Ansible命令使用

Ansible语法使用ansible <pattern\_goes\_here> -m <module\_name> -a <arguments>

-m NAME,–module-name=NAME：指定执行使用的模块

-u USERNAME,–user=USERNAME：指定远程主机以USERNAME运行命令

-s,–sudo：相当于Linux系统下的sudo命令

-USUDO\_USERNAME,–sudo-user=SUDO\_USERNAME：使用sudo，相当于Linux下的sudo命令

-C -check只检查不实际执行

-e EXTRA\_VARS，引用外部参数

-i INVENTORY，指定仓库列表，默认/etc/ansible/hosts

–list-hosts,列出执行主机列

==================================================================================

Ansible的安装部署

Ansible在epel的yum中有提供，所以配置好epel源，直接使用yum命令安装即可

yum install ansible

安装目录

配置文件目录：/etc/ansible/

执行文件目录：/usr/bin/

Lib库依赖目录：/usr/lib/pythonX.X/site-packages/ansible/

Help文档目录：/usr/share/doc/ansible-X.X.X/

Man文档目录：/usr/share/man/man1/

#yum install ansible -y

配置和被管理的主机直接建立基于ssh的密钥认证

[root@Ansible~]#ssh-keygen #生成密码
Generatingpublic/privatersakeypair.
Enterfileinwhichtosavethekey(/root/.ssh/id\_rsa):
Createddirectory'/root/.ssh'.
Enterpassphrase(emptyfornopassphrase):
Entersamepassphraseagain:
Youridentificationhasbeensavedin/root/.ssh/id\_rsa.
Yourpublickeyhasbeensavedin/root/.ssh/id\_rsa.pub.
Thekeyfingerprintis:
2c:b0:df:16:26:8e:c7:e6:b4:c6:6a:22:e1:18:89:e9root@Ansible
Thekey'srandomartimageis:
+--[RSA2048]----+
||
||
|.|
|o.|
|.o.oS|
|\*==.|
|+o..Bo|
|oE..=oo|
|.o.oo|
+-----------------+

==========================================================================================

添加认证

[root@Ansible~]#ssh-copy-id root@172.16.250.149
ssh-copy-id root@172.16.252.245
ssh-copy-id root@172.16.251.163
ssh-copy-id root@172.16.250.217
[root@Ansible~]#sshroot@172.16.250.149 #验证
sshroot@172.16.252.245
sshroot@172.16.251.163
sshroot@172.16.250.217

==========================================================================================

定义主机，将所有被管理的主机加入到/etc/ansible/hosts中，否则无法管理

[root@Ansible~]#vim /etc/ansible/hosts
[web]
172.16.250.149
172.16.252.245
172.16.251.163
172.16.250.217

==========================================================================================

#执行ping存活检测

[root@Ansible~]#ansible web -m ping
172.16.250.217|SUCCESS=>{
"changed":false,
"ping":"pong"
}
172.16.251.163|SUCCESS=>{
"changed":false,
"ping":"pong"
}
172.16.250.149|SUCCESS=>{
"changed":false,
"ping":"pong"
}
172.16.252.245|SUCCESS=>{
"changed":false,
"ping":"pong"
}

#列出执行主机列表

[root@Ansible /etc/ansible]#ansible web --list-hosts
hosts (4):
172.16.250.149
172.16.252.245
172.16.251.163
172.16.250.217

==========================================================================================

[root@Ansible /etc/ansible]#vim ansible.cfg
host\_key\_checking = False
[root@Ansible /etc/ansible]#useradd locy #新建用户
[locy@Ansible ~]$ ssh-keygen
[root@node1 ~]#echo "\*\*\*\*\*\*" | passwd --stdin locy
[locy@Ansible ~]$ ssh locy@172.16.252.245
[locy@node1 ~]$
[root@Ansible ~]#su locy
[locy@Ansible ~]$ ssh locy@172.16.252.245
Last login: Sat Jul 8 10:29:22 2017 from 172.16.250.149
[locy@node1 ~]$ logout
Connection to 172.16.252.245 closed.
[locy@Ansible ~]$ ansible 172.16.252.245 -m ping
172.16.252.245 | SUCCESS => {
"changed": false,
"ping": "pong"
}

==========================================================================================

#做免密sodu

[root@node1 ~]#visudo
%wheel ALL=(ALL) NOPASSWD: ALL
[root@node1 ~]#usermod -G wheel locy

==========================================================================================

对172.16.252.245做ping操作，连接用户locy，以sodu方式运行

[locy@Ansible ~]$ ansible 172.16.252.245 -m ping -u locy -b
172.16.252.245 | SUCCESS => {
"changed": false,
"ping": "pong"
}
或使用root方式 （不建议）
[locy@Ansible ~]$ ansible 172.16.252.245 -m ping -u locy -b --become-user=root

