ansible模块总结

一、yum_repository模块

使用yum_repository管理器来管理软件包，其选项有：
file：配置文件的名字，不包含.repo，会自动加上.repo
name：描述名(yum仓库的名字)
state：状态（present安装，absent卸载，latest最新版本）
description：仓库的描述信息
baseurl:yum源地址
enabled：是否开启yum仓库，yes/no
gpgcheck：是否检查软件包的完整性，yes/no
gpgkey：公钥地址

使用示例：
1、使用软件包完整性检查

```
ansible node1 -m yum_repository -a "file=dvd name=AppStream
description=AppStream baseurl=file:///iso/AppStream enabled=yes gpgcheck=no
gpgkey=/etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release"

ansible node1 -m yum_repository -a "file=dvd name=BaseOS description=BaseOS
baseurl=file:///iso/BaseOS enabled=yes gpgcheck=no gpgkey=/etc/pki/rpm-gpg/RPM-
GPG-KEY-redhat-release"
```

2、不使用软件包完整性检查

```
ansible agent1 -m yum_repository -a 'file=dvd2 name=BaseOS description=BaseOS
baseurl=http://mirror.hust.edu.cn/centos/8.2.2004/BaseOS/x86_64/os/ enabled=yes
gpgcheck=no'

ansible agent1 -m yum_repository -a 'file=dvd2 name=AppStream
description=AppStream
baseurl=http://mirror.hust.edu.cn/centos/8.2.2004/AppStream/x86_64/os/
enabled=yes gpgcheck=no'

ansible agent1 -m yum_repository -a 'file=dvd2 name=epel description=epel
baseurl=http://mirror.hust.edu.cn/epel/8/Everything/x86_64/  enabled=yes
gpgcheck=no'·
```

二、yum模块

name：软件包的名字
state：状态（present安装，absent卸载，latest最新版本）

使用示例

```
ansible test -m yum -a 'name=httpd state=latest'
ansible test -m yum -a 'name=httpd state=absent'
ansible test -m yum -a 'name="@Development Tools" state=present' //装包组
```

三、service模块

name：服务的名字
state：重新加载reloaded，重启restarted，启动started，关闭stopped
enabled：是否开机自启yes/no

使用示例：
```
ansible test -m service -a "name=httpd state=started enabled=yes"
```

四、systemd模块

用于管理服务，该模块包含如下选项：
name：指定服务名称
state：管理服务状态，reloaded|restarted|started|stopped
enabled：是否设置开机自启
daemon_reload：当服务配置文件发生变更时重载服务

使用示例：
```
ansible test -m systemd -a "name=httpd state=started enabled=yes
daemon_reload=yes"
```

五、fetch模块

src：被控端的路径
dest：复制到主控端的路径
flat：默认no，会以被控端主机名的方式组织目录结构

六、copy模块

复制文件到远程主机，copy模块包含如下选项：
*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。
*dest：必选项。要将源文件复制到的远程主机的绝对路径。
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
*backup：在覆盖之前将原文件备份，备份文件包含时间信息。有两个选项：yes|no
*content：用于替代“src”，可以直接设定指定文件的值
*others：所有的file模块里的相关文件属性选项都可以在这里使用
*remote_src：复制远程主机的文件，取值为yes|no，默认是no

使用示例：
```
ansible node1 -m copy -a "src=/tmp/copy.txt dest=/tmp/ansible/copy/"
```

七、file模块

file模块主要用于远程主机上的文件操作，file模块包含如下选项：
path：必选项，定义文件/目录的路径
state：
        file：查看文件状态，默认选项，若文件不存在，也不会被创建，会报错文件不存在
        touch：如果文件不存在，则会创建一个新的文件，如果文件或目录已存在，则更新时间戳
        absent：删除目录、文件或者取消链接文件
        directory：如果目录不存在，创建目录
        link：创建软链接
        hard：创建硬链接

owner：定义文件/目录的属主
group：定义文件/目录的属组
mode：定义文件/目录的权限
src：要被链接的源文件的路径，只应用于state=link的情况
dest：被链接到的路径，只应用于state=link的情况

使用示例
```
ansible agent -m file -a 'path=/tmp/test.txt state=file'
ansible agent -m file -a 'path=/tmp/test.txt state=touch owner=root group=root mode=644'
ansible agent -m file -a 'src=/tmp/test.txt dest=/root/test.txt state=link'
ansible agent -m file -a 'path=/tmp/test state=directory owner=root group=root mode=755'
ansible agent -m file -a 'path=/tmp/test2/test3/aaa/bbb state=directory owner=root group=root mode=755'
ansible agent  -m file -a 'path=/tmp/test2 state=absent'
```

