RedHat

EX294 Exam

Certified Engineer (RHCE) exam for Red Hat Enterprise Linux 8

Product Questions: 35
Version: 5.0

Topic 1, LAB SETUP

You will need to set up your lab by creating 5 managed nodes and one control node.
So 6 machines total. Download the free RHEL8 iso from Red Hat Developers website.

Control node you need to set up

You need to create some static ips on your managed nodes then on the control node set them up in the /etc/hosts file as follows:

```
vim /etc/hosts
10.0.2.21 node1.example.com
10.0.2.22 node2.example.com
10.0.2.23 node3.example.com
10.0.2.24 node4.example.com
10.0.2.25 node5.example.com
```

```
yum -y install ansible
useradd ansible
echo password | passwd --stdin ansible
echo "ansible ALL=(ALL) NOPASSWD:ALL" > /etc/sudoers.d/ansible
su - ansible; ssh-keygen
ssh-copy-id node1.example.com
ssh-copy-id node2.example.com
ssh-copy-id node3.example.com
ssh-copy-id node4.example.com
ssh-copy-id node5.example.com
```

Each manage node setup

First, add an extra 2GB virtual harddisk to each control node 1,2,3. Then add an extra hard disk to control node 4. Do not add an extra hard disk to node 5. When you start up these machines the extra disks should be automatically located at /dev/sdb (or /dev/vdb depending on your hypervisor).

```
useradd ansible
echo password | passwd --stdin ansible
echo "ansible ALL=(ALL) NOPASSWD:ALL" > /etc/sudoers.d/ansible
```

Note python3 should be installed by default, however if it is not then on both the control node and managed nodes you can install it also set the default python3 if you are having trouble with python2 being the default.

```
yum -y install python3
alternatives --set python /usr/bin/python3
```

All machines need the repos available. You did this in RHSC

A. To set up locally you just need to do the same for each machine. Attach the rhel8 iso as a disk to virtualbox, kvm or whatever hypervisor you are using (this will be /dev/sr0). Then inside the machine:

```
mount /dev/sr0 to /mnt
```

Then you will have all the files from the iso in /mnt.

```
mkdir /repo
cp -r /mnt /repo
vim /etc/yum.repos.d/base.repo
```

Inside this file:

```
[baseos]
name=baseos
baseurl=file:///repo/BaseOS
gpgcheck=0
```

Also the appstream

```
vim /etc/yum.repos.d/appstream.repo
```

Inside this file:

```
[appstream]
name=appstream
baseurl=file:///repo/AppStream
gpgcheck=0
```

Question: 1

Install and configure ansible

User bob has been created on your control node. Give him the appropriate permissions on the control node. Install the necessary packages to run ansible on the control node.

Create a configuration file /home/bob/ansible/ansible.cfg to meet the following requirements:

- The roles path should include /home/bob/ansible/roles, as well as any other path that may be required for the course of the sample exam.
- The inventory file path is /home/bob/ansible/inventory.
- Ansible should be able to manage 10 hosts at a single time.
- Ansible should connect to all managed nodes using the bob user.

Create an inventory file for the following five nodes:

nodel.example.com
node2.example.com
node3.example.com
node4.example.com
node5.example.com

Configure these nodes to be in an inventory file where node1 is a member of group dev. nodc2 is a member of group test, nodc3 is a member of group proxy, nodc4 and node 5 are members of group prod. Also, prod is a member of group webservers.

Answer: See the Explanation for complete Solution below.

Explanation:

In/home/sandy/ansible/ansible.cfg

```
[defaults]
inventory=/home/sandy/ansible/inventory
roles_path=/home/sandy/ansible/roles
remote_user= sandy
host_key_checking=false
[privilegeescalation]
become=true
become_user=root
become_method=sudo
become_ask_pass=false
```

In /home/sandy/ansible/inventory

```
[dev]
node 1 .example.com
[test]
node2.example.com
[proxy]
node3 .example.com
[prod]
node4.example.com
node5 .example.com
[webservers:children]
prod
```

Question: 2

Create a file called adhoc.sh in /home/sandy/ansible which will use adhoc commands to set up a new repository. The name of the repo will be 'EPEL' the description 'RHEL8' the baseurl is 'https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rmp' there is no gpgcheck, but you should enable the repo.

* You should be able to use an bash script using adhoc commands to enable repos. Depending on your lab setup, you may need to make this repo "state=absent" after you pass this task.

Answer: See the Explanation for complete Solution below.

Explanation:

```
chmod 0777 adhoc.sh
vim adhoc.sh
#I/bin/bash
ansible all -m yum_repository -a 'name=EPEL description=RHEL8
baseurl=https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rmp
gpgcheck=no enabled=yes'
```

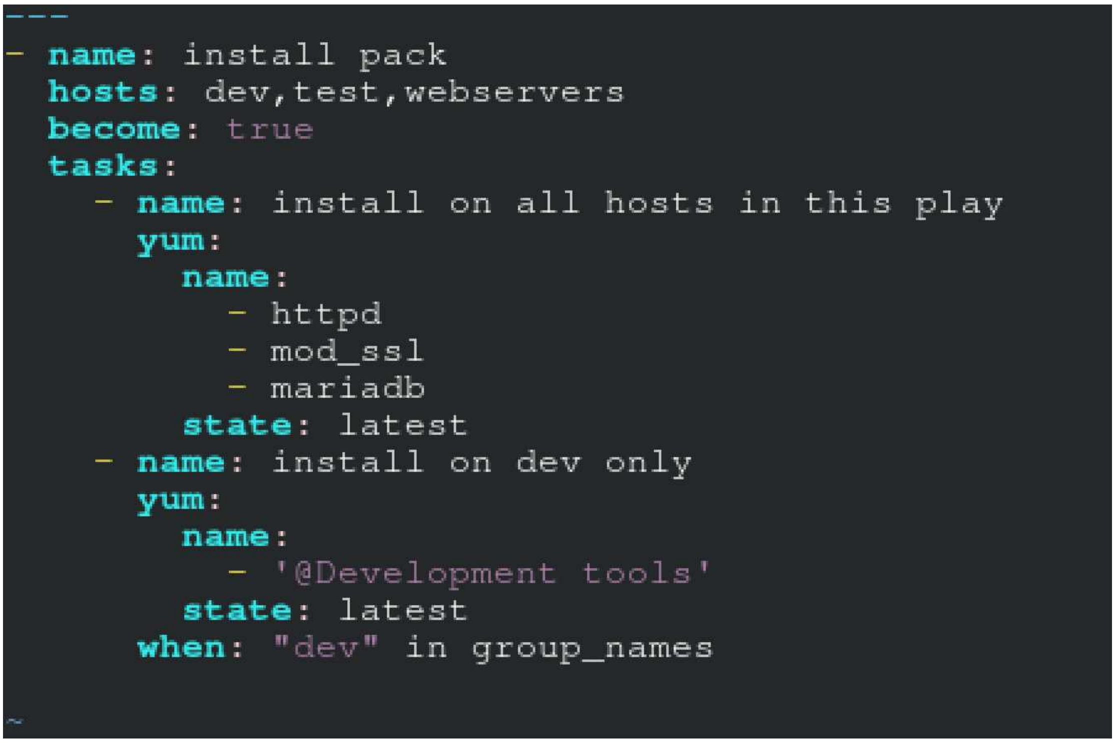
Question: 3

Create a file called packages.yml in /home/sandy/ansible to install some packages for the following hosts. On dev, prod and webservers install packages httpd, mod_ssl, and mariadb. On dev only install the development tools package. Also, on dev host update all the packages to the latest.

Answer: See the Explanation for complete Solution below.

Explanation:

Solution as:



```
---
- name: install pack
  hosts: dev,test,webservers
  become: true
  tasks:
    - name: install on all hosts in this play
      yum:
        name:
          - httpd
          - mod_ssl
          - mariadb
        state: latest
    - name: install on dev only
      yum:
        name:
          - '@Development tools'
        state: latest
      when: "dev" in group_names

~
```

** NOTE 1 a more acceptable answer is likely 'present' since it's not asking to install the latest
state: present
** NOTE 2 need to update the development node

```
- name: update all packages on development node
   yum:
     name: '*'
     state: latest
```

Question: 4

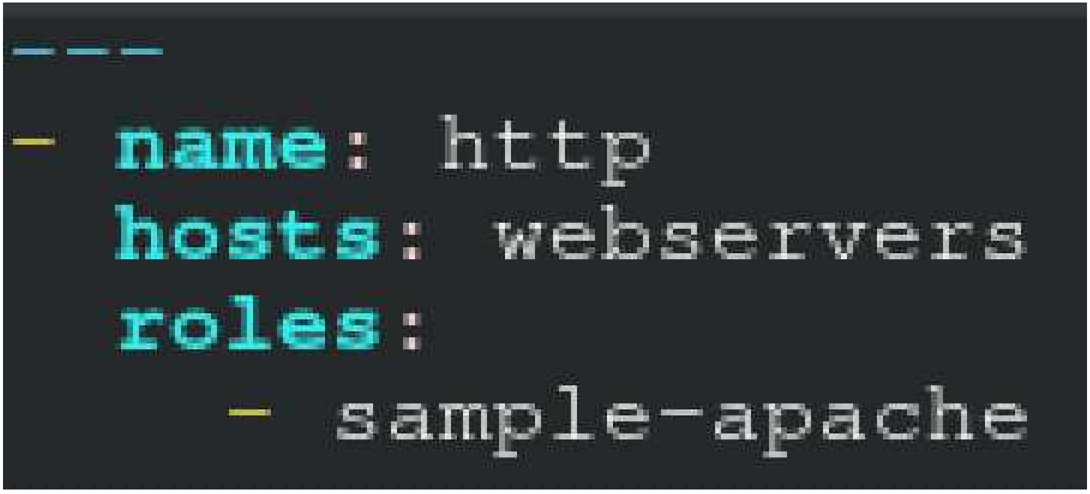
Create a role called sample-apache in /home/sandy/ansible/roles that enables and starts httpd, enables and starts the firewall and allows the webserver service. Create a template called index.html.j2 which creates and serves a message from /var/www/html/index.html Whenever the content of the file changes, restart the webserver service.

Welcome to [FQDN] on [IP]
Replace the FQDN with the fully qualified domain name and IP with the ip address of the node using ansible facts. Lastly, create a playbook in /home/sandy/ansible/ called apache.yml and use the role to serve the index file on webserver hosts.

Answer: See the Explanation for complete Solution below.

