ANSIBLE TRAINING FOR RHCE-9

Salman Francis
TEKCO LLC

Our Setup:

Three RH 9 systems on Virtual Machines

Host file of the host

[salman@RH-Server ansible]$ cat /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain 4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain 6 192.168.0.68 RH-2

Make sure time and date are correct on all the systems.

Make sure the user of remote system is in sudo

Allow root to run any commands anywhere
root ALL=(ALL) ALL salman ALL=(ALL) NOPASSWD:ALL ***Note the above
can also be achieved with echo “salman ALL=(ALL) NOPASSWD:ALL” >
/etc/sudoers.d/salman

1)

Have same users in the systems (I have two redhat 9 systems) use the ssh

commands to make the password less auth.

[salman@RH-Server root]$ ssh-copy-id salman@ 192.168.0.68
/usr/bin/ssh-copy-id: ERROR: No identities found
[salman@RH-Server root]$ ssh-keygen Generating
public/private rsa key pair.
Enter file in which to save the key (/home/salman/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/salman/.ssh/id_rsa Your public key
has been saved in /home/salman/.ssh/id_rsa.pub The key fingerprint is:
SHA256:HocnvCPxjCIRaC3+FwmXVBLDnlkPWnbNuQqfbfCTYaY salman@RH-Server The key's
randomart image is:
+---[RSA 3072]----+
| .=o. o . |
| .. ..+= . + |
|.o.o.oB + . |
|o ..o=.o + = |
| .. o. S % o |

```
|  ..  .* E *     |
|  ....o * . .    |
|   ... . .       |
|                 |
+----[SHA256]-----+
```

[salman@RH-Server root]$ ssh-copy-id salman@ 192.168.0.68 /usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed:
"/home/salman/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys salman@192.168.0.68's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'salman@192.168.0.68'" and check to make sure that only the key(s) you wanted were added.

[salman@RH-Server root]$ ssh salman@ 192.168.0.68
Last login: Thu Jul 13 23:52:03 2023 from 192.168.0.247 [salman@RH-2 ~]$

2- Now Create a directory of ansible in home of salman user

/home/salman
[salman@RH-2 ~]$ mkdir ansible

3- Install Ansible in RH-Server machine

salman@RH-Server root]$ dnf install ansible-core
Not root, Subscription Management repositories not updated
No read/execute access in current directory, moving to /
Error: This command has to be run with superuser privileges (under the root user on most systems).
[salman@RH-Server root]$ sudo dnf install ansible [sudo] password for salman:
Updating Subscription Management repositories.
Last metadata expiration check: 0:05:54 ago on Mon 26 Feb 2024 11:27:34 PM CST.
Package ansible-core-2.13.9-1.el9ap.x86_64 is already installed.
Dependencies resolved. ==========
==========
==========

Package	Architecture	Version	Repository	Size

```
================================================
============================
Upgrading:
 ansible-core       x86_64       2.13.10-1.el9ap       ansible-automation-platform-
2.2- for-rhel-9-x86_64-rpms       1.9 M

Transaction Summary
================================================
================================================
============================
Upgrade  1 Package

Total download size: 1.9 M Is this
ok [y/N]: y
Downloading Packages:
ansible-core-2.13.10-1.el9ap.x86_64.rpm                                    1.0
MB/s | 1.9 MB     00:01
------------------------------------------------------------------------------------------------------------------------
Total                                                       1.0 MB/s | 1.9 MB     00:01
Running transaction check
Transaction check succeeded. Running
transaction test
Transaction test succeeded. Running
transaction
  Preparing       :
1/1
  Upgrading       : ansible-core-2.13.10-1.el9ap.x86_64                      1/2
  Cleanup         : ansible-core-2.13.9-1.el9ap.x86_64                       2/2
  Running scriptlet: ansible-core-2.13.9-1.el9ap.x86_64                      2/2
  Verifying       : ansible-core-2.13.10-1.el9ap.x86_64
1/2
  Verifying       : ansible-core-2.13.9-1.el9ap.x86_64                       2/2
Installed products updated.

Upgraded:
  ansible-core-2.13.10-1.el9ap.x86_64                                   Complete!

[ salman@RH-Server root]$ ansible --version ansible [core
2.13.10]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/salman/.ansible/plugins/modules',
'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3.9/site-packages/ansible   ansible collection
location =
```

