RHCE Ansible(EX294) EXAM-PAPER

- **Six machines are there on your exam environment**
- **Control Node**
- **Control.domainX.example.com**
- **Managed Host**
- **Node1.domainX.exmple.com**
- **Node2.domainX.example.com**
- **Node3.domainX.example.com**
- **Node4.domainX.example.com**
- **Node5.domainX.example.com**

Note- All work you have to do on Control Node (control.domainX.example.com).
First of all you have to login with admin user, then create directory called ansible.
All work you have to do under /home/admin/ansible directory.
Q1.)

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

Install the required packages
b- Create a static inventory file called /home/admin/ansible/inventory as follows:
-- **Node1.domainX.example.com** is a member of the dev host group
-- **Node2.domainX.example.com** is a member of the test host group
-- **Node3.domainXexample.**com and **Node4.domainX.example.**com are members of the prod host group
-- **Node5.domainX.example.com** is a member of the balancers host group
-- The prod group is a member of the webservers host group
c- 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**

Ans-

```
$sudo yum install ansible*
```

(This command configure ansible on your control node)

```
$yum install vim*
```

(In exam only)

(this command configure vim editor on your control node)

```
$vim inventory
```

Press ‘i’

```
[dev]
Node1.domainX.example.com
[test]
Node2.domainX.example.com
[prod]
Node3.domainX.example.com
Node4.domainX.example.com
[balancers]
Node5.domainX.example.com
[webservers:children]
prod
```

‘save file’

Create ansible.cfg

```
$vim ansible.cfg
```

Press ‘i’

```
[defaults]
inventory = /home/admin/ansible/inventory
remote_user = admin
ask_pass = false
roles_path = /home/admin/ansible/roles:/usr/share/ansible/roles
[privilege_escalation]
become = true
become_method = sudo
become_user = root
become_ask_pass = false
```

‘save file’

Q2.)

Create and run an Ansible playbook

As a system administrator, you will need to install software on the managed nodes.

a- Create a playbook called **yum-pack.yml to create a yum repository on each of the managed host as follows** .

1.The name of the repository is EX407
2. The description is "Ex407 Description"
3. The base URL is ftp://192.168.10.254/pub/rhel75/
4. GPG signature checking is enabled
5. The GPG key URL is ftp://192.168.10.254/pub/rhel75/RPM-GPG-KEY-redhat-release
6. The repository is enabled

Ans-

```
      vim yum-pack.yml
- - -
 - name: creating yum repository for all managed host
   hosts: all
   tasks:
        - yum_repository:
                      name: EX407
                      baseurl: ftp://192.168.10.254/pub/rhel75/AppStream
                      gpgcheck: 1
                      enabled: 1
                      gpgkey:
ftp://192.168.10.254/pub/rhel75/RPM-GPG-KEY-redhat-release
                      description: “EX407”
                      state: present
        - yum_repository:
                      name: EX407
                      baseurl: ftp://192.168.10.254/pub/rhel75/BaseOS
                      gpgcheck: 1
                      enabled: 1
                      gpgkey:
ftp://192.168.10.254/pub/rhel75/RPM-GPG-KEY-redhat-release
                      description: “EX407”
                      state: present
        'save file'
 $ansible-playbook yum-pack.yml
```

Q3.)

Install packages

-- 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 RPM 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

Ans-

```
$vim packages.yml
Press 'i'
```

```
---
 - name: php and mariadb install on dev,test,prod
   hosts: dev,test,prod
   tasks:
         - yum:
               name:
                     - php
                     - mariadb
               state: present
 - name: dev tool and update packages
   hosts: dev
   tasks:
         - yum:
               name: "@RPM Development Tools"
               state: present
         - yum:
               name: '*'
               state: latest
```

'save file'

```
    $ansible-playbook packages.yml
```

Q4.)

