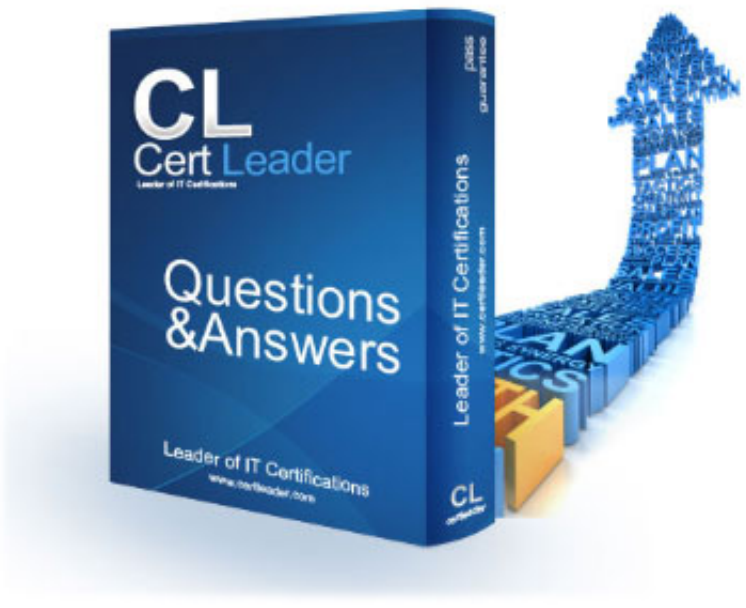
EX407 Dumps

Red Hat Certified Specialist in Ansible Automation exam

https://www.certleader.com/EX407-dumps.html



NEW QUESTION 1
CORRECT TEXT
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 pw_dev and the password is 'dev' and the other is pw_mgr and the password is 'mgr' Create a regular file called secret.txt which contains the password for lock.yml.

A. Mastered
B. Not Mastered

Answer: A

Explanation:
ansible-vault create lock.yml
New Vault Password: reallysafepw Confirm: reallysafepw

In file:

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

NEW QUESTION 2
CORRECT TEXT
Create an empty encrypted file called myvault.yml in /home/sandy/ansible and set the password to notsafepw. Rekey the password to iwejfj2221.

A. Mastered
B. Not Mastered

Answer: A

Explanation:
ansible-vault create myvault.yml
Create new password: notsafepw Confirm password: notsafepw ansible-vault rekey myvault.yml
Current password: notsafepw New password: iwejfj2221 Confirm password: iwejfj2221

NEW QUESTION 3
CORRECT TEXT
==========
==========
control.realmX.example.com _ workstation.lab.example.com
node1.realmX.example.com _ servera.lab.example.com
node2.realmX.example.com _ serverb.lab.example.com
node3.realmX.example.com _ serverc.lab.example.com
node4.realmX.example.com _ serverd.lab.example.com
node5.realmX.example.com
- username:root, password:redhat
- username:admin, password:redhat
note1. don??t change ??root?? or ??admin?? password.
note2. no need to create ssh-keygen for access, its pre-defined
note3. SELinux is in enforcing mode and firewalld is disabled/stop on whole managed hosts.
==========
==========
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 4
CORRECT TEXT
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.

A. Mastered
B. Not Mastered

Answer: A

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

NEW QUESTION 5
CORRECT TEXT
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/invenlory.
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 I is a member of group dev. nodc2 is a member of group test, node3 is a member of group proxy, nodc4 and node 5 are members of group prod. Also, prod is a member of group webservers.

A. Mastered
B. Not Mastered

Answer: A

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]
node 1 .example.com [test] node2.example.com [proxy]
node3 .example.com [prod] node4.example.com node5 .example.com [webservers:children]
prod

NEW QUESTION 6
CORRECT TEXT
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 7
CORRECT TEXT
==========
==========
control.realmX.example.com _ workstation.lab.example.com node1.realmX.example.com _ servera.lab.example.com node2.realmX.example.com _ serverb.lab.example.com node3.realmX.example.com _ serverc.lab.example.com node4.realmX.example.com _ serverd.lab.example.com node5.realmX.example.com
- username:root, password:redhat
- username:admin, password:redhat
note1. don??t change ??root?? or ??admin?? password.
note2. no need to create ssh-keygen for access, its pre-defined
note3. SELinux is in enforcing mode and firewalld is disabled/stop on whole managed hosts.
==========
==========
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 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 8
CORRECT TEXT
==========
==========
control.realmX.example.com _ workstation.lab.example.com node1.realmX.example.com _ servera.lab.example.com node2.realmX.example.com _ serverb.lab.example.com node3.realmX.example.com _ serverc.lab.example.com node4.realmX.example.com _ serverd.lab.example.com node5.realmX.example.com
- username:root, password:redhat
- username:admin, password:redhat
note1. don??t change ??root?? or ??admin?? password.
note2. no need to create ssh-keygen for access, its pre-defined
note3. SELinux is in enforcing mode and firewalld is disabled/stop on whole managed hosts.
==========
==========
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

