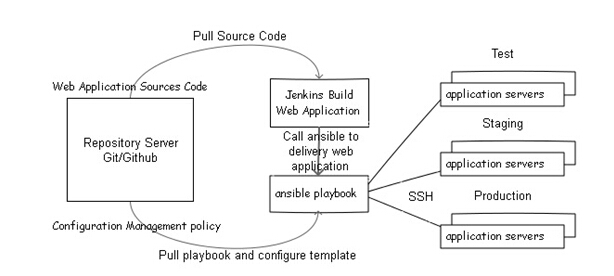
**Jenkins + Ansible + Git 部署**

## **实例**



将应用源代码和playbook及服务器配置文件存放在git服务器中，将需要部署的服务器在hosts中相应分成test，staging，production三个组。

部署的时候，运维人员在ansible服务器上用git从git服务器中取playbook并执行部署，可以一次部署三个环境，也可以选择指定的环境部署;在环境准备好后，开发人员通过jenkins工具取代码，build并构建部署任务。用jenkins调用ansible并执行相应的部署playbook进行自动应用部署，这里的playbook可以定义先判断在test环境是否部署正常，如正常则把当前应用部署到staging环境，否则将test环境回退到之前状态。

## **准备工作**

## JDK安装

## 本试验中Tomcat和Jenkins都需要依赖JDK，其中最新版的Jenkins需要JDK1.8版本，采用yum安装：

## yum install java-1.8.0-openjdk –y

## 安装完毕后执行java -version 确认一下，看到版本号表示成功

1. Git安装

yum -y install git

安装完后执行 git --version 确认，看到版本号表示成功

1. Maven安装

yum -y install maven

安装完后执行 mvn -v确认，看到版本号表示成功

1. Ansible安装

yum install ansible -y

## **安装Jenkins**

1. 下载程序包：

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

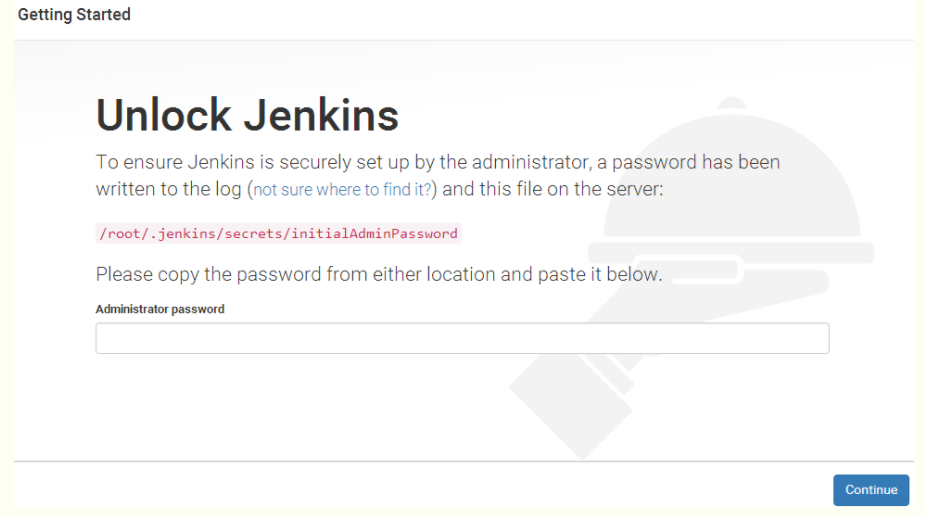
1. 启动程序包：

下载的[Jenkins程序war包可以用Tomcat发布，也可以直接执行启动，本试验中Tomcat 服务用于发布JavaWeb项目，为避免混淆，采用](http://jenkins-ci.org/)直接执行启动。

