本节所讲内容：

21.1 Ansible概述和运行机制

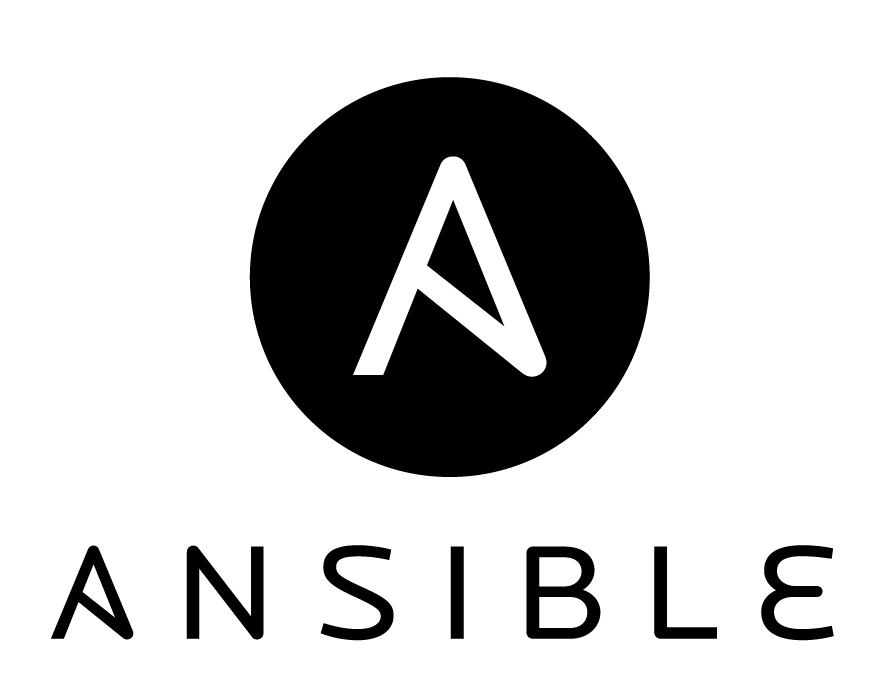
21.2 实战-安装并配置Ansible管理两个节点

21.3 Ansible常见模块高级使用方法

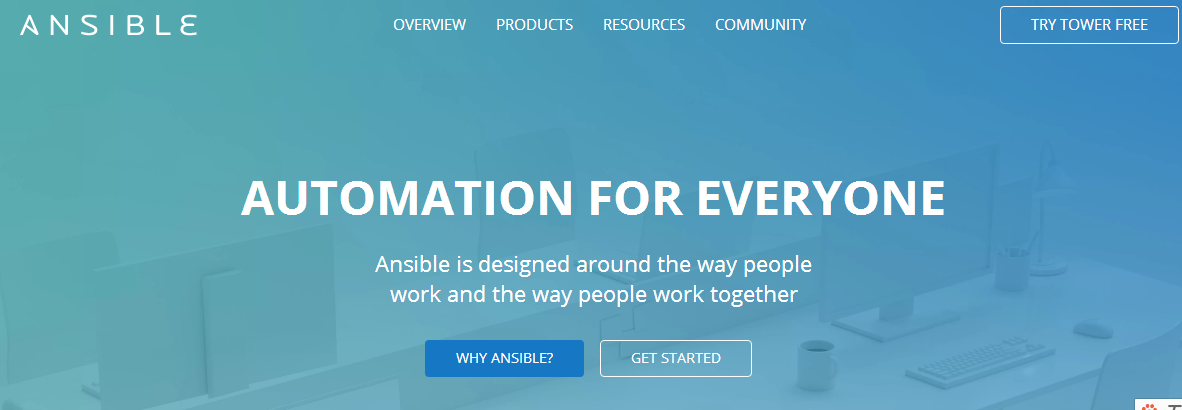
## 21.1 Ansible概述和运行机制

### 21.1.1 Ansible概述

Ansible是一款为类Unix系统开发的自由开源的配置和自动化工具。它用Python写成，类似于saltstack和Puppet，但是有一个不同和优点是我们不需要在节点中安装任何客户端。它使用SSH来和节点进行通信。Ansible基于 Python paramiko 开发，分布式，无需客户端，轻量级，配置语法使用 YAML 及 Jinja2模板语言，更强的远程命令执行操作



官方网站：https://www.ansible.com/



IT业界励志事件：

2015年10月，红帽（Red Hat）宣布收购软件开发公司 Ansible，消息称此次收购耗资逾 1亿美元，也有消息称接近 1.5亿美元。

Ansible 成立于 2013年，总部设在北卡罗来纳州达勒姆，联合创始人 aïd Ziouani 和高级副总裁 Todd Barr 都是红帽的老员工。Ansible 旗下的开源软件 Ansible 十分流行。这家公司还提供 Tower 软件和咨询服务，这个款软件能使开发者轻松地建立和管理规模化应用程序的 IT 基础架构。

