



**Free Questions for EX294 by vceexamstest**

**Shared by Hodge on 22-07-2024**

**For More Free Questions and Preparation Resources**

**Check the Links on Last Page**

# Question 1

---

## Question Type: MultipleChoice

---

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 webservers host group.

## Options:

---

**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: webservers
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
```

## Answer:

---

A

## Question 2

---

### Question Type: MultipleChoice

---

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

--> Runs over all managed hosts.

--> Uses the timesync role.

--> Configures the role to use the time server 192.168.10.254 ( Hear in redhat lab

use "classroom.example.com" )

--> Configures the role to set the iburst parameter as enabled.

### Options:

---

**A-** Explanation:

Solution as:

```
# pwd
```

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

```
# sudo yum install rhel-system-roles.noarch -y
```

```
# cd roles/
```

```
# ansible-galaxy list
```

```
# cp -r /usr/share/ansible/roles/rhelsystem-roles.timesync .
# vim timesync.yml
---
- name: timesynchronization
hosts: all
vars:
timesync_ntp_provider: chrony
timesync_ntp_servers:
- hostname: classroom.example.com _ in exam its ip-address
iburst: yes
timezone: Asia/Kolkata
roles:
- rhel-system-roles.timesync
tasks:
- name: set timezone
timezone:
name: '{{ timezone }}'
:wq!
timedatectl list-timezones | grep india
# ansible-playbook timesync.yml --syntax-check
# ansible-playbook timesync.yml
# ansible all -m shell -a 'chronyc sources -v'
# ansible all -m shell -a 'timedatectl'
# ansible all -m shell -a 'systemctl is-enabled chronyd'
```



**Answer:**

---

A

## Question 3

---

**Question Type:** MultipleChoice

---

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 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.

**Options:**

---

**A-** Explanation:

Solution as:

# pwd

```
home/admin/ansible/  
# vim packages.yml  
---  
- name: Install the packages  
hosts: dev,test,prod  
vars:  
- php_pkg: php  
- mariadb_pkg: mariadb  
tasks:  
- name: install the packages  
yum:  
name:  
- '{{ php_pkg }}'  
- '{{ mariadb_pkg }}'  
state: latest  
- name: install the devops tool packages  
hosts: dev  
tasks:  
- name: install devepment tools  
yum:  
name: '@Development Tools'  
state: latest  
- name: upgrade all the packages  
yum:  
name: '*'  
state: latest
```

```
exclude: kernel*
!wq
# ansible-playbook package.yml ---syntax-check
# ansible-playbook package.yml
```

## Answer:

---

A

## Question 4

---

### Question Type: MultipleChoice

---

Create and run an Ansible ad-hoc command.

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

--> Create a shell script called yum-pack.sh that runs an Ansible ad-hoc command to create yum-repository on each of the managed nodes as follows:

--> repository1

-----

1. The name of the repository is EX407
2. The description is "Ex407 Description"
3. The base URL is [http://content.example.com/rhel8.0/x86\\_64/dvd/BaseOS/](http://content.example.com/rhel8.0/x86_64/dvd/BaseOS/)
4. GPG signature checking is enabled
5. The GPG key URL is [http://content.example.com/rhel8.0/x86\\_64/dvd/RPM-GPG-KEYredhat-release](http://content.example.com/rhel8.0/x86_64/dvd/RPM-GPG-KEYredhat-release)
6. The repository is enabled

--> repository2

-----

1. The name of the repository is EXX407
2. The description is "Exx407 Description"
3. The base URL is [http://content.example.com/rhel8.0/x86\\_64/dvd/AppStream/](http://content.example.com/rhel8.0/x86_64/dvd/AppStream/)
4. GPG signature checking is enabled
5. The GPG key URL is [http://content.example.com/rhel8.0/x86\\_64/dvd/ RPM-GPG-KEYredhat-](http://content.example.com/rhel8.0/x86_64/dvd/RPM-GPG-KEYredhat-)

release

6. The repository is enabled

## Options:

---

**A-** Explanation:

Solution as:

```
# pwd
```

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

```
# vim yum-pack.sh
```

```
#!/bin/bash
```

```
ansible all -m yum_repository -a 'name=EX407 description='Ex407 Description'  
baseurl=http://content.example.com/rhel8.0/x86_64/dvd/BaseOS/ gpgcheck=yes  
gpgkey=http://content.example.com/rhel8.0/x86_64/dvd/RPM-GPG-KEY-redhat-release  
enabled=yes'
```

```
ansible all -m yum_repository -a 'name=EXX407 description='Exx407 Description'  
baseurl=http://content.example.com/rhel8.0/x86_64/dvd/AppStream/ gpgcheck=yes  
gpgkey=http://content.example.com/rhel8.0/x86_64/dvd/RPM-GPG-KEY-redhat-release  
enabled=yes'
```

```
:!wq
```

```
# chmod +x yum-pack.sh
```

```
# bash yum-pack.sh
```

```
# ansible all -m command -a 'yum repolist all'
```

**Answer:**

---

A

## Question 5

---

**Question Type: MultipleChoice**

---

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 webservers 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

## Options:

---

**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
[webservers: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
```



**Answer:**

---

A

## Question 6

---

**Question Type:** MultipleChoice

---

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/inventory.

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 1 is a member of group dev. node2 is a member of group test, node3 is a member of group proxy, node4 and node 5 are members of group prod. Also, prod is a member of group webservers.

## Options:

---

### A- Explanation:

```
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
```

## Answer:

---

A

## Question 7

---

### Question Type: MultipleChoice

---

Create a playbook `/home/bob/ansible/motd.yml` that runs on all inventory hosts and does the following: The playbook should replace 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".

## Options:

---

A- Explanation:

/home/sandy/ansible/apache.yml

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

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

## Answer:

---

A

## Question 8

---

Question Type: MultipleChoice

---

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"

### Options:

---

**A-** Explanation:

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

...

**Answer:**

---

A

## Question 9

---

**Question Type:** MultipleChoice

---

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'

**Options:**

---

**A-** Explanation:

Solution as:

```
- name: Creates a cron file under /etc/cron.d
```

```
cron:
```

```
  name: datejob
```

```
  hour: "12"
```

```
  user: root
```

```
  job: "date >> /root/ datefile"
```

**Answer:**

---

A

## Question 10

---

**Question Type:** MultipleChoice

---

In /home/sandy/ansible/ create a playbook called logvol.yml. In the play create a logical volume called lv0 and make it of size 1500MiB on volume group vg0. If there is not enough space in the volume group print a message "Not enough space for logical volume" and then make a 800MiB lv0 instead. If the volume group still doesn't exist, create a message "Volume group doesn't exist" Create an xfs



filesystem on all lv0 logical volumes. Don't mount the logical volume.

### Options:

---

**A-** Explanation:

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
```

**Answer:**

---

A

## Question 11

---

**Question Type:** MultipleChoice

---

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
```

## Options:

---

A- Explanation:

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
```

## Answer:

---

A

## Question 12

---

**Question Type:** MultipleChoice

---

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.

### Options:

---

**A-** Explanation:

Solution as:

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



```
- name: change bios
  lineinfile:
    line: BIOS={{ansible_bios_version}}
    regex: ^BIOS
    state: present
    path: /root/report.txt
- name: change vda
  lineinfile:
    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
  lineinfile:
    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
```



**Answer:**

---

A

**To Get Premium Files for EX294 Visit**

<https://www.p2pexams.com/products/ex294>

**For More Free Questions Visit**

<https://www.p2pexams.com/redhat/pdf/ex294>

