The IP addresses for these systems are set statically. Do not change these settings.

Hostname resolution is configured to resolve the fully qualified hostnames listed above, but also to resolve host shortnames.

**account information**

The root password for all systems is flectrag .

Do not change the root password. Unless otherwise specified, this will be the password used to access other systems and services. Also, unless otherwise specified, this password also applies to all accounts you create, or any service that requires a password to be set.

For convenience, SSH keys are preinstalled on all systems, allowing root access via SSH without entering a password. Do not make any modifications to the root SSH configuration file on the system.

The user account greg has been created on the Ansible control node. This account is pre-installed with SSH keys that allow SSH logins between the Ansible control node and each Ansible managed node. Do not make any modifications to the greg SSH configuration file on the system. You can use su from the root account to access this user account.

**Important information**

Unless otherwise specified, all your work (including Ansible playbooks, configuration files, host inventory, etc.) should be kept in the directory /home/greg/ansible on the control node and should be owned by the greg user. All Ansible related commands should be run by user greg from this directory on the Ansible control node.

**Additional Information**
Some exam items may require modification of the Ansible host inventory. It is your responsibility to ensure that all previous checklist groups and projects are preserved, coexisting with any other changes. You will also have to make sure that all default groups and hosts in the inventory retain any changes you make.

The firewall on the exam system is not enabled by default, and SELinux is in enforcing mode.

If additional software needs to be installed, your physical system and Ansible control node may already be set up to point to the following repositories on content:

http://content/rhel8.0/x86_64/dvd/BaseOS

http://content/rhel8.0/x86_64/dvd/AppStream

Some projects require additional files, which are already available at:

http://materials

Product documentation can be found at:

http://materials/docs/ansible/html

Other resources are also configured for you to use during the exam. Specific information about these resources will be provided in projects that require them.

**Important information**

Note that before scoring, your Ansible managed node system will be reset to the initial state at the start of the exam, and the Ansible playbook you have written will be run from the /home/greg/ansible directory on the control node by running as user greg to apply. After the playbook is running, your managed nodes are evaluated to see if they are configured as specified.

---

**👇RHCE exam questions👇**

---

**Do all the following steps on your system.**

Install and configure Ansible

Create and run Ansible ad hoc commands

Install the package

Using RHEL system roles

Install roles using Ansible Galaxy

Create and use roles

Using roles from Ansible Galaxy

Create and use logical volumes

Generate hosts file

Modify file content

Create a web content directory

Generate hardware report

Create a password vault

Create user account

Update keys for Ansible repositories

---

## 👇RHCE exam question content👇

---

### 😀 Install and configure Ansible

Install and configure Ansible on the control node 172.25.250.254 as described below:

Install the required packages

Create a static inventory file named /home/greg/ansible/inventory to meet the following requirements:

172.25.250.9 is a member of the dev host group

172.25.250.10 is a member of the test host group

172.25.250.11 and 172.25.250.12 are members of the prod host group

172.25.250.13 is a member of the balancers host group

The prod group is a member of the webservers host group

Create a configuration file named /home/greg/ansible/ansible.cfg to meet the following requirements:

The host inventory file is /home/greg/ansible/inventory

The locations of roles used in playbooks include /home/greg/ansible/roles

Problem solving method:

```
[greg@bastion ~]$ sudo yum install -y ansible
[greg@bastion ~]$ mkdir -p /home/greg/ansible
[greg@bastion ~]$ cd /home/greg/ansible
[greg@bastion ansible]$ vim inventory
[dev]
172.25.250.9
[test]
172.25.250.10
[prod]
172.25.250.11
172.25.250.12
[balancers]
172.25.250.13
[webservers:children]
prod
[all:vars]
ansible_user=root
ansible_password=redhat
[greg@bastion ansible]$ cp /etc/ansible/ansible.cfg ./
[greg@bastion ansible]$ mkdir roles
[greg@bastion ansible]$ vim /home/greg/ansible/ansible.cfg
#Modify only four lines in total
inventory = /home/greg/ansible/inventory
```

```
roles_path = /home/greg/ansible/roles
host_key_checking=False
remote_user = root
[greg@bastion ansible]$ ansible --version
[greg@bastion ansible]$ ansible-inventory --graph
```

### 😀 Create and run Ansible ad hoc commands

As a system administrator, you need to install software on managed nodes.

