**自动化运维的精髓所在**

所谓的运维自动化实际上就是某些运维过程的自动化，比如初始化自动化、测试/部署自动化，监控自动化，简单报警处理自动化，业务降级/恢复自动化...，慢慢的让系统可以承担更多的重复劳动，减少人力投入和学习成本。

**企业级自动化运维工具应用实战-ansible**

使用场景：  
公司计划在年底做一次大型市场促销活动，全面冲刺下交易额，为明年的上市做准备。公司要求各业务组对年底大促做准备，运维部要求所有业务容量进行三倍的扩容，并搭建出多套环境可以共开发和测试人员做测试，运维老大为了在年底有所表现，要求运维部门同学尽快实现，当你接到这个任务时，有没有更快的解决方式？  
（要在短时间内快速的扩充业务容量的最快方法就是横向扩展scale out，那么数量很多的机器，如何快速布置呢，一台一台布置效率就太低了，这时候就可以使用运维自动化工具ansible等等，只要管理一台，其他标准化的机器就都OK了）

**目前多数公司遇到的运维问题现状：**

1、硬件选型多样化  
2、系统多版本并存  
3、目录结构混乱  
4、运维人员水平参差不齐  
5、历史遗留问题多  
6、同一软件出现多个版本  
7、无法批量化操作  
8、工作效率低，故障率高  
9、项目上线操作繁琐

**日常运维工作中的重要事件：**

1、添加监控，收集系统指标  
2、对日志进行归档以及管理  
3、数据备份于恢复  
4、对计划任务进行管理  
5、对软件包进行部署和管理  
6、对脚本进行批量执行  
7、对文件进行批量复制和移动  
8、设置文件或者目录对应的权限  
9、关闭和启动服务  
10、对代码程序进行上下线

**ansible是什么**

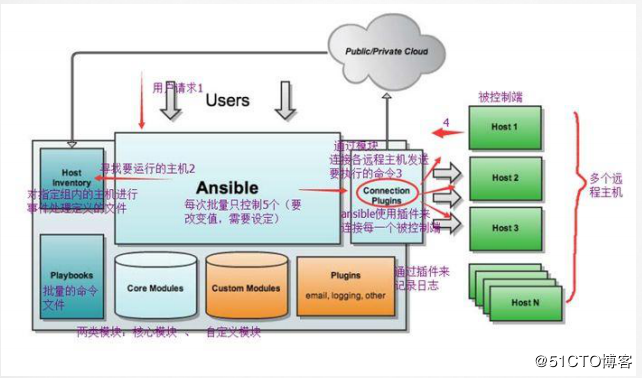
ansible是新出现的自动化运维工具，基于Python开发，集合了众多运维工具（puppet、chef、func、fabric）的优点，实现了批量系统配置、批量程序部署、批量运行命令等功能。

ansible是基于 paramiko 开发的,并且基于**模块化工作**，本身没有批量部署的能力。真正具有批量部署的是ansible所运行的模块，ansible只是提供一种框架。ansible不需要在远程主机上安装client/agents，因为它们是基于**ssh**来和远程主机通讯的。ansible目前已经已经被红帽官方收购，是自动化运维工具中大家认可度最高的，并且上手容易，学习简单。是每位运维工程师必须掌握的技能之一。

**ansible特点**

1、部署简单，只需在主控端部署Ansible环境，被控端无需做任何操作；  
2、默认使用SSH协议对设备进行管理；  
3、有大量常规运维操作模块，可实现日常绝大部分操作。  
4、配置简单、功能强大、扩展性强；  
5、支持API及自定义模块，可通过Python轻松扩展；  
6、通过Playbooks来定制强大的配置、状态管理；  
7、轻量级，无需在客户端安装agent，更新时，只需在操作机上进行一次更新即可；  
8、提供一个功能强大、操作性强的Web管理界面和REST API接口 ——AWX平台。（目前这个应该是付费项目）

**ansible架构图**

  
Ansible：Ansible核心程序。  
**HostInventory**：记录由Ansible管理的主机信息，包括端口、密码、ip等。  
（一般端口都是默认端口不用指定也行，基于ssh私钥登录的话密码也不用了，就指定个ip地址或者主机名就成，主机名要能被解析）  
**Playbooks**：“剧本”YAML格式文件，多个任务定义在一个文件中，定义主机需要调用哪些模块来完成的功能。  
（类似于shell脚本，多条ansible命令的集合，十分注意格式。）  
CoreModules：核心模块，主要操作是通过调用核心模块来完成管理任务。  
（现在已经内嵌了多个模块，已经可以满足我们日常运维的工作了，有特别需要了再自己定义模块）  
CustomModules：自定义模块，完成核心模块无法完成的功能，支持多种语言。  
**ConnectionPlugins**：连接插件，Ansible和Host通信使用

**ansible任务执行模式**

Ansible系统由控制主机对被管节点的操作方式可分为两类，即***ad- hoc***和**playbook**：

·ad-hoc模式使用单个模块，支持批量执行单条命令。 ad\*\*\*\*-hoc 命令是一种可以快速输入的命令，而且不需要保存起来的命令。就相当于bash中的一句话shell。

·playbook模式是Ansible主要管理方式，playbook通过多个task集合完成一类功能，如Web服务的安装部署、数据库服务器的批量备份等。可以简单地把playbook理解为通过组合多条ad-hoc操作作的配置文件。

**Ansible命令执行过程**

1、加载自己的配置文件 默认/etc/ansible/ansible.cfg  
2、查找对应的主机配置文件，找到要执行的主机或者组  
3、加载自己对应的模块文件，如command  
4、通过ansible将模块或命令生成对应的临时py文件，并将该文件传输至远程服务器的对应的目录下  
5、对应执行用户的家目录下的.ansible/tmp/XXX/XXX.PY文件  
6、给文件+x执行  
7、执行并返回结果  
8、删除临时py文件，sleep 0退出

**ansible安装方式**

ansible安装常用两种方式，yum安装和pip程序安装  
这里提供二种安装方式,任选一种即可  
1使用yum安装  
yum install epel-release -y  
yum install ansible –y  
2 使用pip（python的包管理模块）安装  
（与perl的cpanm模式安装类似）  
pip install ansible  
如果没pip,需先安装pip.yum可直接安装：  
yum install python-pip  
pip3 install ansible

**ansible程序结构**

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/

**Ansible配置文件**

设置ansible.cfg配置参数，ansible有许多参数，下面列出常用的参数：

inventory： *#这个参数表示资源清单inventory文件的位置，资源清单就是*

一些Ansible需要连接管理的主 机列表。这个参数的配置实例如下：

inventory = /etc/ansible/hosts 放可管理链接主机的地方

library：这个library参数就是指向存放Ansible模块的目录。配置实例如下：

library = /usr/share/ansible 放ansible平时用的模块的地方（自定义模块）

Ansible支持多个目录方式，只要用冒号（：）隔开就可以，同时也会检查当前执行playbook位置下的./library目录。

forks： 设置默认情况下Ansible最多能有多少个进程同时工作， 从Ansible 1.3开始,

fork数量默认自动设置为主机数量或者潜在的主机数量,默认设置最多5个进程并行处理。

具体需要设置多少个，可以根据控制主机的性能和被管节点的数量来确定，可能是 50或100。默认值5是非常保守的值，

配置实例如下：

forks = 5 ansible能一下管理链接多少台机器，默认5台偏保守

sudo\_user：

这是设置默认执行命令的用户，也可以在playbook中重新设置这个参数。配置实例如下：

sudo\_user = root

remote\_port：

这是指定连接被管节点的管理端口，默认是22。除非设置了特殊的SSH端口，不然这个参数一般是 不需要修改的。

配置实例如下：

remote\_port = 22

host\_key\_checking：

这是设置是否检查SSH主机的密钥。可以设置为True或False，关闭后第一次连接没有提示配置实例

host\_key\_checking = False

timeout：

这是设置SSH连接的超时间隔，单位是秒。配置实例如下：

timeout = 60

log\_path：Ansible系统默认是不记录日志的，如果想把Ansible系统的输出记录到日志文件中，需要设置log\_path

来指定一个存储Ansible日志的文件。配置实例如下：

log\_path = /var/log/ansible.log 建议打开日志

另外需要注意，执行Ansible的用户需要有写入日志的权限，模块将会调用被管节点的syslog来记录

**Ansible主机清单**

可以直接定义一个主机信息，也可以把多个主机放到一个机器组中，起个组名。格式如下（主机匹配还支持统配符来匹配）  
1、直接指明主机地址或主机名：

## **green**.example.com#

# **blue**.example.com#

# 192.168.100.1

# 192.168.100.10

2、定义一个主机组[组名]把地址或主机名加进去

[mysql\_test]

192.168.253.159

192.168.253.160

192.168.253.153

#组成员可以使用通配符来匹配，如下 192.168.2.[1:6]

#表示匹配从192.168.2.1——192.168.2.6的主机

**Ansible常用命令**

/usr/bin/ansible *# Ansibe AD-Hoc 临时命令执行工具，常用于临时命令的执行*

/usr/bin/ansible-doc *# Ansible 模块功能查看工具*

/usr/bin/ansible-galaxy # 下载/上传优秀代码或Roles模块 的官网平台，基于网络的

/usr/bin/ansible-playbook *# Ansible 定制自动化的任务集编排工具*

/usr/bin/ansible-pull *# Ansible远程执行命令的工具，拉取配置而非推送配置（使用较少，海量机器时使用，对运维的架构能力要求较高）*

/usr/bin/ansible-vault *# Ansible 文件加密工具*

/usr/bin/ansible-console *# Ansible基于Linux Consoble界面可与用户交互的命令执行工具*

**Ansible配置公私钥**

配置ansible使用公钥验证  
虽然ansible支持其他主机认证方式，但是我们最常用的的还是基于秘  
钥的认证：  
1、首先生成秘钥  
ssh-keygen -t rsa -P ‘’   
2、然后向主机分发秘钥：  
ssh-copy-id root@ #@后面跟主机名或者IP地址  
3、如果出现以下情况：

**# ssh-copy-id -i ~/.ssh/id\_rsa.pub 10.1.6.72**

-bash: ssh-**copy**-**id**: command not found

请尝试：

yum -y install openssh-clientsansible

**ansible常用模块：**

ping模块：用于测试主机能否连通  
commond模块：用于在主机上执行单条命令  
shell模块：可在管理主机上执行一些复杂命令  
copy模块：大多用于推送配置文件  
yum模块：批量化安装服务包  
setup模块：获取管理的主机的配置信息  
fetch模块：从管理主机上拉取文件  
user模块：批量创建用户  
group模块：批量创建组  
file模块：创建软链接目录什么的  
script模块：在管理机上运行脚本，在被管理机上执行  
service服务模块：服务的启动和重载等等  
cron计划任务模块：添加计划任务

**Ansible playbook简介**

playbook是ansible用于配置，部署，和管理被控节点的剧本。  
通过playbook的详细描述，执行其中的一系列tasks，可以让远端主机达到预期的状态。

**Ansible playbook使用场景**

执行一些简单的任务，使用ad-hoc命令可以方便的解决问题，但是有时一个设施过于复杂，需要大量的操作时候，执行的ad-hoc命令是不适合的，这时最好使用playbook。

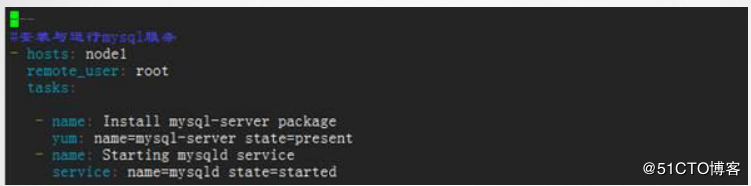
就像执行shell命令与写shell脚本一样，也可以理解为批处理任务，不过playbook有自己的语法格式。

使用playbook你可以方便的重用这些代码，可以移植到不同的机器上面，像函数一样，最大化的利用代码。。

**Ansible playbook格式**

对我们后面书写playbook很有帮助。以下为playbook常用到的YMAL格式。

文件的第一行应该以 ”-” (三个连字符)开始，表明YMAL文件的开始。  
在同一行中，#之后的内容表示注释，类似于shell，python和ruby。  
YMAL中的列表元素以”-”开头然后紧跟着一个空格，后面为元素内容。就像这样

在mysql.yml中，主要由三个部分组成。  
**hosts部分**：使用hosts指示使用哪个主机或主机组来运行下面的tasks，每个playbook都必须指定hosts，   
**remote\_user**：指定远端主机中的哪个用户来登录远端系统，在远端系统执行task的用户，可以任意指定，也可以使用sudo，但是用户必须要有执行相应task的权限。  
**tasks**：指定远端主机将要执行的一系列动作。tasks的核心为ansible的模块，前面已经提到模块的用法。tasks包含name和要执行的模块，name是可选的，只是为了便于用户阅读，不过还是建议加上去，模块是必  
须的，同时也要给予模块相应的参数。  


