



Red-Hat

Exam Questions EX294

Red Hat Certified Engineer (RHCE) exam

NEW QUESTION 1

- (Exam Topic 2)

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/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 and these user should be 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 these user should be 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.

while practising you to create these file hear. But in exam have to download as per questation.

user_list.yml file consist:

--

user:

- name: user1 job: developer

- name: user2 job: manager

A. Mastered

B. Not Mastered

Answer: A

Explanation:

Solution as:

pwd

/home/admin/ansible

#

wget http://classroom.example.com/user_list.yml

cat user_list.yml

vim create_user.yml

--

- name: hosts: all vars_files:

- ./user_list.yml

- ./vault.yml tasks:

- name: creating groups group:

name: "{{ item }}" state: present

loop:

- devops

- opsmgr

- name: creating user user:

name: "{{ item.name }}" state: present

groups: devops

password: "{{ dev_pass|password_hash('sha512') }}" loop: "{{ user }}"

when: (inventory_hostname in groups['dev'] or inventory_hostname in groups['test']) and item.job == "developer"

- name: creating user user:

name: "{{ item.name }}" state: present

groups: opsmgr

password: "{{ mgr_pass|password_hash('sha512') }}" loop: "{{ user }}"

when: inventory_hostname in groups['prod'] and item.job == "manager" wq!

ansible-playbook create_user.yml --vault-password-file=password.txt --syntax-check

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

NEW QUESTION 2

- (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: firewallld http_srv: httpd firewall_srv: firewallld 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 firewallld:
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
```

NEW QUESTION 3

- (Exam Topic 2)

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.

- A. Mastered
- B. Not Mastered

Answer: 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
```

NEW QUESTION 4

- (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 questation.
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 5

- (Exam Topic 2)

Rekey an existing Ansible vault as follows:

*
Download Ansible vault from [http:// classroom.example.com /secret.yml](http://classroom.example.com/secret.yml) to `/home/ admin/ansible/`
* The current vault password is curabete
* The new vault password is newvare
* The vault remains in an encrypted state with the new password

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution as:

```
# pwd
/home/admin/ansible/
#
wget http://classroom.example.com/secret.yml
# chmod 0600 newpassword.txt
# ansible-vault rekey vault.yml --new-vault-password-file=newpassword.txt
```

NEW QUESTION 6

- (Exam Topic 2)

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.

- A. Mastered
- B. Not Mastered

Answer: 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'
```

NEW QUESTION 7

- (Exam Topic 2)

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://classroom.example.com /role1.tar.gz](http://classroom.example.com/role1.tar.gz) The name of this role should be balancer
[http://classroom.example.com /role2.tar.gz](http://classroom.example.com/role2.tar.gz) The name of this role should be phphello

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution as:

```
# pwd
/home/admin/ansible/roles
# vim install.yml
```



```
--
src: http://classroom.example.com/role1.tar.gz name: balancer
src: http://classroom.example.com/role2.tar.gz name: phphello
wq!
# pwd
/home/admin/ansible
# ansible-galaxy install -r roles/install.yml -p roles
```

NEW QUESTION 8

- (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 9

- (Exam Topic 1)

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'

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution as:

```
- name: Creates a cron file under /etc/cron.d
cron:
  name: datejob
  hour: "12"
  user: root
  job: "date >> /root/ datefile"
```

NEW QUESTION 10

- (Exam Topic 1)

Create a role called sample-apache in /home/sandy/ansible/roles that enables and starts httpd, enables and starts the firewall and allows the webserver service.

Create a template called index.html.j2 which creates and serves a message from /var/www/html/index.html Whenever the content of the file changes, restart the webserver service.

Welcome to [FQDN] on [IP]

Replace the FQDN with the fully qualified domain name and IP with the ip address of the node using ansible facts. Lastly, create a playbook in /home/sandy/ansible/ called apache.yml and use the role to serve the index file on webserver hosts.

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

/home/sandy/ansible/apache.yml


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

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