Follow the main text to create a shell script called /home/greg/ansible/adhoc.sh that will install the yum repository on each managed node using Ansible ad hoc commands:

Repository 1:

The name of the repository is EX294_BASE

Described as EX294 base software

The base URL is http://content/rhel8.0/x86_64/dvd/BaseOS

GPG signature checking is enabled

The GPG key URL is http://content/rhel8.0/x86_64/dvd/RPM-GPG-KEY-redhat-release

Repository is enabled

Repository 2:

The name of the repository is EX294_STREAM

Described as EX294 stream software

The base URL is http://content/rhel8.0/x86_64/dvd/AppStream

GPG signature checking is enabled

The GPG key URL is http://content/rhel8.0/x86_64/dvd/RPM-GPG-KEY-redhat-release

Repository is enabled

Problem solving method:

```
[greg@bastion ansible]$ ansible-doc yum_repository
[greg@bastion ansible]$ vim adhoc.sh
#!/bin/bash
ansible all -m yum_repository -a 'name="EX294_BASE" description="EX294 base software"
baseurl="http://content/rhel8.0/x86_64/dvd/BaseOS" gpgcheck=yes enabled=1 gpgkey="http:
//content/rhel8.0/x86_64/dvd/RPM-GPG-KEY-redhat-release"'
ansible all -m yum_repository -a 'name="EX294_STREAM" description="EX294 stream software"
baseurl="http://content/rhel8.0/x86_64/dvd/AppStream" gpgcheck=yes enabled=1
gpgkey="http: //content/rhel8.0/x86_64/dvd/RPM-GPG-KEY-redhat-release"'
[greg@bastion ansible]$ chmod +x adhoc.sh
```

### 😀 Install the package

Create a playbook called /home/greg/ansible/packages.yml :

Install the php and mariadb packages to hosts in the dev , test and prod host groups

Install the RPM Development Tools package group on hosts in the dev host group

Update all packages on hosts in the dev host group to the latest version

Problem solving method:

```
---
- name: install package A
  hosts: dev,test,prod
  tasks:
          - name: one
            yum:
                    name: php
                    state: latest
          - name: two
            yum:
                    name: mariadb
                    state: latest
- name: install package B
  hosts: dev
  tasks:
          - name: one
            yum:
                    name: "@RPM Development Tools"
                    state: latest
          - name: two
            yum:
                    name: "*"
                    state: latest
```

### 😀 Use RHEL system roles

Install the RHEL system role package and create a playbook /home/greg/ansible/timesync.yml with the following conditions :

run on all managed nodes

Use the timesync role

Configure the role to use the currently valid NTP provider

Configure the role to use the time server 172.25.254.254

Configure the role to enable the iburst parameter

Problem solving method:

```
[greg@bastion ansible]$ sudo yum install rhel-system-roles
[greg@bastion ansible]$ vim ansible.cfg
#Modify only one line
```

```
roles_path = /home/greg/ansible/roles:/usr/share/ansible/roles
[greg@bastion ansible]$ ansible-galaxy list
[greg@bastion ansible]$ cp /usr/share/doc/rhel-system-roles/timesync/example-timesync-
playbook.yml timesync.yml
[greg@bastion ansible]$ vim timesync.yml
---
- hosts: all
  vars:
    timesync_ntp_servers:
      - hostname: 172.25.254.254
        iburst: yes
  roles:
    - rhel-system-roles.timesync
```

### 😀 Install roles using Ansible Galaxy

Use Ansible Galaxy and the requirements file /home/greg/ansible/roles/requirements.yml . Download roles from the following URL and install to /home/greg/ansible/roles :

http://materials/haproxy.tar This role should be named balancer

http://materials/phpinfo.tar This role should be named phpinfo

Problem solving method:

```
[greg@bastion ansible]$ mkdir roles
[greg@bastion ansible]$ cd roles
[greg@bastion roles]$ vim requirements.yml
---
- src: http://materials/haproxy.tar
  name: balancer
- src: http://materials/phpinfo.tar
  name: phpinfo
[greg@bastion roles]$ ansible-galaxy install -r requirements.yml
[greg@bastion roles]$ ansible-galaxy list
```

### 😀 Create and use characters

Create a role named apache in /home/greg/ansible/roles with the following requirements :

httpd package installed, set to enable and start at system boot

Firewall is enabled and running with rules that allow access to the web server

The template file index.html.j2 already exists to create the file /var/www/html/index.html with the following output :

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.