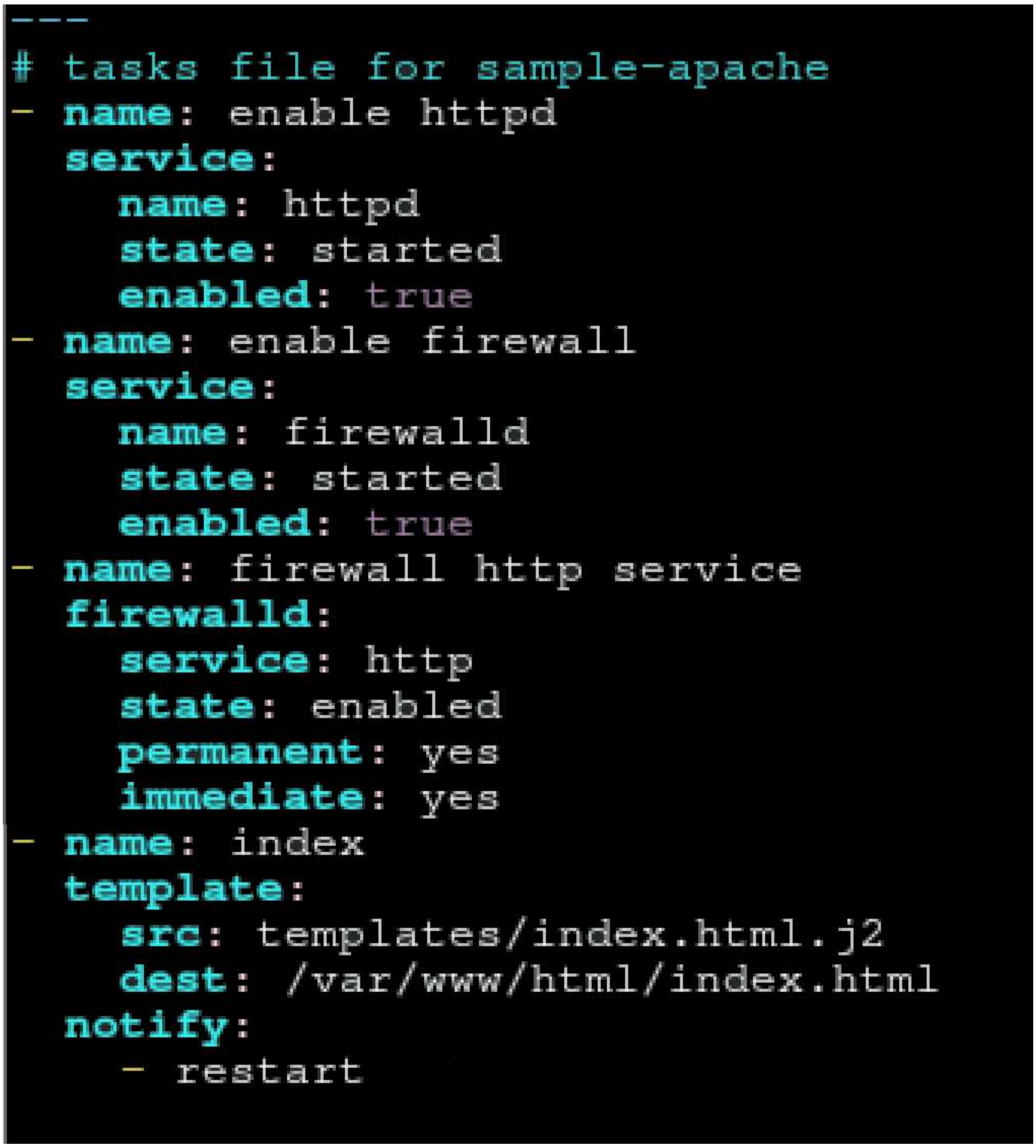
Explanation:

/home/sandy/ansible/apache.yml



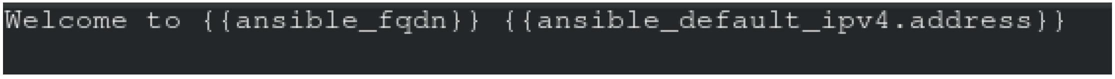
```
---
- name: http
  hosts: webservers
  roles:
    - sample-apache
```

/home/sandy/ansible/roles/sample-apache/tasks/main.yml



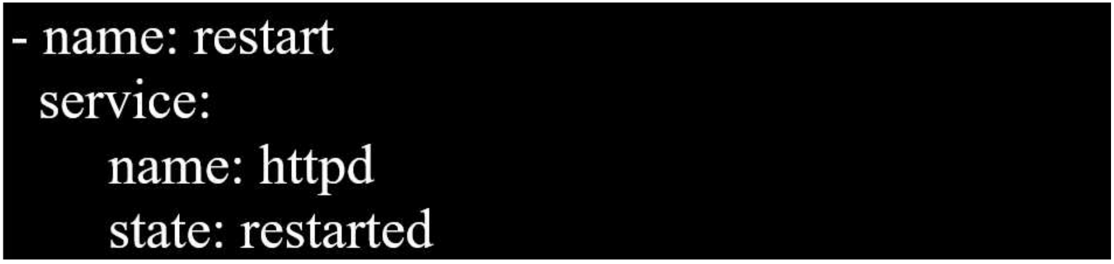
```
---
# tasks file for sample-apache
- name: enable httpd
  service:
    name: httpd
    state: started
    enabled: true
- name: enable firewall
  service:
    name: firewalld
    state: started
    enabled: true
- name: firewall http service
  firewalld:
    service: http
    state: enabled
    permanent: yes
    immediate: yes
- name: index
  template:
    src: templates/index.html.j2
    dest: /var/www/html/index.html
  notify:
    - restart
```

/home/sandy/ansible/roles/sample-apache/templates/index.html.j2



```
Welcome to {{ansible_fqdn}} {{ansible_default_ipv4.address}}
```

In /home/sandy/ansible/roles/sample-apache/handlers/main.yml



```
- name: restart
  service:
      name: httpd
      state: restarted
```

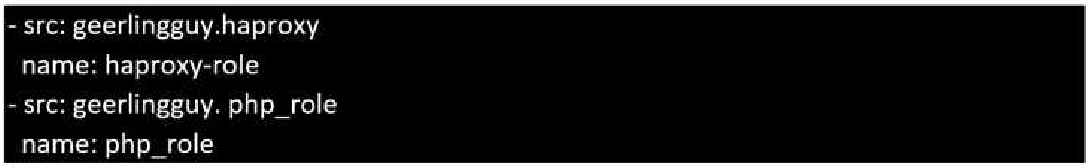
Question: 5

Create a file called requirements.yml in /home/sandy/ansible/roles to install two roles. The source for the first role is geerlingguy.haproxy and geerlingguy.php. Name the first haproxy-role and the second php-role. The roles should be installed in /home/sandy/ansible/roles.

Answer: See the Explanation for complete Solution below.

Explanation:

in /home/sandy/ansible/roles
vim requirements.yml



```
- src: geerlingguy.haproxy
  name: haproxy-role
- src: geerlingguy. php_role
  name: php_role
```

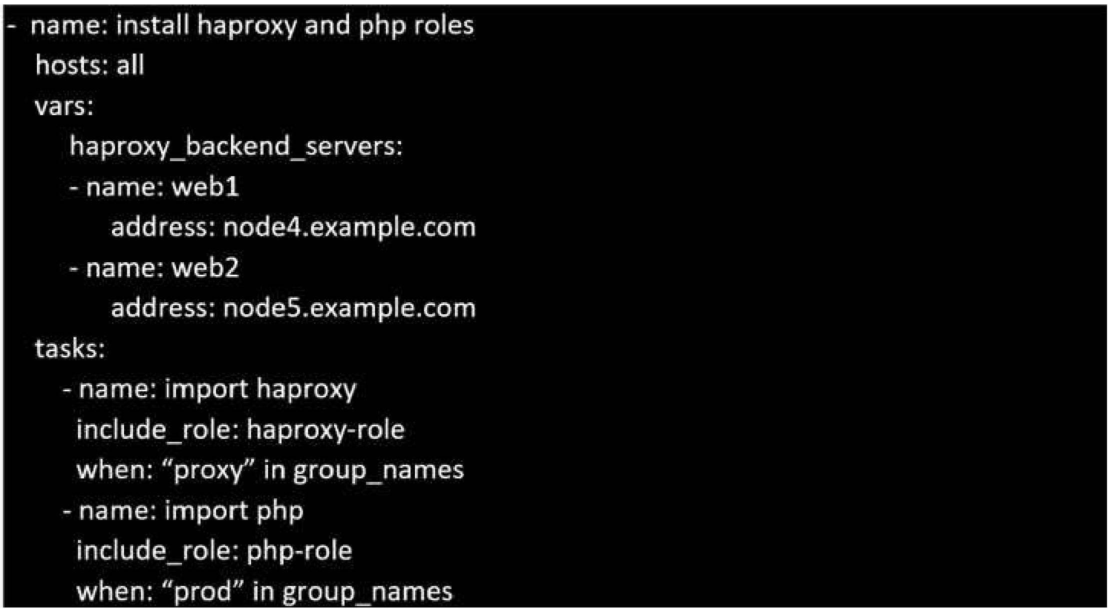
Run the requirements file from the roles directory:
ansible-galaxy install -r requirements.yml -p /home/sandy/ansible/roles

Question: 6

Create a file called requirements.yml in /home/sandy/ansible/roles a file called role.yml in /home/sandy/ansible/. The haproxy-role should be used on the proxy host. And when you curl http://node3.example.com it should display "Welcome to node4.example.com" and when you curl again "Welcome to node5.example.com" The php-role should be used on the prod host.

Answer: See the Explanation for complete Solution below.

Explanation:
Solution as:



```
-  name: install haproxy and php roles
   hosts: all
   vars:
      haproxy_backend_servers:
      - name: web1
          address: node4.example.com
      - name: web2
          address: node5.example.com
   tasks:
     - name: import haproxy
       include_role: haproxy-role
       when: “proxy” in group_names
     - name: import php
       include_role: php-role
       when: “prod” in group_names
```

Check the proxy host by curl http://node3.example.com

Question: 7

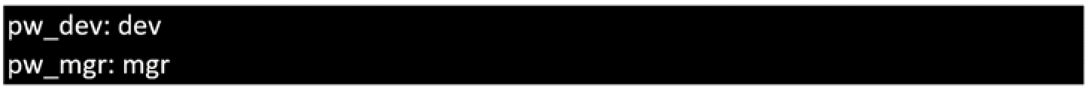
Create an ansible vault password file called lock.yml with the password reallysafepw in the /home/sandy/ansible directory. In the lock.yml file define two variables. One is pw_dev and the password is 'dev' and the other is pw_mgr and the password is 'mgr' Create a regular file called secret.txt which contains the password for lock.yml.

Answer: See the Explanation for complete Solution below.

