Ansible Exam (RHCE 8) full solution with explanation

* [*March 10, 2022*](https://techinfobest.com/2022/03/10/)

*[admin](https://techinfobest.com/author/admin/)*

Comments Offon Ansible Exam (RHCE 8) full solution with explanation

[Ansible](https://techinfobest.com/category/ansible/)[Linux](https://techinfobest.com/category/linux/)

The RHCE 8 exam is totally based on Ansible. Generally, 15-17 questions are asked to solve in the exam. In this article, I will try to solve all questions that you might face in the exam. Please ensure that you have already completed the RHCSA exam and have basic knowledge of YAML.

Question 1: Install and configure Ansible

User ismat 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/ismat/ansible/ansible.cfg and configure to access remote machines via the ismat user.

All roles should be in the path /home/ismat/ansible/roles.  
The inventory path should be in /home/ismat/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 node1 is a member of group dev, node2 is a member of group test, node3 is a member of group proxy, node4 and node5 are members of group prod, Also, prod is a member of group webservers.

Solution:

The environment will remain ready. You just need to install the ansible in the control node by the command. But at home for practice, you need to enable an ansible repository.

sudo yum install -y ansible

For writing code easily you can install vim package also. In a real exam, it helps you a lot as you have to write everything in a text file and vim editor will show you an error if you do a mistake in yml syntax.

sudo yum -y install vim

create a file /home/ismat/ansible/inventory and a directory  
/home/ismat/ansible/roles. Navigate to /home/ismat/ansible directory.

$ mkdir /home/ismat/ansible

$ touch /home/ismat/ansible/inventory

$ mkdir /home/ismat/ansible/roles

$ touch /home/ismat/ansible.cfg

$ cd /home/ismat/ansible

Now edit the inventory file ( vim inventory)

[dev]

node1.example.com

[test]

node2.example.com

[proxy]

node3.example.com

[prod]

node4.example.com

node5.example.com

[webservers:children]

prod

Edit the ansible.cfg file with the following information

[defaults]

inventory=/home/ismat/ansible/inventory

roles\_path= /etc/ansible/roles:/usr/share/ansible/roles:/home/ismat/ansible/roles

remote\_user= ismat

host\_key\_checking=false

[privilege\_escalation]

become=true

become\_user=root

become\_method=sudo

become\_ask\_pass=false

Verify your installation

ansible --version

The above command’s output will shows ansible version installed and also configuration file location : **config file = /home/ismat/ansible/ansible.cfg** . If it comes in this way then your installation is done perfectly. Please run a command always from /home/ismat/ansible directory.

**Tips**: You can copy the /etc/ansible.cfg file to /home/ismat/ansible/ansible.cfg and uncomment and edit the above configuration parameters .  
For practice at home install Ansible on CentOS 8 which is very straightforward. Click [Install Ansible on CentOS 8](http://techinfobest.com/install-ansible-on-centos-8/) to see details.

cp /etc/ansible.cfg /home/ismat/ansible/ansible.cfg

Question2: Repository setup

Create a file called adhoc.sh in /home/ismat/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’ , set gpgcheck and gpgkey is http://example.key and enable the repo

Solution:

vim adhoc.sh

#!/bin/bash

ansible all -m yum\_repository -a 'name=EPEL description=RHEL8 baseurl=https://dl.fedoraproject.org/pub/epel/epelrelease-

latest-8.noarch.rmp gpgcheck=yes gpgkey=http://example.key enabled=yes'

$ chmod 777 adhoc.sh

$ ./adhoc.sh

**Tips**: In the real exam if you forget syntax run the following command and you will get all options and can make sure your syntax is correct.

$ ansible-doc yum\_repository

Question 3: Install packages

Create a file called packages.yml in /home/ismat/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.

Solution:

---

- name: Install Common Packages

hosts: dev,prod,webservers

tasks:

- name: Install httpd,mod\_ssl,mariadb

yum:

name:

- httpd

- mod\_ssl

- mariadb

state: present

- name: Install development tools package on dev host

yum:

name: '@Development tools'

state: latest

when: "'dev' in group\_names"

- name: Update all the packages on dev hosts

yum:

name: '\*'

state: latest

when: "'dev' in group\_names"

...

**Tips:**To know how to write code for the yum module. Run command **ansible-doc yum** , scrolling to the bottom, and copy example code.

Question 4: Create Role including template

Create a role called sample-apache in /home/ismat/ansible/roles that enables and starts httpd, enables and starts the firewall; and allows the webserver service. Create a template called index.html.j2 which creates and serves a message from /var/www/html/index.html. Whenever the content of the file changes, restart the webserver service.  
**Welcome to [FQDN] on [IP]**  
Replace the FQDN with the fully qualified domain name and IP with the ip address of the node using ansible facts. Lastly, create a playbook in /home/ismat/ansible named apache.yml and use the role to serve the index file on webserver hosts.

