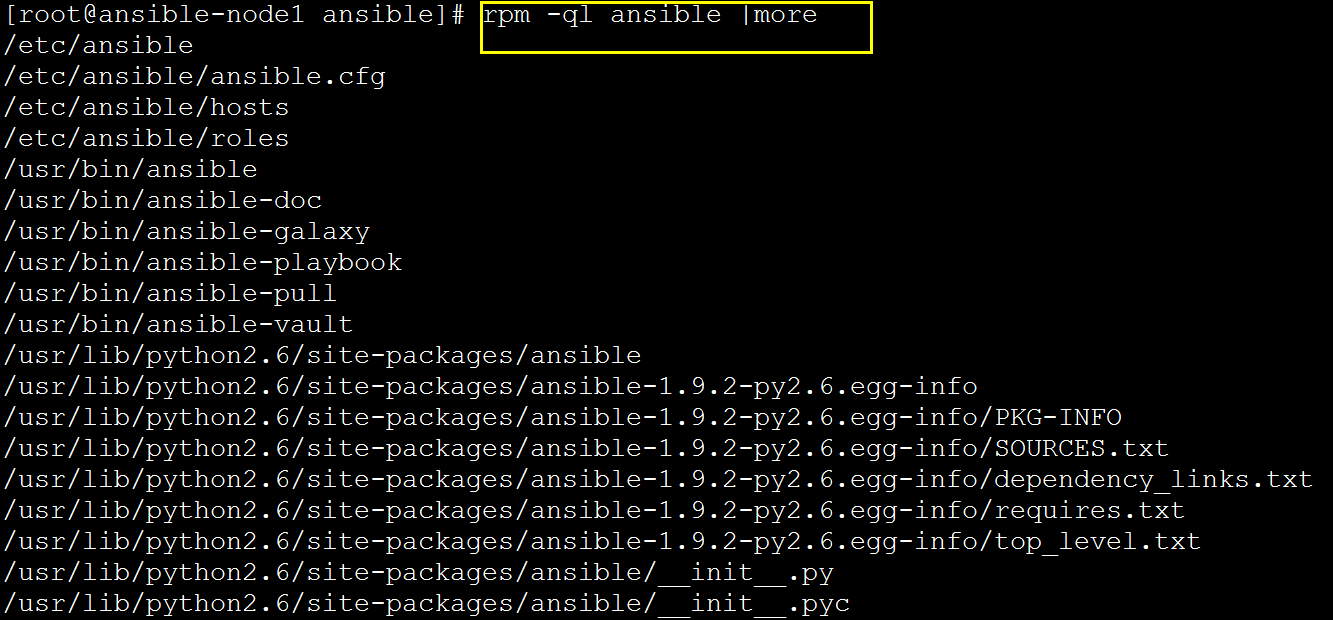


一、安装ansible

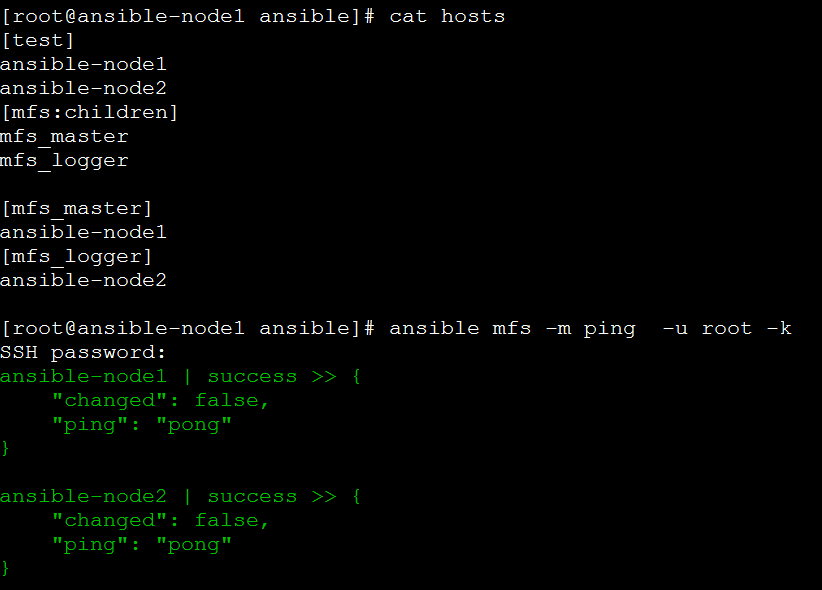
yum install  ansible

2.查看安装后生成的文件：

rpm -ql ansible



2.hosts的组的配置



4.查看ansible的模块

ansible-doc -l

5.查看某个模块下面的参数

ansible-doc -s user

或者ansible-doc user

二、ansible的模块

**1.setup**

从远程节点搜集到的系统信息称为facts。

facts包括远程主机的IP地址，和操作系统类型，磁盘相关信息等等。

可以以一下方式使用上边返回的值：

|  |  |
| --- | --- |
| 1 | {{ ansible\_devices.sda.model }} |

要获取系统的主机名：

|  |  |
| --- | --- |
| 1 | {{ ansible\_hostname }} |

{{ ansible\_default\_ipv4.address }}

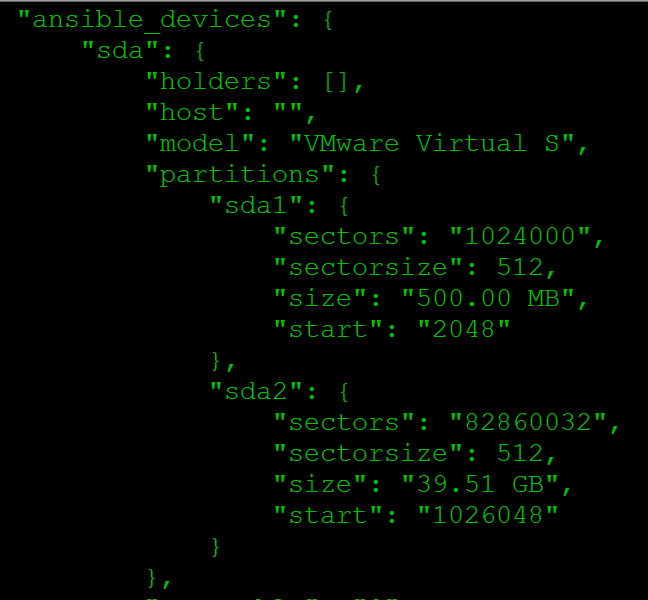
**关闭facts**

如果你确信不需要主机的任何facts信息，而且对远程节点主机都了解的很清楚，那么可以将其关闭。远程操作节点较多的时候，关闭facts会提升ansible的性能。

只需要在play中设置如下：

|  |  |
| --- | --- |