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

-- Runs on all managed hosts
-- Uses the timesync role
-- Configures the role to use the time server 192.168.10.254
-- Configures the role to set the iburst parameter as enabled

Ans-

```
    $vim timesync.yml
```

Press ‘i’

```
---
- hosts: all
  vars:
    timesync_ntp_servers:
      - hostname: 192.168.10.254
        iburst: yes
  roles:
    - rhel-system-roles.timesync
```

‘save file’

```
    $ansible-playbook timesync.yml
```

Q5.)

Create and use a role

- 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
    - 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.

Create a playbook called **httpd.yml** that uses this role as follows:
The playbook runs on hosts in the webservers host group

Ans-

```
        $mkdir roles
        $cd roles
        $ansible-galaxy init apache
        $cd apache/tasks
        $vim main.yml
        Press ‘i’
```

```
---
 - yum:
        name: httpd
        state: present
 - service:
        name: httpd
        state: started
        enabled: yes
 - service:
        name: firewalld
        state: started
        enabled: yes
 - firewalld:
        service: http
        state: enabled
        permanent: yes
        immediate: yes
 - template:
        src: index.html.j2
        dest: /var/www/html/index.html
```

```
        ‘save file’
        $cd ../template
        $vim index.html.j2
        Press ‘i’
```

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

```
        ‘save file’
```

```
          $cd /home/admin/ansible
          $vim httpd.yml
---
- name: apache role
  hosts: webservers
  roles:
           - apache
          ‘save file’
          $ansible-playbook httpd.yml
```

Q6.)

Install roles using Ansible Galaxy

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://192.168.10.254/ex407/role1.tar.gz

The name of this role should be balancer

-- http://192.168.10.254/ex407/role2.tar.gz

The name of this role should be phphello

Ans-

```
          $vim roles/install.yml
---
 - src: content.example.com
   name: balancer
 - src: content.example.com
   name: phphello
          $cd ..
          $ansible-playbook roles/install.yml
```

Q7.)

Create a playbook called balance.yml as follows:

The playbook contains a play that runs on hosts in the balancers host group and uses the balancer role.

This role configures a service to load balance web server requests between hosts in the webservers host group.

When implemented, browsing to hosts in the balancers host group (for example http://node5.example.com) should produce the following output: **Welcome to node3.domainX.example.com on 192.168.10.z**

Reloading the browser should return output from the alternate web server:

Welcome to node4.domainX.example.com on 192.168.10.a

The playbook contains a play that runs on hosts in the 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.domainX.example.com/hello.php, should produce the following output:

Hello PHP World from node3.domainX.example.com

along with various details of the PHP configuration including the version of PHP that is installed.

Similarly, browsing to http://node4.domainX.example.com/hello.php, should produce the following output:

Hello PHP World from node4.domainX.example.com

along with various details of the PHP configuration including the version of PHP that is installed.

Ans-

```
$vim balance.yml
```

Press “i”

```
---
 - name: balancer role
   hosts: balancers
   roles:
        - balancers
 - name: php role
   hosts: webservers
   roles:
        - phphello
```

Save the file

Before run this playbook you have to edit

/home/admin/ansible/roles/balancer/template/balancer.j2

```
# main frontend which proxys to the backends
#---------------------------------------------------------------------
frontend  main node5.domainX.example.com
bind *:80
    acl url_static       path_beg       -i /static /images /javascript
/stylesheets
    acl url_static       path_end       -i .jpg .gif .png .css .js
    use_backend static          if url_static
    default_backend             app
#---------------------------------------------------------------------
# static backend for serving up images, stylesheets and such
#---------------------------------------------------------------------
backend static
    balance     roundrobin
    server      static 127.0.0.1:4331 check

#---------------------------------------------------------------------
# round robin balancing between the various backends
#---------------------------------------------------------------------
backend app
    balance     roundrobin
server node3.domainX.example.com 172.25.250.12:80 check
server node4.domainX.example.com 172.25.250.13:80 check
```

"Save the file"

#ansible-playbook balance.yml

Q8.)