==========================================================================================

Ansible常用模块详解

ansible <host-pattern>     [-m module\_name]    [-a args] [options] #ansible命令格式

指定主机组或ip地址        指定调用模块        传递给模块的参数

ansible-doc -l 查看所有模块

ansible-doc command 查看command模块详细信息

ansible-doc -s command 查看command模块详细用法

==========================================================================================

Command

命令模块接受命令名称，后面是空格分隔的列表参数。给定的命令将在所有选定的节点上执行。它不会通过shell进行处理，比如$HOME和操作如”小于”<“,”>”, “|”, “;”,”&”‘ 工作(需要使用(shell)模块实现这些功能)。

action: command

chdir               # 在执行命令之前，先切换到该目录

creates           # 一个文件名，当这个文件存在，则该命令不执行,可以用来做判断

executable      # 切换shell来执行命令，需要使用命令的绝对路径

free\_form       # 要执行的Linux指令，一般使用Ansible的-a参数代替。

removes         # 一个文件名，这个文件不存在，则该命令不执行,与creates相反的判断

==========================================================================================

对所有机器使用pwd命令

#-m 指定使用的模块command   -a 传递给模块的参数

[root@Ansible ~]#ansible web -m command -a 'pwd'
172.16.250.217 | SUCCESS | rc=0 >>
/root
172.16.252.245 | SUCCESS | rc=0 >>
/root
172.16.251.163 | SUCCESS | rc=0 >>
/root
172.16.250.149 | SUCCESS | rc=0 >>
/root

==========================================================================================

查看磁盘使用情况并将内容传输到/tmp/df.txt中

[root@Ansible ~]#ansible web -m shell -a 'df -h > /tmp/df.txt'

对/tmp/df.txt进行查看

[root@Ansible ~]#ansible web -m command -a 'cat /tmp/df.txt'

批量添加用户

[root@Ansible ~]#ansible web -m command -a 'useradd Tom'

==========================================================================================

shell

执行的命令中有管道或者变量，就需要使用shell

action: shell

chdir           # 执行之前，先cd到指定目录在执行命令

creates       # 一个文件名，当这个文件存在，则该命令不执行

executable  # 切换shell来执行命令，需要使用命令的绝对路径

free\_form   # 执行的命令

removes     # 一个文件名，这个文件不存在，则该命令不执行

==========================================================================================

对/tmp/df.txt进行查看

[root@Ansible ~]#ansible web -m shell -a 'cat /tmp/df.txt'

给上步添加的用户设定密码

[root@Ansible ~]#ansible web -m shell -a 'echo rookie | passwd --stdin Tom'

==========================================================================================

copy

复制模块，将文件复制到被管理主机

action: copy

backup           # 创建一个备份文件包括时间戳信息，如果以某种方式重创错了，还可以拿回原始文件

content         # 取代src=，表示直接用此处指定的信息生成为目标文件内容

dest              # 远程节点存放文件的路径，必须是绝对路径

directory\_mode  # 递归复制设置目录权限，默认为系统默认权限

force             # 如果目标主机包含该文件，但内容不同，如果设置为yes，则强制覆盖，如果设置为no，则只有当目标主机的目标位置不存在该文件时，才复制。默认为yes

group            # 复制到远程主机后，指定文件或目录的属组

mode            # 复制到远程主机后，指定文件或目录权限,类似与chmod指明如 0644

owner           # 复制到远程主机后，指定文件或目录属主

src                # 要复制到远程主机的文件在本地的地址，可以是绝对路径，也可以是相对路径。如果路径是一个目录，它将递归复制。在这种情况下，如果路径使用”/”来结尾，则只复制目录里的内容，如果没有使用”/”来结尾，则包含目录在内的整个内容全部复制，类似于rsync。

==========================================================================================

将本地的/etc/fatab文件复制到目标主机的/tmp/ansible.log，属主为roo，属组为locy,权限为640，并备份

[root@Ansible ~]#ansible web -m copy -a 'src=/etc/fstab dest=/tmp/ansible.log owner=root group=locy mode=640 backup=yes'

对上一步的操作结果进行查看

[root@Ansible ~]#ansible web -m shell -a 'ls -l /tmp/ansible.log'
172.16.250.217 | SUCCESS | rc=0 >>
-rw-r-----. 1 root locy 541 7月 9 20:10 /tmp/ansible.log
172.16.250.149 | SUCCESS | rc=0 >>
-rw-r-----. 1 root locy 541 7月 9 20:10 /tmp/ansible.log
172.16.252.245 | SUCCESS | rc=0 >>
-rw-r----- 1 root locy 541 7月 9 08:10 /tmp/ansible.log
172.16.251.163 | SUCCESS | rc=0 >>
-rw-r----- 1 root locy 541 7月 9 20:10 /tmp/ansible.log

