Thank You for your purchase

RedHat EX294 Exam Question & Answers

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

Product Questions: 20
Version: 4.0

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.

Solution

```
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.

Solution

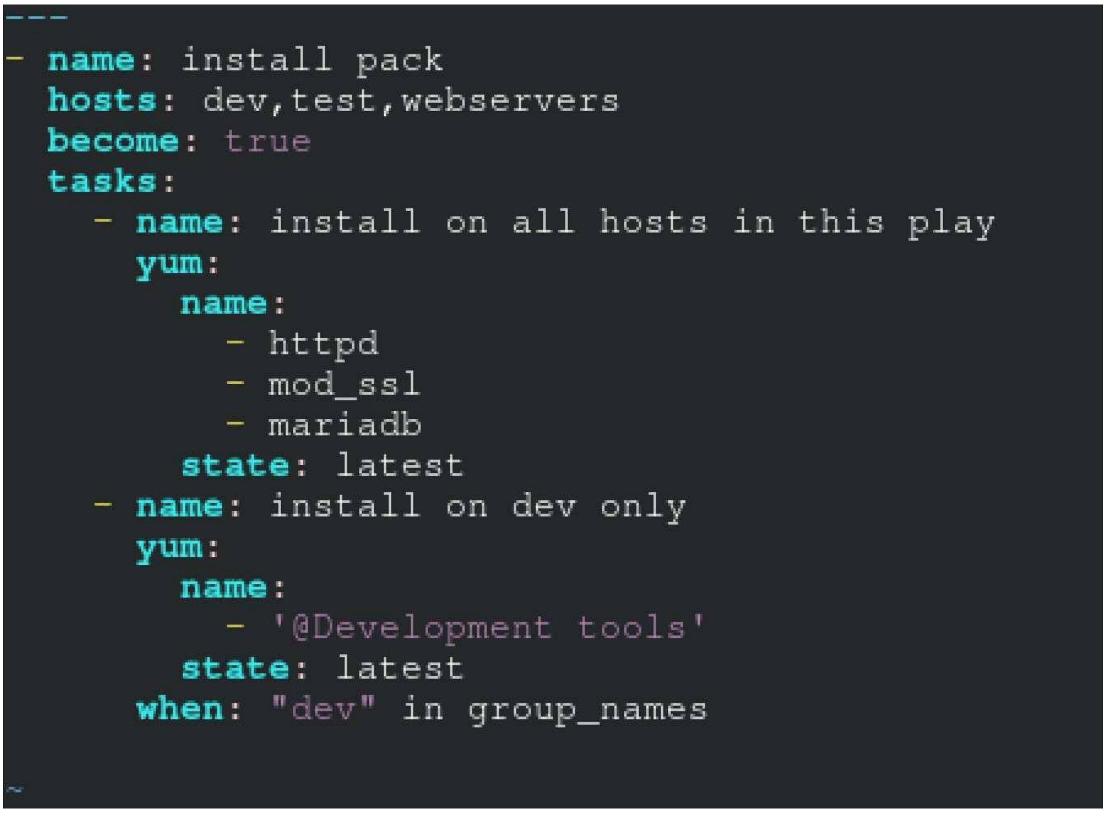
```
chmod 0777 adhoc.sh
vim adhoc.sh
#I/bin/bash
ansible all -m yum_repository -a 'name=EPEL description=RHEL8
baseurl=https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rmp
gpgcheck=no enabled=yes'
```

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.

Solution

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

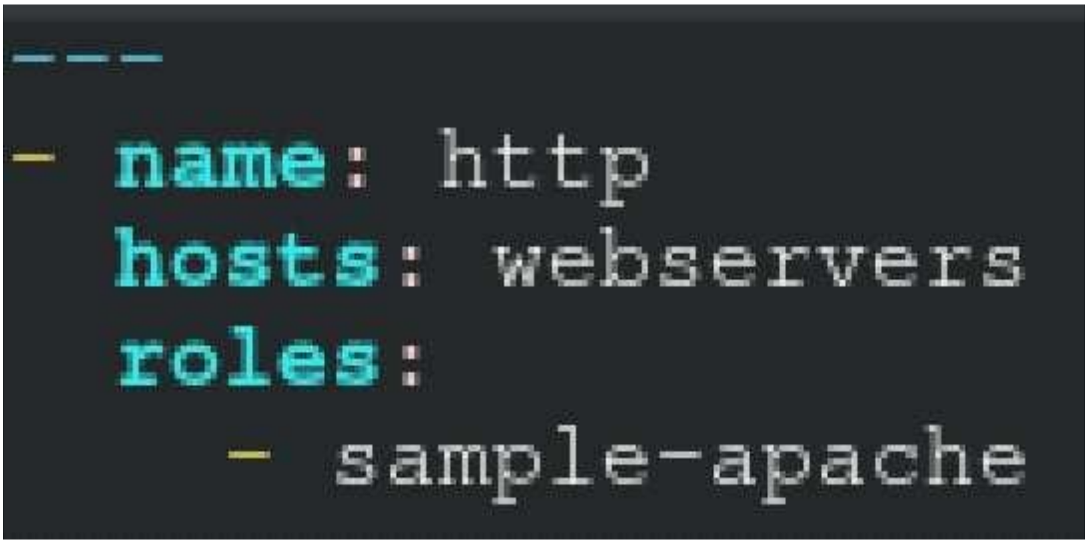
Create a role called sample-apache in /home/sandy/ansible/roles that enables and starts httpd, enables and starts the firewall and allows the webserver