Create a playbook called web 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=read+write+execute, group=read+write+execute, other=read+execute , special permissions: 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

Ans-

$vim web.yml

Press ‘i’

```
---
 - name: webcontent directory
   hosts: dev
   tasks:
         - group:
                 name: apache
                 state: present
         - file:
                 path: /webdev
                 group: apache
                 mode: '2775'
                 state: directory
                 setype: httpd_sys_content_t
         - lineinfile:
                 path: /webdev/index.html
                 line: Development
                 create: yes
                 setype: httpd_sys_content_t
         - file:
                 src: /webdev
                 dest: /var/www/html/webdev
                 state: link
                 force: yes
```

‘save file’

$ansible-playbook web.yml

Q9.)

Create and use a partition

Create a playbook called **partition.yml** that runs on all managed nodes that does the following:

Creates a lvm partition name **mylv** under vg name **myvg** of size 1500MiB on device vdb

Formats the partition with the ext4 filesystem

-- If the requested partition size cannot be created, the error message "Requested size is not present" should be displayed and the size 800MiB should be used instead.

If the device vda does not exist, the error message "Devices is not present" should be displayed.

Note- (No need to mount the partition)

Ans-

$vim partition.yml

Press 'i'

```
---
 - name: create logical volume partition
   hosts: all
   tasks:
      - debug:
          msg: "Device is not present"
        when: ansible_lvm.vgs.myvg is defined
      - lvol:
          vg: myvg
          lv: mylv
          size: 1500m
        when: ansible_lvm.vgs.myvg.free_g >= "1.6"
      - debug:
          msg: "Requested size not present"
        when: ansible_lvm.vgs.myvg.free_g < "1.6"
      - lvol:
          vg: myvg
          lv: mylv
          size: 800m
        when: ansible_lvm.vgs.myvg.free_g < "1.6"
      - filesystem:
          fstype: ext4
          dev: /dev/myvg/mylv
```

```
'save file'
$ansible-playbook partition.yml
```

Q10.)

Create a password vault

-- 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**

Ans-

```
$vim password.txt
Press 'i'
atenorth
'save file'
$ansible-vault create –vault-password-file=password.txt vault.yml
Press 'i'
dev_pass: "wakennym"
mgr_pass: "rocky"
'save file'
```

Q11.)

Generate a hosts file

- Download an initial template file called hosts.j2 from http://192.168.10.254/ex407/
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
Create a playbook called **gen_hosts.yml** that uses this template to generate the file /etc/hosts 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
192.168.10.x node1.domainX.example.com node1
192.168.10.y node2.example.com node2
192.168.10.z node3.domainX.example.com node3
192.168.10.a node4.domainx.example.com node4
192.168.10.b node5.domainX.example.com node5
```

Ans-

```
$wget http://192.168.10.254/ex407/hosts.j2
$vim hosts.j2
```

Press ‘i’

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

‘Save file’

```
$vim gen_hosts.yml
```

Press ‘i’

```
---
 - name: host entry
   hosts: all
   tasks:
         - template:
                 src: hosts.j2
                 dest: /etc/myhost
 - name: delete host entry except dev host group
   hosts: all,!dev
   tasks:
         - file:
                 path: /etc/myhost
                 state: absent
```

‘save file’

```
$ansible-playbook gen_hosts.yml
```

Q12.)

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 keyvalue pair.

Your playbook should:

-- Download the file **hwreport.empty** from the URL http://192.168.10.254/ex407/hwreport.empty and save it as /root/hwreport.txt

-- Modify with the correct values.

If a hardware item does not exist, the associated value should be set to **NONE**

Ans-

$vim hwreport.yml

Press ‘i’

```
---
 - name: generate hardware report
   hosts: all
   tasks:
       - get_url:
              url: http://192.168.10.254/ex407/hwreport.empty
     dest: /root/hwreport
       - replace:
              path: /root/hardware.txt
              regexp: 'memb'
              replace: "{{ ansible_memtotal_mb }}"
         when: ansible_memtotal_mb is defined
       - replace:
              path: /root/hardware.txt
              regexp: 'memb'
```

