

Jenkins+Git+sshd+web

<https://blog.csdn.net/shuizhihun07/article/details/80168076>

作用：持续、自动的构建测试软件项目

持续集成

后端 《jenkins》 tomcat

CI工具

Jenkins的主要目标是监控软件开发流程，快速显示问题。所以能保证开发人员以及相关人
员省时省力提高开发效率。

CI系统在整个开发过程中的主要作用是控制：当系统在代码存储库中探测到修改时，它将
运行构建的任务委托给构建过程本身。如果构建失败了，那么CI系统将通知相关人员，然
后继续监视存储库。它的角色看起来是被动的；但它确能快速反映问题。

该系统的各个组成部分是按如下顺序来发挥作用的：

1. 开发者检入代码到源代码仓库。
2. CI系统会为每一个项目创建了一个单独的工作区。当预设或请求一次新的构建时，它将
把源代码仓库的源码存放到对应的工作区。
3. CI系统会在对应的工作区内执行构建过程。
4. （配置如果存在）构建完成后，CI系统会在一个新的构件中执行定义的一套测试。完成
后触发通知(Email,RSS等等)给相关的当事人。
5. （配置如果存在）如果构建成功，这个构件会被打包并转移到一个部署目标(如应用服务
器)或存储为软件仓库中的一个新版本。软件仓库可以是CI系统的一部分，也可以是一个外
部的仓库，诸如一个文件服务器或者像Java.net、 SourceForge之类的网站。
6. CI系统通常会根据请求发起相应的操作，诸如即时构建、生成报告，或者检索一些构建
好的构件。

总结

git安装包 jdk安装包 tomcat安装包

——部署个git

安装Git依赖包：(网络源)

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

下载安装Git:(源码包)网络源)

```
# wget https://mirrors.edge.kernel.org/pub/software/scm/git/git-2.9.5.tar.gz  
# tar xvf git-2.9.5.tar.gz  
# cd git-2.9.5  
# make prefix=/usr/local/git all && make prefix=/usr/local/git install  
# git --version //如果时1版本就先yum卸载rpm包  
git version 2.9.5
```

```
# mkdir /jenkins_test && cd /jenkins_test && git config --global user.email  
"git@localhost" && git config --global user.name "git" && git init  
# cp index.jsp /jenkins_test/ //拷贝项目到git  
# cd /jenkins_test/  
# git add index.jsp  
# git commit -m "jsp"
```

打开浏览器github.com/settings

创建新的公钥

```
# ssh-keygen && ssh-copy-id 192.168.2.39
```

```
# vim /root/.ssh/os_rsa.pub
```

将公钥发送到github

yq371952690 //github网站用户密码 yangquan19940717

```
# cd /jenkins_test/
```

```
#git remote add yangquan https://github.com/yangquan19940717/jsp.git
```

```
#git remote -v
```

```
#git push -u yangquan master
```

55%
55%

2、部署JDK

下载jdk-8u****-linux-x64.tar.gz （不要用8以上版本的jdk）

解压安装：

```
#tar xvf jdk-8u****-linux-x64.tar.gz -C /usr/local
```

```
#mv /usr/local/jdk-**** /usr/local/java
```

测试jdk：

```
#java -version
```

%
%

3、部署Tomcat

下载Tomcat并解压安装：

```
# tar xf apache-tomcat-9.0.16.tar.gz -C /usr/local/
```

```
# mv /usr/local/apache-tomcat-9.0.16 /usr/local/tomcat
```

优化：

如果8005启动较慢则做如下设置：

```
# vim /usr/local/java/jre/lib/security/java.security
```

```
securerandom.source=file:/dev/urandom //117行
```

%
%

4、部署Jenkins

http://updates.jenkins-ci.org/download/war/ //最新版本在这里下载

下载Jenkins：

```
wget https://mirrors.tuna.tsinghua.edu.cn/jenkins/war/2.165/jenkins.war
```

下载完成以后不需要解压，直接将war包放入到tomcat的网站发布目录即可、

```
#cp jenkins.war /usr/local/tomcat/webapps
```

```
#/usr/local/tomcat/bin/startup.sh
```

5、登陆Jenkins：

http://192.168.2.38:8080/jenkins

优化：

如果出现空白页面一直不跳转：

```
# vim /root/.jenkins/hudson.model.UpdateCenter.xml
<?xml version='1.1' encoding='UTF-8'?>
<sites>
  <site>
    <id>default</id>
    <url>http://mirror.xmission.com/jenkins/updates/update-center.json</url>
  </site>
</sites>
```

如果出现jenkins离线，则如下进行修改（此处是一个大坑，吐个槽！！jenkins的作者估计不知道大中华防火墙的存在！！）

```
# vim /root/.jenkins/updates/default.json
```

将谷歌的网址修改为百度。

部署

部署：

```
cat /root/.jenkins/secrets/initialAdminPassword
ls /usr/java/
cd /usr/java/jdk1.8.0_181/
wget http://mirrors.163.com/.help/CentOS7-Base-163.repo
wget https://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo
yum install -y yum-utils device-mapper-persistent-data lvm2
yum install docker-ce -y
yum install lrzsz -y
wget http://mirrors.jenkins.io/war-stable/latest/jenkins.war
mv jdk-8u181-linux-x64.tar.gz /usr/src/
mkdir /usr/java
tar -zxvf jdk-8u181-linux-x64.tar.gz -C /usr/java
vim /etc/profile
source /etc/profile
java -jar jenkins.war --httpPort=80
nohup java -jar jenkins.war --httpPort=80 &
tail -fn nohup.out
```