启动命令：java -jar jenkins.war --httpPort=8081

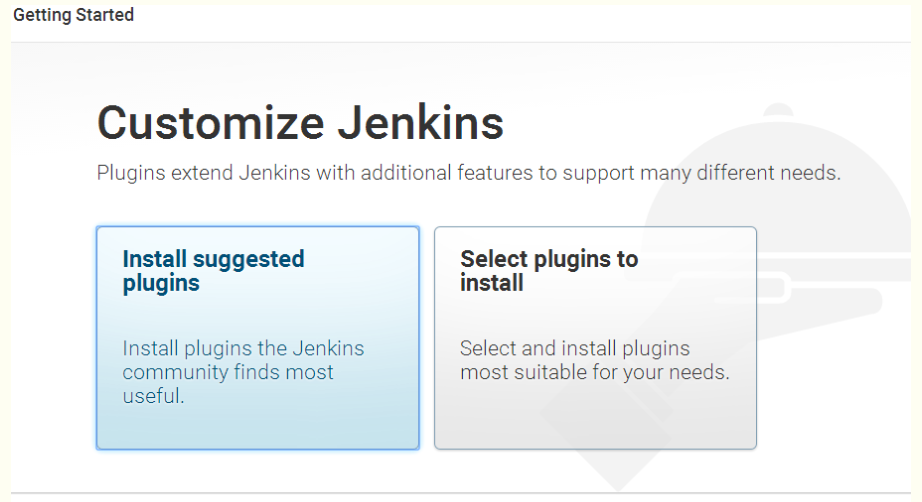
如果不写端口号参数，默认启动端口是8080

初次启动控制台里会生成一个密码，对其进行复制

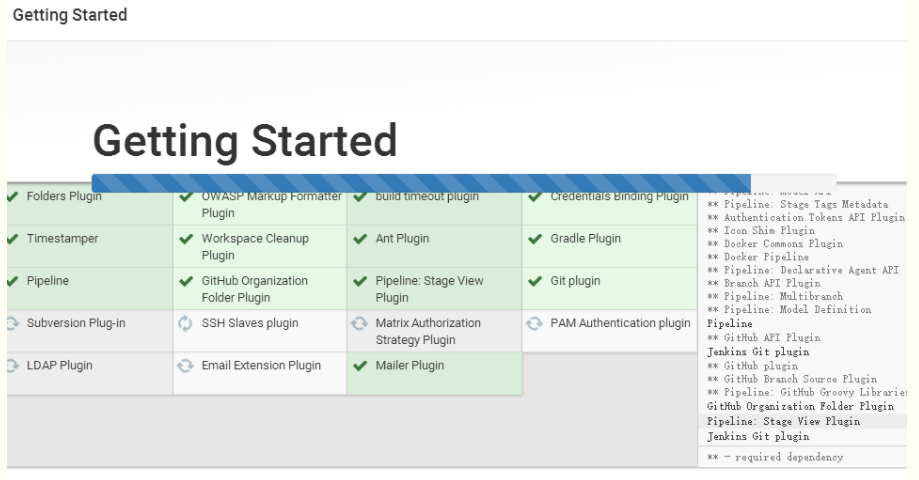
1. 打开访问地址：http://主机IP:8081，出现如下页面

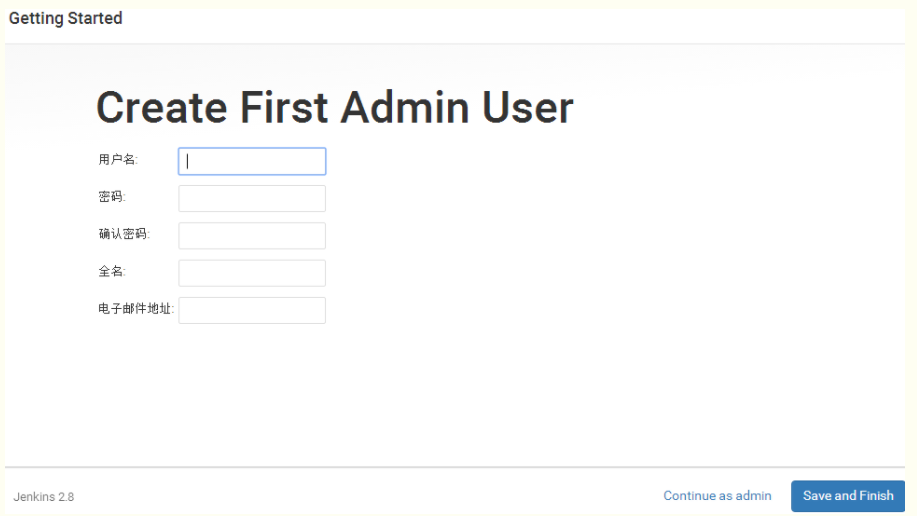
将刚才的密码进行粘贴，也可根据提示到主机上

cat /root/.jenkins/secrets/initialAdminPassword 找到密码进行复制粘贴。来到如下页面

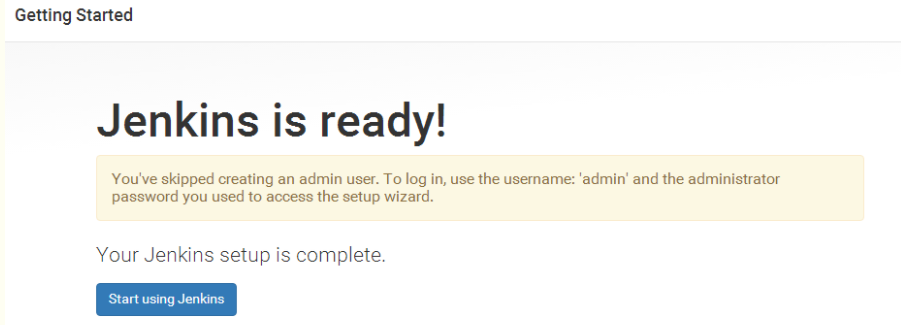


