## 如何利用Jenkins+ansible+maven（java）方式自动化更新微服务

来源：华云公众号 2019-03-12

链接: https://www.huayun.com/news/583.html

一般在开发过程中，都会出现在联调的过程中频繁修改代码，然后将最新代码部署到开发环境中，提供给其他微服务的小伙伴联调使用。传统的方式都需要手动编译代码并打包，然后将打包传到开发环境中，最后重启服务，这种方式非常耗时耗人力。为了节省开发联调的时间，可以使用Jenkins+ansible+maven（java）的方式自动打包上传到开发环境并自动重启服务。

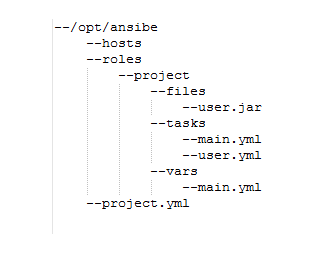
**环境准备**

在服务器中安装好JDK1.8、Jenkins、ansible、git客户端等基础软件，这里就不详细介绍每个软件如何使用。

**编写好ansible脚本**

在服务器事先准备好微服务的ansible脚本。

ansible文件目录：



文件说明：

1.       hosts文件主要记录要更新微服务的服务器地址详情，这里只需要ip地址和端口号，这里的地址详情可以随意写，下面会在Jenkins任务中动态修改，文件内容如下：

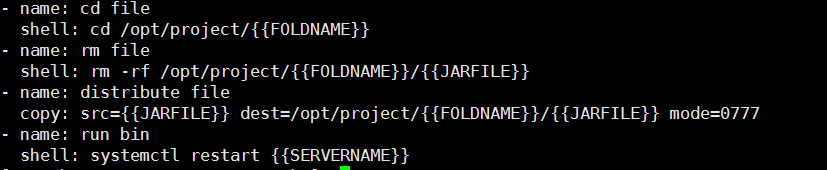
2.png

2.       roles文件夹下表示是每个项目执行ansible所需要的配置和文件。roles/project文件夹表示执行的一个任务，project文件夹就是该任务所需要的ansible配置和文件，roles/project/files是ansible任务执行所需要的文件，roles/project/tasks是任务执行的配置，这里可以配置需要在目前服务器执行的shell脚本，roles/project/vars是任务执行的所需要的参数,注意var下的配置都会在下面的Jenkins任务中动态修改。