网站服务器部署

1、部署JDK

下载jdk-8u****-linux-x64.tar.gz （**不要用8以上版本的jdk**）

解压安装：

```
#tar xvf jdk-8u****-linux-x64.tar.gz -C /usr/local
```

```
#mv /usr/local/jdk-**** /usr/local/java
```

修改环境变量：

```
#vim /etc/profile
```

```
export JAVA_HOME=/usr/local/java
```

```
export PATH=$JAVA_HOME/bin:$PATH
```

```
#source /etc/profile
```

测试jdk：

```
#java -version
```

(需要注意如果装了openjdk的话，需要卸载openjdk)

2、部署tomcat

下载Tomcat并解压安装：

```
# tar xf apache-tomcat-***.tar.gz -C /usr/local/
```

```
# mv /usr/local/apache-tomcat-**** /usr/local/tomcat
```

3、配置tomcat

设置tomcat账号密码：

```
# vim /usr/local/tomcat/conf/tomcat-users.xml
```

```
<role rolename="admin"/>
```

```
<role rolename="admin-gui"/>
```

```
<role rolename="admin-script"/>
```

```
<role rolename="manager"/>
```

```
<role rolename="manager-gui"/>
```

```
<role rolename="manager-script"/>
```

```
<role rolename="manager-jmx"/>
```

```
<role rolename="manager-status"/>
```

```
<user username="admin" password="alan" roles="admin,manager,admin-gui ,admin-script,manager-gui,manager-script,manager-jmx,manager-status"/>
```

```
37 <!--
38 <role rolename="tomcat"/>
39 <role rolename="role1"/>
40 <user username="tomcat" password="<must-be-changed>" roles="tomcat"/>
41 <user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
42 <user username="role1" password="<must-be-changed>" roles="role1"/>
43 -->
44 <role rolename="admin"/>
45 <role rolename="admin-gui"/>
46 <role rolename="admin-script"/>
47 <role rolename="manager"/>
48 <role rolename="manager-gui"/>
49 <role rolename="manager-script"/>
50 <role rolename="manager-jmx"/>
51 <role rolename="manager-status"/>
52 <user username="admin" password="alan" roles="admin,manager,admin-gui ,admin-script,manager-gui,manager-script,manager-jmx,manager-status"/>
53 </tomcat-users>
```

设置安全登录的IP：范围

```
# vim /usr/local/tomcat/webapps/manager/META-INF/context.xml
```

```
<Context antiResourceLocking="false" privileged="true" >
  <Valve className="org.apache.catalina.valves.RemoteAddrValve"
    allow="127\.\d+\.\d+\.\d+|:::1|192.168.2.*" />
  <Manager sessionAttributeValueClassNameFilter="java\.lang\.(?:Boolean|Integer|Long|Number|string)|org\.apache\.catalina
\filters\.CsrfPreventionFilter\$LruCache(?:\$1)?|java\.util\.(?:Linked)?HashMap"/>
</Context>
```

启动tomcat：

```
# /usr/local/tomcat/bin/startup.sh
```

如果8005启动较慢则做如下设置：

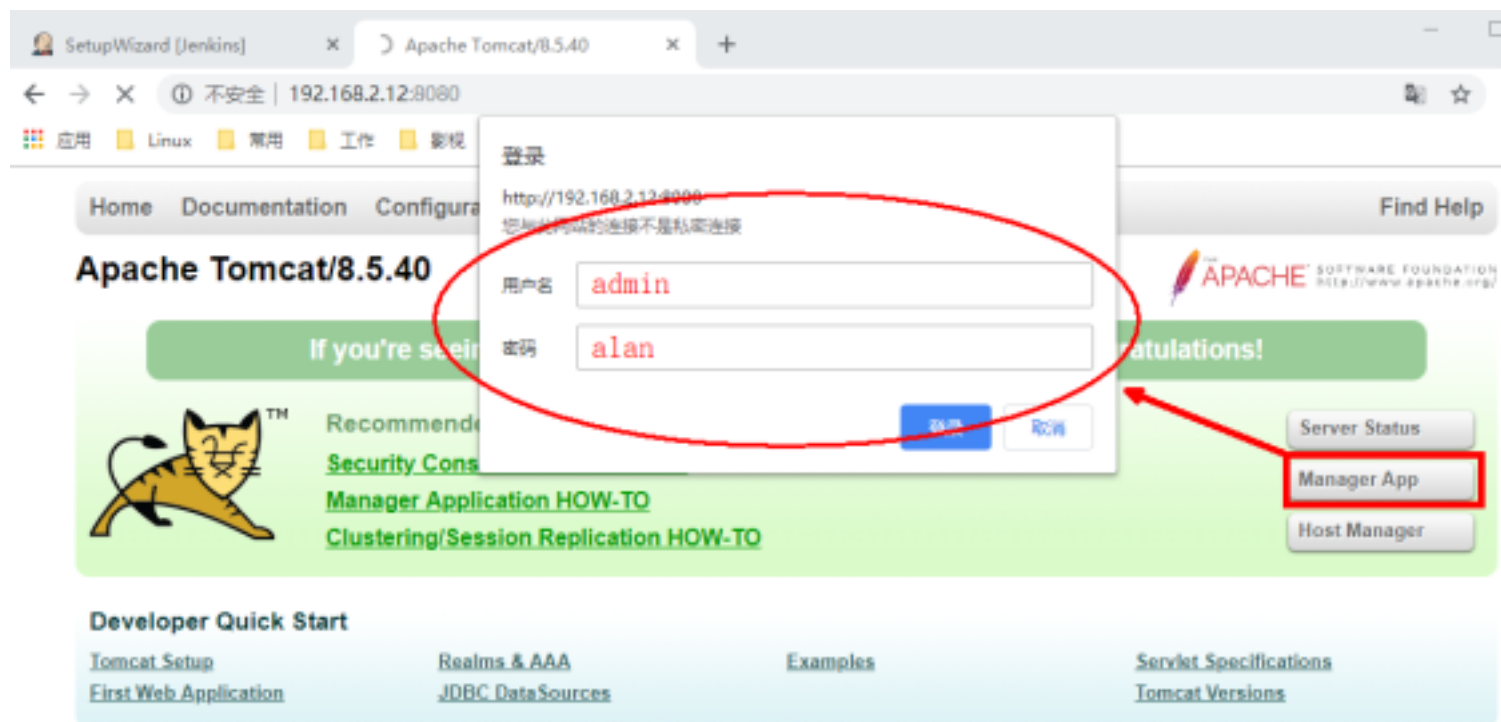
```
# vim /usr/local/java/jre/lib/security/java.security
```

```
117 securerandom.source=file:/dev/urandom
```