service. Create a template called index.html.j2 which creates and serves a message from /var/www/html/index.html Whenever the content of the file changes,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

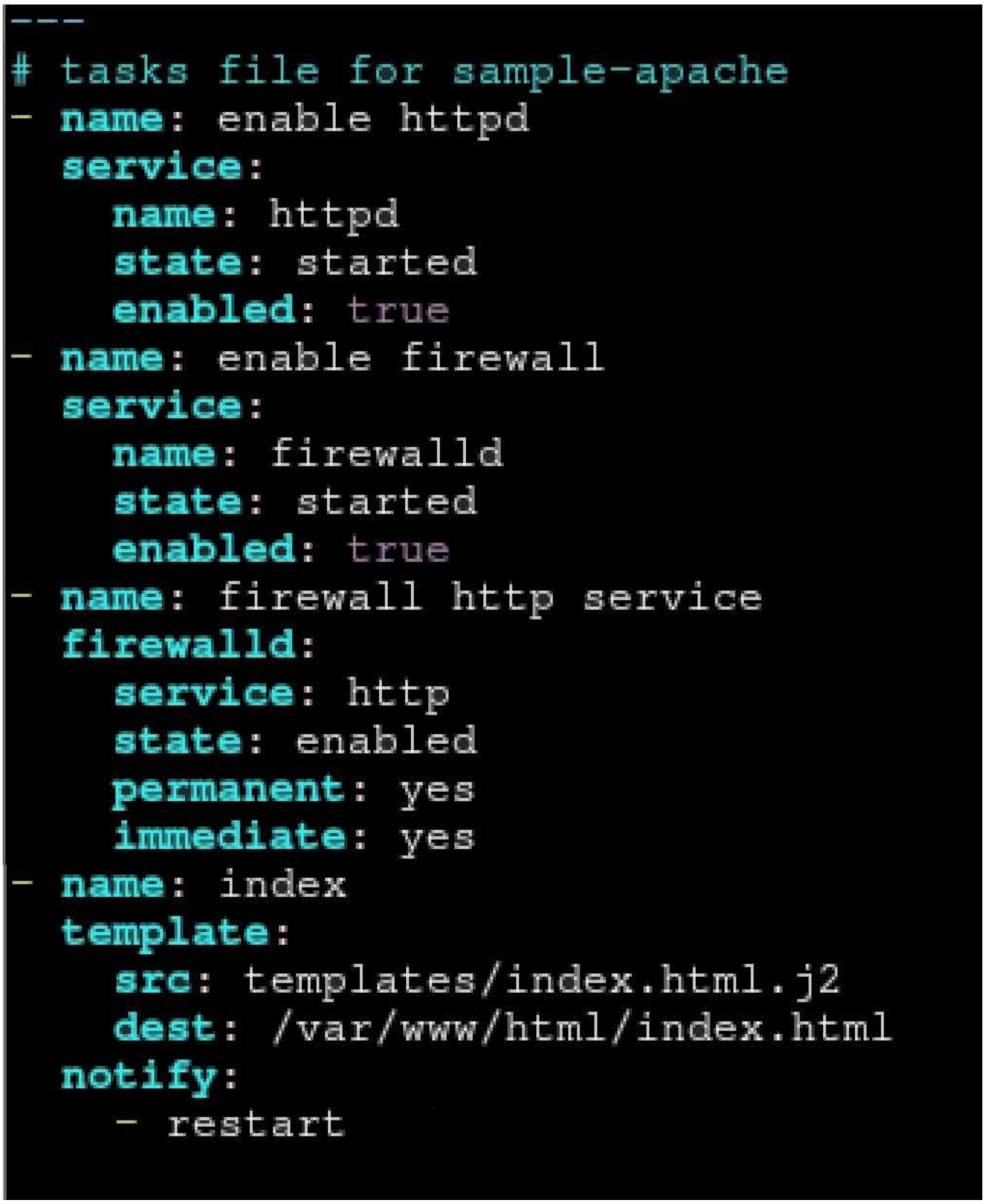
Solution

/home/sandy/ansible/apache.yml



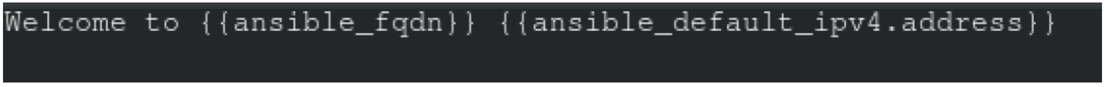
```
---
- name: http
  hosts: webservers
  roles:
    - sample-apache
```

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



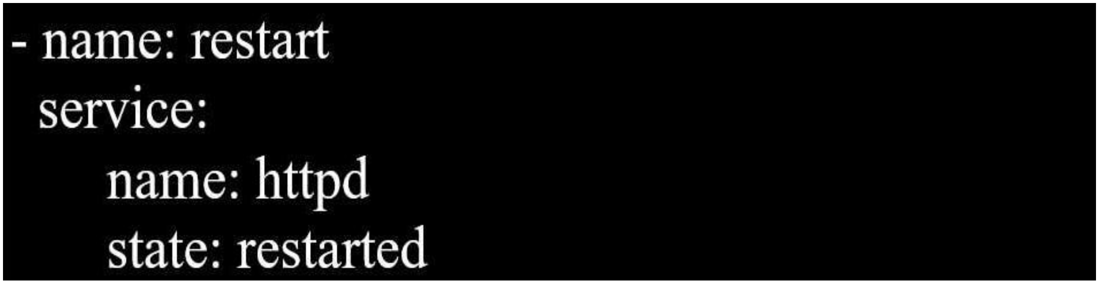
```
---
# tasks file for sample-apache
- name: enable httpd
  service:
    name: httpd
    state: started
    enabled: true
- name: enable firewall
  service:
    name: firewalld
    state: started
    enabled: true
- name: firewall http service
  firewalld:
    service: http
    state: enabled
    permanent: 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