ansiblle具有如下特点：

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、主从集中化管理；

4、配置简单、功能强大、扩展性强；

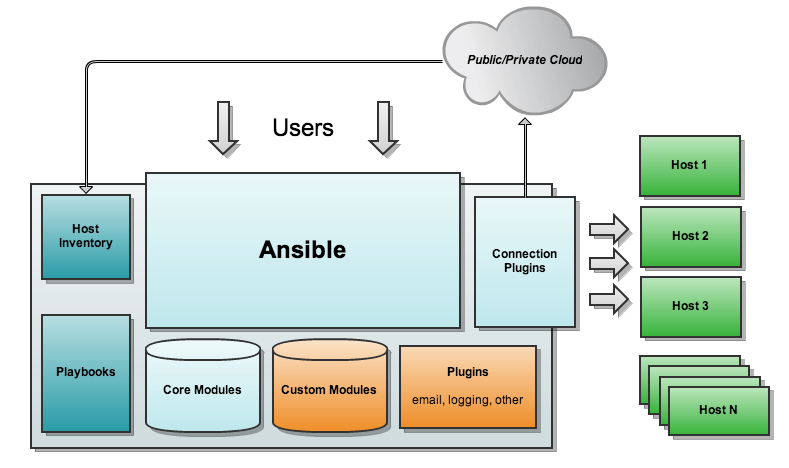
5、支持API及自定义模块，可通过Python轻松扩展；

6、通过Playbooks来定制强大的配置、状态管理

7、对云计算平台、大数据都有很好的支持；

### 21.1.2 Ansible 工作机制

Ansible 在管理节点将 Ansible 模块通过 SSH 协议推送到被管理端执行，执行完之后自动删除, 可以使用 SVN 等来管理自定义模块及编排。



由上面的图可以看到 Ansible 的组成由 5 个部分组成：

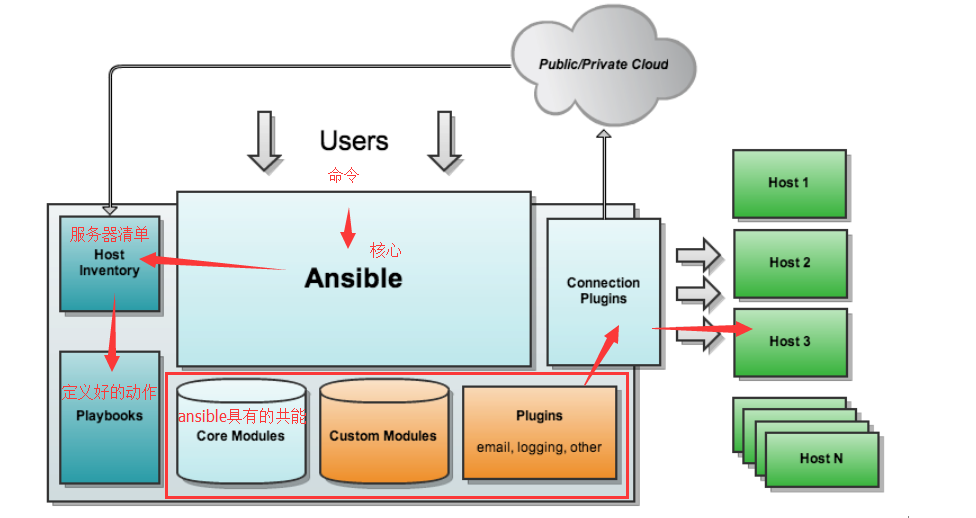
Ansible ：     ansible核心

Modules ：    包括 Ansible 自带的核心模块及自定义模块

Plugins ：      完成模块功能的补充，包括连接插件、邮件插件等

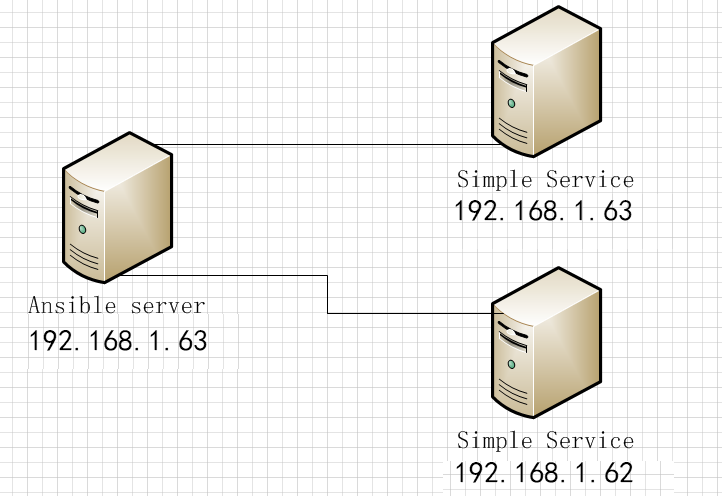
Playbooks ：   剧本；定义 Ansible 多任务配置文件，由Ansible 自动执行