查看tomcat端口：

```
# netstat -tanlp|grep java
```

打开tomcat网页：http://192.168.2.12:8080





跳转如下图：

SetupWizard [Jenkins] x /manager x +

← → ↻ ① 不安全 | 192.168.2.12:8080/manager/html

应用 Linux 常用 工作 影视 娱乐

Tomcat Web Application Manager

Message: OK

Manager

[List Applications](#) [HTML Manager Help](#) [Manager Help](#) [Server Status](#)

Applications



Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

通过谷歌翻译后的页面如下：

SetupWizard [Jenkins] x /经理 x +

← → ↻ ① 不安全 | 192.168.2.12:8080/manager/html

应用 Linux 常用 工作 影视 娱乐

Tomcat Web应用程序管理器

信息: 好

经理

[列出应用程序](#) [HTML管理帮助](#) [经理帮助](#) [设备状态](#)

应用

路径	版	显示名称	运行	会话	命令
/	没有指定	欢迎来到Tomcat	真正	0	开始 停止 刷新 取消部署 空闲 时间过期 ≥ 30 分钟
/文档	没有指定	Tomcat文档	真正	0	开始 停止 刷新 取消部署 空闲 时间过期 ≥ 30 分钟
/例子	没有指定	Servlet和JSP示例	真正	0	开始 停止 刷新 取消部署 空闲 时间过期 ≥ 30 分钟
/主机经理	没有指定	Tomcat主机管理器应用程序	真正	0	开始 停止 刷新 取消部署 空闲 时间过期 ≥ 30 分钟
/经理	没有指定	Tomcat Manager应用程序	真正	1	开始 停止 刷新 取消部署 空闲 时间过期 ≥ 30 分钟

jenkins部署

环境：

两台机器：

一台部署部署Jenkins

一台部署tomcat网站(网站服务器)

关闭防火墙与selinux

1、部署Git（4G内存的Centos-64一台）

安装Git依赖包：(网络源)

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

```
export JAVA_HOME=/usr/local/java  
export GIT_HOME=/usr/local/git  
export PATH=$JAVA_HOME/bin:$GIT_HOME/bin:$PATH
```

下载安装Git:(源码包)

```
(网络源) #wget https://mirrors.edge.kernel.org/pub/software/scm/git/  
git-2.9.5.tar.gz  
#tar xvf git-2.9.5.tar.gz  
#cd git-2.9.5  
#make prefix=/usr/local/git all && make prefix=/usr/local/git install
```

```
#git --version //如果时1版本就先yum卸载rpm包  
git version 2.9.5
```

```
# mkdir /jenkins_test && cd /jenkins_test && git config --global user.email  
"git@localhost" && git config --global user.name "git" && git init  
# cp index.jsp /jenkins_test/ //拷贝项目到git  
# cd /jenkins_test/  
# git add index.jsp  
# git commit -m "jsp"
```

打开浏览器github.com/settings
创建新的公钥

```
# ssh-keygen && ssh-copy-id 192.168.2.39
```

```
# vim /root/.ssh/os_rsa.pub
```

将公钥发送到github

yq371952690

```
# cd /jenkins_test/
```

```
#git remote add yangquan https://github.com/yangquan19940717/jsp.git
```

```
#git remote -v
```

```
#git push -u yangquan master
```

2、部署JDK

下载jdk-8u****-linux-x64.tar.gz （不要用8以上版本的jdk）

解压安装：

```
#tar xvf jdk-8u****-linux-x64.tar.gz -C /usr/local
```

```
#mv /usr/local/jdk-**** /usr/local/java
```

测试jdk：

```
#java -version
```

(需要注意如果装了openjdk的话，需要卸载openjdk)

3、部署maven(war包工具)

———下载maven：———

———

```
#wget http://mirror.bit.edu.cn/apache/maven/maven-3/3.5.4/binaries/apache-maven-3.5.4-bin.tar.gz
```