/home/salman/.ansible/collections:/usr/share/ansible/collections executable location = /usr/bin/ansible
python version = 3.9.16 (main, Dec 8 2022, 00:00:00) [GCC 11.3.1 20221121 (Red Hat 11.3.1-4)] jinja
version = 3.0.3 libyaml =
True

As we can see that Ansible's config is in /etc/ansible/ directory we need to make a config file in home/salman/ansible directory

vim /home/salman/ansible.cfg

[defaults]
inventory = myinventory # Means that inventory file is present in the same directory as ansible.cfg remote_user = tekco host_key_checking = false

[privilege_escalation] become = True # Become root where ever and when ever required.
become_method = sudo become_user = root
become_ask_pass = False

Now let's create our host in the inventory.

Vim /home/salman/myinventory.yml RH-2

save and quit.

Done.

Now we can run the ad-hoc command

```
[ salman@RH-Server ansible]$ ansible all -i myinventory -m ping
RH-2 | SUCCESS => {
   "ansible_facts": {
      "discovered_interpreter_python": "/usr/bin/python3"
   },
   "changed": false,
   "ping": "pong"
}
[ salman@RH-Server ansible]$ ansible all  -m ping
RH-2 | SUCCESS => {
   "ansible_facts": {
```

```
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
[salman@RH-Server ansible]$
```

Working.

Now let's create a yaml file to install repository in our host machiine. First let's create

.vimrc file and add the following

syntax on set bg=dark autocmd FileType yaml setlocal ai et ts=2 sw=2
cuc cul

Note:
ai = auto indent , ts = tab spacing, sw = shift width, cuc = cursor colum , cul = cursor line , et = expands tabs
setlocal will setlocal command sets a localoption. Local options apply to the current window or buffer

Q-1 Create a File called mypackages.yml in "/home/tekco/ansible" to install few packages for the following hosts:

On dev install httpd, mod_ssl and mariadb.
On prod install httpd, mariadb, mod_ssl and Developmentools. On dev update all the host's packages to the latest

Solution:

```
- name: "Installing my Pacakges"   hosts:  dev, prod   become:
  True   tasks:
- name: "Install the packages on all of the hosts"       dnf:
  name:
- httpd
- mod_ssl          - mariadb-server        state: present
- name: "Make sure the services are started and enabled"
  service:
```

```
      name: "{{item}}"
state: started       enabled:
yes      loop:       - httpd
- mariadb
- name: " Install Devtools on Prod"      dnf:       name:
- '@Development tools'       state: present

    when: "'prod' in group_names"

- name: "Make sure all the packages on dev hosts are latest"
  dnf:       name: '*'
     state: latest
    when: "'dev' in group_names"
```

Q-2 Create a Role called webserver in "/home/tekco/ansible/roles" with the following requirements:

1- The httpd package should be installed, httpd service should be enabled on boot and its started.

2- The firewall is enabled and running allowing web access

3- Create a Jinja 2 Template index.html.j2 with the following output

"Welcome to Hostname, whose IP is IPADDRESS"

Solution:

Create a roles directory in ansible directory

```
# ansible-galaxy role init  roles/webserver -  Role
webserver was created successfully

**Note roles directory will be automaticall created.

[ tekco@server1 ansible]$ tree roles roles
└── webserver
    ├── defaults
    │   └── main.yml
    ├── files
```

```
   ├── handlers
   │   └── main.yml
   ├── meta
   │   └── main.yml
   ├── README.md     ├──
tasks
   │   └── main.yml
   ├── templates
   ├── tests
   │   ├── inventory
   │   └── test.yml
   └── vars
       └── main.yml

9  directories, 8 files
[tekco@server1 ansible]$
```

Now create variables in var directory of webserver role

vim vars/main.yml pkgs:
- httpd - firewalld

myrules:
- http
- https

In template directory of webserver role create jinja 2 template vim

templates/index.html.j2

Welcome To {{ansible_facts ['hostname']}} whose IP is {{ansible_facts ['default_ipv4'] ['address']}}

