RHCE8（PM）

练习环境默认没有虚拟机，需要reset生成虚拟机（虚拟机会自动启动）

[root@server1 ~]# rht-vmctl reset control

[root@server1 ~]# rht-vmctl reset node1

[root@server1 ~]# rht-vmctl reset node2

[root@server1 ~]# rht-vmctl reset node3

[root@server1 ~]# rht-vmctl reset node4

[root@server1 ~]# rht-vmctl reset node5

模拟环境的所有虚拟机，可以通过virt-manager管理

真正考试，所有虚拟机都已经准备OK，不需要reset，可以直接使用虚拟机

考试共计15题，4个小时！

**第一题：**

真机ssh远程control虚拟机(使用域名或IP地址）:

[root@server1 ~]# ssh [alice@control](mailto:alice@control)

备注：control主机的yum源是提前配置OK的

[alice@control ~]$ sudo yum -y install ansible #安装软件包

[alice@control ~]$ pwd #确认自己的位置

/home/alice

[alice@control ~]$ mkdir -p /home/alice/ansible/roles

[alice@control ~]$ cd /home/alice/ansible/

写主配置文件，要参考/etc/ansible/ansible.cfg

[alice@control ansible]$ vim ansible.cfg #创建主配置文件

[defaults]

inventory = /home/alice/ansible/inventory

roles\_path = /home/alice/ansible/roles

remote\_user = alice

[privilege\_escalation]

become=True

become\_method=sudo

become\_user=root

become\_ask\_pass=False

[alice@control ansible]$ vim inventory #创建主机清单文件

[test01]

node1

[test02]

node2

[web]

node3

node4

[test05]

node5

[webtest:children]

web

[alice@control ansible]$ ansible all -m ping #测试一下环境

**第二题：**

参考ansible-doc yum\_repository【EXAMPLES】

ansible-doc -l | grep yum // -l 过滤，

# yum\_repository 是陪yum源的 ，yum是装包的；

[alice@control ansible]$ vim adhoc.sh

#!/bin/bash

ansible all -m yum\_repository -a 'name=BASE description="software base" baseurl="http://study.lab0.example.com/rhel8/BaseOS" gpgcheck=yes gpgkey="http://study.lab0.example.com/rhel8/RPM-GPG-KEY-redhat-release" enabled=yes'

ansible all -m yum\_repository -a 'name=STREAM description="software stream" baseurl="http://study.lab0.example.com/rhel8/AppStream" gpgcheck=yes gpgkey="http://study.lab0.example.com/rhel8/RPM-GPG-KEY-redhat-release" enabled=yes'

[alice@control ansible]$ chmod +x adhoc.sh

[alice@control ansible]$ ./adhoc.sh

测试：可以去node1-node5任意安装一个软件包测试

**第三题：**

参考ansible-doc yum【EXAMPLES】

[alice@control ansible]$ vi tools.yml

---

- hosts: test01,test02,web

tasks:

- name: Install a list of packages

yum:

name:

- php

- tftp

state: present

- hosts: test01

tasks:

- name: groups

yum:

name: "@RPM Development Tools"

state: present

- name: update

yum:

name: '\*'

state: latest

也可以参考QQ群共享中的文件格式填写剧本！

[alice@control ansible]$ ansible-playbook tools.yml

验证：ssh远程node1

# rpm -qa |grep php

# yum grouplist 看有没有installed group的软件组包

**第四题：**

[alice@control ansible]$ sudo yum list |grep roles

#软件包记不住，可以grep过滤

[alice@control ansible]$ sudo yum -y install rhel-system-roles

#安装软件包（题目有要求）

[alice@control ansible]$ sudo rpm -ql rhel-system-roles

#查看软件安装到哪里

[alice@control roles]$ cp -r /usr/share/ansible/roles/rhel-system-roles.timesync/ /home/alice/ansible/roles/

#根据考试题目第一题，要求：所有角色必须放在/home/alice/ansible/roles目录

使用角色之前需要看帮助（README.md,找Example Playbook)

[alice@control ansible]$ vim roles/rhel-system-roles.timesync/README.md

[root@control ansible]# vi timesync.yml #写剧本做题

---

- hosts: all

vars:

timesync\_ntp\_servers:

- hostname: 172.25.254.250

iburst: yes

roles:

- rhel-system-roles.timesync

验证：ssh远程node1，查看chrony配置文件

[root@node1 ~]# head /etc/chrony.conf

**第五题：**

[alice@control ansible]$ vi roles/down.yml

- src: http://study.lab0.example.com/roles/haproxy.tar

name: haproxy

- src: http://study.lab0.example.com/roles/myphp.tar

name: myphp

[alice@control ansible]$ ansible-galaxy install -r roles/down.yml

ansible-galaxy默认是去官网下载角色，通过-r指定上面一步创建的文件名，到特定的链接下载role

验证：ls roles

多了两个角色（haproxy，myphp）

**第六题：**

[alice@control ansible]$ ansible-galaxy init roles/httpd

#创建一个名称为httpd的角色，在roles目录下创建

验证：[alice@control ansible]$ ls roles/httpd/

参考ansible-doc yum

ansible-doc service

ansible-doc firewalld

[alice@control ansible]$ vim roles/httpd/tasks/main.yml #编写任务

- name: install httpd

yum:

name: httpd

state: present

#调用的模块是yum，安装的软件名称是httpd，present代表安装

- name: run httpd

service:

name: httpd

state: started

enabled: yes

#调用的模块是service，启动的服务名称是httpd，started代表启动服务

#enabled: yes代表设置开机自启

- firewalld:

service: http

permanent: yes

immediate: yes

state: enabled

#调用的模块是firewalld，state:enabled是添加防火墙规则

#运行http服务的访问（放行，不要拦截）

#permanent:yes代表永久规则，immediate: yes代表立刻生效

- template:

src: index.html.j2

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

#调用template模块，将index.html.j2文件拷贝到被管理主机

#放到被管理主机的/var/www/html/index.html

参考：ansible node3 -m setup | less（找facts变量）

[alice@control ansible]$ vi roles/httpd/templates/index.html.j2

Welcome to {{ansible\_fqdn}} on {{ansible\_eth0.ipv4.address}}

[alice@control ansible]$ vi myrole.yml #编写剧本调用role角色

---

- hosts: webtest

roles:

- httpd

[alice@control ansible]$ ansible-playbook myrole.yml

测试：

[alice@control ansible]$ curl 172.25.254.103

[alice@control ansible]$ curl 172.25.254.104

**第七题：**

角色是前面第五题已经下载好的haproxy和myphp

参考cat roles/httpd/tasks/main.yml（参考之前写的防火墙规则）

[alice@control ansible]$ vim web.yml

---

- hosts: webtest

roles:

- myphp

#远程webtest主机(node3,node4)，调用myphp角色，自动配置php动态网页

- hosts: test05

roles:

- haproxy

#远程test05(node5)，调用haproxy角色，自动配置负载均衡调度器

tasks:

- firewalld:

service: http

permanent: yes

immediate: yes

state: enabled

#调用firewalld设置防火墙规则（如第6题）

[alice@control ansible]$ ansible-playbook web.yml #执行剧本

验证：

[alice@control ansible]$ curl http://node5 #访问静态

[alice@control ansible]$ curl http://node5/index.php #访问动态

**第八题：**

参考：ansble node1 -m setup | less

ansible-doc lvol （逻辑卷）

[alice@control ansible]$ vi lvm.yml

---

- hosts: all

tasks:

- name: display a error info

debug:

msg: "VG not found"

when: "'search' not in ansible\_lvm.vgs"

failed\_when: "'search' not in ansible\_lvm.vgs"

#调用debug模块，显示报错信息，msg：后面是报错信息的内容

#when定义什么时候执行debug模块，条件是在ansible\_lvm.vgs变量里面找不到search卷组

#failed\_when定义什么情况是剧本失败，条件是在ansible\_lvm.vgs变量里面找不到search卷组

#找不到search这个卷组，就立刻停止执行剧本，不在往下执行后面的任务

- block:

- lvol:

vg: search

lv: mylv

size: 1000m

rescue:

- debug:

msg: "insufficient free space"

- lvol:

vg: search

lv: mylv

size: 500m

always:

- filesystem:

fstype: ext4

dev: /dev/search/mylv

#先通过block尝试创建1000M的逻辑卷

#使用的是lvol模块，vg指定使用哪个卷组创建逻辑卷

#使用的是search这个卷组

#lv指定创建的卷组叫什么名字，这里名称为mylv

#size指定逻辑卷大小（1000m或者500m）

#如果前面block创建1000m无法成功，那么才执行rescue后面的任务

#调用相同的lvol模块，创建500m的逻辑卷

#最后不管创建多大的逻辑卷，都要格式化

#使用的是filesytem模块，格式化的格式是ext4，dev后面写对谁格式化

**第九题：**

参考：ansible node1 -m setup |less （查变量名）

下载文件

修改文件（变量)

写剧本，拷贝文件到所有主机（template）

[alice@control ansible]$ sudo yum -y install wget

#装包

[alice@control ansible]$ wget http://study.lab0.example.com/materials/newhosts.j2