A. Mastered
B. Not Mastered

Answer: 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 363838623761643164363536653437656433313934333735646137626665313130343364
38353662
346434633134646130633763363239356364353137613961 0a343531326130663266613 5
33633562
386234393166313064636237613439393732633331343532643338343532643439343737
65643737
353530363062666637 0a643663366634383863393338616661666632353139306436316 4
30616334
653861343933636431333637386561306365323464313762656130663261626434376430
64313863
663333353730333433343764616334366666613231663937653 1
ansible-vault view vault.yml password:******

- dev_pass: wakennym
- mgr_pass: rocky

NEW QUESTION 9
CORRECT TEXT
==========
==========
control.realmX.example.com _ workstation.lab.example.com
node1.realmX.example.com _ servera.lab.example.com
node2.realmX.example.com _ serverb.lab.example.com
node3.realmX.example.com _ serverc.lab.example.com
node4.realmX.example.com _ serverd.lab.example.com
node5.realmX.example.com
- username:root, password:redhat
- username:admin, password:redhat
note1. don??t change ??root?? or ??admin?? password.
note2. no need to create ssh-keygen for access, its pre-defined
note3. SELinux is in enforcing mode and firewalld is disabled/stop on whole managed hosts.
==========
==========
Generate a hosts file:
* Download an initial template file hosts.j2 from 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

A. Mastered
B. Not Mastered

Answer: 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

NEW QUESTION 10
CORRECT TEXT
==========
==========
control.realmX.example.com _ workstation.lab.example.com
node1.realmX.example.com _ servera.lab.example.com
node2.realmX.example.com _ serverb.lab.example.com
node3.realmX.example.com _ serverc.lab.example.com
node4.realmX.example.com _ serverd.lab.example.com
node5.realmX.example.com
- username:root, password:redhat
- username:admin, password:redhat
note1. don??t change ??root?? or ??admin?? password.
note2. no need to create ssh-keygen for access, its pre-defined
note3. SELinux is in enforcing mode and firewalld is disabled/stop on whole managed hosts.
==========
==========
Install and configure Ansible on the control-node control.realmX.example.com as follows:

--> Install the required packages
--> Create a static inventory file called /home/admin/ansible/inventory as follows: node1.realmX.example.com is a member of the dev host group
node2.realmX.example.com is a member of the test host group node3.realmX.example.com & node4.realmX.example.com are members of the prod host group
node5.realmX.example.com is a member of the balancers host group. prod group is a member of the webservers host group
--> Create a configuration file called ansible.cfg as follows:
--> The host inventory file /home/admin/ansible/inventory is defined
--> The location of roles used in playbooks is defined as /home/admin/ansible/ roles

A. Mastered
B. Not Mastered

Answer: A

Explanation:
Solution as:
Through physical host, login to workstation.lab.example.com with user root.
ssh root@workstation.lab.example.com
hostname workstation.lab.example.com
yum install platform-python*
su - admin
pwd
/home/admin/
vim .vimrc
mkdir -p ansible/roles
cd ansible
vim inventory [dev]
servera.lab.example.com [test] serverb.example.com [prod] serverc.example.com
serverd.example.com [balancer] serverd.lab.example.com [webservers:children] prod
!wq
# vim ansible.cfg [defaults]
inventory = ./inventory role_path = ./roles remote_user = admin ask_pass = false [privilege_escalation] become = true become_method = sudo become_user = root become_ask_pass = false
!wq
ansible all -–list-hosts

NEW QUESTION 10
CORRECT TEXT
==========
==========
control.realmX.example.com _ workstation.lab.example.com node1.realmX.example.com _ servera.lab.example.com node2.realmX.example.com _ serverb.lab.example.com node3.realmX.example.com _ serverc.lab.example.com node4.realmX.example.com _ serverd.lab.example.com
node5.realmX.example.com
- username:root, password:redhat
- username:admin, password:redhat
note1. don??t change ??root?? or ??admin?? password.
note2. no need to create ssh-keygen for access, its pre-defined
note3. SELinux is in enforcing mode and firewalld is disabled/stop on whole managed hosts.
==========
==========
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 14
CORRECT TEXT
Create a playbook called webdev.yml in 'home/sandy/ansible. The playbook will create a directory Avcbdev on dev host. The permission of the directory are 2755 and owner is webdev. Create a symbolic link from /Webdev to /var/www/html/webdev. Serve a file from Avebdev7index.html which displays the text "Development" Curl http://node1.example.com/webdev/index.html to test

A. Mastered
B. Not Mastered

Answer: A

Explanation:
Solution as:

```
- name: webdev
  hosts: dev
  tasks:
    - name: create webdev user
      user:
          name: webdev
          state: present
    - name: create a directory
       file:
          mode: '2755'
          path: /webdev
          state: directory
    - name: create symbolic link
      file:
         src: /webdev
         path: /var/www/html/webdev
         state: link
- name: create index.html
  copy:
     content: Development
     dest: /webdev/ index.html
- name: Install selinux policies
  yum:
        name: python3-policycoreutils
        state: present
- name: allow httpd from this directory
  sefcontext:
       target: '/webdev(/.*)?'
       setype: httpd_sys_content_t
       state: present
- name: restore the context
   shell: restorecon -vR /webdev
```