**Playbook的核心元素**

Hosts：主机  
Tasks：任务列表  
Variables 设置方式有4种  
Templates：包含了模板语法的文本文件；  
Handlers：由特定条件触发的任务；

Playbooks配置文件的基础组件：  
Hosts：运行指定任务的目标主机；  
remoute\_user: 在远程主机上执行任务的用户；  
sudo\_user：  
tasks：任务列表  
模块，模块参数；

        Host Inventory：主机清单，也就是被管理的主机列表

        Playbooks：ansible的剧本，可想象为将多个任务放置在一起，一块执行

        Core Modules：ansible的核心模块

        Custom Modules：自定义模块

        Connection Plugins：连接插件，用于与被管控主机之间基于SSH建立连接关系

        Plugins：其他插件，包括记录日志等

    2、ansible的特性

        <1>模块化：调用特定的模块，完成特定任务

        <2>基于python语言实现，由Paramiko(完成基于ssh的连接)，PyYAML(对YAML文件的支持)，jinja2(python的模板库)三个关键的模块

        <3>部署简单：是没有客户端的

        <4>支持自定义模块，使用任意编程语言

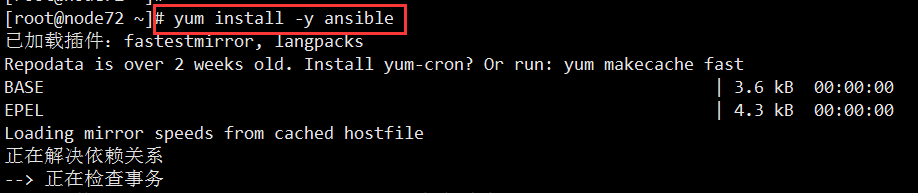
        <5>支持强大的playbook

        <6>具有幂等性：一个操作在一个主机上执行一遍和执行N遍的结果是一样的

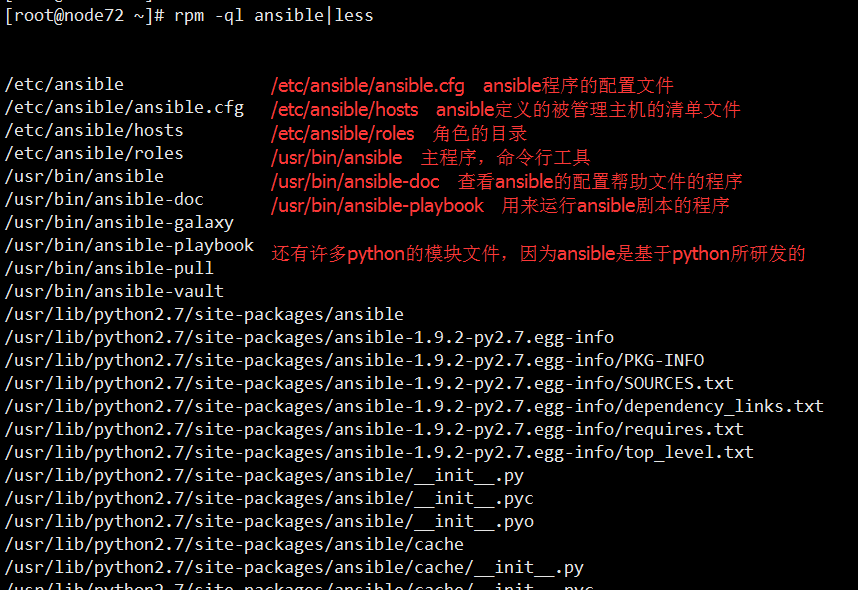
第二章    ansible的基础应用

    1、ansible管理端的安装

        在EPEL源中，有包含ansible的软件包安装只需要配置好EPEL的yum源，yum安装即可



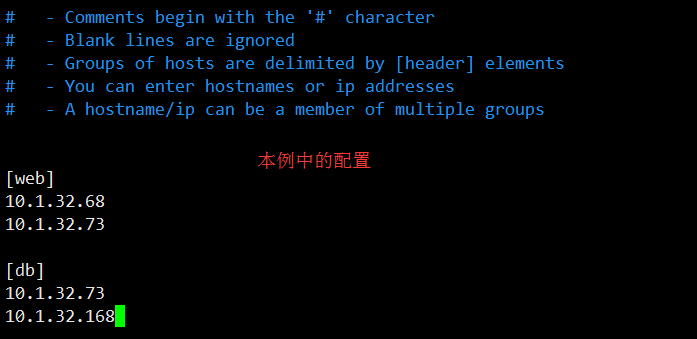
    安装生成的文件



    2、ansible被管控主机的定义

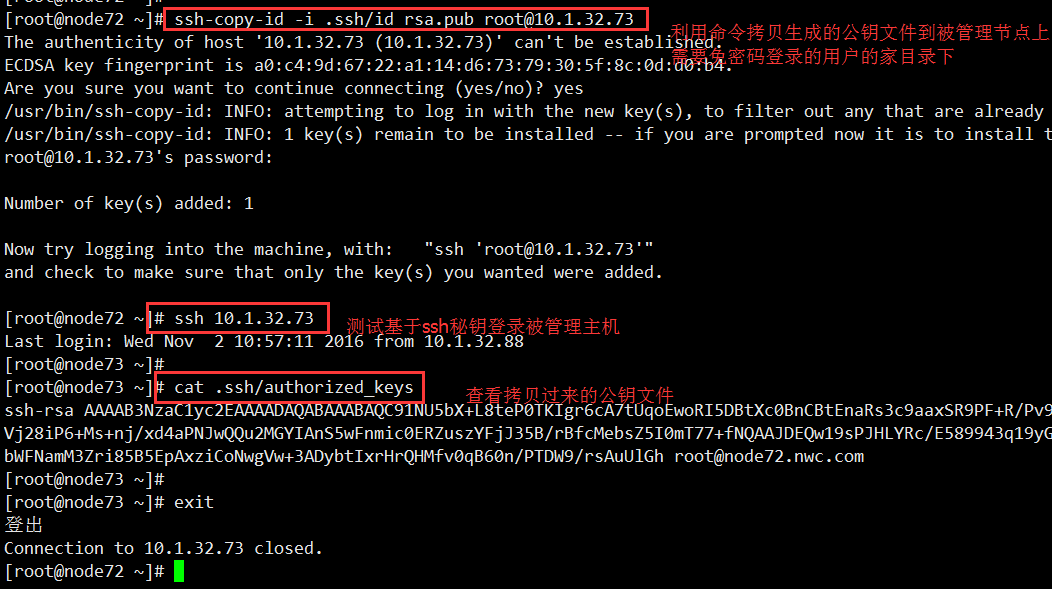
        对希望被管控的主机的定义需要实现在ansible管理端的针对被管理主机的配置文件(/etc/ansible/hosts)中进行定义





    3、配置管理节点可以基于ssh秘钥登录被管理节点





    4、ansible命令的用法介绍

ansible HOST-PATTERN [-f FORKS] [-m MOD\_NAME] [-a MOD\_ARGS]

            HOST\_PATTERN：指明对哪些被管控主机进行操作

            -f FORKS：表示一批处理几台主机，也就是当被管控主机很多时，ansible不是对所有主机同时发起管理操作，而是一批处理几台，然后再换一批，直到所有主机被处理完成，如果不指定，则默认是5台

            -m MOD\_NAME：指明调用哪个模块执行操作，各个模块所能实现的功能不同，如果不指定，默认是用-m command模块

            -a MOD\_ARGS：指明使用该模块的执行操作时的参数

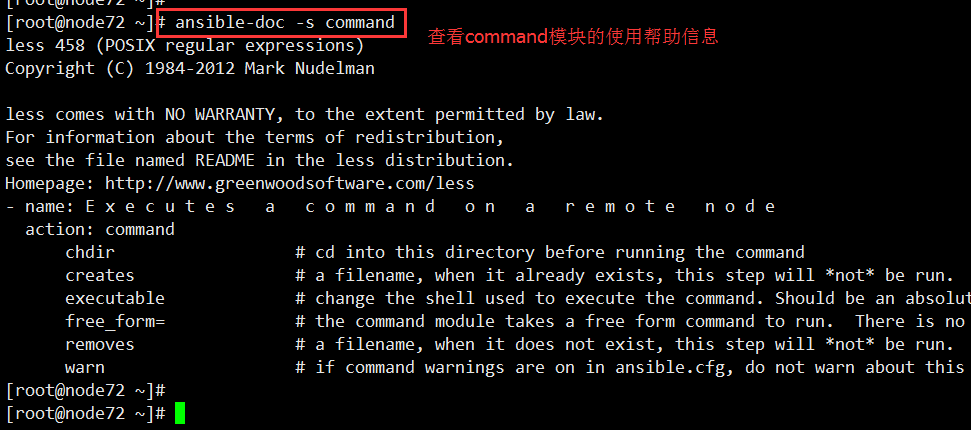


第三章    ansible常用模块介绍

    1、获取常用模块的列表和对应模块的使用帮助信息

        可用ansible-doc -l  来查看所有可用的模块列表

        可用ansible-doc -s MOD\_NAME  来查看对应模块的帮助信息



    2、command模块

        在远程主机执行命令,不支持管道，重定向等shell的特性

        常用参数有：

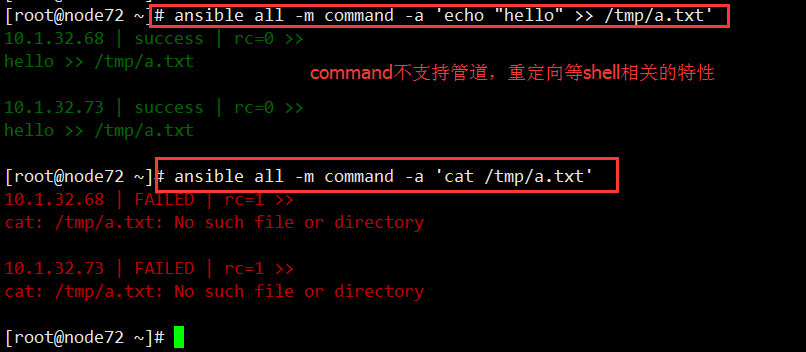
        chdir=   表示指明命令在远程主机上哪个目录下运行，也就是在命令执行前切换到哪个目录下

        creates=   在命令运行时创建一个文件，如果文件已存在，则不会执行创建任务

        removes=   在命令运行时移除一个文件，如果文件不存在，则不会执行移除任务

        executeble=   指明运行命令的shell程序





    3、shell模块

        在远程主机执行命令，相当于调用远程主机的shell进程，然后在该shell下打开一个子shell运行命令

        支持shell特性，如管道，重定向等

        注意：command和shell模块的核心参数直接为命令本身，而其他模块的核心参数一般是"key=value"格式

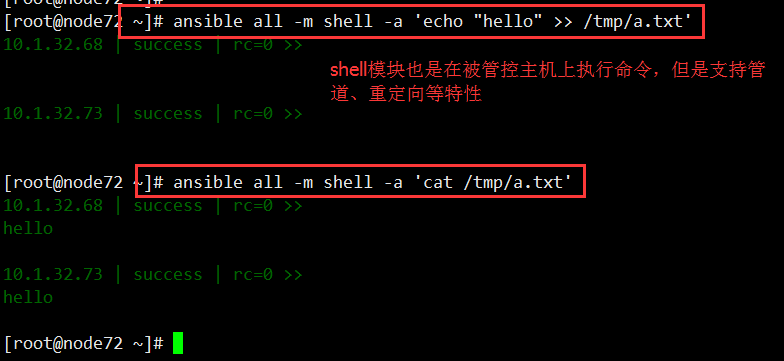
        常见参数有：

        chdir=   表示指明命令在远程主机上哪个目录下运行

        creates=   在命令运行时创建一个文件，如果文件已存在，则不会执行创建任务

        removes=  在命令运行时移除一个文件，如果文件不存在，则不会执行移除任务

        executeble=   指明运行命令的shell程序



    4、copy模块

        拷贝ansible管理端的文件到远程主机的指定位置

        常见参数有：

        dest=   指明拷贝文件的目标目录位置，使用绝对路径，如果源是目录，则目标也要是目录,如果目标文件已存在，会覆盖原有内容

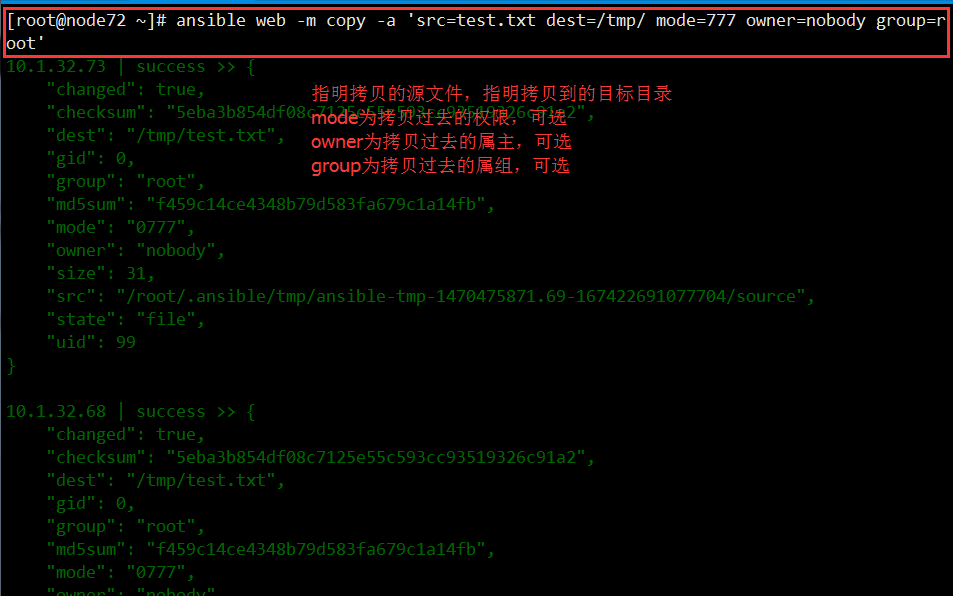
        src=   指明本地路径下的某个文件，可以使用相对路径和绝对路径，支持直接指定目录，如果源是目录，则目标也要是目录