==========================================================================================

cron

定时任务模块，设置管理节点生成定时任务

action: cron

backup              # 如果设置，创建一个crontab备份

cron\_file            # 如果指定, 使用这个文件cron.d，而不是单个用户crontab

day                   # 日应该运行的工作( 1-31, \*, \*/2, etc )

hour                 # 小时( 0-23, \*, \*/2, etc )

job                  # 指明运行的命令是什么

minute             # 分钟( 0-59, \*, \*/2, etc )

month              # 月( 1-12, \*, \*/2, etc )

name               # 定时任务描述

reboot              # 任务在重启时运行，不建议使用，建议使用special\_time

special\_time       # 特殊的时间范围，参数：reboot（重启时）,annually（每年）,monthly（每月）,weekly（每周）,daily（每天）,hourly（每小时）

state                # 指定状态，prsent表示添加定时任务，也是默认设置，absent表示删除定时任务

user                 # 以哪个用户的身份执行

weekday           # 周( 0-6 for Sunday-Saturday, \*, etc )

==========================================================================================

每天凌晨三点、四点、五点、六点将磁盘使用情况保存在/tmp/df.log

[root@Ansible ~]#ansible web -m cron -a 'name="harddrive check" minute="15" hour="3,4,5,6" job="df -lh >> /tmp/df.log"'

每十分钟将磁盘使用情况保存在/tmp/df.log

[root@Ansible ~]#ansible web -m cron -a 'name="harddrive check2" minute="\*/10" job="df -lh >> /tmp/df.log"'
[root@Ansible ~]#crontab -l
#Ansible: harddrive check
15 3,4,5,6 \* \* \* df -lh >> /tmp/df.log
#Ansible: harddrive check2
\*/10 \* \* \* \* df -lh >> /tmp/df.log

将harddrive check删除

[root@Ansible ~]#ansible web -m cron -a 'name="harddrive check" state=absent'

======================================================================================

fetch

远程文件复制到本地

dest                           #保存文件的目录

fail\_on\_missing             #当设置为yes时，如果源文件丢失，任务将会失败

flat                            #允许覆盖将主机名/路径/文件/文件附加到目的地的默认行为

src                            #获取远程系统上的文件。这必须是一个文件，而不是一个文件目录

validate\_checksum      #在获取文件之后验证源和目标校验和

==========================================================================================

将远程文件/tmp/df.txt复制到本地/root/下

[root@Ansible ~]#ansible web -m fetch -a 'src=/tmp/df.txt dest=/root/'

==========================================================================================

file

文件操作模块，设置文件属性

action: file

force          # 需要在两种情况下强制创建软连接，一种是源文件不存在但之后会建立的情况下；另一种是目标连接已存在，需要先取消之前的软连接，有两个选项：yes|no

group          # 设置文件或目录的属组

mode          # 设置文件或目录的权限

owner         # 设置文件或目录的属主

path           # 必选项，定义文件或目录的路径

recurse       # 递归设置文件的属性，只对目录有效

src             # 要被链接到的路径，只应用与state=link的情况

state          # directory：如果目录不存在，创建目录

==========================================================================================

查看web组下的所有主机的/tmp/df.txt

[root@Ansible ~]#ansible web -m shell -a 'ls -l /tmp/df.txt'
172.16.250.217 | SUCCESS | rc=0 >>
-rw-r--r--. 1 root root 562 7月 9 19:18 /tmp/df.txt
172.16.250.149 | SUCCESS | rc=0 >>
-rw-r--r--. 1 root root 535 7月 9 19:18 /tmp/df.txt
172.16.251.163 | SUCCESS | rc=0 >>
-rw-r--r-- 1 root root 615 7月 9 19:18 /tmp/df.txt
172.16.252.245 | SUCCESS | rc=0 >>
-rw-r--r-- 1 root root 535 7月 9 07:18 /tmp/df.txt

将web组下的所有主机的/tmp/df.txt权限改为600属主属组为locy

[root@Ansible ~]#ansible web -m file -a 'path=/tmp/df.txt state=touch mode="600" owner=locy group=locy'
172.16.250.217 | SUCCESS | rc=0 >>
-rw-------. 1 locy locy 562 7月 9 21:41 /tmp/df.txt
172.16.250.149 | SUCCESS | rc=0 >>
-rw-------. 1 locy locy 535 7月 9 21:41 /tmp/df.txt
172.16.252.245 | SUCCESS | rc=0 >>
-rw------- 1 locy locy 535 7月 9 09:41 /tmp/df.txt
172.16.251.163 | SUCCESS | rc=0 >>
-rw------- 1 locy locy 615 7月 9 21:41 /tmp/df.txt

==========================================================================================

