**测试环境Jenkins推送更换为Ansible**

安装

yum install -y ansible

在使用ansible之前需要配置ssh免密钥登陆

生成公钥和私钥ssh-keygen -N

ssh-copy-id ipaddress

ansible.conf解析：

hostfile = /etc/ansible/hosts //hosts文件的位置

library = /usr/share/ansible //ansible默认搜寻模块的位置

pattern = \* //如果没有提供hosts节点,这是playbook要通信的默认主机组.默认值是对所有主机通信

remote\_tmp = $HOME/.ansible/tmp //Ansible通过远程传输模块到远程主机,然后远程执行,执行后在清理现场.在有些场景下,你也许想使用默认路径希望像更换补丁一样使用

forks = 5 //在与主机通信时的默认并行进程数 ，默认是5d

poll\_interval = 15 //当具体的poll interval 没有定义时,多少时间回查一下这些任务的状态, 默认值是5秒

sudo\_user = root //sudo使用的默认用户 ，默认是root

#ask\_sudo\_pass = True //用来控制Ansible playbook 在执行sudo之前是否询问sudo密码.默认为no

#ask\_pass = True //控制Ansible playbook 是否会自动默认弹出密码

transport = smart //通信机制.默认 值为’smart’。如果本地系统支持 ControlPersist技术的话,将会使用(基于OpenSSH)‘ssh’,如果不支持将使用‘paramiko’.其他传输选项‘local’,‘chroot’,’jail’等等

#remote\_port = 22 //远程SSH端口。 默认是22

module\_lang = C //模块和系统之间通信的计算机语言，默认是C语言

#host\_key\_checking = False //检查主机密钥

timeout = 10 //SSH超时时间

#log\_path = /var/log/ansible.log //日志文件存放路径

#module\_name = command //ansible命令执行默认的模块

#private\_key\_file = /path/to/file //私钥文件存储位置

常用模块：

shell模块

默认ansible使用的module 是 command，这个模块并不支持 shell 变量和管道等，若想使用shell 来执行模块，请使用-m 参数指定 shell 模块,但是值得注意的是普通的命令执行模块是通过python的ssh执行。

ansible web -m shell -a ‘echo $PATH’

raw模块

Raw也是命令执行模块，而raw模块则是直接用ssh模块进行执行，通常用在客户机还没有python的环境的时候。

ansible web -m raw -a ‘echo $TERM’

copy模块

实现主控端向目标主机拷贝文件，类似于scp的功能

ansible web -m copy -a "src=/etc/hosts dest=/tmp/hosts"

file模块

file模块称之为文件属性模块，可以做的操作如下：

创建文件

ansible web -m file -a "dest=/tmp/zhao/a.txt state=touch"

更改文件的用户及权限

ansible web -m file -a "dest=/tmp/zhao/b.txt mode=600 owner=deploy group=root"

创建目录

ansible web -m file -a "dest=/test mode=755 owner=deploy group=deploy state=directory"

删除文件或者目录

ansible web -m file -a "dest=/tmp/yong state=absent"

state的其他选项：link(链接)、hard(硬链接)

template模块

template使用了Jinjia2格式作为文件模板，进行文档内变量的替换的模块。他的每次使用都会被ansible标记为changed状态。

stat模块

获取远程文件状态信息，包含atime、ctime、mtime、md5、uid、gid等

ansible web -m stat -a "path=/tmp/zhao/a.txt"

管理软件模块

apt、yum模块分别用于管理ubuntu系列和redhat系列系统软件包（redhat用yum，ubuntu用apt，此处以yum为例。）

安装

ansible web -m yum -a "name=httpd state=present"

安装包一个特定的版本

ansible web -m yum -a "name=nginx-1.6.2 state=present"

指定某个源仓库安装某软件包

ansible web -m yum -a "name=php55w enablerepo= remi state=present"

更新一个软件包是最新版本

ansible web -m yum -a "name=httpd state=latest"

卸载一个软件

ansible web -m yum -a "name=httpd state=absent"

User模块

创建新用户和更改、删除已存在用户非常方便

创建用户并更新密码

ansible all -m user -a "name=test password= $6$YyF5qLN8$edF1l.d/xcd9kv4ZQD/VVq5g2Uavlwo."

