

EX294^{Q&As}

Red Hat Certified Engineer (RHCE) exam for Red Hat Enterprise Linux
8 Exam

Pass RedHat EX294 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

<https://www.pass2lead.com/ex294.html>

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by RedHat
Official Exam Center

- ⚙️ **Instant Download** After Purchase
- ⚙️ **100% Money Back** Guarantee
- ⚙️ **365 Days** Free Update
- ⚙️ **800,000+** Satisfied Customers



QUESTION 1

There are Mail servers, Web Servers, DNS Servers and Log Server. Log Server is already configured. You should configure the mail server, web server and dns server to send the logs to log server.

A. Answer: See the for complete Solution below.

Correct Answer: A

According to question, log server is already configured. We have to configure the mail, web and dns server for log redirection. In mail, web and dns server:

1.

```
vi /etc/syslog.conf mail.* @logserveraddress
```

2.

```
service syslog restart
```

mail is the facility and * means the priority. It sends logs of mail services into log server.

QUESTION 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 webservers host group.curl

--> 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 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.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. Answer: See the for complete Solution below.

Correct Answer: A

Solution as:

```
# pwd
```

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

```
# vim balancer.yml
```

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

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

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

```
roles:
```

```
  -./roles/balancer wq! # ansible-playbook balancer.yml --syntax-check # ansible-playbook balancer.yml
```

QUESTION 3

Prevent Mary from performing user configuration tasks in your system.

A. Answer: See the for complete Solution below.

Correct Answer: A



```
Modify the /etc/cron.deny, add:  
[root@server1 ~]# cat /etc/cron.deny  
mary
```

Conclusions:

1. I find that it is common to add various service access limits in the exam RHCE. The exercises like:

require one network segment can be accessed another network segments can not be accessed, the following are some conclusions for various service:

tcp_wrappers:/etc/hosts.allow,/etc/hosts.deny

tcp_wrappers can filter the TCP's accessing service. TCP whether has the filtering function which depends on this service whether use the function library of tcp_wrappers, or this service whether has the xinetd process of starting function of

tcp_wrappers. tcp_wrappers's main configuration file is /etc/hosts.allow,/etc/hosts.deny.

And the priority of the documents in hosts.allow is higher than hosts.deny. Visit will be passed if no match was found.

sshd,vsftpd can use the filtering service of tcp_wrappers.

Configuration example:

```
sshd:.example.com 192.168.0. 192.168.0.0/255.255.255.0 150.203.  
EXCEPT 150.203.6.66
```

Notice:

The two configuration files' syntax can refer to hosts_access (5) and hosts_options(5) sshd_config There are four parameters in this configuration file: DenyUsers, AllowUsers, DenyGroups, AllowGroups, they are used to limit some users or

user groups to proceed Remote Login through the SSH. These parameters' priority level is DenyUsers->AllowUsers->DenyGroups->AllowGroups Configuration example:

```
AllowUsers tim rain@192.168.1.121 kim@*.example.com
```

httpd Service Through the /etc/httpd/conf/httpd.conf in parameters, can add to control the url access. Just as:

```
<VirtualHost *:80>
DocumentRoot /var/http/virtual
ServerName www1.example.com
<Directory /var/http/virtual/limited>
Options Indexes MultiViews FollowSymlinks
order deny,allow
deny from all
allow from 192.168.0.
</Directory>
</VirtualHost>
```

Notice:

So pay attention, deny\'s and allow\'s priority level in order deny,allow is: the backer has the higher priority level. But here, allow\'s priority has a higher priority level.

nfs Service

nfs service directly control the visits through file /etc/exports, just as:

```
/common *.example.com(rw, sync) 192.168.0.0/24(ro, sync)
```

samba Service

Parameter hosts allow in /etc/samba/smb.conf which is used as Access Control, just as:

```
hosts allow = 192.168.0. 192.168.1.0/255.255.255.0 .example.com
```

2.

Paying attention to use Mount parameters: _netdev,defaults when you are mounting ISCSI disk.

3.

Stop the NetworkManager /etc/init.d/NetworkManager stop chkconfig NetworkManager off

4.

When you are deploying ifcfg-ethX, add parameters: PEERDNS=no

5.

Empty the firewall in RHCSARHCE:

6.

Narrow lv steps:

7.

Mount the using command - swap which is newly added in /etc/fstab

8.

If Verification is not passed when you are installing software, can import public key: rpm import /etc/pki/ rpm.../...release and so on. In yum.repo, you also can deploy gpgkey, for example, gpgkey=/etc/pki/rpm.../ ...release

9.