Now in task directory under webserver role vim tasks/main.yml

```
- name: "Install the packages"   dnf:
   name: "{{item}}"     state:
present
  loop: "{{ pkgs}}"

- name: "Copying Jinja 2 Template"
  template:                                                    src: index.html.j2
    dest: /var/www/html/index.html

- name: "Starting & Enabling Services"   service:
```

name: "{{item}}"
state: started enabled:
yes
loop: "{{ pkgs}}"

- name: "Enabling Firewall Rules" firewalld:
service: "{{item}}"
permanent: yes
immediate: yes state:
enabled loop:
"{{myrules}}" Now in roles
directory we will create
playbook.

Vim webserver-role.yml

- name: "Webserver Playbook" hosts: dev become:
True roles:
- webserver

Q-3 Install a RHEL sysem Role package & Create a playbook called timesync.yml in

"/home/tekco/ansible/" with the following Conditions: 1- Run it on all managed hosts

2- It should use timesync role

3- The role should use the time server 2 .rhel.pool.ntp.org

4 . Set the iburst parameter as enabled.

Solution:

[root@server1 ~]# su tekco
[tekco@server1 root]$ cd /home/tekco/ansible/
[tekco@server1 ansible]$

root@server1 ~]# yum -y install rhel-system-roles Updating Subscription
Management repositories.

```
Last metadata expiration check: 23:13:50 ago on Sun 10 Mar 2024 01:23:25 AM EST.
Dependencies resolved.
================================================
================================================
================================
 Package                 Architecture       Version                Repository           Size
================================================
================================================
================================
Installing:  rhel-system-roles           noarch          1.22.0-2.el9           rhel-9-for-x86_64appstream-
rpms           2.7 M

Transaction Summary
================================================
================================================
================================
Install  1 Package

Total download size: 2.7 M Installed size:
11 M
Downloading Packages:
rhel-system-roles-1.22.0-2.el9.noarch.rpm                                    3.0 MB/s | 2.7 MB     00:00
-------------------------------------------------------------------------------------------------------------------------------
Total                                                                        3.0 MB/s | 2.7 MB 00:00
Running transaction check
Transaction check succeeded. Running
transaction test
Transaction test succeeded.
Running transaction
  Running scriptlet: rhel-system-roles-1.22.0-2.el9.noarch                                 1/1
  Preparing       :
1/1
  Installing      : rhel-system-roles-1.22.0-2.el9.noarch                                  1/1
  Verifying       : rhel-system-roles-1.22.0-2.el9.noarch                                  1/1
Installed products updated. vim
```

timesync.ymal

```
- name: Time Sync with All hosts   hosts: Test   become: yes   vars:    timesync_ntp_servers:
```

```
- hostname: 2.rhel.pool.ntp.org
      pool: yes        iburst:
yes   roles:
- rhel-system-roles.timesync
```

Testing:

```
[ tekco@server1 ansible]$ ansible Test -i testinventory -a "chronyc -N authdata" testnode | CHANGED
| rc=0 >>
Name/IP address          Mode KeyID Type KLen Last Atmp  NAK Cook CLen
==========================================================
==========================
2 .rhel.pool.ntp.org        -   0   0   0   -   0   0   0   0
2 .rhel.pool.ntp.org        -   0   0   0   -   0   0   0   0
2 .rhel.pool.ntp.org        -   0   0   0   -   0   0   0   0
2 .rhel.pool.ntp.org        -   0   0   0   -   0   0   0   0
[tekco@server1 ansible]$
```

Q-4 Use SELinux Role to Create a playbook called selinux.yml in "/home/tekco/ansible/"

with the following Conditions:

1- **Run it on all managed hosts & set selinux mode to permissive on all.**

2- **Verify selinux with ad-hoc command**

3- **Create another copy fo selinux.yml with name selinuxdefault.yml and setup default mode of enforcing for all manages hosts in this yml file.**