———

———

```
#tar xvf apache-maven-3.5.4-bin.tar.gz -C /usr/local
```

———

———

```
#mv /usr/local/apache-maven-3.5.4 /usr/local/maven
```

———

———测试：———

———

```
mvn -version
```

———

4、部署Tomcat

下载Tomcat并解压安装：

```
# tar xf apache-tomcat-9.0.16.tar.gz -C /usr/local/
```

```
# mv /usr/local/apache-tomcat-9.0.16 /usr/local/tomcat
```

5、部署Jenkins

<http://updates.jenkins-ci.org/download/war/> //最新版本在这里下载

下载Jenkins：

```
wget https://mirrors.tuna.tsinghua.edu.cn/jenkins/war/2.165/jenkins.war
```

下载完成以后不需要解压，直接将war包放入到tomcat的网站发布目录即可、

```
#cp jenkins.war /usr/local/tomcat/webapps
```

```
#!/usr/local/tomcat/bin/startup.sh
```


如果8005启动较慢则做如下设置：

```
# vim /usr/local/java/jre/lib/security/java.security
```

```
117 securerandom.source=file:/dev/urandom
```

6、登陆Jenkins：

```
http://192.168.2.38:8080/jenkins
```

7、设置jenkins的账号密码：

查看密钥文件，并将密钥站调制Tomcat网页内：

```
# cat /root/.jenkins/secrets/initialAdminPassword
```



Please wait while Jenkins is getting ready to work ...

Your browser will reload automatically when Jenkins is ready.

出现上方这个界面不要着急，这个是启动中.....

第一步：

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

如果出现空白页面一直不跳转：

```
# vim /root/.jenkins/hudson.model.UpdateCenter.xml
```

```
<?xml version='1.1' encoding='UTF-8'?>
```

```
<sites>
```

```
<site>
```

```
<id>default</id>
```

```
<url>http://mirror.xmission.com/jenkins/updates/update-center.json</url>
```

```
</site>
```

```
</sites>
```

如果出现jenkins离线，则如下进行修改（此处是一个大坑，吐个槽！！jenkins的作者估计不知道大中华防火墙的存在！！）

```
# vim /root/.jenkins/updates/default.json
```

将谷歌的网址修改为百度。

第二步：

新手入门

自定义Jenkins

插件通过附加特性来扩展Jenkins以满足不同的需求。

安装推荐的插件

安装Jenkins社区推荐的插件。

选择插件来安装

选择并安装最适合的插件。

8、安装jenkins社区推荐的插件：

安装插件过程中可能会因为网速原因出现插件安装失败。

因此，需要你讲这些未安装的插件记录下来，到jenkins内进行手动安装未安装的插件。

新手入门

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding
✓ Timestampers	✓ Workspace Cleanup	✓ Ant	Gradle
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View
Git	Subversion	SSH Slaves	Matrix Authorization Strategy
PAM Authentication	LDAP	Email Extension	Mailer

@@ SonarSource API

@@ Struts

@@ Pipeline: Step API

@@ SCM API

@@ Pipeline: API

@@ JUnit

OWASP Markup Formatter

@@ Token Macro

Build Timeout

@@ Credentials

@@ SSH Credentials

@@ Plain Credentials

Credentials Binding

Timestampers

@@ Pipeline: Supporting APIs

@@ Dashboard Task

9、创建第一个管理员用户：

创建第一个管理员用户

用户名:

密码:

确认密码:

全名:

电子邮件地址:

完成后会给与一个jenkins分享目录，然后点击继续：

实例配置

Jenkins URL:

Jenkins URL 用于给各种Jenkins资源提供绝对路径链接的根地址。这意味着对于很多Jenkins特色是需要正确设置的，例如：邮件通知、PR状态更新以及提供给构建步骤的BUILD_URL环境变量。

推荐的默认值显示在尚未保存，如果可能的话这是根据当前请求生成的。最佳实践就是要设置这个值，用户可能会需要用到。这将会避免在分享或查看资源时出现问题。

点击使用jenkins：

Jenkins已就绪！

Jenkins安装已完成。

[开始使用Jenkins](#)

如果出现白色页面不跳转：

```
# vim /root/.jenkins/config.xml
```

```
<authorizationStrategy class="hudson.security.AuthorizationStrategy$Unsecured"/>
```

弹出页面后如果出现方向代理设置有误，这个是因为没有做相应配置造成的，不过不需要管他。

10、安装必须插件：

a、安装Deploy to container

作用：自动化将代码部署到tomcat容器

安装：



第二步、（搜索的时候可能因网络延迟，造成搜索过程会比较慢）

a、安装GitHub Authentication、GitHub Branch Source Plugin（该插件如果找不到，就去已安装中查看，默认是安装的）、GitHub Organization Folder 这三个插件

作用：调用Git的相关插件

安装：方法同第一个插件

b、安装Publish Over SSH

作用：服务器之间传送文件

安装：方法同第一个插件（这个是最后一个插件，所以点击安装完成后重启jenkins，重启过程很漫长.....请等待，下方是重启成功后的登陆界面）



欢迎来到 Jenkins!

登录

☐

保持登录状态

11、检查错误配置

a、如果出现“编码问题”错误，则在Tomcat----conf---server.xml文件中修改即可。

```
#vim /usr/local/tomcat/conf/server.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
```

b、如果有反向代理错误，请忽略。

c、由于更新较快，如果遇到版本漏洞问题，请自行下载更高版本的jenkins进行安装.

