令进行切换。

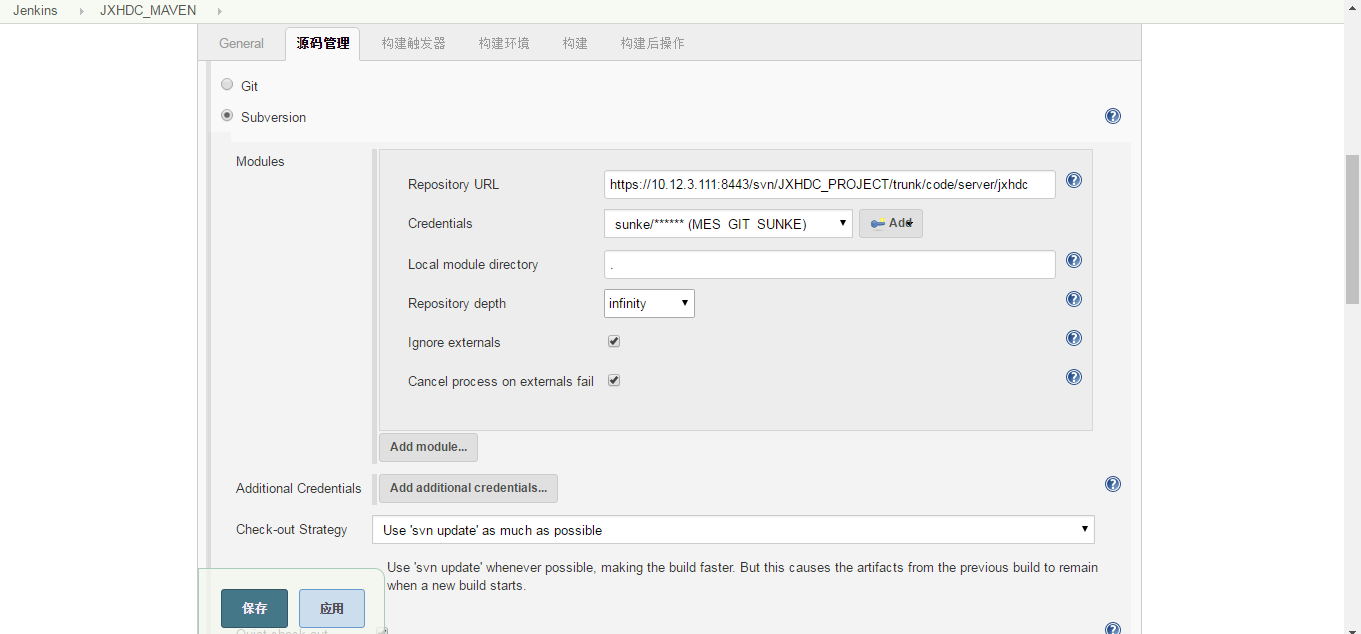


## Svn+ Maven　项目配置

### 创建项目



### 配置源码

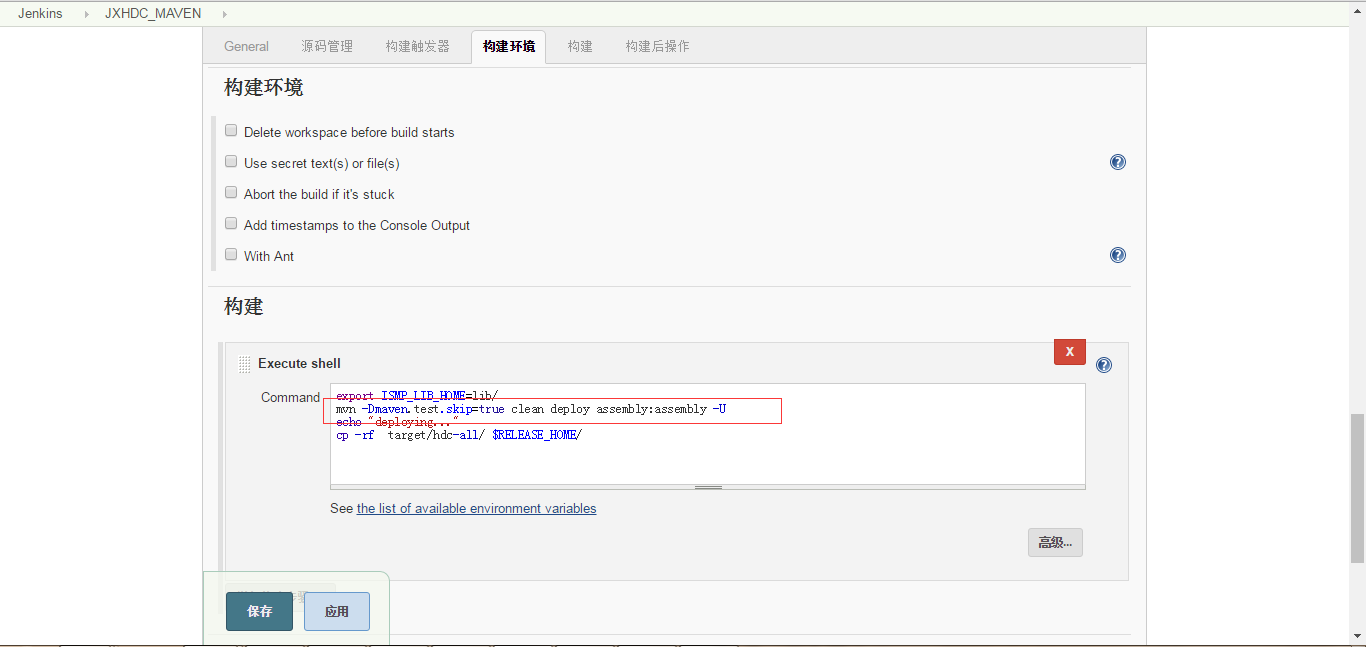


### 配置触发器



### 构建

通过shell直接调用mvn命令进行编译。



# 离线插件安装

jenkins 离线插件下载地址

<http://updates.jenkins-ci.org/download/plugins/script-security/>

linux 安装 Jenkins 步骤

1. 使用root权限下载

wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo

rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key

1. 安装jenkins

yum install jenkins

1. 安装成功后的文件目录

/usr/lib/jenkins/jenkins.war ## jenkins安装目录，WAR包会放在这里

/etc/sysconfig/jenkins  ## 配置文件

# Jenkins集成 sonar

<https://blog.csdn.net/kefengwang/article/details/54377055>

一 安装mysql

二 安装sonar并修改配置

## sudo vim /opt/sonarqube-6.2/conf/sonar.properties

sonar.web.javaOpts=-server -Xms256m -Xmx768m -XX:+HeapDumpOnOutOfMemoryError

