

RHCE 9 (EX 294) Exam Sample Paper Solution

Max. Marks: 300

Passing Score: 70% = 210/300

Time: 4 Hours (240 Minutes)

Note:

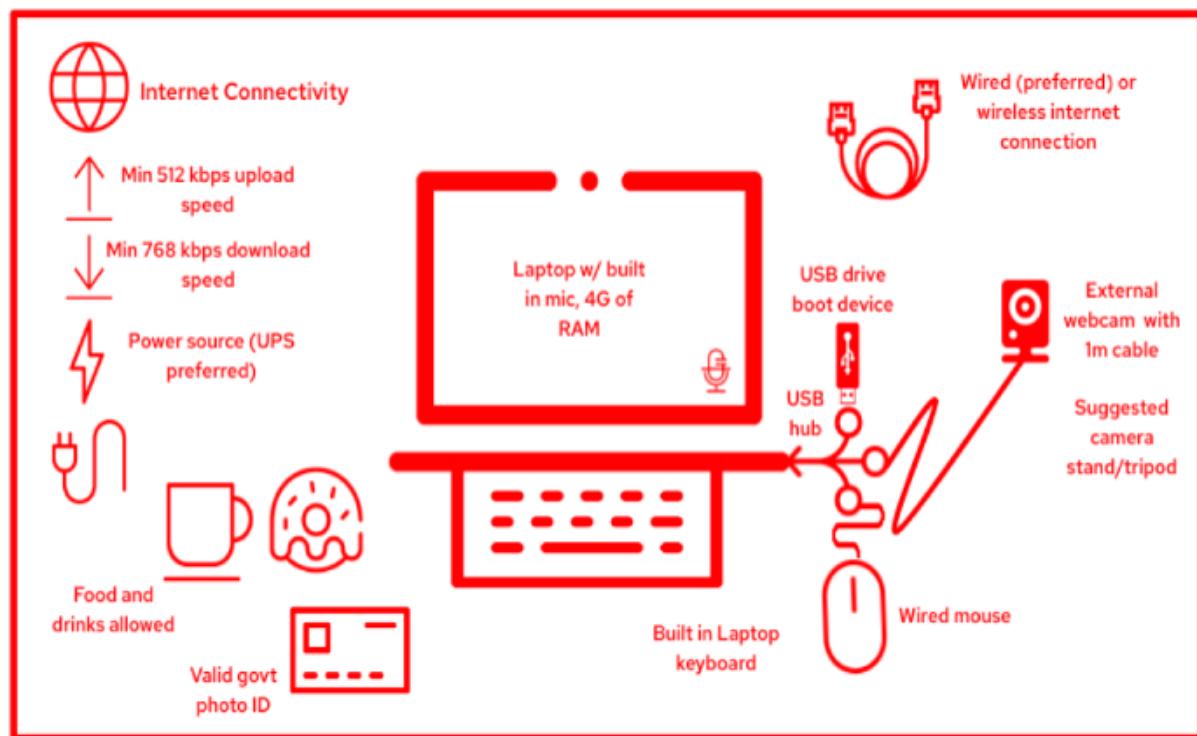
1. Ensure that all the tasks are implemented with firewalld and SELinux enabled.
2. Your server should be able to survive the reboot.

Introduction:

This practice examination question paper is for those that have gone through the RHCE 9 (EX294) course/book and want to test their knowledge before sitting in the actual RHCE 9 (EX294) examination.

Remote Exam Setup Requirements:

Please setup the environment as per the diagram below to appear in the RedHat certification exam remotely from home.



Points To Remember:

- a) Practice a lot and try to solve all these questions before 2.5 hrs.
- b) Trust yourself, you can easily score 100% marks.
- c) First of all, read & understand the instructions carefully.
- d) You don't need to create the local user accounts on the different nodes for the (key based) password-less authentication until not asked in the question itself.
- e) Try to solve the questions at your own first before you check the solutions.