这步是让选择推荐安装还是自定义安装，之后可以随时更改，点第一个，知道用什么插件点第二个也可以这样装的包少一点。安装的插件都在/root/.jenkins/plugins/下面





直接选择continue as admin



开始使用，点击就会进来



重置admin密码：

以后访问时会需要账号密码，在“系统管理”-“管理用户”中对admin账户进行密码重置修改。

1. 至此完成Jenkins安装以及初次登录。

此时Jenkins运行在CentOS当前页面进程里，如果关掉当前会话会导致Jenkins服务停止，可先将服务停掉，运行如下命令进行后台启动：

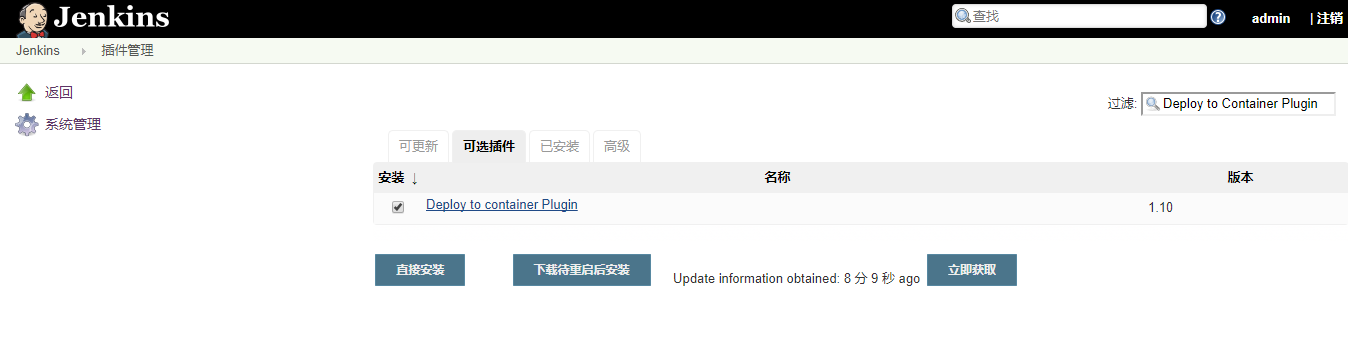
nohup java -jar /root/jenkins.war --httpPort=8081 &