#下载考题给的文件

[alice@control ansible]$ cat newhosts.j2

#看一看

[alice@control ansible]$ vim newhosts.j2 #修改文件方法一

直接复制考题的内容，到文件中

[alice@control ansible]$ vim newhosts.j2 #修改文件方法二

在源文件下面，手写如下内容

{% for i in groups.all %}

{{hostvars[i].ansible\_eth0.ipv4.address}} {{hostvars[i].ansible\_fqdn}} {{hostvars[i].ansible\_hostname}} {% endfor %}

#groups是ansible自带的变量（魔法变量）被管理的组，all是所有主机

#groups.all代表，被管理组中的所有主机

#hostvars是ansible自带的变量（魔法变量），他可以提取某个主机的变量信息，如hostvars[node1]

[alice@control ansible]$ vi newhosts.yml #编写剧本拷贝上面准备的文件

---

- hosts: all

- hosts: test01

tasks:

- template:

src: newhosts.j2

dest: /etc/newhosts

#注意事项：先远程所有主机，获取所有主机的变量

#然后再远程test01,拷贝模版文件newhosts.j2到对方主机

#模版文件里通过for循环逐一写入所有主机的IP，域名和主机名

**第十题：**

[alice@control ansible]$ vim inventory #修改主机清单文件，添加如下内容：

[test01:vars]

xx=test01

[test02:vars]

xx=test02

[web:vars]

xx=Webserver

#给test01组创建变量,变量名称为xx（名称任意），变量的值为test01

#其他所有组类似

参考：ansible-doc copy

[alice@control ansible]$ vi newissue.yml #编写剧本

---

- hosts: all

tasks:

- copy:

content: "{{xx}}"

dest: /etc/issue

[alice@control ansible]$ ansible-playbook newissue.yml

#执行剧本

验证：

[alice@control ansible]$ ansible all -a "cat /etc/issue"

**第十一题：**

参考：ansible-doc group

ansible-doc file

[alice@control ansible]$ vi webdev.yml #编写剧本

---

- hosts: test01

tasks:

- group:

name: webdev

state: present

#调用group组，创建webdev

- file:

path: /webdev

group: webdev

mode: '2775'

state: directory

#调用file模块创建目录，设置权限，所属组

- file:

src: /webdev

dest: /var/www/html/webdev

state: link

#调用file模块，src后面是源文件或目录，dest后面是软链接

#state的值是link，表示创建软链接

- copy:

content: "It's works!"

dest: /webdev/index.html

#调用copy模块，创建一个index.html文件，内容是：It's works!

- shell: setenforce 0

#调用shell模块执行setenforce 0关闭selinux

#参考前面题目写的防火墙规则，复制过来

参考 cat roles/httpd/tasks/main.yml

- firewalld:

service: http

permanent: yes

immediate: yes

state: enabled

- service:

name: httpd

state: started

enabled: yes

#确保服务是启动

[alice@control ansible]$ ansible-playbook webdev.yml

#执行剧本

测试：在真机使用firefox <http://node1/webdev>

**第十二题：**

使用真机浏览器访问一下http://study.lab0.example.com/materials/hardware.empty

看看大致的内容

原文件内容如下：

hostname=inventoryhostname

mem=memory\_in\_MB

bios=BIOS\_version

vdasize=disk\_vda\_size

vdbsize=disk\_vdb\_siz

参考：ansible-doc get\_url

ansible-doc replace

ansible node1 -m setup | less

[alice@control ansible]$ vim hardware.yml #编写剧本

---

- hosts: all

ignore\_errors: yes

tasks:

- get\_url:

url: http://study.lab0.example.com/materials/hardware.empty

dest: /root/hardware.txt

- replace:

path: /root/hardware.txt

regexp: inventoryhostname

replace: "{{ansible\_hostname}}"

- replace:

path: /root/hardware.txt

regexp: memory\_in\_MB

replace: "{{ansible\_memtotal\_mb}}"

- replace:

path: /root/hardware.txt

regexp: BIOS\_version

replace: "{{ansible\_bios\_version}}"

- replace:

path: /root/hardware.txt

regexp: disk\_vda\_size

replace: "{{ansible\_devices.vda.size}}"

- replace:

path: /root/hardware.txt

regexp: disk\_vdb\_size

replace: "{{ansible\_devices.vdb.size if ansible\_devices.vdb.size is defined else 'NONE'}}"