Instructions:

- a) There will be one controller machine (ansible server) and 5 managed nodes.
- b) You will be provided with the root login and root password, and you have to create playbooks via vikasnehra user.
- c) You have to login on controller node with root user account then switch to vikasnehra user account.
- d) Create all your playbooks in the /home/vikasnehra/ansible directory if the playbook path is not well-defined in the question.
- e) Create config and inventory file in the /home/vikasnehra/ansible directory.
- f) Create a roles directory in the /home/vikasnehra/ansible directory.

RHCE 9 (EX 294) Exam Sample Paper Solution

1. Install and configure Ansible on the control node (ansible-server):
 - a) Install the required packages.
 - b) Create a static inventory file called /home/vikasnehra/ansible/inventory so that:
 - i) node1 is a member of the dev host group.
 - ii) node2 is a member of the test host group.
 - iii) node3 and node4 are the members of the prod host group.
 - iv) node5 is a member of the balancers host group.
 - c) The prod group is a member of the webservers host group.
 - d) Create a configuration file called /home/vikasnehra/ansible/ansible.cfg so that:
 - i) The host inventory file is /home/vikasnehra/ansible/inventory
 - ii) The default content collections directory is /home/vikasnehra/ansible/mycollection
 - iii) The default roles directory is /home/vikasnehra/ansible/roles

Solution 1:

```
# dnf install ansible-core python3-pip vim -y
$ su - vikasnehra
$ pip install ansible-navigator
$ ping node1
$ ping node2
$ ping node3
$ ping node4
$ ping node5

$ vim .vimrc
set ai
set ts=2
set et
set cursorcolumn

$ mkdir -p /home/vikasnehra/ansible/mycollection
$ mkdir -p /home/vikasnehra/ansible/roles

$ vim /home/vikasnehra/ansible/inventory
[dev]
node1

[test]
node2

[prod]
node[3:4]

[balancers]
node5

[webservers:children]
prod

$ ansible -i /home/vikasnehra/ansible/inventory all --list-hosts
$ ansible-inventory -i /home/vikasnehra/ansible/inventory --list
```

```
$ vim /home/vikasnehra/ansible/ansible.cfg
[defaults]
inventory = /home/vikasnehra/ansible/inventory
roles_path = /home/vikasnehra/ansible/roles
collections_path = /home/vikasnehra/ansible/mycollection
remote_user = vikasnehra
```

RHCE 9 (EX 294) Exam Sample Paper Solution

```

ask_pass = false
host_key_checking = false

[privilegeEscalation]
become = true
becomeMethod = sudo
becomeUser = root
becomeAskPass = false

$ ansible --version
$ echo "export ANSIBLE_CONFIG=/home/vikasnehra/ansible/ansible.cfg" >> .bashrc
$ source .bashrc
$ ansible --version
$ ansible -m ping all

```

2. Create and run an Ansible ad-hoc command. As a system administrator, you will need to install software on the managed nodes:
 - a) Create a shell script called yum-repo.sh that runs Ansible ad-hoc commands to create the yum repositories on each of the managed nodes as per the following details:
 - b) NOTE: you need to create 2 repos (BaseOS & AppStream) in the managed nodes.

BaseOS:

```

name: BaseOS
baseurl: file:///mnt/BaseOS/
description: Base OS Repo
gpgcheck: yes
enabled: yes
key: file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

```

AppStream:

```

name: AppStream
baseurl: file:///mnt/AppStream/
description: AppStream Repo
gpgcheck: yes
enabled: yes
key: file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

```

Solution 2:

```

$ ansible all -m command -a 'dnf repolist all'
$ vim yum-repo.sh
#!/bin/bash
ansible all -m yum_repository -a 'file=external.repo name=BaseOS description="Base OS Repo" baseurl=file:///mnt/BaseOS/ gpgcheck=yes gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release enabled=yes state=present' -b

ansible all -m yum_repository -a 'file=external.repo name=AppStream description="AppStream Repo" baseurl=file:///mnt/AppStream/ gpgcheck=yes gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release enabled=yes state=present' -b

$ chmod +x yum-repo.sh
$ ./yum-repo.sh
$ ansible all -m command -a 'dnf repolist all'

```

3. Create a playbook called /home/vikasnehra/ansible/packages.yml that:

- a) Installs the php and mariadb packages on hosts in the dev, test, and prod host groups only.
- b) Installs the RPM Development Tools package group on hosts in the dev host group only.
- c) Updates all packages to the latest version on hosts in the dev host group only.