管理Jenkins

反向代理设置有误

更多信息

不再显示

12、jenkins系统配置：



点击系统管理----Global Too configuration（全局工具配置），配置JDK，GIT的路径。

- a、没有特殊情况，请选择默认配置
- b、全程禁止点击自动安装

JDK

JDK 安装

新增 JDK

图1

JDK

别名

java

JAVA_HOME

/usr/local/java

☐ 自动安装

删除 JDK

Git

Git installations

Git

Name

Default

Path to Git executable

/usr/local/git/bin/git

☐ 自动安装

Delete Git

图2

13、ssh配置

目的：Jenkins服务器上的maven将开发、生产使用的*.war通过ssh推送到远程的Tomcat服务器上。

a、准备双机互信：

#ssh-keygen

#ssh-copy-id Tomcat服务器IP

b、在Jenkins上配置ssh信息：

①、进入系统设置：

系统管理-----系统设置-----Publish over SSH

②、准备ssh私钥：

cat /root/.ssh/id_rsa

复制私钥串粘贴到Jenkins系统设置内。


```
[root@localhost jenkins]# cat /root/.ssh/id_rsa
```

```
-----BEGIN RSA PRIVATE KEY-----
```

```
MIIEowIBAAKCAQEA vQgrPqr0Ix0Y4etvGniTjMljGbccaS5BJBQ9rvylm8qgio0r  
Bm0BvPGsmJCLH5P5zIC48u3Ki/010kkDWRTDYgWlFWEu5h0TD/WjGu1HsbogRok0  
sdZ2DyMoXKTqhViovTc8gKXoGrnJpfYAeyWBtXdh79yXgU5eAcR0kP3If1wB9Em0  
XBL3SR/co2S2i00gZaLzvqC0YiT8J+tAWxh59tUKXNNTXMkCR/QivNnFQCNGIYS0  
ZKnXmmzppJ8LRDM70eZD9TyN+a80m9LXb1VBih6Xy2wu+jIg+3MSk2TXhzqRF6LJ  
LV0aqBpXpzogQSxJf/EYAl1PoU/50XX36WTnoQIDAQABAOIBAGwTpZX47Ai6Baj7  
kNpTohjYvRZQyR+h4ICesf4oYjUmjs9okvbeIalDi+jWi5SMaD/yQMw0BKAXLy02  
5x4hbuw2SgKFQ+kxKlrvmeEekp0cKgbWSwyJ2eZ2B/muppqSC4kb8s8ziDmhpXM9U  
bUUNfbYnGianur+lbL4dlCj10JJvrQbTfb6LrqqsTFsTYWw//UCfFrw8hE+q0CFb  
ieAQbuFdoHfJIO4+kUwFMFewIrWYrmxsKh5sn0MkI8kaSFqeveIi9pmb03hrgoDL  
C5uSlwfpblvCL0PRpifCUgByEczZaiIxtcc2atEvsGqPiSKDQ/UGKkEbXhbTzGIA  
L/BoHHkCgYEA+KGh5G3ZPdnZy40inWXG4D7dvSia0igVYkHPgFOHATiNciIAdCY  
UNRiSsp4cv1NRiQY0jnp1rC5DkMH80QMC1JVc/WNtzpwIYalWzmREMRgQ098FKmI  
NWfRpCR7DqvViWXPnLnoghoAD1fTPy7ZFGn3adnu8PrI18E04uV7P4cCgYEAwqJa  
50DBpe7urT8WvPTH0Wdp5jSpJb2c4Jw6XQHhviU8hxlLW8gG2XhV0F7tAiY6JS0v  
5UBcpEjW/r/fiB+WskbaarcfcE3r3k9A/roTfcDmjIbb2GmNxaKtd9e7NxESwDS6  
L9/z5ho9an8XKfhYtfqXXvAAE8hcg033UmCf2ZcCgYB36MQCPfQyvzkp0URrcqVc  
iwc37jhUuJ4viBvCAQawUQzuiXV8js7VVMie120l0AIPn0MXqHSdZReyCsyaJQ0  
dc/eoA0b5aQklyJyj53l3z+pltc5ylVlpD9YEXwDxF206RP5WFqjvFRhEfNAmqB5  
Fu4K798r+lcXrvU1+oTpSwKBgFNLhhizc0xfH7f26vU80FHpxvFn2ikzp4KceY6  
P3DPoMK0eYKy7PN+w0LkjQvBblrgpsG2tK7+GTx2wuQPmJL7FZhb/FFkl4/RXSBn  
Y9cKieAUctvn+GtfmZszkQR1AxBMvdxLPKb6DdSYxUIJ7XUb5FIV5gKRwb03mg0g  
JGjLAoGBANJpy1RfxYYSXIKGBlCP1FjsTIT48uVYSM2hpQkXxCnGnTgbVaY1BtLs  
Bxte/UXW4jTBZlaAAt4fQm0x2UHEADx3krBVc6fR8YIujwr8wCTpoy3bHYfeZdc0  
JuTcZVdq8NgZPDH/8ds4TR4eKLDrnbf0eP7vZqMQB7I/iTmHomSU
```

```
-----END RSA PRIVATE KEY-----
```

```
[root@localhost jenkins]#
```