如要停止服务，可找到进程pid号，执行 kill -9 pid号

## **配置Jenkins**

1. 插件安装：
2. Maven工程插件





选择可选插件，过滤搜索Maven Integration plugin 插件

勾选，点击下边的按钮：直接安装，这个可能时间较久，等待即可。

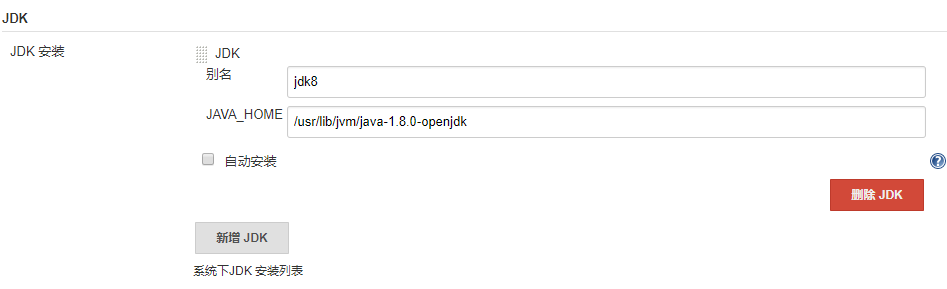


1. 系统配置



1. JDK，别名是任意的,路径填写刚才yum安装的jdk位置

/usr/lib/jvm/java-1.8.0-openjdk

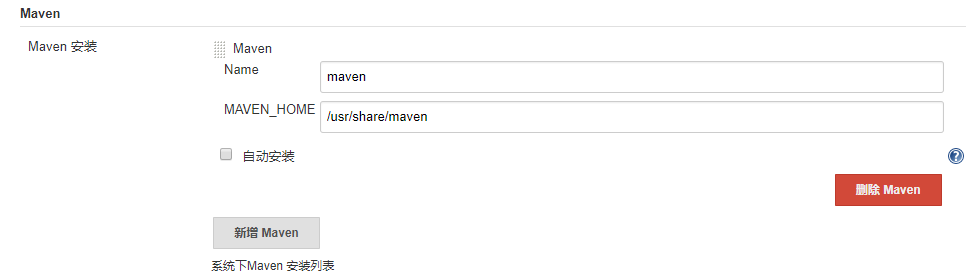


1. Git，注意这里的git位置，是可执行文件的地址（类似于[Java](http://lib.csdn.net/base/java)中bin下的java可执行文件位置），可通过 git --exec-path 命令查看路径，本试验yum安装的git可执行文件位置为/usr/libexec/git-core/git