| 1  2 | - hosts: whatever    gather\_facts: no |





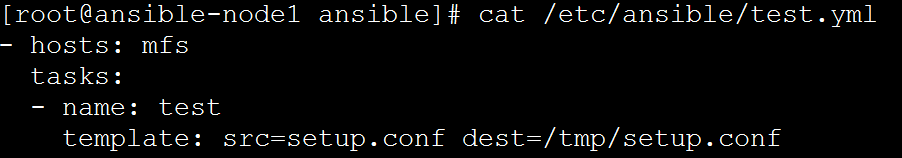
以 [] 中括号括起来的属于列表

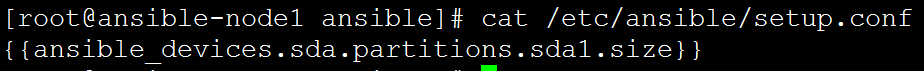
以 {} 大括号括起来的属于元组

"ansible\_devices": { } 属于字典

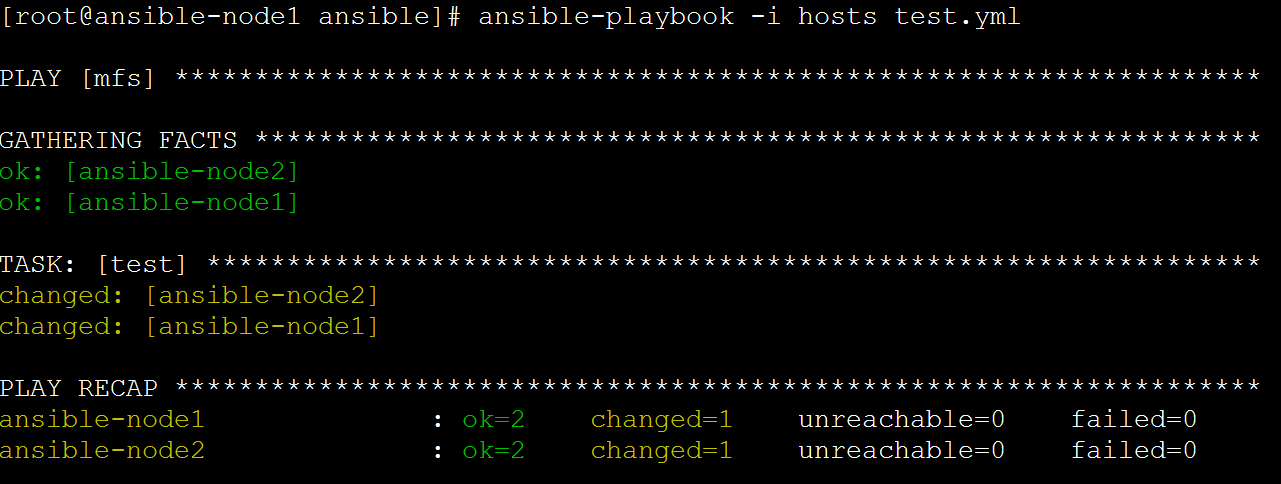
例1：我要获取sda1的硬盘大小：（获取字典中的内容）

{{ansible\_devices.sda.partitions.sda1.size}}

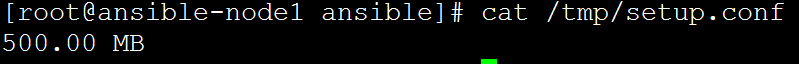




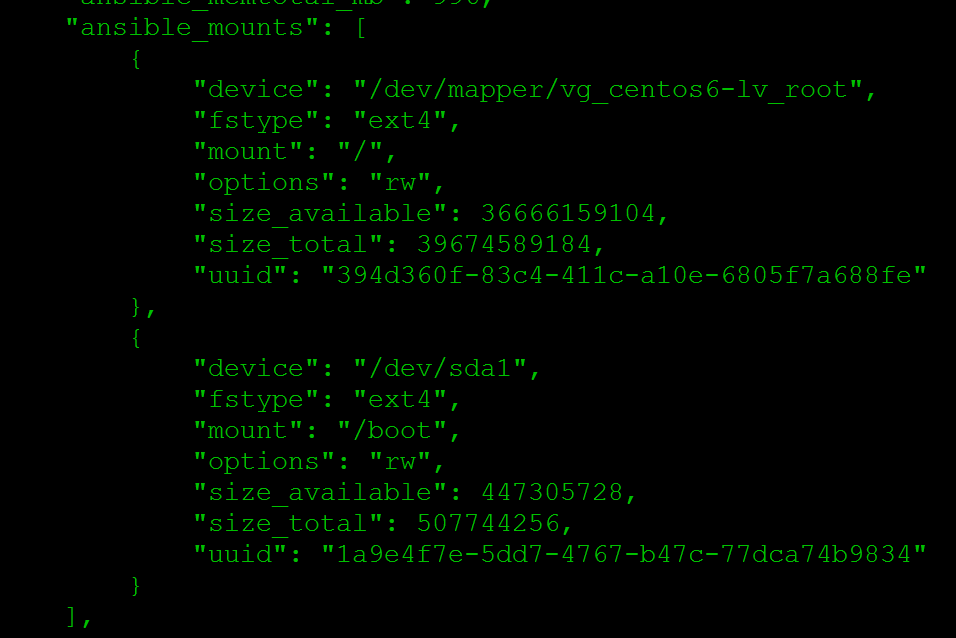
执行：



检查：



例2：获取列表中的内容：

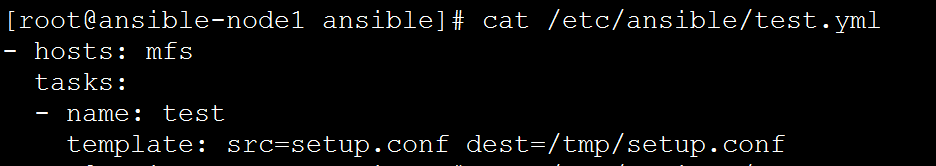


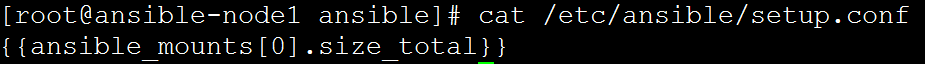
ansible\_mounts [{},{}] 列表中有两个值，每一个{}代表一个值。

获取第一个值中的内容ansible\_mounts[0] ,这样获取的是第一个{}中的整个内容，如果想获取size\_total的值，可以这么写{{ ansible\_mounts[0].size\_total }}

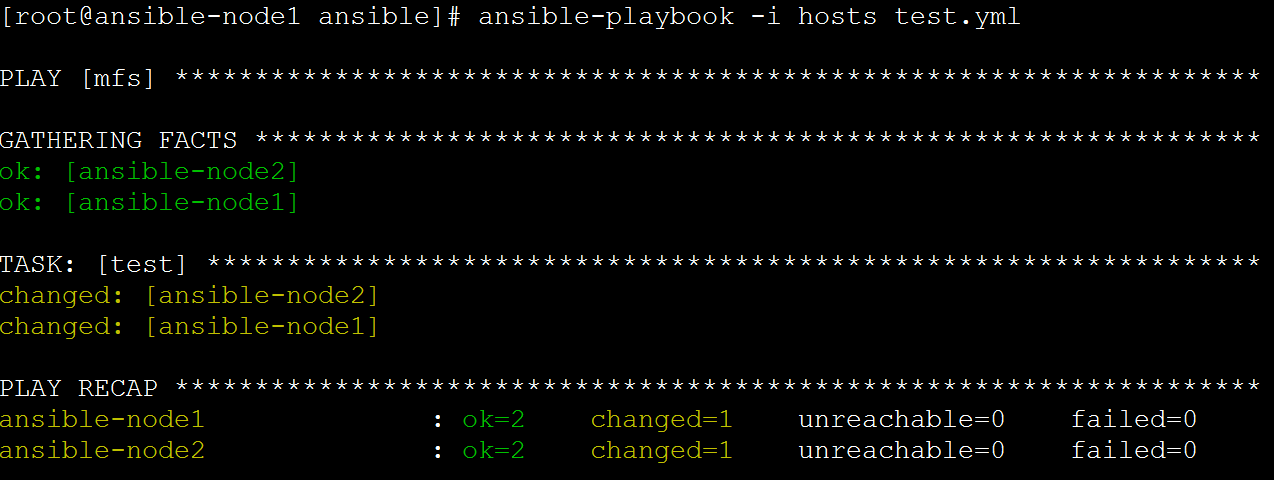
获取第二个值中的内容ansible\_mounts[1]