Solution 3:

```
$ vim /home/vikasnehra/ansible/packages.yml
```

RHCE 9 (EX 294) Exam Sample Paper Solution

```
---
- name: Package Management Ansible Playbook
  hosts: all
  tasks:
    - name: install the packages
      dnf:
        name: "{{ item }}"
        state: present
      loop:
        - php
        - mariadb
      when: inventory_hostname in groups['dev'] or inventory_hostname in groups['test'] or inventory_hostname in groups['prod']
    - name: install the RPM development tool package group
      dnf:
        name: "@RPM Development tools"
        state: present
      when: inventory_hostname in groups['dev']
    - name: update all packages
      dnf:
        name: '*'
        state: latest
      when: inventory_hostname in groups['dev']
...
$ ansible-playbook /home/vikasnehra/ansible/packages.yml
```

4. Install the RHEL system roles package and create a playbook called /home/vikasnehra/ansible/timesync.yml that:
- Runs on all the managed hosts.
 - Uses the timesync role.
 - Configures the role to use the time server 172.25.254.250
 - Configures the role to set the iburst parameter as enabled.

Solution 4:

```
$ sudo yum install rhel-system-roles -y
$ vim /home/vikasnehra/ansible/timesync.yml
---
```

```
- name: Time Sync Using Roles Playbook
  hosts: all
  become: true
  vars:
    timesync_ntp_servers:
      - hostname: 172.25.254.250
        iburst: yes
  roles:
    - /usr/share/ansible/roles/rhel-system-roles.timesync
...
$ ansible-playbook /home/vikasnehra/ansible/timesync.yml
```

5. Create a role called apache in /home/vikasnehra/ansible/roles with the following requirements:
- The httpd package should be installed, httpd service should be enabled on boot, and started.
 - The firewall is enabled and running with a rule to allow access to the web server.
 - A template file index.html.j2 exists (you have to create this file) and is used to create the file /var/www/html/index.html with the following output: Welcome to HOSTNAME on IPADDRESS, where HOSTNAME is the fully qualified domain name of the managed node and IPADDRESS is the IP address of the managed node.

RHCE 9 (EX 294) Exam Sample Paper Solution

- d) Create a playbook called /home/vikasnehra/ansible/apache-role.yml that uses this role as follows: The playbook runs on hosts in the webservers host group.

Solution 5:

```
$ cd /home/vikasnehra/ansible/
$ ansible-galaxy init --init-path=roles apache
$ ls -lh roles/
$ vim /home/vikasnehra/ansible/roles/apache/tasks/main.yml
---
```

```
- name: Installing all the required packages
```

```
yum:
  name: "{{ item }}"
  state: present
  loop: "{{ pkgs }}"
```

```
- name: Creating index.html file from ansible jinja2 template
  template:
```

```
  src: index.html.j2
  dest: /var/www/html/index.html
```

```
- name: Starting and Enabling the services
```

```
service:
  name: "{{ item }}"
  state: started
  enabled: true
  loop: "{{ pkgs }}"
```

```
- name: Managing firewall rules.
```

```
firewalld:
  service: "{{ item }}"
  permanent: true
  immediate: true
  state: enabled
  loop: "{{ rule }}"
...
```

```
$ vim /home/vikasnehra/ansible/roles/apache/vars/main.yml
---
```

```
pkgs:
  - httpd
  - firewalld
rule:
  - http
  - https
...
```

```
$ vim /home/vikasnehra/ansible/roles/apache/templates/index.html.j2
```

```
Welcome to {{ ansible_fqdn }} on {{ ansible_default_ipv4['address'] }}
```

```
$ vim /home/vikasnehra/ansible/roles/apache-role.yml
---
```

```
- name: Apache Roles Playbook
  hosts: webservers
  roles:
    - role: /home/vikasnehra/ansible/roles/apache
...
```

```
$ ansible-galaxy collection install ansible.posix
```

RHCE 9 (EX 294) Exam Sample Paper Solution

```
$ ansible-playbook /home/vikasnehra/ansible/roles/apache-role.yml
$ curl node3
$ curl node4
```

6. Use Ansible Galaxy with the requirements file called /home/vikasnehra/ansible/roles/requirements.yml to download and install roles to /home/admin/ansible/roles from the following URLs:
 - a) <https://galaxy.ansible.com/download/zabbix-zabbix-1.0.6.tar.gz> The name of this role should be zabbix.
 - b) https://galaxy.ansible.com/download/openafs_contrib-openafs-1.9.0.tar.gz The name of this role should be security.
 - c) <https://galaxy.ansible.com/download/mafalb-squid-0.2.0.tar.gz> The name of this role should be squid.