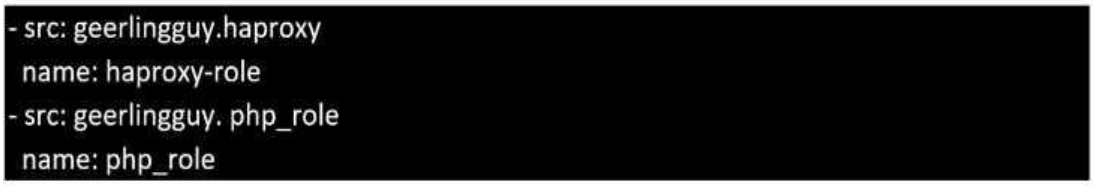
```
- name: restart
  service:
      name: httpd
      state: restarted
```

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.

Solution

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



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

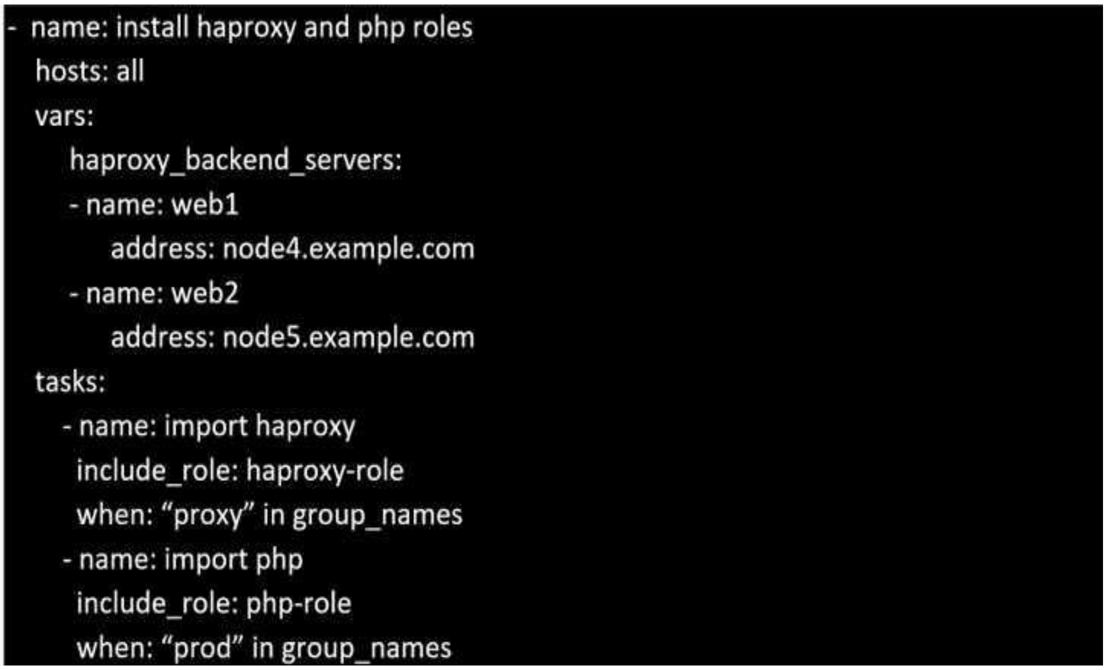
Run the requirements file from the roles directory:
ansible-galaxy install -r requirements.yml -p /home/sandy/ansible/roles

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.

Solution

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.

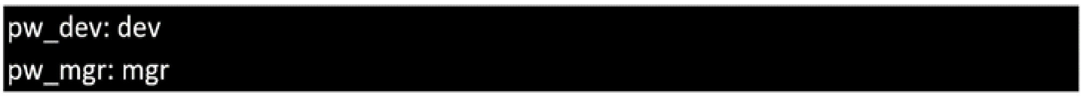
Solution

ansible-vault create lock.yml

New Vault Password: reallysafepw

Confirm: reallysafepw

In file:



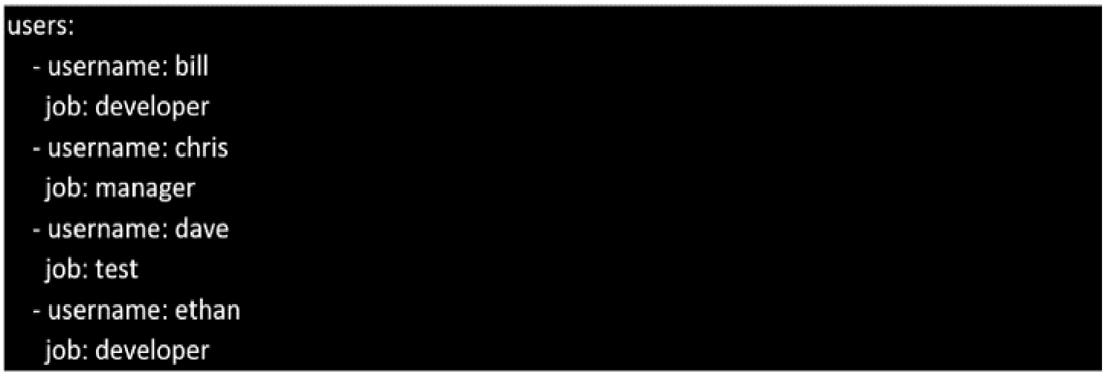
```
pw_dev: dev
pw_mgr: mgr
```

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.