③、粘贴到Jenkins系统配置：

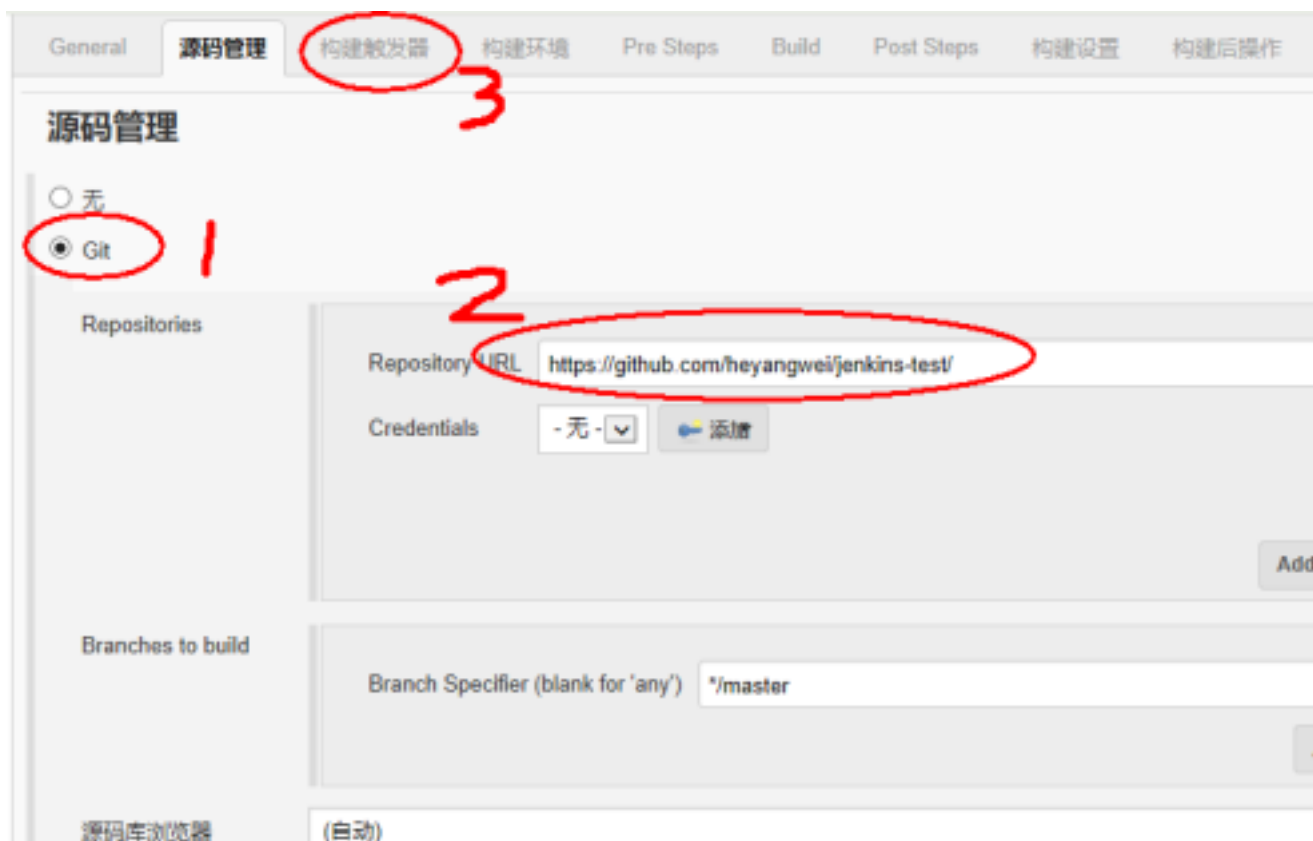
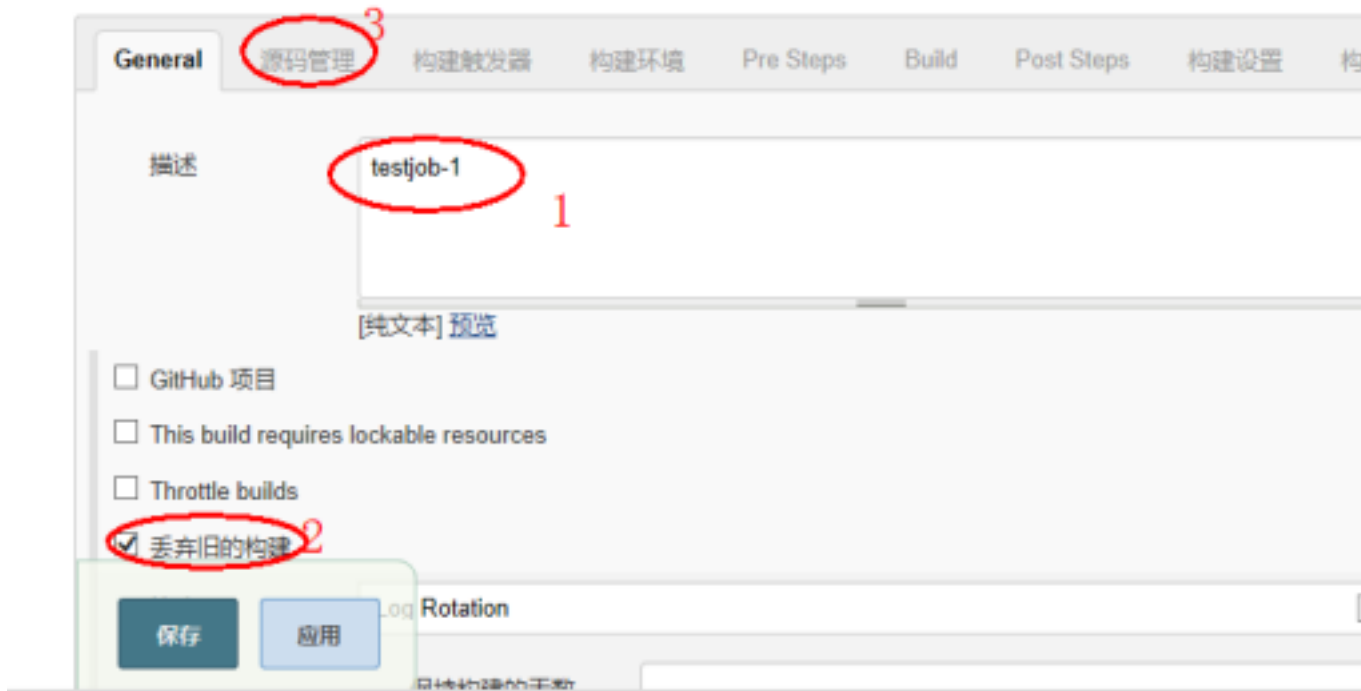