Inventory ：    定义 Ansible 管理主机的清单 [ˈɪnvəntri] 清单



## 21.2 实战-安装并配置Ansible管理两个节点

实验环境：



Ansible server 192.168.1.63

Simple service 192.168.1.63

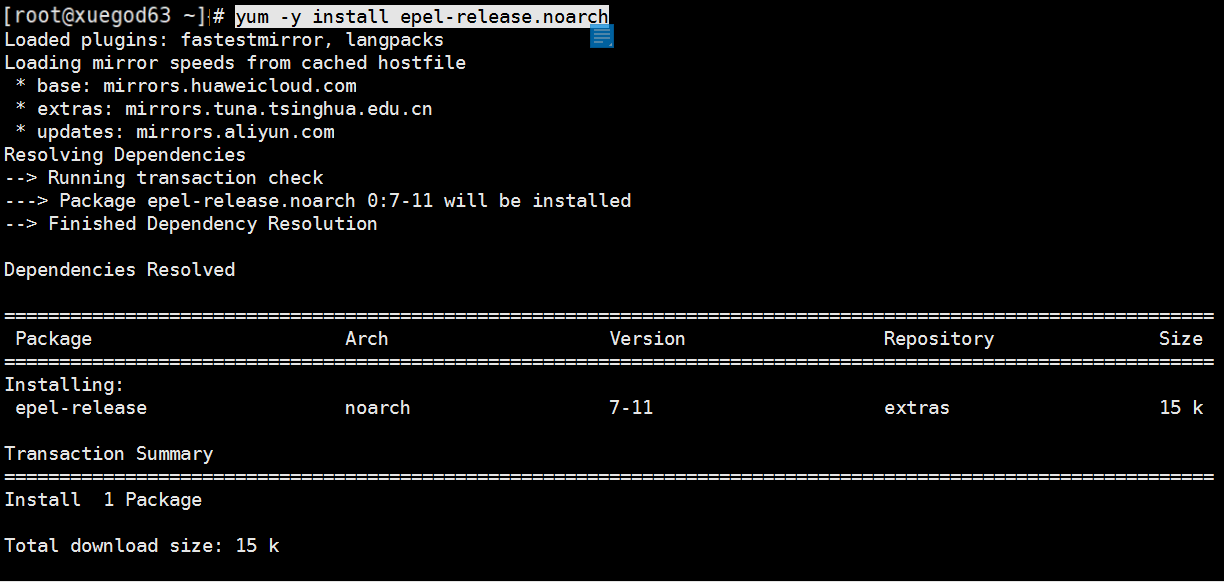
Simple service 192.168.1.62

在server 192.168.1.63安装ansible

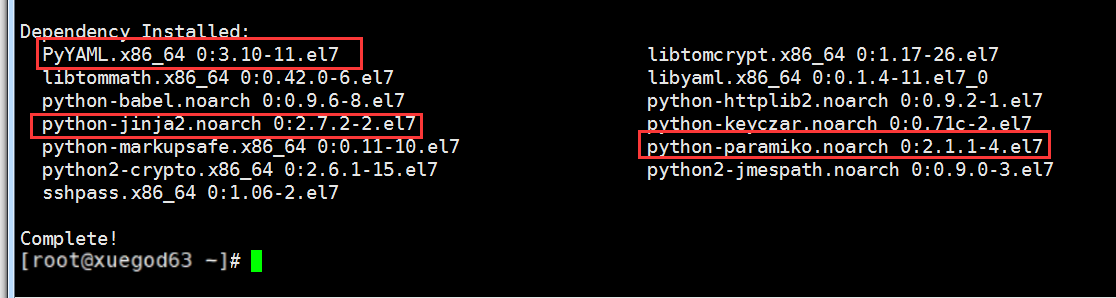
1、 设置EPEL仓库

Ansible仓库默认不在yum仓库中，因此我们需要使用下面的命令启用epel仓库。

yum -y install epel-release.noarch

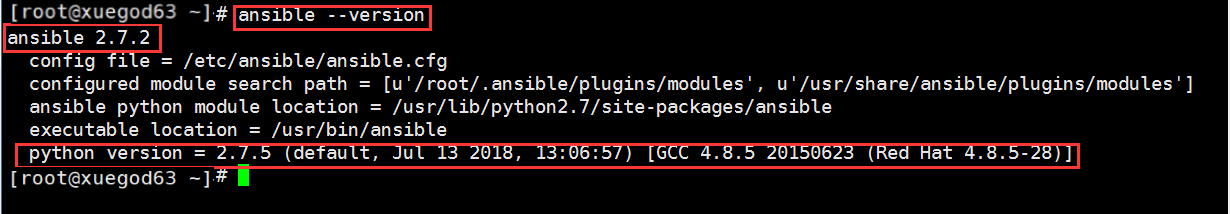


yum install ansible -y



扩展 yum -y install zabbix salt-stack docker

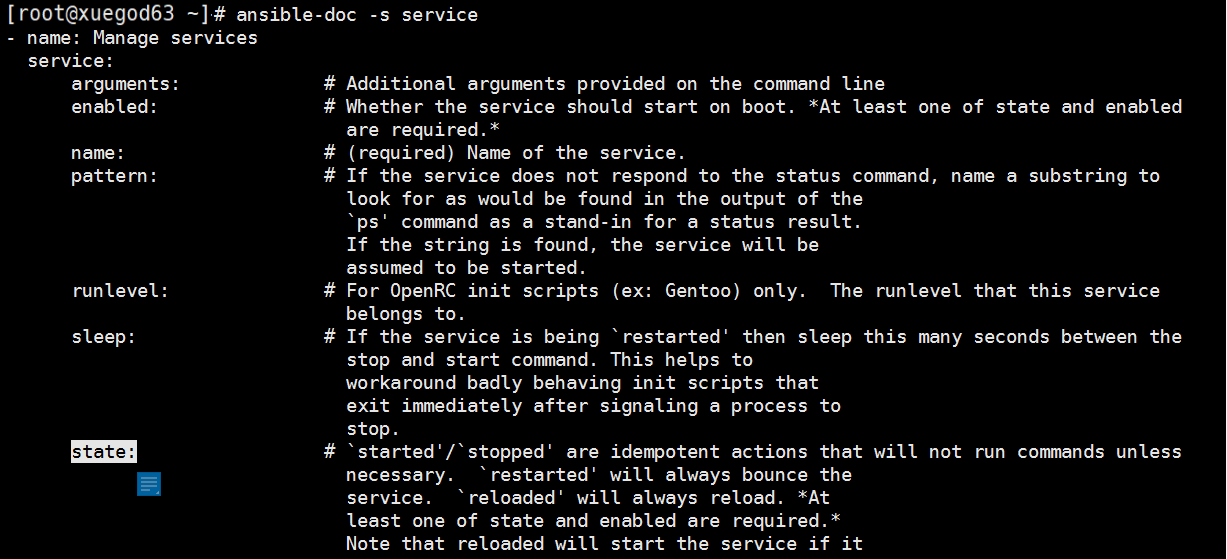
ansible --version



ansible命令参数

我们首先写一个ansible的命令，查看ansible的文档

ansible-doc -s service



anisble命令语法： ansible [-i 主机文件] [-f 批次] [组名] [-m 模块名称] [-a 模块参数]

ansible object method argument

环境变量 对象 方法 调用方法需要的参数

-v,-verbose #  详细模式，如果命令执行成功，输出详细的结果 (-vv –vvv -vvvv)