Solution:

Navigate to /home/ismat/ansible/roles directory and run command

$ ansible-galaxy init sample-apache

vim sample-apache/tasks/main.yml

---

# tasks file for sample-apache

- name: Enable httpd

service:

name: httpd

state: started

enabled: yes

- name: Enable firewalld

service:

name: firewalld

state: started

enabled: yes

- name: Allow webserver service

firewalld:

service: http

state: enabled

permanent: yes

immediate: yes

- name: Create index file from index.html.j2

template:

src: index.html.j2

dest: /var/www/html/index.html

notify:

- restart\_webservers

vim sample-apache/templates/index.html.j2

Welcome to {{ ansible\_fqdn }} on {{ ansible\_default\_ipv4.address }}

**Tips:**There is a chance you can’t remember ansible\_fqdn or ansible\_default\_ipv4.address these kinds of facts. You can run adhoc command of the setup module to get the whole remote machine facts and save them to a file. **ansible node2.example.com -m setup > facts.json.** Then vim facts.json file and find the facts variables. It is extremely helpful if you are afraid of a spelling mistake.

vim sample-apache/handlers/main.yml

---

# handlers file for sample-apache

- name: restart\_webservers

service:

name: httpd

state: restarted

Navigate back to your ansible working directory /home/ismat/ansible and write in apache.yml ( vim apache.yml)

---

- name: Install apachec from apache-role

hosts: webservers

roles:

- name: sample-apache

...

Question 5: Install Role from ansible-galaxy repository

Create a file called requirements.yml in /home/ismat/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/ismat/ansible/roles.

Solution:

Create a file requirements.yml in /home/ismat/ansible/roles and write the following code

- name: haproxy-role

src: geerlingguy.haproxy

- name: php-role

src: geerlingguy.php

Navigate to the roles directory and run the following command

$ ansible-galaxy install -r requirements.yml -p /home/sandy/ansible/roles/

Check the roles were installed

$ ls \*

haproxy-role php-role sample-apache

Question 6: Call Role from playbook

Use the roles from Task 5 in a file called role.yml in /home/ismat/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.

Solution:

vim /home/ismat/ansible/role.yml

---

- name: install haproxy and php roles

hosts: all

vars:

haproxy\_backend\_servers:

- name: web1

address: node4.example.com:80

- name: web2

address: node5.example.com:80

tasks:

- name: import hasproxy

include\_role:

name: haproxy-role

when: "'proxy' in group\_names"

- name: import php

include\_role:

name: php-role

when: "'prod' in group\_names"

...

Question 7: Create secret file

Create an ansible vault password file called lock.yml with the password reallysafepw in the /home/ismat/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.

Solution:

Run the following command

$ ansible-vault create lock.yml

New Vault password:

Confirm New Vault password:

$ ansible-vault view lock.yml

Vault password:

pw\_dev: dev

pw\_mgr: mgr

Create a file /home/ismat/ansible/secret.txt and write reallysafepw

$ vim /home/ismat/ansible/secret.txt

$ cat /home/ismat/ansible/secret.txt

reallysafepw

Question 8: Read secret from vault

Create the users in the file users\_list.yml file provided. Do this in a playbook called users.yml located at /home/ismat/ansible. The passwords for these users should be set using the lock.yml file from TASK7. When running the playbook, the lock.yml file should be unlocked with secret.txt file from Question 7.

All users with the job of ‘developer’ should be created on the dev hosts, add them to the group devops, their password should be set using the pw\_dev variable. Likewise create users with the job of ‘manager’ on the proxy host and add the users to the group ‘managers’, their password should be set using the pw\_mgr variable.

**users\_list.yml**

users:

- username: bill

job: developer

- username: chris

job: manager

- username: dave

job: test

- username: ethan

job: developer

Solution:

Write your code in /home/ismat/ansible/users.yml file

---

- name: user create

hosts: dev,proxy

vars\_files:

- lock.yml

- users\_list.yml

tasks:

- name: developer user create

user:

name: "{{ item.username }}"

group: devops

state: present

password: "{{ pw\_dev | password\_hash('sha512') }}"

when: item.job=="developer" and "dev" in group\_names

loop: "{{ users }}"

- name: manager user create

user:

name: "{{ item.username }}"

group: managers

state: present

password: "{{ pw\_mgr | password\_hash('sha512') }}"

when: item.job=="manager" and "proxy" in group\_names

loop: "{{ users }}"

...

Run the playbook as follows

$ ansible-playbook users.yml --vault-password-file secret.txt

Question 9: Download and modify file

Create a file in /home/ismat/ansible/ called report.yml. Using this playbook, get a file called report.txt. Download the file from[http://classroom.example.com](https://classroom.example.com/) to all remote hosts at /root/report.txt. Then edit the lines in the file to provide the real information of the hosts. If a disk does not exist then write NONE. The file content of report.txt is

HOST= inventory hostname  
MEMORY=total memory in mb  
BIOS=bios version  
VDA\_DISK\_SIZE=disk size  
VDB\_DISK\_SIZE=disk size

Solution:

---

- name: copy file with custom information

hosts: all

tasks:

- name: get file

get\_url:

url: https://classroom.example.com/report.txt

dest: /root/report.txt

- name: chahge hostname

lineinfile:

path: /root/report.txt

line: HOST={{ ansible\_hostname }}

regex: ^HOST

state: present

- name: chahge memory

lineinfile:

path: /root/report.txt

line: MEMORY={{ ansible\_memtotal\_mb }}

regex: ^MEMORY

state: present

- name: chahge bios version

lineinfile:

path: /root/report.txt

line: BIOS={{ ansible\_bios\_version }}

regex: ^BIOS

state: present

- name: chahge vda

lineinfile:

path: /root/report.txt

line: VDA\_DISK\_SIZE={% if ansible\_devices.vda is defined %}{{ ansible\_devices.vda.size }} {% else %} NONE {% endif %}

regex: ^VDA\_DISK\_SIZE

state: present

- name: chahge vdb

lineinfile:

path: /root/report.txt

line: VDB\_DISK\_SIZE={% if ansible\_devices.sda is defined %}{{ ansible\_devices.sda.size }} {% else %} NONE {% endif %}

regex: ^VDB\_DISK\_SIZE

state: present

...