例如：获取第一个值中的size\_total的值

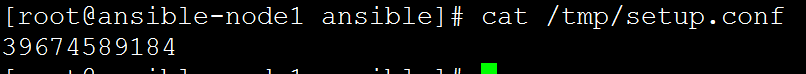




执行：



检查结果：



1.1获取eth0的网卡信息

# ansible ansible-node1 -m setup -a 'filter=ansible\_eth0' -uroot -k

1.2获取eth0的IP信息：

# ansible ansible-node1 -m setup -a 'filter=ansible\_all\_ipv4\_addresses' -uroot -k

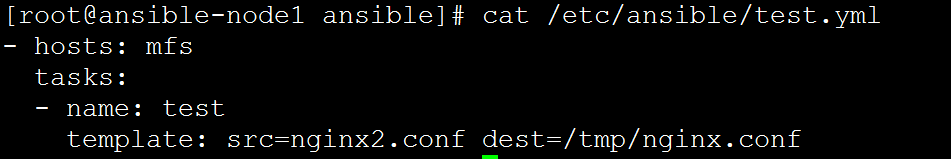
1.3获取ipv4信息：

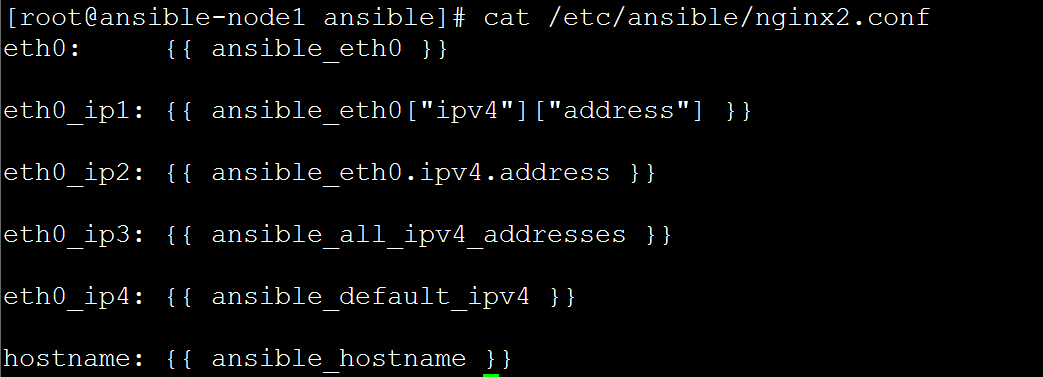
# ansible ansible-node1 -m setup -a 'filter=ansible\_default\_ipv4' -uroot -k

1.4 获取主机名：

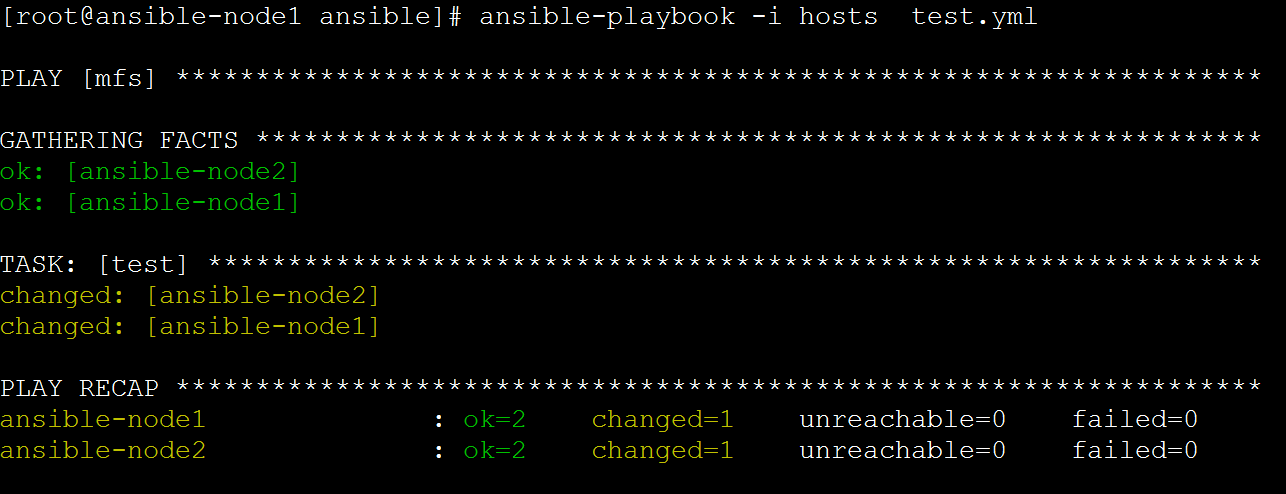
# ansible ansible-node1 -m setup -a 'filter=ansible\_hostname' -uroot -k