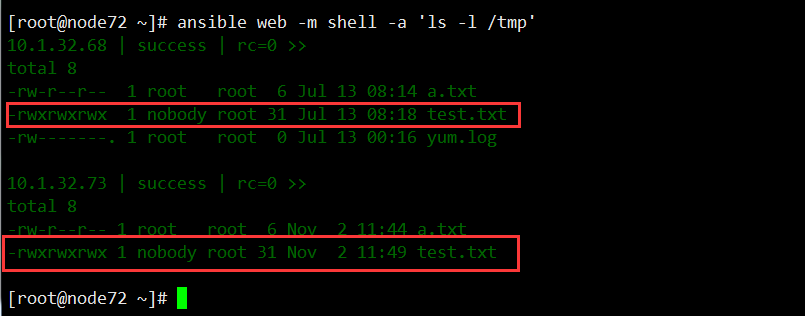
        mode=   指明复制时，目标文件的权限

        owner=   指明复制时，目标文件的属主

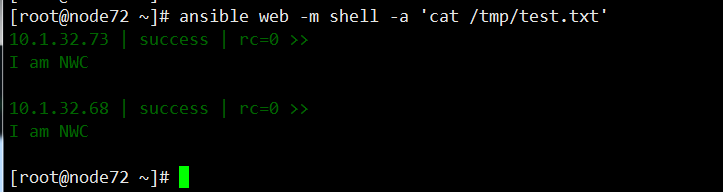
        group=   指明复制时，目标文件的属组

        content=  指明复制到目标主机上的内容，不能与src一起使用，相当于复制content指明的数据，到目标文件中









    5、cron模块

        管理计划任务的模块

        常见参数有：

        minute=  指明计划任务的分钟，支持格式：0-59，\*，\*/2等，与正常cron任务定义的一样的语法,省略时，默认为\*，也就是每分钟都执行

        hour=  指明计划任务的小时，支持的语法：0-23，\*，\*/2等，省略时，默认为\*，也就是每小时都执行

        day=  指明计划任务的天，支持的语法：1-31，\*，\*/2等，省略时，默认为\*，也就是每天都执行

        month=  指明计划任务的月，支持的语法为：1-12，\*，\*/2等，省略时，默认为\*，也就是每月都执行

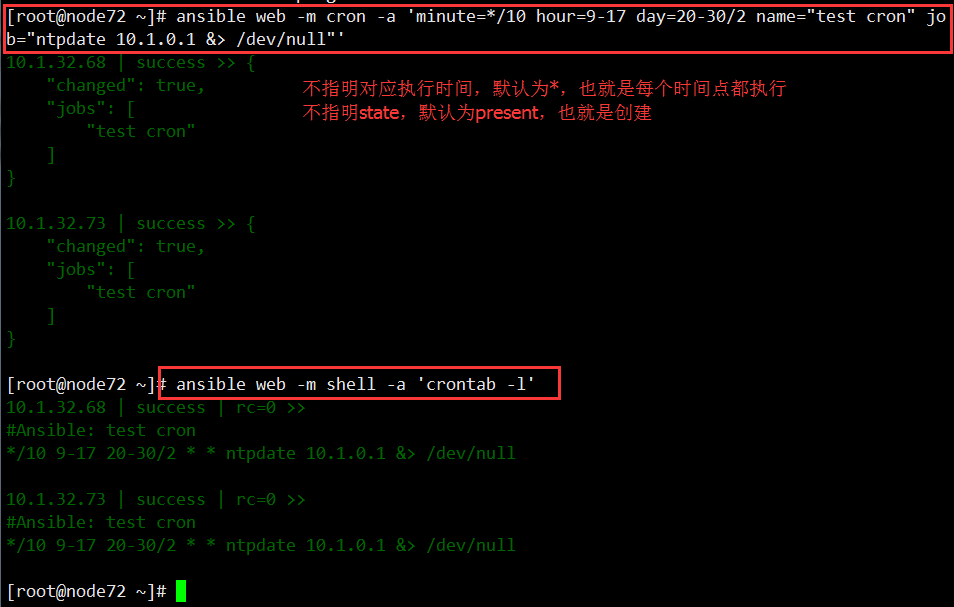
        weekday=  指明计划任务的星期几，支持的语法为：0-6，\*等，省略时，默认为\*，也就是每星期几都执行

        reboot  指明计划任务执行的时间为每次重启之后

        name=   给该计划任务取个名称,必须要给明。每个任务的名称不能一样。删除任务时，只需要给明任务的名称即可

        job=  执行的任务是什么，当state=present时才有意义

        state=present|absent   表示这个任务是创建还是删除，present表示创建，absent表示删除，默认是present





    6、fetch模块

        从远程主机拉取文件到本地

        一般情况下，只会从一个远程节点拉取数据

        常见参数有：

        dest=  从远程主机上拉取的文件存放在本地的位置，一般只能是目录

        src=   指明远程主机上要拉取的文件，只能是文件，不能是目录



    7、file模块

        用于设定远程主机上的文件属性

        常见参数有：

        path=   指明对哪个文件修改其属性

        src=   指明path=指明的文件是软链接文件，其对应的源文件是谁，必须要在state=link时才有用

        state=directory|link|absent   表示创建的文件是目录还是软链接

        owner=   指明文件的属主

        group=   指明文件的属组

        mode=   指明文件的权限

        创建软链接的用法：

            src=  path=  state=link

        修改文件属性的用法：

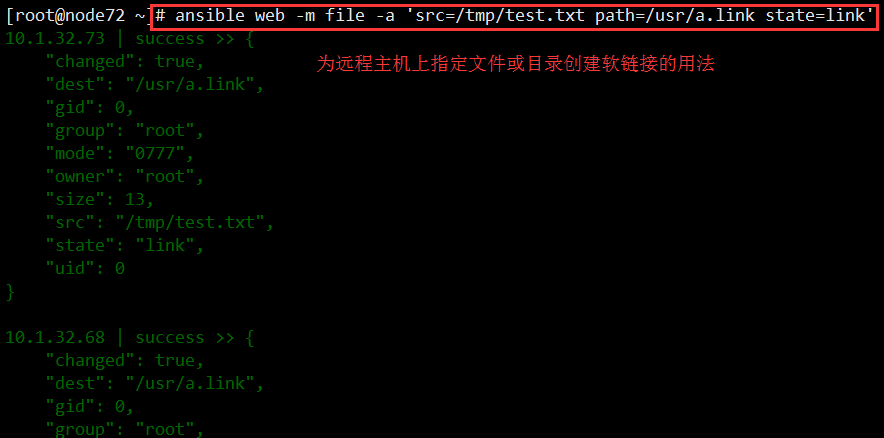
            path=  owner=  mode=  group=

