

RHCE v8:

Q1. Install and configure Ansible on the control node control.example.com as follows:

- * Install the required packages
- * Create a static inventory file called /home/admin/ansible/inventory as follows:
 - node1.example.com is a member of the dev host group
 - node2.example.com is a member of the balancers host group
 - node3.example.com and node4.example.com are members of the prod host group

 - The 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

Q2. Create and run an Ansible ad-hoc command. As a system administrator, you will need to install software on the managed nodes. Create a shell script called yum-pack.sh that runs an Ansible ad-hoc command to create a yum repository on each of the managed nodes as follows: create a appstream and baseos repo.

1. The name of the repository is baseos and appstream/
2. The description is "RH294 Description"
3. The base URL for appstream is "http://content.example.com/rhel8.0/x86_64/dvd/AppStream/" and for baseos "http://content.example.com/rhel8.0/x86_64/dvd/BaseOS/"
4. GPG signature checking is enabled
5. The GPG key URL is "http://content.example.com/rhel8.0/x86_64/dvd/RPM-GPG-KEY-redhat-release"
6. The repository is enabled

Q3. Create a playbook called packages.yml that:

- Installs the php and mariadb packages on hosts in the dev, balancers, and prod host groups
- Installs the Development Tools package group on hosts in the dev host group
- Updates all packages to the latest version on hosts in the dev host group

Q4. Install the RHEL system roles package and create a playbook called timesync.yml that:

- Runs on all managed hosts
- Uses the timesync role
- Configures the role to use the time server 172.25.254.254
- Configures the role to set the iburst parameter as enabled

Q5. 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
- A template file index.html.j2 exists (you have to create this file) and is used to create the file /var/www/html/index.html with
Welcome to HOSTNAME on IPADDRESS
- where HOSTNAME is the fully qualified domain name of the managed node and IPADDRESS is the IP address of the managed node.
- Create a playbook called httpd.yml that uses this role as follows:

the following output:

- * The playbook runs on hosts in the webservers host group

Q6. 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://content.example.com/rh294/role1.tar.gz
The name of this role should be balancer
- http://content.example.com/rh294/role2.tar.gz
The name of this role should be phphello

Q7. Create a playbook called balance.yml as follows:

The playbook contains a play that runs on hosts in the balancers host group and uses the balancer role.

- This role configures a service to load balance web server requests between hosts in the webservers host group.
- When implemented, browsing to hosts in the balancers host group (for example http://node5.example.com) should produce the
Welcome to node3.example.com on 192.168.10.2
- Reloading the browser should return output from the alternate web server:
Welcome to node4.example.com on 192.168.10.2

following output:

- This role configures a service to load balance web server requests between hosts in the webservers host group.
- 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 192.168.10.2

- 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 the 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

along with various details of the PHP configuration including the version of PHP that is installed.

- * Similarly, browsing to `http://node4.example.com/hello.php`, should produce the following output:

Hello PHP World from node4.example.com

along with various details of the PHP configuration including the version of PHP that is installed.

Q8. Create a playbook called web 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=read+write+execute`, `group=read+write+execute`, `other=read+execute`
special permissions: `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`

- * Symbolically link `/var/www/html/webdev` to `/webdev`
- * Create the file `/webdev/index.html` with a single line of text that reads: Development

Q9. 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`

Q10. Generate a hosts file:

- * Download an initial template file called `hosts.j2` from <http://192.168.10.254/ex407/> 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`
- * Create a playbook called `gen_hosts.yml` that uses this template to generate the file `/etc/hosts` on hosts in the dev host group.
- * When completed, the file `/etc/hosts` 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

192.168.10.x node1.example.com node1
192.168.10.y node2.example.com node2
192.168.10.z node3.example.com node3
192.168.10.a node4.example.com node4
192.168.10.b node5.example.com node5
```

Q11. 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 keyvalue pair.

* Your playbook should:

- Download the file hwreport.empty from the URL <http://192.168.10.254/ex407/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

Q12. 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 balancers host group, the line reads: Test
 - On hosts in the prod host group, the line reads: Production

Q13. Rekey an existing Ansible vault as follows:

- * Download the Ansible vault from `"http://content.example.com/rh294/secret.yml"`
- * The current vault password is curabete
- * The new vault password is newvare
- * The vault remains in an encrypted state with the new password

Q14. 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://content.example.com/rh294/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 balancers host groups assigned the password from the dev_pass variable a 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 a 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.

Q15. Create Logical volumes in all nodes according to following requirements.

- * Create a new Logical volume named as 'data'

- * LV should be the member of 'research' Volume Group

- * LV size should be 1500M

- * It should be formatted with ext4 filesystem.

- If Volume Group does not exist then it should print the message "VG Not found"

- If the VG can not accomodate 1500M size then it should print "LV Can not be created with following size"

- then the LV should be created with 800M of size.

- Do not perform any mounting for this LV.