Explanation:

ansible-vault create lock.yml
New Vault Password: reallysafepw
Confirm: reallysafepw

In file:



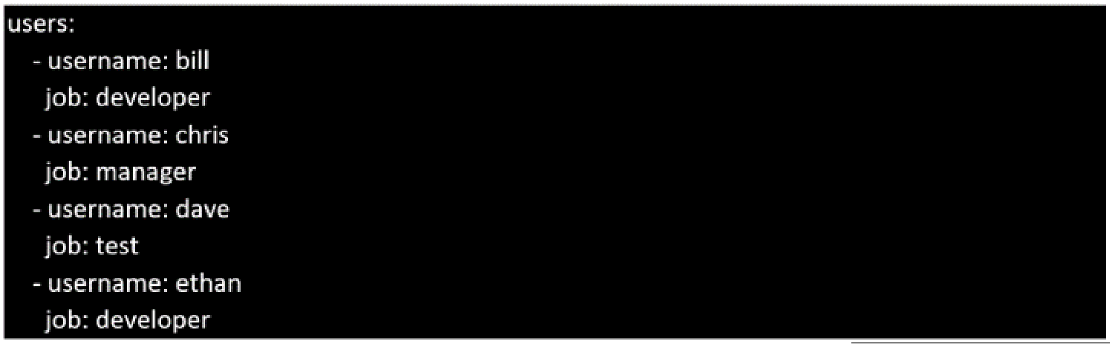
```
pw_dev: dev
pw_mgr: mgr
```

Question: 8

Create the users in the file usersjist.yml file provided. Do this in a playbook called users.yml located at /home/sandy/ansible. The passwords for these users should be set using the lock.yml file from TASK7. When running the playbook, the lock.yml file should be unlocked with secret.txt file from TASK 7.

All users with the job of 'developer' should be created on the dev hosts, add them to the group devops, their password should be set using the pw_dev variable. Likewise create users with the job of 'manager' on the proxy host and add the users to the group 'managers', their password should be set using the pw_mgr variable.

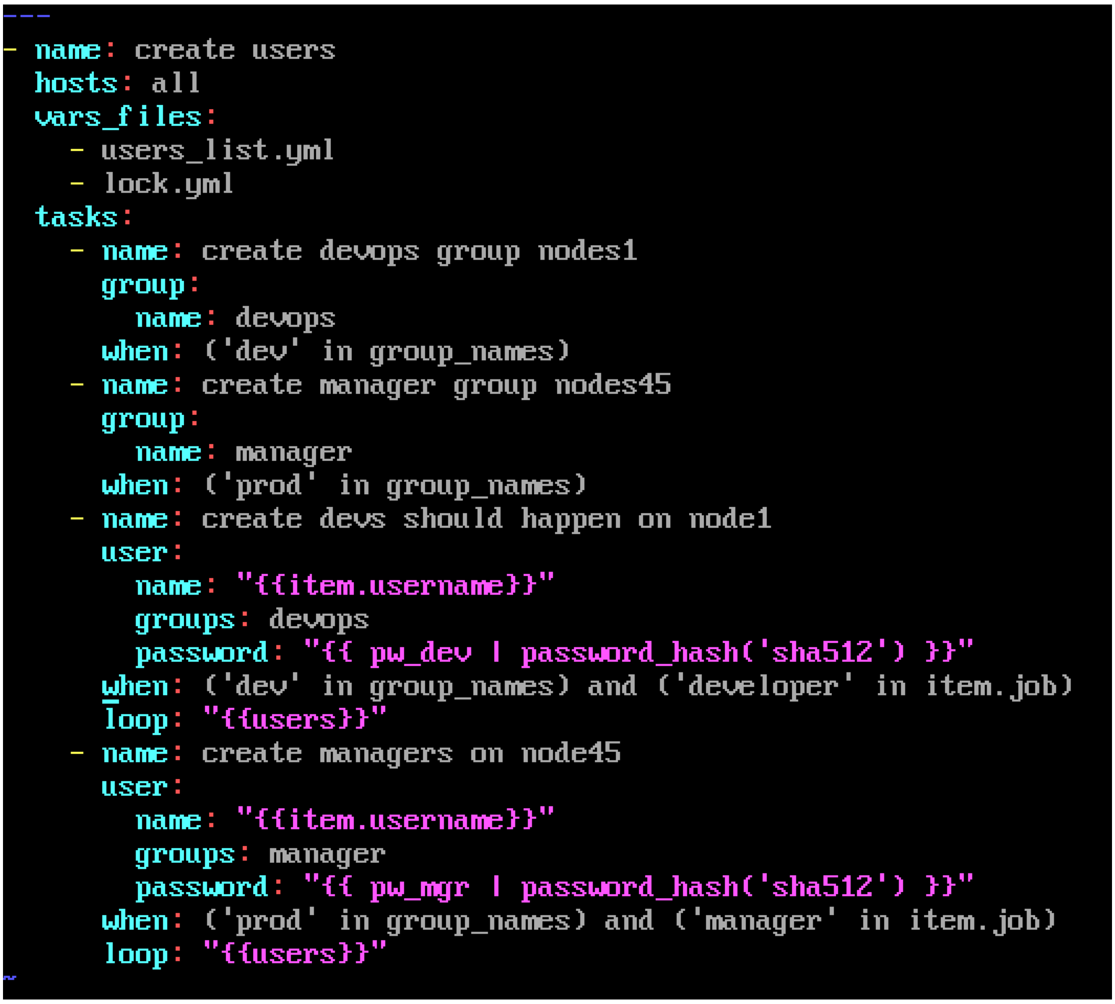
users_list.yml



```yaml
users:
  - username: bill
    job: developer
  - username: chris
    job: manager
  - username: dave
    job: test
  - username: ethan
    job: developer
```

Answer: See the Explanation for complete Solution below.

Explanation:
ansible-playbook users.yml –vault-password-file=secret.txt



```
---
- name: create users
  hosts: all
  vars_files:
    - users_list.yml
    - lock.yml
  tasks:
    - name: create devops group nodes1
      group:
        name: devops
      when: ('dev' in group_names)
    - name: create manager group nodes45
      group:
        name: manager
      when: ('prod' in group_names)
    - name: create devs should happen on node1
      user:
        name: "{{item.username}}"
        groups: devops
        password: "{{ pw_dev | password_hash('sha512') }}"
      when: ('dev' in group_names) and ('developer' in item.job)
      loop: "{{users}}"
    - name: create managers on node45
      user:
        name: "{{item.username}}"
        groups: manager
        password: "{{ pw_mgr | password_hash('sha512') }}"
      when: ('prod' in group_names) and ('manager' in item.job)
      loop: "{{users}}"
~
```

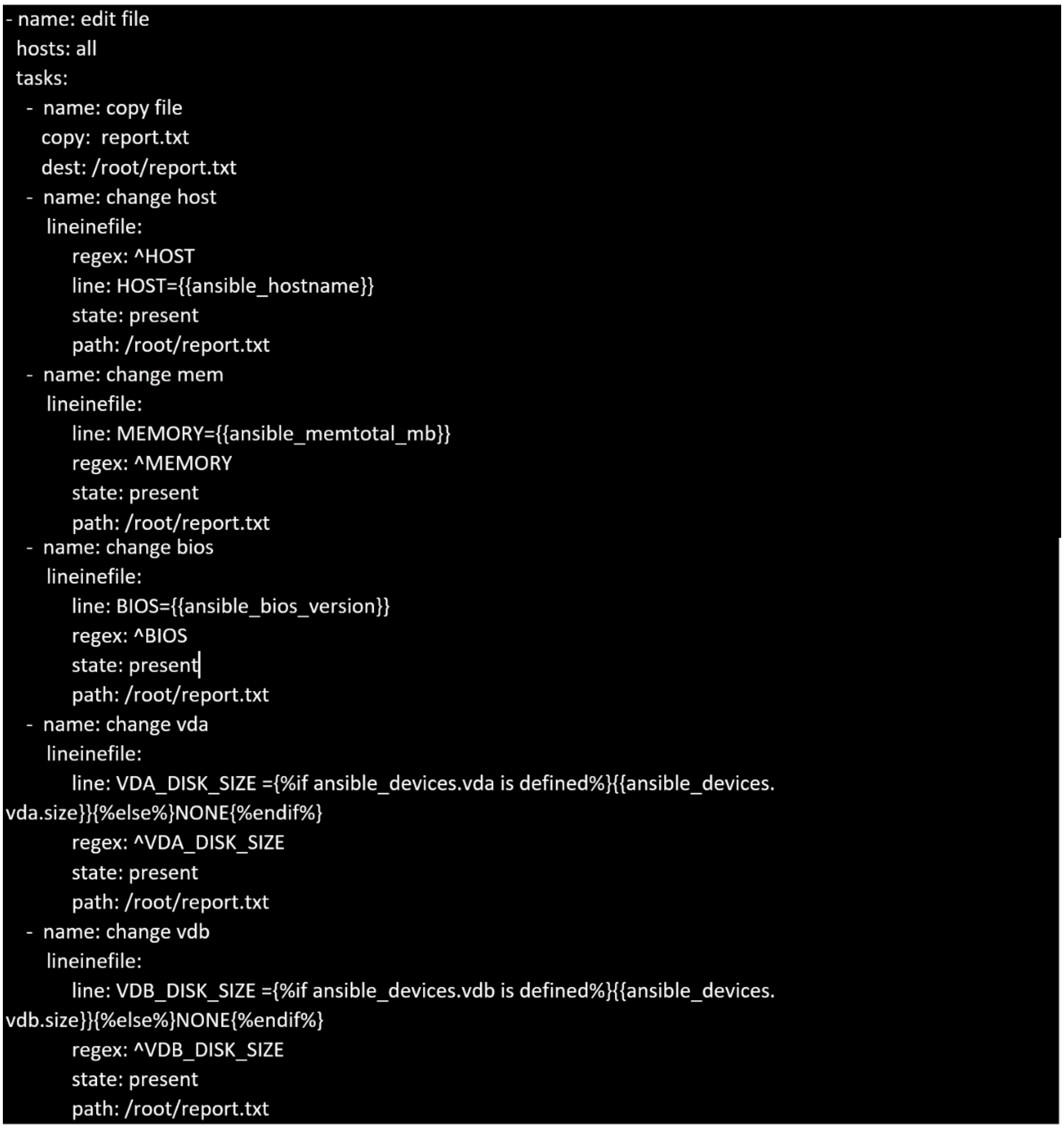
Question: 9

Create a file called specs.empty in home/bob/ansible on the local machine as follows:

HOST=
MEMORY=
BIOS=
VDA_DISK_SIZE=
VDB_DISK_SIZE=
Create the playbook /home/bob/ansible/specs.yml which copies specs.empty to all remote nodes' path /root/specs.txt. Using the specs.yml playbook then edit specs.txt on the remote machines to reflect the appropriate ansible facts.

Answer: See the Explanation for complete Solution below.

