

DUMPS ARENA

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

RedHat EX294

Version Demo

Total Demo Questions: 5

Total Premium Questions: 35

Buy Premium PDF

<https://dumpsarena.co>

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

Topic Break Down

Topic	No. of Questions
Topic 1, LAB SETUP	19
Topic 2, LAB SETUP – 2	16
Total	35

QUESTION NO: 1 - (SIMULATION)

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 question.

`user_list.yml` file consist:

user:

- name: user1

job: developer

- name: user2

job: manager

ANSWER: SeetheExplanationforcompleteSolutionbelow.

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
```

QUESTION NO: 2 - (SIMULATION)

Create a file called requirements.yml in /home/sandy/ansible/roles a file called role.yml in /home/sandy/ansible/. The haproxy-role should be used on the proxy host. And when you curl <http://node3.example.com> it should display "Welcome to node4.example.com" and when you curl again "Welcome to node5.example.com" The php-role should be used on the prod host.

ANSWER: SeetheExplanationforcompleteSolutionbelow.

Explanation:

Solution as:

```
- name: install haproxy and php roles
hosts: all
vars:
  haproxy_backend_servers:
    - name: web1
      address: node4.example.com
    - name: web2
      address: node5.example.com
tasks:
  - name: import haproxy
    include_role: haproxy-role
    when: "proxy" in group_names
  - name: import php
    include_role: php-role
    when: "prod" in group_names
```

Check the proxy host by curl <http://node3.example.com>

QUESTION NO: 3 - (SIMULATION)

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

ANSWER: SeetheExplanationforcompleteSolutionbelow.

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
```

QUESTION NO: 4 - (SIMULATION)

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.

ANSWER: SeetheExplanationforcompleteSolutionbelow.

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'
```

QUESTION NO: 5 - (SIMULATION)

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

ANSWER: SeetheExplanationforcompleteSolutionbelow.

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:

```
regex: "{{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
```