

《Jenkins 自动化发包》

编辑整理: Alinx

技术 Q 群 : 578549158

一、	jenkins 简介	2
二、	Jenkins 安装	3
1	安装 jdk.....	3
2	Maven 安装	3
3	jenkins 安装.....	4
三、	Jenkins 完善	9
1、	对接全局工具——JDK.....	9
2、	对接全局工具——Maven	10
3、	安装插件——maven.....	10
4、	安装插件——SSH	11
四、	Jenkins 部署 war 包程序.....	12
1、	服务器之间做好免密码登录.....	12
2、	服务器主机添加	12
3、	部署 jenkins.war 包程序.....	13
五、	Jenkins 部署 jar 包程序	18
六、	Jenkins 自动部署构建	23

- 一、 Jenkins 简介；
- 二、 Jenkins 安装；
- 三、 Jenkins 完善；
- 四、 Jenkins 部署 war 包程序；
- 五、 Jenkins 部署 jar 包程序；
- 六、 jenkins 自动部署构建；

系统环境：Centos7.2

Java 版本：JAVA1.8

Maven 版本：maven-3.5.4

Jenkins 版本：jenkins 最新版

一、 jenkins 简介

Jenkins 是一个开源软件项目，是基于 Java 开发的一种持续集成工具，用于监控持续重复的工作，旨在提供一个开放易用的软件平台，使软件的持续集成变成可能。

是一个独立的开源自动化服务器，可用于自动化各种任务，如构建，测试和部署软件。Jenkins 可以通过本机系统包 Docker 安装，甚至可以通过安装 Java Runtime Environment（java 运行环境）的任何机器独立运行。

下载 Jenkins 之后在下载目录中打开终端并运行

```
java -jar jenkins.war --httpPort=8080 &
```

浏览 <http://localhost:8080> 并按照说明完成安装。

二、 Jenkins 安装

1 安装 jdk

环境变量添加：

下载地址：

```
wget
http://download.oracle.com/otn-pub/java/jdk/8u171-b11/512cd62ec5174c3487ac17c61aaa89e8/jdk-8u171-linux-x64.tar.gz?AuthParam=1530757933_06ca50dfb743337fc0b4b072ec571c1f
```

解压包：

```
[root@iZe8t6k3ccrp36Z opt]# ls
apache-maven-3.5.4-bin.tar.gz  jdk-8u171-linux-x64.tar.gz  jenkins.war
[root@iZe8t6k3ccrp36Z opt]# tar -zxf jdk-8u171-linux-x64.tar.gz -C /usr/local/
[root@iZe8t6k3ccrp36Z opt]# cd !$
[root@iZe8t6k3ccrp36Z local]# ls
aegis      bin  games  jdk1.8.0_171  lib64  sbin  src
apache-maven-3.5.4  etc  include  lib          libexec  share
```

添加环境变量到 profile 文件

```
[root@iZe8t6k3ccrp36Z local]# vim /etc/profile
```

```
export JAVA_HOME=/usr/local/jdk1.8.0_171
export JAVA_BIN=/usr/local/jdk1.8.0_171/bin
export PATH=${JAVA_HOME}/bin:$PATH
export CLASSPATH=.:${JAVA_HOME}/lib/dt.jar:${JAVA_HOME}/lib/tools.jar
```

2 Maven 安装

包下载：

```
wget http://mirrors.hust.edu.cn/apache/maven/maven-3/3.5.4/binaries/apache-maven-3.5.4-bin.zip
```

```
[root@iZe8t6k3ccrp36Z opt]# tar -zxf apache-maven-3.5.4-bin.tar.gz -C /usr/local/
[root@iZe8t6k3ccrp36Z opt]# cd !$
[root@iZe8t6k3ccrp36Z local]# ls
aegis  apache-maven-3.5.4  bin  etc  games  include  jdk1.8.0_171  lib  lib64  libexec  sbin
share  src
```

添加环境变量

```
[root@iZe8t6k3ccrp36Z local]# vim /etc/profile
export MAVEN_HOME=/usr/local/apache-maven
export PATH=/usr/local/apache-maven/bin:$PATH
```

加载一下变量文件

验证环境：

```
[root@iZe8t6k3ccrp36Z local]# source /etc/profile
```

Java 环境：

```
[root@iZe8t6k3ccrp36Z local]# 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)
[root@iZe8t6k3ccrp36Z local]#
```

Maven 环境：

```
[root@iZe8t6k3ccrp36Z local]# mvn -v
Apache Maven 3.5.4 (1edded0938998edf8bf061f1ceb3cfdeccf443fe; 2018-06-18T02:33:14+08:00)
Maven home: /usr/local/apache-maven-3.5.4
Java version: 1.8.0_171, vendor: Oracle Corporation, runtime: /usr/local/jdk1.8.0_171/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "3.10.0-693.2.2.el7.x86_64", arch: "amd64", family: "unix"
[root@iZe8t6k3ccrp36Z local]#
```

然后开始运行 jenkins 对 直接运行 jenkins 。

3 jenkins 安装

下载 jenkins 安装 war 包

wget <http://mirrors.jenkins.io/war-stable/latest/jenkins.war>

直接运行即可

```
[root@iZe8t6k3ccrp36Z opt]# java -jar jenkins.war --httpPort=8888 &
```

Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:

a16368df7b7e4091acee67da4826217a

This may also be found at: /root/.jenkins/secrets/initialAdminPassword


```
Jul 10, 2018 12:22:17 PM hudson.model.UpdateSite updateData
INFO: Obtained the latest update center data file for UpdateSource default
Jul 10, 2018 12:22:20 PM hudson.model.DownloadService$Downloadable load
INFO: Obtained the updated data file for hudson.tasks.Maven.MavenInstaller
Jul 10, 2018 12:22:20 PM hudson.model.AsyncPeriodicWork$1 run
INFO: Finished Download metadata. 11,944 ms
Jul 10, 2018 12:22:22 PM hudson.model.UpdateSite updateData
INFO: Obtained the latest update center data file for UpdateSource default
Jul 10, 2018 12:22:23 PM jenkins.InitReactorRunner$1 onAttained
INFO: Completed initialization
Jul 10, 2018 12:22:23 PM hudson.WebAppMain$3 run
INFO: Jenkins is fully up and running
```

按照指示进入安装页面：

密码文件存放路径

[root@iZe8t6k3ccrp36Z opt]# **cat /root/.jenkins/secrets/initialAdminPassword**

```
[root@iZe8t6k3ccrp36Z opt]# cat /root/.jenkins/secrets/initialAdminPassword
a16368df7b7e4091acee67da4826217a
```