 -i PATH, -inventory=PATH      #  指定 host 文件的路径，默认是在 /etc/ansible/hosts

inventory [ˈɪnvəntri] 库存

 -f NUM,-forks=NUM     # NUM 是指定一个整数，默认是 5 ，指定 fork 开启同步进程的个数。

 -m NAME,-module-name=NAME    #   指定使用的 module 名称，默认使用 command模块

 -a,MODULE\_ARGS   #指定 module 模块的参数

 -k,-ask-pass          #提示输入 ssh 的密码，而不是使用基于 ssh 的密钥认证

 -sudo        # 指定使用 sudo 获得 root 权限

 -K,-ask-sudo-pass             #提示输入 sudo 密码，与 -sudo 一起使用

 -u USERNAME,-user=USERNAME          # 指定移动端的执行用户

 -C,-check             #测试此命令执行会改变什么内容，不会真正的去执行

ansible-doc详细参数：

ansible-doc -l #列出所有的模块列表

ansible-doc -s 模块名 #查看指定模块的参数 -s, snippet [ˈsnɪpɪt] 片断

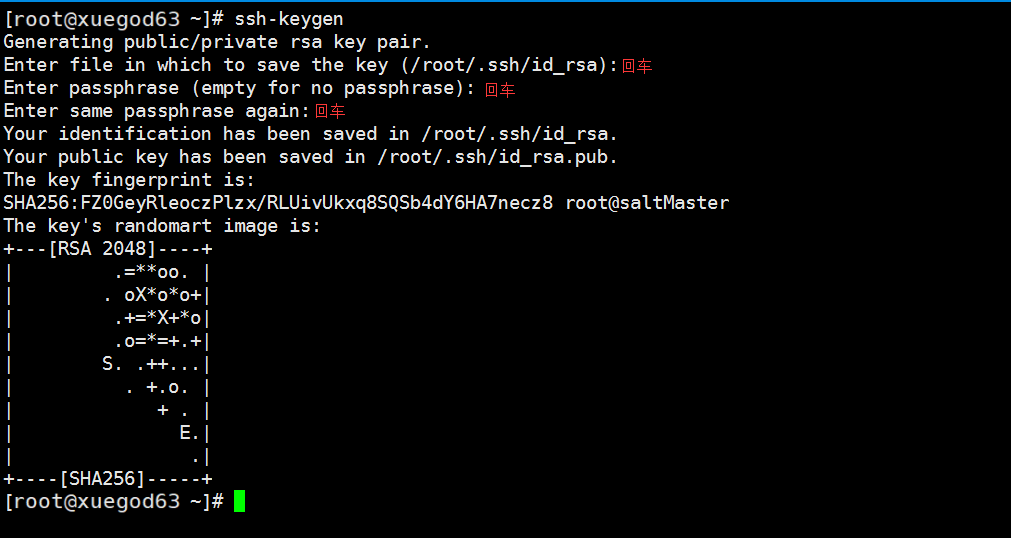
基于ssh密钥来访问定义主机清单

一般来说，使用明文密码不安全，所以增加主机无密码访问。

在Ansible服务端生成密钥，并且复制公钥到节点中。

生成秘钥

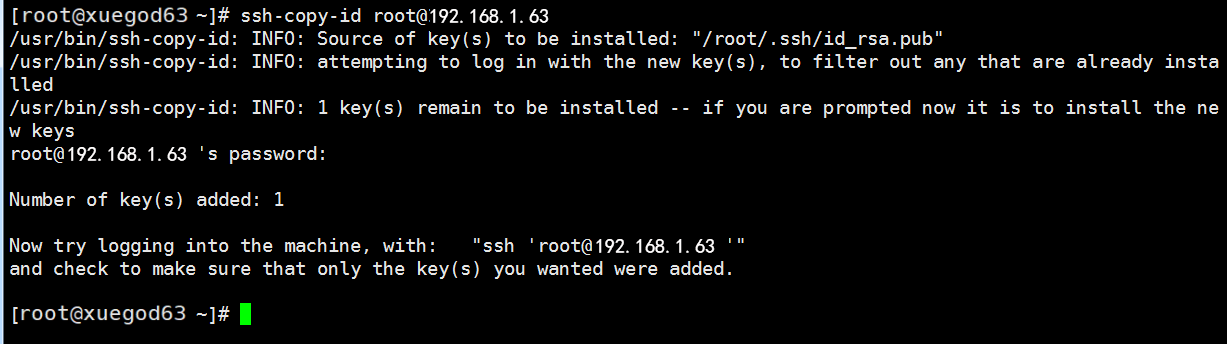
ssh-keygen