14、构建项目：

a、创建任务并定义任务名称和类型：主页----新建任务



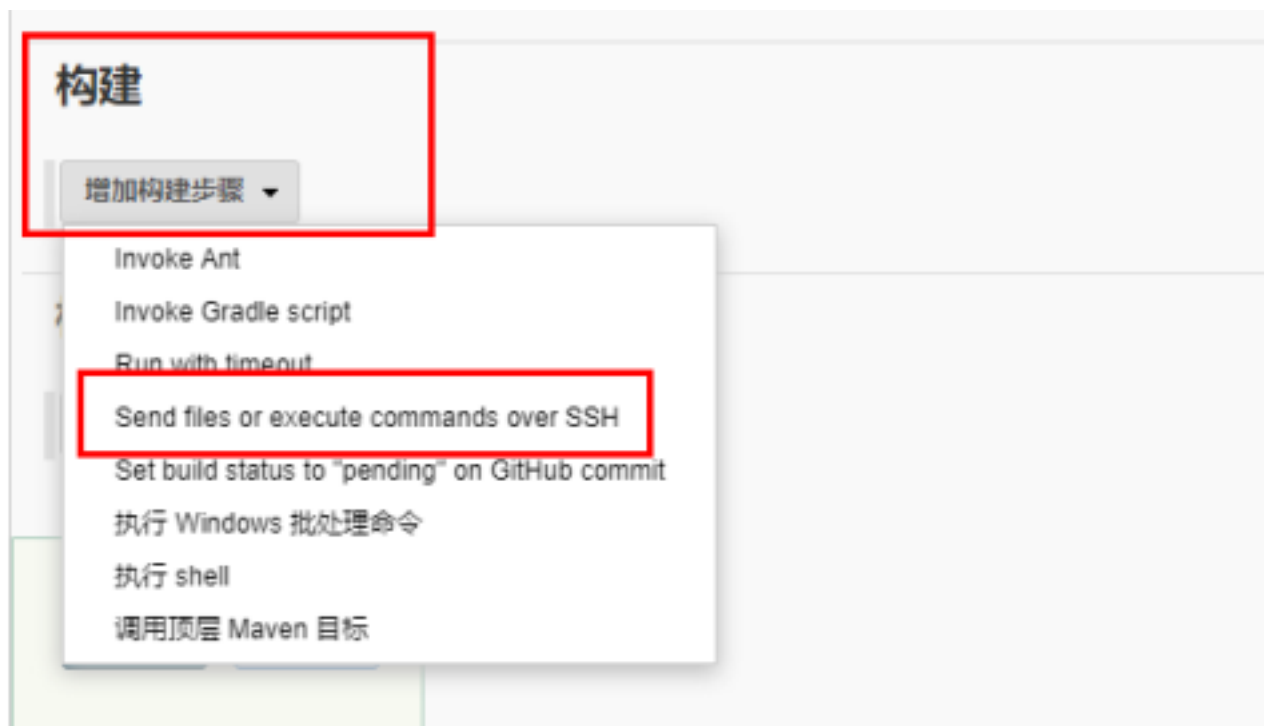
b、设置源码管理



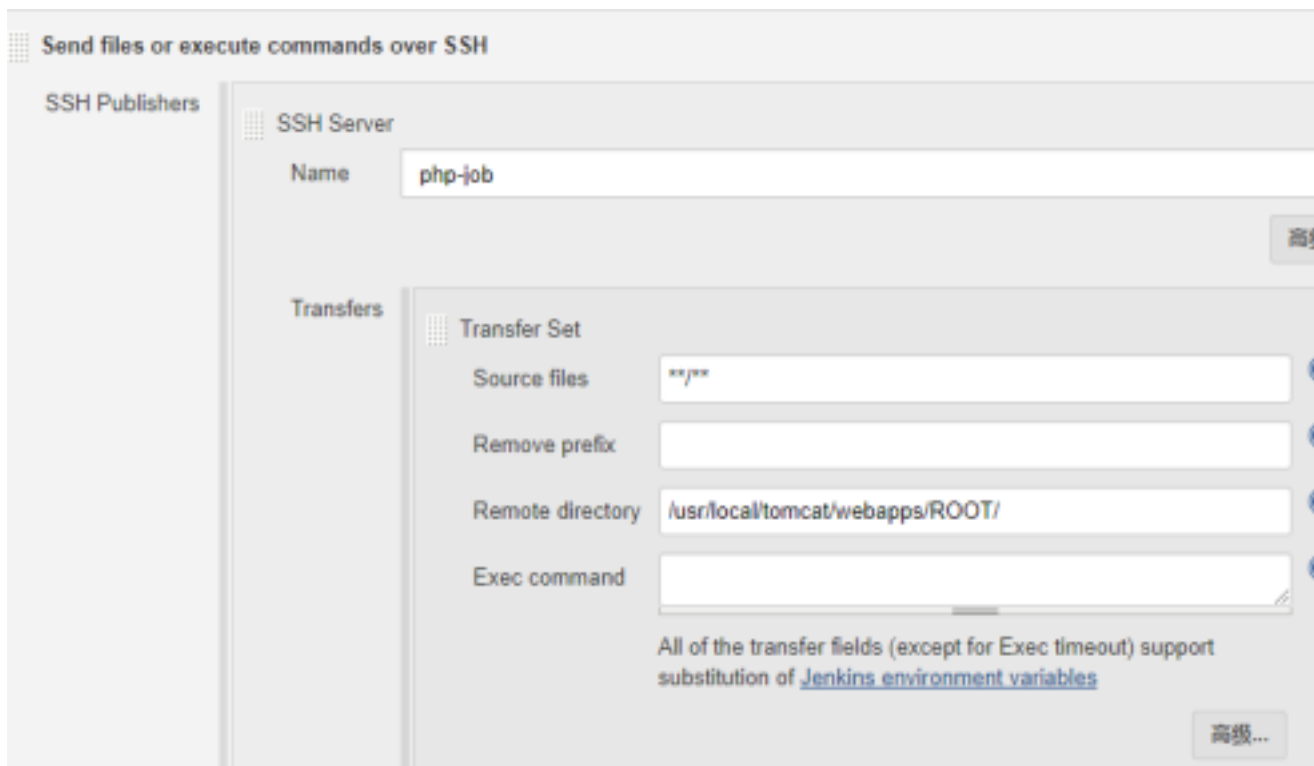
c、设置构建触发器：默认不需要动



d、构建操作：



设置ssh发送构建工件



①ssh server：是13-b-③的配置名称

②source files： Jenkins设置完成之后Jenkins会自动创建该路径，用存放GitHub所推送过来的文件

③remove prefix：自动删除路径前缀(可省略)。

④remote directory：这一项所指定的路径需要提前在Tomcat服务器上创建，用来存放网站源代码。

⑤Exec command：指定Tomcat收到源代码之后需要执行的动作脚本。

15、测试（立即构建）

查看文件是否推送到了Tomcat主机

控制台输出

由用户 [alan](#) 启动

Running as SYSTEM

构建中 在工作空间 /root/.jenkins/workspace/job 中

No credentials specified

```
> /usr/local/git/bin/git rev-parse --is-inside-work-tree # timeout=10
```

Fetching changes from the remote Git repository

```
> /usr/local/git/bin/git config remote.origin.url git@github.com:heyangwei/jsp.git # timeout=10
```

Fetching upstream changes from git@github.com:heyangwei/jsp.git

```
> /usr/local/git/bin/git --version # timeout=10
```

```
> /usr/local/git/bin/git fetch --tags --progress git@github.com:heyangwei/jsp.git +refs/heads/*:refs/remotes/origin/
```

```
> /usr/local/git/bin/git rev-parse refs/remotes/origin/master^{commit} # timeout=10
```

```
> /usr/local/git/bin/git rev-parse refs/remotes/origin/origin/master^{commit} # timeout=10
```