```
        replace: "NONE"
  when: ansible_memtotal_mb is not defined
- replace:
        path: /root/hardware.txt
        regexp: 'bios'
        replace: "{{ ansible_bios_version }}"
  when: ansible_bios_version is defined
- replace:
        path: /root/hardware.txt
        regexp: 'bios'
        replace: "NONE"
  when: ansible_bios_version is not defined
- replace:
        path: /root/hardware.txt
        regexp: 'vda'
        replace: "{{ ansible_devices.vdc.size }}"
  when: ansible_devices.vdc.size is defined
- replace:
        path: /root/hardware.txt
        regexp: 'vda'
        replace: "NONE"
  when: ansible_devices.vdc.size is not defined
- replace:
        path: /root/hardware.txt
        regexp: 'vdb'
        replace: "{{ ansible_devices.vdb.size }}"
  when: ansible_devices.vdb.size is defined
- replace:
        path: /root/hardware.txt
        regexp: 'vdb'
        replace: "NONE"
  when: ansible_devices.vdb.size is not defined
- replace:
        path: /root/hardware.txt
        regexp: 'hostname'
```

```
          replace: "{{ ansible_hostname }}"
       when: ansible_hostname is defined
      - replace:
          path: /root/hardware.txt
          regexp: 'hostname'
          replace: "NONE"
       when: ansible_hostname is not defined
```

‘save file’

```
$ansible-playbook hwreport.yml
```

Q13.)

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: **Testing**

On hosts in the prod host group, the line reads: **Production**

Ans-

```
$vim modify.yml
```

Press ‘i’

```
---
 - hosts: dev
  tasks:
      - copy:
          content: "Development"
          dest: /modify.txt
 - hosts: test
  tasks:
      - copy:
          content: "Testing"
          dest: /modify.txt
 - hosts: prod
  tasks:
```

```
  - copy:
        content: "Production"
        dest: /modify.txt
```

'save file'

$ansible-playbook modify.yml

Q14.)

Rekey an Ansible vault

Rekey an existing Ansible vault as follows:
Download the Ansible vault from
http://192.168.10.254/ex407/secret.yml and save it as **secret.yml**
The current vault password is **curabete**
The new vault password is **newvare**
The vault remains in an encrypted state with the new password

Ans-

```
$ansible-vault rekey secret.yml
'type old password'
'type new password'
```

Q15.)

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://192.168.10.254/ex407/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 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 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.

Ans-

```
$wget http://192.168.10.254/ex407/user_list.yml
$vim user_list.yml
```

Press ‘i’

```
---
 - hosts: dev,test
   vars_files:
       - vault.yml
       - user_list.yml
   tasks:
       - group:
            name: devops
            state: present
       - user:
            name: "{{ item.name }}"
            groups: devops
            password: "{{ dev_pass | password_hash ('sha512') }}"
            state: present
         when: item.job == "developer"
         loop: "{{ user }}"
 - hosts: prod
   vars_files:
       - vault.yml
       - user_list.yml
   tasks:
       - group:
            name: opsmgr
            state: present
       - user:
            name: "{{ item.name }}"
            groups: opsmgr
            password: "{{ mgr_pass | password_hash ('sha512') }}"
```

```
              state: present
           when: item.job == "manager"
           loop: "{{ user }}"
        ‘save file’
```

Note- after write this playbook, you have to define password file path in ansible.cfg

```
        $ansible-playbook create_user.yml
```

Q 17)   Use a RHEL system role

Create a playbook name selinux.yml and use system roles

- Set selinux mode as enforcing in all manage node

Soln:

```
       vim selinux.yml
- - -
 - name: using rhel-system-roles.selinux for all managed host
   hosts: all
   vars:
       - selinux_state: enforcing
   roles:
       -  role: rhel-system-roles.selinux
   ‘Save file’
$ansible-playbook selinux.yml
```

Q 18) Create a cronjob for user natasha in all nodes, the playbook name crontab.yml and the job details are Every 2 minutes the job will execute logger "EX294 in progress"

Soln:

```
    vim crontab.yml
- - -
 - name: use crontab
   hosts: all
```

```
tasks:
   - name: create crontab for natasha user
     cron:
       name: “test”
       User: natasha
       Minute: “*/2”
       Job: “logger EX294”
       State: present
```