← → ① 不安全 | 47.106.113.221:8888/login?from=%2F

应用 淘宝购物 百度一下

入门


解锁jenkins

为了确保管理员安全地安装jenkins，密码已写入到日志中（[不知道在哪里？](#)）该文件在服务器上：

`/root/.jenkins/secrets/initialAdminPassword`

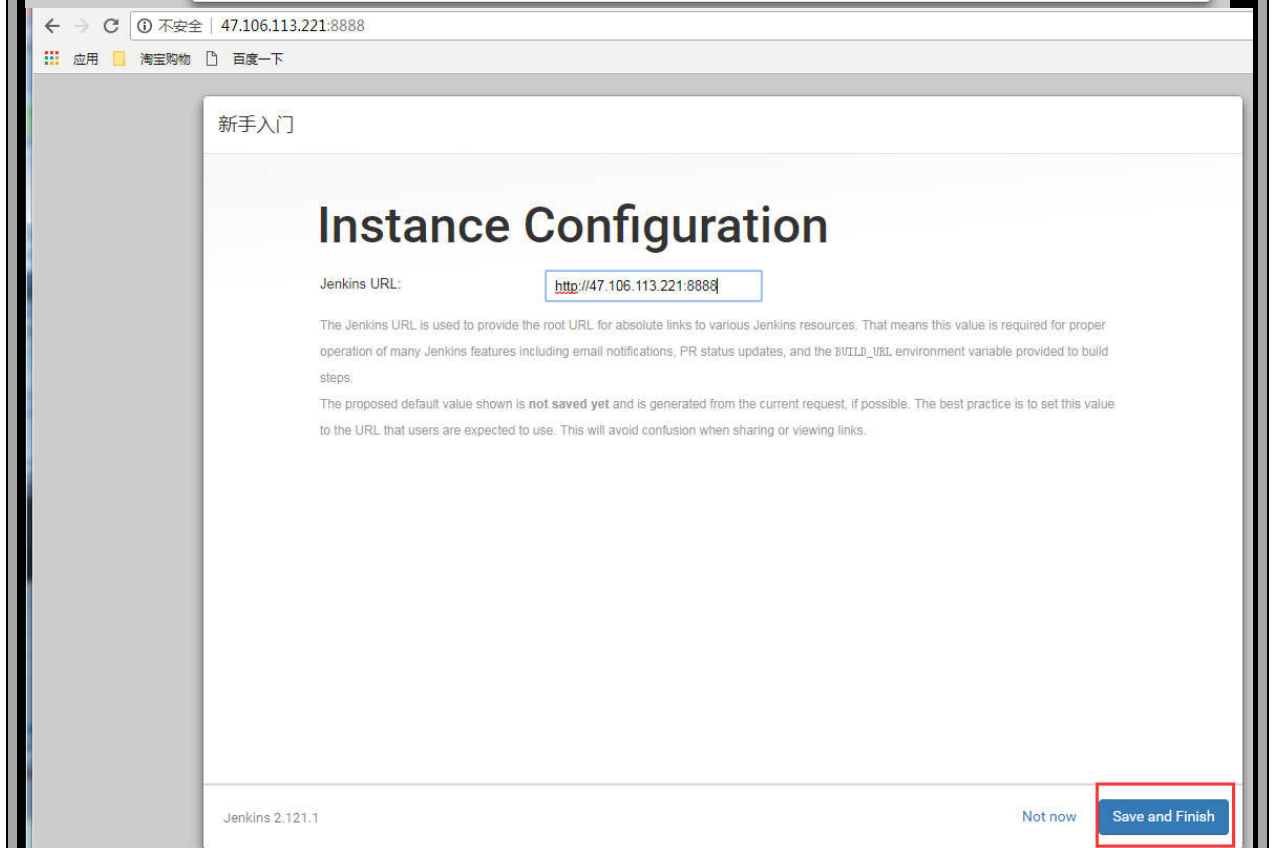
请从本地复制密码并粘贴到下面。

管理员密码



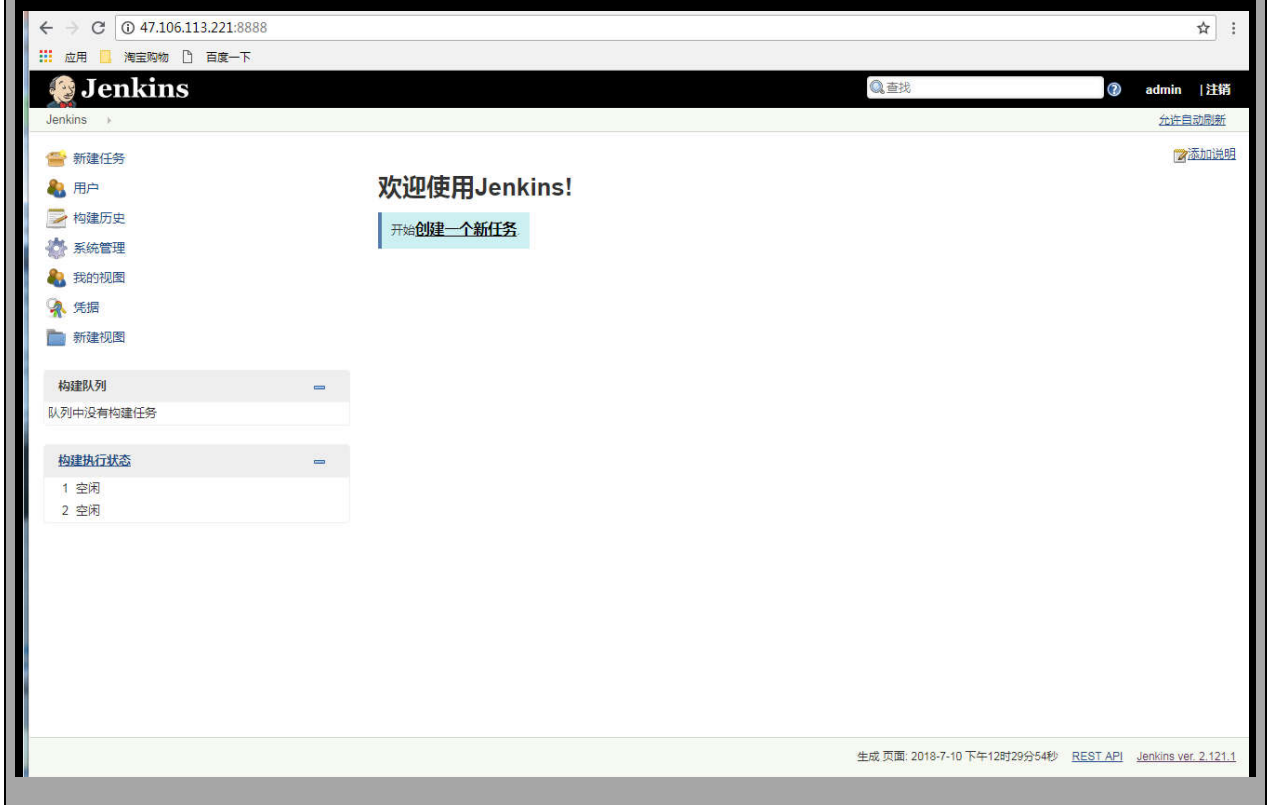
继续







安装到此就已经完成!!!



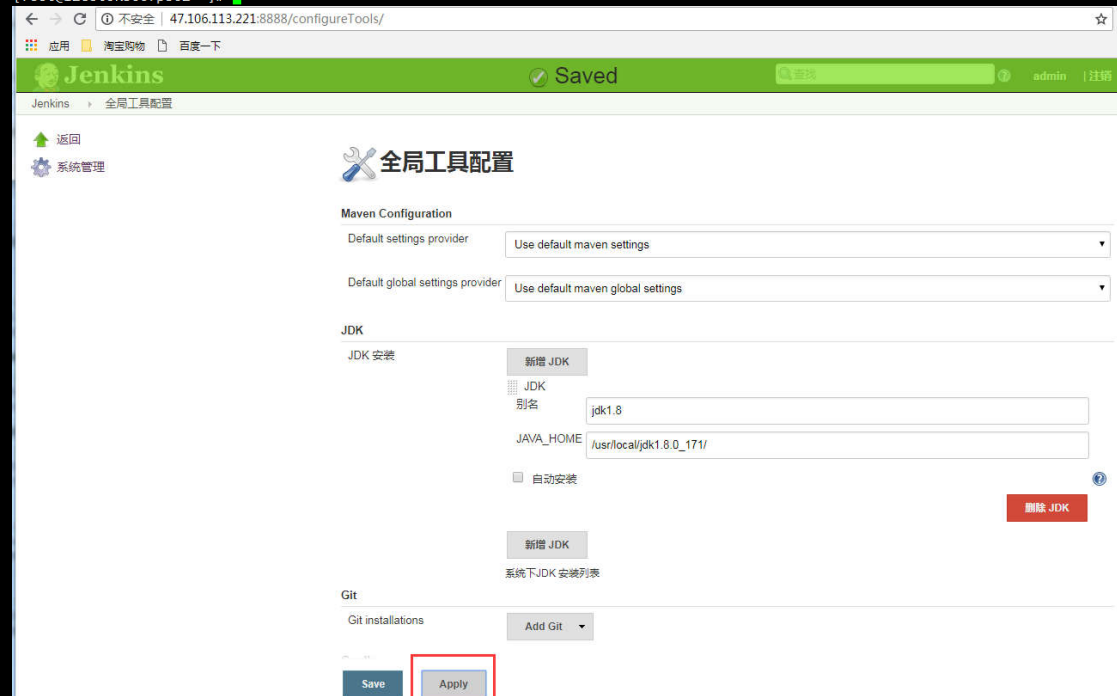
