

## Valid Red Hat Certified Architect (RHCA) EX447 Dumps Ensure Your Passing [Q14-Q35]



Valid Red Hat Certified Architect (RHCA) EX447 Dumps Ensure Your Passing  
EX447 Dumps Real Exam Questions Test Engine Dumps Training

### RedHat EX447 Exam Syllabus Topics:

TopicDetailsTopic 1- Set up directories containing multiple host variable files for some of your managed hosts- Create machine credentials to access inventory hostsTopic 2- Use special variables to override the host, port, or remote user Ansible uses for a specific host- Transform data with filters and pluginsTopic 3- Perform basic configuration of Ansible Tower after configuration- Populate variables with data from external sources using lookup pluginsTopic 4- Write an API scriptlet to launch a job- Manage inventories and credentials- Create a job workflow templateTopic 5- Implement loops using structures other than simple lists using lookup plugins and filters- Add those modified files back into the Git repositoryTopic 6- Run a task for a managed host on a different host, then control whether facts gathered by that task are delegated to the managed host or the other hostTopic 7- Inspect, validate, and manipulate variables containing networking information with filters- Update, modify and create files in a Git repositoryTopic 8- Override the name used in the inventory file with a different name or IP address- Create Ansible Tower users and teams and make associations of one to the otherTopic 9- Create a source control credential- Control privilege execution- Manage inventory variablesTopic 10- Use lookup and query functions to template data from external sources into playbooks and deployed template files

## NEW QUESTION 14

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

```
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
```

See the Explanation for complete Solution below.

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 15

Using the Simulation Program, perform the following tasks:

1. Use an ansible ad-hoc command, check the connectivity of your servers.
2. Use an ad-hoc ansible command, find the free space of your servers.
3. Use an ad-hoc ansible command, find out the memory usage of your servers.

4. Do an ls -l on the targets /var/log/messages file.
5. Tail the contents of the targets /var/log/messages file.  
See the Explanation for complete Solution below.

#### Explanation

1. ansible all -m ping
2. ansible all -a &#8220;/bin/df -h&#8221;;
3. ansible all-a &#8220;/usr/bin/free&#8221;;
4. ansible all -a &#8220;ls -l /var/log/messages&#8221;;
5. ansible local -b -a &#8220;tail /var/log/messages&#8221;;

#### NEW QUESTION 16

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.

See the Explanation for complete Solution below.

#### 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 17

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. See the Explanation for complete Solution below.

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]

node1 .example.com

[test]

node2.example.com

[proxy]

node3 .example.com

[prod]

node4.example.com

node5 .example.com

[webservers:children]

prod

## NEW QUESTION 18

Create the users in the fileusersjist.ymlfile provided. Do this in a playbook called users.yml located at

/home/sandy/ansible.The passwords for these users should be set using thelock.ymlfile from TASK7. When running the playbook, the lock.yml file should be unlocked withsecret.txtfile from TASK 7.

All users with the job of 'developer' should be created on thedevhosts, add them to the group devops, their password should be set using thepw\_devvariable. Likewise create users with the job of 'manager' on theproxy host and add the users to the group 'managers', their password should be set using thepw\_mgrvariable.

users\_list.yml

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

certify.vceprep.com

See the Explanation for complete Solution below.

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 devops should happen on node
      user:
        name: "{{item.username}}"
        groups: devops
        password: "{{ pw_dev | password_hash('sha512') }}"
      when: ('dev' in group_names) and ('devops' in group_names)
      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 group_names)
      loop: "{{users}}"
```

## NEW QUESTION 19

Create a playbook called `timesync.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.

See the Explanation for complete Solution below.

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
```

[certify.vceprep.com](https://certify.vceprep.com)

## NEW QUESTION 20

Using the Simulation Program, perform the following tasks:

Static Inventories Task:

1. Add a new group to your default ansible host file. call the group `[ec2]`
  2. Add a new host to the new group you created.
  3. Add a variable to a new host entry in the `/etc/ansible/hosts` file. Add the following. `localhost http_port=80 maxRequestsPerChild=808`
  4. Check to see if `maxRequestsPerChild` is pulled out with an ad-hoc command.
  5. Create a local host file and put a target group and then a host into it. Then ping it with an ad-hoc command.
- See the Explanation for complete Solution below.

Explanation

1. Edit the `/etc/ansible/hosts` file. Add a group.
2. Edit the `/etc/ansible/hosts` file. Add a user under the group you created.

3. Edit the /etc/ansible/hosts file. Find a host. if we add a variable called maxRequestsPerChild to the host it would look like this.

```
host1 maxRequestsPerChild=808
```

4. ansible ec2 -m shell -a '#echo {{ maxRequestsPerChild }}#';

5. Edit a local file. It could be called anything. Lets call it myhosts. Inside the file it would have a host like the following. [mygroup]

```
myusername1.mylabserver.com
```

## NEW QUESTION 21

Create a file called `adhoc.sh` in `/home/sandy/ansible` which will use `adhoc` commands to set up a new repository.

The name of the repo will be `EPEL`; the description `RHEL8`; the `baseurl` is `https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm`; there is no `gpgcheck`, but you should enable the repo.

\* You should be able to use an `bash` script using `adhoc` commands to enable repos. Depending on your lab setup, you may need to make this repo `state=absent`; after you pass this task.

See the Explanation for complete Solution below.

Explanation

```
chmod 0777 adhoc.sh
```

```
vim adhoc.sh
```

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

```
ansible all -m yum_repository -a '#name=EPEL description=RHEL8
```

```
baseurl=https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm gpgcheck=no enabled=yes#';
```

## NEW QUESTION 22

Create an `ansible vault` password file called `lock.yml` with the password `reallysafepw` in the

`/home/sandy/ansible` directory. In the `lock.yml` file define two variables. One is `ispw_dev` and the password is

`dev`; and the other is `ispw_mgr` and the password is `mgr`; Create a regular file called `secret.txt` which contains the password for `lock.yml`.

See the Explanation for complete Solution below.

Explanation

```
ansible-vault create lock.yml
```

New Vault Password: reallysafepw

Confirm: reallysafepw



In file:

```
pw_dev: dev  
pw_mgr: mgr
```

### NEW QUESTION 23

In `/home/sandy/ansible/` create a playbook called `the play` create a logical volume called `lv0` and make it of size `1500MiB` on volume group `vg`. If there is not enough space in the volume group print a message

Not enough space for logical vol `lv0` instead. If the volume group still doesn't exist, create a message; Volume group doesn't exist; filesystem on all `lv0` logical volumes. Don't mount the logical volume. See the Explanation for complete Solution below.

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
  debug:
    msg: Not enough space for logical volume
  when: ansible_lvm.vgs.vg0 is defined and ( (ansible_lvm.vgs.vg0.size_g | float ) < 1.5)
- name: create a smaller 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: create fs
  filesystem:
    dev: /dev/vg0/lv0
    fstype: xfs
  when: ansible_lvm.vgs.vg0 is defined
```

certify.vceprep.com

**RedHat EX447: Selling Red Hat Certified Architect (RHCA) Products and Solutions:**  
<https://www.vceprep.com/EX447-latest-vce-prep.html>