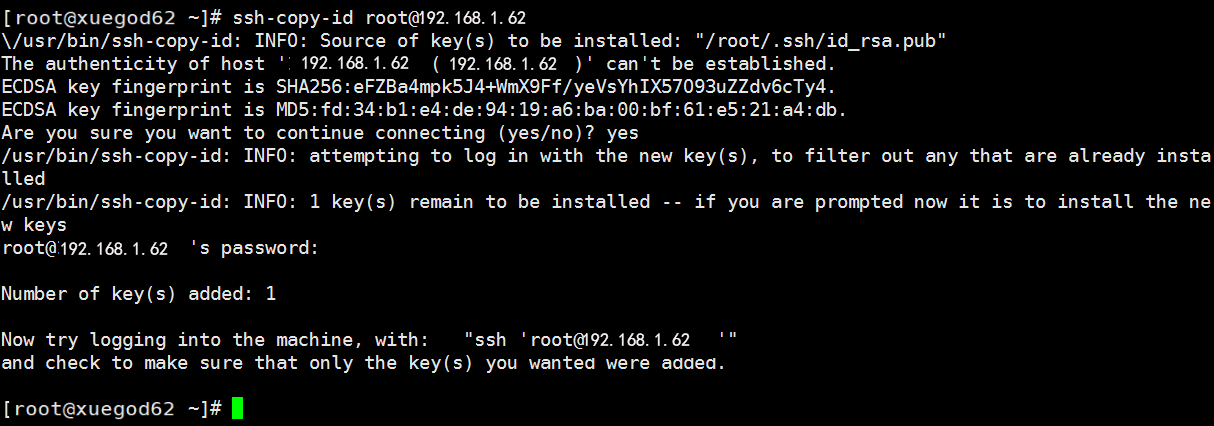
复制到客户机

ssh-copy-id

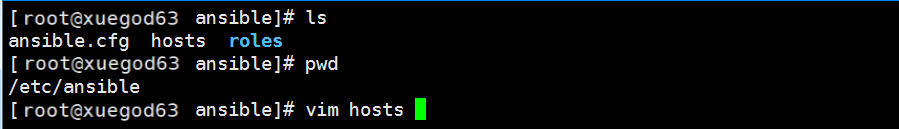
ssh-copy-id root@192.168.1.63

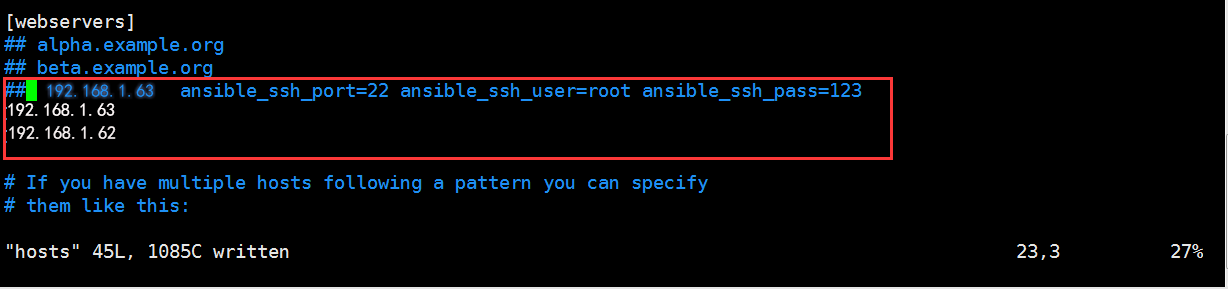


ssh-copy-id root@192.168.1.62



然后修改hosts配置





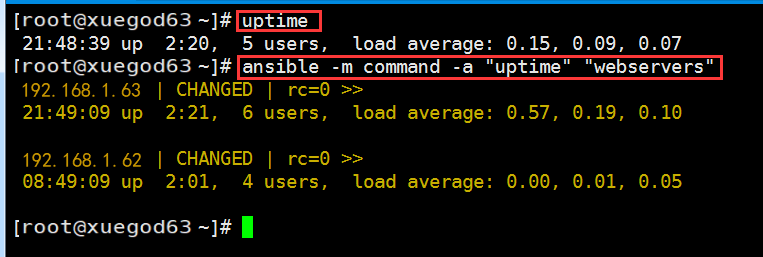
注意：两个#注释，只是为了和上面一致

进行测试

ansible 'webservers' -m ping



例2：检查Ansible节点的运行时间（uptime）



[root@xuegod63~]# uptime

21:48:39 up 2:20, 5 users, load average: 0.15, 0.09, 0.07

[root@xuegod63~]# ansible -m command -a "uptime" "webservers"

192.168.1.63 | CHANGED | rc=0 >>

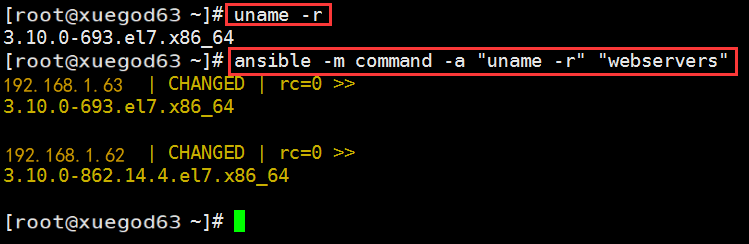
21:49:09 up 2:21, 6 users, load average: 0.57, 0.19, 0.10

192.168.1.62 | CHANGED | rc=0 >>

08:49:09 up 2:01, 4 users, load average: 0.00, 0.01, 0.05

[root@xuegod63~]#

例3：检查节点的内核版本



例4：给节点增加用户

[root@xuegod63~]# useradd while123

