



Free Questions for EX294 by certsinside

Shared by Hampton on 24-05-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

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

Options:

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
```

Answer:

A

Question 2

Question Type: MultipleChoice

Rekey an existing Ansible vault as follows:

* Download Ansible vault from `http://classroom.example.com/secret.yml` to `/home/`

`admin/ansible/`

* The current vault password is `curabete`

* The new vault password is `newware`

* The vault remains in an encrypted state with the new password

Options:

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
```

Answer:

A

Question 3

Question Type: MultipleChoice

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"

Options:

A- Explanation:

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
```

Answer:

A

Question 4

Question Type: MultipleChoice

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/](http://classroom.example.com/hwreport.empty)

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

Options:

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
```

Answer:

A

Question 5

Question Type: MultipleChoice

Generate a hosts file:

* [Download an initial template file hosts.j2 from http://classroom.example.com/](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 serevrb.lab.example.com serverb

172.25.250.12 serevrc.lab.example.com serverc

172.25.250.13 serevrd.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

Options:

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
```


Answer:

A

Question 6

Question Type: MultipleChoice

Create an Ansible vault to store user passwords as follows:

- * The name of the vault is valut.yml
- * The vault contains two variables as follows:
 - dev_pass with value wakennym
 - mgr_pass with value rocky
- * The password to encrypt and decrypt the vault is atenorth
- * The password is stored in the file /home/admin/ansible/password.txt

Options:

A- Explanation:

Solution as:

```
# pwd
```

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

```
# echo 'atenorth' >password.txt
```

```
# chmod 0600 password.txt
```

```
# ansible-vault create vault.yml --vault-password-file=password.txt
```

```
---
```

```
- dev_pass: wakennym
```

```
- mgr_pass: rocky
```

```
:wq
```

```
# cat vault.yml
```

```
$ANSIBLE_VAULT;1.1;AES256
```

```
36383862376164316436353665343765643331393433373564613762666531313034336438353662
```

```
3464346331346461306337633632393563643531376139610a343531326130663266613533633562
```

```
38623439316631306463623761343939373263333134353264333834353264343934373765643737
```

```
3535303630626666370a643663366634383863393338616661666632353139306436316430616334
```

```
65386134393363643133363738656130636532346431376265613066326162643437643064313863
```

```
6633333537303334333437646163343666666132316639376531
```

```
# ansible-vault view vault.yml
```

```
password:*****
```

```
---
```

```
- dev_pass: wakennym
```

```
- mgr_pass: rocky
```

Answer:

A

Question 7

Question Type: MultipleChoice

Create a playbook called web.yml as follows:

* The playbook runs on managed nodes in the "dev" host group

* Create the directory /webdev with the following requirements:

--> membership in the apache group

--> regular permissions: owner=r+w+execute, group=r+w+execute, other=r+execute

s.p=set group-id

* Symbolically link /var/www/html/webdev to /webdev

* Create the file /webdev/index.html with a single line of text that reads:

"Development"

--> it should be available on <http://servera.lab.example.com/webdev/index.html>

Options:

A- Explanation:

Solution as:

```
# pwd
```

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

```
# vim web.yml
```

```
---
```

```
- name:
```

```
hosts: dev
```

```
tasks:
```

```
- name: create group
```

```
  yum:
```

```
    name: httpd
```

```
    state: latest
```

```
- name: create group
```

```
  group:
```

```
    name: apache
```

```
    state: present
```

```
- name: creating directory
```

```
  file:
```

```
    path: /webdev
```

```
    state: directory
```

```
    mode: '2775'
```

group: apache

- sefcontext:

target: '/webdev/index.html'

setype: httpd_sys_content_t

state: present

- name: Apply new SELinux file context to filesystem

command: restorecon -irv

- name: creating symbolic link

file:

src: /webdev

dest: /var/www/html/webdev

state: link

force: yes

- name: creating file

file:

path: /webdev/index.html

sate: touch

- name: Adding content to index.html file

copy:

dest: /webdev/index.html

content: 'Development'

- name: add service to the firewall

firewalld:

service: http

permanent: yes

state: enabled

```
immediate: yes
- name: active http service
service:
name: httpd
state: restarted
enabled: yes
:wq
# ansible-playbook web.yml ---syntax-check
# ansible-playbook web.yml
```

Answer:

A

Question 8

Question Type: MultipleChoice

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

Options:

A- Explanation:

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
```


Answer:

A

Question 9

Question Type: MultipleChoice

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

Options:

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
```

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>