使用play-book获取信息

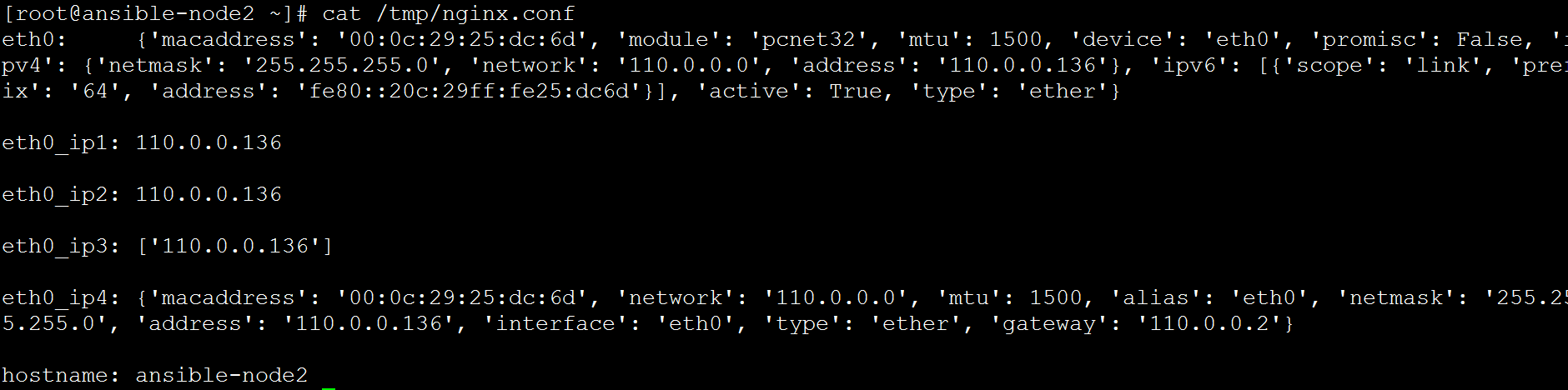


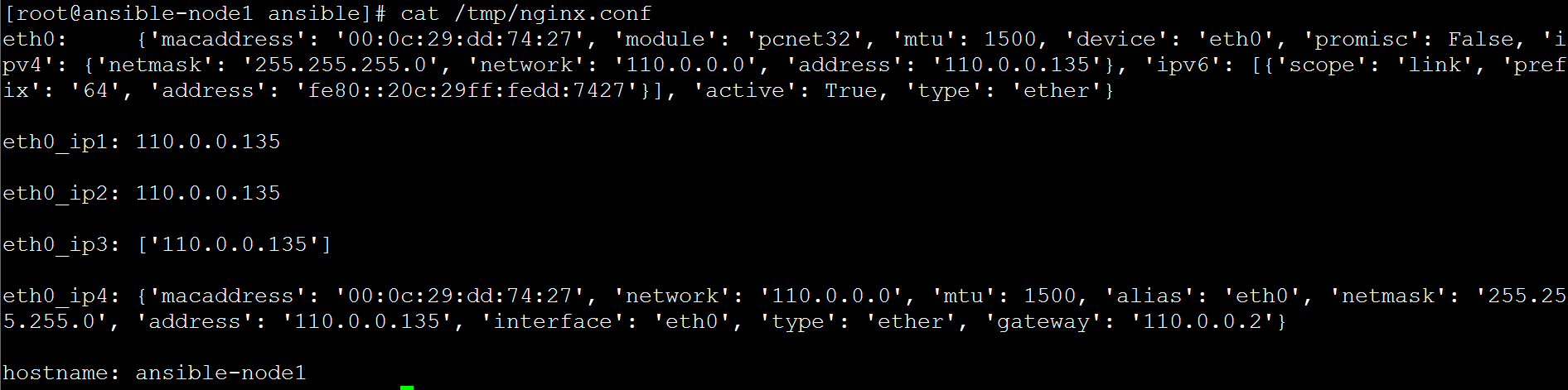


# ansible-playbook -i hosts  test.yml



验证测试：





**2.file**

设置文件的属性

file模块包含如下选项：

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

    group：定义文件/目录的属组

    mode：定义文件/目录的权限

    owner：定义文件/目录的属主

    path：必选项，定义文件/目录的路径

    recurse：递归的设置文件的属性，只对目录有效

    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

    state：

            directory：如果目录不存在，创建目录

            file：即使文件不存在，也不会被创建

            link：创建软链接

            hard：创建硬链接

            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，则更新其最后修改时间

            absent：删除目录、文件或者取消链接文件

示例：

    ansible test -m file -a "src=/etc/fstab dest=/tmp/fstab state=link"

    ansible test -m file -a "path=/tmp/fstab state=absent"

    ansible test -m file -a "path=/tmp/test state=touch"

2.1创建软链接

# ansible ansible-node1 -m file -a "src=/etc/fstab dest=/tmp/fstab state=link"

2.2创建文件

# ansible ansible-node1 -m file -a "path=/tmp/file1 state=touch"

2.3删除文件

# ansible ansible-node1 -m file -a "path=/tmp/file1 state=absent"

2.4创建目录

# ansible ansible-node1 -m file -a "path=/tmp/d1 state=directory"

**3.copy**

复制文件到远程主机

copy模块包含如下选项：

    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

    dest：必选项。要将源文件复制到的远程主机的绝对路径，如果源文件是一个目录，那么该路径也必须是个目录

    directory\_mode：递归的设定目录的权限，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

    others：所有的file模块里的选项都可以在这里使用

    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。

    validate ：The validation command to run before copying into place. The path to the file to validate is passed in via '%s' which must be present as in the visudo example below.

示例：

    ansible test -m copy -a "src=/srv/myfiles/foo.conf dest=/etc/foo.conf owner=foo group=foo mode=0644"

    ansible test -m copy -a "src=/mine/ntp.conf dest=/etc/ntp.conf owner=root group=root mode=644 backup=yes"

    ansible test -m copy -a "src=/mine/sudoers dest=/etc/sudoers validate='visudo -cf %s'"

3.1复制文件并备份

 ansible ansible-node1 -m copy -a "src=/etc/hosts dest=/tmp/file1 backup=yes"

4.command模块

在远程主机上执行命令

command模块包含如下选项：

    creates：一个文件名，当该文件存在，则该命令不执行

    free\_form：要执行的linux指令

    chdir：在执行指令之前，先切换到该指定的目录

    removes：一个文件名，当该文件不存在，则该选项不执行

    executable：切换shell来执行指令，该执行路径必须是一个绝对路径

示例：

    ansible test -a "/sbin/reboot"

4.1创建压缩文件

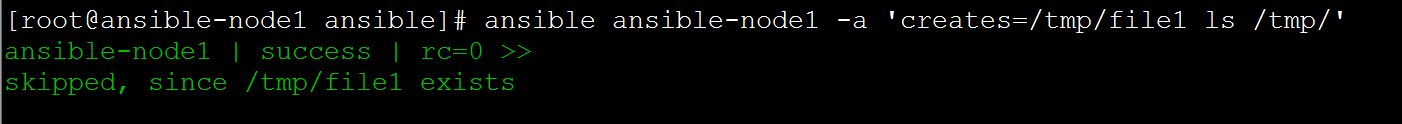
假如/usr/local/src/目录下有个hosts文件，我们需要压缩为aaa.tar.gz

# ansible ansible-node1 -a 'chdir=/usr/local/src  tar zcf aaa.tar.gz  hosts'

4.2creates参数

例如# ansible ansible-node1 -a 'creates=/tmp/file1 ls /tmp/'

执行ls /tmp/之前，先判断/tmp/file1 文件是否存在。如果/tmp/file1存在那么不执行ls /tmp/。否则执行。



5.shell

切换到某个shell执行指定的指令，参数与command相同。

示例：

    ansible test -m shell -a "somescript.sh >> somelog.txt"

shell模块与command的区别：shell模块支持管道，command模块不支持管道。

6.service

用于管理服务

该模块包含如下选项：

    arguments：给命令行提供一些选项

    enabled：是否开机启动  yes|no

    name：必选项，服务名称

    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：运行级别

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

    state：对当前服务执行启动，停止、重启、重新加载等操作（started,stopped,restarted,reloaded）

示例：

    ansible test -m service -a "name=httpd state=started enabled=yes"

    ansible test -m service -a "name=foo pattern=/usr/bin/foo state=started"

    ansible test -m service -a "name=network state=restarted args=eth0"

6.1启动httpd并设置为开机自启动

# ansible ansible-node1 -m service -a 'name=httpd state=started enabled=yes'

6.2停止httpd服务

# ansible ansible-node1 -m service -a 'name=httpd state=stopped'

6.3重启httpd服务，重启过程暂停3s

# ansible ansible-node1 -m service -a 'name=httpd state=restarted sleep=3'

7.cron

用于管理计划任务

包含如下选项：

    backup：对远程主机上的原任务计划内容修改之前做备份

    cron\_file：如果指定该选项，则用该文件替换远程主机上的cron.d目录下的用户的任务计划

    day：日（1-31，\*，\*/2,……）

    hour：小时（0-23，\*，\*/2，……）

    minute：分钟（0-59，\*，\*/2，……）

    month：月（1-12，\*，\*/2，……）

    weekday：周（0-7，\*，……）

    job：要执行的任务，依赖于state=present

    name：该任务的描述

    special\_time：指定什么时候执行，参数：reboot,yearly,annually,monthly,weekly,daily,hourly

    state：确认该任务计划是创建还是删除

    user：以哪个用户的身份执行

示例：

    ansible test -m cron -a 'name="check dirs" hour="5,2" job="ls -alh > /dev/null"'

    ansible test -m cron -a 'name="a job for reboot" special\_time=reboot job="/some/job.sh"'

    ansible test -m cron -a 'name="yum autoupdate" weekday="2" minute=0 hour=12 user="root" job="YUMINTERACTIVE=0 /usr/sbin/yum-autoupdate" cron\_file=ansible\_yum-autoupdate'

    ansilbe test -m cron -a 'cron\_file=ansible\_yum-autoupdate state=absent'