**Tips:** At real exam you will not see the content of report.txt file. So first download it to your working directory by the command **curl  
http://dl.example.com/report.txt –output report.txt .**After the task is completed you can delete the file from your working directory

Question 10: Create file using jinja2 template

Download the jinja template from http://dl.example.com/hosts.j2 in /home/ismat/ansible/ and Edit this file so it looks like the one below. The order of the nodes doesn’t matter. Then create a playbook in /home/ismat/ansible called hosts.yml and install the template on dev node at /root/myhosts

Download the jinja template from http://dl.example.com/hosts.j2 in /home/ismat/ansible/ and edit it. Then create a playbook in /home/ismat/ansible called hosts.yml and install the template on dev node at /root/myhosts so that /root/myhosts content will be like below. The order of the nodes doesn’t matter.

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

Solution:

Download the template

$ curl http://dl.example.com/hosts.j2 --output hosts.j2

Edit the hosts.j2 file as follows

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\_default\_ipv4.address }} {{ hostvars[host].ansible\_fqdn }} {{ hostvars[host].ansible\_hostname }}

{% endfor %}

Write a playbook named hosts.yml

---

- name: use hosts.j2 template

hosts: all

tasks:

- name: template a file to dev hosts

template:

src: /home/ismat/ansible/hosts.j2

dest: /root/myhosts

when: "'dev' in group\_names"

...

Question 11: Create LVM

In /home/ismat/ansible/ create a playbook called logvol.yml. In the play create a logical volume called lv0 and make it of size 12GiB on volume group vg0 If there is not enough space in the volume group print a message “Not enough space for logical volume” and then make a 6GiB Iv0 instead. If the volume group still doesn’t exist, create a message “Volume group doesn’t exist” Create an xfs filesystem on all lv0 logical volumes. Don’t mount the logical volume.

Write logvol.yml playbook

---

- name: create volume

hosts: all

tasks:

- name: create partition

parted:

device: /dev/sdb

number: 1

flags: [lvm]

state: present

- name: create vg

lvg:

vg: vg0

pvs: /dev/sdb1

when: ansible\_devices.sdb.partitions.sdb1 is defined

- name: create lv

lvol:

vg: vg0

lv: lv0

size: 12000m

when: ansible\_lvm.vgs.vg0 is defined and ((ansible\_lvm.vgs.vg0.size\_g | float) > 12)

- name: send message if volume group is 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) < 12)

- name: create similiar volume

lvol:

vg: vg0

lv: lv0

size: 6000m

when: ansible\_lvm.vgs.vg0 is defined and ((ansible\_lvm.vgs.vg0.size\_g | float) < 12)

- name: create fs

filesystem:

dev: /dev/vg0/lv0

fstype: xfs

when: ansible\_lvm.vgs.vg0 is defined

...

Question 12: Read content from custom web directory

Create a playbook called webdev.yml in /home/ismat/ansible. The playbook will create a directory /webdev on dev host. The permission of the directory is u=rwx,g=rw, other has no permission. Set group id for the folder and owner is webdev. Create a symbolic link from /webdev to /var/www/html/webdev. Serve a file from /webdev/index.html which displays the text “Development”. Curl http://node1.example.com/webdev/index.html to test

Write the playbook webdev.yml

---

- name: web development

hosts: dev

tasks:

- name: create webdev user

user:

name: webdev

state: present

- name: create directory

file:

path: /webdev

state: directory

owner: webdev

mode: u=rwx,g=rw,0=---,g+s

- name: create symbolic link

file:

src: /webdev

path: /var/www/html/webdev

state: link

- name: create index.html

copy:

content: Development

dest: /var/www/html/webdev/index.html

- name: install selinux policy

yum:

name: python3-policycoreutils

state: present

- name: allow httpd from custom directory

sefcontext:

target: '/webdev(/.\*)?'

setype: httpd\_sys\_content\_t

state: present

- name: restore the context

shell: restorecon -vR /webdev

...

Question 13: Set up NTP using Red Hat system roles

Create a playbook called timesync.yml in /home/ismat/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.

Solution:

Install Redhat System Roles

$ sudo yum install rhel-system-roles

Check timesync role is installed properly

$ ls -d /usr/share/ansible/roles/\*.timesync

/usr/share/ansible/roles/rhel-system-roles.timesync

Write a playbook /home/ismat/ansible/timesync.yml

---

- name: use rhel system role to timesync

hosts: all

roles:

- rhel-system-roles.timesync

vars:

timesync\_ntp\_servers:

- hostname: 0.uk.pool.ntp.org

iburst: yes

...

Question 14: Write content filtered by host

Create a playbook called issue.yml in /home/ismat/ansible which changes the file /etc/issue on all managed nodes: If the host is a member of dev then write “Development” If the host is a member of test then write “Test” If the host is a member of prod then write “Production”

Solution:

Write a playbook /home/ismat/ansible/issue.yml

---

- name: change issue file

hosts: all

tasks:

- name: change dev hosts issue file

copy:

content: "Development"

dest: /etc/issue

when: '"dev" in group\_names'

- name: change test hosts issue file

copy:

content: "Test"

dest: /etc/issue

when: '"test" in group\_names'

- name: change prod hosts issue file

copy:

content: "Producation"

dest: /etc/issue

when: '"prod" in group\_names'

...