Lao Liu's warm reminder: The role can be created. There is no need to perform a separate call test. Once it is called and used, it will cause the following question to report an error. The following question will immediately call the apache role, just follow the webpage content and steps~

Problem solving method:

```
[greg@bastion roles]$ ansible-galaxy init apache
[greg@bastion roles]$ vim apache/tasks/main.yml
---
- name: one
  yum:
          name: httpd
          state: latest
- name: two
  service:
          name: httpd
          state: started
          enabled: yes
- name: three
  firewalld:
          service: http
          permanent: yes
          state: enabled
          immediate: yes
- name: four
  template:
          src: index.html.j2
          dest: /var/www/html/index.html
[greg@bastion roles]$ vim apache/templates/index.html.j2
Welcome to {{ ansible_fqdn }} on {{ ansible_default_ipv4.address }}
```

### 😀 Using roles from Ansible Galaxy

Create a playbook named /home/greg/ansible/roles.yml with the following requirements :

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

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

Browsing to a host in the balancers host group (eg http://172.25.250.13 ) will generate the following output:

Welcom to serverb.lab.example.com on 172.25.250.11

Reloading the browser will generate output from another web server:

`Welcom to serverc.lab.example.com on 172.25.250.12`

The playbook contains a play that runs on a host in the webservers host group and will use the phpinfo role.

Browsing to a host in the webservers host group at the URL /hello.php will generate the following output:

Hello PHP World from FQDN

where FQDN is the fully qualified name of the host.

Hello PHP World from

serverb.lab.example.com

There are also various details of the PHP configuration, such as the installed PHP version, etc.

Likewise, browsing to http://172.25.250.12/hello.php produces the following output:

Hello PHP World from

serverc.lab.example.com

There are also various details of the PHP configuration, such as the installed PHP version, etc.

Problem solving method:

```
[greg@bastion ansible]$ vim roles.yml
---
- name: one
  hosts: balancers
  roles:
          - balancer
- name: two
  hosts: webservers
  roles:
          - phpinfo
- name: three
  hosts: webservers
  roles:
          - apache
```

### 😀 Create and use logical volumes

Create a playbook named /home/greg/ansible/lv.yml that will run on all managed nodes to perform the following tasks:

Create logical volumes that meet the following requirements:

Logical volumes are created in the research volume group

The logical volume name is data

Logical volume size is 1500 MiB

Format a logical volume with the ext4 filesystem

An error message should be displayed if the requested logical volume size cannot be created

Could not create logical volume of that size

, and a size of 800 MiB should be used instead .

If the volume group research does not exist, an error message should be displayed

Volume group done not exist

.

Do not mount logical volumes in any way

Problem solving method:

```
[greg@bastion ansible]$ vim lv.yml
---
- name: create and use logical volumes
  hosts: all
  tasks:
          - block:
                - name: one
                  lvol:
                          vg: research
                          lv: data
                          size: 1500
                - name: two
                  filesystem:
                          fstype: ext4
                          dev: /dev/research/data
            rescue:
                  - debug:
                          msg: Could not create logical volume of that size
                  - name: three
                    lvol:
                        vg: research
                        lv: data
                        size: 800
                    when: ansible_lvm.vgs.research is defined
                  - debug:
                          msg: Volume group done not exist
                    when: ansible_lvm.vgs.research is undefined
```

### 😀 Generate hosts file

Download an initial template file from http://materials/hosts.j2 to /home/greg/ansible

Complete the template so that it generates the following file: one line for each inventory host, in the same format as / etc/hosts

Create a playbook named /home/greg/ansible/hosts.yml that will use this template to generate the file /etc/myhosts on hosts in the dev host group .