Solution 6:

```
$ vim /home/vikasnehra/ansible/roles/requirements.yml
---
- src: https://galaxy.ansible.com/download/zabbix-zabbix-1.0.6.tar.gz
  name: zabbix

- src: https://galaxy.ansible.com/download/openafs_contrib-openafs-1.9.0.tar.gz
  name: security

- src: https://galaxy.ansible.com/download/mafalb-squid-0.2.0.tar.gz
  name: squid
...

$ cd /home/vikasnehra/ansible
$ ls -lh roles
$ ansible-galaxy install -r /home/vikasnehra/ansible/roles/requirements.yml -p roles/
$ ls -lh roles
```

7. Create a playbook called squid.yml as per the following details:
 - a) The playbook contains a play that runs on hosts in the balancers host group and uses the squid role present in your machine.

Solution 7:

```
$ vim /home/vikasnehra/ansible/squid.yml
---
- name: play for balancer group
  hosts: balancers
  roles:
    - /home/vikasnehra/ansible/roles/squid
...

$ ansible-playbook /home/vikasnehra/ansible/squid.yml
```

```
$ ansible balancers -m command -a 'systemctl status squid' -b
$ ansible balancers -m command -a 'sudo cat /etc/squid/squid.conf'
```

8. Create a playbook called test.yml as per the following details:
 - a) The playbook runs on managed nodes in the test host group.
 - b) Create the directory /webtest with the group ownership webtest group and having the regular permissions rwx for the owner and group and rx for the others.
 - c) Apply the special permissions: set group ID
 - d) Symbolically link /var/www/html/webtest to /webtest directory.
 - e) Create the file /webtest/index.html with a single line of text that reads: Testing

Solution 8:

```
$ vim /home/vikasnehra/ansible/test.yml
```

RHCE 9 (EX 294) Exam Sample Paper Solution

```
---
- name: Webtest server Playbook
  hosts: test
  tasks:
    - name: Creating the webtest group.
      group:
        name: webtest

    - name: Creating the /webtest directory.
      file:
        state: directory
        path: /webtest
        mode: '2775'
        group: webtest
        owner: apache

    - name: Creating the /webtest directory symbolic link.
      file:
        src: /webtest
        dest: /var/www/html/myweb
        state: link
        mode: '2775'
        owner: apache

    - name: Creating the index file.
      copy:
        dest: /webtest/index.html
        mode: '0640'
        owner: apache
        content: "Testing"
...

```

```
$ ansible-playbook /home/vikasnehra/ansible/test.yml
$ ansible test -m command -a 'ls -lh /webtest' -b
```

- 9. Create an Ansible vault to store user passwords with the following conditions:**
- The name of the vault is `vault.yml`
 - The vault contains two variables, `dev_pass` with value as `redhat` and `mgr_pass` with value as `linux` respectively.
 - The password to encrypt and decrypt the vault is `nehraclasses`
 - The password is stored in the file `/home/vikasnehra/ansible/password.txt` file.

Solution 9:

```
$ cd /home/vikasnehra/ansible/
$ vim /home/vikasnehra/ansible/password.txt
nehraclasses
```

```
$ ansible-vault create --vault-password-file=/home/vikasnehra/ansible/password.txt vault.yml
dev_pass: redhat
mgr_pass: linux
```

```
$ ansible-vault view /home/vikasnehra/ansible/vault.yml
```

10. Generate hosts files:

- Download an initial template file called `hosts.j2` from the below URL: <http://classroom.example.com/content/hosts.j2> to `/home/vikasnehra/ansible/` directory. 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`.

