ansible 01

1 ansible 192.168.1.10 管理机器

2 web1 192.168.1.11 托管机器

3 web2 192.168.1.12 托管机器

4 db1 192.168.1.21 托管机器

5 db2 192.168.1.22 托管机器

6 cache 192.168.1.33 托管机器

在 ansible 管理机器上,配置源

yum install -y ansible

安装完成以后执行，没有报错，正确显示版本即可

ansible --version

ansible 的配置文件是 ansible.cfg

ansible.cfg 的查找顺序是

1 ANSIBLE\_CONFIG 变量定义的配置文件

2 当前目录下的 ./ansible.cfg 文件

3 前用户家目录下 ~/ansible.cfg 文件

4 /etc/ansible/ansible.cfg 文件

ansible.cfg 中 inventony 指定主机分组文件的路径和地址，默认分组文件 hosts

hosts 的配置

[web]

web[1:2]

[db]

db1

db2

[app:children] # 指定子组

web

db

[app:vars]

ansible\_ssh\_user="root"

ansible\_ssh\_pass="123456"

[other]

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

动态主机

ansible 的inventony 文件可以是静态也可以是脚本(输出格式 json)

修改 ansible.cfg

inventory = urscript

一个shell 脚本样例

#!/bin/bash

echo '

{

"web" : ["web1", "web2"],

"db" : ["db1", "db2"],

"other" : ["cache"]

}'

ansible 命令基础

ansible 主机分组 -m 模块 -a '命令和参数'

创建密钥对 id\_rsa 是私钥， id\_rsa.pub 是公钥

ssh-keygen -t rsa -b 2048 -N ''

给所有主机部署密钥

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

模块

ansible-doc 查看帮助，必须掌握

ansible-doc -l 列出所有模块

ansible-doc 模块名 查看该模块的帮助信息

ping

没有参数，检测主机的连通性，与 ping 无关，主要检测 ssh 是否可以连接

command | shell | raw

command 是默认模块，没有启用 shell ，所有shell 相关特性命令无法使用，例如 < > | &

raw 模块，没有 chdir create remove 等参数，能执行大部分操作

shell 模块，启动 /bin/sh 运行命令，可以执行所有操作

测试

ansible cache -m command -a 'chdir=/tmp touch f1' 创建成功

ansible cache -m shell -a 'chdir=/tmp touch f2' 创建成功

ansible cache -m raw -a 'chdir=/tmp touch f3' 文件可以创建，但无法切换目录，文件在用户家目录下生成

复杂操作怎么办，使用脚本来解决

#!/bin/bash

id zhang3

if [ $? != 0 ];then

useradd li4

echo 123456 |passwd --stdin li4

fi

ansible all -m scriopt -a "urscriopt.sh"

copy lineinfile replace 模块

copy 把文件发布到远程其他主机上面

lineinfile 修改一个文件的一行，以行为基础，整行修改

replace 修改文件的某一部分，以正则表达式匹配为基础修改

利用 copy 模块修改所有机器的 /etc/resolv.conf 为

nameserver 8.8.8.8

利用 lineinfile 修改 /etc/sysconfig/network-scriopts/ifcfg-eth0

ONBOOT=yes|no

ansible cache -m lineinfile -a 'path=/etc/sysconfig/network-scripts/ifcfg-eth0 regexp="^ONBOOT" line="ONBOOT=\"no\""

利用 replace 修改 /etc/sysconfig/network-scriopts/ifcfg-eth0

ONBOOT=no|yes

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

yum 模块 installed 安装， removed 删除

ansible other -m yum -a 'name="lrzsz" state=removed'

ansible other -m yum -a 'name="lftp" state=removed'

ansible other -m yum -a 'name="lrzsz,lftp" state=installed'

service 模块 name 指定服务名称，enabled= yes|no 设置开机启动， state=stopped|started 启动关闭服务

设置 chronyd 服务开启启动，并启动服务

ansible other -m service -a 'name="chronyd" enabled="yes" state="started"'

setup 模块，查看信息 filter 过滤指定的关键字

playbook httpd.yml

---

- hosts: web

remote\_user: root

tasks:

- name: install the latest version of Apache

yum:

name: httpd

state: installed

- lineinfile:

path: /etc/httpd/conf/httpd.conf

regexp: '^Listen '

insertafter: '^#Listen '

line: 'Listen 8080'

- lineinfile:

path: /etc/httpd/conf/httpd.conf

regexp: '^#ServerName'

line: 'ServerName localhost'

- copy:

src: /root/index.html

dest: /var/www/html/index.html

owner: apache

group: apache

mode: 0644

- service:

name: httpd

state: started

enabled: yes

使用变量和过滤器添加用户

---

- hosts: cache

remote\_user: root

vars:

un: nb

tasks:

- user:

name: "{{un}}"

group: users

password: "{{'123456'|password\_hash('sha512')}}"

- name: ooxx

shell: chage -d 0 "{{un}}"

添加用户，忽略错误

---

- hosts: db

remote\_user: root

vars:

un: zhang3

tasks:

- shell: adduser "{{un}}"

ignore\_errors: True

- shell: echo 123456|passwd --stdin "{{un}}"

- name: ooxx

shell: chage -d 0 "{{un}}"

handlers 触发

---

- hosts: web

remote\_user: root

tasks:

- name: install the latest version of Apache

yum:

name: httpd

state: installed

- lineinfile:

path: /etc/httpd/conf/httpd.conf

regexp: '^Listen '

insertafter: '^#Listen '

line: 'Listen 8080'

notify:

- restart\_httpd

- lineinfile:

path: /etc/httpd/conf/httpd.conf

regexp: '^#ServerName'

line: 'ServerName localhost'

notify:

- restart\_httpd

- copy:

src: /root/index.html

dest: /var/www/html/index.html

owner: apache

group: apache

mode: 0644

- copy:

src: /root/httpd.conf

dest: /etc/httpd/conf/httpd.conf

owner: root

group: root

mode: 0644

tags: config\_httpd

notify:

- restart\_httpd

handlers:

- name: restart\_httpd

service:

name: httpd

state: restarted

enabled: yes

把负载高的web服务器停止

---

- hosts: web

remote\_user: root

tasks:

- shell: uptime |awk '{printf("%.2f",$(NF-2))}'

register: result

- service: name=httpd state=stopped

when: result.stdout|float > 0.7

追加 debug 调试信息

- name: Show debug info

debug: var=ooxx

测试命令 awk 'BEGIN{while(1){}}'

循环添加多用户

---

- hosts: web2

remote\_user: root

tasks:

- user:

name: "{{item.name}}"

group: "{{item.group}}"

password: "{{'123456'|password\_hash('sha512')}}"

with\_items:

- {name: "nb", group: "users"}

- {name: "dd", group: "mail" }

- {name: "jj", group: "wheel"}

- {name: "lx", group: "root" }

嵌套循环

---

- hosts: cache

remote\_user: root

vars:

un: [a, b, c]

id: [1, 2, 3]

tasks:

- name: add users

shell: echo {{item}}

with\_nested:

- "{{un}}"

- "{{id}}"

标签设置 tags: config\_httpd

检测语法

ansible-playbook --syntax-check playbook.yaml

测试运行

ansible-playbook -C playbook.yaml

显示受到影响到主机 --list-hosts

显示工作的 task --list-tasks

显示将要运行的 tag --list-tags