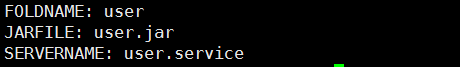
roles/project/task/main.yml文件内容如下：

3.png

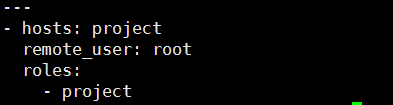
roles/project/task/user.yml文件内容如下：



roles/project/var/main.yml文件内容如下：

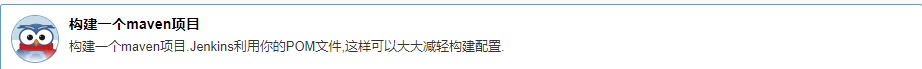


1.       project.xml是执行project这个任务的关联主机和roles信息，文件内容如下：

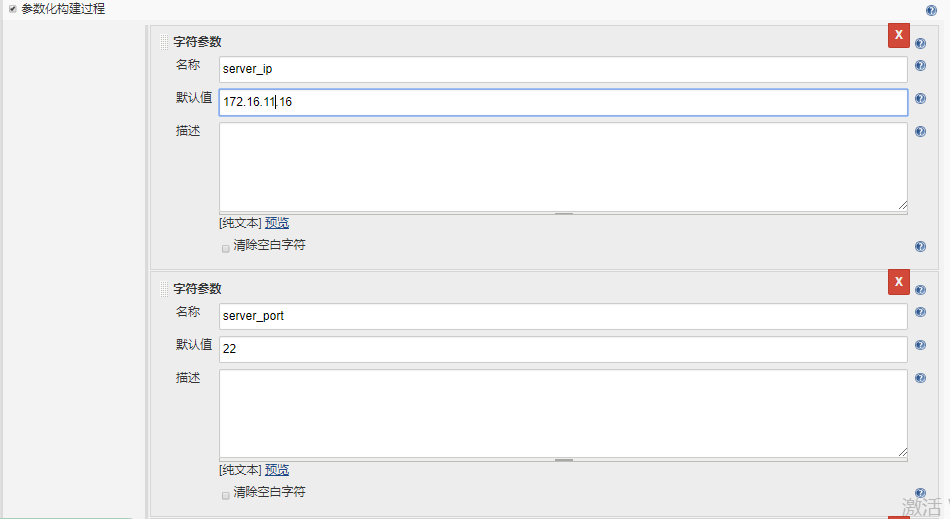


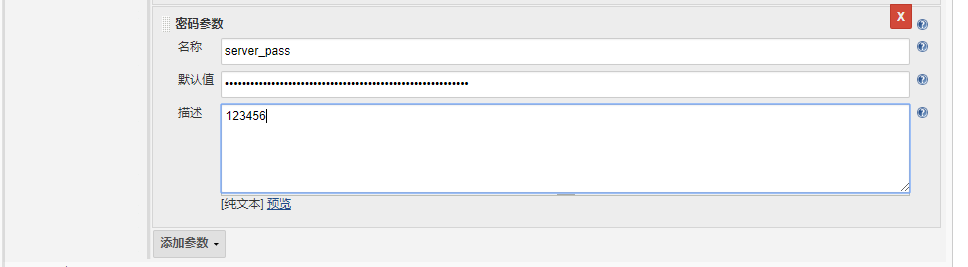
**Jenkins新建任务**

1.       新建一个maven项目任务，这里maven项目中必须要包含pom.xml文件。

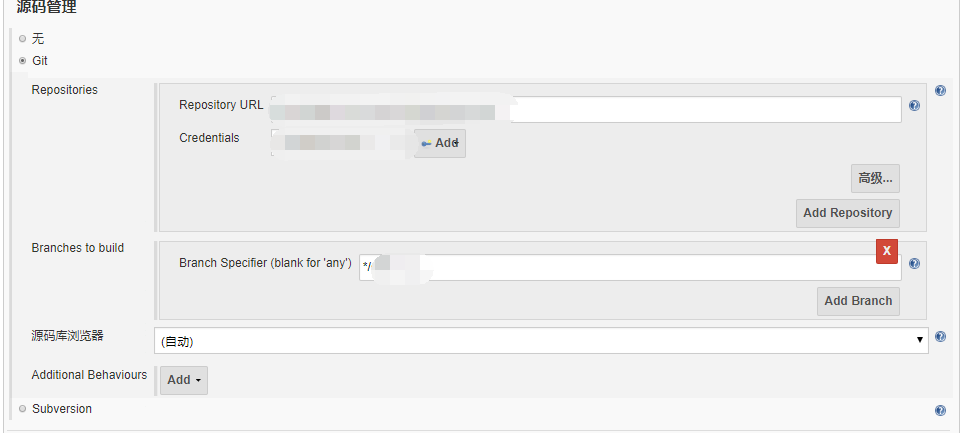


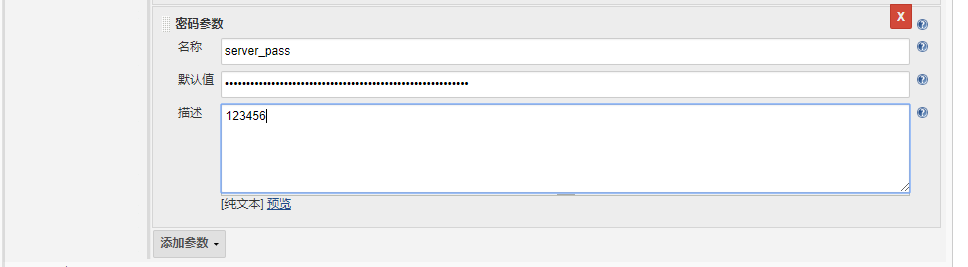
2.       在新建的任务中添加构建过程中的所需要的参数，这里只要增加server\_ip、server\_port、server\_pass这2个参数