RHCE 9 (EX 294) Exam Sample Paper Solution

- b) Create a playbook called gen_hosts.yml that uses this template to generate the file /etc/myhosts on hosts in the dev host group.
- c) 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
192.168.10.x node1.example.com node1
192.168.10.y node2.example.com node2
192.168.10.z node3.example.com node3
192.168.10.a node4.example.com node4
192.168.10.b node5.example.com node5
```

Solution 10:

```
$ cd /home/vikasnehra/ansible/
$ wget http://classroom.example.com/content/hosts.j2
```

```
$ vim hosts.j2
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
{% for x in groups['all'] %}
{{ hostvars[x]['ansible_facts']['default_ipv4']['address'] }} {{ hostvars[x]['fqdn'] }} {{ hostvars[x]['ansible_facts']['hostname'] }}
{% endfor %}
```

```
$ vim gen_hosts.yml
---
```

```
- name: Hosts File Playbook
hosts: all
tasks: []
- hosts: dev
tasks:
- name: Generating the hosts file.
  template:
    src: "/home/vikasnehra/ansible/hosts.j2"
    dest: "/etc/myhosts"
...

```

```
$ ansible-playbook gen_hosts.yml
$ ansible dev -m command -a 'sudo cat /etc/myhosts'
```

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

- a) Inventory hostname
- b) Total memory in MB
- c) BIOS version

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

Solution 11:

```
$ cd /home/vikasnehra/ansible/
```

```
$ vim hwreport.j2
-- Inventory hostname : {{ ansible_facts['fqdn'] }}
-- Total memory in MB: {{ ansible_facts['memtotal_mb'] }}
-- BIOS version: {{ ansible_facts['bios_version'] }}
```

```
$ vim hwreport.yml
---
```

```
- name: Playbook for the hardware report.
```

RHCE 9 (EX 294) Exam Sample Paper Solution

```

hosts: all
become: true
tasks:
- name: Generating the hardware report
  template:
    src: /home/vikasnehra/ansible/hwreport.j2
    dest: /root/hwreport.txt
...

```

```

$ ansible-playbook hwreport.yml
$ ansible all -m command -a 'sudo cat /root/hwreport.txt'

```

12. Create a playbook called /home/vikasnehra/ansible/issue.yml as per the following requirements:

- a) The playbook runs on all inventory hosts.
- b) The playbook replaces the contents of /etc/issue with a single line of text as:
 - i. On hosts in the dev host group, the line reads: Development
 - ii. On hosts in the test host group, the line reads: Test
 - iii. On hosts in the prod host group, the line reads: Production

Solution 12:

```
$ cd /home/vikasnehra/ansible/
```

```
$ vim issue.yml
```

```
---
```

```
- name: Ansible playbook for replacing the contents in the /etc/issue file.
```

```
hosts: all
```

```
tasks:
```

```
- name: Replacing the content in dev group.
```

```
copy:
```

```
  content: "Development"
```

```
  dest: /etc/issue
```

```
  when: inventory_hostname in groups['dev']
```

```
- name: Replacing the content in test group.
```

```
copy:
```

```
  content: "Test"
```

```
  dest: /etc/issue
```

```
  when: inventory_hostname in groups['test']
```

```
- name: Replacing the content in prod group.
```

```
copy:
```

```
  content: "Production"
```

```
  dest: /etc/issue
```

```
  when: inventory_hostname in groups['prod']
```

```
...
```

```
$ ansible-playbook issue.yml
```

```
$ ansible prod -m command -a 'sudo cat /etc/issue'
```

```
$ ansible test -m command -a 'sudo cat /etc/issue'
```

```
$ ansible dev -m command -a 'sudo cat /etc/issue'
```

13. Rekey an existing Ansible vault as per the following conditions:

- a) Use the vault.yml file which have you created earlier.
- b) Set the new vault password as ansible.
- c) The vault remains in an encrypted state with the new password.

Solution 13:

RHCE 9 (EX 294) Exam Sample Paper Solution

```
$ cd /home/vikasnehra/ansible/
$ ls -lh vault.yml
$ ansible-vault rekey --ask-vault-pass vault.yml
Vault password: nehraclasses
New Vault password: ansible
Confirm New Vault password: ansible
Rekey successful
```