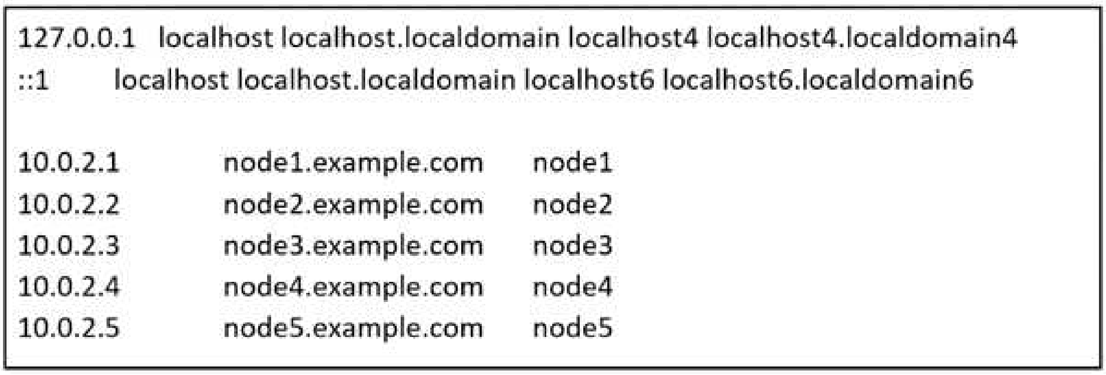
Explanation:
Solution as:



```
- name: edit file
 hosts: all
 tasks:
  - name: copy file
    copy:  report.txt
    dest: /root/report.txt
  - name: change host
    lineinefile:
        regex: ^HOST
        line: HOST={{ansible_hostname}}
        state: present
        path: /root/report.txt
  - name: change mem
    lineinefile:
        line: MEMORY={{ansible_memtotal_mb}}
        regex: ^MEMORY
        state: present
        path: /root/report.txt
  - name: change bios
    lineinefile:
        line: BIOS={{ansible_bios_version}}
        regex: ^BIOS
        state: present
        path: /root/report.txt
  - name: change vda
    lineinefile:
        line: VDA_DISK_SIZE ={%if ansible_devices.vda is defined%}{{ansible_devices.
vda.size}}{%else%}NONE{%endif%}
        regex: ^VDA_DISK_SIZE
        state: present
        path: /root/report.txt
  - name: change vdb
    lineinefile:
        line: VDB_DISK_SIZE ={%if ansible_devices.vdb is defined%}{{ansible_devices.
vdb.size}}{%else%}NONE{%endif%}
        regex: ^VDB_DISK_SIZE
        state: present
        path: /root/report.txt
```

Question: 10

Create a jinja template in /home/sandy/ansible/ and name it hosts.j2. Edit this file so it looks like the one below. The order of the nodes doesn't matter. Then create a playbook in /home/sandy/ansible called hosts.yml and install the template on dev node at /root/myhosts

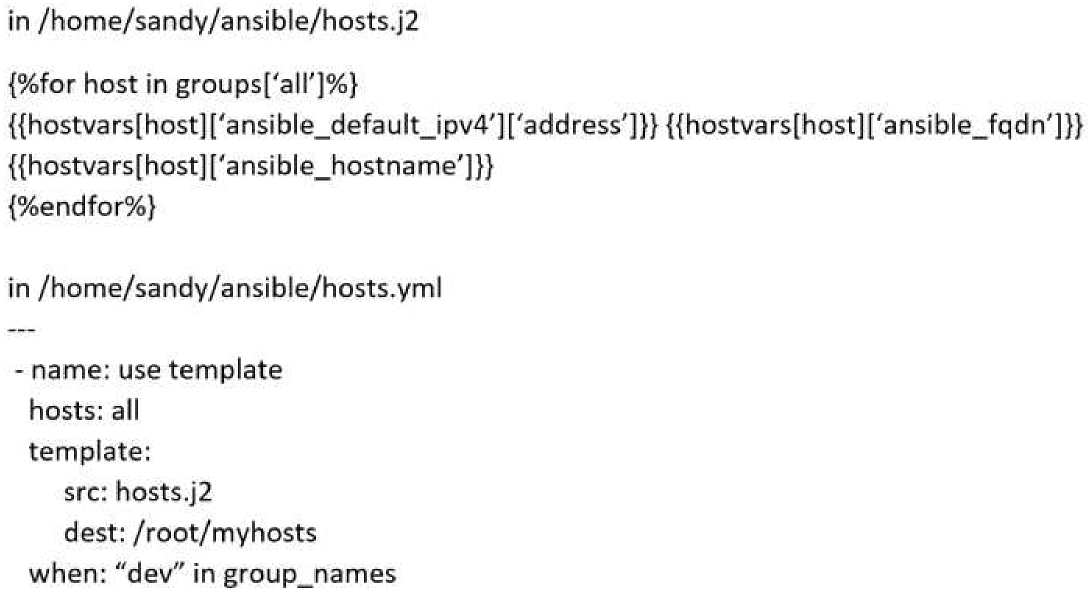


```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6

10.0.2.1         node1.example.com     node1
10.0.2.2         node2.example.com     node2
10.0.2.3         node3.example.com     node3
10.0.2.4         node4.example.com     node4
10.0.2.5         node5.example.com     node5
```

Answer: See the Explanation for complete Solution below.

Explanation:
Solution as:



```
in /home/sandy/ansible/hosts.j2

{%for host in groups['all']%}
{{hostvars[host]['ansible_default_ipv4']['address']}} {{hostvars[host]['ansible_fqdn']}}
{{hostvars[host]['ansible_hostname']}}
{%endfor%}

in /home/sandy/ansible/hosts.yml
---
 - name: use template
   hosts: all
   template:
      src: hosts.j2
      dest: /root/myhosts
   when: "dev" in group_names
```

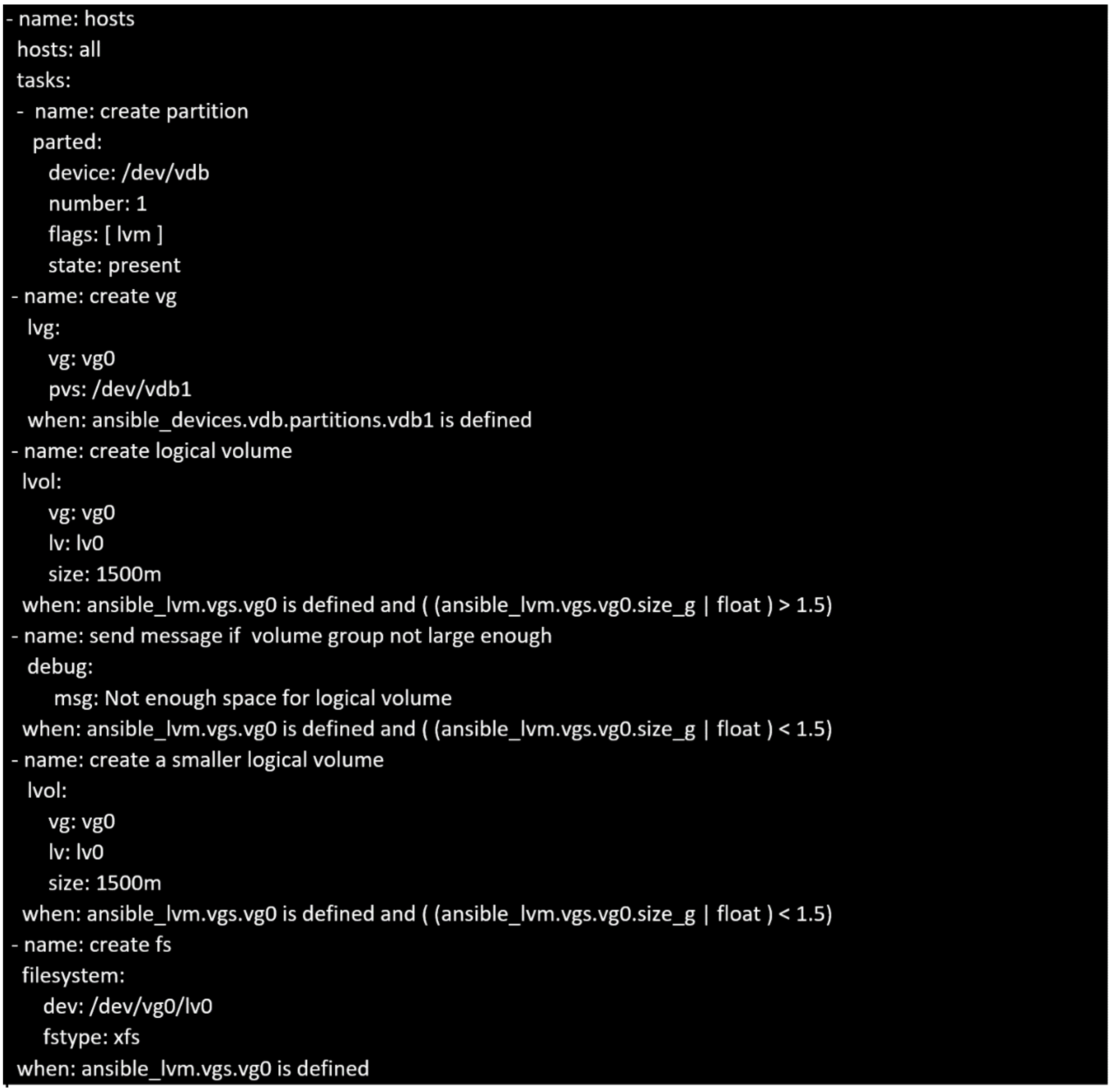
Question: 11

In /home/sandy/ansible/ create a playbook called logvol.yml. In the play create a logical volume called Iv0 and make it of size 1500MiB on volume group vgO If there is not enough space in the volume group print a message "Not enough space for logical volume" and then make a 800MiB Iv0 instead. If the volume group still doesn't exist, create a message "Volume group doesn't exist" Create an xfs filesystem on all Iv0 logical volumes. Don't mount the logical volume.

Answer: See the Explanation for complete Solution

below.