1. Maven，选择刚才yum安装的maven存放位置

/usr/share/maven



至此Jenkins的基本配置完成。

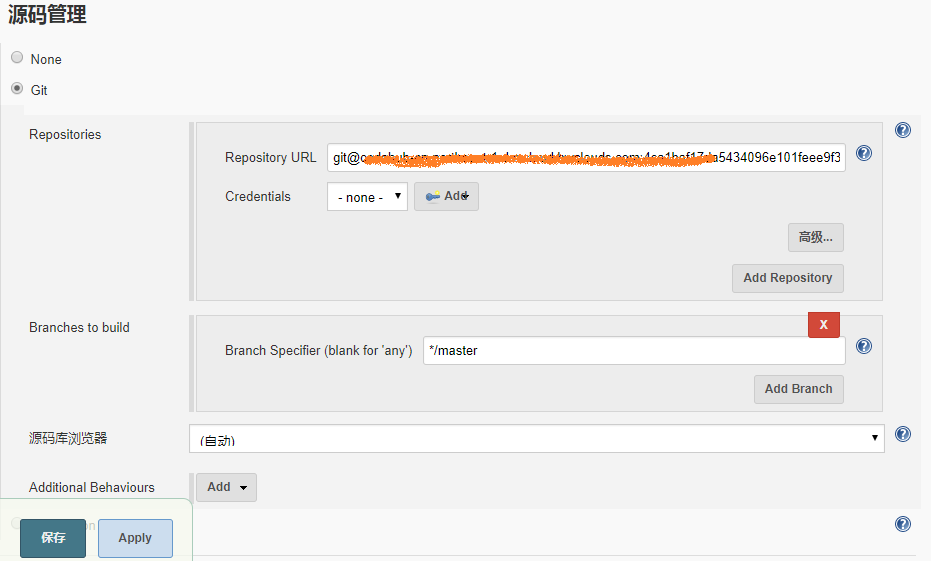
## **创建任务**

1. 新建任务



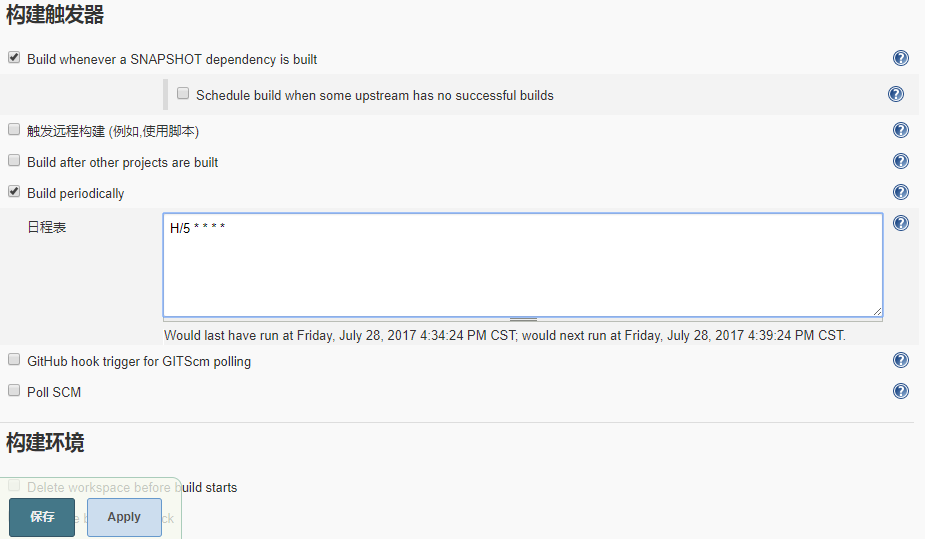
1. 配置任务
2. 源码管理

源码管理选择Git，这个时候添加url之后，下边会报错，显示让去认证，认证即可（其他的安装中又遇到这个问题），如果认证失败，请下载认证Github Authentication plugin插件，这个在插件管理的可选插件中搜索安装



