DUMPSBOSS.

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

RedHat EX294

Version Demo

Total Demo Questions: 6

Total Premium Questions: 61

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, Understand core components of Ansible	6
Topic 2, Install and configure an Ansible control node	4
Topic 3, Script administration tasks	1
Topic 4, Create Ansible plays and playbooks	4
Topic 5, Use Ansible roles and Ansible Content Collections	10
Topic 6, Create and use templates to create customized configuration files	11
Topic 7, Automate common Red Hat Enterprise Linux administration tasks	23
Topic 8, Mix Questions	2
Total	61

QUESTION NO: 1 - (SIMULATION)

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.**

ANSWER: See the explanation for the answer

Explanation:

Create an executable script that runs the `yum_repository` ad hoc module against all managed hosts.

```
cd /home/sandy/ansible
cat > adhoc.sh <<'EOF'
#!/bin/bash
ansible all -m ansible.builtin.yum_repository -a 'name=EPEL description=RHEL8
baseurl=https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rmp gpgcheck=no
enabled=yes state=present'
EOF
chmod 755 adhoc.sh
```

The resulting `/home/sandy/ansible/adhoc.sh` creates/enables the repository named `EPEL`, sets its description and requested base URL, and disables GPG checking. Run it with:

```
/home/sandy/ansible/adhoc.sh
```

If cleanup is required after validation, change `state=present` to `state=absent` and run the script again.

QUESTION NO: 2 - (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: See the explanation for the answer

Explanation:

Create `hwreport.yml` in the Ansible project directory with the following content. The `get_url` task first obtains the required empty report file, and the `replace` task substitutes each placeholder. Using `get(...)` ensures that an absent disk (or absent BIOS fact) is written as `NONE` rather than causing the play to fail.

```
- name: Create hardware reports
  hosts: all
  become: true
  gather_facts: true

  tasks:
    - name: Download the empty hardware report
      ansible.builtin.get_url:
        url: http://classroom.example.com/hwreport.empty
        dest: /root/hwreport.txt
        mode: '0644'

    - name: Insert hardware information into the report
      ansible.builtin.replace:
        path: /root/hwreport.txt
        regexp: '{{ item.placeholder }}'
        replace: '{{ item.value }}'
      loop:
        - placeholder: hostname
          value: '{{ inventory_hostname }}'
        - placeholder: biosversion
          value: '{{ ansible_facts.get("bios_version", "NONE") }}'
        - placeholder: memory
          value: '{{ ansible_facts.get("memtotal_mb", "NONE") }}'
        - placeholder: vdasize
          value: '{{ ansible_facts.get("devices", {}).get("vda", {}).get("size", "NONE")
}}'
        - placeholder: vdbsize
          value: '{{ ansible_facts.get("devices", {}).get("vdb", {}).get("size", "NONE")
}}'
```

Run it with:

```
ansible-playbook hwreport.yml
```

The resulting `/root/hwreport.txt` on every managed host contains the five required key-value lines, for example `my_sys=servera.example.com`. If the exam environment explicitly provides a different path after `classroom.example.com/`, use that exact supplied URL in the `get_url` task.

QUESTION NO: 3 - (SIMULATION)

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”

ANSWER: See the explanation for the answer

Explanation:

Create `/home/admin/ansible/modify.yml` with one task that selects the required content based on the host's inventory group membership:

```
---
- name: Set the issue banner by environment
  hosts: all
  become: true
  tasks:
    - name: Replace /etc/issue content
      ansible.builtin.copy:
        dest: /etc/issue
        content: "{{ issue_text }}\n"
        owner: root
        group: root
        mode: '0644'
      vars:
        issue_text: >-
          {% if inventory_hostname in groups['dev'] %}
          Development
          {% elif inventory_hostname in groups['test'] %}
          Test
          {% elif inventory_hostname in groups['prod'] %}
          Production
          {% endif %}
```

Run it from the Ansible project directory:

```
cd /home/admin/ansible
ansible-playbook --syntax-check modify.yml
ansible-playbook modify.yml
```

The `copy` module replaces the destination file, and the trailing newline in `content` ensures `/etc/issue` contains exactly one text line. `become: true` is required because normal users generally cannot modify `/etc/issue`.

QUESTION NO: 4 - (SIMULATION)

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.

ANSWER: See the explanation for the answer

Explanation:

Create `/home/sandy/ansible/packages.yml` with three plays: one for the common packages, and one for the development-host-only requirements.

```
---
- name: Install required packages on development, production, and web servers
  hosts: dev,prod,webservers
  become: true
  tasks:
    - name: Install HTTPD, SSL, and MariaDB packages
      yum:
        name:
          - httpd
          - mod_ssl
          - mariadb
        state: present

- name: Configure the development host
  hosts: dev
  become: true
  tasks:
    - name: Install Development Tools package group
      yum:
        name: '@Development Tools'
        state: present

    - name: Update every installed package to its latest version
      yum:
        name: '*'
        state: latest
```

`state: present` installs the requested common packages without unnecessarily upgrading them. The wildcard package name with `state: latest` updates all packages, and is limited to the `dev` host/group as required.

QUESTION NO: 5 - (SIMULATION)

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.

ANSWER: See the explanation for the answer

Explanation:

From `/home/sandy/ansible`, create the vault password file and then use it to create the encrypted vault file:

```
cd /home/sandy/ansible
printf '%s\n' 'reallysafepw' > secret.txt
chmod 600 secret.txt
ansible-vault create --vault-password-file=secret.txt lock.yml
```

When the editor opens for `lock.yml`, add these two variables:

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

Save and exit the editor. The resulting `lock.yml` must be Ansible Vault encrypted (its first line will be similar to `$ANSIBLE_VAULT;1.1;AES256`), while `secret.txt` is a regular file containing exactly:

```
reallysafepw
```

Alternatively, if using the interactive command shown in the prompt, run `ansible-vault create lock.yml`, enter `reallysafepw` twice when prompted, add the variables above, and then create `secret.txt` with the same password.

QUESTION NO: 6 - (SIMULATION)

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: â€œDevelopmentâ€□

--> On hosts in the test host group, the line reads: â€œTestâ€□

--> On hosts in the prod host group, the line reads: â€œProductionâ€□

ANSWER: See the explanation for the answer

Explanation:

Create `/home/admin/ansible/modify.yml` with the following content:

```
---
- name: Set /etc/issue according to host group
  hosts: all
  become: true
  tasks:
    - name: Set development issue message
      ansible.builtin.copy:
        dest: /etc/issue
        content: "Development\n"
      when: "'dev' in group_names"

    - name: Set test issue message
      ansible.builtin.copy:
        dest: /etc/issue
        content: "Test\n"
      when: "'test' in group_names"

    - name: Set production issue message
      ansible.builtin.copy:
        dest: /etc/issue
        content: "Production\n"
      when: "'prod' in group_names"
```

The `copy` module replaces the complete content of `/etc/issue`. The trailing newline makes each value a proper single text line. Verify and run it with:

```
cd /home/admin/ansible
ansible-playbook --syntax-check modify.yml
ansible-playbook modify.yml
```