Explanation:
Solution as:



```
- name: hosts
 hosts: all
 tasks:
 -  name: create partition
    parted:
       device: /dev/vdb
       number: 1
       flags: [ lvm ]
       state: present
- name: create vg
   lvg:
       vg: vg0
       pvs: /dev/vdb1
   when: ansible_devices.vdb.partitions.vdb1 is defined
- name: create logical volume
  lvol:
       vg: vg0
       lv: lv0
       size: 1500m
  when: ansible_lvm.vgs.vg0 is defined and ( (ansible_lvm.vgs.vg0.size_g | float ) > 1.5)
- name: send message if  volume group not large enough
   debug:
        msg: Not enough space for logical volume
  when: ansible_lvm.vgs.vg0 is defined and ( (ansible_lvm.vgs.vg0.size_g | float ) < 1.5)
- name: create a smaller logical volume
   lvol:
       vg: vg0
       lv: lv0
       size: 1500m
  when: ansible_lvm.vgs.vg0 is defined and ( (ansible_lvm.vgs.vg0.size_g | float ) < 1.5)
- name: create fs
  filesystem:
      dev: /dev/vg0/lv0
      fstype: xfs
 when: ansible_lvm.vgs.vg0 is defined
```

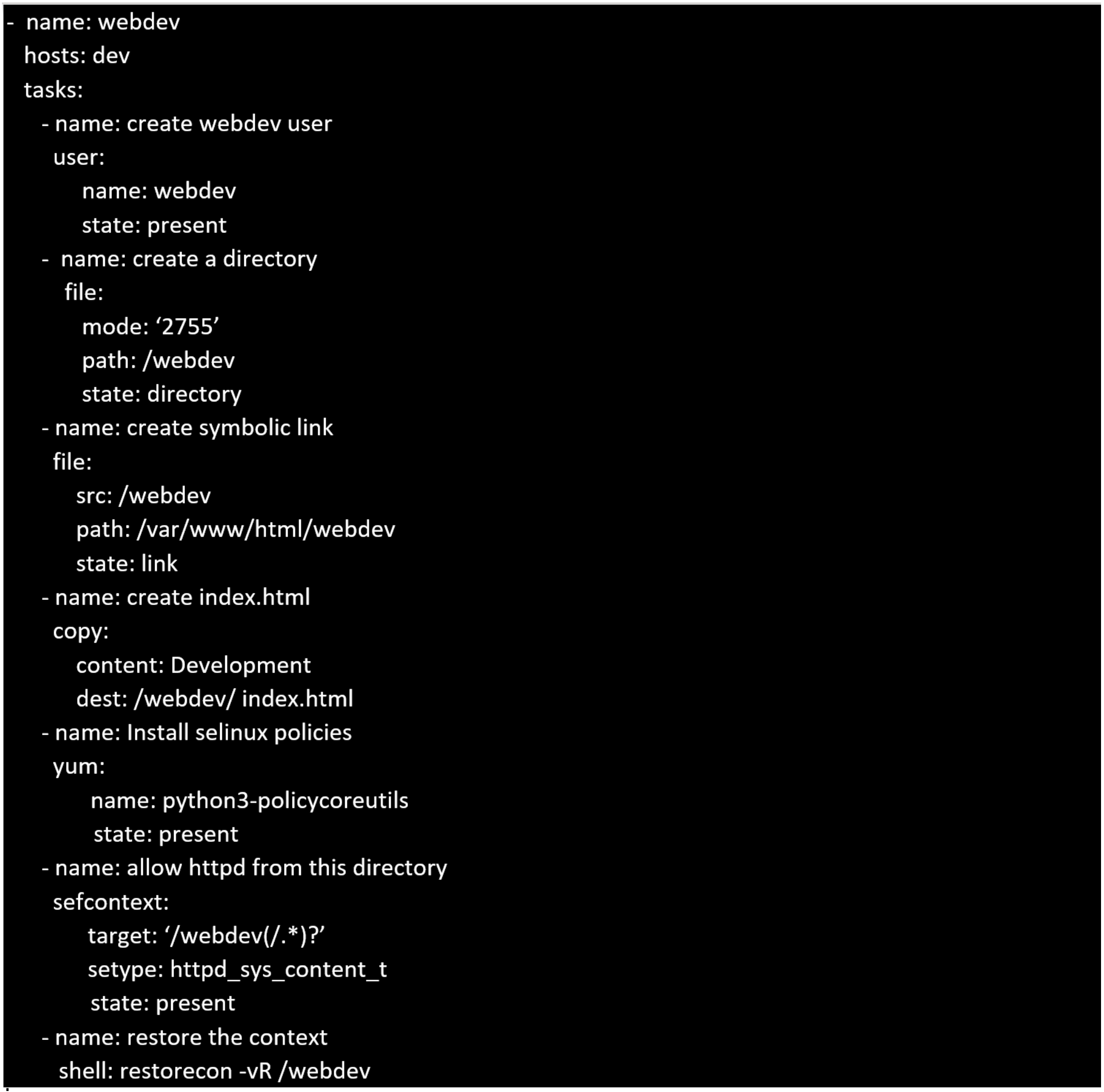
Question: 12

Create a playbook called webdev.yml in 'home/sandy/ansible. The playbook will create a directory Avcbdev on dev host. The permission of the directory are 2755 and owner is webdev. Create a symbolic link from /Webdev to /var/www/html/webdev. Serve a file from Avebdev7index.html which displays the text "Development" Curl http://node1.example.com/webdev/index.html to test

Answer: See the Explanation for complete Solution below.

Explanation:

Solution as:



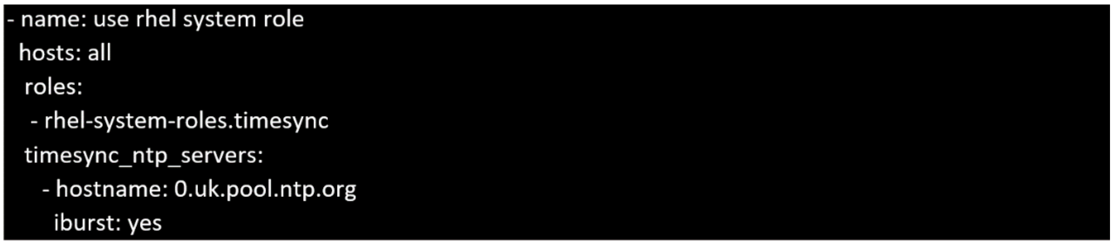
```
-  name: webdev
   hosts: dev
   tasks:
     - name: create webdev user
       user:
           name: webdev
           state: present
     -  name: create a directory
         file:
           mode: ‘2755’
           path: /webdev
           state: directory
     - name: create symbolic link
       file:
          src: /webdev
          path: /var/www/html/webdev
          state: link
     - name: create index.html
       copy:
          content: Development
          dest: /webdev/ index.html
     - name: Install selinux policies
       yum:
            name: python3-policycoreutils
            state: present
     - name: allow httpd from this directory
       sefcontext:
            target: ‘/webdev(/.*)?’
            setype: httpd_sys_content_t
            state: present
     - name: restore the context
        shell: restorecon -vR /webdev
```

Question: 13

Create a playbook called timesvnc.yml in /home/sandy/ansible using rhel system role timesync. Set the time to use currently configured nip with the server 0.uk.pool.ntp.org. Enable burst. Do this on all hosts.

Answer: See the Explanation for complete Solution below.

Explanation:
Solution as:



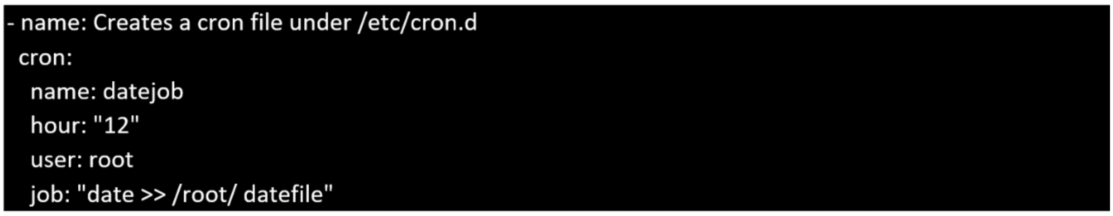
```
- name: use rhel system role
 hosts: all
  roles:
   - rhel-system-roles.timesync
  timesync_ntp_servers:
    - hostname: 0.uk.pool.ntp.org
      iburst: yes
```

Question: 14

Create a playbook called regulartasks.yml which has the system that append the date to /root/datefile every day at noon. Name is job 'datejob'

Answer: See the Explanation for complete Solution below.

Explanation:
Solution as:



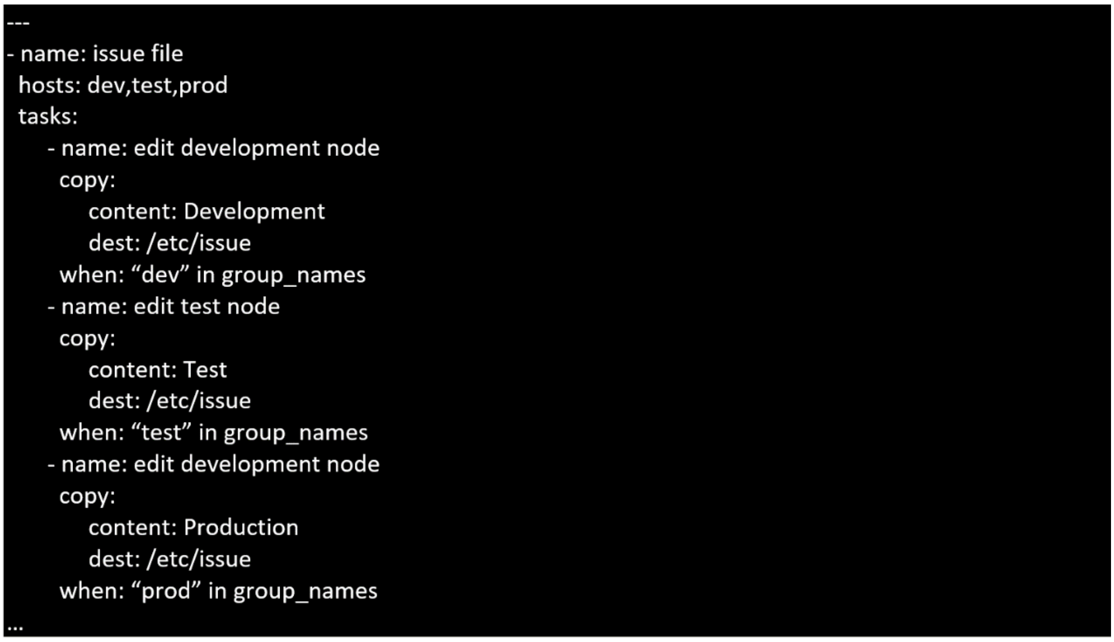
```
- name: Creates a cron file under /etc/cron.d
 cron:
   name: datejob
   hour: "12"
   user: root
   job: "date >> /root/ datefile"
```

Question: 15

Create a playbook called issue.yml in /home/sandy/ansible which changes the file /etc/issue on all managed nodes: If host is a member of (lev then write "Development" If host is a member of test then write "Test" If host is a member of prod then write "Production"

Answer: See the Explanation for complete Solution below.

Explanation:
Solution as:



```
---
- name: issue file
  hosts: dev,test,prod
  tasks:
      - name: edit development node
        copy:
            content: Development
            dest: /etc/issue
        when: “dev” in group_names
      - name: edit test node
        copy:
            content: Test
            dest: /etc/issue
        when: “test” in group_names
      - name: edit development node
        copy:
            content: Production
            dest: /etc/issue
        when: “prod” in group_names
...
```

Question: 16

Create an empty encrypted file called myvault.yml in /home/sandy/ansible and set the password to notsafepw. Rekey the password to iwejfj2221.

Answer: See the Explanation for complete Solution below.

Explanation:
ansible-vault create myvault.yml
Create new password: notsafepw Confirm password: notsafepw ansible-vault rekey myvault.yml
Current password: notsafepw New password: iwejfj2221 Confirm password: iwejfj2221

Question: 17

Create a playbook that changes the default target on all nodes to multi-user tarqet. Do this in playbook file called target.yml in /home/sandy/ansible

Answer: See the Explanation for complete Solution below.

