一 ansible

１，静态主机

[root@ansible ansible]# vim /etc/ansible/ansible.cfg

14 inventory = /etc/ansible/hosts　　　　#打开注释

61 host\_key\_checking = False　　　#打开注释，关闭key检查

[root@ansible ansible]# vim hosts

[web]

web1 ansible\_ssh\_user="root" ansible\_ssh\_pass="123456" ansible\_ssh\_port=22

web2 ansible\_ssh\_pass="123456"

#web[01:10]

#[web:var]

#ansible\_ssh\_user="root"

#ansible\_ssh\_pass="123456"

#ansible\_ssh\_port=22

[db]

db1 ansible\_ssh\_user="root" ansible\_ssh\_pass="123456"

db2 ansible\_ssh\_user="root" ansible\_ssh\_pass="123456"

[other]

cache ansible\_ssh\_pass="123456"

[app:children]

web

db

[root@ansible ansible]# ansible web -m ping -k

-k:交互输入密码

2，动态主机

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

－－部署key

[root@ansible .ssh]# ssh-keygen -f "/root/.ssh/id\_rsa" -N ""

[root@ansible .ssh]# bb=$(<id\_rsa.pub)

[root@ansible .ssh]# ansible all -m authorized\_key -a "user=root exclusive=true manage\_dir=true key='$(< /root/.ssh/id\_rsa.pub)'"

#exclusive=true :强制执行命令

#manage\_dir=true　：.ssh没有时新建目录

#key=公钥 :设置公钥的值

－－模块

[root@ansible ~]# ansible-doc -l

#command 默认模块

执行命令不开启bash

#shell

#raw

#script

-ansible web -m script -a “myscript”

注意：该脚本包含但是不限于shell脚本，只要指定sha-bang解释器的脚本都可以

#copy

#lineinfile / replace

-path:目的文件

-regexp：正则表达式

-line：修改后的样子

-replace：匹配修改的样子

例子：

[root@ansible ~]# ansible cache -m lineinfile -a 'path=/etc/sysconfig/network-scripts/ifcfg-eth0 regexp="^ONBOOT" line="ONBOOT=\"no\""'

[root@ansible ~]# ansible cache -m replace -a 'path=/etc/sysconfig/network-scripts/ifcfg-eth0 regexp="^(ONBOOT=).\*" replace="\1\"yes\""'

#yum

-name：安装的服务名称

-state :动作 install=> installed ，remove => removed

#service

-name：必选项，服务名称

-enabled：是否开机自启　yes|no

-sleep：执行restarted,会在stop和start之间沉睡几秒钟

-state：对当前服务启动，停止，重启，重新加载等操作(started, stopped, restarted, reloaded)

２，playbook

[root@ansible ~]# vim aa

{

"username": "mm"

}

[root@ansible ~]# ansible-playbook -e '@aa' useradd.yml