sonar.jdbc.username=sonar

sonar.jdbc.password=Sonar6.2

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

## 下面设定访问网址为 http://centos:9000/sonar

sonar.web.host=0.0.0.0

sonar.web.port=9000

sonar.web.context=/sonar

启动服务

sudo /opt/sonarqube-6.2/bin/linux-x86-64/sonar.sh stop ## 停止服务

sudo /opt/sonarqube-6.2/bin/linux-x86-64/sonar.sh start ## 启动服务

sudo /opt/sonarqube-6.2/bin/linux-x86-64/sonar.sh restart ## 重启服务

sudo /opt/sonarqube-6.2/bin/linux-x86-64/sonar.sh console ## 控制台启动

sudo tail -f /opt/sonarqube-6.2/logs/sonar.log ## 查看日志

sonar配置

aspire-test: **e291dc80e4e7f6e94a8ae26c2af2c68eb9aad3d4**

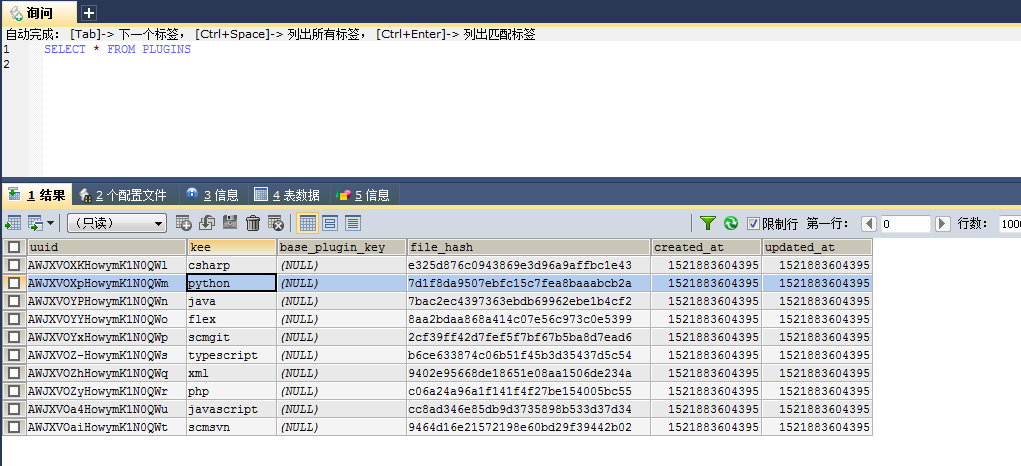
mvn sonar:sonar \

-Dsonar.host.url=http://127.0.0.1:9000 \

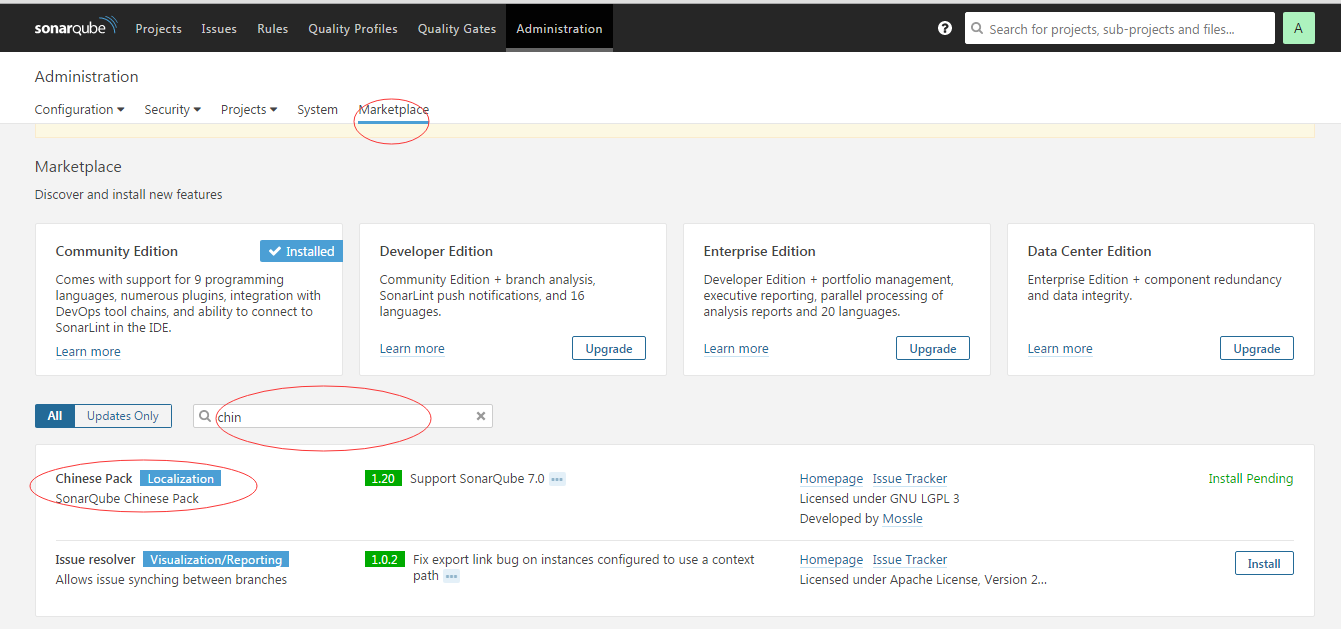
-Dsonar.login=e291dc80e4e7f6e94a8ae26c2af2c68eb9aad3d4