[root@xuegod63~]# grep while123 /etc/passwd

while123:x:1001:1001::/home/while123:/bin/bash

[root@xuegod63~]# ansible -m command -a "useradd while123" "webservers"

192.168.1.63 | CHANGED | rc=0 >>

192.168.1.62 | CHANGED | rc=0 >>

[root@xuegod63~]# ansible -m command -a "grep while123 /etc/passwd" "webservers"

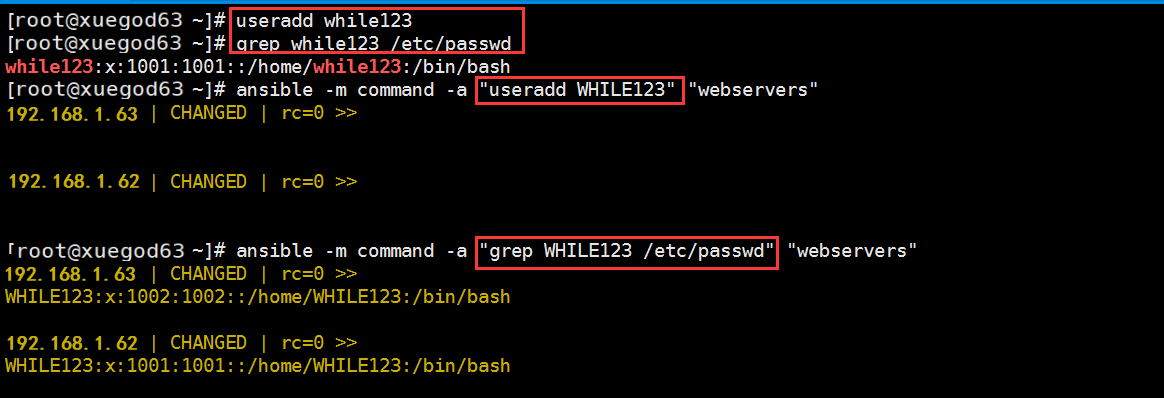
192.168.1.63 | CHANGED | rc=0 >>

while123:x:1002:1002::/home/while123:/bin/bash

192.168.1.62 | CHANGED | rc=0 >>

while123:x:1001:1001::/home/while123:/bin/bash

[root@xuegod63~]#



## 21.3 Ansible常见模块高级使用方法

Ansible常用模块

Ansible 已经给开发者定义了好多可用的模块，我们除了拓展之外，大部分模块可以直接使用

1、3个远程命令模块的区别

（1）、command模块为ansible默认模块，不指定-m参数时，使用的就是command模块； comand模块比较简单，常见的命令都可以使用，但其命令的执行不是通过shell执行的，所以，像这些 "<", ">", "|", and "&"操作都不可以，当然，也就不支持管道； 缺点：不支持管道，没法批量执行命令；

