

# Red-Hat

## Exam Questions EX294

Red Hat Certified Engineer (RHCE) exam



**NEW QUESTION 1**

- (Exam Topic 2)

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 webserver host group.

A. Mastered

B. Not Mastered

**Answer: A**

**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: webserver

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
```

**NEW QUESTION 2**

- (Exam Topic 2)

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 webserver 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.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 webserver host group and uses the phphello role.

--> When implemented, browsing to hosts in the webserver 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

A. Mastered

B. Not Mastered

**Answer: A**

**Explanation:**

Solution as:

```
# pwd
```

```
/home/admin/ansible/
```

```
# vim balancer.yml
```

```
--
```

```
- name: Including phphello role hosts: webserver
```

```
roles:
```

```
- ./roles/phphello
```

```
- name: Including balancer role hosts: balancer
```

```
roles:
```

```
- ./roles/balancer wq!
```

```
# ansible-playbook balancer.yml --syntax-check
```

```
# ansible-playbook balancer.yml
```

**NEW QUESTION 3**

- (Exam Topic 2)

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 question.

hwreport.txt file consists. my\_sys=hostname

my\_BIOS=biosversion my\_MEMORY=memory my\_vda=vdasize my\_vdb=vdbsize

A. Mastered

B. Not Mastered

**Answer: A**

**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
```

#### NEW QUESTION 4

- (Exam Topic 2)

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 webserver's 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

- A. Mastered
- B. Not Mastered

**Answer:** A

#### 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 [webserver: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
```

#### NEW QUESTION 5

- (Exam Topic 2)

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 serevrblab.example.com serverb
* 172.25.250.12 serevrc.lab.example.com serverc
* 172.25.250.13 serevrld.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
```

- A. Mastered  
B. Not Mastered

**Answer:** A

**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
```

**NEW QUESTION 6**

- (Exam Topic 1)

Create a Shell script /root/program:

The shell script will come back to "user" parameter when you are entering "kernel" parameter.

The shell script will come back to "kernel" when you are entering "user" parameter.

It will output the standard error when this script "usage:/root/program kernel|user" don't input any parameter or the parameter you inputted is entered as the requirements.

- A. Mastered  
B. Not Mastered

**Answer:** A

**Explanation:**



```
[root@server1 virtual]# cat /root/program
#!/bin/bash
param1="$1"
if [ "$param1" == "kernel" ]; then
echo "user"
elif [ "$param1" == "user" ]; then
echo "kernel"
else
echo "usage:/root/program kernel|user"
if
[root@server1 ~]# chmod +x /root/program
```

**NEW QUESTION 7**

- (Exam Topic 1)

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

- A. Mastered  
B. Not Mastered

**Answer:** A

**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

#### NEW QUESTION 8

- (Exam Topic 1)

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

A. Mastered

B. Not Mastered

**Answer:** A

**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
```

#### NEW QUESTION 10

.....



## Thank You for Trying Our Product

### We offer two products:

1st - We have Practice Tests Software with Actual Exam Questions

2nd - Questions and Answers in PDF Format

### EX294 Practice Exam Features:

- \* EX294 Questions and Answers Updated Frequently
- \* EX294 Practice Questions Verified by Expert Senior Certified Staff
- \* EX294 Most Realistic Questions that Guarantee you a Pass on Your First Try
- \* EX294 Practice Test Questions in Multiple Choice Formats and Updates for 1 Year

**100% Actual & Verified — Instant Download, Please Click**  
**[Order The EX294 Practice Test Here](#)**