When you are using "Find" command to search and keep these files, paying attention to use cp -a to copy files if you use user name and authority as your searching methods.

```
iptables -F
```

```
iptables -X
```

```
iptables -Z
```

```
/etc/init.d/iptables save
```

```
1.umount /dev/mapper/lv
```

```
2.e2fsck -f /dev/mapper/lv
```

```
3.resize2fs /dev/mapper/lv 100M
```

```
4.lvreduce -L 50M /dev/mapper/lv
```

```
5.mount -a
```

QUESTION 4

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. Answer: See the for complete Solution below.

Correct Answer: A

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}}"
```

QUESTION 5

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. Answer: See the for complete Solution below.

Correct Answer: A

/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

/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
```

QUESTION 6

SIMULATION

Create the directory /storage and group owner should be the sysusers group.

A.

Correct Answer: Please see explanation

QUESTION 7

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` The name of this role should be balancer

`http://classroom.example.com/role2.tar.gz` The name of this role should be phphello

A. Answer: See the for complete Solution below.

Correct Answer: A

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
```

QUESTION 8

SIMULATION

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.

Correct Answer: Please see explanation

QUESTION 9

Add a cron schedule to take full backup of /home on every day at 5:30 pm to /dev/st0 device.

A. Answer: See the for complete Solution below.

Correct Answer: A

1.

```
vi /var/schedule 30 17 * * * /sbin/dump -0u /dev/st0 /dev/hda7
```

2.

```
crontab /var/schedule
```

3.

```
service crond restart
```

We can add the cron schedule either by specifying the scripts path on /etc/crontab file or by creating on text file on crontab pattern.

cron helps to schedule on recurring events. Pattern of cron is: Minute Hour Day of Month Month Day of Week Commands

0-59 0-23 1-31 1-12 0-7 where 0 and 7 mean Sunday.

Note * means every. To execute the command on every two minutes */2.

QUESTION 10

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. Answer: See the for complete Solution below.

Correct Answer: A

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
```

QUESTION 11

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: "Test"

--> On hosts in the prod host group, the line reads: "Production"

A. Answer: See the for complete Solution below.

Correct Answer: A

Solution as: # pwd /home/admin/ansible # vim modify.yml

-name:

hosts: all

tasks:

-

name:

copy:

content: "Development"

dest: /etc/issue

when: inventory_hostname in groups[\'dev\']

-

name:

copy:

content: "Test"

dest: /etc/issue

when: inventory_hostname in groups[\'test\']

-

name:

copy:

content: "Production"

dest: /etc/issue

when: inventory_hostname in groups[\'prod\']

wq

ansible-playbook modify.yml ?syntax-check

ansible-playbook modify.yml

QUESTION 12

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.

A. Answer: See the for complete Solution below.

Correct Answer: A

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
```

QUESTION 13

Given the kernel of a permanent kernel parameters: sysctl=1. It can be shown on cmdline after restarting the system. Kernel of /boot/grub/grub.conf should be a34dded finally, as:

A. Answer: See the for complete Solution below.

Correct Answer: A

```
Kernel of /boot/grub/grub.conf should be added finally, as:

kernel /vmlinuz-2.6.32-279.1.1.e16.x86_64 ro
root=/dev/mapper/vgsrv-root
rd_LVM_LV=vgsrv/root      rd_NO_LUKS  LANG=en_US.UTF-8
rd_LVM_LV=vgsrv/swap rd_NO_MD
SYSFONT=latercyrheb-sun16 crashkernel=auto KEYBOARDTYPE=pc
KEYTABLE=us rd_NO_DM rhgb quiet
rhgb quiet sysctl=1
```

QUESTION 14

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.

A. Answer: See the for complete Solution below.

Correct Answer: A

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]

node1.example.com

[test]

node2.example.com

[proxy]

node3.example.com [prod] node4.example.com node5.example.com [webservers:children] prod

QUESTION 15

```
# vim /root/program
# !/bin/bash

if [ $# -ne 1 ];then
    echo "usage:/root/program kernel|user"
else
    if [ "$1" -eq "kernel"];then
        echo "user"
    elif [ "$1" -eq "user"];then
        echo "kernel"
    else
        echo "usage:/root/program kernel|user"
    fi
fi
```

Test:

```
# chmod a+x /root/program
./root/program kernel
./root/program user
./root/program lll
```

SIMULATION

You access the iscsi shared storage. The storage server ip is 172.24.30.100. Separate of 1500M space, format as ext3

file system, mount under /mnt/data, and make sure the root-start automatically mount.

A.

Correct Answer: Please see explanation

[Latest EX294 Dumps](#)

[EX294 Exam Questions](#)

[EX294 Braindumps](#)