```
$ cat vault.yml
```

```
$ ansible-vault view vault.yml
Vault password: ansible
dev_pass: redhat
mgr_pass: linux
```

14. 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/content/user_list.yml" and save to `/home/vikasnehra/ansible/` directory. 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 is a 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 is a 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.

Solution 14:

```
$ cd /home/vikasnehra/ansible/
$ wget http://classroom.example.com/content/user\_list.yml
```

```
$ vim user_list.yml
users:
- name: adam
  job: developer
  uid: 3000
- name: gabriel
  job: manager
  uid: 3001
- name: lucifer
  job: developer
  uid: 3002
```

```
$ vim create_user.yml
---
- name: Users & groups management playbook
  hosts: test,dev
  vars_files:
    - /home/vikasnehra/ansible/user_list.yml
    - /home/vikasnehra/ansible/vault.yml
  tasks:
    - name: creating the devops group
      group:
        name: devops
        state: present
    - name: creating the users having developer profile.
      user:
        name: "{{ item.name }}"

```



Nehra Classes
Igniting The Minds

RHCE 9 (EX 294) Exam Sample Paper Solution

```

uid: "{{ item.uid }}"
groups: devops
state: present
password: "{{ dev_pass | password_hash('sha512') }}"
when: item.job == "developer"
loop: "{{ users }}"

- hosts: prod
vars_files:
- /home/vikasnehra/ansible/user_list.yml
- /home/vikasnehra/ansible/vault.yml
tasks:
- name: creating the opsmgr group
group:
  name: opsmgr
  state: present
- name: creating the users having manager profile.
user:
  name: "{{ item.name }}"
  uid: "{{ item.uid }}"
  groups: opsmgr
  state: present
  password: "{{ mgr_pass | password_hash('sha512') }}"
when: item.job == "manager"
loop: "{{ users }}"
...

```

\$ vim password.txt
ansible

\$ ansible-playbook create_user.yml --vault-password-file=password.txt

\$ ansible all -m command -a 'sudo id -a lucifer'
\$ ansible all -m command -a 'sudo id -a gabriel'
\$ ansible all -m command -a 'sudo id -a adam'

15. Configure cron jobs:

Create /home/vikasnehra/ansible/cron.yml playbook as per the following requirements:

- This playbook runs on all managed nodes in the hostgroup.
- Configure cronjob , which runs every 2 minutes and executes the following commands: logger "EX294 exam in progress" and run as user natasha.

Solution 15:

```

$ cd /home/vikasnehra/ansible/
$ vim cron.yml
---
- name: Cron job playbook
hosts: all
tasks:
- name: Creating the user natasha for running the cron job.
user:
  name: natasha
  state: present
- name: Scheduling the cron job.
cron:
  name: "cron job for the natasha user"
  minute: "*/2"
  job: 'logger "EX294 exam in progress"'
  user: natasha

```

RHCE 9 (EX 294) Exam Sample Paper Solution

...
\$ ansible-playbook cron.yml

\$ ansible all -m command -a 'crontab -l -u natasha'
\$ ansible all -m command -a 'sudo tail /var/log/cron'

16. Create & use a logical volume:

Create a playbook called /home/vikasnehra/ansible/lvm.yml that runs on all the managed nodes and does the following:

- a) Creates a logical volume with the following requirements:
 - i. The logical volume is created in the research volume group.
 - ii. The logical volume name is data.
 - iii. The logical volume size is 1200 Mib.
 - iv. Format the logical volume with the ext4 file-system.
 - v. if the requested logical volume size cannot be created, the error message "could not create logical volume of that size" should be displayed and size 800 MiB should be used instead.
 - vi. if the volume research does not exist, the error message "volume group does not exist" should be displayed.
 - vii. Don't mount the logical volume in any way.

Solution 16:

\$ cd /home/vikasnehra/ansible/

\$ vim lvm.yml