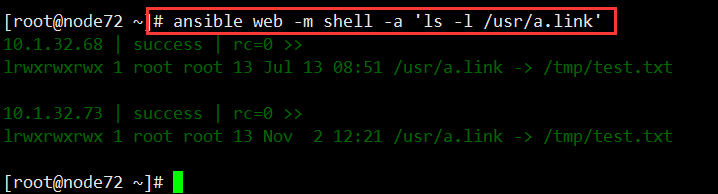
        创建目录的用法：

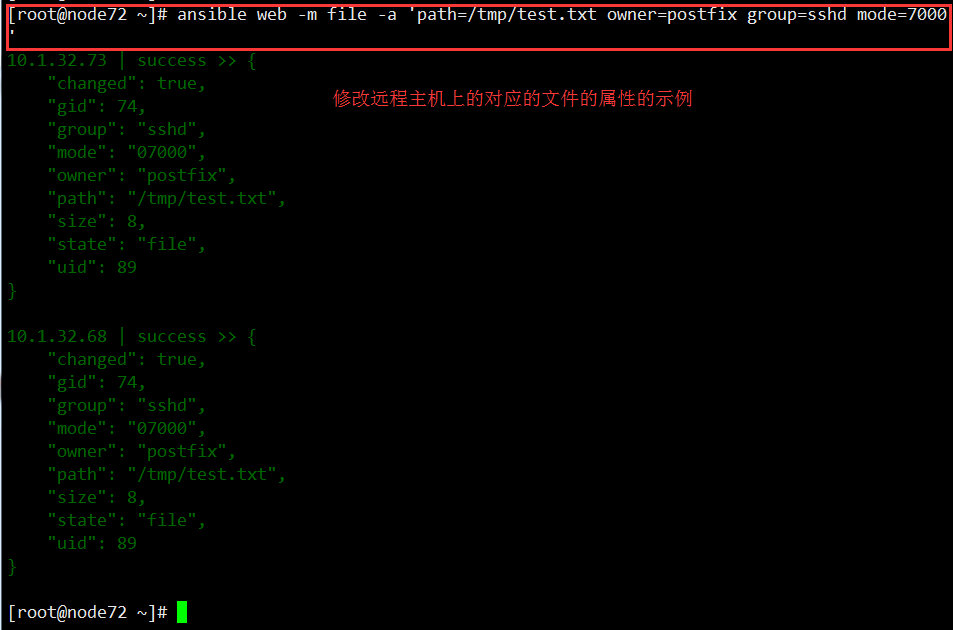
            path=  state=directory

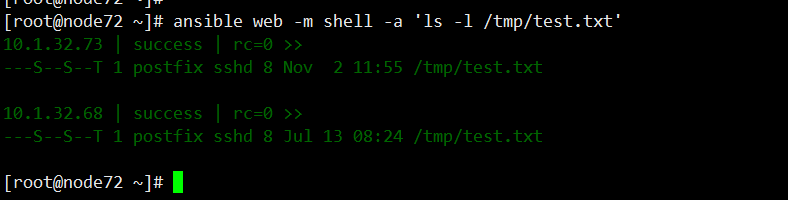
        删除文件：

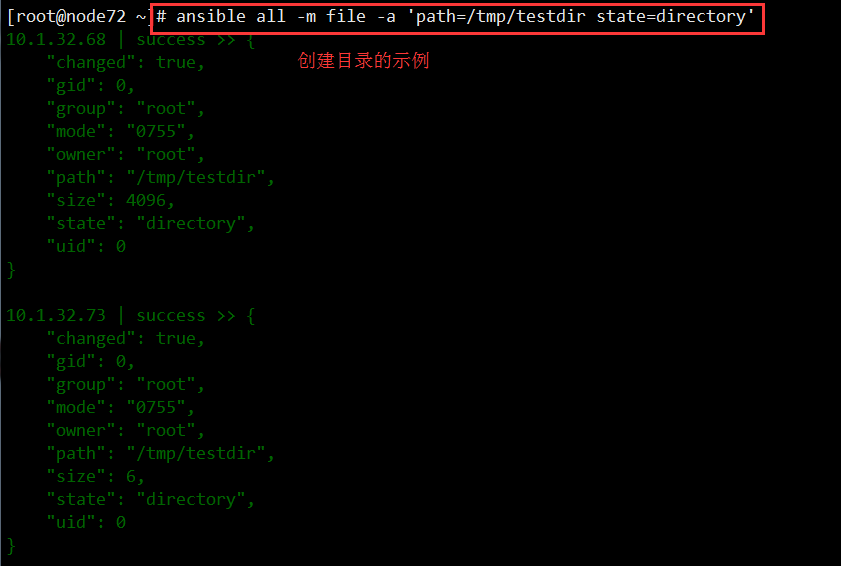
            path= state=absent

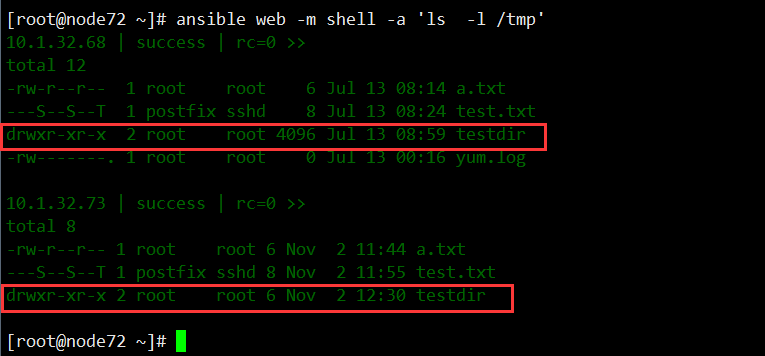


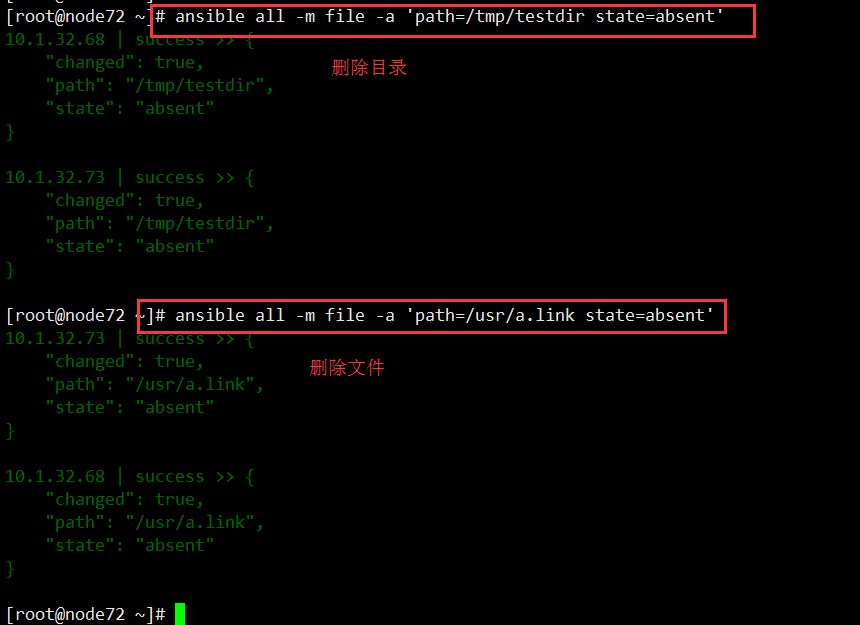


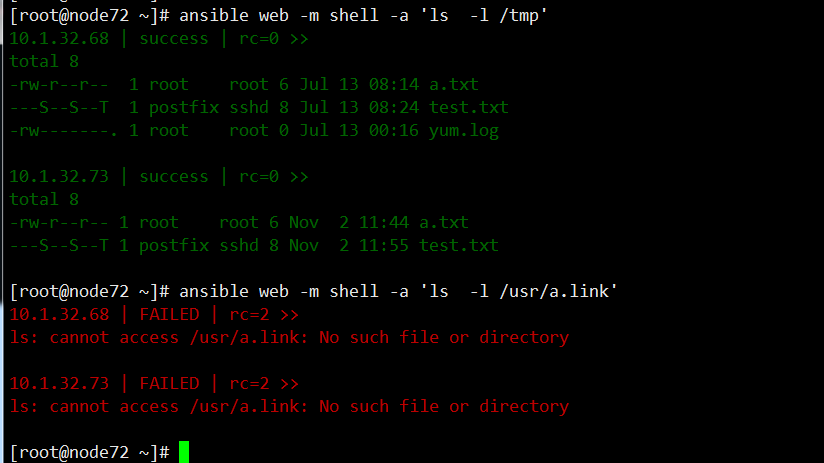










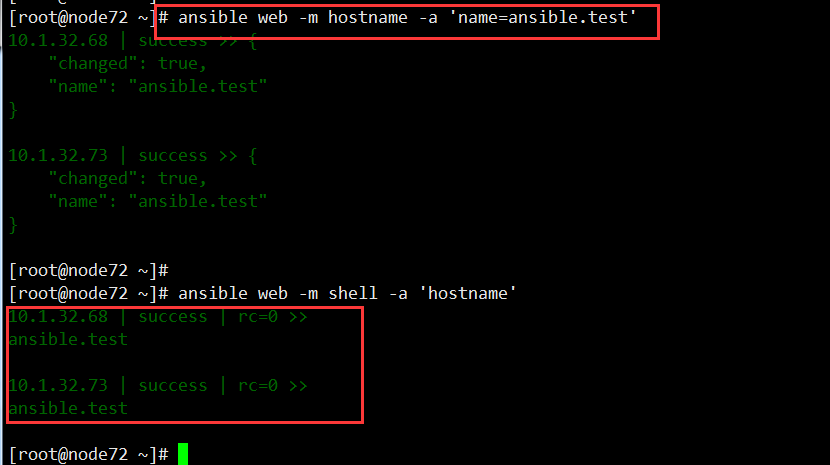


    8、hostname模块

        管理远程主机上的主机名

        常用参数有

        name=  指明主机名



    9、yum模块

        基于yum机制，对远程主机管理程序包

        常用参数有：

        name=   指明程序包的名称，可以带上版本号，不指明版本，就是默认最新版本。

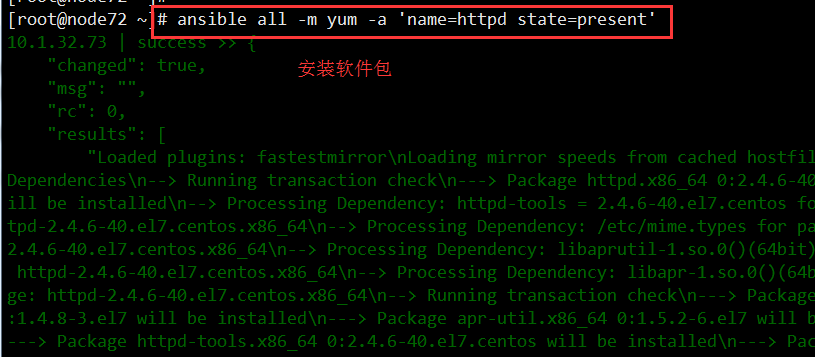
        state=present|latest|absent   指明对程序包执行的操作，present表示安装程序包，latest表示安装最新版本的程序包，absent表示卸载程序包

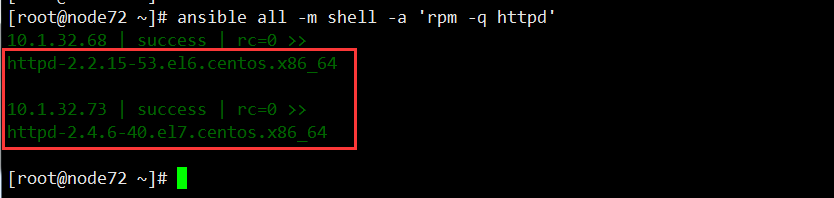
        disablerepo=    在用yum安装时，临时禁用某个仓库，仓库的ID

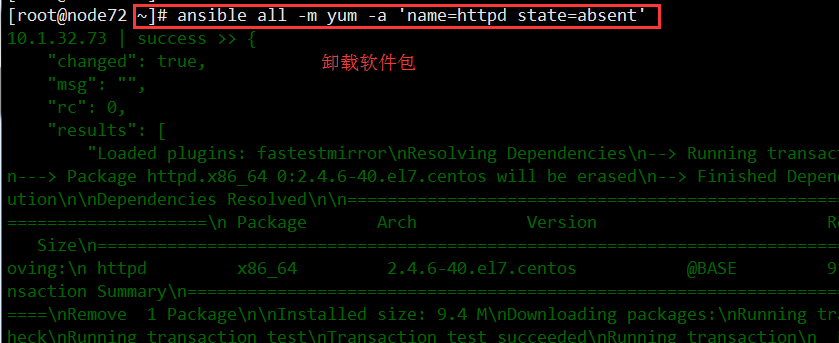
        enablerepo=    在用yum安装时，临时启用某个仓库,仓库的ID

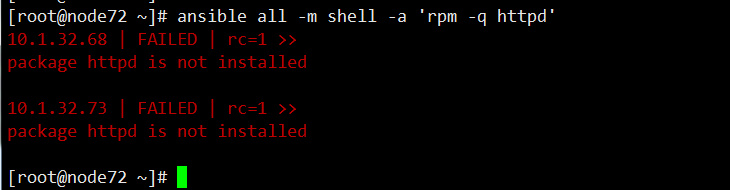
        conf\_file=   指明yum运行时采用哪个配置文件，而不是使用默认的配置文件

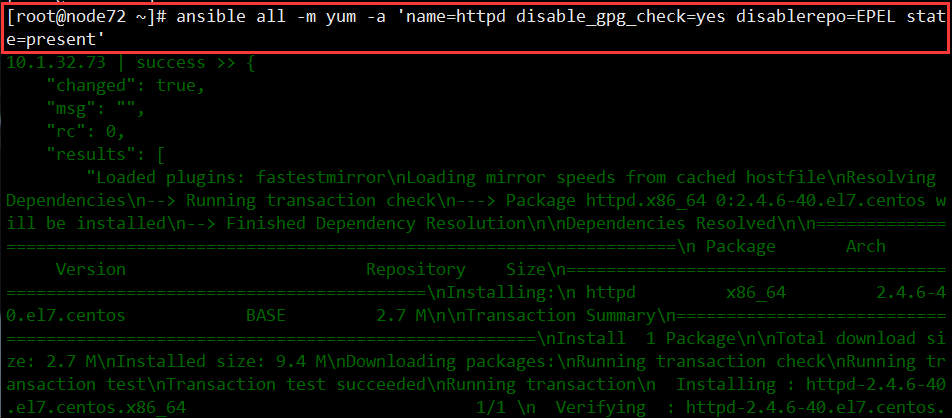
        diable\_gpg\_check=yes|no  是否启用gpg-check











    10、service模块

        用来管理远程主机上的服务的模块

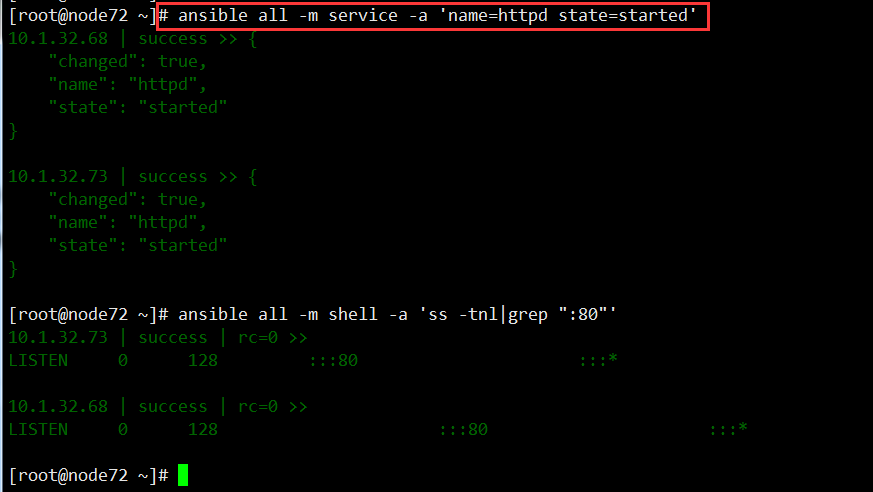
        常见参数有：

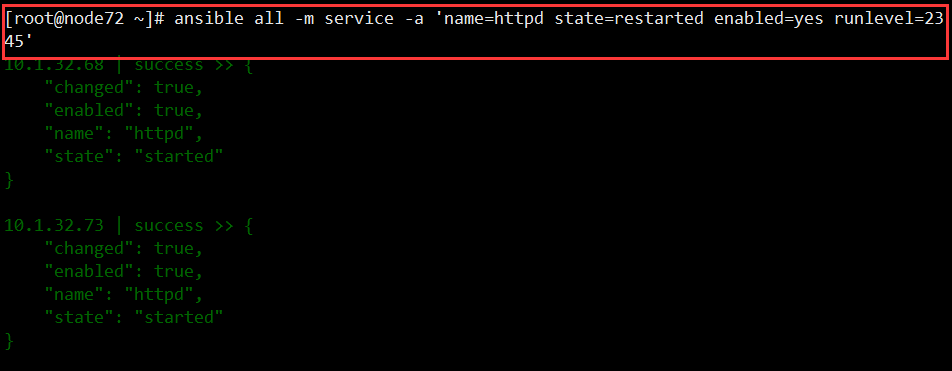
        name=   被管理的服务名称

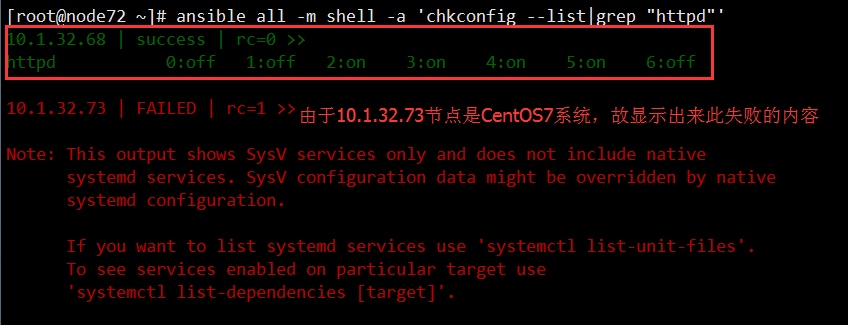
        state=started|stopped|restarted   表示启动或关闭或重启