在root下创建file目录

[root@Ansible ~]#ansible web -m file -a 'path=/root/file state=directory'
[root@Ansible ~]#ls
file

==========================================================================================

hostname

设置系统的主机名

将172.16.250.149主机名改为master

[root@Ansible ~]#ansible 172.16.250.149 -m hostname -a 'name=master'
[root@Ansible ~]#hostname
master

==========================================================================================

yum

基于yum源安装程序

action: yum

conf\_file                # yum的配置文件

disable\_gpg\_check  # 关闭gpg\_check

disablerepo            # 不启用某个源

enablerepo            # 启用某个源

name=                 # 指定要安装的包，如果有多个版本需要指定版本，否则安装最新的包

state                    # 安装(present)，安装最新版(latest)，卸载程序包(absent)

==========================================================================================

为web组所有主机安装nginx 且为最新版本

[root@Ansible ~]#ansible web -m yum -a 'name=nginx state=latest'

==========================================================================================

service

服务管理模块

action: service

arguments     # 向服务传递的命令行参数

enabled       # 设置服务开机自动启动，参数为yes|no

name          # 控制服务的名称

pattern       # 定义一个模式，如果通过status指令来查看服务的状态时，没有响应，就会通过ps指令在进程中根据该模式进行查找，如果匹配到，则认为该服务依然在运行

runlevel      # 设置服务自启动级别

sleep         # 如果执行了restarted，则在stop和start之间沉睡几秒钟

state         # 启动started  关闭stopped  重新启动restarted  重载reloaded

==========================================================================================

web组所有主机启动nginx

[root@Ansible ~]#ansible web -m service -a 'name=nginx state=started'

web组所有主机关闭nginx

[root@Ansible ~]#ansible web -m service -a 'name=nginx state=stopped'

web组所有主机重启nginx

[root@Ansible ~]#ansible web -m service -a 'name=nginx state=restarted'

web组所有主机重载nginx配置文件

[root@Ansible ~]#ansible web -m service -a 'name=nginx state=reloaded'

web组所有主机启动nginx，并开机启动/不启动

[root@Ansible ~]#ansible web -m service -a 'name=nginx state=started enabled=yes/no'

==========================================================================================

group

用户组模块,添加或删除组

action: group

gid           # 设置组的GID号

name=     # 管理组的名称

state        # 指定组状态，默认为创建，设置值为absent为删除

system     # 设置值为yes，表示为创建系统组

==========================================================================================

创建名为tom的组

[root@Ansible ~]#ansible web -m group -a 'name=tom state=present'

==========================================================================================

user

用户模块,管理用户帐号

action: user

comment           # 用户的描述信息

createhome       # 是否创建家目录

force                 # 在使用state=absent是, 行为与userdel –force一致.

group               # 指定基本组

groups              # 指定附加组，如果指定为(groups=)表示删除所有组

home               # 指定用户家目录

login\_class        # 可以设置用户的登录类 FreeBSD, OpenBSD and NetBSD系统.

move\_home       # 如果设置为home=时, 试图将用户主目录移动到指定的目录

name                # 指定用户名

non\_unique       # 该选项允许改变非唯一的用户ID值

password         # 指定用户密码

remove           # 在使用state=absent时, 行为是与userdel –remove一致

shell               # 指定默认shell

state            # 设置帐号状态，不指定为创建，指定值为absent表示删除

system           # 当创建一个用户，设置这个用户是系统用户。这个设置不能更改现有用户

uid              # 指定用户的uid

update\_password  # 更新用户密码

==========================================================================================

创建用户tom，用户信息为tom is tom，uid为1066，基本组为tom，附加组为wheel，shell类型为zshell，用户家目录为/home/tomhome

[root@Ansible ~]#ansible web -m user -a 'name=tom comment="tom is tom" uid=1066 group=tom groups=wheel shell=/bin/zshell home=/home/tomhome'
[root@Ansible ~]#getent passwd tom
tom:x:1066:1002:tom is tom:/home/tomhome:/bin/zshell

==========================================================================================

script

在指定节点运行服务端的脚本

[root@Ansible ~]#vim test.sh
#/bin/bash
touch /tmp/test.sh.log #创建/tmp/test.sh.log
echo "hello" >> /tmp/test.sh.log #将date命令结果输出到/tmp/test.sh.log
在web组中所有主机执行/root/test.sh脚本
[root@Ansible ~]#ansible web -m script -a '/root/test.sh'
[root@Ansible ~]#cat /tmp/test.sh.log
hello
[root@node1 ~]#cat /tmp/test.sh.log
hello
查看172.16.251.163主机下的/tmp/test.sh.log
[root@Ansible ~]#ansible 172.16.251.163 -m shell -a ‘cat /tmp/test.sh.log’
172.16.251.163 | SUCCESS | rc=0 >>
hello