```
---
- name: lvm playbook
  hosts: all
  tasks:
    - name: checking details
      block:
        - name: if the volume research does not exist, print the error message.
          debug:
            msg: "volume group does not exist"
            when: "'research' not in ansible_lvm.vgs"
    - name: creating the 1200m lvm
      lvol:
        vg: research
        lv: data
        size: 1200m
        when: "'research' in ansible_lvm.vgs"

    rescue:
      - name: if the requested logical volume size cannot be created, the print.
        debug:
          msg: "Could not create logical volume of that size"
          when: "'research' in ansible_lvm.vgs"
    - name: creating the logical volume of 800m
      lvol:
        vg: research
        lv: data
        size: 800m
        when: "'research' in ansible_lvm.vgs"

  always:
    - name: format filesystem
      filesystem:
        fstype: ext4
        dev: /dev/research/data
        when: "'research' in ansible_lvm.lvs"
```

RHCE 9 (EX 294) Exam Sample Paper Solution

...

```
$ ansible-galaxy collection install community.general
$ ansible-playbook lvm.yml
```

17. Create and use partitions:

Create /home/vikasnehra/ansible/partition.yml, which will create partitions on all the managed nodes:

- a) After vdb creating a 1200M primary partition, partition number 1, and format it into ext4 filesystem.
- b) On the prod group to permanently mount the partition to /srv directory.
- c) If there is not enough disk space, give prompt information "Could not create partition of that size" and create a 800M partition.
- d) If vdb does not exist, a prompt message will be given "this disk does not exist."

Solution 17:

```
$ cd /home/vikasnehra/ansible/
```

```
$ vim partition.yml
```

```
- name: Partition Playbook
hosts: all
tasks:
  - name: check block stroage availability.
    block:
      - name: If vdb does not exist, a prompt message will be given
        debug:
          msg: "this disk does not exist."
          when: "'vdb' not in ansible_devices"
      - name: Creating the 1200m partition
        parted:
          device: /dev/vdb
          number: 1
          part_end: 1200MiB
          state: present
          when: "'vdb' in ansible_devices"
    rescue:
      - name: If there is not enough disk space, give the prompt information
        debug:
          msg: "Could not create partition of that size"
          when: "'vdb1' not in ansible_devices.vdb"
      - name: Creating the 800m partition
        parted:
          device: /dev/vdb
          number: 1
          part_end: 800MiB
          state: present
  always:
    - name: Creating the ext4 filesystem
      filesystem:
        fstype: ext4
        dev: /dev/vdb1
        when: "'vdb' in ansible_devices"
    - name: mounting the filesystem.
      mount:
        path: /srv
        src: /dev/vdb1
        fstype: ext4
        state: mounted
        when: "'prod' in group_names"
```

RHCE 9 (EX 294) Exam Sample Paper Solution

...
\$ ansible-playbook partition.yml

18. Using a selinux role create a selinux.yml playbook with the following conditions:
- Configure on all managed hosts to set the default selinux mode as permissive.
 - Verify the selinux mode on all the nodes using ansible ad-hoc command.
 - Create another copy of the selinux.yml playbook with the name as selinux2.yml and make changes there in it to configure the selinux default mode as enforcing for all the managed nodes.
 - Execute the selinux2.yml playbook using ansible navigator.
 - Verify the selinux mode on all the node machines.

Solution 18:

```
$ cd /home/vikasnehra/ansible/  
$ ansible-galaxy install linux-system-roles.selinux
```

```
$ vim selinux.yml
```

```
---
```

```
- name: creating a play for selinux mode.  
hosts: all  
vars:  
  - selinux_state: permissive  
roles:  
  - linux-system-roles.selinux  
...
```

```
$ ansible all -m command -a 'sestatus'  
$ ansible-playbook selinux.yml  
$ ansible all -m command -a 'sestatus'
```

```
$ cp selinux.yml selinux2.yml
```

```
$ vim selinux2.yml
```

```
---
```

```
- name: creating a play for selinux mode.  
hosts: all  
vars:  
  - selinux_state: enforcing  
roles:  
  - linux-system-roles.selinux  
...
```

```
$ sudo loginctl enable-linger 1000  
$ ansible-navigator run -m stdout selinux2.yml  
$ ansible all -m command -a 'sestatus'
```

All the best for your RHCE (EX294) exam, best wishes from NEHRA CLASSES.

Thank You