（2）、shell模块：使用shell模块，在远程命令通过/bin/sh来执行；所以，我们在终端输入的各种命令方式，都可以使用。

例1：运行free -m 命令

[root@xuegod63~]# ansible webservers -m shell -a "free -m"

192.168.1.63 | CHANGED | rc=0 >>

total used free shared buff/cache available

Mem: 1823 781 82 12 959 783

Swap: 2047 6 2041

192.168.1.62 | CHANGED | rc=0 >>

total used free shared buff/cache available

Mem: 974 686 60 6 227 63

Swap: 2047 383 1664

对shell模块的使用可以分成两块：

1. 如果待执行的语句少，可以直接写在一句话中,如上例。
2. 如果在远程待执行的语句比较多，可写成一个脚本，通过copy模块传到远端，然后再执行；但这样就又涉及到两次ansible调用；对于这种需求，ansible已经为我们考虑到了，script模块就是干这事的；

3）、scripts模块

使用scripts模块可以在本地写一个脚本，在远程服务器上执行：

[root@xuegod63~]# vim /etc/ansible/net.sh

#!/bin/bash

date

hostname

[root@xuegod63~]# ansible webservers -m script -a "/etc/ansible/net.sh"

192.168.1.63 | CHANGED => {

"changed": true,

"rc": 0,

"stderr": "Shared connection to 192.168.1.63 closed.\r\n",

"stderr\_lines": [

"Shared connection to 192.168.1.63 closed."

],

"stdout": "Mon Dec 3 22:08:55 CST 2018\r\nsaltMaster\r\n",

"stdout\_lines": [

"Mon Dec 3 22:08:55 CST 2018",

"saltMaster"

]

}

192.168.1.62 | CHANGED => {

"changed": true,

"rc": 0,

"stderr": "Shared connection to 192.168.1.62 closed.\r\n",

"stderr\_lines": [

"Shared connection to 192.168.1.62 closed."

],

"stdout": "Mon Dec 3 09:09:06 EST 2018\r\nlocalhost.localdomain\r\n",

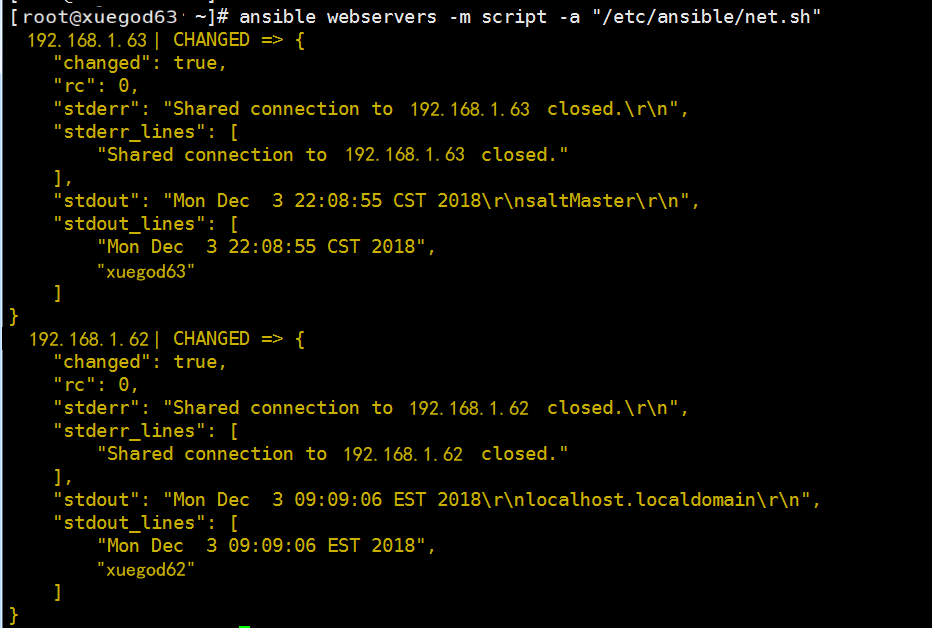
"stdout\_lines": [

"Mon Dec 3 09:09:06 EST 2018",

"localhost.localdomain"

]

}



2、copy模块:实现主控端向目标主机拷贝文件，类似scp功能

例1：把ansible主机上的/etc/hosts文件复制到主机组中机器的/tmp目录下

[root@xuegod63/]# ansible webservers -m copy -a "src=/etc/hosts dest=/tmp/ owner=root group=root mode=0755"

