ansible dev -m shell -a "firewall-cmd --zone=publıic --permanent --add-service=http"

ansible dev -m shell -a "firewall-cmd --reload"

ansible dev -m shell -a "firewall-cmd --list service"

ansible dev -m shell -a "systemctl enable firewalld"

if you see a questions with and parted commands, and if your ansible playbook cant mount a device to file system

you can fix it by connecting to a machine with ssh where disk devices like vdb, vdc exist. then you should run

mkfs.ext4 /dev/vdb1

mkfs.ext4 /dev/vdc1

After that you should run your playbook

Considering the below case, you should know how to use mount and parted commands in playbook. you should also know how to use debug module to include message and how to use lvol command to create logical volume

ansible all -a "df -h" , to check mount points

ansible all -a "lsblk -fb" , to check file system type

you might verify your playbooks after they run with above commands

In your first questions you might asked to define repositories

you should know how to use yum\_repository module.

you should use

ansible-playbook playbookname --syntax-check

you should make syntax check before you run your commands

ansible dev -m shell -a "systemctl enable firewalld"

you mignt need these commands in some questions when you run your playbook if you in the error message that firewalld is not active

if you see a questions with and parted commands, and if your ansible playbook cant mount a device to file system

you can fix it by connecting to a machine with ssh where disk devices like vdb, vdc exist. then you should run

mkfs.ext4 /dev/vdb1

mkfs.ext4 /dev/vdc1

After that you should run your playbook

Considering the below case, you should know how to use mount and parted commands in playbook. you should also know how to use debug module to include message and how to use lvol command to create logical volume

ansible all -a "df -h" , to check mount points

ansible all -a "lsblk -fb" , to check file system type

you might verify your playbooks after they run with above commands

In your first questions you might asked to define repositories

you should know how to use yum\_repository module.

you should use

ansible-playbook playbookname --syntax-check

you should make syntax check before you run your commands

**The most important thing !!!**

When you set ansible environment

run:

1. ansible all -m ping

2. in root and user prompts check ssh auto connection with

for i in 1 2 3 4 5; do ssh node$i; done (run this in both root and user prompts)

# for i in 1 2 3 4 5; do ssh node$i; done

su - username

for i in 1 2 3 4 5; do ssh node$i; done

run below commands in root prompt

for i in 1 2 3 4 5; do ssh node$i 'echo "username ALL=(ALL) NOPASSWD:ALL" > /etc/sudoers.d/username'; done

The last command is important to ensure username user has sudo privileges assigned

su - username

mkdir -p ansible/roles

mkdir -p ansible/mycollections

[defaults]

inventory=./inventory

roles\_path=./roles

collections\_path=./mycollections

remote\_user=username

host\_key\_checking =false

[privilege\_escalation]

become:=true

become\_user=root

become\_method=sudo

become\_ask\_pass=false

take not above ansible.cfg configuration

mkdir -p ansible/mycollections

NEW

cd mycollections

sample collection

vim requirements.yml

---

collections:

- ansible.posix

- community.general

or

---

collections:

- http://path1

- http://path2

install collection:

ansible-galaxy collection install -r requirements.yml -p .

you can have such question

Learn how to use yum\_repository module to define repos:

Sample yum\_repository configuration:

---

- name: Add repos

hosts: all

become: true

tasks:

- name: Add BaseOS repo

yum\_repository:

name: BaseOS

description: "CentOS8 Linux BaseOS"

baseurl: http://rhgls.example.local/repo/BaseOS/

gpgcheck: yes

gpgkey: file:///etc/pki/rpm-gpg/RPM-GPG-KEY-centosofficial

enabled: yes

- name: Add AppStream repo

yum\_repository:

name: AppStream

description: "CentOS8 Linux AppStream"

baseurl: http://rhgls.example.local/repo/AppStream/

gpgcheck: yes

gpgkey: file:///etc/pki/rpm-gpg/RPM-GPG-KEY-centosofficial

enabled: yes

Take a look at the difference below config :

---

- name: collect hosts

hosts: all

become: true

tasks:

- name: put collected hosts on dev hosts

template:

src: hosts.j2

dest: /root/myhosts

when: inventory\_hostname in groups['dev']

---

- name: collect hosts

hosts: all

become: true

gather\_facts: true

tasks:

- name: put collected hosts on dev hosts

template:

src: hosts.j2

dest: /root/myhosts

when: inventory\_hostname in groups['dev']

can you tell me what is the difference between the two ?