密码必须为加密过的字符串

删除用户

ansible all -m user -a "name=test state=absent"

service模块

启动服务

ansible web -m service -a "name=httpd state=started"

重启服务

ansible web -m service -a "name=httpd state=restarted"

关闭服务

ansible web -m service -a "name=httpd state=stopped"

cron模块

在指定节点上定义一个计划任务

ansible db -m cron -a 'name="custom job" minute=\*/3 hour=\* day=\* month=\* weekday=\* job="/usr/sbin/ntpdate 172.16.254.139"'

每隔3分钟到主控端更新一次时间

group模块

创建一个组

ansible all -m group -a 'gid=2017 name=grouptest'

script模块

在指定节点上执行脚本

该脚本是在ansible控制节点上的

ansible 172.25.254.\* -m script -a '/root/a.sh'

ping模块

检查指定节点机器是否还能连通

ansible all -m ping

command模块

默认模块，用于运行系统命令，比如echo hello。不支持shell变量和管道。

ansible all -m command -a 'hostname'

也许shell模块可以替代command模块呢？

get\_url模块

下载

下载百度的

ansible host31 -m get\_url -a "url=http://nginx.org/download/nginx-1.12.1.tar.gz dest=/tmp

synchronize模块

将主控方目录推送到指定节点目录下

ansible web -m synchronize -a 'src=/root/a dest=/tmp/ compress=yes'

1

delete=yes 使两边的内容一样（即以推送方为主）

compress=yes 开启压缩，默认为开启

–exclude=.Git 忽略同步.git结尾的文件

该模板默认使用的是push，即推送。若要实现拉取得功能，则指定mode=pull

将指定节点的目录拉取到控制结点下

ansible 10.1.1.113 -m synchronize -a 'mode=pull src=/tmp/a dest=/root/'

主机列表文件 /etc/ansible/host

Ansible 通过读取主机清单/etc/ansible/hosts连接到多个远程主机上执行任务。

使用[]来对主机进行分类，可按照功能，机房，系统，业务等内容进行分类，方便对同一类型的主机进行批量操作。

例如：

[B52-test-2]

test@172.16.51.52

playbook文件

Playbook可以定制配置，可以按照指定的操作步骤有序执行，支持同步和异步方式（以YAML格式为标准）

playbook文件以 .yml 结尾

- hosts: "{{ tomcat\_host }}"

tasks:

- name: stop tomcat

command: bash /usr/local/bin/tomde {{tomcat\_number}} stop

- name : delete old war

shell: rm -rf /data/tomcat{{tomcat\_number}}/webapps/{{war\_name}}\*

- name: copy

copy: src={{WORKSPACE}}/jjjr2-web-gateway/target/{{war\_name}}.war dest=/data/tomcat{{tomcat\_number}}/webapps/

- name: start tomcat

shell: nohup /usr/local/bin/tomde {{tomcat\_number}} start &

- hosts:host主机 //定义主机

tasks: //定义任务

{{$tomcat\_host}} : 双花括号定义的参数是在ansible命令中传递过来

以上用到了 shell、command、copy模块，相关解析已在上文中

由于ansible 并不知道原先在Jenkins 中定义的变量，所以要在Jenkins命令行中把相关的变量传递到yml文件中，再执行yml文件。

例子：

remote\_dir=/share/Software/服务上架/金佳金融2/测试环境/服务端/$JOB\_NAME/`/usr/bin/date "+%Y-%m/%d"`\_$SVN\_REVISION$svnpath

tomcat\_host="test@172.16.49.102"

war\_name="jjjr2-mobile-gateway"

tomcat\_number="1"

echo "推送到共享目录"

/usr/bin/ssh server246 mkdir -p $remote\_dir

/usr/bin/find $WORKSPACE -name ${war\_name}.war -exec scp {} server246:$remote\_dir/ \;

echo "$remote\_dir"

if [[ $deploy == true ]];then

echo "测试环境部署"

/usr/bin/ansible-playbook /etc/ansible/play.yml --extra-vars "tomcat\_host=$tomcat\_host war\_name=$war\_name tomcat\_number=$tomcat\_number WORKSPACE=/root/.jenkins/workspace/$war\_name"

fi

传递参数有三种方法：

ansible-playbook e33\_var\_in\_command.yml --extra-vars "hosts=web user=root"

还可以用json格式传递参数

ansible-playbook e33\_var\_in\_command.yml --extra-vars "{'hosts':'vm-rhel7-1', 'user':'root'}"

还可以将参数放在文件里面：

ansible-playbook e33\_var\_in\_command.yml --extra-vars "@vars.json"

我们这里最适合的是第一种，因为Jenkins所有打完包生成的地址有时候不是一样的，执行推送的时候会报找不到地址，所以我们选择第一种传递变量的时候可以自由定制。

例

1./root/.jenkins/workspace/jjjr2-mobile-gateway/jjjr2-mobile-gateway/target/jjjr2-mobile-gateway.war

2./root/.jenkins/workspace/jjcm-web/target/jjcm-web.war

yml的 copy模块

copy: src={{WORKSPACE}}/{{war\_name}}/target/{{war\_name}}.war dest=/data/tomcat{{tomcat\_number}}/webapps/

这时候，根据例1，Jenkins命令行中的WORKSPACE可以匹配到.war包

例2就不能了，会报“"Unable to find '/root/.jenkins/workspace/jjcm-web/jjcm-web/target/jjcm-web.war' in expected paths."，这时候就是找不到目录下的war包，路径有问题，多了一层jjcm-web目录，所以Jenkins命令行中WORKSPACE变量应该改成WORKSPACE=/root/.jenkins/workspace