**\***需要在jenkins服务器端生成密钥，并且添加到相应的git仓库服务器端

1. 构建触发器



此处默认为第一个选项，可以根据实际业务选择其他选项，

例如：

Build periodically：周期性触发执行

Poll SCM：周期性扫描代码仓库，源代码发生变化触发执行

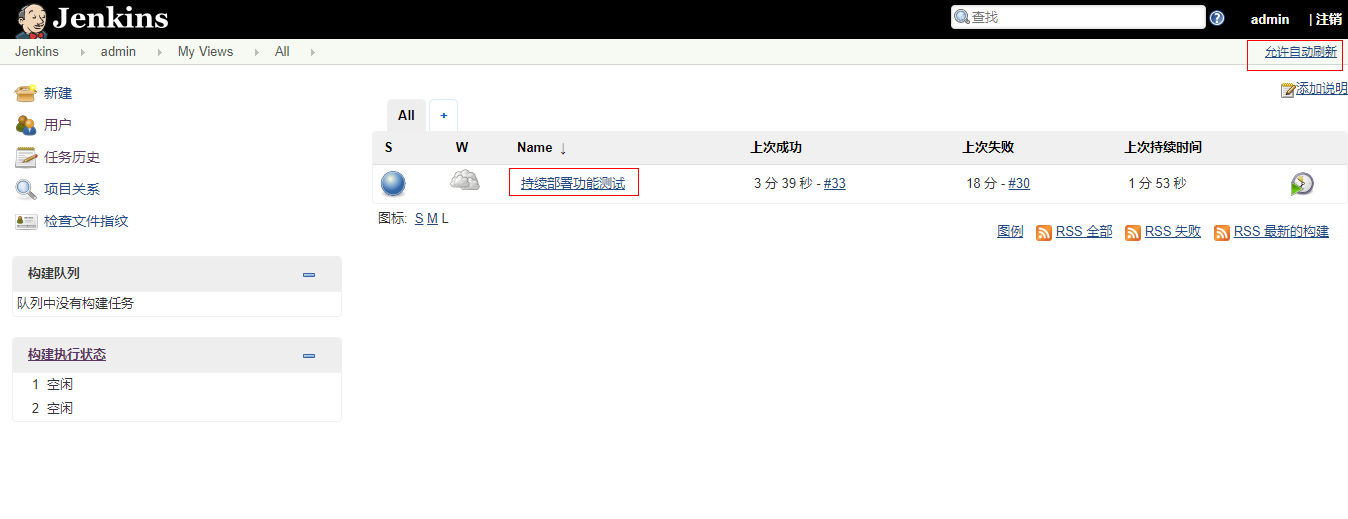
配置内容可以点选后面的问号圆圈查看填写规则说明

此试验配置为：每五分钟执行一次，H/5 \* \* \* \*

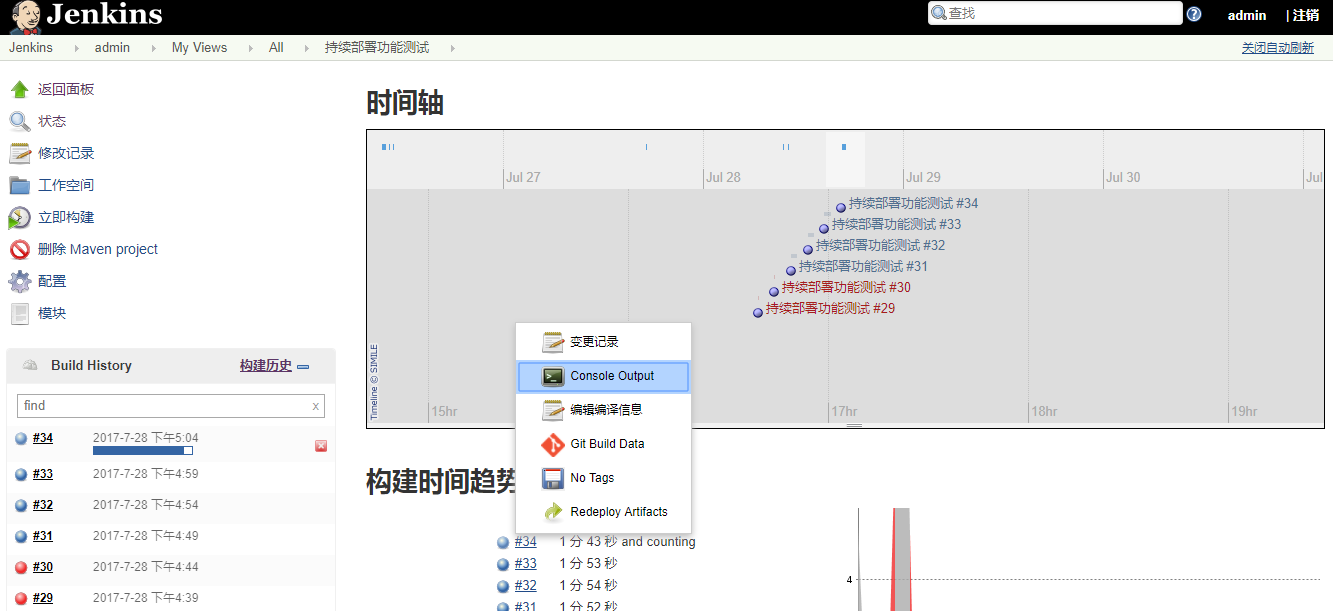
1. 至此任务创建配置完成

## **查看任务**

此时可以点击“立即构建”，也可以等待定时任务自动触发。









## **配置Ansible**

1. 为Ansible定义节点清单，编辑/etc/ansible/hosts，加入（123.56.219.140是我们要进行部署的服务器）：

[test-servers]

123.56.219.140

1. 在Ansible服务端生成密钥，并且复制公钥到节点中。

使用ssh-copy-id命令来复制Ansible公钥到节点中。

ssh-copy-id -i [root@123.56.219.140](mailto:root@123.56.219.140)

ps：如果出现host key验证失败的提示，可以进入/root/.ssh/ known\_hosts文件中，把之前的节点信息删除。

1. 尝试在Ansible服务端运行命令

使用ping检查‘test-servers’或者ansible节点的连通性。

[root@jdu4e00u53f7 ansible]# ansible -m ping 'test-servers'

123.56.219.140 | SUCCESS => {

"changed": false,

"ping": "pong"

