RHCE Ansible (EX294)
EXAM PAPER

- **Six machines are there on your exam environment**

  A. **Control Node**

     1. **Control.domainX.example.com**

  B. **Managed Host**

     1. **Node1.domainX.example.com**
     2. **Node2.domainX.example.com**
     3. **Node3.domainX.example.com**
     4. **Node4.domainX.example.com**
     5. **Node5.domainX.example.com**

- **We will take access of control node via ssh, so that we can easily copy content from question paper directly to terminal for fast solving.**
- **All work you have to do on Control Node (control.domainX.example.com). First of all you have to login with the given user for example 'admin' user, then create directory called ansible in the home directory of that user. All work you have to do under /home/admin/ansible directory.**

Q1. Setting up Ansible Environment

$ sudo yum install ansible*

$ mkdir /home/admin/ansible

$ vim inventory

```
[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
```

$ vim ansible.cfg

```
 [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
```

- **To Verify:**
- **ansible all -m ping**

Q2). Create a playbook for yum configuration

```
$ vim yum.yml
```

```
---
- name: Setting up Repository
  hosts: all
  tasks:
        name: Configuring BaseOS
        yum_repository:
                name: EX294_BASE
                description: “EX294 Base Software”
                baseurl: http://server.network.example.com/BaseOS
                 gpgcheck: true
                gpgkey: http://server.network.example.com/RHEL/RPM-GPG-KEY-release
                enabled: true


       name: Configuring AppStream
        yum_repository:
                name: EX294_STREAM
                description: “EX294 Base Software”
                baseurl: http://server.network.example.com/AppStream
                 gpgcheck: true
                gpgkey: http://server.network.example.com/RHEL/RPM-GPG-KEY-release
                enabled: true
```

- **To Verify:**
- **ansible all -m command -a “yum repolist”**

Q3). Install Packages

```
$vim packages.yml
```

```
---
  - 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
```

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

- **To Verify:**
- **ansible all -m command -a “rpm -q php”**

Q4). Install Collections

- **$ mkdir /home/admin/ansible/mycollection**

- Add this line to ansible.cfg below roles_path:

  collections_path = /home/admin/ansible/mycollection

Run these commands staying inside mycollections directory

- **$ wget http://utility.lab13.example.com/materials/redhat.rhel_system_roles-1.19.3.tar.gz**
- **$ wget http://utility.lab13.example.com/materials/posix-1.20.2.tar.gz**
- **$ wget http://utility.lab13.example.com/materials/community-general-5.5.0.tar.gz**

Then

- **$ ansible-galaxy collection install redhat.rhel_system_roles-1.19.3.tar.gz**
- **$ ansible-galaxy collection install posix-1.20.2.tar.gz**
- **$ ansible-galaxy collection install community-general-5.5.0.tar.gz**

- To Verify

- You will see the collections installed in the given directory and they will be decompressed from their original .tar.gz format

Q5). Using System Roles

Come as Root User and run:

$ yum install rhel-system-roles
$ cd /usr/share/ansible/roles/rhel-system-roles.timesync
$ vim README.md

Scroll down to Example Playbook and copy the first one completely

$ su – admin
$ cd ansible
$ vim timesync.yml

(Paste what we copied and correct the alignment)

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

Save file

$ ansible-playbook timesync.yml

- To Verify:
- ansible all -m command -a “cat /etc/chrony.conf”

Q6). Installing Roles using Ansible Galaxy

Go to given path and create requirement file

$ vim /home/admin/ansible/roles/requirement.yml

Enter:

```
---
 - src: http://server.network.example.com/materials/haproxy.tar
   name: balancer

 - src:  http://server.network.example.com/materials/phpinfo.tar
   name: phphello
```

Then run command in same location:

$ ansible-galaxy install -r roles/requirement.yml -p roles/

- To Verify:
- Go to /home/admin/ansible/roles and check to see roles installed

Q7). Creating and Using a Custom Role

```
$ cd roles
$ ansible-galaxy init apache
$ cd apache/tasks
$ vim main.yml
```

```
---

 - 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
```

```
$ cd ../template
$ vim index.html.j2
```

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

$ cd /home/admin/ansible
$ vim httpd.yml

```
---
 - name: apache role
   hosts: webserver
   roles:
            -  apache
```

$ ansible-playbook httpd.yml

- To Verify:
- Give IP or hostname in web browser and it will show proper output as per our file entry, or
- ansible all -m command -a “curl localhost”

Q8). Using Roles using Ansible Galaxy

Go as Root User:

$ yum install haproxy

$ vim /etc/haproxy/haproxy.cfg

(Use this file as a reference file for the next part of the question)

Switch back to admin user:

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

(Orange is the text that will already be there)
(Black Text are the lines we need to add or modify accordingly)

```
# main frontend which proxys to the backends
#---------------------------------------------------------------------
frontend  main
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_staticdefault_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
```