Explanation:

```
- name: change default target
hosts: all
tasks:
- name: change target
```

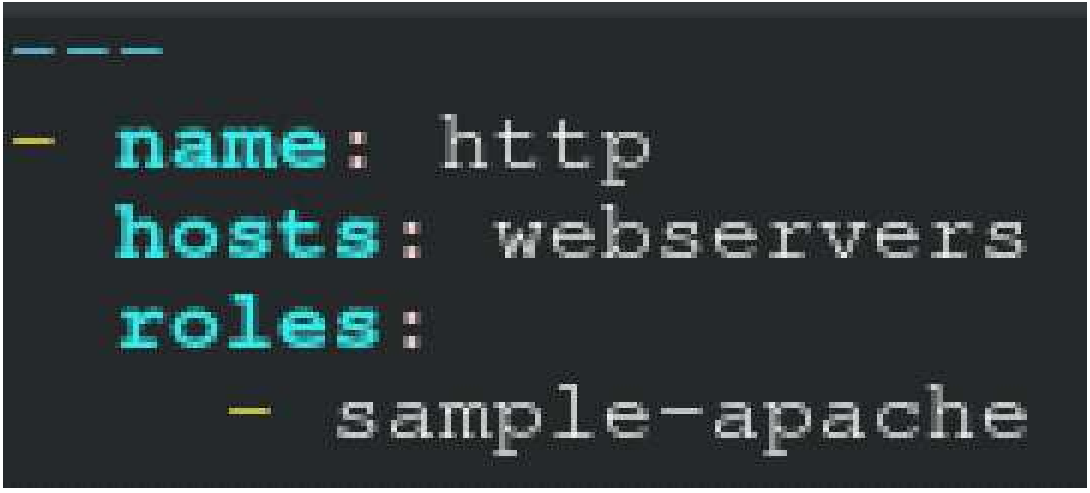
file:
src: /usr/lib/systemd/system/multi-user.target dest: /etc/systemd/system/default.target state: link

Question: 18

Create a playbook /home/bob /ansible/motd.yml that runs on all inventory hosts and docs the following: The playbook should replaee any existing content of/etc/motd in the following text. Use ansible facts to display the FQDN of each host
On hosts in the dev host group the line should be "Welcome to Dev Server FQDN".
On hosts in the webserver host group the line should be "Welcome to Apache Server FQDN".
On hosts in the database host group the line should be "Welcome to MySQL Server FQDN".

Answer: See the Explanation for complete Solution below.

Explanation:
/home/sandy/ansible/apache.yml



```
---
- name: http
  hosts: webservers
  roles:
    - sample-apache
```

/home/sandy/ansible/roles/sample-apache/tasks/main.yml

Question: 19

Install and configure ansible
User sandy has been created on your control node with the appropriate permissions already, do not change or modify ssh keys. Install the necessary packages to run ansible on the control node. Configure ansible.cfg to be in folder /home/sandy/ansible/ansible.cfg and configure to access remote machines via the sandy user. All roles should be in the path /home/sandy/ansible/roles. The inventory path should be in /home/sandy/ansible/invenlory.
You will have access to 5 nodes.

node1.example.com
node2.example.com

node3.example.com
node4.example.com
node5.example.com

Configure these nodes to be in an inventory file where node I is a member of group dev. nodc2 is a member of group test, node3 is a member of group proxy, nodc4 and node 5 are members of group prod. Also, prod is a member of group webservers.

Answer: See the Explanation for complete Solution below.

Explanation:

In/home/sandy/ansible/ansible.cfg

```
[defaults]
inventory=/home/sandy/ansible/inventory
roles_path=/home/sandy/ansible/roles
remote_user= sandy
host_key_checking=false
[privilegeescalation]
become=true
become_user=root
become_method=sudo
become_ask_pass=false
```

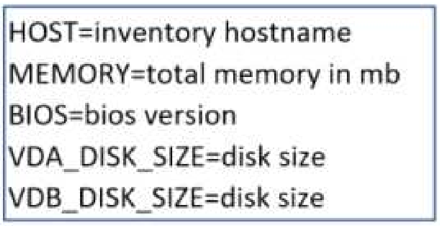
In /home/sandy/ansible/inventory

```
[dev]
node 1 .example.com
[test]
node2.example.com
[proxy]
node3 .example.com
[prod]
node4.example.com
node5 .example.com
[webservers:children]
prod
```

Question: 20

Create a file in /home/sandy/ansible/ called report.yml. Using this playbook, get a file called report.txt (make it look exactly as below). Copy this file over to all remote hosts at /root/report.txt. Then edit the lines in the file to provide the real information of the hosts. If a disk does not exist then write NONE.

report.txt

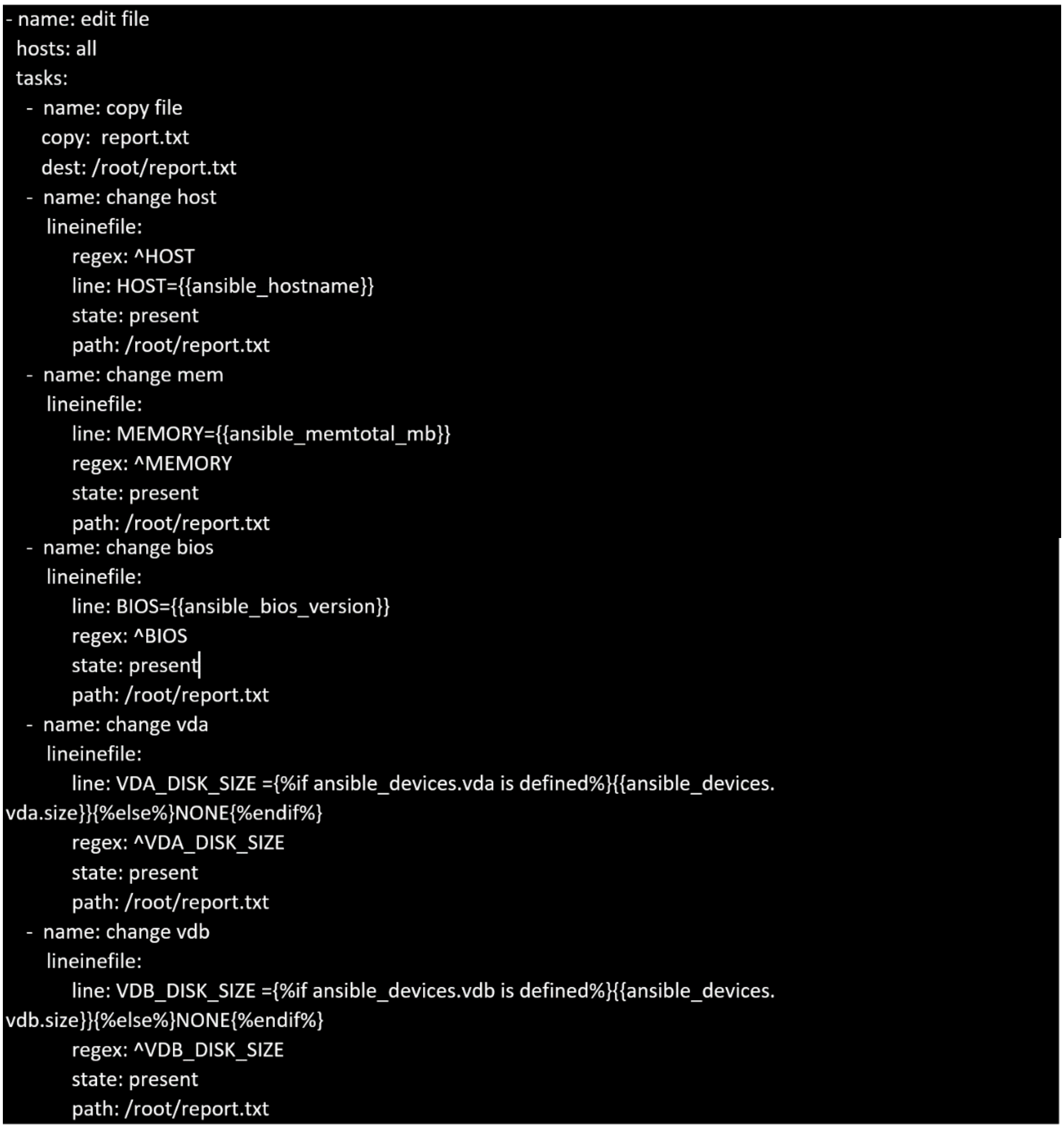


```
HOST=inventory hostname
MEMORY=total memory in mb
BIOS=bios version
VDA_DISK_SIZE=disk size
VDB_DISK_SIZE=disk size
```

Answer: See the Explanation for complete Solution below.

Explanation:

Solution as:



```
- name: edit file
  hosts: all
  tasks:
    - name: copy file
      copy:  report.txt
      dest: /root/report.txt
    - name: change host
      lineinefile:
          regex: ^HOST
          line: HOST={{ansible_hostname}}
          state: present
          path: /root/report.txt
    - name: change mem
      lineinefile:
          line: MEMORY={{ansible_memtotal_mb}}
          regex: ^MEMORY
          state: present
          path: /root/report.txt
    - name: change bios
      lineinefile:
          line: BIOS={{ansible_bios_version}}
          regex: ^BIOS
          state: present
          path: /root/report.txt
    - name: change vda
      lineinefile:
          line: VDA_DISK_SIZE ={%if ansible_devices.vda is defined%}{{ansible_devices.
vda.size}}{%else%}NONE{%endif%}
          regex: ^VDA_DISK_SIZE
          state: present
          path: /root/report.txt
    - name: change vdb
      lineinefile:
          line: VDB_DISK_SIZE ={%if ansible_devices.vdb is defined%}{{ansible_devices.
vdb.size}}{%else%}NONE{%endif%}
          regex: ^VDB_DISK_SIZE
          state: present
          path: /root/report.txt
```

Topic 2, LAB SETUP – 2

control.realmX.example.com _ workstation.lab.example.com
node1.realmX.example.com _ servera.lab.example.com
node2.realmX.example.com _ serverb.lab.example.com
node3.realmX.example.com _ serverc.lab.example.com
node4.realmX.example.com _ serverd.lab.example.com
node5.realmX.example.com
- username:root, password:redhat
- username:admin, password:redhat
note1. don't change 'root' or 'admin' password.
note2. no need to create ssh-keygen for access, its pre-defined
note3. SELinux is in enforcing mode and firewalld is disabled/stop on whole managed hosts.