}

这里的test-servers 就是在/etc/ansible/hosts设置的主机组名称

返回结果SUCCESS，证明成功

1. 在ansible主机的/devcloud/Resource/目录下，放置所需要如下安装包

/devcloud/Resource/java/libpng-1.2.49-2.el6\_7.x86\_64.rpm

/devcloud/Resource/java/java-1.7.0-openjdk-1.7.0.45-2.4.3.3.el6.x86\_64.rpm

/devcloud/Resource/tomcat/apache-tomcat-7.0.61.tar.gz

1. 编写playbook脚本

|  |
| --- |
| - hosts: test-servers  remote\_user: root  tasks:  - name: copy libpng rpm 6  copy: src=/devcloud/Resource/java/libpng-1.2.49-2.el6\_7.x86\_64.rpm dest=/tmp/trial/ mode=0500    - name: install libpng 6  command: chdir=/tmp/trial rpm -ivh libpng-1.2.49-2.el6\_7.x86\_64.rpm  become: True  become\_method: sudo  ignore\_errors: yes  # - name: copy jdk-1.7.0-openjdk rpm 6  # copy: src=/devcloud/Resource/java/java-1.7.0-openjdk-1.7.0.45-2.4.3.3.el6.x86\_64.rpm dest=/tmp/trial/ mode=0500  # when: ansible\_distribution == "CentOS" and ansible\_distribution\_major\_version == "6"  # - name: install openjdk 6  # command: chdir=/tmp/trial rpm -ivh java-1.7.0-openjdk-1.7.0.45-2.4.3.3.el6.x86\_64.rpm  # become: True  # become\_method: sudo  # ignore\_errors: yes  - name: install openjdk 6  shell: yum -y install java-1.7.0-openjdk.x86\_64  - name: add group "tomcat"  group: name=tomcat  - name: add user "tomcat"  user: name=tomcat group=tomcat home=/usr/share/tomcat createhome=no  become: True  become\_method: sudo  - name: mkdir tomcat tomcat\_wksp  file: path=/usr/local/devcloud owner=tomcat group=tomcat state=directory mode=0540  become: True  become\_method: sudo  - name: Download Tomcat  copy: src=/devcloud/Resource/tomcat/apache-tomcat-7.0.61.tar.gz dest=/usr/local/devcloud/apache-tomcat-7.0.61.tar.gz mode=0640  #get\_url: url={{tomcat\_url}} dest=/usr/local/devcloud/apache-tomcat-7.0.61.tar.gz  - name: Extract archive  command: chdir=/usr/share /bin/tar xvf /usr/local/devcloud/apache-tomcat-7.0.61.tar.gz -C /usr/local/devcloud/ creates=/usr/local/devcloud/apache-tomcat-7.0.61  - name: copy war  copy: src=/root/.jenkins/workspace/Demo/target/demo-0.0.1-SNAPSHOT.war dest=/usr/local/devcloud/apache-tomcat-7.0.61/webapps/ mode=0500  - name: Start Tomcat  shell: 'nohup /usr/bin/java -Djava.util.logging.config.file=/usr/local/devcloud/apache-tomcat-7.0.61/conf/logging.properties -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djava.endorsed.dirs=/usr/local/devcloud/apache-tomcat-7.0.61/endorsed -classpath /usr/local/devcloud/apache-tomcat-7.0.61/bin/bootstrap.jar:/usr/local/devcloud/apache-tomcat-7.0.61/bin/tomcat-juli.jar -Dcatalina.base=/usr/local/devcloud/apache-tomcat-7.0.61 -Dcatalina.home=/usr/local/devcloud/apache-tomcat-7.0.61 -Djava.io.tmpdir=/usr/local/devcloud/apache-tomcat-7.0.61/temp org.apache.catalina.startup.Bootstrap start &'  #service: name=tomcat state=started enabled=yes  become: True  become\_method: sudo  - name: insert iptables rule 6  lineinfile: dest=/etc/sysconfig/iptables create=yes state=present regexp="8080" insertafter="^:OUTPUT "  line="-A INPUT -p tcp -m state --state NEW -m tcp --dport 8080 -j ACCEPT"  when: ansible\_distribution == "CentOS" and ansible\_distribution\_major\_version == "6"  notify:  - service: name=iptables state=restarted  become: yes  become\_method: sudo |

1. 在Jenkins的构建任务里，点击配置-Post Steps，选择Execute shell，添加如下shell命令：

ansible-playbook /tmp/bielei/00-setup-CentOS6.yml

1. 如第一部分实例中所述，对于playbook的脚本尽量使用Git仓库进行管理。