        enabled=yes|no  表示要不要设定该服务开机自启动

        runlevel=   如果设定了enabled开机自动启动，则要定义在哪些运行级别下自动启动







    11、uri模块

        如果远端是web服务器，可以利用ansible直接请求某个网页

        常见参数有：

        url=  指明请求的url的路径，如：http://10.1.32.68/test.jpg

        user=  如果请求的url需要认证，则认证的用户名是什么

        password=  如果请求的url需要认证，则认证的密码是什么

        method=  指明请求的方法，如GET、POST…

        body=   指明报文中实体部分的内容，一般是POST方法或PUT方法时用到

        HEADER\_   自定义请求报文中的添加的首部

    12、user模块

        管理远程主机上的用户的账号

        常见参数有：

        name=   指明要管理的账号名称

        state=present|absent   指明是创建账号还是删除账号，present表示创建，absent表示删除

        system=yes|no   指明是否为系统账号

        uid=   指明用户UID

        group=   指明用户的基本组

        groups=   指明用户的附加组

        shell=   指明默认的shell

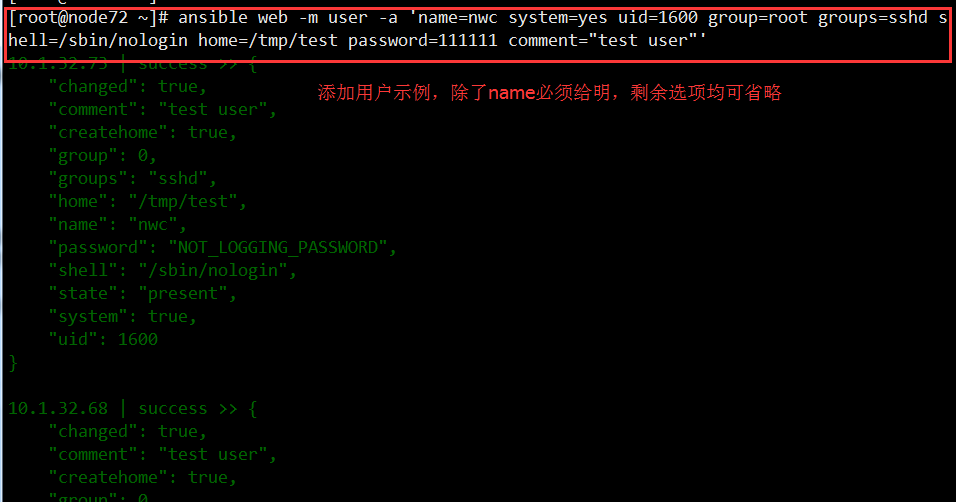
        home=   指明用户的家目录

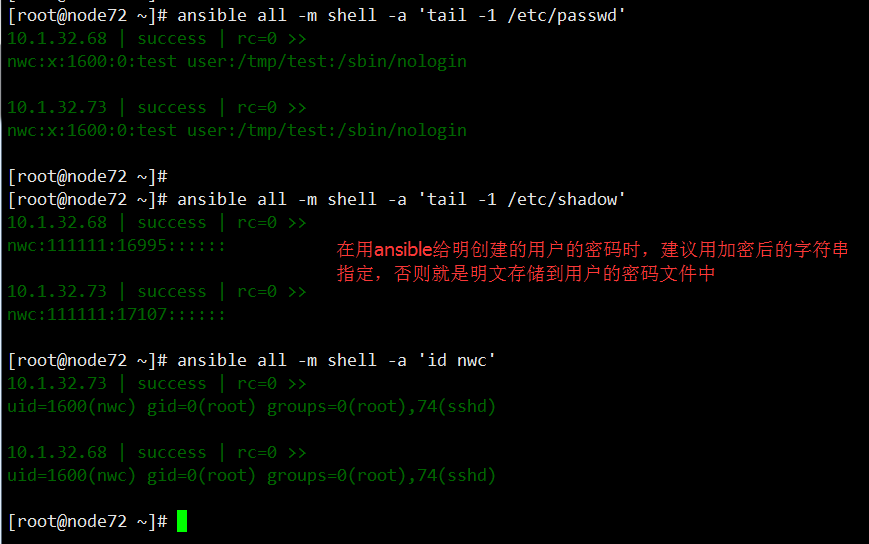
        move\_home=yes|no   当home设定了家目录，如果要创建的家目录已存在，是否将已存在的家目录进行移动