NEW QUESTION 18
CORRECT TEXT
Create a playbook that changes the default target on all nodes to multi-user tarqet. Do this in playbook file called target.yml in /home/sandy/ansible

A. Mastered
B. Not Mastered

Answer: A

Explanation:
- name: change default target hosts: all
tasks:
- name: change target file:
src: /usr/lib/systemd/system/multi-user.target dest: /etc/systemd/system/default.target state: link

NEW QUESTION 22
CORRECT TEXT
==========
control.realmX.example.com _ workstation.lab.example.com node1.realmX.example.com _ servera.lab.example.com node2.realmX.example.com _ serverb.lab.example.com node3.realmX.example.com _ serverc.lab.example.com node4.realmX.example.com _ serverd.lab.example.com node5.realmX.example.com

- username:root, password:redhat
- username:admin, password:redhat
note1. don??t change ??root?? or ??admin?? password.
note2. no need to create ssh-keygen for access, its pre-defined
note3. SELinux is in enforcing mode and firewalld is disabled/stop on whole managed hosts.
==========
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. 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 25
CORRECT TEXT
Create a file called packages.yml in /home/sandy/ansible to install some packages for the following hosts. On dev, prod and webservers install packages httpd, mod_ssl, and mariadb. On dev only install the development tools package. Also, on dev host update all the packages to the latest.

A. Mastered
B. Not Mastered

Answer: A

Explanation:
Solution as:

```
---
- name: install pack
  hosts: dev,test,webservers
  become: true
  tasks:
    - name: install on all hosts in this play
      yum:
        name:
          - httpd
          - mod_ssl
          - mariadb
        state: latest
  - name: install on dev only
    yum:
      name:
        - '@Development tools'
      state: latest
    when: "dev" in group_names
~
```

** NOTE 1 a more acceptable answer is likely 'present' since it's not asking to install the latest
state: present

** NOTE 2 need to update the development node
- name: update all packages on development node yum:
name: '*' state: latest

NEW QUESTION 28
CORRECT TEXT
==========
control.realmX.example.com _ workstation.lab.example.com
node1.realmX.example.com _ servera.lab.example.com
node2.realmX.example.com _ serverb.lab.example.com
node3.realmX.example.com _ serverc.lab.example.com
node4.realmX.example.com _ serverd.lab.example.com
node5.realmX.example.com
- username:root, password:redhat
- username:admin, password:redhat
note1. don??t change ??root?? or ??admin?? password.
note2. no need to create ssh-keygen for access, its pre-defined
note3. SELinux is in enforcing mode and firewalld is disabled/stop on whole managed hosts.
==========
==========
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. Mastered
B. Not Mastered

Answer: 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

NEW QUESTION 32
CORRECT TEXT
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.rmp' 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.

A. Mastered
B. Not Mastered

Answer: A

Explanation:
chmod 0777 adhoc.sh vim adhoc.sh
#I/bin/bash
ansible all -m yum_repository -a 'name=EPEL description=RHEL8 baseurl=https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rmp gpgcheck=no enabled=yes'

NEW QUESTION 36
CORRECT TEXT
Using the Simulation Program, perform the following tasks:
Ad-Hoc Ansible Commands (Number Two) Task:
* 1. Use the ad-hoc command to make sure php is installed.
* 2. Use the ad-hoc command to make sure that php is installed and is the latest version.
* 3. Use the ad-hoc command to make sure that httpd is installed.
* 4. Use the ad-hoc command to remove httpd from the servers.

A. Mastered
B. Not Mastered

Answer: A

Explanation:
* 1. ansible all -b -m yum -a 'name=php state=present'
* 2. ansible all -b -m yum -a 'name=php state=latest'
* 3. ansible all -b -m yum -a 'name=httpd state=latest'
* 4. ansible all -b -m yum -a 'name=httpd state=absent'

NEW QUESTION 38
CORRECT TEXT
==========
control.realmX.example.com _ workstation.lab.example.com
node1.realmX.example.com _ servera.lab.example.com
node2.realmX.example.com _ serverb.lab.example.com
node3.realmX.example.com _ serverc.lab.example.com
node4.realmX.example.com _ serverd.lab.example.com
node5.realmX.example.com
- username:root, password:redhat
- username:admin, password:redhat
note1. don??t change ??root?? or ??admin?? password.
note2. no need to create ssh-keygen for access, its pre-defined
note3. SELinux is in enforcing mode and firewalld is disabled/stop on whole managed hosts.
==========
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. 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: 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

NEW QUESTION 43
.....

Thank You for Trying Our Product

* 100% Pass or Money Back

All our products come with a 90-day Money Back Guarantee.

* One year free update

You can enjoy free update one year. 24x7 online support.

* Trusted by Millions

We currently serve more than 30,000,000 customers.

* Shop Securely

All transactions are protected by VeriSign!

100% Pass Your EX407 Exam with Our Prep Materials Via below:

https://www.certleader.com/EX407-dumps.html