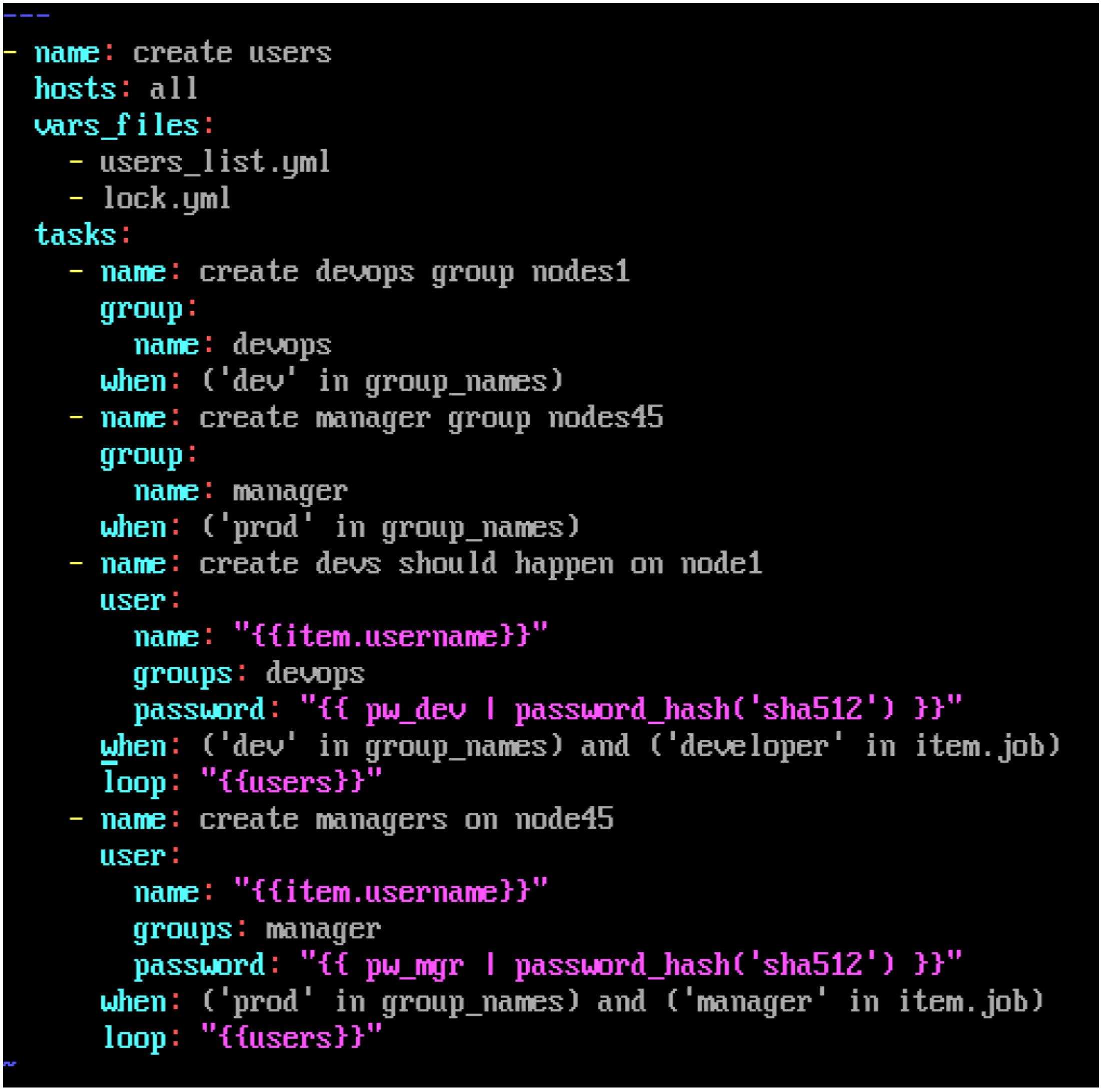
users_list.yml



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

Solution

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



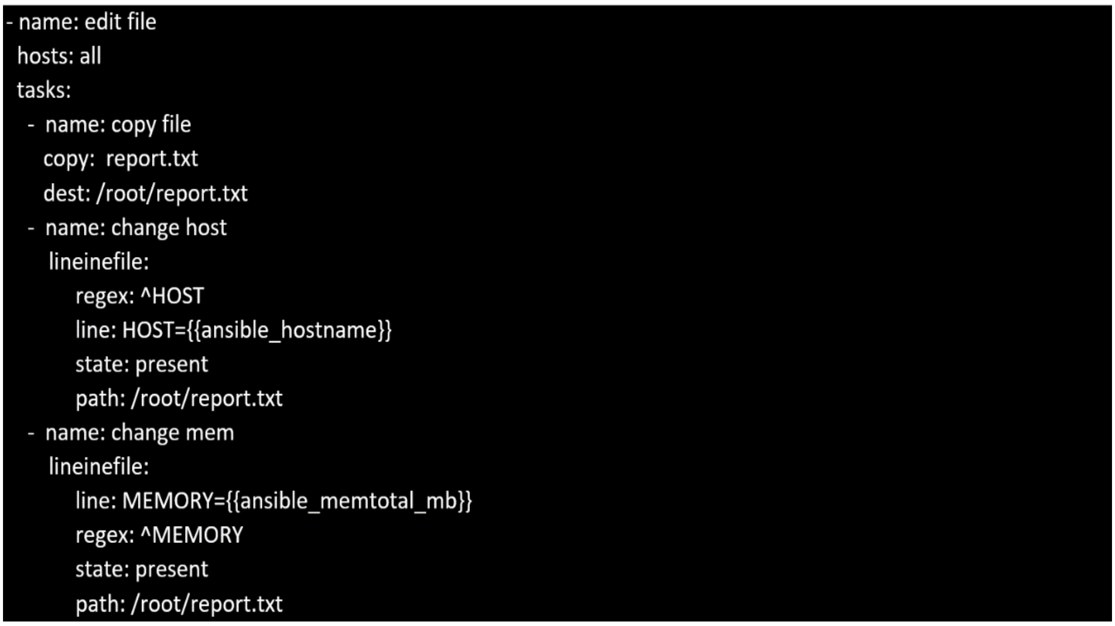
```
---
- name: create users
  hosts: all
  vars_files:
    - users_list.yml
    - lock.yml
  tasks:
    - name: create devops group nodes1
      group:
        name: devops
      when: ('dev' in group_names)
    - name: create manager group nodes45
      group:
        name: manager
      when: ('prod' 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

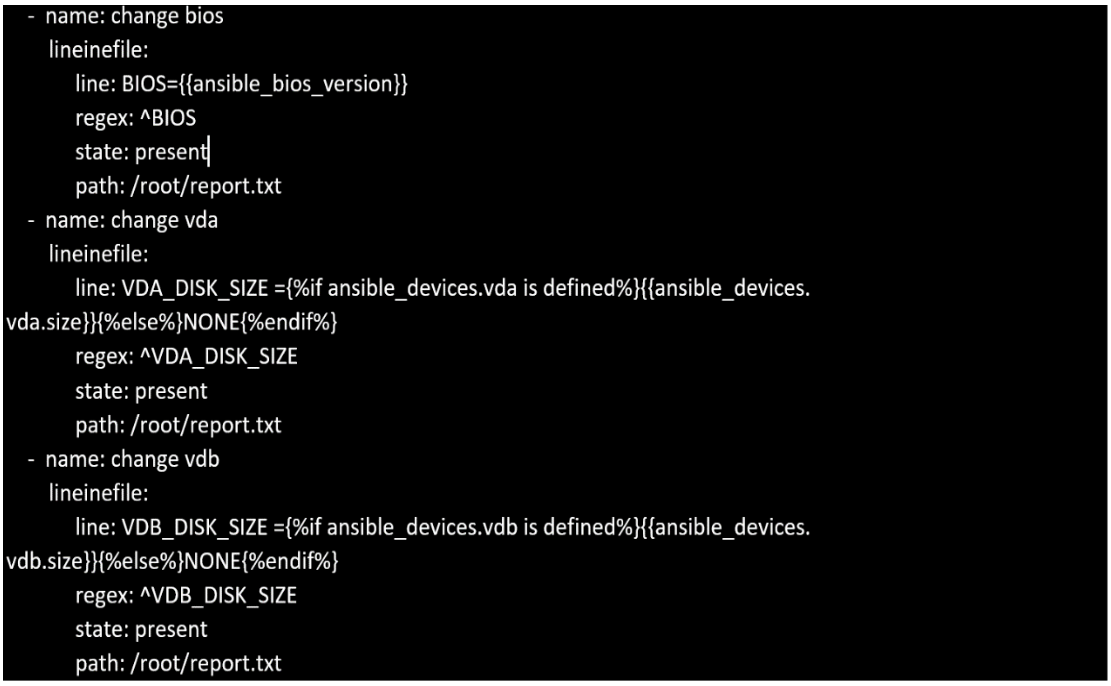
Create a file called specs.empty in home/bob/ansible on the local machine as follows:
HOST=
MEMORY=
BIOS=
VDA_DISK_SIZE=
VDB_DISK_SIZE=
Create the playbook /home/bob/ansible/specs.yml which copies specs.empty to all remote nodes' path /root/specs.txt. Using the specs.yml playbook then edit specs.txt on the remote machines to reflect the appropriate ansible facts.

Solution

Solution as:



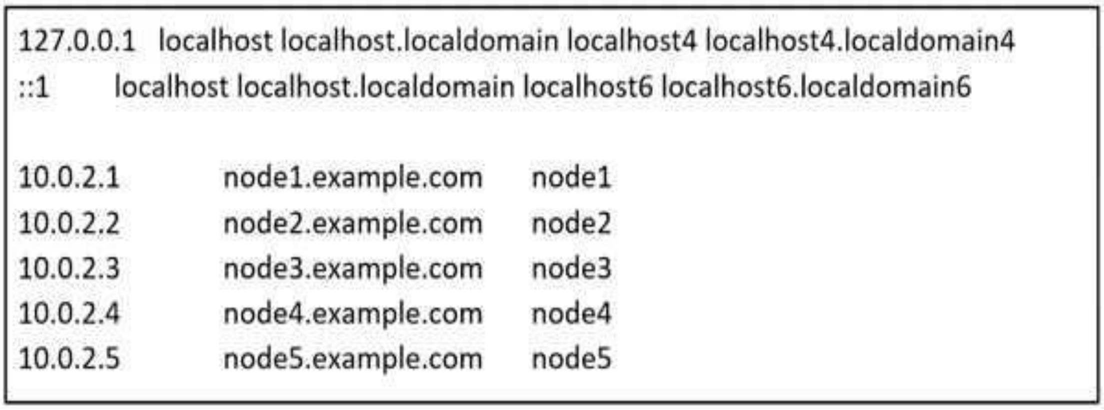
```
- name: edit file
  hosts: all
  tasks:
   -  name: copy file
      copy:  report.txt
      dest: /root/report.txt
   -  name: change host
       lineinefile:
          regex: ^HOST
          line: HOST={{ansible_hostname}}
          state: present
          path: /root/report.txt
   -  name: change 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