Question 15: Schedule cron job

Create a playbook called regulartasks.yml which has the system that appends the date to /root/datefile every day at noon. Name is job ‘datejob’

Solution:

Write a playbook /home/sandy/ismat/regulartasks.yml

---

- name: cron

hosts: all

tasks:

- name: ensure a job that runs at noon

cron:

name: "datejob"

minute: "0"

hour: "12"

job: "date >> /root/datefile"

...

Question 16: Create and change password of vault file

Create an empty encrypted file called myvault.yml in /home/ismat/ansible and set the password to notsafepw. Rekey the password to iaiai202112.

Solution:

$ ansible-vault create myvault.yml

New Vault password:

Confirm New Vault password:

Check you set up the right password

$ ansible-vault view myvault.yml

Vault password:

Rekey the password

$ ansible-vault rekey myvault.yml

Vault password:

New Vault password:

Confirm New Vault password:

Rekey successful

Check rekey is successful

$ ansible-vault view myvault.yml

Vault password:

**Ansible Sample Exam Questions and Answers For RHCE EX294**

[](https://www.seimaxim.com/author/ali)

*by*[**K.M. Ali Qamar**](https://www.seimaxim.com/author/ali)

SEPTEMBER 15, 2022

**Introduction**

This article has practice questions for ansible, which can be very useful to prepare for your Red Hat Certified Engineer exam.

**LAB**

Your Lab should consist of one controller node and four managed nodes. If you think your laptop does not have enough MEMORY or storage to run 6 VMs, you can order VPS from [SeiMaxim](https://www.seimaxim.com/vps-hosting). At least one node should have a separate disk attached to the nodes so you can practice partitions, LVMs, etc.

**This article is not designed as a sample exam. I‘ve tried to create the sample questions in blocks of similar topics. Before attempting, you need to configure your systems to create an environment for your question. So please utilize snapshots to revert to your original state in a short period.**

**Questions about Ansible Installation and configuration**

**Question:1**

Creating inventory files is the most critical aspect of an ansible environment, and this exam question is imminent.

* Install ansible in controller node.
* Group creation for your inventory file should be as shown below:

node1 should be the member of proxy group

node2 and node3 should be the member of webservers group.

node4 should be the member of database group.

Create a new group called “prod” and put your webservers group in that new group.

**Answer:1**

To install Ansible, you can follow this link seimaxim.  
You can create an inventory file like shown below:

[proxy]

node1

[webservers]

node2

node3

[database]

node4

[prod:children]

webservers

**Questions about ad-hoc commands**

**Question:1**

* + Create repository file for MariaDB server using the ad-hoc command with the following commands:
  + name: MariaDB
  + description: Repository MariaDB-Server
  + base URL: http://yum.mariadb.org/10.5/centos8-amd64
  + GPGKEY: https://yum.mariadb.org/RPM-GPG-KEY-MariaDB
  + GPGCHECK: yes
  + enabled: active

\*You can even create a playbook instead of a shell script.

**Question:2**

This question prepares your environment so your ansible user can perform the tasks.

* Generate the script which creates the user named “ali” and copies the SSH public key of user “ansible”.
* User ali, which you have created, should be able to be a SUDO.

[Note] Your environment of 4 nodes and one controller already has root access to all the nodes. so manually, you are generating your ssh-key of user ali on the controller node, which you will copy on all the nodes.

**Question:3**

* + Set Red Hat Enterprise Linux ISO as a disk to the node:
  + whatever hypervisor you are using, attach the ISO image with the repo server, so mount /dev/sr0 to /mnt and have all the files from the ISO into the /mnt.
  + make a directory named /repo and copy everything to /mnt/repo.
  + In the last part, you can generate repo files with the following information:
  + The first repo file should contain the BaseOS, and the second repo file should contain the AppStream.
  + baseuls of both files are file:///repo/BaseOS & file:///repo/AppStream, respectively.
  + gpgcheck must of 0.

**Answers: Questions about ad-hoc commands**

**Answer: 1**

ansible database -m yum\_repository -a "name=mariadb description='Repository MariaDB-Server' baseurl='http://yum.mariadb.org/10.5/centos8-amd64' gpgkey='https://yum.mariadb.org/RPM-GPG-KEY-MariaDB' gpgcheck=yes enabled=yes" -b

You can create a file repo.sh and put the above-mentioned query into that file, make it executable, and run it.

**Answer2:** Create a file “vim adhoc.sh”.

#!/bin/bash

/usr/local/bin/ansible all -b -m user -a "name=ali state=present"

/usr/local/bin/ansible all -b -m file -a "path=/home/ali/.ssh state=directory owner=ali"

/usr/local/bin/ansible all -b -m copy -a "src=/home/ali/.ssh/id\_rsa.pub dest=/home/ali/.ssh/authorized\_keys directory\_mode=yes"

/usr/local/bin/ansible all -b -m lineinfile -a "path=/etc/sudoers state=present line='ali ALL=(ALL) NOPASSWD: ALL'"

once you exit from adhoc.sh file, make it executable with ” chmod +x adhoc.sh”.