After the playbook is running, the file /etc/myhosts on the hosts in the dev host group should contain one line for each managed host:

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

172.25.250.9 workstation.lab.example.com workstation

172.25.250.10 servera.lab.example.com servera

172.25.250.11 serverb.lab.example.com serverb

172.25.250.12 serverc.lab.example.com serverc

172.25.250.13 serverd.lab.example.com serverd

Note: The order in which the inventory host names are displayed is not important.

Lao Liu's warm reminder: At present, there is no absolute good or bad for problem-solving methods A and B. It is easier to use A, but a small number of students report that points will be deducted. of. The B method is the standard answer of Red Hat's original factory, which is more reliable in theory, but the disadvantage is that the memory is complicated. Students are requested to make choices based on their own preparation progress and their own circumstances.

Solution A:

```
[greg@bastion ansible]$ wget http://materials/hosts.j2
[greg@bastion ansible]$ vim hosts.j2
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
172.25.250.9 workstation.lab.example.com workstation
172.25.250.10 servera.lab.example.com servera
172.25.250.11 serverb.lab.example.com serverb
172.25.250.12 serverc.lab.example.com serverc
172.25.250.13 serverd.lab.example.com serverd
[greg@bastion ansible]$ vim hosts.yml
---
- name: generate hosts file
  hosts: dev
  tasks:
          - name: one
            template:
                    src: /home/greg/ansible/hosts.j2
                    dest: /etc/myhosts
```

Problem solving method:

```
[greg@bastion ansible]$ wget http://materials/hosts.j2
[greg@bastion ansible]$ vim hosts.j2
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain
{% for host in groups['all'] %}
{{ hostvars[host]['ansible_facts']['default_ipv4']['address'] }} {{
hostvars[host]['ansible_facts']['fqdn'] }} {{ hostvars[host]['ansible_facts
']['hostname'] }}
{% end for %}
[greg@bastion ansible]$ vim hosts.yml
---
- name: generate hosts file
  hosts: dev
  tasks:
          - name: one
            template:
                    src: /home/greg/ansible/hosts.j2
                    dest: /etc/myhosts
```

### 😀 Modify file content

Create a playbook named /home/greg/ansible/issue.yml as described below :

The playbook will run on all inventory hosts

The playbook will replace the contents of /etc/issue with a line of text shown below:

On hosts in the dev host group, this line of text reads: Development

On hosts in the test host group, this line of text reads: Test

On hosts in the prod host group, this line of text reads: Production

Problem solving method:

```
[greg@bastion ansible]$ vim issue.yml
---
- name: modify file content
  hosts: all
  tasks:
          - name: one
            copy:
                    content: 'Development'
                    dest: /etc/issue
            when: "inventory_hostname in groups.dev"
          - name: two
            copy:
                    content: 'Test'
                    dest: /etc/issue
            when: "inventory_hostname in groups.test"
          - name: three
            copy:
                    content: 'Production'
                    dest: /etc/issue
            when: "inventory_hostname in groups.prod"
```

### 😀 Create Web Content Directory

Create a playbook named /home/greg/ansible/webcontent.yml as described below :

The playbook runs on managed nodes in the dev host group

Create a directory /webdev that meets the following requirements :

The owner is the webdev group

With regular permissions: owner=read+write+execute , group=read+write+execute , other=read+execute

Has special permissions : set group ID

symlink /var/www/html/webdev to /webdev

Create the file /webdev/index.html with a one-line file like this: Development

Browsing this directory on a host in the dev host group (eg http://172.25.250.9/webdev/ ) will produce the following output:

Development

Problem solving method:

```
[greg@bastion ansible]$ vim webcontent.yml
```

```
---
- name: create web content directory
  hosts: dev
  tasks:
          - name: one
            file:
                    path: /webdev
                    state: directory
                    group: webdev
                    mode: '2775'
          - name: two
            file:
                    src: /webdev
                    dest: /var/www/html/webdev
                    state: link
          - name: three
            copy:
                    content: 'Development'
                    dest: /webdev/index.html
                    setype: httpd_sys_content_t
```