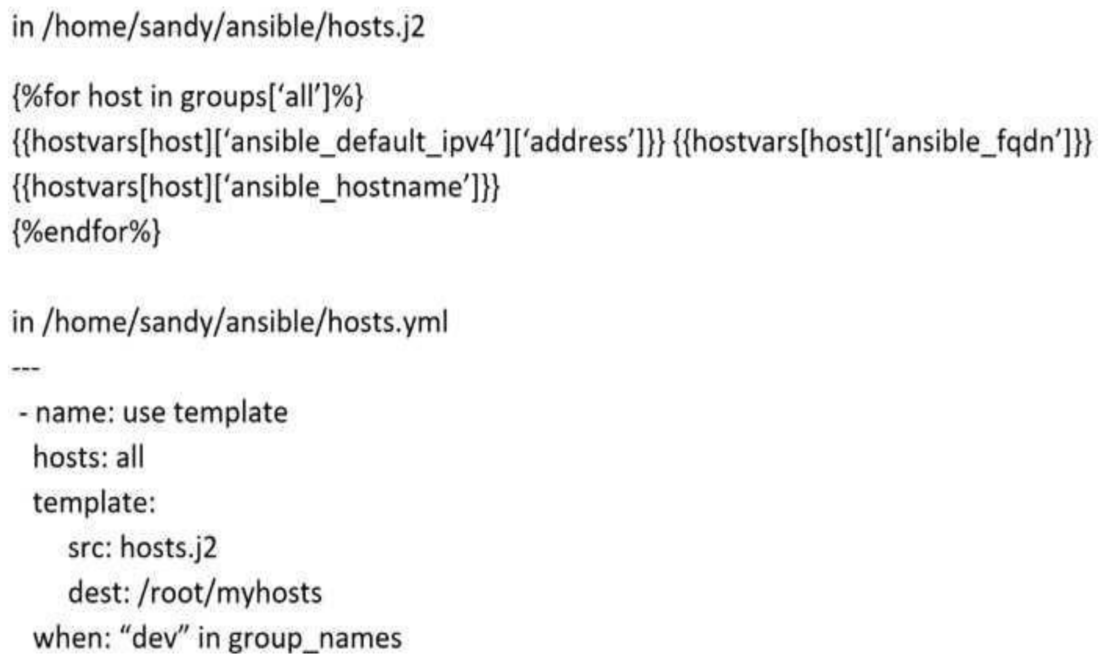
Create a jinja template in /home/sandy/ansible/ and name it hosts.j2. Edit this file so it looks like the one below. The order of the nodes doesn't matter. Then create a playbook in /home/sandy/ansible called hosts.yml and install the template on dev node at /root/myhosts



```
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
```

Solution

Solution as:



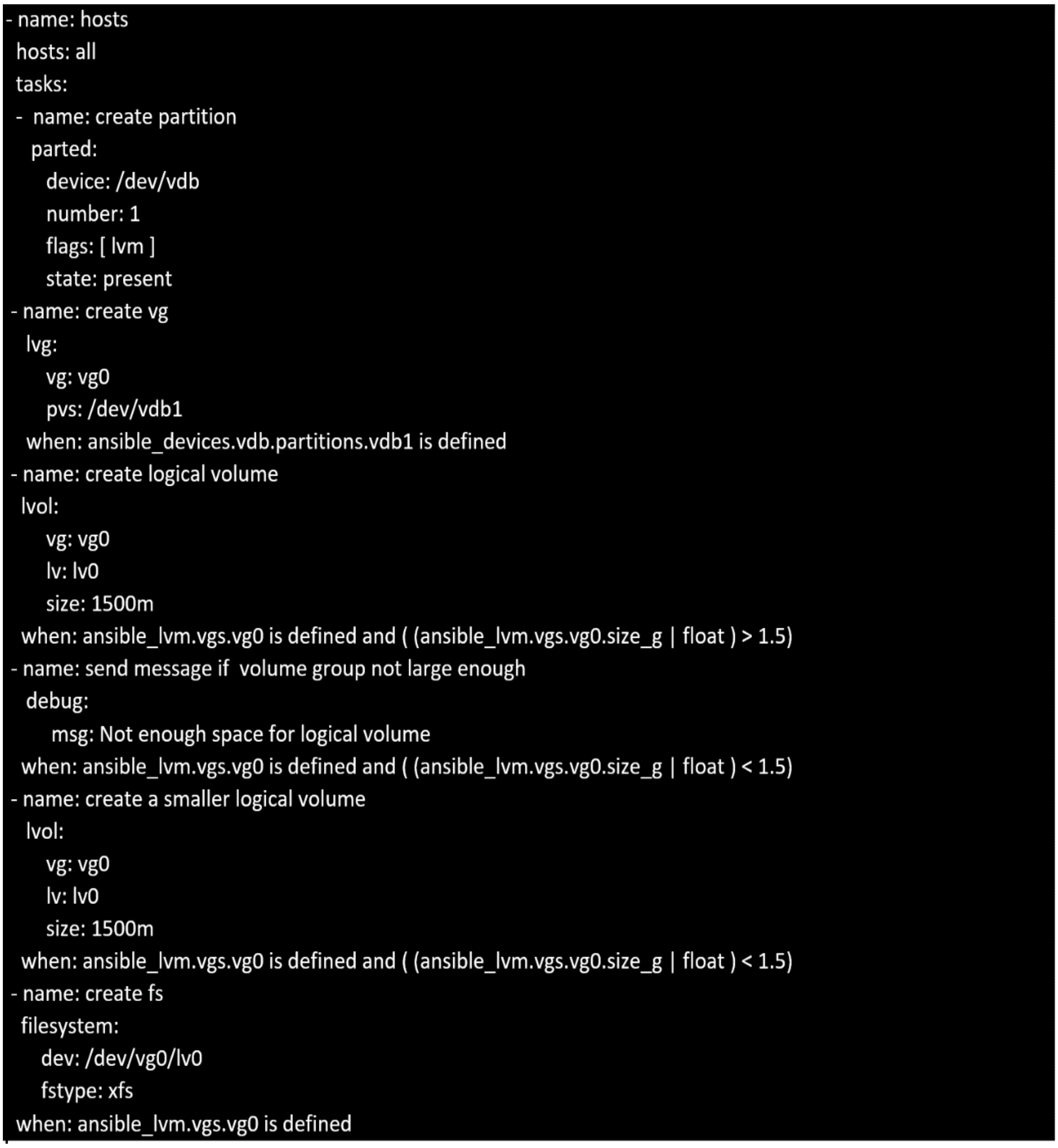
```
in /home/sandy/ansible/hosts.j2

{%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.

Solution

Solution as:



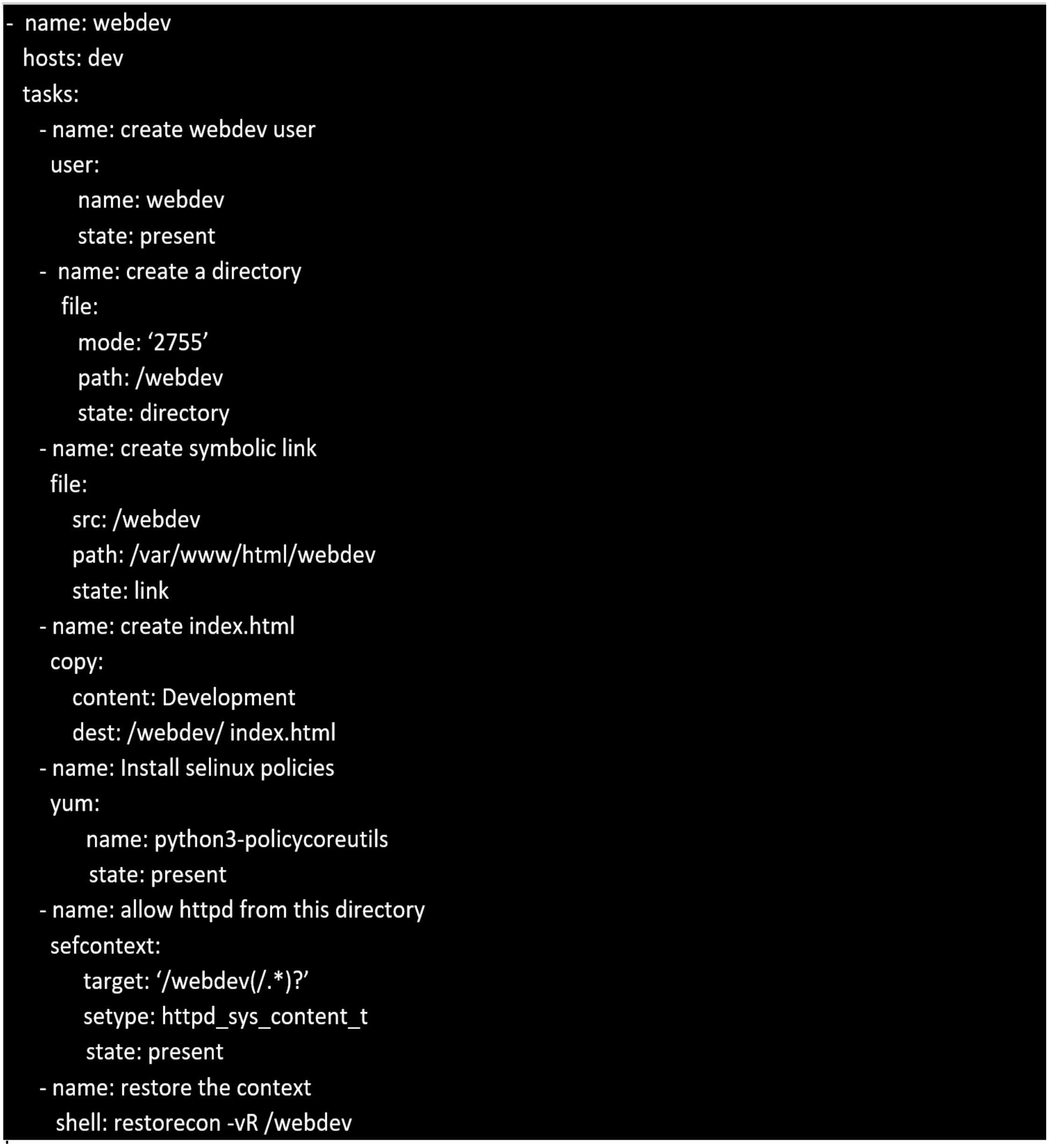
```
- name: hosts
  hosts: all
  tasks:
  -  name: create partition
     parted:
        device: /dev/vdb
        number: 1
        flags: [ lvm ]
        state: present
 - name: create vg
   lvg:
        vg: vg0
        pvs: /dev/vdb1
   when: ansible_devices.vdb.partitions.vdb1 is defined
 - name: 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