Question: 21

Install and configure Ansible on the control-node control.realmX.example.com as follows:

--> Install the required packages
--> Create a static inventory file called /home/admin/ansible/inventory as follows:
node1.realmX.example.com is a member of the dev host group
node2.realmX.example.com is a member of the test host group
node3.realmX.example.com & node4.realmX.example.com are members of the prod host group
node5.realmX.example.com is a member of the balancers host group.
prod group is a member of the webservers host group
--> Create a configuration file called ansible.cfg as follows:
--> The host inventory file /home/admin/ansible/inventory is defined
--> The location of roles used in playbooks is defined as /home/admin/ansible/ roles

Answer: See the Explanation for complete Solution below.

Explanation:
Solution as:
Through physical host, login to workstation.lab.example.com with user root.

```
# ssh root@workstation.lab.example.com
# hostname
workstation.lab.example.com
# yum install platform-python*
# su - admin
# pwd
/home/admin/
# vim .vimrc
# mkdir -p ansible/roles
# cd ansible
# vim inventory
[dev]
servera.lab.example.com
[test]
serverb.example.com
[prod]
serverc.example.com

serverd.example.com
[balancer]
serverd.lab.example.com
[webservers:children]
prod
:!wq
# vim ansible.cfg
[defaults]
inventory = ./inventory
role_path = ./roles
remote_user = admin
ask_pass = false
[privilege_escalation]
become = true
become_method = sudo
become_user = root
become_ask_pass = false
:!wq
# ansible all -–list-hosts
```

Question: 22

Create and run an Ansible ad-hoc command.

--> As a system administrator, you will need to install software on the managed nodes.
--> Create a shell script called yum-pack.sh that runs an Ansible ad-hoc command to create yum-repository on each of the managed nodes as follows:
--> repository1
----
1. The name of the repository is EX407
2. The description is "Ex407 Description"
3. The base URL is http://content.example.com/rhel8.0/x86_64/dvd/BaseOS/
4. GPG signature checking is enabled
5. The GPG key URL is http://content.example.com/rhel8.0/x86_64/dvd/RPM-GPG-KEYredhat-release
6. The repository is enabled
--> repository2

----

1. The name of the repository is EXX407
2. The description is "Exx407 Description"
3. The base URL is http://content.example.com/rhel8.0/x86_64/dvd/AppStream/
4. GPG signature checking is enabled
5. The GPG key URL is http://content.example.com/rhel8.0/x86_64/dvd/ RPM-GPG-KEYredhat-release
6. The repository is enabled

Answer: See the Explanation for complete Solution below.

Explanation:
Solution as:

```
# pwd
/home/admin/ansible
# vim yum-pack.sh
#!/bin/bash
ansible all -m yum_repository -a 'name=EX407 description="Ex407 Description"
baseurl=http://content.example.com/rhel8.0/x86_64/dvd/BaseOS/ gpgcheck=yes
gpgkey=http://content.example.com/rhel8.0/x86_64/dvd/RPM-GPG-KEY-redhat-release
enabled=yes'

ansible all -m yum_repository -a 'name=EXX407 description="Exx407 Description"
baseurl=http://content.example.com/rhel8.0/x86_64/dvd/AppStream/ gpgcheck=yes
gpgkey=http://content.example.com/rhel8.0/x86_64/dvd/RPM-GPG-KEY-redhat-release
enabled=yes'
:!wq
# chmod +x yum-pack.sh
# bash yum-pack.sh
# ansible all -m command -a 'yum repolist all'
```

Question: 23

Create a playbook called packages.yml that:

--> Installs the php and mariadb packages on hosts in the dev, test, and prod host groups.
--> Installs the Development Tools package group on hosts in the dev host group.
--> Updates all packages to the latest version on hosts in the dev host group.

Answer: See the Explanation for complete Solution

below.

Explanation:

Solution as:

```
# pwd
home/admin/ansible/
# vim packages.yml
---
- name: Install the packages
hosts: dev,test,prod
vars:
- php_pkg: php
- mariadb_pkg: mariadb
tasks:
- name: install the packages
yum:
name:
- "{{ php_pkg }}"
- "{{ mariadb_pkg }}"
state: latest
- name: install the devops tool packages
hosts: dev
tasks:
- name: install devepment tools
yum:
name: "@Development Tools"
state: latest
- name: upgrade all the packages
yum:
name: "*"
state: latest
exclude: kernel*

!wq
# ansible-playbook package.yml –-syntax-check
# ansible-playbook package.yml
```

Question: 24

Install the RHEL system roles package and create a playbook called timesync.yml that:

--> Runs over all managed hosts.
--> Uses the timesync role.
--> Configures the role to use the time server 192.168.10.254 (Hear in redhat lab use "classroom.example.com")
--> Configures the role to set the iburst parameter as enabled.

Answer: See the

Explanation for complete Solution below.

Explanation:

Solution as:

```
# pwd
home/admin/ansible/
# sudo yum install rhel-system-roles.noarch -y
# cd roles/
# ansible-galaxy list
# cp -r /usr/share/ansible/roles/rhelsystem-roles.timesync .
# vim timesync.yml
---
- name: timesynchronization
hosts: all
vars:
timesync_ntp_provider: chrony
timesync_ntp_servers:
- hostname: classroom.example.com _ in exam its ip-address
iburst: yes
timezone: Asia/Kolkata
roles:
- rhel-system-roles.timesync

tasks:
- name: set timezone
timezone:
name: "{{ timezone }}"
:wq!
timedatectl list-timezones | grep india
# ansible-playbook timesync.yml --syntax-check
# ansible-playbook timesync.yml
# ansible all -m shell -a 'chronyc sources -v'
# ansible all -m shell -a 'timedatectl'
# ansible all -m shell -a 'systemctl is-enabled chronyd'
```

Question: 25

Create a role called apache in "/home/admin/ansible/roles" with the following requirements:

--> The httpd package is installed, enabled on boot, and started.
--> The firewall is enabled and running with a rule to allow access to the web server.
--> template file index.html.j2 is used to create the file /var/www/html/index.html with the output:
Welcome to HOSTNAME on IPADDRESS
--> Where HOSTNAME is the fqdn of the managed node and IPADDRESS is the IP-Address of

the managed node.
note: you have to create index.html.j2 file.
--> Create a playbook called httpd.yml that uses this role and the playbook runs on hosts in the webservers host group.

Answer: See the Explanation for complete Solution below.

Explanation:
Solution as:

```
----------
# pwd
/home/admin/ansible/roles/
# ansible-galaxy init apache
# vim apache/vars/main.yml
---
# vars file for apache
http_pkg: httpd
firewall_pkg: firewalld
http_srv: httpd
firewall_srv: firewalld
rule: http
webpage: /var/www/html/index.html
template: index.html.j2
:wq!
# vim apache/tasks/package.yml
---
- name: Installing packages
yum:
name:
- "{{http_pkg}}"
- "{{firewall_pkg}}"
state: latest
:wq!
# vim apache/tasks/service.yml
---
- name: start and enable http service
service:
name: "{{http_srv}}"
enabled: true
state: started
- name: start and enable firewall service
service:
name: "{{firewall_srv}}"
enabled: true
state: started
```

```
:wq!
# vim apache/tasks/firewall.yml
---
- name: Adding http service to firewall
firewalld:
service: "{{rule}}"
state: enabled
permanent: true
immediate: true
:wq!
# vim apache/tasks/webpage.yml
---
- name: creating template file
template:
src: "{{template}}"
dest: "{{webpage}}"
notify: restart_httpd
!wq
# vim apache/tasks/main.yml

# tasks file for apache
- import_tasks: package.yml
- import_tasks: service.yml
- import_tasks: firewall.yml
- import_tasks: webpage.yml
:wq!
# vim apache/templates/index.html.j2
Welcome to {{ ansible_facts.fqdn }} on {{ ansible_facts.default_ipv4.address }}
# vim apache/handlers/main.yml
---
# handlers file for apache
- name: restart_httpd
service:
name: httpd
state: restarted
:wq!
# cd ..
# pwd
/home/admin/ansible/
# vim httpd.yml
---
- name: Including apache role
hosts: webservers
pre_tasks:
- name: pretask message
debug:
msg: 'Ensure webserver configuration'
roles:
```

```
- ./roles/apache
post_tasks:
- name: Check webserver
uri:
url: "http://{{ ansible_facts.default_ipv4.address }}"
return_content: yes
status_code: 200
:wq!
# ansible-playbook httpd.yml –-syntax-check
# ansible-playbook httpd.yml
# curl http://serverx
```

Question: 26

Use Ansible Galaxy with a requirements file called /home/admin/ansible/roles/install.yml to download and install roles to /home/admin/ansible/roles from the following URLs:

http:// classroom.example.com /role1.tar.gz The name of this role should be balancer
http:// classroom.example.com /role2.tar.gz The name of this role should be phphello

Answer: See the Explanation for complete Solution below.

Explanation:
Solution as:

```
# pwd

/home/admin/ansible/roles
# vim install.yml
---
- src: http://classroom.example.com/role1.tar.gz
name: balancer
- src: http://classroom.example.com/role2.tar.gz
name: phphello
:wq!
# pwd
/home/admin/ansible
# ansible-galaxy install -r roles/install.yml -p roles
```

Question: 27

Create a playbook called balance.yml as follows:

* The playbook contains a play that runs on hosts in balancers host group and uses

the balancer role.
--> This role configures a service to loadbalance webserver requests between hosts in the webservers host group.curl
--> When implemented, browsing to hosts in the balancers host group (for example http://node5.example.com) should produce the following output:
Welcome to node3.example.com on 192.168.10.z
--> Reloading the browser should return output from the alternate web server:
Welcome to node4.example.com on 192.168.10.a
* The playbook contains a play that runs on hosts in webservers host group and uses the phphello role.
--> When implemented, browsing to hosts in the webservers host group with the URL / hello.php should produce the following output:
Hello PHP World from FQDN
--> where FQDN is the fully qualified domain name of the host. For example, browsing to http://node3.example.com/hello.php, should produce the following output:
Hello PHP World from node3.example.com
* Similarly, browsing to http://node4.example.com/hello.php, should produce the following output:
Hello PHP World from node4.example.com

Answer: See the Explanation for complete Solution below.

Explanation:
Solution as:

```
# pwd
/home/admin/ansible/
# vim balancer.yml
---
- name: Including phphello role
hosts: webservers
roles:
- ./roles/phphello
- name: Including balancer role
hosts: balancer

roles:
- ./roles/balancer
:wq!
# ansible-playbook balancer.yml --syntax-check
# ansible-playbook balancer.yml
```

Question: 28

Create a playbook called web.yml as follows:

* The playbook runs on managed nodes in the "dev" host group
* Create the directory /webdev with the following requirements:
--> membership in the apache group
--> regular permissions: owner=r+w+execute, group=r+w+execute, other=r+execute
s.p=set group-id
* Symbolically link /var/www/html/webdev to /webdev
* Create the file /webdev/index.html with a single line of text that reads:
“Development”
--> it should be available on http://servera.lab.example.com/webdev/index.html

Answer: See the Explanation for complete Solution below.