三、 Jenkins 完善

装好 jenkins 之后需要安装插件与添加服务器的主机信息



1、对接全局工具——JDK

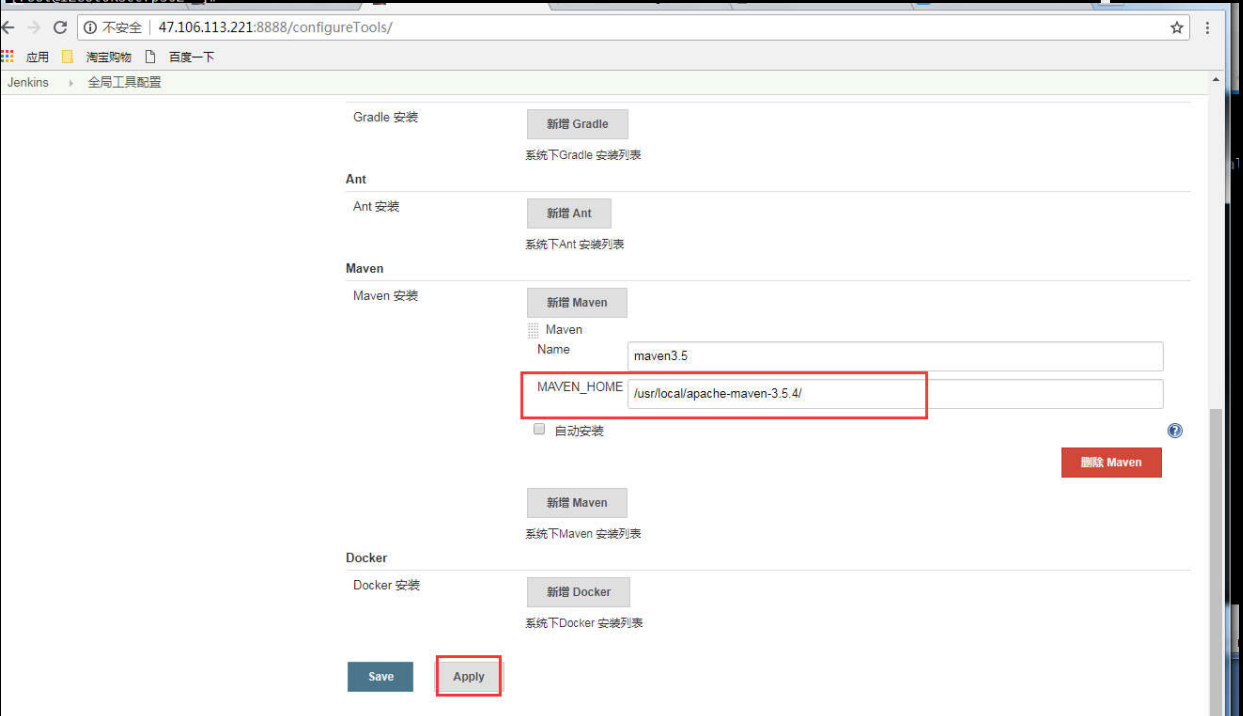
```
[root@iZe8t6k3ccrp36Z ~]# echo $PATH
/usr/local/apache-maven-3.5.4/bin:/usr/local/jdk1.8.0_171/bin:/usr/local/jdk1.8.0_171/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin:/bin:/bin:/bin:/bin
```