Solution

Solution as:



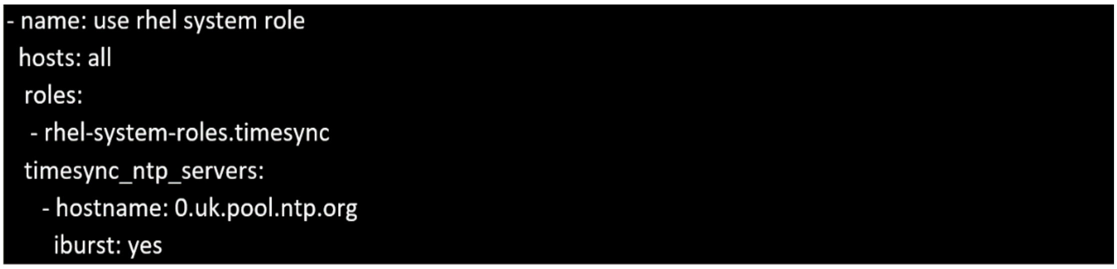
```
-  name: webdev
  hosts: dev
  tasks:
    - name: create webdev user
      user:
          name: webdev
          state: present
    -  name: create a directory
       file:
          mode: ‘2755’
          path: /webdev
          state: directory
    - name: create symbolic link
      file:
         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.

Solution

Solution as:



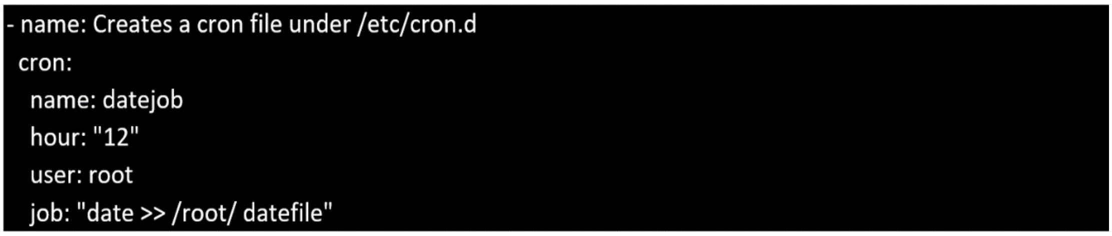
```
- name: use rhel system role
 hosts: all
  roles:
   - rhel-system-roles.timesync
  timesync_ntp_servers:
    - hostname: 0.uk.pool.ntp.org
      iburst: yes
```

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'

Solution

Solution as:



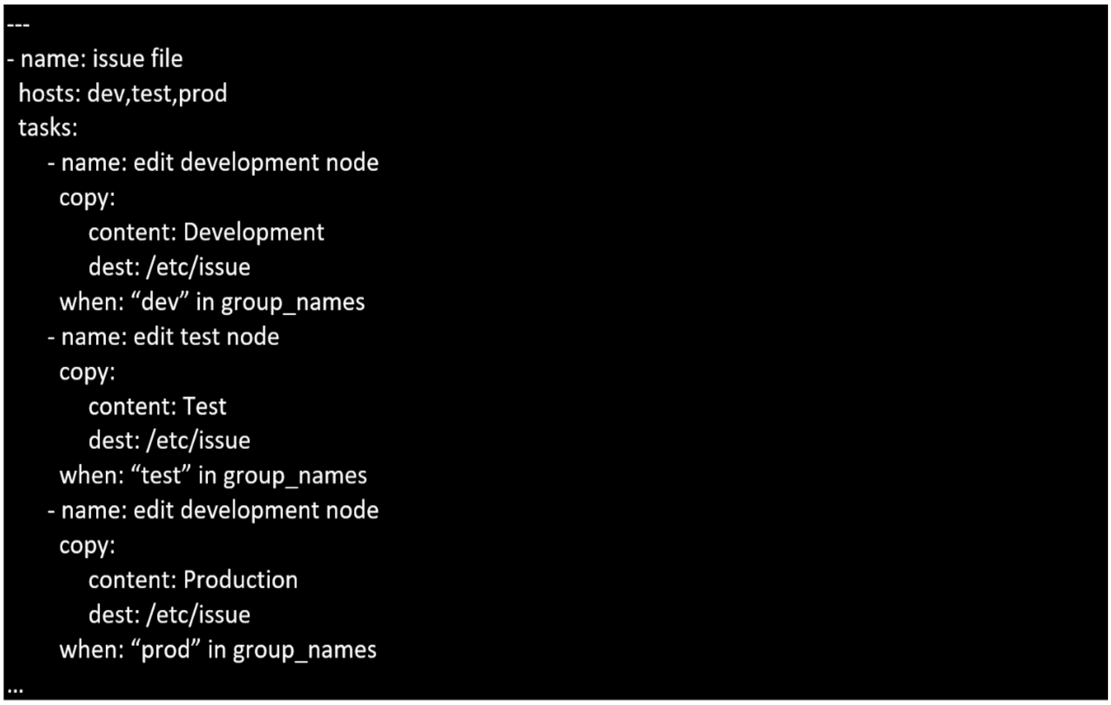
```
- name: Creates a cron file under /etc/cron.d
 cron:
   name: datejob
   hour: "12"
   user: root
   job: "date >> /root/ datefile"
```

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"

Solution

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.

Solution

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

Solution

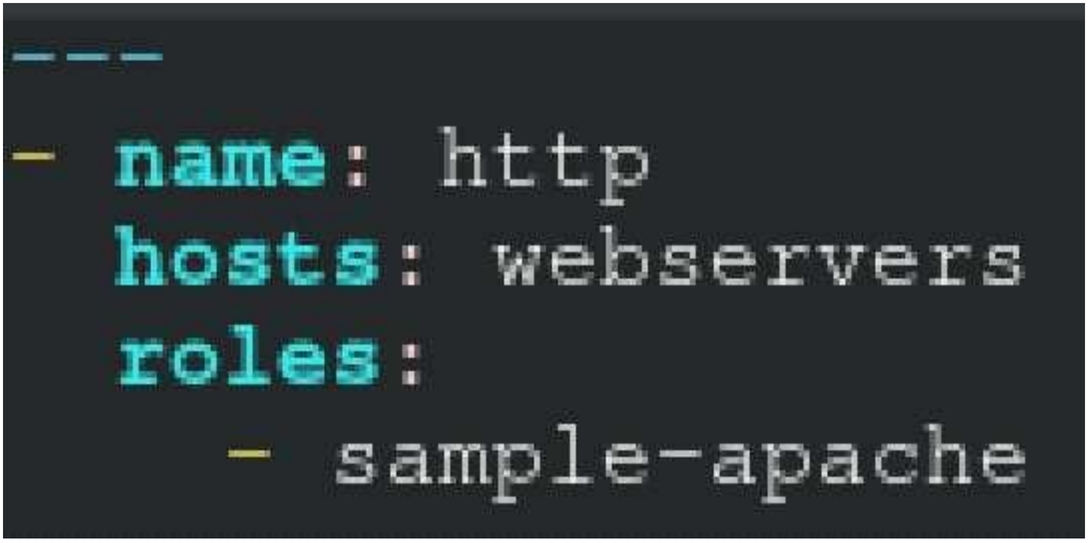
- name: change default target
hosts: all
tasks:
- name: change target
file:
src: /usr/lib/systemd/system/multi-user.target dest: /etc/systemd/system/default.target state: link

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".

Solution

/home/sandy/ansible/apache.yml



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

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

Question: 19

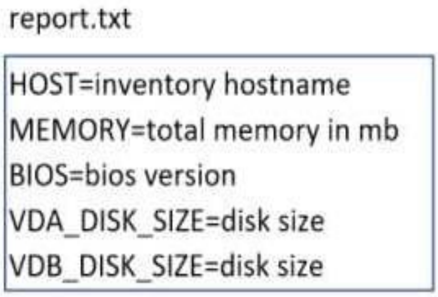
Install and configure ansible
User sandy has been created on your control node with the appropriate permissions already, do not change or modify ssh keys. Install the necessary packages to run ansible on the control node. Configure ansible.cfg to be in folder /home/sandy/ansible/ansible.cfg and configure 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.

Solution