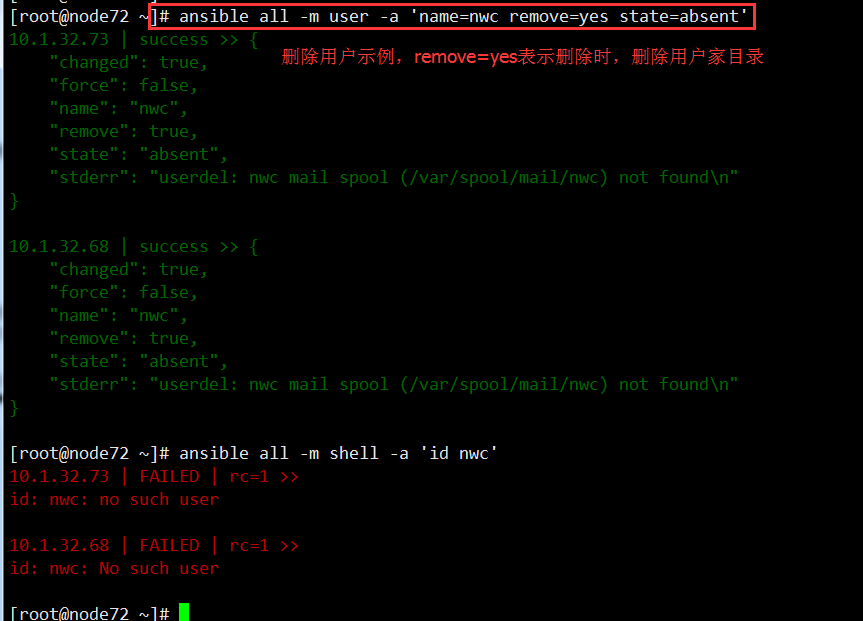
        password=   指明用户的密码，最好使用加密好的字符串

        comment=   指明用户的注释信息

        remove=yes|no   当state=absent时，也就是删除用户时，是否要删除用户的而家目录







    13、group模块

        用来添加或删除远端主机的用户组

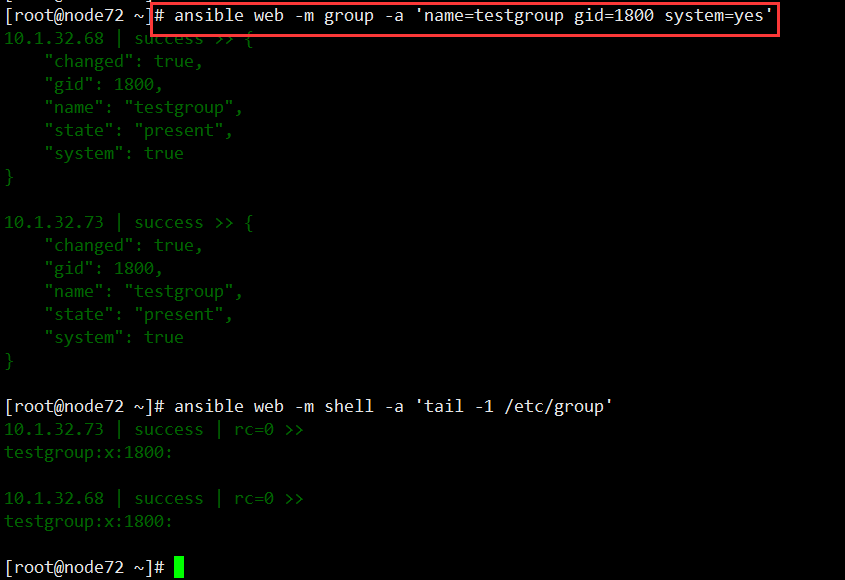
        常见参数有：

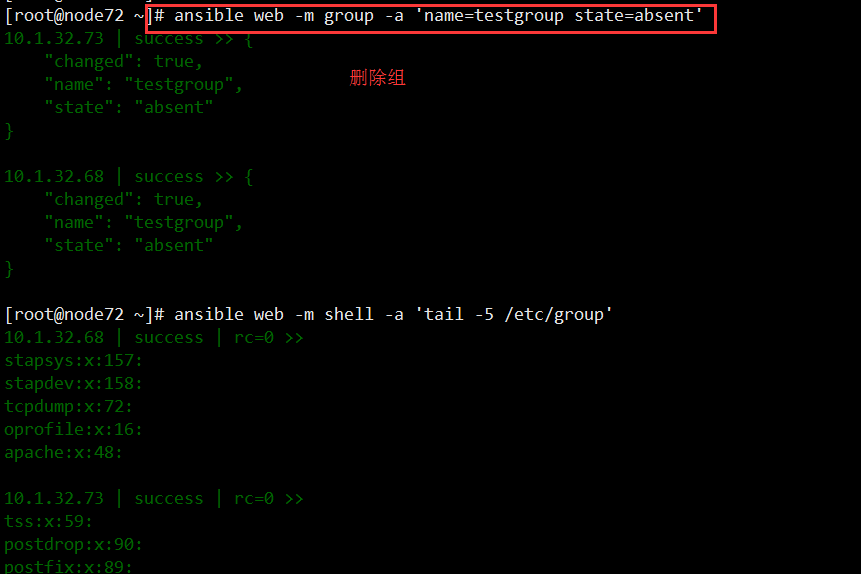
        name=   被管理的组名

        state=present|absent   是添加还是删除,不指名默认为添加

        gid=   指明GID

        system=yes|no   是否为系统组





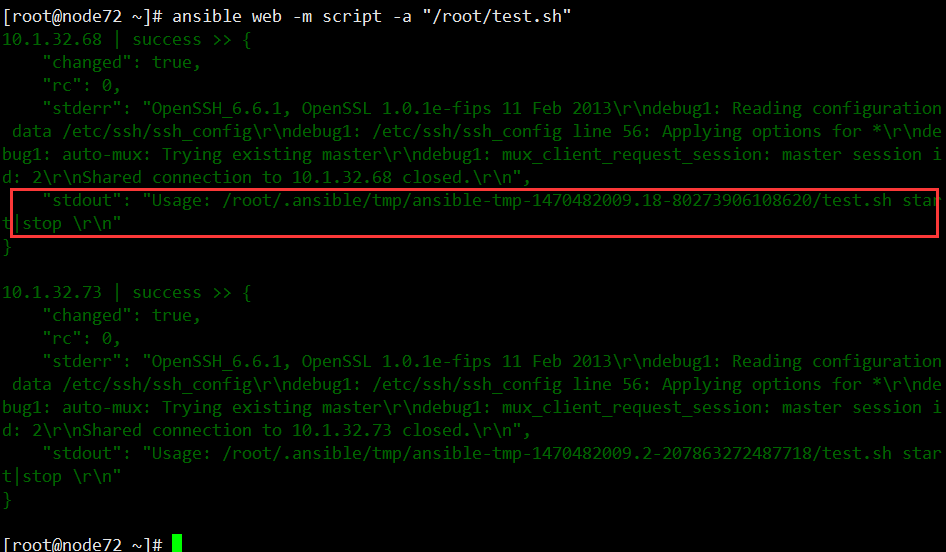
    14、script模块

        将管理端的某个脚本，移动到远端主机(不需要指明传递到远端主机的哪个路径下，系统会自动移动，然后执行)，然后执行

        一般是自动移动到远端主机的/root/.ansible/tmp目录下，然后自动给予其权限，然后再开个子shell然后运行脚本，运行完成后删除脚本

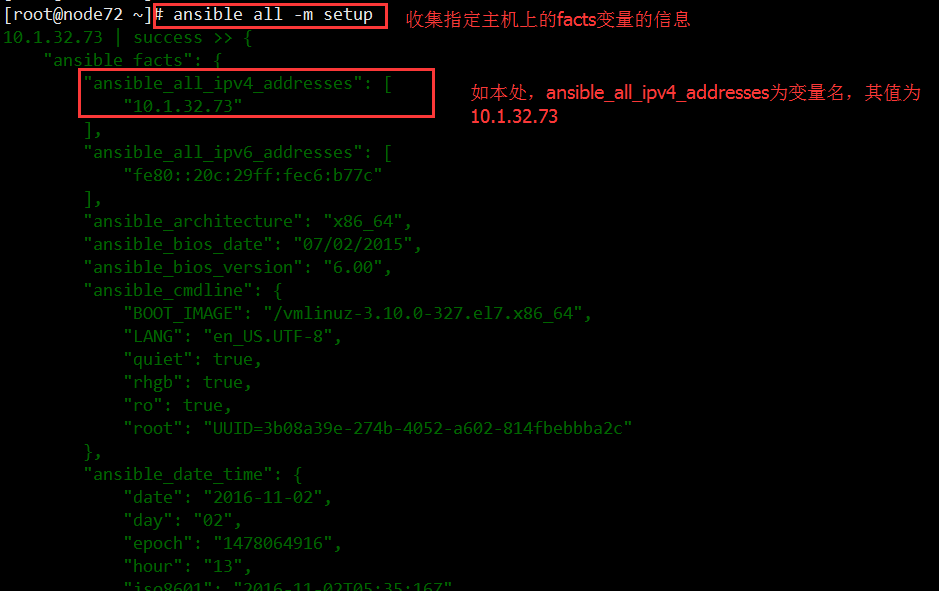






    15、setup模块

        可收集远程主机的facts变量的信息，相当于收集了目标主机的相关信息(如内核版本、操作系统信息、cpu、…)，保存在ansible的内置变量中，之后我们有需要用到时，直接调用变量即可



    16、template模块的使用

        基于模板方式，生成一个模板文件，复制到远程主机，让远程主机基于模板，生成符合远程主机自身的文件

        注意：此模块不能在命令行使用，只能用在playbook中

        常见的参数有：

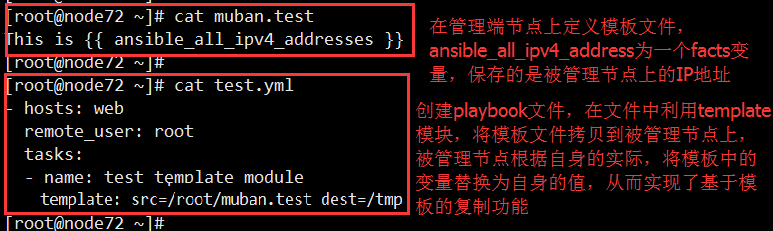
        src=  指明管理端本地的模板文件的目录

        dest=   指明将模板文件拷贝到远程主机的哪个目录下

        owner=  指明拷贝到远程主机的文件的属主

        group=  指明拷贝到远程主机的文件的属组

        mode=   指明拷贝到远程主机的文件的权限





第四章    ansible的playbook基础应用介绍

    当需要执行的任务有多个时，需要一条一条编辑ansible命令，然后执行，而且当需要重复执行时，又要重新编辑执行，这样效率不高，因此ansible就可以利用playbook来完成将任务写到一个YAML格式的文件中，然后利用ansible-playbook进行调用该文件，从而实现了多条语句，可重复执行的效果，类似shell脚本的效果，ansible的playbook要借助YAML文件来实现，YAML文件扩展名通常为.yaml或.yml

    1、YAML文件的语法

        YAML语法和其他高阶语言类似，并可以简单表达清单，散列表、标量等数据结构。其结构通过空格来展示，序列里的项用“-”来代表，Map你的键值用“；”分隔

    YAML文件中列表的表示：列表中的所有元素均使用"-" 开头，例如：

        – apple

        – orange

        – mango

    YAML文件中字典的表示：字典通过key与value进行标识，如：

        name: nwc

        job: manager

        sex: M

        也可以将key:value放置于{}中进行表示，如

        {name: nwc,job: manager,sex: M}

    2、playbook的核心元素

        Hosts：运行在哪些主机之上

        Users：远程主机上，运行此任务的身份，不指名默认为root

        Tasks：任务，也就是定义的具体任务，由模块定义的操作的列表

        Variables：变量

        Templates：模板，包含了模板语法编写的模板的文本文件

        Handlers：处理器，类似Tasks，只是在特定的条件下才会触发的任务

            某任务的状态在运行后为changed时，可通过"notify"通知给相应的handlers进行触发执行

        Roles：角色，将Hosts剥离出去，由Tasks、Variables、Templates、Handlers所组成的一种特定的结构的集合

    3、playbook的基础组件

        hosts：运行指定任务的而目标主机，多个主机用:冒号分隔

        remote\_user：在远程主机上执行任务的用户;可以全局指定，也可以单个任务指定

        sudo\_user：表示以sudo方式运行任务时，切换为哪个用户身份运行

        tasks：

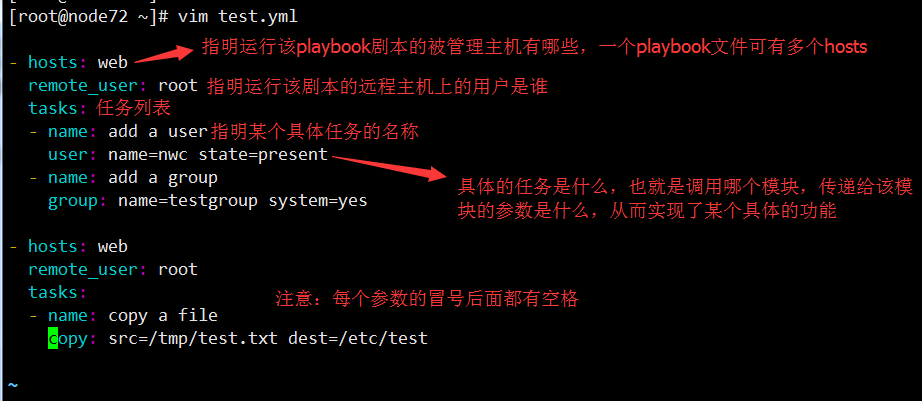
            任务列表，ansible运行任务的方式为，将第一个任务在所有主机上运行完成，然后再将第二个任务在所有主机上运行…，当某个任务在某个主机上运行出现故障，会造成任务终止，再次执行任务只需直接执行即可

            定义任务列表，实际就是指明使用的模块和对应的模块参数来完成的任务的列表，其格式有两种：

                (1)action:MODULE  ARGUMENTS

                (2)MODULE:ARGUMENTS

            注意：shell和command模块后面直接跟命令，而不是key=value的参数列表



    4、playbook文件的执行

        playbook文件定义的任务要向执行，需要利用ansible-playbook命令进行调用

        ansible-playbook命令用法：

        <1> 检测语法

            ansible-playbook –syntax-check /PATH/TO/PLAYBOOK.yaml

        <2> 测试运行

            ansible-playbook -C|–check /PATH/TO/PLAYBOOK.yaml

            只检测执行指定的YAML文件可能会发生改变，但不真正执行操作，相当于测试运行

                –list-hosts  检测YAML文件可能影响到的主机列表

                –list-tasks   列出YAML文件的任务列表

                –list-tags    列出YAML文件中的标签

        <3> 运行

            ansible-playbook /PATH/TO/PLAYBOOK.yml

            可用选项：

            不加任何选项表示完整运行整个playbook文件

            -t TAGS,–tags=TAGS  表示只执行那个标签的任务

            –skip-tags=SKIP\_TAGS  表示除了指明的标签的任务，其他任务都执行

            –start-at-task=START\_AT   从指明的任务开始往下运行

        <4> 通常情况下剧本的执行过程

            先要利用 ansible-playbook -C|–check /PATH/TO/PLAYBOOK.yaml进行测试，测试没问题后

            再利用 ansible-playbook /PATH/TO/PLAYBOOK.yml正式执行





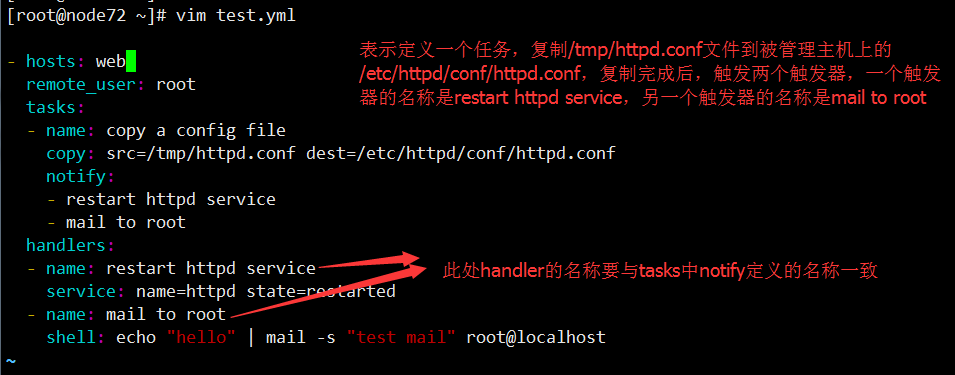
第五章    playbook中的handlers(触发器)的介绍

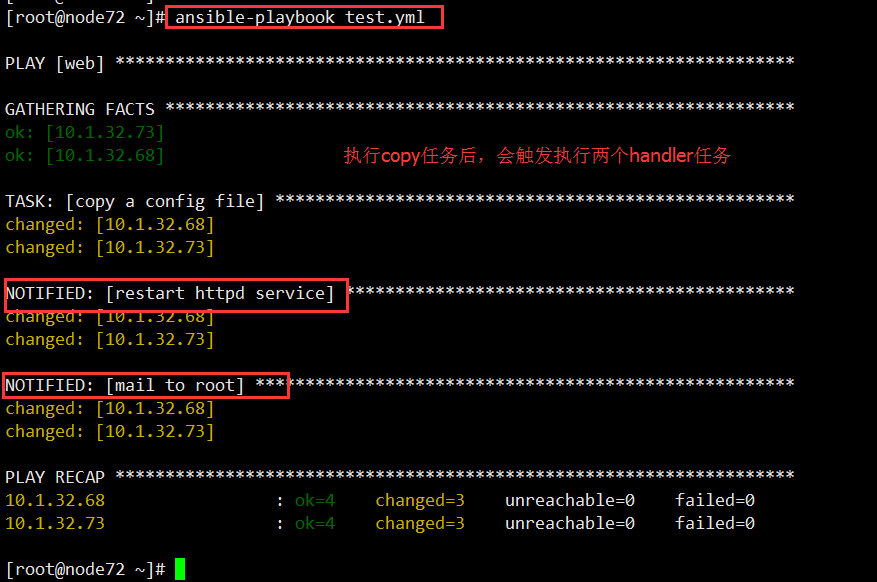
    1、handlers的作用

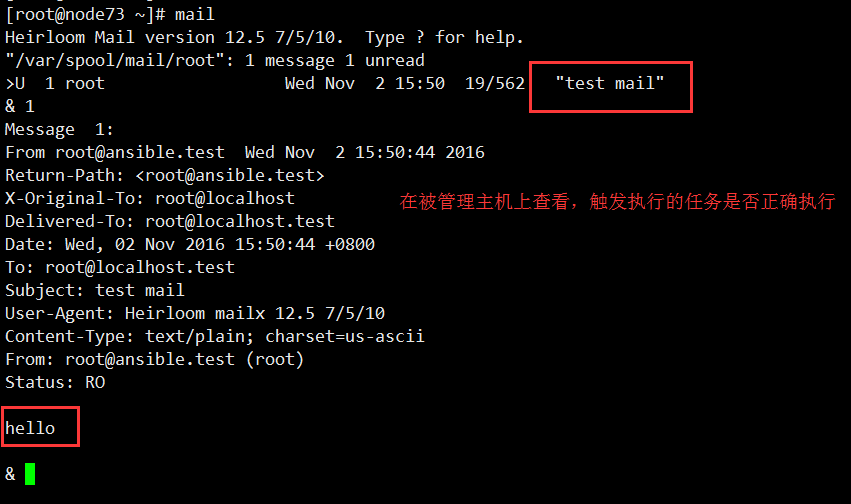
        用于当关注资源发生变化时采取一定的操作，可理解为：当之前定义在tasks中的任务，如果执行成功后，我们希望在此基础上触发某个别的任务，这时就需要定义handlers。