### 😀 Configure cron jobs ( new )

Create a playbook called /home/greg/ansible/cron.yml ,

Configure a `cron` job that runs `every 2 minutes` and executes the following commands:

`logger "EX294 in progress"`, running as user `natasha`

Problem solving method:

```
[greg@bastion ansible]$ vim cron.yml
---
- name: configure cron job
  hosts: all
  tasks:
          - name: one
            user:
                    name: natasha
                    state: present
          - name: two
            cron:
                    name: "logger info"
                    minute: "*/2"
                    user: natasha
                    job: logger "EX294 in progress"
```

### 😀 Generate hardware report

Create a playbook named /home/greg/ansible/hwreport.yml that will generate an output file /root/hwreport.txt on all managed nodes with the following information :

Inventory hostname

Total memory size in MB

BIOS version

size of disk device vda

size of disk device vdb

Each line in the output file contains a key=value pair.

Your playbook should:

Download the file from http://materials/hwreport.empty and save it as /root/hwreport.txt

Change to /root/hwreport.txt with the correct value

If the hardware item does not exist, the associated value should be set to NONE

Problem solving method:

```
[greg@bastion ansible]$ vim hwreport.yml
---
- name: generate hardware report
  hosts: all
  vars:
          hw_all:
                  - hw_name: HOST
                    hw_cont: "{{ inventory_hostname | default('NONE',true) }}"
                  - hw_name: MEMORY
                    hw_cont: "{{ ansible_memtotal_mb | default('NONE',true) }}"
                  - hw_name: BIOS
                    hw_cont: "{{ ansible_bios_version | default('NONE',true) }}"
                  - hw_name: DISK_SIZE_VDA
                    hw_cont: "{{ ansible_devices.vda.size | default('NONE',true) }}"
                  - hw_name: DISK_SIZE_VDB
                    hw_cont: "{{ ansible_devices.vdb.size | default('NONE',true) }}"
  tasks:
          - name: one
            get_url:
                    url: http://materials/hwreport.empty
                    dest: /root/hwreport.txt
          - name: two
            lineinfile:
                    path: /root/hwreport.txt
                    regexp: "^{{ item.hw_name }}"
                    line: "{{ item.hw_name }}={{ item.hw_cont }}"
            loop: "{{ hw_all }}"
```

### 😀 Create password vault

Create an Ansible library to store user passwords as described below:

The library name is /home/greg/ansible/locker.yml

The library contains two variables with the following names:

pw_developer , the value is Imadev

pw_manager , the value is Imamgr

The password used to encrypt and decrypt the library is whenyouwishuponastar

The password is stored in the file /home/greg/ansible/secret.txt

Problem solving method:

```
[greg@bastion ansible]$ vim ansible.cfg
#Modify only one line
vault_password_file = /home/greg/ansible/secret.txt
[greg@bastion ansible]$ vim locker.yml
---
pw_developer: Imadev
pw_manager: Imamgr
[greg@bastion ansible]$ echo whenyouwishuponastar > secret.txt
[greg@bastion ansible]$ ansible-vault encrypt locker.yml
```

### 😀 Create user account

Download the list of users to create from http://materials/user_list.yml and save it to /home/greg/ansible

The password library /home/greg/ansible/locker.yml created elsewhere is used for this exam . Create a playbook named /home/greg/ansible/users.yml to create user accounts as follows:

Users with the job description developer should:

Created on managed nodes in dev and test host groups

Assign password from pw_developer variable

is a member of the supplementary group devops

A user whose job description is manager should:

Created on a managed node in the prod host group

Assign password from pw_manager variable

is a member of the supplementary group opsmgr

The password is in SHA512 hash format.

Your playbook should function properly for this exam using the vault secret file /home/greg/ansible/secret.txt created elsewhere.