填入 Java 的安装路径，将自动安装的勾去掉，以免在构建的时间自动的去下载 Jdk 来构建，浪费大量的时间。

2、对接全局工具——Maven

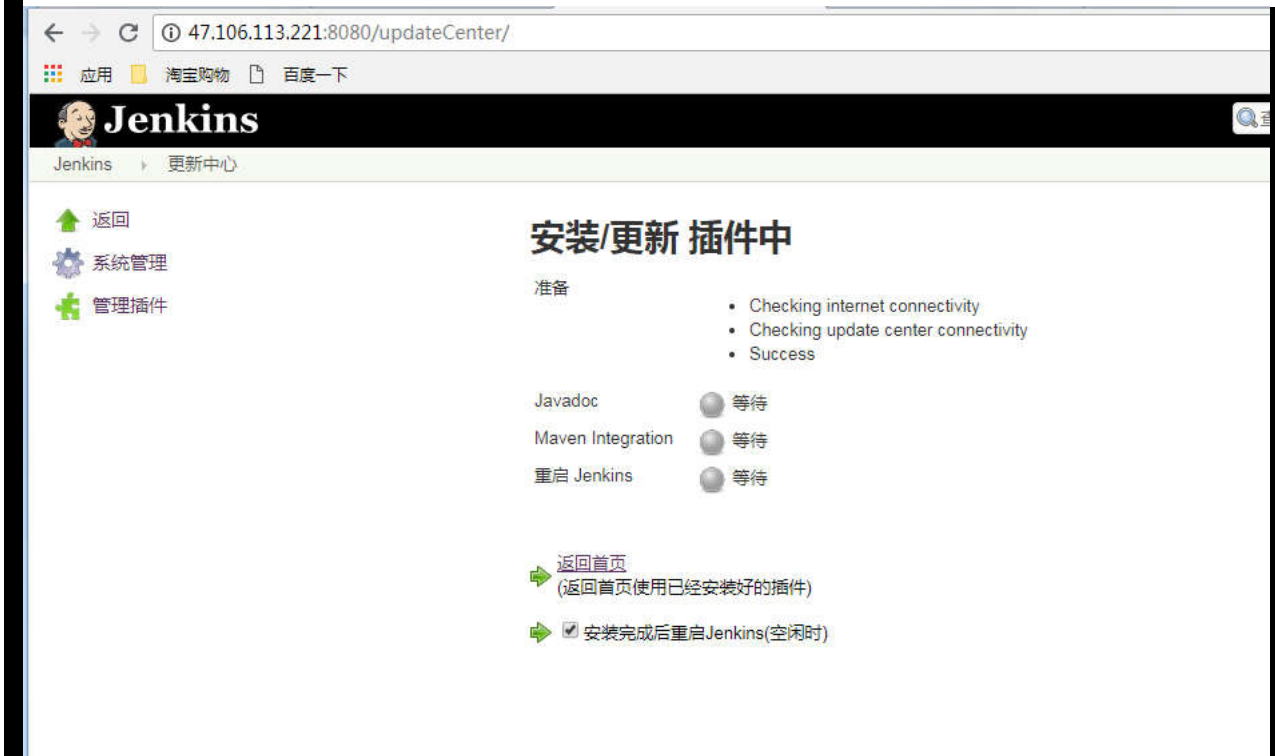
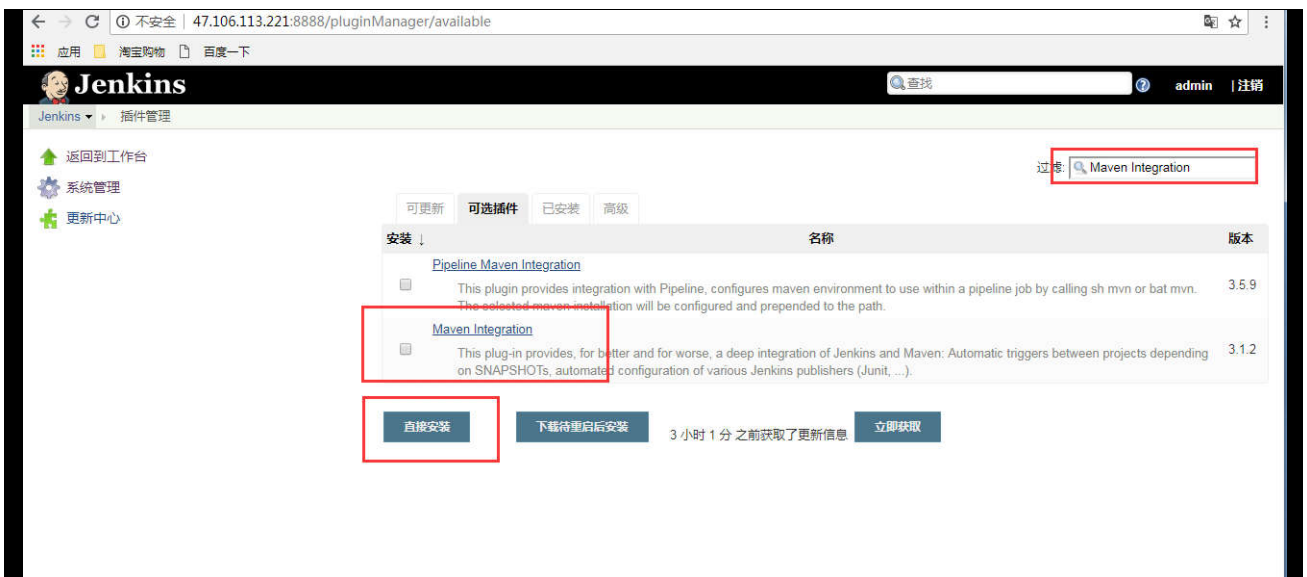
```
[root@iZe8t6k3ccrp36Z ~]# echo $PATH
/usr/local/apache-maven-3.5.4/bin:/usr/local/jdk1.8.0_171/bin:/usr/local/jdk1.8.0_171/bin:/usr/local/jdk1.8.0_171/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin:/bin:/bin:/bin:/bin
```