4. **Use ansible-navigator to execurte selinuxdefault playbook**

5. **Verify the selinux mode on all the managed nodes**

Solution:

For this again we will use RHEL-System-Roles

All the documents and roles with example are present in "/usr/share/doc/rhelsystem-roles/ Direcotory.

```
[ root@server1 ~]# su tekco
[ tekco@server1 root]$ cd /home/tekco/ansible/
[ tekco@server1 ansible]$
```

```
[ tekco@server1 ansible]$ dnf -y install rhel-system-roles
```

Lets Check our current Selinux in all nodes

```
[ tekco@server1 ansible]$ ansible Test -i testinventory -a "getenforce" testnode |
CHANGED | rc=0 >> Enforcing
```

Let's Create a playbook selinux.yml

```
- name: "SELinux System Role to Change Default Values"   hosts: Test   become: yes   vars:
- selinux_state: permissive                                                          roles:
- rhel-system-roles.selinux
```

Run the playbook and then check the result

```
[ tekco@server1 ansible]$ ansible-playbook -i testinventory selinux.yml
```

```
[ tekco@server1 ansible]$ ansible Test -i testinventory -a "getenforce" testnode |
CHANGED | rc=0 >> Permissive
```

First task is completed

Now we will create selinuxdefault.yml file and use ansible-navigator to run it. vim

selinuxdefault.yml

```
- name: "SELinux System Role to Change Default Values"   hosts: Test   become: yes   vars:
- selinux_state: Enforcing                                                           roles:
- rhel-system-roles.selinux
```

```
[ tekco@server1 ansible]$ sudo dnf install ansible-navigator
```

**Note if you don't have ansible-navigator , you can install it as follows:

First install pip

[root@server1 ~]# yum -y install pip
Updating Subscription Management repositories.
Last metadata expiration check: 0:41:44 ago on Mon 11 Mar 2024 02:30:43 AM EDT.
Dependencies resolved.
==========
==========
==========
Package Architecture Version Repository Size
==========
==========
==========
Installing: python3-pip noarch 21.2.3-7.el9_3.1 rhel-9-for-x86_64appstream-rpms 2.0 M

Then use pip to install ansible navigator

[root@server1 ~]# pip install ansible-navigator
Collecting ansible-navigator
Downloading ansible_navigator-24.2.0-py3-none-any.whl (297 kB)
| 297 kB 1.1 MB/s

To run playbook with ansbile-navigator use the following:

Roles: Note:

By default roles for RHEL such as rhel-system-roles are downloaded in

“/usr/share/ansible/roles” directory
with Ansible galaxy the role if role path is not give will be downloaded in “/home/user/.ansible/roles directory

ERROR & SOLUTION:

LI

tekco@server1 ansible]$ ansible-navigator run -m stdout selinux.yml WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available WARN[0000] For using

systemd, you may need to log in using a user session WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root) WARN[0000] Falling back to --cgroupmanager=cgroupfs WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available WARN[0000] For using systemd, you may need to log in using a user session WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root) WARN[0000] Falling back to --cgroup-manager=cgroupfs

The warning messages you're seeing are related to the cgroupv2 manager and systemd. It suggests that there is no systemd user session available.

Here are a few steps you can take to address this:

1. **Enable Lingering:**

   - As suggested in the warning, you can enable lingering for the user. Lingering allows the user's session to persist even when they are not logged in. Run the following command:

#loginctl enable-linger tekco

The following commands also helped me to install ansible-navigator

$ echo $XDG_RUNTIME_DIR
/run/user/1000

$ cd /run/user/1000
$ rm -rf libpod crun containers

Q-No.5 Use Requirments file in "home/tekco/ansible/" directory with the name requirements.yml to download and install roles with the following conditions:

1)Role name should be squid – download url: https://galaxy.ansible.com/download/mafalb-squid-0.2.0.tar.gz

2) Install git role with the following parameters :

Role name should be git

Repository https://github.com/geerlingguy/ansible-role-git

Install git if not present

Solution:

Create a file requirements.yml in the home directory with the following code.

```
- name: "Installing Squid Roles"
  src: https://galaxy.ansible.com/download/mafalb-squid-0.2.0.tar.gz   name: squid

- name: "Installing git"                                                scm: git
  src: https://github.com/geerlingguy/ansible-role-git   name: git
```