**Answer3:** You can try it by yourself.

**Questions about file content –  variables & facts**

**Question: 1**

* + Create a playbook, which can do as shown below:
  + Create a file on all the servers at this path /etc/motd.
  + The file contains “MY SERVER”.

**Question: 2**

* + Create a playbook, which can do as shown below:
  + Create a file on all the servers at this path /etc/motd.
  + The file contains  “MY SERVER: <Server Name> “.

you replace the server Name with the actual HOSTNAME of the server.

**Question: 3**

* + Create a playbook, which can do as shown below:
  + The existing file (/etc/motd) should be removed ( if any).
  + host group proxy should have the line “This Is PROXY Server”.
  + Host group webservers should have the line “These Are The WEBSERVERS”.
  + Host group database should have the line “This is “DATABASE server

**Question: 4**

* + In this question, you are going to practice the configuration of the SSH Server.
  + Create a playbook, which can do the following tasks:

Set banner to /etc/motd

X11Forwarding is disabled

MaxAuthTries is set to 3

**Question: 5**

* Create a playbook such as servers.yml, which should generate the network file as shown below:
* Servers.yml must use a jinja2 file called servers.j2, and when servers.yml is executed, it creates the file servers.txt on the servers group of webservers and database servers.  
  for example, your file should contain  information as shown below:

<FQDN> <HOSTNAME> <IPADDRESS>

node1.seimaxim.com node1 192.168.1.12

**Question: 6**

* Create a playbook, which generates a file named “details.txt” file on each server with the following information:  
  The file should be placed in the root directory: /root

HOSTNAME: <hostname>

MEMORY\_TOTAL: <Total actual memory should be presented here>

MEMORY\_FREE: <Free Memory available on the system>

IPV4: <IP address of the system>

FQDN: <fqdn>

SDA\_DISK\_SIZE: </dev/sda1 disk size>

SDC\_DISK\_SIZE: </dev/sdc> #if disk is not attach,it should put NONE

**Question: 7**

* Create a playbook that checks the file’s existence.  
  print the message “File Exists” if the file exists and print the message “File does not exist”if the file does not exist.

**Question: 8**

* Create a playbook that performs the shown below actions:
* Change the line “Listen 80” in the /etc/httpd/conf/httpd.conf to “Listen 8080”.
* Allow the port 8080 in firewall and restart the httpd services

**Question: 9**

* Create a playbook that performs the shown below actions:
* Change the default target to the multi-user.target with the file module by using links. don’t use shell command to set-default target.

**Question: 10**

* Create a playbook that performs the shown below actions:
* Run this playbook against the webservers group.
* A custom Ansible fact **server\_role=webservers** is created that can be retrieved from **ansible\_local.custom.sample\_exam** when using Ansible setup module.

**Answers : Questions about file content –  variables & facts**

**Answer: 1**

[ali@control ansible]$ vi motd.yml

---

- hosts: all

become: yes

tasks:

- name: Write content in the file

copy:

dest: /etc/motd

content: "MY SERVER"

**Answer: 2**

[ali@control ansible]$ vi motd.q2.yml

---

- hosts: webservers

become: yes

tasks:

- name: Write content in the file

copy:

dest: /etc/motd

content: "MY SERVER: {{ ansible\_facts['hostname']}}"

**Answer: 3**

[ali@control ansible]$ cat motd.q3.yml

---

- hosts: all

become: yes

tasks:

- name: Generate the file motd content

file:

path: /etc/mtd

state: absent

- name: Proxy Server file content

copy:

dest: /etc/motd

content: "This Is PROXY Server"

when: "'proxy' in group\_names"

- name: WEBSERVERS file content

copy:

dest: /etc/motd

content: "These Are The WEBSERVERS"

when: "'webservers' in group\_names"

- name: Database Server file content

copy:

dest: /etc/motd

content: "This is DATABASE Server&amp;amp;amp;amp;amp;amp;amp;quot"

when: "'database' in group\_names"

**Answer: 4**

---

- hosts: all

become: yes

tasks:

- name: Set Banner to /etc/motd

lineinfile:

path: /etc/ssh/sshd\_config

regex: "^Banner"

line: "Banner /etc/motd"

- name: Disable X11Forwarding

lineinfile:

path: /etc/ssh/sshd\_config

regex: "^X11Forwarding"

line: "X11Forwarding no"

- name: Set MaxAuthTries to 3

lineinfile:

path: /etc/ssh/sshd\_config

regex: "^MaxAuthTries"

line: "MaxAuthTries 3"

**Answer: 5**

cat servers.yml

---

- hosts: all

become: yes

tasks:

- name: Generating a file by the jinja2 file

template:

src: servers.j2

dest: /root/servers.txt

[ali@control ansible]$ vi servers.j2

{% for host in groups['webservers']%}

{{ hostvars[host]['ansible\_facts']['fqdn']}} {{ hostvars[host]['ansible\_facts']['hostname']}} {{ hostvars[host]['ansible\_facts']['default\_ipv4']['address']}}

{% endfor %}

To check if the files has been generated.