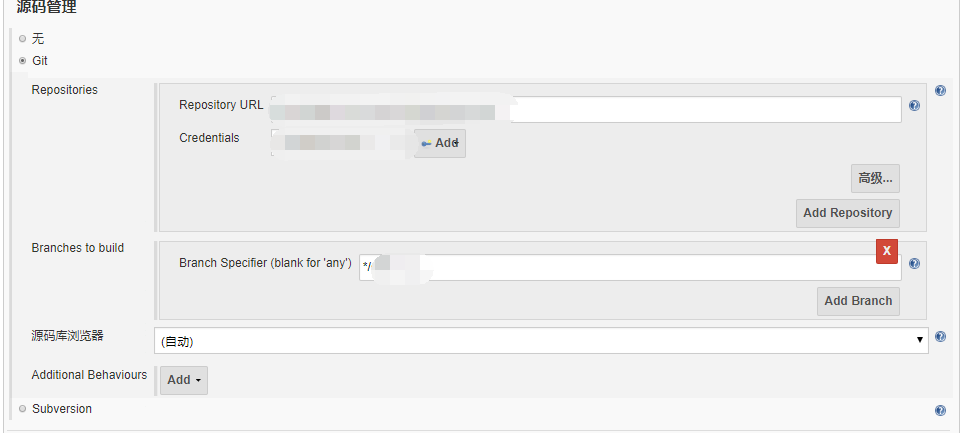


3.       添加版本控制地址，这里以git作为例子，可以执行响应的版本号，这里的版本号也可以通过参数来指定。





3.       添加版本控制地址，这里以git作为例子，可以执行响应的版本号，这里的版本号也可以通过参数来指定。



ansible\_install=/opt/ansible ##之前写好的ansible脚本

FOLDNAME=user

JARFILE=user.jar

SERVERNAME=user.service

cd ${ansible\_install}/roles/project/vars

rm -rf main.yml

echo FOLDNAME: ${FOLDNAME} >> main.yml

echo JARFILE: ${JARFILE} >> main.yml

echo SERVERNAME: ${SERVERNAME} >> main.yml

cd ${ansible\_install}

rm -rf hosts

echo '' >> hosts

echo '[project]' >> hosts

echo ${server\_ip} ansible\_ssh\_port=${server\_port} >> hosts

cd ${ansible\_install}

cd roles/project/files/

rm -rf ./\*.jar

cp  ${WORKSPACE}/user/target/${JARFILE} ./

expect -c "

set timeout 300

    spawn ssh-copy-id -i /root/.ssh/id\_rsa.pub -p ${server\_port} root@${server\_ip}

    expect {

     \"(yes/no)?\" {send \"yes\r\";exp\_continue}

        \"\*password:\" {send \"$server\_pass\r\"}

        \"\*again.\" {send \"$server\_pass\r\"}

    }

    expect eof

"

cd ${ansible\_install}

expect -c "

set timeout 3000

    spawn ansible-playbook -i ${ansible\_install}/hosts user.yml

    expect {

     \"(yes/no)?\" {send \"yes\r\";exp\_continue}

        \"\*password:\" {send \"$server\_pass\r\"}

        \"\*again.\" {send \"$server\_pass\r\"}

    }

    expect eof

"

**总结**

完成自动化脚本之后，在开发联调时就不用再手动打包部署了，只要在Jenkins中构建下对应Job就可以完成服务自动打包更新，提高了开发人员联调效率。本文是针对mavne项目为例，其他语言（go、PHP、react等）只要事先准备好脚本，同样可以实现自动化更新服务。当然还有缺点，目前还不可以自定化更新数据库、安装服务、多节点服务安装和更新等场景，后续有兴趣的同学可以实现这些场景。