7.1创建定时任务

# ansible ansible-node1 -m cron -a 'name="reboot system" hour=2 user=root job="/sbin/reboot"'

7.2删除定时任务

# ansible ansible-node1 -m cron -a 'name="reboot system" hour=2 user=root job="/sbin/reboot" state=absent'

8.yum

使用yum包管理器来管理软件包

选项：

    config\_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

    list

    name：要进行操作的软件包的名字，也可以传递一个url或者一个本地的rpm包的路径

    state：状态（present，absent，latest）

示例：

    ansible test -m yum -a 'name=httpd state=latest'

    ansible test -m yum -a 'name="@Development tools" state=present'

    ansible test -m yum -a 'name=http://nginx.org/packages/centos/6/noarch/RPMS/nginx-release-centos-6-0.el6.ngx.noarch.rpm state=present'

8.1安装httpd

# ansible ansible-node1 -m yum -a 'name=httpd state=installed'

9.user

管理用户

   - name: M a n a g e   u s e r   a c c o u n t s

- name: M a n a g e   u s e r   a c c o u n t s

  action: user  
      append                 # If `yes', will only add groups, not set them to just the list in `groups'.  
      comment                # Optionally sets the description (aka `GECOS') of user account.  
      createhome             # Unless set to `no', a home directory will be made for the user when the account is created or if the home directory does not exist.  
      expires                # An expiry time for the user in epoch, it will be ignored on platforms that do not support this. Currently supported on Linux and FreeBSD.  
      force                  # When used with `state=absent', behavior is as with `userdel --force'.  
      generate\_ssh\_key       # Whether to generate a SSH key for the user in question. This will \*not\* overwrite an existing SSH key.  
      group                  # Optionally sets the user's primary group (takes a group name).  
      groups                 # Puts the user in this comma-delimited list of groups. When set to the empty string ('groups='), the user is removed from all groups except the primary group.  
      home                   # Optionally set the user's home directory.  
      login\_class            # Optionally sets the user's login class for FreeBSD, OpenBSD and NetBSD systems.  
      move\_home              # If set to `yes' when used with `home=', attempt to move the user's home directory to the specified directory if it isn't there already.  
      name=                  # Name of the user to create, remove or modify.  
      non\_unique             # Optionally when used with the -u option, this option allows to change the user ID to a non-unique value.  
      password               # Optionally set the user's password to this crypted value.  See the user example in the github examples directory for what this looks like in a playbook. The `FAQ  <http://docs.ansible.com/faq.htm  
      remove                 # When used with `state=absent', behavior is as with `userdel --remove'.  
      shell                  # Optionally set the user's shell.  
      ssh\_key\_bits           # Optionally specify number of bits in SSH key to create.  
      ssh\_key\_comment        # Optionally define the comment for the SSH key.  
      ssh\_key\_file           # Optionally specify the SSH key filename. If this is a relative filename then it will be relative to the user's home directory.  
      ssh\_key\_passphrase     # Set a passphrase for the SSH key.  If no passphrase is provided, the SSH key will default to having no passphrase.  
      ssh\_key\_type           # Optionally specify the type of SSH key to generate. Available SSH key types will depend on implementation present on target host.  
      state                  # Whether the account should exist or not, taking action if the state is different from what is stated.  
      system                 # When creating an account, setting this to `yes' makes the user a system account.  This setting cannot be changed on existing users.  
      uid                    # Optionally sets the `UID' of the user.  
      update\_password        # `always' will update passwords if they differ.  `on\_create' will only set the password for newly created users.

    需要特别说明的是，password后面指定的密码不能是明文，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，而登陆的时候输入的密码会被hash加密以后再去与/etc/shadow中存放的密码去做对比，会出现不一致的现象。所以需要先将密码字符串进行加密处理：openssl passwd -salt -1 "123456"，然后将得到的字符串放到password中即可。

9.1创建用户

生成加密的密码：

# openssl passwd -salt -1 "123456"

-1yEWqqJQLC66

然后创建用户

# ansible ansible-node1 -m user -a 'createhome=yes home=/home/user1 password=-1yEWqqJQLC66 name=user1 state=present shell=/bin/bash'

92.删除用户

# ansible ansible-node1 -m user -a 'remove=yes state=absent name=user1'

10.synchronize

使用rsync同步文件

    archive

    checksum

    delete

    dest

    src

    dest\_port

    existing\_only: skip createing new files on receiver

    links

    owner

    mode:(push, pull)

    recursive

    rsync\_path

    times:Preserve modification times

示例：

    src=some/relative/path dest=/some/absolute/path rsync\_path="sudo rsync"

    src=some/relative/path dest=/some/absolute/path archive=no links=yes

    src=some/relative/path dest=/some/absolute/path checksum=yes times=no

    src=/tmp/helloworld dest=/var/www/helloword rsync\_opts=--no-motd,--exclude=.git mode=pull

9.1同步文件

# ansible ansible-node1 -m synchronize -a 'src=/tmp/helloworld dest=/var/www/'

9.2 同步目录

# ansible ansible-node1 -m synchronize -a 'src=/etc/ansible dest=/usr/local/src'

9.3

# ansible ansible-node1 -m synchronize -a 'src=/etc/ansible.cfg dest=/tmp/ mode=push rsync\_path=/usr/bin/rsync rsync\_opts="-avz.--exclude=.git"'

10.mount

配置挂载点

选项：

    dump

    fstype：必选项，挂载文件的类型

    name：必选项，挂载点

    opts：传递给mount命令的参数

    passno

    src：必选项，要挂载的文件

    state：必选项

            present：只处理fstab中的配置

            absent：删除挂载点

            mounted：自动创建挂载点并挂载之

            umounted：卸载

示例：

    name=/mnt/dvd src=/dev/sr0 fstype=iso9660 opts=ro state=present

    name=/srv/disk src='LABEL=SOME\_LABEL' state=present

    name=/home src='UUID=b3e48f45-f933-4c8e-a700-22a159ec9077' opts=noatime state=present

    ansible test -a 'dd if=/dev/zero of=/disk.img bs=4k count=1024'

    ansible test -a 'losetup /dev/loop0 /disk.img'

    ansible test -m filesystem 'fstype=ext4 force=yes opts=-F dev=/dev/loop0'