192.168.1.63 | CHANGED => {

"changed": true,

"checksum": "7335999eb54c15c67566186bdfc46f64e0d5a1aa",

"dest": "/tmp/hosts",

"gid": 0,

"group": "root",

"md5sum": "54fb6627dbaa37721048e4549db3224d",

"mode": "0755",

"owner": "root",

"size": 158,

"src": "/root/.ansible/tmp/ansible-tmp-1543846330.63-199478885009462/source",

"state": "file",

"uid": 0

}

192.168.1.62 | CHANGED => {

"changed": true,

"checksum": "7335999eb54c15c67566186bdfc46f64e0d5a1aa",

"dest": "/tmp/hosts",

"gid": 0,

"group": "root",

"md5sum": "54fb6627dbaa37721048e4549db3224d",

"mode": "0755",

"owner": "root",

"secontext": "unconfined\_u:object\_r:admin\_home\_t:s0",

"size": 158,

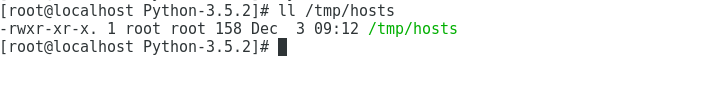
"src": "/root/.ansible/tmp/ansible-tmp-1543846330.65-125711974784987/source",

"state": "file",

"uid": 0

}

我们看一眼



3、file模块设置文件属性。

例如：

[root@xuegod63 ~]# ansible -i /etc/ansible/hosts webservers -m file -a "path=/tmp/hosts mode=0777"

验证:

[root@xuegod63 ~]# ll /tmp/hosts

-rwxrwxrwx 1 root root 158 11月 14 22:39 /tmp/hosts

4、stat模块获取远程文件信息

[root@xuegod63 ~]# ansible -i /etc/ansible/hosts webservers -m stat -a "path=/tmp/hosts"

5、get\_url模块实现远程主机下载指定url到本地，支持sha256sum文件校验。

例如：下载epel-release-latest-7.noarch.rpm到主机清单中的/tmp/目录下

[root@xuegod63 ~]# ansible -i /etc/ansible/hosts webservers -m get\_url -a "url=https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm dest=/tmp/ mode=0440 force=yes"

注：url=https://xxx 的等号=前后不能有空格

扩展:查看force=yes的作用

[root@xuegod63 ~]# ansible-doc -s get\_url #在弹出的信息中找到force

如果force=yes，当下载文件时，如果所下的内容和原目录下的文件内容不一样，则替换原文件，如果一样，就不下载了。

如果为“否”，则仅在目标不存在时才下载文件。 一般来说，只有小型本地文件才应该为“是”。 在0.6之前，该模块表现为默认为“是”。

查看下载的文件：

[root@xuegod63 ~]# ll /tmp/epel-release-latest-7.noarch.rpm

-r--r----- 1 root root 15080 11月 14 22:43 /tmp/epel-release-latest-7.noarch.rpm

6、yum模块linux平台软件包管理。

yum模块可以提供的status状态： latest ，present，installed #这3个代表安装；removed, absent #后面2个是卸载

例子：安装httpd软件

[root@xuegod63 ~]# ansible -i /etc/ansible/hosts webservers -m yum -a "name=httpd state=latest"

7、cron模块远程主机crontab配置。

例如：增加每30分钟执行ls /tmp

[root@xuegod63 ~]# ansible -i /etc/ansible/hosts webservers -m cron -a "name='list dir' minute='\*/30' job='ls /tmp'"

在xuegod63上查看

[root@xuegod63 ~]# crontab -l

#Ansible: list dir

\*/30 \* \* \* \* ls /tmp

8、service模块远程主机系统服务管理。

service模块常用参数：

（1）、name参数：此参数用于指定需要操作的服务名称，比如 nginx，httpd。

（2）、state参数：此参数用于指定服务的状态，比如，我们想要启动远程主机中的httpd，则可以将 state 的值设置为 started；如果想要停止远程主机中的服务，则可以将 state 的值设置为 stopped。此参数的可用值有 started、stopped、restarted（重启）、reloaded。

enabled参数：此参数用于指定是否将服务设置为开机 启动项，设置为 yes 表示将对应服务设置为开机启动，设置为 no 表示不会开机启动。

注：想使用service模块启动服务，被启动的服务，必须可以使用service 命令启动或关闭

例如：远程启动Apache服务

[root@xuegod63 ~]# ansible -i /etc/ansible/hosts webservers -m service -a "name=httpd state=restarted"

9、sysctl模块远程主机sysctl配置。

例：开启路由转发功能

[root@xuegod63 ~]# ansible -i /etc/ansible/hosts webservers -m sysctl -a "name=net.ipv4.ip\_forward value=1 reload=yes"

验证：

[root@xuegod63 ~]# cat /proc/sys/net/ipv4/ip\_forward

1

10、user模块远程主机用户管理

例如：

[root@xuegod63 ~]# ansible -i /etc/ansible/hosts webservers -m user -a "name=xuegod6 state=present"

# present [ˈpreznt] 目前

验证：

[root@xuegod63 ~]# id xuegod6

uid=1001(xuegod6) gid=1001(xuegod6) 组=1001(xuegod6)

playbook lamp 安装 启服务 测试文件

l a m p 多个playbook

yaml tasks 任务目录

总结：

21.1 Ansible概述和运行机制

21.2 实战-安装并配置Ansible管理两个节点

21.3 Ansible常见模块高级使用方法