3、安装插件——maven

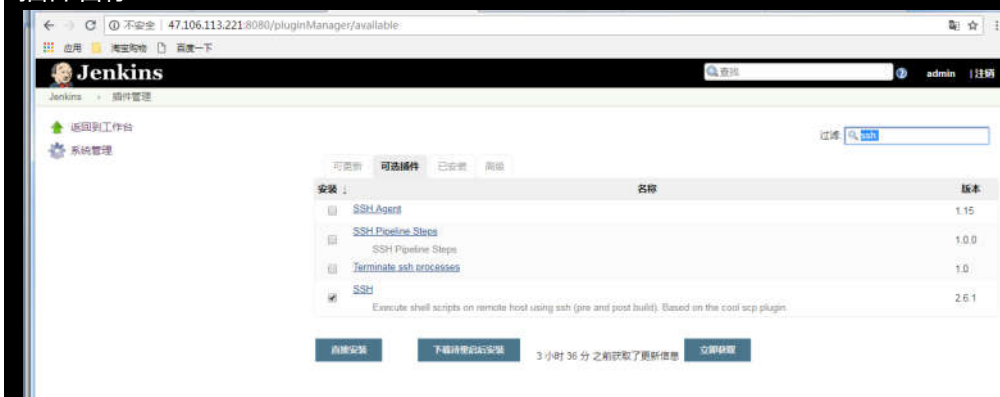
插件名称：Maven Integration





4、安装插件——SSH

插件名称：SSH



四、 Jenkins 部署 war 包程序

1、 服务器之间做好免密码登录

将你的公钥复制到你需要免密码登录的服务器上即可

查看是否存在密钥文件

```
[root@iZe8t6k3ccrp36Z ~]# ls /root/.ssh/  
authorized_keys id_rsa id_rsa.pub  
[root@iZe8t6k3ccrp36Z ~]#
```

如果不存在，创建它

```
[root@iZe8t6k3ccrp36Z ~]# ssh-keygen
```

一路回车到底就行

然后去服务器上生成授权文件

```
[root@iZe8t6k3ccrp36Z .ssh]# vim authorized_keys
```

>将本地电脑的 id_rsa.pub 内容拷贝到服务器上的 authorized_keys 里面，并保存

>现在服务器已经有了本地这台电脑的公钥

>修改服务器上面的 authorized_keys 权限

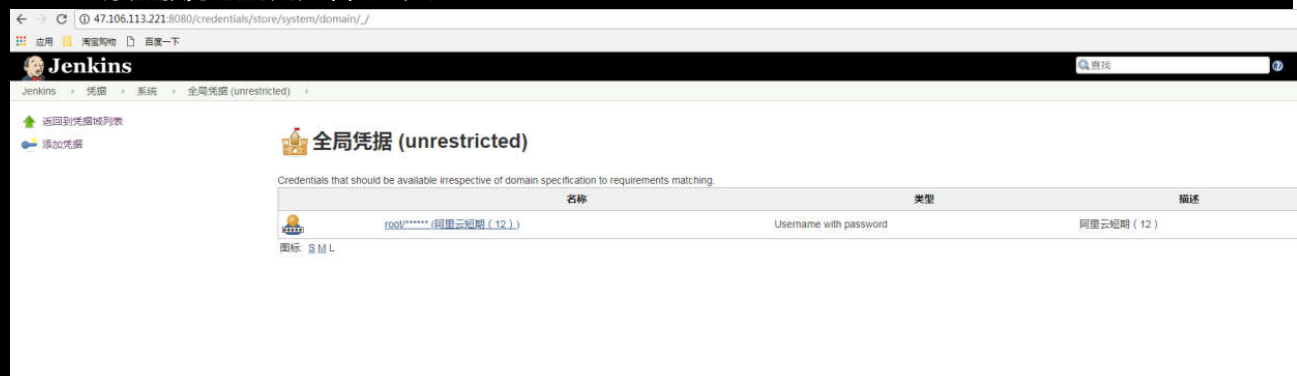
>chmod 600 authorized_keys

>重启服务器 ssh

```
[root@izwz9gyzplgtrem5yjm3hwz ~]# cd /root/.ssh/  
[root@izwz9gyzplgtrem5yjm3hwz .ssh]# ls  
authorized_keys id_rsa id_rsa.pub known_hosts  
[root@izwz9gyzplgtrem5yjm3hwz .ssh]# vim authorized_keys  
[root@izwz9gyzplgtrem5yjm3hwz .ssh]# cd ..  
[root@izwz9gyzplgtrem5yjm3hwz ~]# chmod 600 .ssh/authorized_keys  
[root@izwz9gyzplgtrem5yjm3hwz ~]# systemctl restart crond  
[root@izwz9gyzplgtrem5yjm3hwz ~]#
```

2、 服务器主机添加

添加服务器的用户名与密码：



添加一个免密码登录的云主机

SSH remote hosts

SSH sites

Hostname: 119.23.109.12

Port: 22

Credentials: root (阿里云短期 (12))