Explanation:
Solution as:

```
# pwd
/home/admin/ansible/
# vim web.yml
---
- name:
hosts: dev
tasks:
- name: create group
yum:
name: httpd
state: latest
- name: create group
group:
name: apache
state: present
- name: creating directiory
file:
path: /webdev
state: directory
mode: '2775'
group: apache
- sefcontext:
target: '/webdev/index.html'
setype: httpd_sys_content_t
state: present
- name: Apply new SELinux file context to filesystem
command: restorecon -irv
- name: creating symbolic link
file:
src: /webdev
```

```
dest: /var/www/html/webdev
state: link
force: yes
- name: creating file
file:
path: /webdev/index.html
sate: touch
- name: Adding content to index.html file
copy:
dest: /webdev/index.html
content: "Development"
- name: add service to the firewall
firewalld:
service: http
permanent: yes
state: enabled
immediate: yes
- name: active http service
service:
name: httpd
state: restarted
enabled: yes
:wq
# ansible-playbook web.yml -—syntax-check
# ansible-playbook web.yml
```

Question: 29

Create an Ansible vault to store user passwords as follows:

* The name of the vault is valut.yml
* The vault contains two variables as follows:
- dev_pass with value wakennym
- mgr_pass with value rocky
* The password to encrypt and decrypt the vault is atenorth
* The password is stored in the file /home/admin/ansible/password.txt

Answer: See the Explanation for complete Solution below.

Explanation:
Solution as:

```
# pwd
/home/admin/ansible
# echo "atenorth" >password.txt
# chmod 0600 password.txt
# ansible-vault create vault.yml --vault-password-file=password.txt
---
- dev_pass: wakennym
- mgr_pass: rocky
:wq
# cat vault.yml
$ANSIBLE_VAULT;1.1;AES256
36383862376164316436353665343765643331393433373564613762666531313034336438353662
34643463313464613063376336323935636435313761396 10a34353132613066326661353363356 2
38623439316631306463623761343939373263333134353264333834353264343934373765643737
35353036306266663 70a643663366634383863393338616661666632353139306436316430616334
65386134393363643133363738656130636532346431376265613066326162643437643064313863
66333335373033343334376461633436666661323166393765 31
# ansible-vault view vault.yml
password:******
---
- dev_pass: wakennym
- mgr_pass: rocky
```

Question: 30

Generate a hosts file:

* Download an initial template file hosts.j2 from http://classroom.example.com/
hosts.j2 to
/home/admin/ansible/ Complete the template so that it can be used to generate a file
with a
line for each inventory host in the same format as /etc/hosts:
172.25.250.9 workstation.lab.example.com workstation
* Create a playbook called gen_hosts.yml that uses this template to generate the file
/etc/myhosts on hosts in the dev host group.
* When completed, the file /etc/myhosts on hosts in the dev host group should have a
line for
each managed host:
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
172.25.250.10 serevra.lab.example.com servera
172.25.250.11 serevrb.lab.example.com serverb
172.25.250.12 serevrc.lab.example.com serverc
172.25.250.13 serevrd.lab.example.com serverd
----
while practising you to create these file hear. But in exam have to download as per
questation.
hosts.j2 file consists.
localhost localhost.localdomain localhost4 localhost4.localdomain4
::1
localhost localhost.localdomain localhost6 localhost6.localdomain6

Answer: See the Explanation for complete Solution below.

Explanation:

Solution as:

```
# pwd
/home/admin/ansible
# wget http://classroom.example.com/hosts.j2
# vim hosts.j2
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_facts']['default_ipv4']['address'] }} {{ hostvars[host]
['ansible_facts']['fqdn'] }} {{ hostvars[host]['ansible_facts']['hostname'] }}
{% endfor %}

:wq!
# vim gen_hosts.yml
---
- name: collecting all host information
hosts: all
tasks:
- name:
template:
src: hosts.j2
dest: /etc/myhosts
when: inventory_hostname in groups['dev']
:wq
# ansible-playbook gen_hosts.yml -—syntax-check
# ansible-playbook gen_hosts.yml
```

Question: 31

Create a playbook called hwreport.yml that produces an output file called /root/hwreport.txt on all managed nodes with the following information:

--> Inventory host name
--> Total memory in MB
--> BIOS version
--> Size of disk device vda
--> Size of disk device vdb

Each line of the output file contains a single key-value pair.

* Your playbook should:

--> Download the file hwreport.empty from the URL http://classroom.example.com/
hwreport.empty and
save it as /root/hwreport.txt
--> Modify with the correct values.
note: If a hardware item does not exist, the associated value should be set to NONE
----
while practising you to create these file hear. But in exam have to download as per questation.
hwreport.txt file consists.

```
my_sys=hostname
my_BIOS=biosversion
my_MEMORY=memory
my_vda=vdasize
my_vdb=vdbsize
```

Answer: See the Explanation for complete Solution below.

Explanation:
Solution as:

```
# pwd
/home/admin/ansible
# vim hwreport.yml

- name:
hosts: all
ignore_errors: yes
tasks:
- name: download file
get_url:
url: http://classroom.example.com/content/ex407/hwreport.empty
dest: /root/hwreport.txt
- name: vdasize
replace:
regexp: "vdasize"
replace: "{{ ansible_facts.devices.vda.size }}"
dest: /root/hwreport.txt
register: op1
- debug:
var: op1
- name: none
replace:
regexp: "vdasize"
replace: NONE
dest: /root/hwreport.txt
```

```
when:
op1.failed == true
- name: vdbsize
replace:
regexp: "vdbsize"
replace: "{{ ansible_facts.devices.vdb.size }}"
dest: /root/hwreport.txt
register: op2
- debug:
var: op2
- name: none
replace:
regexp: "vdbsize"
replace: NONE
dest: /root/hwreport.txt
when:
op2.failed == true
- name: sysinfo
replace:
regexp: "{{item.src}}"
replace: "{{item.dest}}"
dest: /root/hwreport.txt
loop:
- src: "hostname"
dest: "{{ ansible_facts.fqdn }}"
- src: "biosversion"
dest: "{{ ansible_facts.bios_version }}"
- src: "memory"
dest: "{{ ansible_facts.memtotal_mb }}"
:wq!
# ansible-playbook hwreport.yml -–syntax-check
# ansible-playbook hwreport.yml
```

Question: 32

Modify file content.

Create a playbook called /home/admin/ansible/modify.yml as follows:

* The playbook runs on all inventory hosts

* The playbook replaces the contents of /etc/issue with a single line of text as follows:

--> On hosts in the dev host group, the line reads: “Development”

--> On hosts in the test host group, the line reads: “Test”

--> On hosts in the prod host group, the line reads: “Production”

Answer: See the Explanation for

complete Solution below.

Explanation:

Solution as:

```
# pwd
/home/admin/ansible
# vim modify.yml
---
- name:
hosts: all
tasks:
- name:
copy:
content: "Development"
dest: /etc/issue
when: inventory_hostname in groups['dev']
- name:
copy:
content: "Test"
dest: /etc/issue
when: inventory_hostname in groups['test']

- name:
copy:
content: "Production"
dest: /etc/issue
when: inventory_hostname in groups['prod']
:wq
# ansible-playbook modify.yml –-syntax-check
# ansible-playbook modify.yml
```

Question: 33

Rekey an existing Ansible vault as follows:

----

* Download Ansible vault from http:// classroom.example.com /secret.yml to /home/admin/ansible/

* The current vault password is curabete

* The new vault password is newvare

* The vault remains in an encrypted state with the new password

Answer: See the Explanation for complete Solution below.

Explanation:

Solution as:

```
# pwd
/home/admin/ansible/
# wget http://classroom.example.com/secret.yml
# chmod 0600 newpassword.txt
# ansible-vault rekey vault.yml --new-vault-password-file=newpassword.txt
```

Question: 34

Create user accounts

--> A list of users to be created can be found in the file called user_list.yml
which you should download from http://classroom.example.com/user_list.yml and
save to /home/admin/ansible/
--> Using the password vault created elsewhere in this exam, create a playbook called
create_user.yml
that creates user accounts as follows:
--> Users with a job description of developer should be:
--> created on managed nodes in the "dev" and "test" host groups assigned the
password from the "dev_pass"
variable and these user should be member of supplementary group "devops".
--> Users with a job description of manager should be:
--> created on managed nodes in the "prod" host group assigned the password from
the "mgr_pass" variable
and these user should be member of supplementary group "opsmgr"
--> Passwords should use the "SHA512" hash format. Your playbook should work using
the vault password file
created elsewhere in this exam.

while practising you to create these file hear. But in exam have to download as per questation.
user_list.yml file consist:

```
---
user:
- name: user1
job: developer
- name: user2
job: manager
```

Answer: See the Explanation for complete Solution below.

Explanation:
Solution as:

```
# pwd
/home/admin/ansible
# wget http://classroom.example.com/user_list.yml
# cat user_list.yml
# vim create_user.yml
---
- name:
hosts: all
vars_files:
- ./user_list.yml
- ./vault.yml
tasks:
- name: creating groups
group:
name: "{{ item }}"

state: present
loop:
- devops
- opsmgr
- name: creating user
user:
name: "{{ item.name }}"
state: present
groups: devops
password: "{{ dev_pass|password_hash ('sha512') }}"
loop: "{{ user }}"
when: (inventory_hostname in groups['dev'] or inventory_hostname in
groups['test']) and item.job == "developer"
- name: creating user
user:
name: "{{ item.name }}"
state: present
groups: opsmgr
password: "{{ mgr_pass|password_hash ('sha512') }}"
loop: "{{ user }}"
when: inventory_hostname in groups['prod'] and item.job == "manager"
:wq!
# ansible-playbook create_user.yml -—vault-password-file=password.txt -—syntax-check
# ansible-playbook create_user.yml -—vault-password-file=password.txt
```

Question: 35

Create Logical volumes with lvm.yml in all nodes according to following requirements.

----

* Create a new Logical volume named as 'data'
* LV should be the member of 'research' Volume Group

* LV size should be 1500M
* It should be formatted with ext4 file-system.
--> If Volume Group does not exist then it should print the message "VG Not found"
--> If the VG can not accommodate 1500M size then it should print "LV Can not be created with
following size", then the LV should be created with 800M of size.
--> Do not perform any mounting for this LV.

Answer: See the Explanation for complete Solution below.

Explanation:
Solution as:

```
# pwd
/home/admin/ansible
# vim lvm.yml
---
- name:
hosts: all
ignore_errors: yes
tasks:
- name:
lvol:
lv: data
vg: research
size: "1500"
- debug:

msg: "VG Not found"
when: ansible_lvm.vgs.research is not defined
- debug:
msg: "LV Can not be created with following size"
when: ansible_lvm.vgs.research.size_g < "1.5"
- name:
lvol:
lv: data
vg: research
size: "800"
when: ansible_lvm.vgs.research.size_g < "1.5"
- name:
filesystem:
fstype: ext4
dev: /dev/research/data
:wq!
# ansible-playbook lvm.yml –-syntax-check
# ansible-playbook lvm.yml
```