

day04

day04

ansible

环境准备

配置ansible管理环境

ansible管理

adhoc临时命令

ansible模块

command模块

shell模块

script模块

file模块

copy模块

fetch模块

lineinfile模块

replace模块

文件操作综合练习

user模块

group模块

ansible

- 批量管理服务器的工具
- 2015年被红帽公司收购
- 使用Python语言编写的
- 基于ssh进行管理，所以不需要在被管端安装任何软件
- ansible在管理远程主机的时候，主要是通过各种模块进行操作的

环境准备

主机名	IP地址	角色
Control	192.168.4.253	控制节点 (manager)
Node1	192.168.4.11	被控制节点 (test)
Node2	192.168.4.12	被控制节点 (proxy)
Node3	192.168.4.13	被控制节点 (web1)
Node4	192.168.4.14	被控制节点 (web2)
Node5	192.168.4.15	被控制节点 (database)

- 6台主机，需要配置主机名、IP地址、YUM。关闭SELINUX和防火墙
- control节点要求：
 - 配置名称解析，能够通过名字访问所有节点
 - 配置可以通过ssh到所有节点免密登陆

- 拷贝/linux-soft/2/ansible_soft.tar.gz到control，并解压安装

配置名称解析

```
[root@control ~]# echo -e "192.168.4.253\tcontrol" >> /etc/hosts
```

```
[root@control ~]# for i in {1..5}
> do
> echo -e "192.168.4.1$i\tnode$i" >> /etc/hosts
> done
```

```
[root@control ~]# tail -6 /etc/hosts
```

```
192.168.4.253    control
192.168.4.11     node1
192.168.4.12     node2
192.168.4.13     node3
192.168.4.14     node4
192.168.4.15     node5
```

配置免密登陆

```
[root@control ~]# ssh-keygen    # 三个问题都直接回车，使用默认值
```

```
[root@control ~]# for i in node{1..5}    # 回答yes和密码
> do
> ssh-copy-id $i
> done
```

装包

```
[root@zzgrhel8 ~]# scp /linux-soft/2/ansible_soft.tar.gz 192.168.4.253:/root
[root@control ~]# yum install -y tar
[root@control ~]# tar xf ansible_soft.tar.gz
```

```
[root@control ~]# cd ansible_soft/  
[root@control ansible_soft]# yum install -y *.rpm
```

配置ansible管理环境

- 因为要管理的远程主机可能不一样。所以具有相同管理方式的配置放到一个目录下。

创建ansible工作目录，目录名自己定义，不是固定的。

```
[root@control ~]# mkdir ansible
```

```
[root@control ~]# cd ansible
```

创建配置文件。默认的配置文

件是/etc/ansible/ansible.cfg，但是一般不使用它，而是在工作目录下创建自己的配置文件

```
[root@control ansible]# vim ansible.cfg    # 文件名必须是ansible.cfg
```

```
[defaults]
```

```
inventory = hosts    # 管理的主机，配置在当前目录的hosts文件中，hosts名是自定义的。=号两边空格可有可无。
```

创建主机清单文件。写在[]里的是组名，[]下面的是组内的主机名

```
[root@control ansible]# vim hosts
```

```
[test]
```

```
node1
```

```
[proxy]
```

node2

[webserver]

node[3:4] # node3和node4的简化写法，表示从3到4

[database]

node5

cluster是组名，自定义的；:children是固定写法，表示下面的组名是cluster的子组。

[cluster:children]

webserver

database

查看被管理的所有的主机。注意，一定在工作目录下执行命令。

[root@control ansible]# ansible all --list-hosts

hosts (5):

node1

node2

node3

node4

node5

查看webserver组中的所有的主机

[root@control ansible]# ansible webserver --list-hosts

hosts (2):

node3

node4

ansible管理

- ansible进行远程管理的两个方法：
 - adhoc临时命令。就是在命令行上执行管理命令。
 - playbook剧本。把管理任务用特定格式写到文件中。
- 无论哪种方式，都是通过模块加参数进行管理。

adhoc临时命令

- 语法：

```
ansible 主机或组列表 -m 模块 -a "参数" # -a是可选的
```

- 测试到远程主机的连通性

```
[root@control ansible]# ansible all -m ping
```

ansible模块

- 模块基本信息查看

```
# 列出ansible的所有模块数量
[root@control ansible]# ansible-doc -l | wc -l
2834

# 列出ansible的所有模块
[root@control ansible]# ansible-doc -l

# 查看与yum相关的模块
[root@control ansible]# ansible-doc -l | grep yum

# 查看yum模块的使用说明，主要查看下方的EXAMPLE示例
[root@control ansible]# ansible-doc yum
```