Checking out Revision fd1450f0be25800fccc94b936f886e7711282c08 (refs/remotes/origin/master)

```
> /usr/local/git/bin/git config core.sparsecheckout # timeout=10
```

```
> /usr/local/git/bin/git checkout -f fd1450f0be25800fccc94b936f886e7711282c08
```

Commit message: "jsp"

First time build. Skipping changelog.

SSH: Connecting from host [jenkins]

SSH: Connecting with configuration [php-job] ...

SSH: Disconnecting configuration [php-job] ...







SSH: Transferred 1 file(s)

Build step 'Send files or execute commands over SSH' changed build result to SUCCESS

Finished: SUCCESS

登录Tomcat网站验证

← → ↺ ⓘ 不安全 | 192.168.2.12:8080

 应用  Linux  常用  工作  影视  娱乐

jenkins-----OK!!

插件安装失败实例

✔ JDK Tool	✔ Build Timeout	✘ Credentials	✘ SSH Credentials
✘ Plain Credentials	✘ Credentials Binding	✔ Ant	✔ Pipeline: Milestone Step
✘ Pipeline: Declarative Extension Points API	✘ JSch dependency	✘ Git client	✘ GIT server
✘ Pipeline: Shared Groovy Libraries	✘ Authentication Tokens API	✘ Docker Commons	✘ Docker Pipeline
✘ Pipeline: Declarative Agent API	✘ Pipeline: Declarative	✘ Pipeline	✔ GitHub API
✘ Git	✘ GitHub	✘ GitHub Branch Source	✘ Pipeline: GitHub Groovy Libraries
✘ Subversion	✘ SSH Slaves	✔ LDAP	

Localization: Chinese (Simplified)

如果安装冒红就把冒红的记下来。再进入到jenkins软件里安装插件