[ali@control ansible]$ ansible all -m shell -a "cat /root/servers\*" -b

node3 | CHANGED |

node2.seimaxim.com node2 10.0.2.15

node3.seimaxim.com node3 10.0.2.15

node4 | CHANGED |

node2.seimaxim.com node2 10.0.2.15

node3.seimaxim.com node3 10.0.2.15

node1 | CHANGED |

node2.seimaxim.com node2 10.0.2.15

node3.seimaxim.com node3 10.0.2.15

node2 | CHANGED |

node2.seimaxim.com node2 10.0.2.15

node3.seimaxim.com node3 10.0.2.15

**Answer: 6**

---

- hosts: all

become: yes

tasks:

- name: generate a report

blockinfile:

path: /root/report.txt

create: yes

block: |

MEMORY\_TOTAL: {{ ansible\_facts['memtotal\_mb' ]}}

MEMORY\_FREE: {{ ansible\_facts['memfree\_mb' ]}}

HOSTNAME: {{ ansible\_facts['hostname']}}

IPV4: {{ ansible\_facts['default\_ipv4']['address']}}

FQDN: {{ ansible\_facts['fqdn'] }}

SDA\_DISK\_SIZE: {{ ansible\_facts['devices']['sda']['partitions']['sda1']['size'] }}

SDB\_DISK\_SIZE: {{ ansible\_devices.sdc.size | default('NONE') }}

You can check it by this command

#To check the report's content run shown below command

[ali@control ansible]$ ansible all -m shell -a "cat /root/report.txt" -b

**Answer: 7**

---

- hosts: all

become: yes

tasks:

- name: Check if the file exists

stat:

path: /var/www/html/index.html # or you can put the path of your desired file

register: result

- debug:

msg: "THE FILE EXISTS"

when: result.stat.exists == true

- debug:

msg: "THE FILE DOES'T EXIST"

when: result.stat.exists == false

**Answer: 8**

# you must have installed httpd or you can put any other file to practice this senario

---

- hosts: webservers

become: yes

tasks:

- name: Modify the content of httpd.conf

replace:

path: /etc/httpd/conf/httpd.conf

regexp: 'Listen 80'

replace: 'Listen 8080'

notify: restart\_httpd

handlers:

- name: restart\_httpd # must be same as notify

service:

name: httpd

state: restarted

**Answer: 9**

---

- hosts: database

become: yes

tasks:

- name: Change the default target

file:

src: /usr/lib/systemd/system/multi-user.target

dest: /etc/systemd/system/default.target

state: link

**Answer: 10**

---

- hosts: webservers

become: yes

tasks:

- name: Ensure the directory exists

file:

path: "/etc/ansible/facts.d/"

state: directory

- name: Copy content to file in new directory

copy:

content: "[sample\_exam]nserver\_role=webserversn"

dest: "/etc/ansible/facts.d/custom.fact"

**Questions about packages installation**

**Question: 1**

* Create a yml file, which performs the below tasks on webservers(httpd).
* Install the latest firewall and httpd.
* Make sure that your httpd service is allowed through firewalls.
* Create a file /var/www/html/index.html and put the entry “Server name is: hostname” so the hostname in the index.html should be replaced with the actual hostname from the system.

For example, when we do curl http://node2 it should give me “Server name is: node2”.

**Question: 2**

* + Create a playbook that performs the shown below activities:
  + The playbook will create a directory /webdir with the permission of 2755, and the owner is also webdir user.
  + You should have already configured apache to run your page from /var/www/html/index.html.
  + Create a link of your index.html in the /webdir, which should be accessible with the http://node1/webdir/index.html.
  + Your /var/www/webdir/index.hml should contain “I AM FROM WEBDIR”.
* ANSWERS: QUESTIONS ABOUT PACKAGES INSTALLATION

**Answer: 1**

---

- hosts: webservers

become: yes

tasks:

- name: Install packages on webservers

yum:

name:

- httpd

- firewalld

state: latest

- name: Start firewalld and httpd on webservers

service:

name: "{{ item }}"

state: started

enabled: yes

loop:

- httpd

- firewalld

- name: Opening Port through FW

firewalld:

service: http

permanent: yes

immediate: yes

state: enabled

- name: Creation of index.html

copy:

dest: /var/www/html/index.html

content: "Server name is: {{ ansible\_facts['hostname'] }}"

**Answer: 2**

---

- hosts: node2

become: yes

tasks:

- name: Create the webdir user

user:

name: webdir

state: present

- name: Create a directory

file:

mode: '2755'

path: /webdir

state: directory

- name: Create Symbolic Link

file:

src: /webdir

dest: /var/www/html/webdir

state: link

- name: Create index.html

copy:

dest: /var/www/html/webdir/index.html

content: " I AM FROM WEBDIR"

**Questions about users & groups**

**Question: 1**

* Create a playbook that performs the shown below tasks:
* Create a group on all the servers named “admin”.
* Create a group on all the servers named “members”.

Here is the file user\_list.yml, In the exam, it will be provided.

---

users:

- username: eli

uid: 1201

password: abcdeXYZ

group: admin

- username: vincent

uid: 1202

password: cdefgXYZ

group: admin

- username: sandy

uid: 2201

password: edfegZBC

group: members

- username: patrick

uid: 2202

password: gjfezBCG

group: members

* Users whose user ID starts with 1 should be created on servers in the **webservers** host group. The user password, shell, and UID should be used from the **user\_list.yml** variable file.
* Users whose user ID starts with 2 should be created on servers in the **database** host group. The user password, shell, and UID should be used from the **user\_list.yml** variable.
* Users uid starting from 1\* should be the member of supplementary group **admin**.
* Users uid starting from 2\* should be the member of supplementary group **members**.
* The shell should be set to /bin/bash for all users.
* Account passwords should use the SHA512 hash format.
* The SSH Key of your main ansible user should be uploaded. Dont generate the new SSH key just use your primary ansible user key, which you are using to connect from the controller node.