插件安装

SonarQu 安装完成后自带了一些插件，具体已经安装的插件可以查询sonar库的plugins表



1. Chinese pack汉化包



1. Findbug
2. Checkstyle
3. Pmd
4. sonarGraph

5.

# FAQ

## Job for jenkins.service failed

Starting jenkins (via systemctl): Job for jenkins.service failed because the control process exited with error code. See "systemctl status jenkins.service" and "journalctl -xe" for details.

出现如上错误有两种可能：

1. Jenkins java路径与安装的java路径不一致 通过sudo vi /etc/init.d/jenkins修改java路径

candidates="

/etc/alternatives/java

/usr/lib/jvm/java-1.8.0/bin/java

/usr/lib/jvm/jre-1.8.0/bin/java

/usr/lib/jvm/java-1.7.0/bin/java

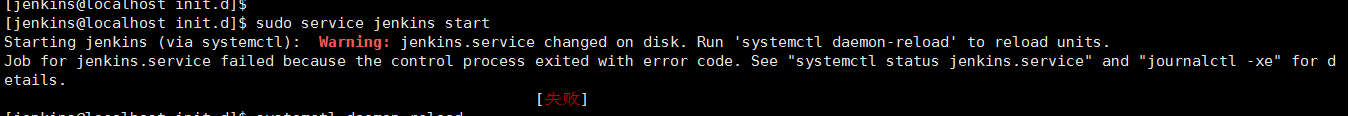
/usr/lib/jvm/jre-1.7.0/bin/java

/usr/bin/java

"

1. 没有java未安装或者版本不对 yum install java

## systemctl daemon-reload



解决方式： systemctl daemon-reload