八、script模块

```
# 要执行的脚本文件script.sh内容如下：
#/bin/bash
ifconfig
df -hT

使用示例：
ansible 10.212.52.252 -m script -a 'script.sh'
说明： script模块的脚本文件在主控端，而不需要传到被控端。
```

九、cron模块

```
用于管理计划任务：
name：该任务的描述
state：present创建，absent删除
user：以哪个用户的身份执行
day：日
hour：小时
minute：分钟
month：月
weekday：周
job：要执行的任务
cron_file：指定配置文件，如/etc/crontab，/etc/cron.d/*

使用示例：
ansible node1 -m cron -a "name=hello state=present user=root minute=0 hour=15 job='echo hello cron' cron_file=/etc/crontab"

ansible agent1 -m cron -a "name=hello state=present user=root job='echo hello' minute=*/1"
查看
[root@agent1 yum.repos.d]# crontab -l
#Ansible: hello
*/1 * * * * echo hello
```

十、user模块

```
user模块请求的是useradd，userdel，usermod 三个指令：
state：是创建还是删除
name：指定用户名
uid：指定用的uid
group：指定用户属组
groups：指定用户的附加组
comment：定义用户描述信息
create_home：是否创建家目录 yes|no
home：指定用户的家目录，需要与create_home配合使用
shell：指定用户的shell环境
password：指定用户的密码
remove：当state=absent时，remove=yes则表示连同家目录一起删除，等价于userdel -r


使用示例：
ansible test -m user -a ‘name=johnd comment="John Doe" uid=1040 group=admin’
ansible test -m user -a ‘name=james shell=/bin/bash groups=admins,developers
append=yes ’
ansible test -m user -a  ‘name=johnd state=absent remove=yes’

#需要说明的是，在指定password参数时，不能使用明文密码，因为后面这一串密码会被直接传送到被管理主机的/etc/shadow文件中，所以需要先将密码字符串进行加密处理。然后将得到的字符串放到password中即可。
#使用下面的密码创建用户
# echo "yutian" | openssl passwd -1 -salt $(< /dev/urandom tr -dc '[:alnum:]' |
head -c 32) -stdin
$1$4P4PlFuE$ur9ObJiT5iHNrb9QnjaIB0
ansible all -m user -a 'name=foo password="$1$4P4PlFuE$ur9ObJiT5iHNrb9QnjaIB0"'
```

十一、group模块

```
goup模块请求的是groupadd，groupdel，groupmod 三个指令：
gid：指定组id
name：指定组名
state：创建还是删除组，选项：present|absent

使用示例：
ansible test -m group -a 'name=test gid=1001 state=present'
```

十二、firewalld模块

```
service：放行的服务
port：80/tcp、8080/udp
permanent：yes 永久生效
immediate：yes 立即生效，相当于reload
state：enabled放行、disabled禁止
```

十三、 get_url模块

```
src：被控端的路径
dest：复制到主控端的路径
```

十四、unarchive模块

```
src：压缩文件的原路径
dest：远程主机的路径，即文件解压的路径
remote_src：默认为yes，文件会从master端复制到目标端，否则会直接尝试从目标端查找文件
owner：解压后文件或目录的属主
group：解压后文件或目录的属组
mode：解压后文件的权限
creates：指定一个文件名，当该文件存在时，则解压指令不执行
list_files：默认为no，否则会列出压缩包里的文件

使用示例：
ansible node1 -m unarchive -a "src=/tmp/ac.tar dest=/tmp/tar owner=root
group=root mode=644 list_files=yes"
```

十五、synchronize模块

```
src：要复制的文件
dest：目标路径
archive：归档，默认开启
compress：是否开启压缩
rsync_opts：rsync的选项，选项之间用逗号隔开
delect：删除不存在的文件，默认no
dest_port：默认主机上的端口，默认22
使用实例：
ansible agent1 -m synchronize -a 'src=/etc/passwd dest=/mnt/'
```



十六、selinx

1、sefcontext模块：设置文件的context值

```
target：‘/data(/.*)?’
setype:httpd_git_rw_content_t
state:present
```

2.seboolean模块：设置布尔值

```
name：布尔值
persistent：yes | no
state：开启或关闭 on off
```