**Answers: Questions about users & groups**

Answer:1

---

- hosts: all

become: yes

vars\_files:

- user\_list.yml

tasks:

- name: Create the group named admin

group:

name: admin

state: present

- name: Create the group named admin

group:

name: members

state: present

- name: Create users on webservers whose user ID starts with 1

user:

name: "{{ item.username }}"

shell: /bin/bash

groups: "{{ item.group }}"

append: yes

uid: "{{ item.uid }}"

password: "{{ item.password | password\_hash('sha512') }}"

with\_items: "{{ users }}"

when:

- "'webservers' in group\_names"

- "item.uid|string|first == '1'"

- name: Create users on database whose user ID starts with 2

user:

name: "{{ item.username }}"

shell: /bin/bash

groups: "{{ item.group }}"

append: yes

uid: "{{ item.uid }}"

password: "{{ item.password | password\_hash('sha512') }}"

with\_items: "{{ users }}"

when:

- "'database' in group\_names"

- "item.uid|string|first == '2'"

- name: Create SSH directory for users on webservers hosts

file:

path: "/home/{{ item.username }}/.ssh/"

state: directory

owner: "{{ item.username }}"

group: "{{ item.group}}"

mode: "0700"

with\_items: "{{ users }}"

when:

- "'webservers' in group\_names"

- "item.uid|string|first == '1'"

- name: Create SSH directory for users on database hosts

file:

path: "/home/{{ item.username }}/.ssh/"

state: directory

owner: "{{ item.username }}"

group: "{{ item.group }}"

mode: "0700"

with\_items: "{{ users }}"

when:

- "'database' in group\_names"

- "item.uid|string|first == '2'"

- name: Copy private SSH-key to users on the webserver hosts

copy:

src: /home/ali/.ssh/id\_rsa.pub

dest: "/home/{{ item.username }}/.ssh/authorized\_keys"

owner: "{{ item.username }}"

group: "{{ item.group }}"

mode: "0600"

with\_items: "{{ users }}"

when:

- "'webservers' in group\_names"

- "item.uid|string|first == '1'"

- name: Copy private SSH-key to users on the database hosts

copy:

src: /home/ali/.ssh/id\_rsa.pub

dest: "/home/{{ item.username }}/.ssh/authorized\_keys"

owner: "{{ item.username }}"

group: "{{ item.username }}"

mode: "0600"

with\_items: "{{ users }}"

when:

- "'database' in group\_names"

- "item.uid|string|first == '2'"

**Questions about roles**

**Question: 1**

* Use ansible Galaxy to download and Install the ansible-galaxy role named geerlingguy.nginx
* The requirement file should install this role.
* change the name to Nginx

**Question: 2**

Create a role called **apache** and store it in your ansible directory “roles”. This role should satisfy the requirements below:

* The **httpd**, **mod\_ssl** and **php** packages are installed. Apache service is running and enabled on boot.
* The firewall is configured to allow all incoming traffic on HTTP port TCP 80 and HTTPS port TCP 443.
* Apache service should be restarted every time the file /var/www/html/index.html is modified.
* A Jinja2 template file index.html.j2 is used to create the file /var/www/html/index.html with the following content:

The address of the server is: IPV4ADDRESS

IPV4ADDRESS is the IP address of the managed node.

Create a playbook /home/automation/plays/apache.yml that uses the role and runs on hosts in the **webservers** host group.

**Question: 3**

* Create a playbook of any name, and that playbook should do as shown below:
* The playbook runs over all the managed hosts and uses the time sync role ( Red Hat system role)
* This role should be able to change the time server to nl.pool.ntp.org .

**Answers: Questions about roles**

**Answer:1**

**Step 1:** Type "ansible-galaxy search nginx --author geerlingguy --platforms EL" to find the geeglinguy.nginx role

Step 2: "ansible-galaxy info geerlingguy.nginx" to check more information about this role.

Step 3: Create requirement.yml file as show below:

- src: geerlingguy.nginx

version: "2.7.0"

name: Nginx

Step 4: Run this commad "ansible-galaxy install -r requirement.yml" Step 5: Run "ansible-galaxy list" to make sure that new role has been installed sucessfully. [Note] Prior to this task you need to have ansible-galaxy in your system. Make sure you have added the correct path of roles directory in your ansible.cfg file with roles\_path =< Path of your ansible directory to have the roles >

**Answer: 2**

Step 1: Create the empty role by "ansible-galaxy init apache" run this command in your roles directory, for it is /home/ali/ansible/roles

Step 2: Create the handler file roles/apache/handlers/main.yml.

---

- name: restart\_apache

service:

name: httpd

state: restarted

enabled: yes

Step 3: Create the main task file roles/apache/tasks/main.yml

---

- name: Ensure the packages httpd, mod\_ssl and php are installed

yum:

name: "{{ item }}"

state: latest

loop:

- httpd

- mod\_ssl

- php

- name: Ensure that the service httpd is enabled

service:

name: httpd

state: started

enabled: yes

- name: Ensure the firewall ports 80 and 443 are open

firewalld:

service: "{{ item }}"

permanent: yes

immediate: yes

state: enabled

loop:

- http

- https

- name: Create index.html from template

template:

src: index.html.j2

dest: /var/www/html/index.html

notify: restart\_apache

...