```
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.

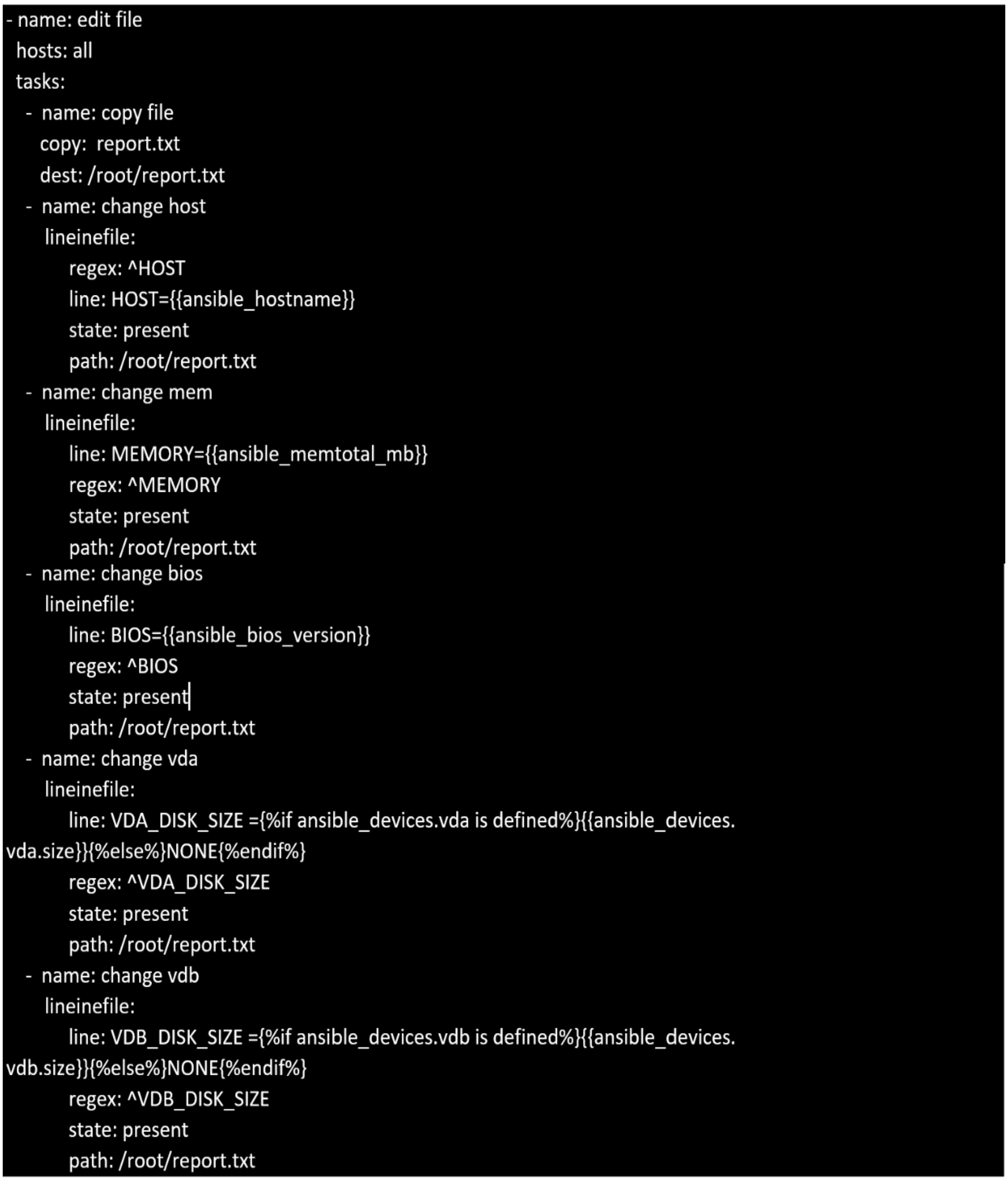


report.txt

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

Solution

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
```