        要想handlers生效，首先需要在tasks的任务中定义一个notify，表示执行成功后，通知执行哪个handler，然后再定义handlers中，定义handler任务，handler任务的name要与notify中定义通知给哪个handler的名称一致

    2、handlers触发器的使用示例：







第六章    playbook中的tags(标签)的介绍

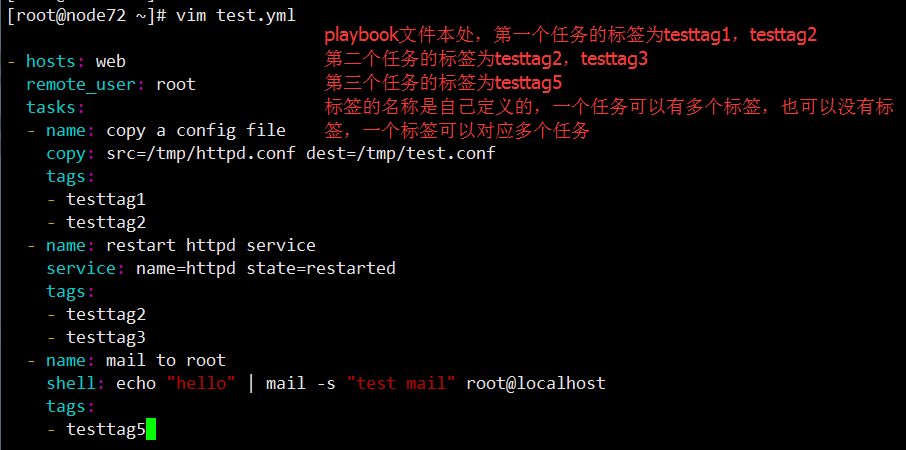
    1、tags标签的作用

        当我们定义了一个playbook文件，文件有很多任务要执行，如果我们只是希望执行其中的某一个任务，则可以在编写该任务时，为该任务加上标签，然后利用ansible-playbook调用时，指明只执行那个tags标签的任务(ansible-playbook -t TAG\_NANE YAML文件)

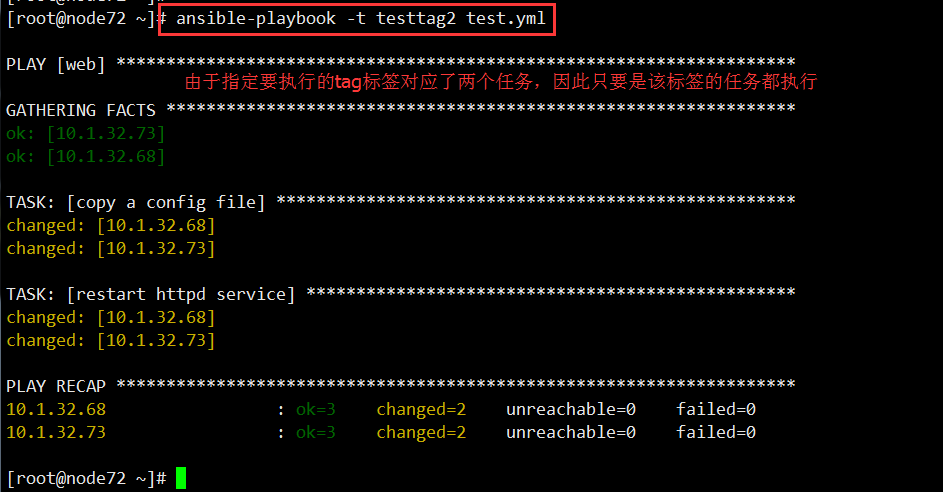
        可以将多个任务提供一样的标签，这样，就可以实现指定运行某标签的任务时，同时运行多个任务；也支持一个任务定义多个标签

        可以在用ansible-playbook利用-t指明执行的标签的任务时，支持用逗号,隔开的多个标签，则也是多个标签的任务都执行

    2、tags标签的示例









第七章    playbook中的variables(变量)的介绍

    在playbook中可在各个任意地方使用变量，引用变量的格式为：{{ VAR\_NAME }}，变量名与大括号之间有空格

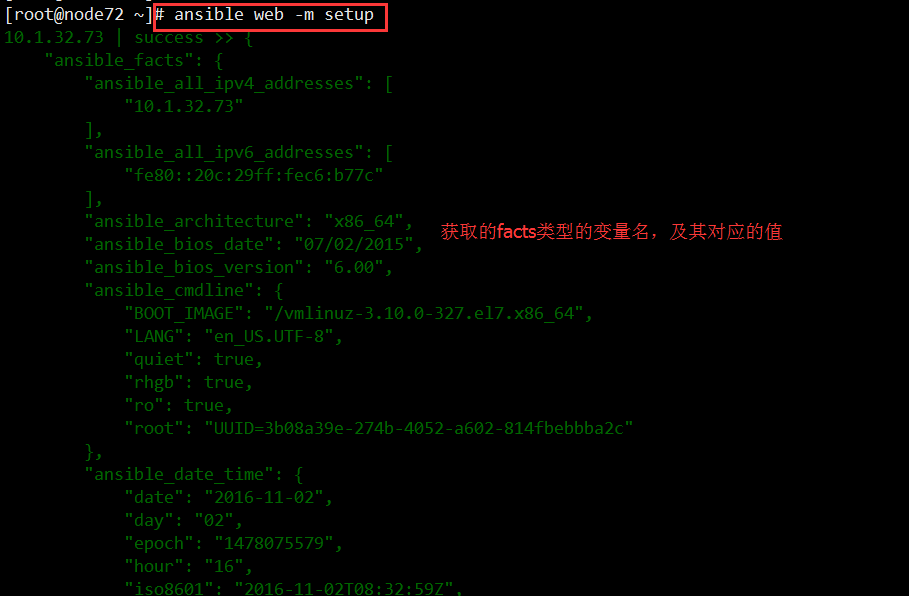
    定义变量的方式分别为：

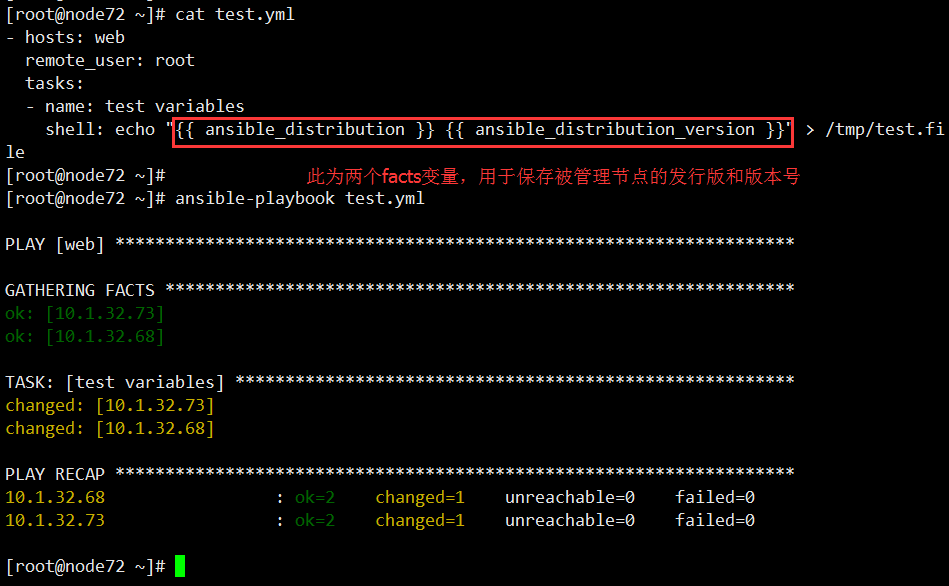
    1、facts类型的变量：

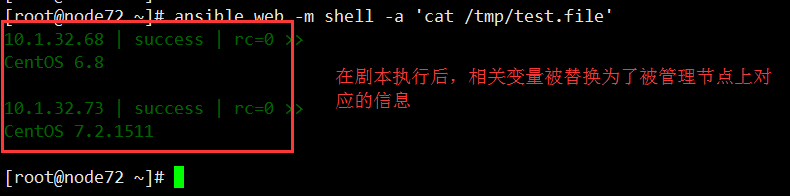
        可直接调用，是ansible收集的关于被管理主机的相关信息，其被保存在ansible的一些变量中，如果要查看某个被管理主机有哪些facts变量可用，则可以执行：

        ansible 10.1.32.72 -m setup

        可列出10.1.32.72主机上可用的所有的facts变量及其值



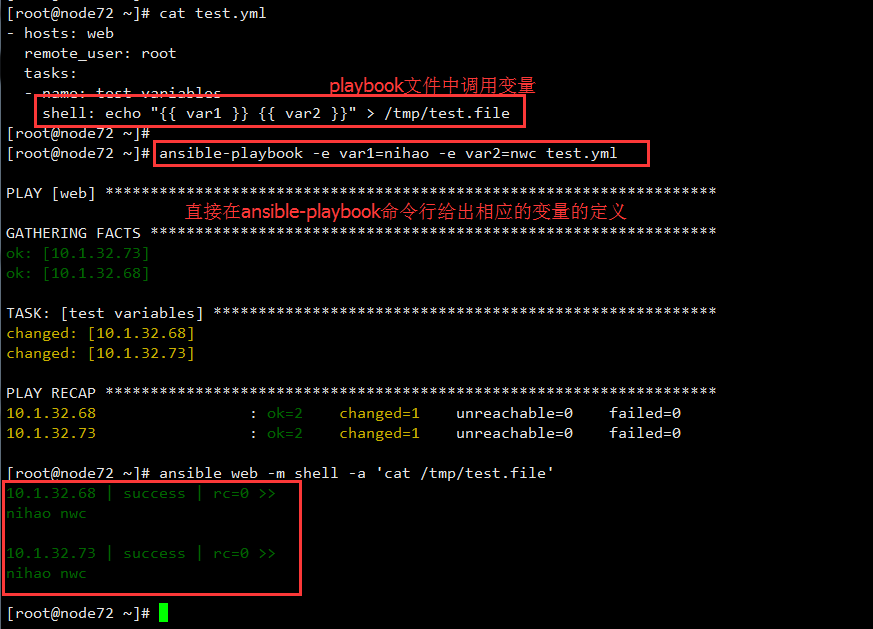




    2、ansible-playbook命令的命令行中的自定义变量：

        ansible-playbook -e VARS=VALUE

        如果要指定多个变量，则用多个-e引导即可



    3、在定义主机的hosts中(也就是/etc/ansible/hosts文件中)定义变量

        <1>实现向不同的主机传递不同的变量

            如：vim /etc/ansible/hosts

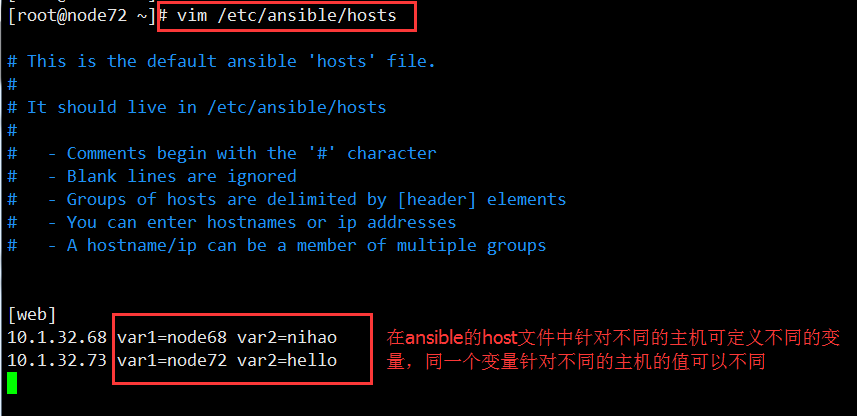
            [webserver]