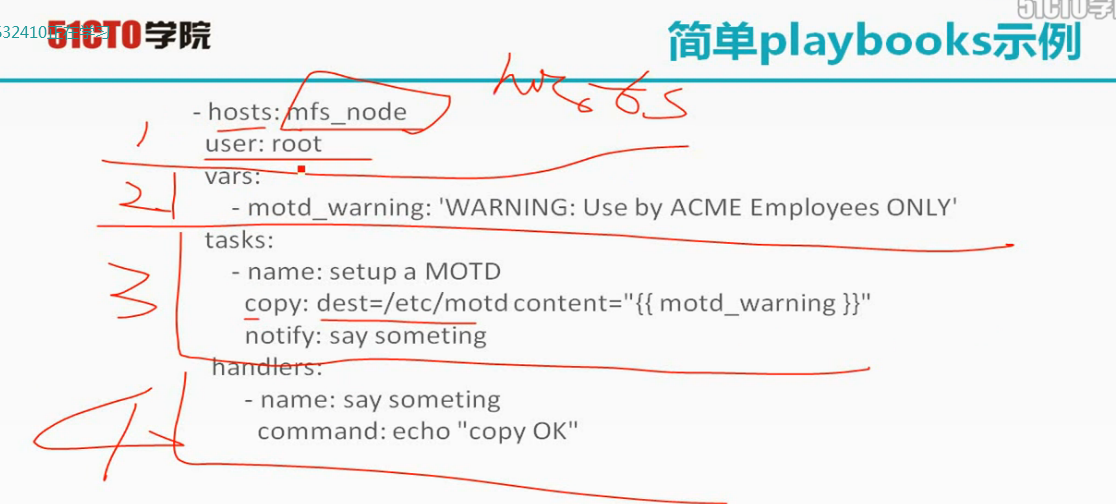
    ansible test -m mount 'name=/mnt src=/dev/loop0 fstype=ext4 state=mounted opts=rw'

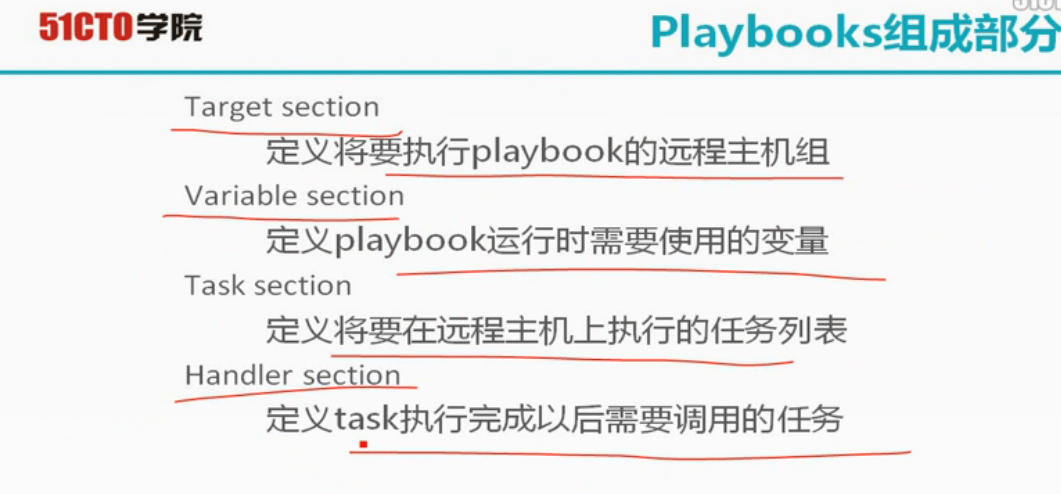
ansible ansible-node1 -m command -a 'dd if=/dev/zero of=/disk.img bs=4k count=1024'

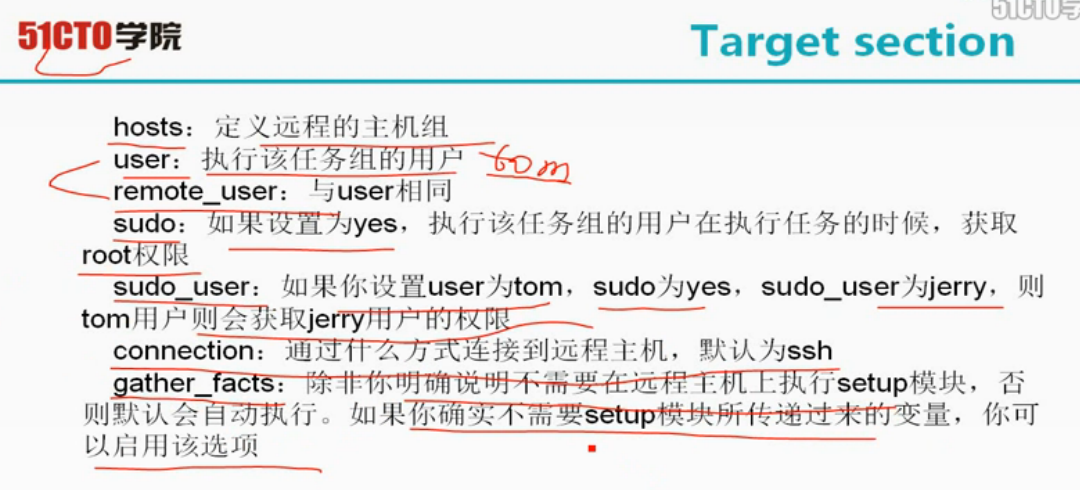
ansible ansible-node1 -m command -a 'losetup /dev/loop0 /disk.img'

ansible ansible-node1 -m filesystem -a 'dev=/disk.img fstype=ext4'