Step 4: Create a template file index.html.j2.

The address of the server is: {{ ansible\_facts['default\_ipv4']['address']}}

Step 5: Create a general apache file to run the role.

---

- hosts: webservers

become: yes

roles:

- apache

**Answer: 3**

Step 1: you must have RHEL system roles installed and settled all the paths.

Step 2: Copy the roles into your ansible directory or change the path of roles\_path in ansible.cfg file.

Step 3: Copy /usr/share/doc/rhel-sytem-roles/timesync/example-timesync-playbook.yml to home directory.

Step 4: Edit the file has shome below:

---

- hosts: all

vars:

timesync\_ntp\_servers:

- hostname: nl.pool.ntp.org

ibrust: yes

roles:

- rhel-system-roles.timesync

Step 5: Run shown above yaml file to settle the ntp server.

**Questions about disk partitions and LVM**

**Question: 1-A**

* Create a playbook that should perform the following tasks:
* Create a primary partition on /dev/sdb disk with the size of 2GB ( or anything according to your availability: just keep a couple of GBs for the next question).
* Create a volume group named “RedHat”.
* Create an lv named “exam”.
* Formate the new lv as xfs.
* Create a new directory named /mydir on it.
* Mount /mydir so it can sustain the reboot.

**Question: 2-B**

* Create a playbook that performs the following tasks:
* Use the remaining VG “RedHat” space and create a new LV named “ex294”.
* Formate the new lv with ext4.
* Create a folder named “/exam” and mount it so it can sustain the reboot.

**Answers: Questions about disk partitions and LVM**

**Answer: Q1A**

---

- hosts: database

become: yes

tasks:

- name: Create The Partition

parted:

device: "/dev/sdb"

number: 1

state: present

part\_end: 1GiB

- name: Create VG

lvg:

vg: "RedHat"

pvs: "/dev/sdb1"

- name: Create LV

lvol:

lv: "exam"

vg: "RedHat"

size: 500m

- name: Formate

filesystem:

fstype: "xfs"

dev: "/dev/mapper/RedHat-exam"

- name: Mount the FS

mount:

path: "/mydir"

src: "/dev/mapper/RedHat-exam"

fstype: "xfs"

state: mounted

**Answer: Q1B**

---

- hosts: database

become: yes

tasks:

- name: Create New LV On existing VG

lvol:

lv: "ex294"

vg: "RedHat"

size: 100%FREE

- name: Format new LV

filesystem:

fstype: "ext4"

dev: "/dev/mapper/RedHat-ex294"

- name: Mount new volume

mount:

path: "/exam"

src: "/dev/mapper/RedHat-ex294"

fstype: "ext4"

state: mounted

**Questions about cron**

**Question: 1**

* Create a playbook that performs the following tasks:
* Should create a crontab file /etc/cron.d/uptime.
* The task must run by the user ( any user you wish in your server).
* Every 5th minute from 9 to 5 on weekdays.
* The task should run uptime and put an entry here: /home/ali/ansible/my\_uptime\_cron.txt.

**Question: 2**

* Create a playbook that performs the following tasks:
* Create a root crontab record that runs every hour.
* The cron job appends the file /var/log/time.log with the output from the **date** command.

**Question: 3**

* Create a playbook, which should perform shown below tasks:
* Runs a fstrim command every day at 8:30 AM and 4:30 PM.

**Question: 4**

* Create a playbook, which should perform shown below tasks:
* Remove the corn job which you created in Q3.

**Question: 5**

* Create a yaml file, which should perform shown below tasks:
* Run a backup of /etc/passwd, /etc/group & /etc/shadow

**Answers: Questions about cron**

**Answer:1**

---

- hosts: all

become: yes

tasks:

- name: Crontab file

cron:

name: uptime command

user: ali

minute: "\*/5"

hour: 9-5

days: 1-5

job: "uptime &amp;amp;amp;amp;amp;amp;amp;gt;&amp;amp;amp;amp;amp;amp;amp;gt; /home/ali/ansible/my\_uptime\_cron.txt"

cron\_file: my\_uptime\_cron.txt

state: present

**Answer: 2**

---

- hosts: all

become: yes

tasks:

- name: Create crontab-record on proxy hosts

cron:

name: "Append the output of 'date' to /var/log/time.log"

minute: "0"

job: "date &amp;amp;amp;amp;amp;amp;amp;gt;&amp;amp;amp;amp;amp;amp;amp;gt; /var/log/time.log"

**Answer: 3**

---

- hosts: all

become:

tasks:

- name: Run Fstrim

cron:

name: "Run FSTRIM"

minute: "30"

hour: "8,4"

job: "fstrim -a"

**Answer: 4**

---

- hosts: all

become:

tasks:

- name: Run Fstrim

cron:

name: "Run FSTRIM"

state: absent

**Answer: 5**

---

- hosts: all

become: yes

tasks:

- name: Backup user data

cron:

name: Backup Users

hour: 5

minute: 30

weekday: 1-5

user: root

job: 'tar -czf /root/user.tgz /etc/passwd /etc/shadow'

cron\_file: user\_backup