- 学习模块，主要知道实现某种功能，需要哪个模块。
- 模块的使用方式都一样。主要是查看该模块有哪些参数。

command模块

- ansible默认模块，用于在远程主机上执行任意命令
- command不支持shell特性，如管道、重定向。

```
# 在所有被管主机上创建目录/tmp/demo
[root@control ansible]# ansible all -a "mkdir
/tmp/demo"

# 查看node1的ip地址
[root@control ansible]# ansible node1 -a "ip a s"
[root@control ansible]# ansible node1 -a "ip a s |
head"    # 报错
```

shell模块

- 与command模块类似，但是支持shell特性，如管道、重定向。

```
# 查看node1的ip地址，只显示前10行  
[root@control ansible]# ansible node1 -m shell -a "ip a  
s | head"
```

script模块

- 用于在远程主机上执行脚本

```
# 在控制端创建脚本即可  
[root@control ansible]# vim test.sh  
#!/bin/bash  
  
yum install -y httpd  
systemctl start httpd  
  
# 在test组的主机上执行脚本  
[root@control ansible]# ansible test -m script -a  
"test.sh"
```

file模块

- 可以创建文件、目录、链接等，还可以修改权限、属性等
- 常用的选项：
 - path: 指定文件路径

- owner: 设置文件所有者
- group: 设置文件所属组
- state: 状态。touch表示创建文件，directory表示创建目录，link表示创建软链接，absent表示删除
- mode: 设置权限
- src: source的简写，源
- dest: destination的简写，目标

查看使用帮助

```
[root@control ansible]# ansible-doc file
```

... ..

EXAMPLES:

- name: Change file ownership, group and permissions #
忽略

file: # 模块名。以下是它的
各种参数

path: /etc/foo.conf # 要修改的文件的路径

owner: foo # 文件所有者

group: foo # 文件的所有组

mode: '0644' # 权限

... ..

根据上面的example，-m file -a的内容就是doc中把各参数的冒号换成=号

在test主机上创建/tmp/file.txt

```
[root@control ansible]# ansible test -m file -a
```

"path=/tmp/file.txt state=touch" # touch是指如果文件不存在，则创建

在test主机上创建/tmp/demo目录

```
[root@control ansible]# ansible test -m file -a
"path=/tmp/demo state=directory"

# 将test主机上/tmp/file.txt的属主改为sshd，属组改为
adm，权限改为0777
[root@control ansible]# ansible test -m file -a
"path=/tmp/file.txt owner=sshd group=adm mode='0777'"
[root@control ansible]# ansible test -a "ls -l
/tmp/file.txt"

# 删除test主机上/tmp/file.txt
[root@control ansible]# ansible test -m file -a
"path=/tmp/file.txt state=absent"    # absent英文缺席
的、不存在的

# 删除test主机上/tmp/demo
[root@control ansible]# ansible test -m file -a
"path=/tmp/demo state=absent"

# 在test主机上创建/etc/hosts的软链接，目标
是/tmp/hosts.txt
[root@control ansible]# ansible test -m file -a
"src=/etc/hosts dest=/tmp/hosts.txt state=link"
```

copy模块

- 用于将文件从控制端拷贝到被控端
- 常用选项：
 - src：源。控制端的文件路径
 - dest：目标。被控制端的文件路径

- content: 内容。需要写到文件中的内容

```
[root@control ansible]# echo "AAA" > a3.txt
# 将a3.txt拷贝到test主机的/root/
[root@control ansible]# ansible test -m copy -a
"src=a3.txt dest=/root/"

# 在目标主机上创建/tmp/mytest.txt，内容是Hello World
[root@control ansible]# ansible test -m copy -a
"content='Hello World' dest=/tmp/mytest.txt"
```

fetch模块

- 与copy模块相反，copy是上传，fetch是下载
- 常用选项：
 - src: 源。被控制端的文件路径
 - dest: 目标。控制端的文件路径

```
# 将test主机上的/etc/hostname下载到本地用户的家目录下
[root@control ansible]# ansible test -m fetch -a
"src=/etc/hostname dest=~/"
[root@control ansible]# ls ~/node1/etc/    # node1是test
组中的主机
hostname
```

lineinfile模块

- 用于确保存目标文件中有某一行内容
- 常用选项:

- path: 待修改的文件路径
- line: 写入文件的一行内容
- regexp: 正则表达式, 用于查找文件中的内容

```
# test组中的主机, /etc/issue中一定要有一行Hello World。  
如果该行不存在, 则默认添加到文件结尾
```

```
[root@control ansible]# ansible test -m lineinfile -a  
"path=/etc/issue line='Hello World'"
```

```
# test组中的主机, 把/etc/issue中有Hello的行, 替换成chi  
le ma
```

```
[root@control ansible]# ansible test -m lineinfile -a  
"path=/etc/issue line='chi le ma' regexp='Hello'"
```

replace模块

- lineinfile会替换一行, replace可以替换关键词
- 常用选项:
 - path: 待修改的文件路径
 - replace: 将正则表达式查到的内容, 替换成replace的内容
 - regexp: 正则表达式, 用于查找文件中的内容

```
# 把test组中主机上/etc/issue文件中的chi, 替换成he  
[root@control ansible]# ansible test -m replace -a  
"path=/etc/issue regexp='chi' replace='he'"
```

文件操作综合练习

- 所有操作均对test组中的主机生效
- 在目标主机上创建/tmp/mydemo目录，属主和属组都是adm，权限为0777
- 将控制端的/etc/hosts文件上传到目标主机的/tmp/mydemo目录中，属主和属组都是adm，权限为0600
- 替换目标主机/tmp/mydemo/hosts文件中的node5为server5
- 将目标主机/tmp/mydemo/hosts文件下载到控制端的当前目录

在目标主机上创建/tmp/mydemo目录，属主和属组都是adm，权限为0777

```
[root@control ansible]# ansible test -m file -a  
"path=/tmp/mydemo owner=adm group=adm mode='0777'  
state=directory"
```

将控制端的/etc/hosts文件上传到目标主机的/tmp/mydemo目录中，属主和属组都是adm，权限为0600

```
[root@control ansible]# ansible test -m copy -a  
"src=/etc/hosts dest=/tmp/mydemo owner=adm group=adm  
mode='0600'"
```

替换目标主机/tmp/mydemo/hosts文件中的node5为server5

```
[root@control ansible]# ansible test -m replace -a  
"path=/tmp/mydemo/hosts regexp='node5'  
replace='server5'"
```

将目标主机/tmp/mydemo/hosts文件下载到控制端的当前目录。文件将会保存到控制端当前目录的node1/tmp/mydemo/

```
[root@control ansible]# ansible test -m fetch -a  
"src=/tmp/mydemo/hosts dest=."
```

user模块

- 实现linux用户管理
- 常用选项：
 - name: 待创建的用户名
 - uid: 用户ID
 - group: 设置主组

- groups: 设置附加组
- home: 设置家目录
- password: 设置用户密码
- state: 状态。present表示创建，它是默认选项。absent表示删除
- remove: 删除家目录、邮箱等。值为yes或true都可以。

```
# 在test组中的主机上，创建tom用户
```

```
[root@control ansible]# ansible test -m user -a  
"name=tom"
```

```
# 在test组中的主机上，创建jerry用户。设置其uid为1010，  
主组是adm，附加组是daemon和root，家目录是/home/jerry
```

```
[root@control ansible]# ansible test -m user -a  
"name=jerry uid=1010 group=adm groups=daemon,root  
home=/home/jerry"
```

```
# 设置tom的密码是123456
```

```
# {{}}是固定格式，表示执行命令。password_hash是函数，  
sha512是加密算法，则password_hash函数将会把123456通过  
sha512加密变成tom的密码
```

```
[root@control ansible]# ansible test -m user -a  
"name=tom password=  
{{ '123456' | password_hash('sha512') }}"
```

```
# 删除tom用户，不删除家目录
```

```
[root@control ansible]# ansible test -m user -a  
"name=tom state=absent"
```

```
# 删除jerry用户，同时删除家目录
```

```
[root@control ansible]# ansible test -m user -a  
"name=jerry state=absent remove=yes"
```

group模块

- 创建、删除组
- 常用选项：
 - name: 待创建的组名
 - gid: 组的ID号
 - state: present表示创建，它是默认选项。absent表示删除

```
# 在test组中的主机上创建名为devops的组  
[root@control ansible]# ansible test -m group -a  
"name=devops"
```

```
# 在test组中的主机上删除名为devops的组  
[root@control ansible]# ansible test -m group -a  
"name=devops state=absent"
```