$ vim balance.yml

```
---
- name: balancer role
- hosts: balancers
- roles:
    - balancer

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

$ ansible-playbook balance.yml

- To Verify:
- We will ping the balancer group IP/hostname, each time we refresh it in the browser we see the output changing as the server alternates between node3 and node4 as per our entry.

Q9). Generate a Hosts File

$ wget http://192.168.10.254/ex407/hosts.j2
$ vim hosts.j2
(Do not modify the existing entries, we will append at the end of file)

```
{% for host in groups['all'] %}

{{ hostvars[host].ansible_default_ipv4.address }} {{ hostvars[host].ansible_fqdn }}
{{ hostvars[host].ansible_hostname }}

{% endfor %}
```

$ wget http://server.network.example.com/materials/gen_hosts.yml
(Do not make any changes in this file)

$ ansible-playbook gen_hosts.yml

- To Verify:
- ansible dev -m command -a “cat /etc/myhosts”

Q10). Modify File Content

$ vim modify.yml

```
- hosts: dev
  tasks:
        - copy:
              content: "Development"
              dest: /etc/issue

- hosts: test
  tasks:
        - copy:
              content: "Testing"
              dest: /etc/issue

- hosts: prod
  tasks:
        - copy:
              content: "Production"
              dest: /etc/issue
```

$ ansible-playbook modify.yml

- To Verify:
- ansible all -m command -a “cat /etc/issue”

Q11). Create Web Content Directory

vim web.yml

```
- 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
```

```
$ ansible-playbook web.yml
```

- To Verify:
- ansible all -m command -a “ls -ldZ /webdev”
- ansible all -m command -a “cat /var/www/html/webdev/index.html”

Q12). Generate Hardware Report

$ vim hwreport.yml

```
---
 - name: Generate Hardware Report
   hosts: all
   tasks:
        - name:
          get_url:
                url: http://192.168.10.254/ex407/hwreport.empty
                dest: /root/hwreport.txt

 - replace:
          path: /root/hwreport.txt
          regexp: "{{ item.oldvalue }}"
          replace: "{{ item.newvalue }}"
   loop:
          - oldvalue: hostname
            newvalue: "{{  ansible_hostname | default('NONE')  }}"
          - oldvalue: biosversion
            newvalue: "{{  ansible_bios_version | default('NONE')  }}"
          - oldvalue: memory
            newvalue: "{{  ansible_memtotal_mb | default('NONE')  }}"
          - oldvalue: vdasize
            newvalue: "{{  ansible_devices.vda.size | default('NONE')  }}"
          - oldvalue: vdbsize
            newvalue: "{{  ansible_devices.vda.size | default('NONE')  }}"
```

$ ansible-playbook hwreport.yml

- To Verify:
- ansible all -m command -a "cat /root/hwreport.txt"

Q13). Create Password Vault

$ vim password.txt

atenorth

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

dev_pass: “wakennym”

mgr_pass: “rocky”

- To Verify:
- ansible-vault view –vault-password-file=password.txt vault.yml
- The content of vault should be visible

Q14). Create User Accounts

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

```
- 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 }}"
```

After this is complete need to make a change inside ansible.cfg file:
(Make entry below collections_paths)

vault_password_file= password.txt

- To Verify:
- ansible all -m command -a “id bob”
- ansible all -m command -a “id alice”

Q15). Rekey Ansible Vault

$ wget http://192.168.10.254/ex407/secret.yml

$ ansible-vault rekey secret.yml

old password: jaishreeram
new password: jaimatadi

- To Verify:
- ansible-vault view secret.yml
- New password should make it work

Q16). Storage LVM

$ vim partition.yml

```
---

 - name: create logical volume partition
   hosts: all
   tasks:
           - debug:
                msg: "Device is not present"
             when: ansible_lvm.vgs.myvg is not 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
```

$ ansible-playbook partition.yml

- To Verify:
- ansible all -m command -a “lv display”

Q17). Using RHEL System Role

Go to SELinux System Role

$ cd /usr/share/ansible/roles/rhel-system-roles.selinux
$ vim README.md

Go to “” ### selinux role “” line
We will find the sample playbook as well the parameter to set selinux_state as enforcing so we will

$ vim selinux.yml

```
---
 - name: Setting SELinux enforcing
   hosts: all
   vars:
        selinux_state: enforcing
   roles:
        - role: rhel-system-roles.selinux
          become: true
```

(This is how to playbook will look after copying and setting the alignments)

- To Verify:
- ansible all -m command -a “cat /etc/selinux/config”

Q18). Setting a Cronjob

$ vim crontab.yml

```
---
 - name: Cronjob setup
   hosts: all
   tasks:
         cron:
               name: Cron
               user: natasha
               state: present
               minute: */2
               job: 'logger " EX294 in Progress " '
```

$ ansible-playbook crontab.yml

- To Verify:
- ansible all -m command -a "crontab -l"