Pty: ☒

serverAliveInterval: 0

timeout: 0

Successful connection

Check connection

删除

3、部署 jenkins.war 包程序

← → ↻ ① 不安全 | 47.106.113.221:8080/view/all/newJob

应用 淘宝购物 百度一下


Jenkins


Jenkins > 所有 >


输入一个任务名称


jenkins


» 任务名 'jenkins' 已存在

 **构建一个自由风格的软件项目**
这是Jenkins的主要功能.Jenkins将会结合任何SCM和任何构建系统来构建你的项目, 甚至可以构建软件以外的系统.


 **构建一个maven项目**
构建一个maven项目.Jenkins利用你的POM文件,这样可以大大减轻构建配置.

 **流水线**
精心地组织一个可以长期运行在多个节点上的任务。适用于构建流水线（更加正式地应当称为工作流），增加或者组织难以采用自由风格的构建。


 **构建一个多配置项目**
适用于多配置项目,例如多环境测试,平台指定构建,等等.

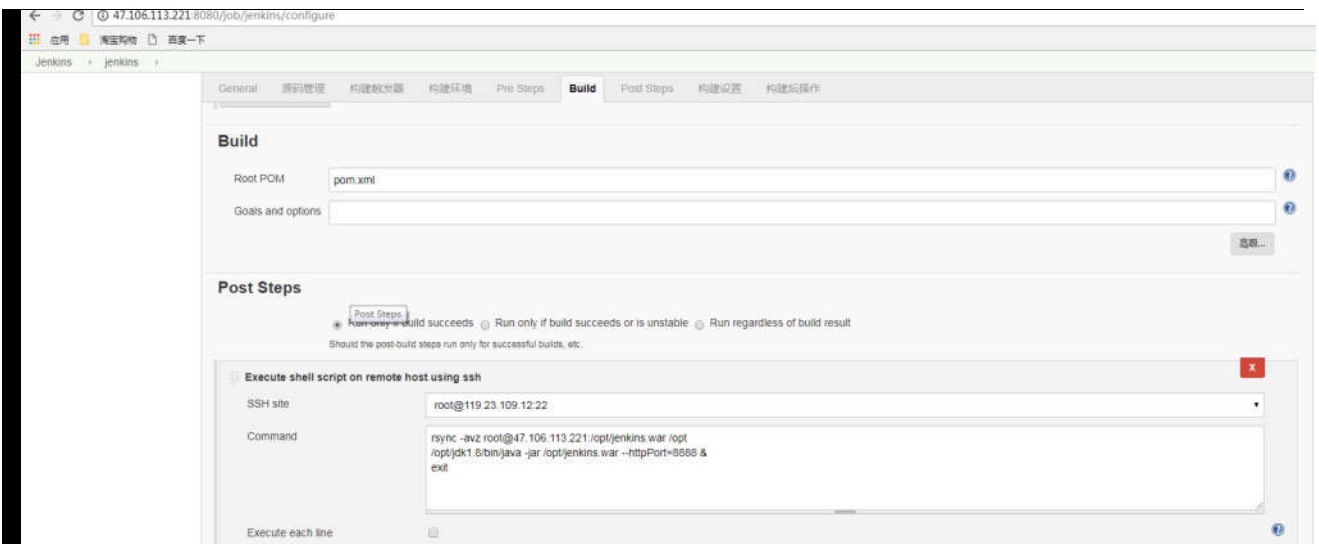
 **文件夹**
创建一个可以嵌套存储的容器。利用它可以进行分组。视图仅仅是一个过滤器，而文件夹则是一个独立的命名空间，因此你可以有多个相同名称的文件夹。

 **GitHub Organization**
Scans a GitHub organization (or user account) for all repositories matching some defined markers.

 **多分支流水线**
根据一个SCM仓库中检测到的分支创建一系列流水线。

如果你想根据一个已经存在的任务创建，可以使用这个选项

 复制 输入自动完成



```
rsync -avz root@47.106.113.221:/opt/jenkins.war /opt/
/opt/jdk1.8/bin/java -jar /opt/jenkins.war --httpPort=8888 &
exit
```

安装 rsync 同步工具

```
yum list |grep telnet
```

```
yum install telnet-server.x86_64 -y
```

```
yum install telnet.x86_64 -y
```

```
yum list |grep xinetd
```

```
yum install xinetd.x86_64 -y
```

```
systemctl enable xinetd.service
```

```
systemctl enable telnet.socket
```

```
systemctl start telnet.socket
```

```
systemctl start xinetd
```

```
systemctl start rsyncd.service
```

```
systemctl start rsyncd.service
```

```
rsync -avz root@ 47.106.113.221:/opt/jenkins.war /opt/
```

注意：这里如果不做免密码登录，是需要输入密码的，那就无法这样在脚本里面执行了

怎么办？去做免密码就好啦啊

```
cat /root/.ssh/id_rsa.pub
[ root@izwz9gyzplgtrem5yjm3hwz ~]# cat /root/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDAQ/tUq+R+I57L4q5Zy0TaWi+6DQ8NlYnvGHenwDp/n98/yK+SKNmX22TEQf+a0zg3m2azQdGJK+jlHJj
TTXILiCd0VY8toiduZZ0h/+0eDFLVFdc3v15m5pnCJgr24qyGL9Fjp8Wo9sqZagIMJpioCa8KYRBkjcqiSxYlg2KSKX3nXMMJEjaSFgwiJm9nr+QuM7Cz
1956RA7TWSG4AE24W8eSDKr6J6Qy9IniaL/la0dky0xUGjdd5a/6i58xfdjl8IwnyZu0pQneCqQNaN0L5ujqd'j82Ys6v7ZZMht
Y4G72HK6HQ57TrBDvvQD8oZ+Z root@izwz9gyzplgtrem5yjm3hwz
```

复制到需要同步文件的服务器授权文件里面即可

```
[root@iZe8t6k3ccrp36Z .ssh]# pwd
/root/.ssh
[root@iZe8t6k3ccrp36Z .ssh]# vim authorized_keys
```

开始构建：

A screenshot of the Jenkins web interface showing the console output of a build. The left sidebar contains navigation links: '返回到工程', '状态集', '变更记录', '控制台输出' (selected), '文本方式查看', '编辑编译信息', and '删除本次生成'. The main area is titled '控制台输出' and displays the following text:

```
Started by user admin
Building in workspace /root/.jenkins/workspace/Jenkins构建
[SSH] script:

/bin/sh /data/rsync.sh
exit

[SSH] executing...
receiving incremental file list

sent 19 bytes received 45 bytes 128.00 bytes/sec
total size is 74736772 speedup is 1167762.06
grep: unrecognized option '--httpPrt=8888'
Usage: grep [OPTION]... PATTERN [FILE]...
Try 'grep --help' for more information.
/data/rsync.sh: line 6: java: command not found
Jenkins is Down

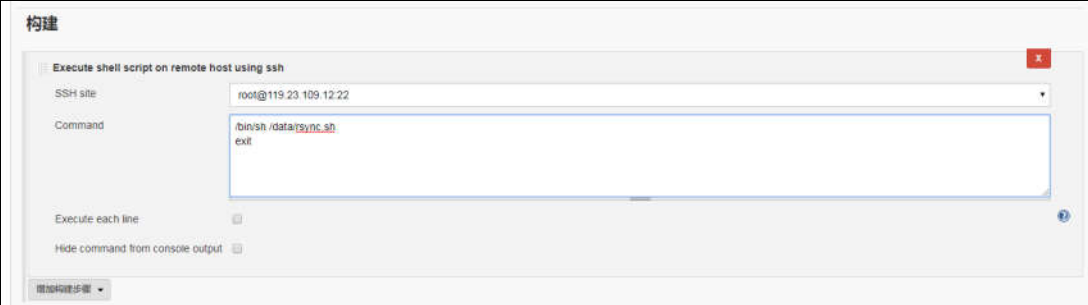
[SSH] completed
[SSH] exit-status: 0

Finished: SUCCESS
```