#通过get\_url下载题目指定的文件

#再通过replace修改文件内容

#replace里面的path指定需要修改的文件名，regexp指定原来的旧内容（需要被替换的内容）

#replace里面的replace后面是需要提出为的新内容{新内容都是变量}

验证：

远程node1-node5，逐一查看/root/hardware.txt文件的内容

**第十三题：**

[root@node1 ~]# vi passdb.yml

---

pw\_dev: ab1234

pw\_man: cd5678

#设置变量名和变量的值，：后面有空格

[alice@control ansible]$ vim secret.txt #加密上面passdb.yml文件的密码

[pwd@1234](mailto:pwd@1234)

参考：ansible-vault --help

[alice@control ansible]$ ansible-vault encrypt --vault-id secret.txt passdb.yml

#ansible-vault是密码用的命令，encrypt代表加密，decrypt代表解密

#--vault-id后面跟的是密码文件，最后是对passdb.yml文件加密

**第十四题：**

[alice@control ansible]$ wget <http://study.lab0.example.com/materials/name_list.yml>

[alice@control ansible]$ cat name\_list.yml

users:

- name: tom

job: dev

- name: jerry

job: man

参考：ansible-doc user

方法一：

[alice@control ansible]$ vi users.yml

---

- hosts: test01,test02,web

vars\_files:

- name\_list.yml

- passdb.yml

tasks:

- group: name=devops

when: ('test01' in group\_names or 'test02' in group\_names)

- group: name=opsmgr

when: ('web' in group\_names)

- user: name={{item.name}} password={{pw\_dev|password\_hash('sha512')}} groups=devops append=yes

when: (item.job == 'dev') and ('test01' in group\_names or 'test02' in group\_names)

loop: "{{users}}"

- user: name={{item.name}} password={{pw\_man|password\_hash('sha512')}} groups=opsmgr append=yes

when: (item.job == 'man') and ('web' in group\_names)

loop: "{{users}}"

#远程所有test01，test02，web组

#通过when判断，现在远程的组是不是test01或者test02，如果是，则创建devops组

#通过when判断，现在远程的组是不是web组，如果是，则创建opsmgr组

#group.names是ansible自带得的变量（魔法变量）代表正在远程的组的名称

#user模块通过item对loop进行循环，loop的值是users变量（这个变量在前面的文件中）

#users变量里面有两组数据，一组是tom的数据（名称和职位），一组是jerry的数据（名称和职位）

#users变量里面有name和job

#user模块创建用户使用item对loop（也就是user循环）调用users.name（也就是users变量里面的name）创建用户，再通过password设置对应的密码

#如果直接循环，test01，test02，web都将创建tom和jerry用户

#下面再通过when设置条件,仅当item.job(也就是users.job)的值是dev时才创建tom

#仅当item.job(也就是users.job)的值是man时才创建jerry

方法二：

[alice@control ansible]$ vi users.yml

---

- hosts: test01,test02

vars\_files:

- name\_list.yml

- passdb.yml

tasks:

- group: name=devops

- user:

name: "{{item.name}}"

groups: devops

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

loop: "{{users}}"

when: item.job == 'dev'

- hosts: web

vars\_files:

- name\_list.yml

- passdb.yml

tasks:

- group: name=opsmgr

- user:

name: "{{item.name}}"

groups: opsmgr

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

loop: "{{users}}"

when: item.job == 'man'

#编写一个剧目，远程test01和test02，创建devops组，创建用户tom，设置密码

#首先通过vars\_files读取两个文件中的变量（name\_list.yml和passdb.yml)

#使用group模块给test01,test02组创建固定的组，名称为devops

#使用user模块创建系统账户，name后面跟账户的名称，账户名称使用item调用loop循环

#item是关键词不可修改，下面loop对变量{{users}}循环，

#users中包含两组数据，一组是tom的数据，一组是jerry的数据

#本来user模块通过loop循环可以创建两个账户，因为user中包含两组数据

#但是，下面又通过when语句做条件判断，仅当item.job的值=='关键词'时才创建用户

#对于test01和test02组的主机，就可以判断仅当item.job的值为dev时才创建账户

#创建的账户名称为item.name，也就是users.name（系统账户名称）

#item.job也就是users.job（职位的描述）

#账户对应的密码指定读取passdb.yml中定义的变量pw\_dev和pw\_man

#第一个剧本编写好后，使用数字+yy复制，p粘贴，编写第二个剧目

#适当修改第二个剧目的内容即可

**第十五题：**

[alice@control ansible]$ wget <http://study.lab0.example.com/materials/topsec.yml>

[alice@control ansible]$ ansible-vault rekey topsec.yml #rekey是改密码

Vault password: 【老密码】

New Vault password: 【新密码】

Confirm New Vault password: 【新密码】