You might be asked this

gather\_facts: true

When you use when: inventory\_hostname in groups['dev'] statement, you should use gather\_facts: true

……………………………………………………………

You should learn how to write cron job

---

- name: create logger job

hosts: all

become: true

tasks:

- name: create loggerjob

cron:

name: loggerjob

minute: "\*/2"

hour: "\*"

day: "\*"

month: "\*"

weekday: "\*"

user: "root"

job: "/usr/bin/logger First\_Log"

You might be also asked this

You might have such question with lvm:

---

- name: create logical volume and file system

hosts: all

become: true

gather\_facts: true

tasks:

- name: create debug message

debug:

msg: "VG Not found"

when: ansible\_lvm.vgs.research is not defined

- name: create logical volume with size of 1500m

lvol:

lv: date

vg: research

size: 1500

when: ansible\_lvm.vgs.research.size\_g > '1.5'

- name: create debug message

debug:

msg: "LV Can not be created with following size"

when: ansible\_lvm.vgs.research.size\_g < '1.5'

- name: create smaller logical volume

lvol:

lv: data

vg: research

size: 800

when: ansible\_lvm.vgs.research.size\_g < '1.5'

- name: create file system

filesystem:

dev: /dev/research/data

fstype: ext4

In the exam, you will have a below question

---

- name: set motd on the hosts

hosts: all

become: true

tasks:

- name: set motd on dev hosts

copy:

content: Development

dest: /etc/motd

when: ('dev' in group\_names)

- name: set motd on test hosts

copy:

content: Test

dest: /etc/motd

when: ('test' in group\_names)

- name: set motd on prod hosts

copy:

content: Production

dest: /etc/motd

when: ('prod' in group\_names)

or

you can be asked to set /etc/issue

---

- name: set motd on the hosts

hosts: all

become: true

tasks:

- name: set motd on dev hosts

copy:

content: Development

dest: /etc/issue

when: ('dev' in group\_names)

- name: set motd on test hosts

copy:

content: Test

dest: /etc/issue

when: ('test' in group\_names)

- name: set motd on prod hosts

copy:

content: Production

dest: /etc/issue

when: ('prod' in group\_names)

View Hakan’s profileHakan Yıldırğan

Hakan Yıldırğan 2:41 AM

This message may contain unwanted or harmful content. View message anyway

Olu Familusi sent the following messages at 2:42 AM

View Olu’s profileOlu Familusi

Olu Familusi 2:42 AM

This is where i have been struggling

This templates

Hakan Yıldırğan sent the following message at 2:42 AM

View Hakan’s profileHakan Yıldırğan

Hakan Yıldırğan 2:42 AM

You will be asked this question:

---

- name: install packages

hosts: dev,test,prod

become: true

tasks:

- name: install packages on all hosts

yum:

name:

- php

- mariadb

state: latest

- name: install development tools on dev hosts

yum:

name: '@Development Tools'

state: latest

when: ('dev' in group\_names)

- name: undate packages on dev hosts

yum:

name: '\*'

state: latest

when: ('dev' in group\_names)

nsible packages.yml --syntax-check

ansible packages.yml

after running

ansible dev -a 'yum list installed | grep php'

ansible dev -a 'yum list installed | grep mariadb'

ansible test -a 'yum list installed | grep php'

ansible dev -a 'yum list installed | grep mariadb'

ansible dev -a 'yum list installed | grep php'

ansible prod -a 'yum list installed | grep mariadb'

ansible dev -a 'yum grouplist installed'

You should check your playbooks after running

my\_sys=

my\_BIOS=

my\_MEMORY=

my\_sda=

my\_sdb=

---

- name: collect hardware info

hosts: all

become: true

gather\_facts: true

tasks:

- name: copy file

copy:

src: hwreport.txt

dest: /root/hwreport.txt

- name: change my\_sys

lineinfile:

regex: ^my\_sys

line: my\_sys={{ ansible\_hostname }}

state: present

path: /root/hwreport.txt

- name: change bios

lineinfile:

regex: ^my\_BIOS

line: my\_BIOS={{ ansible\_bios\_version }}

state: present

path: /root/hwreport.txt

- name: change memory

lineinfile:

regex: ^my\_MEMORY

line: my\_BIOS={{ ansible\_memtotal\_mb }}

state: present

path: /root/hwreport.txt

- name: change sda

lineinfile:

regex: ^my\_sda

line: my\_sda={%if ansible\_devices.sda is defined%}{{ansible\_devices.sda.size}}{%else%}NONE{%endif%}

state: present

path: /root/hwreport.txt

- name: change sdb

lineinfile:

regex: ^my\_sdb

line: my\_sdb={%if ansible\_devices.sdb is defined%}{{ansible\_devices.sdb.size}}{%else%}NONE{%endif%}

state: present

path: /root/hwreport.txt

You might also asked this question

ansible all -a 'cat /root/hwreport.txt'

Olu Familusi sent the following messages at 2:48 AM

View Olu’s profileOlu Familusi

Olu Familusi 2:48 AM

Wow!!

I will try my hardest and God will take the glory

Hakan Yıldırğan sent the following messages at 2:48 AM

View Hakan’s profileHakan Yıldırğan

Hakan Yıldırğan 2:48 AM

sample apache role

cd roles

ansible-galaxy init apache

vim /apache/tasks/main.yml

---

# tasks file for apache

- name: install packages

yum:

name:

- httpd

- firewalld

state: latest

- name: start and enable http service

service:

name: httpd

state: started

enabled: true

- name: start and enable firewall service

service:

name: firewalld

state: started

enabled: true

- name: add http service to firewall

firewalld:

service: http

state: enabled

immediate: true

permanent: true

- name: create web content

template:

src: index.html.j2

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

notify:

- restart\_httpd

vim /apache/templates/index.html.j2

Welcome to {{ansible\_fqdn}} on {{ansible\_default\_ipv4.address}}

vim /apache/handlers/main.yml

---

# handlers file for apache

- name: restart\_httpd

service:

name: httpd

state: restarted

cd ../

the playbook of this role

---

- name: use apache role

hosts: webservers

become: true

roles:

- apache

check it

curl node3

curl node4

sample roles question

cd roles

vim install.yml

---

- src: http://classroom.example.local/materials/role1.tar.gz

name: balancer

- src: http://classroom.example.local/materials/role2.tar.gz

name: phphello

ansible-galaxy install -r install.yml -p .

its play playbook

cd ../

---

- name: use php hello role

hosts: webservers

become: true

roles:

- phphello

- name: use balancer role

hosts: balancers

become: true

roles:

- balancer

check curk node5/index.php

run this two times to check load balancer service

Olu Familusi sent the following message at 2:55 AM

View Olu’s profileOlu Familusi

Olu Familusi 2:55 AM

Wow!

Hakan Yıldırğan sent the following messages at 2:55 AM

View Hakan’s profileHakan Yıldırğan

Hakan Yıldırğan 2:55 AM

---

- name: create webdev

hosts: dev

become: true

tasks:

- name: install packages

yum:

name:

- httpd

- python3-policycoreutils

state: present

- name: start and enable http service

service:

name: httpd

state: started

enabled: true

- name: add http service to firewall

firewalld:

service: http

state: enabled

permanent: true

immediate: true

- name: create apache group

group:

name: apache

- name: create webdev directory

file:

path: /webdev

group: apache

mode: '2775'

state: directory

- name: create symbolic link

file:

src: /webdev

dest: /var/www/html/webdev

state: link

- name: create web content

copy:

content: Development

dest: /webdev/index.html

- name: restart http service

service:

name: httpd

state: restarted

- name: allow httpd from this directory

sefcontext:

target: '/webdev/index.html'

setype: httpd\_sys\_content\_t

state: present

- name: restore context

shell: restorecon -vR /webdev

You will have such question

ansible dev -m shell -a "firewall-cmd --zone=publıic --permanent --add-service=http"

ansible dev -m shell -a "firewall-cmd --reload"

ansible dev -m shell -a "firewall-cmd --list service"

if you get firewalld is not active run these firewalld commands and after that run your playbook

you should know how to create fault file

take these notes:

echo abbbccdd > password.txt

ansible vault create vault.yml --vault-password-file=password.txt

vault file content should be like:

---

- dev\_pass: abc123

- mgr\_pass: dff456

you should know how to rekey vault password

ansible-vault create vault.yml --vault-password-file=password.txt

correction

you should know how to rekey vault password

ansible-vault rekey secret.yml

you should enter the current password and new password two times

You might be asked such a question

---

- name: create users and groups

hosts: all

become: true

gather\_facts: true

vars\_files:

- vault.yml

- user\_list.yml

You will be asked such a questions

you should run this playbook as

ansible-playbook create\_user.yml --vault-password-file=password.txt --syntax-check

ansible-playbook create\_user.yml --vault-password-file=password.txt

tasks:

- name: create devops group on dev and test hosts

group:

name: devops

when: (inventory\_hostname in groups['dev'] or inventory\_hostname in groups['test'])

- name: create opsmgr group on prod hosts

group:

name: opsmgr

when: inventory\_hostname in groups['prod']

- name: create users and dev and test hosts

user:

name: "{{item.name}}"

groups: devops

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

uid: "{{item.uid}}"

when: (inventory\_hostname in groups['dev'] or inventory\_hostname in groups['test']) and item.job == "developer"

loop: "{{user}}"

- name: create users on prod hosts

user:

name: "{{item.name}}"

groups: opsmgr

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

uid: "{{item.uid}}"

when: inventory\_hostname in groups['prod'] and item.job == "manager"

loop: "{{user}}"

---

- hosts: all

tasks:

- name: Enable SELinux in enforcing mode

selinux:

policy: targeted

state: enforcing

ansible all -a 'cat /etc/selinux/config'

You will also be asked this question

You will need to use

ansible-doc parted

ansible-doc mount

commands in the exam

you should study

ansible-doc parted

ansible-doc mount

ansible-doc selinux

ansible-doc lvol

There is a bug with mount module it doesnt be mounted as parted command cant tun fs\_type: ext4 parameter

in that case you should ssh to machine where vdc and vdb exist

and you should tun below commands manually

mkfs.ext4 /dev/vdb1

mkfs.ext4 /dev/vdc1

these comamnds will create file systems on the devices

after that your playbooks will run. I got such bug in the exam and solved it this way

keep this jinja in your mind: [127.0.0.1](http://127.0.0.1/) localhost localhost.localdomain localhost4 localhost4.localdomain4 ::1 localhost localhost.localdomain localhost6 localhost6.localdomain6 {% for host in groups['all']%} {{hostvars[host]['ansible\_default\_ipv4']['address']}} {{hostvars[host]['ansible\_fqdn']}} {{hostvars[host]['ansible\_hostname']}} {%endfor%}

you should know that magic variable in that syntax gets ip, hostname and fqdn

mean it this way

sudo yum install -y rhel-system-roles

cp -r /usr/share/ansible/roles/rhel-system-roles.timesync /roles

vim timesync.yml

---

- name: time synchronization

hosts: all

become: true

vars:

timesync\_ntp\_servers:

- hostname: classroom.example.local

iburst: yes

roles:

- rhel-system-roles.timesync

for check

ansible all -a 'chroncy sources -v'

you should see something like that at the command prompt when checking

classroom.example.local ....

when you write your inventory and ansible.cfg you should control your ansible enviroment with ansible all -m ping command

You

might also have such question

---

- name: change default target

hosts: all

become: True

tasks:

- name: change target

file:

src: /usr/lib/systemd/system/multi-user.target

dest: /etc/systemd/system/default.target

state: link

...

check it with:

ansible all -a 'systemctl get-default'

what you need to do to run this command

in both root and user prompts

for i in 1 2 3 4 5; do ssh node$i; done to check auto connection

then you should exit from nodes to come to control node

BUT

HERE IS IMPORTANT !!!