问题 1：

直接写命令在构建中看起来不是很美观，建议写脚本在服务器中，执行脚本这样会好很多。

脚本的执行方式如下：

A screenshot of the Jenkins 'Build' configuration page. The 'Execute shell script on remote host using ssh' section is expanded. The 'SSH site' dropdown is set to 'root@119.23.109.12.22'. The 'Command' text area contains the following script:

```
/bin/sh /data/rsync.sh
exit
```

Below the command area, there are checkboxes for 'Execute each line' (unchecked) and 'Hide command from console output' (checked). At the bottom left, there is a '增加构建步骤' (Add build step) button.





```
#!/bin/bash

rsync -avz root@47.106.113.221:/opt/jenkins.war /data/

java -jar /data/jenkins.war > /var/log/jenkins.log 2>&1 &

ps aux | grep -v grep | grep "/data/jenkins.war"
if [ $? -eq 0 ];then
echo "Jenkins is Running"
else
echo "Jenkins is Down"
fi
```

Jenkins Jenkins构建

-  返回面板
-  状态
-  修改记录
-  工作空间
-  立即构建
-  删除工程
-  配置
-  重命名

工程 Jenki

 工作区

 最新修改记

相关链接

- [最近一次构建\(#](#)
- [最近稳定构建\(#](#)
- [最近成功的构建](#)
- [最近完成的构建](#)