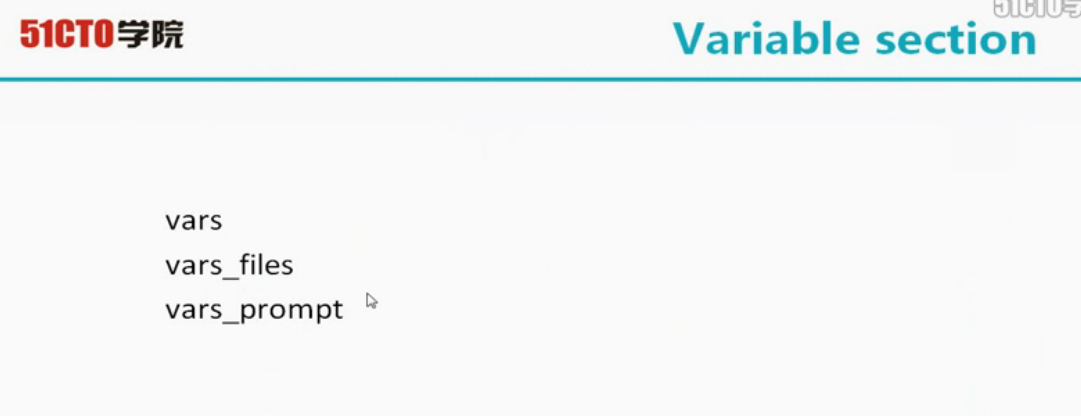
二，playbooks



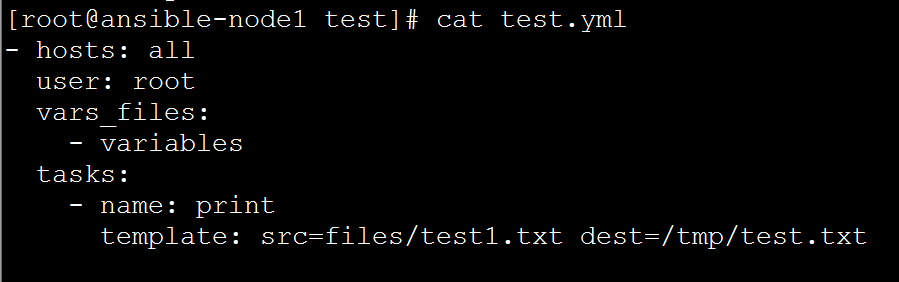


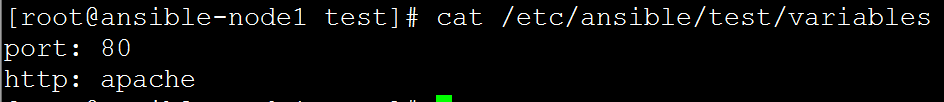


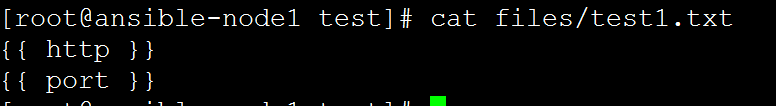
1.vars 变量部分







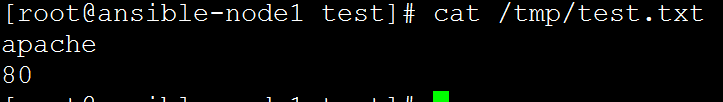




执行：

# ansible-playbook test.yml

测试：



vars\_prompt

交互式输入变量

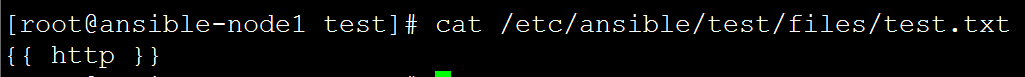
实例1：



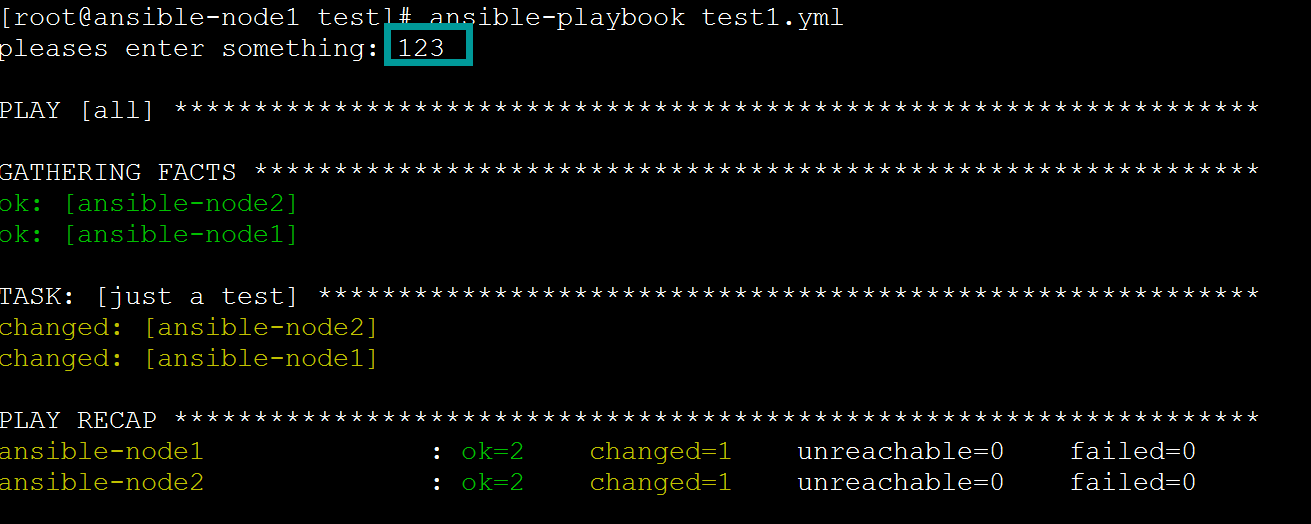
- name: http     #变量名称

   prompt: pleases enter something:  #执行时提示输入值，这个值会赋给上面的name的变量

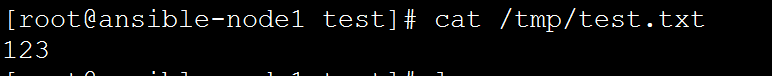
   private: no  # 是否设置为私有，如果设为yes，那么执行时是不显示变量的值，如果设为no，那么执行时会显示变量的值；



执行：



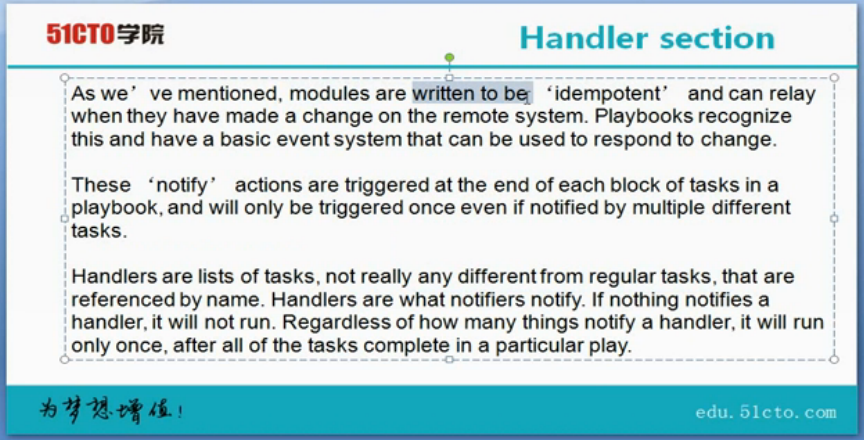
检查结果：



2.Task section 任务部分：

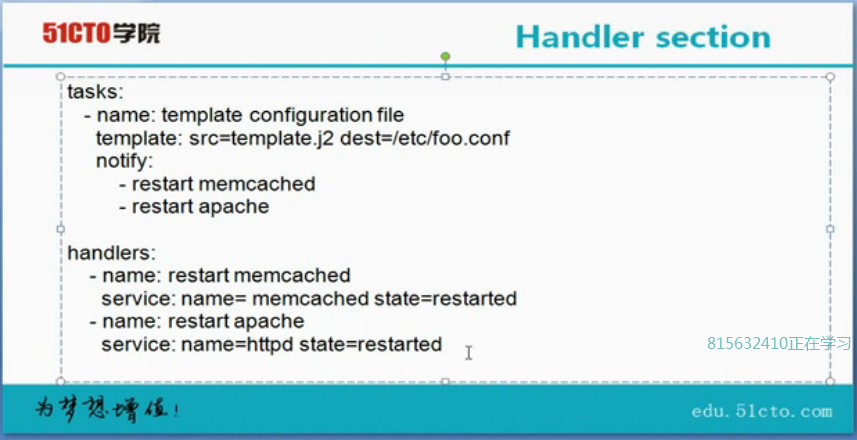


3.Handler section



notify： notify是在playbook种的每一个tasks块的末尾被触发的，不管有多个不同的tasks通知多少次，而且只能触发一次。

handlers：handlers也是任务列表。Handlers 是notify通知的。如果没有通知handler，那么handler将不会执行的。不管多少个任务通知同一个handler，但是，handler只能在所有的tasks执行完成后，被执行一次。



notify 通知handlers:

notify 中写handles中name：后面的值就可以调用handles

notify:

    - restart memcached

    - restart apache

handlers:

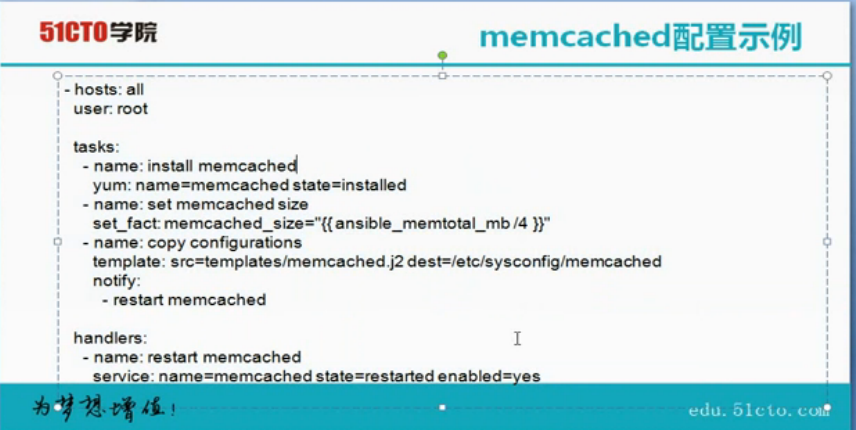
    - name: restart memcached

       service: name=memcached state=restarted

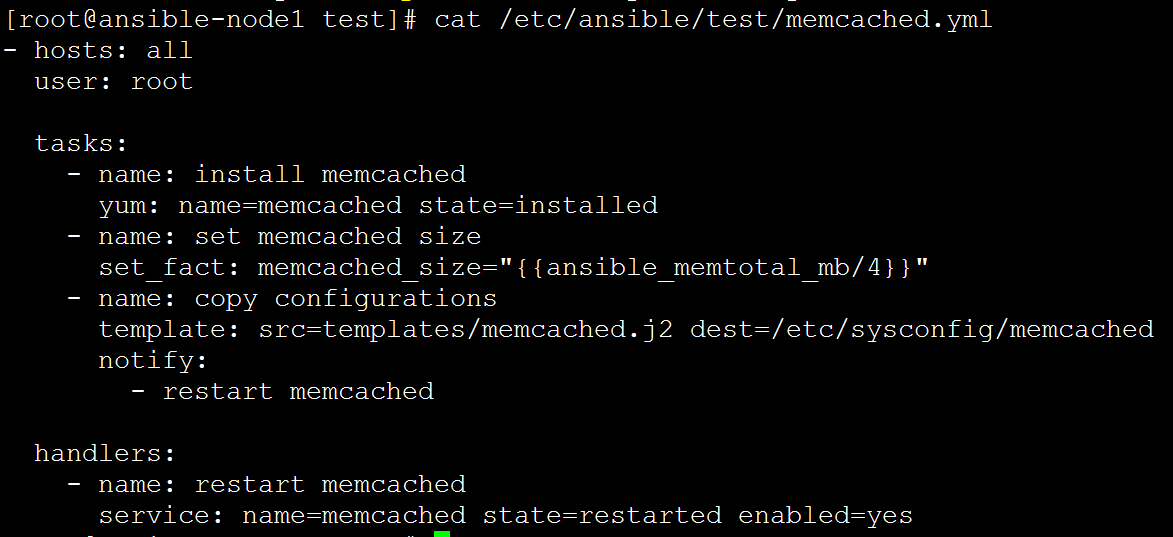
    - name: restart apache

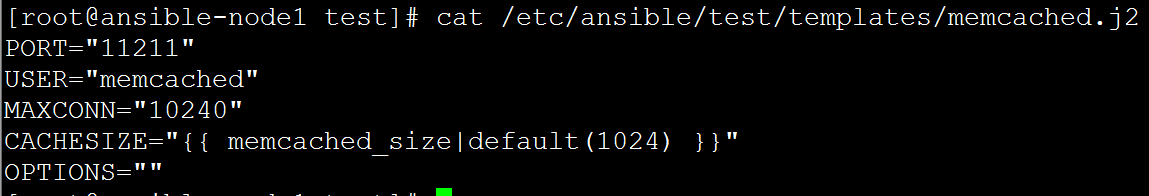
       service: name=httpd state=restarted

3.1 memcached安装配置实例：



set\_fact: 是收集客户端的信息，并进行计算。



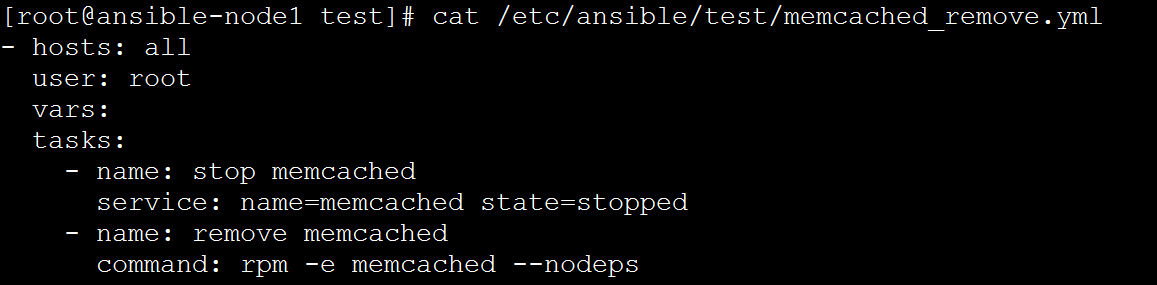


CACHESIZE="{{ memcached\_size|default(1024) }}" 如果memcached\_size的值存在，CACHESIZE的值就等于memcached\_size ,如果memcached\_size的值不存在，CACHESIZE的值，就用default(1024)，也是1024

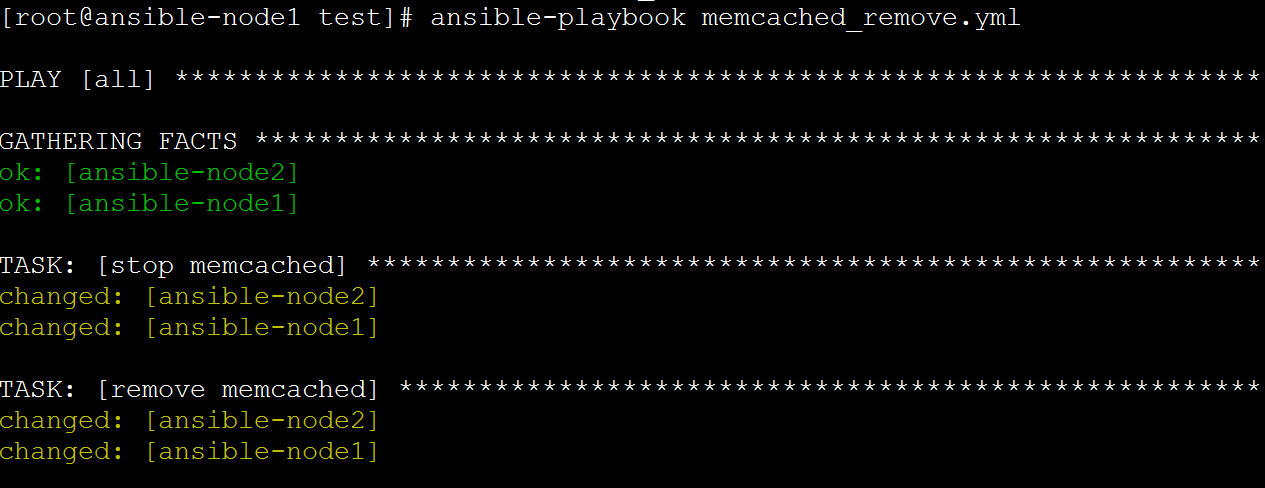
执行：



3.2 memcached的卸载：



执行：



三、playbook 的常用模块





1.template模块