for i in 1 2 3 4 5; do ssh node$i 'echo "username ALL=(ALL) NOPASSWD:ALL" > /etc/sudoers.d/username'; done

here username is the user that you will be given in the exam

you should run this command from root prompt

to ensure username user root privileges when you run playbooks from user prompt

Hakan Yıldırğan 3:37 AM

# for i in 1 2 3 4 5; do ssh node$i; done

# for i in 1 2 3 4 5; do ssh node$i 'echo "username ALL=(ALL) NOPASSWD:ALL" > /etc/sudoers.d/username'; done

# for i in 1 2 3 4 5; do ssh node$i 'cat /etc/sudoers.d/username'; done

su - username

for i in 1 2 3 4 5; do ssh node$i; done

mkdir -p ansible/roles

mkdir -p ansible/mycollections

You should directly run this command on root and user command

It checks all nodes at the same time if they are connected automatically via ssh

You should also dont need to create ssh keys

ssh-keys are created by redhat staff so you dont need to create ssh-keys

this information is also given in exam instructions

for i in 1 2 3 4 5; do ssh node$i; done

When you connect you redhat exam enviroment open nodes console from there connect to root console

run cat /etr/hosts

to learn the ip of control node

then ssh to control node from the redhat exam environment by openinng command prompt

this way you dont need to go to nodes console everytime

when you access to root prompt of control node you should run

# for i in 1 2 3 4 5; do ssh node$i; done

to check you can access all nodes from control node automatically

after that switch to username user (username user will be given in exam)

su - username

$ or i in 1 2 3 4 5; do ssh node$i; done

IMPORTANT NOTES !!!

---

- name: set motd on the hosts

hosts: all

become: true

tasks:

- name: set motd on dev hosts

copy:

content: Development

dest: /etc/issue

when: ('dev' in group\_names)

- name: set motd on test hosts

copy:

content: Test

dest: /etc/issue

when: ('test' in group\_names)

- name: set motd on prod hosts

copy:

content: Production

dest: /etc/issue

when: ('prod' in group\_names)

View Hakan’s profileHakan Yıldırğan

Hakan Yıldırğan 4:16 AM

When you set /etc/issue, it leaves message to Monitor console. You can check it out with logout after set

---

- name: set motd on the hosts

hosts: all

become: true

tasks:

- name: set motd on dev hosts

copy:

content: Development

dest: /etc/motd

when: ('dev' in group\_names)

- name: set motd on test hosts

copy:

content: Test

dest: /etc/motd

when: ('test' in group\_names)

- name: set motd on prod hosts

copy:

content: Production

dest: /etc/motd

when: ('prod' in group\_names)

[defaults]

inventory=./inventory

roles\_path=./roles

collections\_path=./mycollections

remote\_user=username

host\_key\_checking =false

[privilege\_escalation]

become:=true

become\_user=root

become\_method=sudo

become\_ask\_pass=false

keep in mind those firewall comands, I had to use them in two questions in the exam

and mkfs.ext4 /dev/vdb1

I tried very tricks in my lab works so I knew how to pass the exam.

for the ansible collections you will need three things.

1. in asibble.cfg file add

collections\_path= ./mycollections (mycollections name şs given in the exam)

Olu Familusi sent the following message at 4:50 AM

View Olu’s profileOlu Familusi

Olu Familusi 4:50 AM

Yea...I dont know anything about that right now but will have to look into my book to see how to work on that

Hakan Yıldırğan sent the following messages at 4:51 AM

View Hakan’s profileHakan Yıldırğan

Hakan Yıldırğan 4:51 AM

2.

cd mycollections (mycollections should be in the same path with roles)

cd mycollections

vim requirements.yml

---

collections:

- http://path..

- http://path..

3. run collection

ansible-galaxy collection install -r requirements.yml -p .

You will have one question from collections part

if you know these 3 steps it is enough to answer question

you have to install rhel-system-roles with

sudo yum install -y rhel-system-roles

cp -r /usr/share/ansible/roles/rhel-system-roles.timesync /roles

if you need to use such a role:

---

- name: time synchronization

hosts: all

become: true

vars:

timesync\_ntp\_servers:

- hostname: classroom.example.local

iburst: true

roles:

- rhel-system-roles.timesync