 **Build History** 构建历史

#2 2018-7-11 下午12:04

#1

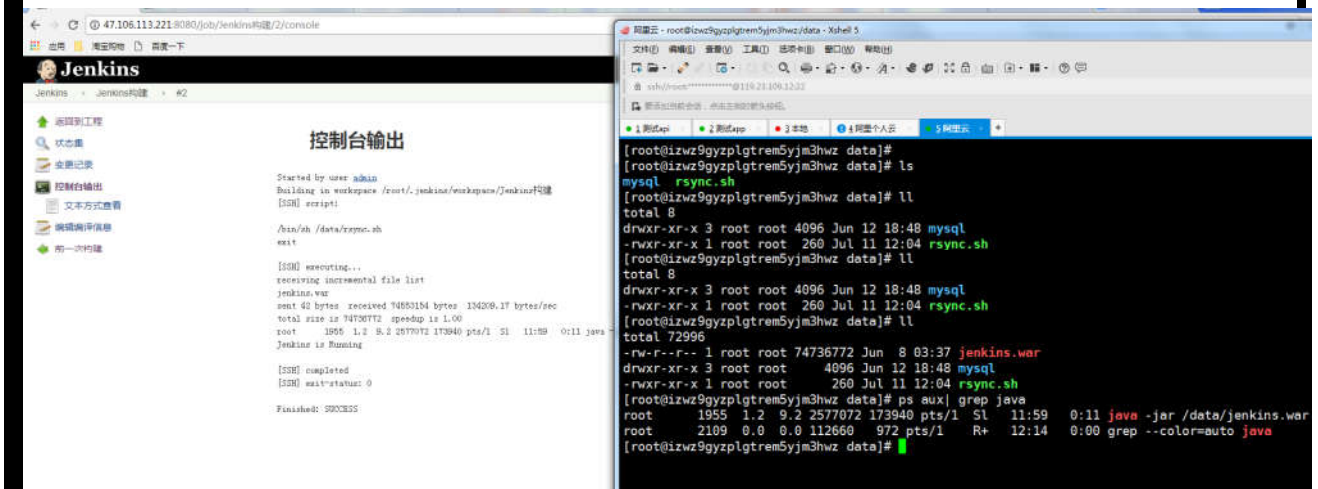
 变更记录

 控制台输出

 编辑编译信息

RSS 失败

没有对接代码仓库，但原理是一样



成功的构建。

Tomcat 构建也可以类似于这样，把打好包的 war 包通过同步或者 scp 传到服务器 tomcat 中，重启 tomcat 即可完成发布。

Ps: 因为没有 war 包项目的源码，所以使用 jenkins.war 来代替，哈哈

五、 Jenkins 部署 jar 包程序

申请或者去码云平台获取 git 地址（仓库），存放项目代码，然后使用 https 的方式对接认证；

1、新建构建任务（Maven 构建）

使用 maven 工具对项目进行打包，得到的 jar 包会放在

/root/.jenkins/workspace 包构建路径

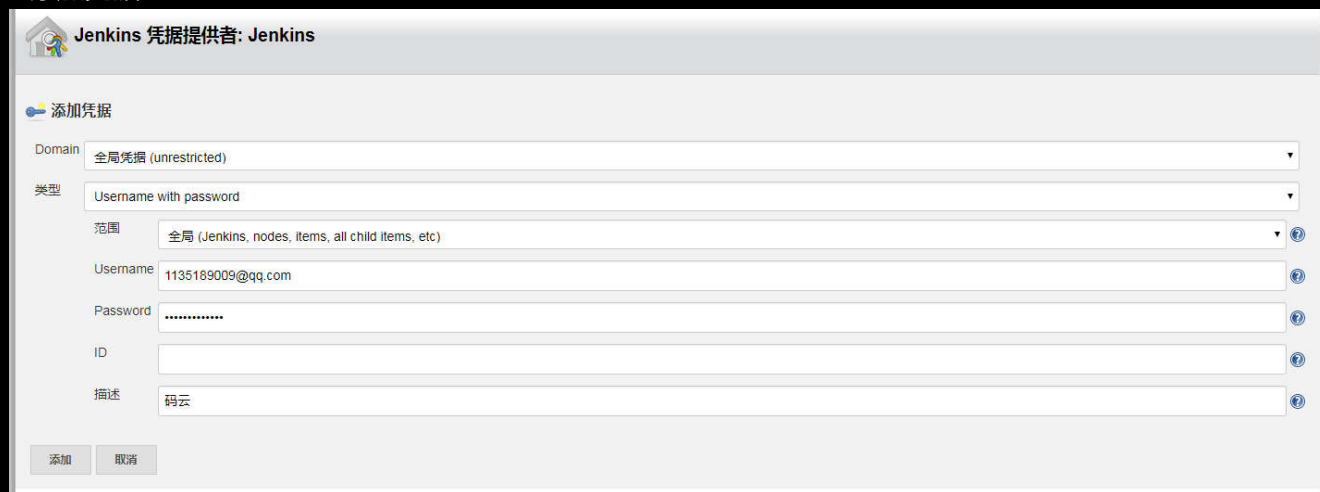
/root/.jenkins/jobs 配置与历史路径

使用 gitee.com 来对接项目：

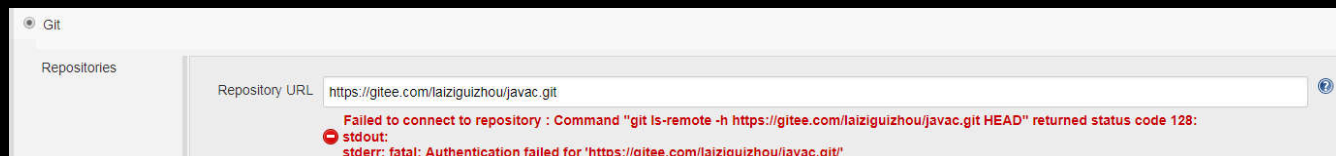




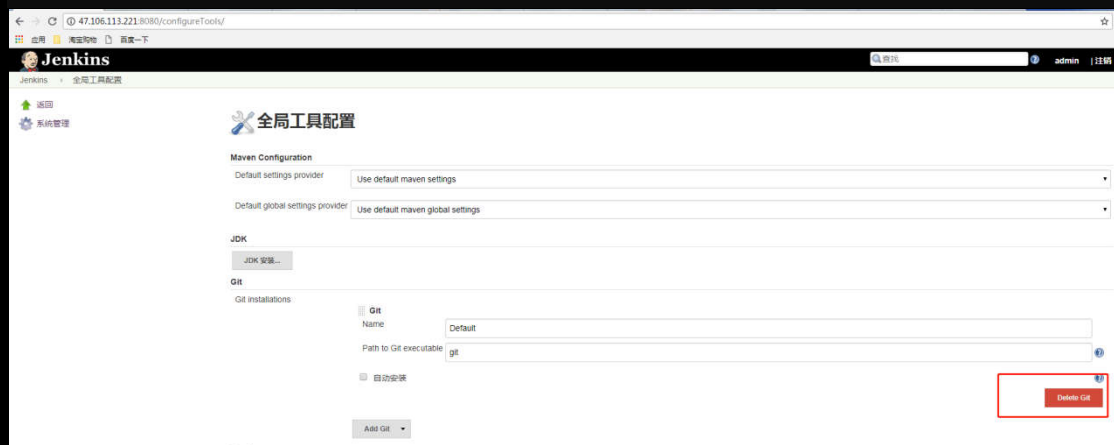
2. 添加凭据



如果对接有问题



首先去把 git 升级一下或者重装一下 git 工具，然后 git 的相关依赖包装一下



重装 git 环境：

yum remove git -y

yum install git -y

yum install curl-devel expat-devel gettext-devel openssl-devel zlib-devel gcc perl-ExtUtils-MakeMaker

源码管理

☐ 无
☒ Git

Repositories

Repository URL:

Credentials:

Branches to build

Branch Specifier (blank for 'any'):

源码库浏览器:

Additional Behaviours:

完美对接！

3、输出到控制台

构建环境

☐ Delete workspace before build starts

☐ Use secret text(s) or file(s)

☐ Abort the build if it's stuck

☒ Add timestamps to the Console Output

☐ Execute shell script on remote host using ssh

☐ With Ant

4、轮询自动构建：

定时构建示例：**H/15 * * * *** 每隔十五分钟构建一次

与计划任务一样的时间格式：**分 时 日 月 周**

构建触发器

☒ Build whenever a SNAPSHOT dependency is built

☐ Schedule build when some upstream has no successful builds

☐ 触发远程构建 (例如,使用脚本)

☐ 其他工程构建后触发

☐ 定时构建

☒ GitHub hook trigger for GITScm polling

☒ 轮询 SCM

日程表:

上次运行的时间 Wednesday, July 11, 2018 2:45:57 PM CST; 下次运行的时间 Wednesday, July 11, 2018 3:00:57 PM CST.

忽略钩子 post-commit ☐

5、打包构建

这里说一下 打包命令

The screenshot shows the Maven IDE configuration interface. The 'Build' section is active, with 'Root POM' set to 'pom.xml' and 'Goals and options' set to 'clean package'. A dropdown menu for 'Add post-build step' is open, showing various options. The '执行 shell' option is selected. The 'Post Steps' section shows radio buttons for 'Run only if build succeeds', 'Run only if build succeeds or is unstable', and 'Run regardless of build result'. The '保存' (Save) button is visible at the bottom left of the configuration area.

Pom.xml 文件是相对接和打包的标准，一般是开发来对接。不说了

重点说下：

Clean package 命令执行过程

- 1、使用清理插件：maven-clean-plugin 执行清理删除已有 target 目录；
- 2、使用资源插件：maven-resources-plugin 执行资源文件的处理；
- 3、使用编译插件：maven-compiler-plugin 编译所有源文件生成 class 文件至 target\classes 目录下；
- 4、使用资源插件：maven-resources-plugin 执行测试资源文件的处理；
- 5、使用编译插件：maven-compiler-plugin 编译测试目录下的所有源代码；
- 6、使用插件：maven-surefire-plugin 运行测试用例；
- 7、使用插件：maven-jar-plugin 对编译后生成的文件进行打包，包名称默认为：artifactId-version，比如本例生成的 jar 文件：rtp-front-1.0-SNAPSHOT，包文件保存在 target 目录下（这个生成的包不能在命令行中直接执行，因为我们还没有入口类配置到 Manifest 资源配置文件中，后续会阐述）。

注意：不管是 compile、package 还是 install 等前三个步骤都是必不可少的。

所以命令使用，最好使用符合项目需求的；

比如：clean install -Dmaven.test.skip=true 跳过测试打包，可靠性只能让程序员来完

5、开始构建 jar 包

包已经存在，通过脚本同步或者发送到目标服务器，或者对接好服务器主机直接同步过去然后执行重启操作即可完成整个构建。

六、 Jenkins 自动部署构建

自动检测代码更新状态，检测到代码更新就开始构建，构建好后执行脚本去重启程序包，并且发送告警到钉钉；

定制部署：捐助后联系群主即可。

捐助微信/支付宝： **金额随意**



声明：本文来自原创，转载需声明，如有问题请联系技术 Q 群管理/所有捐助属于自愿、捐后不予退还。