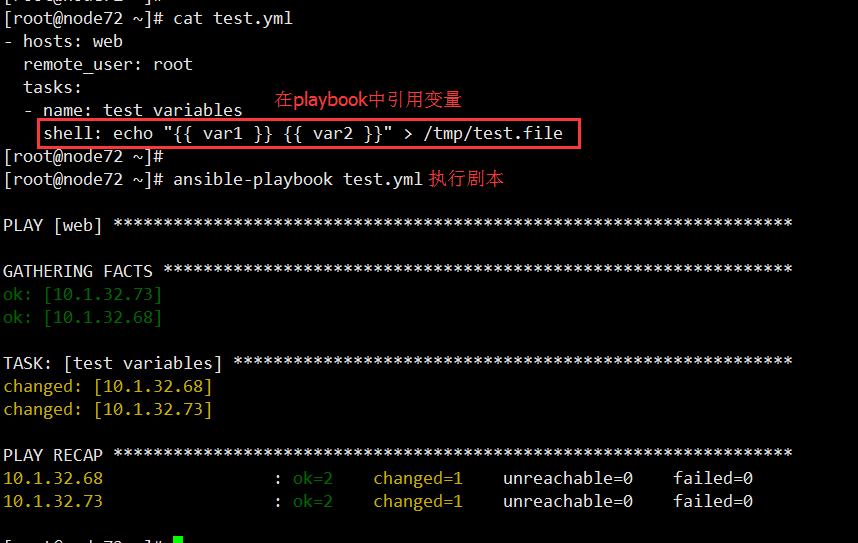
            10.1.32.72 hname=web1 aaa=111 bbb=test

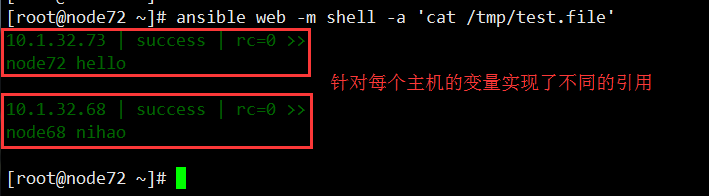
            10.1.32.73 hname=web2 aaa=222

            表示针对10.1.32.72这台主机，hname这个变量的值为8080，aaa变量的值为111，bbb变量的值为test

            针对10.1.32.73这台主机，hname这个变量的值为8090，aaa变量的值为222







        <2>实现向某个组内的主机，传递相同的变量

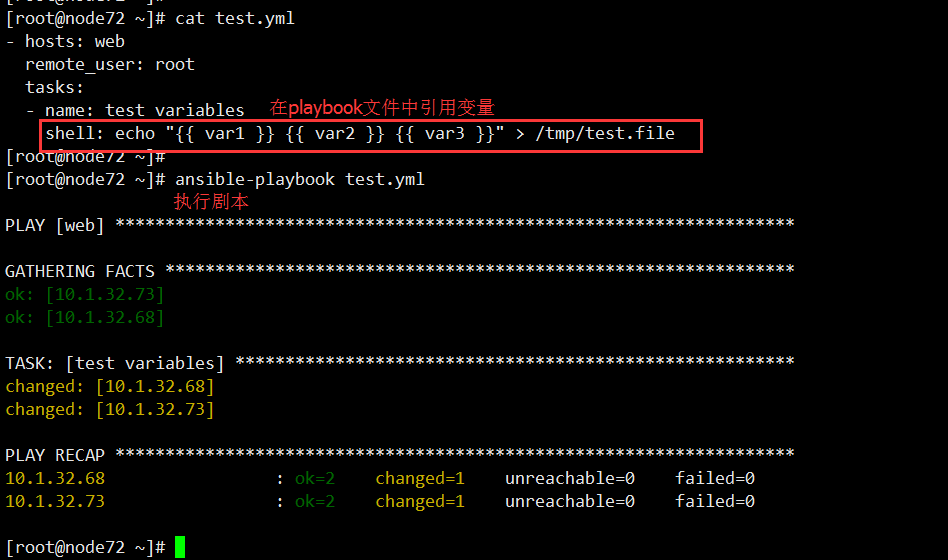
            如：vim /etc/ansible/hosts

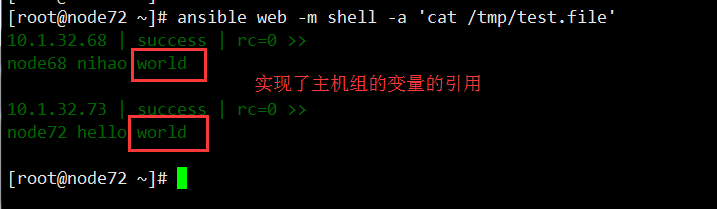
            [webserver:vars]

            http\_port=8080

            表示向webserver组内的主机定义相同的变量http\_port,其值都为8080







        <3>hosts文件中在每个主机后面可以用ansible\_ssh\_user和ansible\_ssh\_pass来指明ansible连接该主机时，不是采用我们之前自己手动执行的利用ssh秘钥登录，而是直接将用户名密码写入到/etc/ansible/hosts文件中，每次连接都是基于用户名密码的登录

            如：vim /etc/ansible/hosts

            [dbserver]

            10.1.32.68 ansible\_ssh\_user=root ansible\_ssh\_pass=123456

            10.1.32.73

            注意，此类参数不能传递给playbook，也就是无法进行调用，只是用来ansible连接远程主机时的定义

            除了上面的两个参数，还有：

                ansible\_ssh\_host  连接的远程主机

                ansible\_ssh\_port  连接的远程主机的端口

                ansible\_sudo\_pass   以sudo方式运行任务时的sudo用户的密码

    4、在playbook的yaml文件中定义变量

        如:有个YAML文件为/root/test.yaml，内容为

        – hosts: webserver

          remote\_user: root

          vars:

          – pkname: httpd

          – yname: php

         tasks:

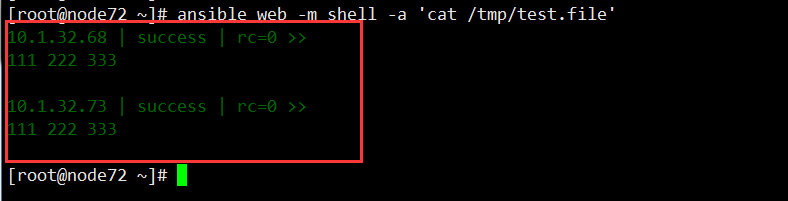
         – name: install packages

           yum: name={{ pkname }} state=present

         – name: install packages2

           yum: name={{ yname }} state=present





第八章    playbook中的templates(模板)的介绍

    1、templates模板文件的说明

        templates是模板文本文件，但是此文本文件内部嵌套有脚本(脚本是使用模板编程语言编写)，主要用来提供文件模板，让被管理主机根据模板中定义的脚本，生成符合远程主机环境的文件，也就是相当于提供一个模板文件，拷贝到被管理主机上，被管理主机根据自身的情况，生成符合自身实际的文件，模板文件中的语法，是语句模板编程语言所定义的，在ansible中，是jinja2的语法格式(因为ansible是python语言开发，而python嵌入文本中的语言是jinja2)

    2、jinja2常用的语法

        数据类型(字面量)：

            字符串：使用单引号或双引号引用起来的都被认为是字符串

            数字：整数、浮点数，不能用引号

            列表：[item1,item2,…]

            元组：(item1,item2,…)

            字典：{key1:value1,key2:value2,…}

                字典的key一般是字符串，所以要用引号引起来

            布尔型：true/false

        算数运算：

            +、-、\*、/、//(除完以后只保留商)、%(取模，除完以后只留余数)、\*\*(次方)

        比较操作：

            ==、!=、>、>=、<、<=

        逻辑运算：

            and、or、not

        变量引用：与YAML语法一样

            {{ VAR\_NAME }}

        迭代(循环)、条件判断

    3、template模块

        当模板文件生成后，就可以借助template模块，将模板文件拷贝到被管控主机上，生成符合远端主机环境的文件，注意不能用copy模块进行拷贝，因为copy模块拷贝时，模板文件中定义的一些jinja2的语法结构会被当做纯文本信息进行拷贝，而用template模块进行拷贝时，则会识别jinja2的语法，将对应的语法替换为符合远端主机的具体的值，生成符合远端主机环境的文件

        template模块的参数有：

            src=  指明管理端本地的模板文件的目录

            dest=   指明将模板文件拷贝到远程主机的哪个目录下

            owner=  指明拷贝到远程主机的文件的属主

            group=  指明拷贝到远程主机的文件的属组

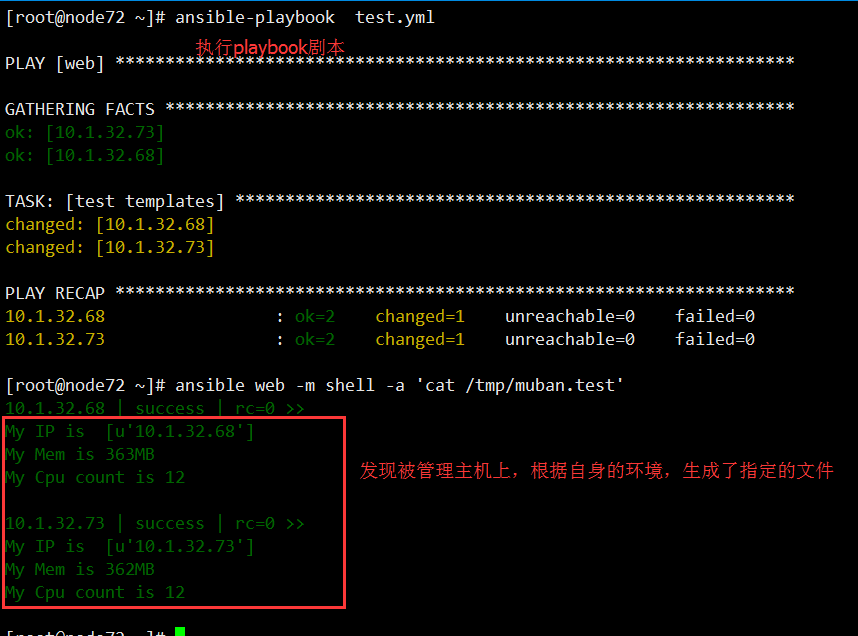
            mode=   指明拷贝到远程主机的文件的权限

    4、template模板使用配置示例



变量的获取可以参考这个

<https://docs.ansible.com/ansible/latest/user_guide/playbooks_variables.html>



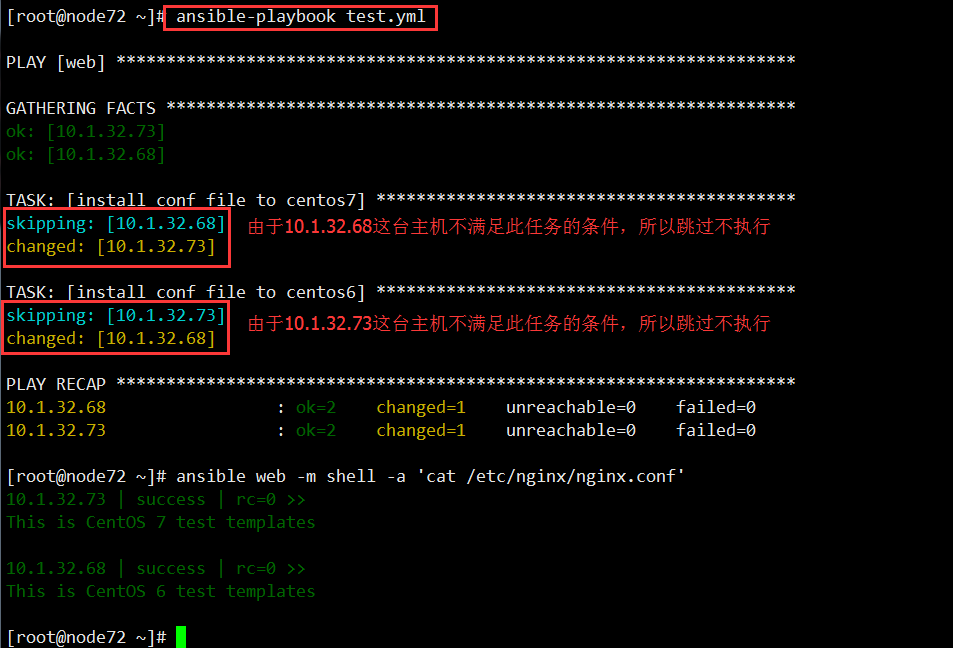
第九章    playbook中的条件判断机制的介绍

    当我们希望在playbook文件中，完成诸如在某条件满足时，才执行指定的任务时，就需要借助条件判断机制。

    要想使用条件判断，可以在tasks中使用when语句，标明在什么情况下，才执行该任务，when语句支持jinja2的语法格式

    示例：





第十章    playbook中的循环(迭代)机制的介绍

    1、循环的相关概念