```
---
# tasks file for sample-apache
- name: enable httpd
  service:
    name: httpd
    state: started
    enabled: true
- name: enable firewall
  service:
    name: firewalld
    state: started
    enabled: true
- name: firewall http service
  firewalld:
    service: http
    state: enabled
    permanent: yes
    immediate: yes
- name: index
  template:
    src: templates/index.html.j2
    dest: /var/www/html/index.html
  notify:
    - restart
```

/home/sandy/ansible/roles/sample-apache/templates/index.html.j2

```
Welcome to ({ansible_fqdn}) ({ansible_default_ipv4.address})
```

In /home/sandy/ansible/roles/sample-apache/handlers/main.yml

```
- name: restart
  service:
    name: httpd
    state: restarted
```

NEW QUESTION 10

- (Exam Topic 1)

Create a file called requirements.yml in /home/sandy/ansible/roles to install two roles. The source for the first role is geerlingguy.haproxy and geerlingguy.php. Name the first haproxy-role and the second php-role. The roles should be installed in /home/sandy/ansible/roles.

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

in /home/sandy/ansible/roles vim requirements.yml

```
- src: geerlingguy.haproxy
  name: haproxy-role
- src: geerlingguy.php_role
  name: php_role
```

Run the requirements file from the roles directory:

ansible-galaxy install -r requirements.yml -p /home/sandy/ansible/roles

NEW QUESTION 11

- (Exam Topic 1)

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.

A. Mastered

B. Not Mastered

Answer: 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
```

NEW QUESTION 16

- (Exam Topic 1)

Create the users in the file usersjist.yml file provided. Do this in a playbook called users.yml located at

/home/sandy/ansible. The passwords for these users should be set using the lock.yml file from TASK7. When running the playbook, the lock.yml file should be unlocked with secret.txt file from TASK 7.

All users with the job of 'developer' should be created on the dev hosts, add them to the group devops, their password should be set using the pw_dev variable.

Likewise create users with the job of 'manager' on the proxy host and add the users to the group 'managers', their password should be set using the pw_mgr variable.

users_list.yml

```
users:
  - username: bill
    job: developer
  - username: chris
    job: manager
  - username: dave
    job: test
  - username: ethan
    job: developer
```

- A. Mastered
B. Not Mastered

Answer: A

Explanation:

ansible-playbook users.yml --vault-password-file=secret.txt

```
- name: create users
  hosts: all
  vars_files:
    - users_list.yml
    - lock.yml
  tasks:
    - name: create devops group nodes1
      group:
        name: devops
      when: ('dev' in group_names)
    - name: create manager group nodes45
      group:
        name: manager
      when: ('prod' in group_names)
    - name: create devs should happen on node1
      user:
        name: "{{item.username}}"
        groups: devops
        password: "{{ pw_dev | password_hash('sha512') }}"
      when: ('dev' in group_names) and ('developer' in item.job)
      loop: "{{users}}"
    - name: create managers on node45
      user:
        name: "{{item.username}}"
        groups: manager
        password: "{{ pw_mgr | password_hash('sha512') }}"
      when: ('prod' in group_names) and ('manager' in item.job)
      loop: "{{users}}"
```

NEW QUESTION 21

- (Exam Topic 1)

Create a playbook called timesvnc.yml in /home/sandy/ansible using rhel system role timesync. Set the time to use currently configured ntp with the server 0.uk.pool.ntp.org. Enable burst. Do this on all hosts.

- A. Mastered
B. Not Mastered

Answer: A

Explanation:

Solution as:


```
- name: use rhel system role
hosts: all
roles:
  - rhel-system-roles.timesync
timesync_ntp_servers:
  - hostname: 0.uk.pool.ntp.org
  iburst: yes
```

NEW QUESTION 26

- (Exam Topic 1)

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
```

- A. Mastered
B. Not Mastered

Answer: 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
```

NEW QUESTION 30

.....

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](#)