Problem solving method:

```
[greg@bastion ansible]$ wget http://materials/user_list.yml
[greg@bastion ansible]$ vim users.yml
---
- name: create user account
  hosts: all
  vars_files:
          - /home/greg/ansible/locker.yml
          - /home/greg/ansible/user_list.yml
```

```
  tasks:
          - name: one
            group:
                    name: devops
            loop: "{{ users }}"
            when: item.job == 'developer' and (inventory_hostname in groups.dev or
inventory_hostname in groups.test)
          - name: two
            user:
                    name: "{{ item.name }}"
                    password: "{{ pw_developer | password_hash('sha512','mysecretsalt')
}}"
                    groups: devops
            loop: "{{ users }}"
            when: item.job == 'developer' and (inventory_hostname in groups.dev or
inventory_hostname in groups.test)
          - name: three
            group:
                    name: opsmgr
            loop: "{{ users }}"
            when: item.job == 'manager' and inventory_hostname in groups.prod
          - name: four
            user:
                    name: "{{ item.name }}"
                    password: "{{ pw_manager | password_hash('sha512','mysecretsalt') }}"
                    groups: opsmgr
            loop: "{{ users }}"
            when: item.job == 'manager' and inventory_hostname in groups.prod
```

### 😀 Update keys for Ansible repositories

Update keys for existing Ansible repositories as described below:

Download the Ansible library from http://materials/salaries.yml to /home/greg/ansible

The current library password is insecure8sure

The new library password is bbs2you9527

The library remains encrypted with the new password

Problem solving method:

```
[greg@bastion ansible]$ wget http://materials/salaries.yml
[greg@bastion ansible]$ ansible-vault rekey --ask-vault-pass salaries.yml
Vault password: Paste current password
New Vault password: Paste new password
Confirm New Vault password: Paste new password
[greg@bastion ansible]$ ansible-vault view salaries.yml
```

### 😀 Create partitions (hidden questions, have a chance to appear)

On the balancers host, divide a new partition, the device is /dev/vdd, number 1, size 1500m, format it as ext4, mount it to the /newpart1 directory, if the space is not enough, divide it into 800m, if there is no vdd, an error is reported

Problem solving method:

```
---
- name: create partition
  hosts: balancers
  tasks:
          - block:
                  - name: one
                    file:
                            path: /newpart1
                            state: directory
                  - name: two
                    parted:
                            device: /dev/vdd
                            number: 1
                            state: present
                            part_end: 1500MiB
                  - name: three
                    filesystem:
                            fstype: ext4
                            dev: /dev/vdd1
                  - name: four
                    mount:
                            path: /newpart1
                            src: /dev/vdd1
                            fstype: ext4
                            state: mounted
            rescue:
                    - debug:
                            msg: Could not create partation of that size
                    - name: five
                      parted:
                              device: /dev/vdd
                              number: 1
                              state: present
                              part_end: 800MiB
                      when: ansible_facts.devices.vdd is defined
                    - name: six
                      filesystem:
                              fstype: ext4
                              dev: /dev/vdd1
                      when: ansible_facts.devices.vdd is defined
                    - name: seven
                      mount:
                              path: /newpart1
                              src: /dev/vdd1
                              fstype: ext4
                              state: mounted
                      when: ansible_facts.devices.vdd is defined
                    - debug:
                            msg: Disk does not exist
                      when: ansible_facts.devices.vdd is undefined
```

### 😀 Install RHEL character (hidden question, there is a chance to appear)

Install the RHEL role, and use the SELinux role, which is required to run on all nodes, with SELinux set to enforcing mode.

```
[greg@bastion ansible]$ dnf install rhel-system-roles -y
[greg@bastion ansible]$ cp -rf /usr/share/ansible/roles/linux-system-roles.selinux ./
[greg@bastion ansible]$ vim selinux.yml
#After installation, you can change it or not. If you change it, you can leave the rest,
and delete the excess.
---
```

```
- hosts: all
  vars:
          selinux_policy: targeted
          selinux_state: enforcing
  roles:
    - role: roles/rhel-system-roles.selinux
  tasks:
          - name: apply SElinux role
            block:
                    - include_roles:
                            name: roles/rhel-system-roles.selinux
            rescue:
                    - name: check
                      fail:
                      when: not selinux_reboot_required
                    - name: reboot
                      reboot:
                    - name: changed
                      include_role:
                        name: roles/rhel-system-roles.selinux
```