Install git if not installed:

```
dnf -y install git
```

Q-No. 6 Create a file squid.yml in your home directory with the following conditions:

1) The playbook should run on dev group

2) Use squid role available in your roles

Solution:

Create a squid.yml in "/home/tekco/ansbile" directory with the following code:

```
- name: "Setup Squid"
hosts: dev   become: True
roles: -  squid
```

Testing with ad-hoc command

```
[ tekco@server1 ansible]$ ansible dev -a "systemctl status squid" node1 | CHANGED |
rc=0 >>
● squid.service - Squid caching proxy
    Loaded: loaded (/usr/lib/systemd/system/squid.service; enabled; preset: disabled)
    Active: active (running) since Thu 2024-03-14 02:47:46 EDT; 29s ago
      Docs: man:squid(8)
   Process: 22128 ExecStartPre=/usr/libexec/squid/cache_swap.sh (code=exited, status=0/SUCCESS)
```

Q-No. 7 Create a Logical Volume with lvm.yml on all the nodes with the following conditions:

1) Logical Volume name: data, which is a member of research vol group

2) Size 2000 M

3) Formatting should be with ext4 File-System

4) If Vol Group is not found then it should print the message
"VG Not Found"
5) If the VG cannot accommodate 2000M size then it should print
"LV cannot be created with the given size and then the LV should be created with 100M of size

**Note: Do not perform any Mounting
Solution:

We can use the following command to list the lvm

[tekco@server1 ansible]$ ansible localhost -m setup -a filter=ansible_lvm Now create a lvm.yml

file with the following content

```
 become: True   tasks:
- name: "LVM Testing"      lvol:
      lv: data        vg:
research        size: 2000M
register: lvm_result
ignore_errors: True
- name: "Display MSG if VG not found"

    debug:
      msg: "VG NOT FOUND"
    when: lvm_result.failed

- name: "Display Message if LV cannot be
  Created with the given size"      debug:
  msg: "LV Cannot be Created with the
  Following Size"      when: lvm_result.failed

- name: "If above task fails"      lvol:
      lv: data        vg:
research        size:
100M
    when: lvm_result.failed

- name: "Formatting with EXT4 FS"
  filesystem:        fstype: ext4
```

dev: /dev/sdb/research/data
when: lvm_result is defined & lvm_result|Success

Q-No. 8 Create a partition on all the nodes with the following conditions:

1) **Create a 500M Primary Partition, 1 on dev sdb.**

2) **Format it using ext4 FileSystem**

3) **Permanently mount the partition to /data dir in the prod group**

4) **Display a mesasge " Size not Enough" in case the partition creation fails with the given size and then Create a Partition with 200M size.**

5) **Display a message " The device doesnot exist" in case there is no sdb device present on the nodes.**

Solution:

We will create a play book called partition.yml with the following content.

```
 tasks:
- name: "Check if device /dev/sdb exists"      stat:
      path: /dev/sdb


- name: "If sdb does not exist, display the
  message"      debug:       msg: "The disk does
  not exist"     when: ansible_devices.sdb is not
  defined

- name: "Creating Partition of 500 M"      parted:
  device: /dev/sdb       number: 1       part_end:
  500MiB       state: present
    when: ansible_devices.sdb is defined

- name: "If there is not Enough Space"      debug:
  msg: "Size not Enough"
    when: ansible_devices.sdb.size < 500MiB

- name: "Creating 200M Partition Size"
```

```
    parted:       device:
/dev/sdb       number: 1
part_end: 200MiB
      state: present

- name: " Creating FileSystem EXT4"
  filesystem:       fstype: ext4       device:
  /dev/sdb1
    when: ansible_devices.sdb is defined

- name: "Mounting the device"      mount:
      path: /data       src: /dev/sdb1       fstype: ext4
state: mounted      when: inventory_hostname in
groups.prod
```

Q-No. 9 Create an Ansible Vault to store password with the following conditions:

1) **File name is tekco-vault.yml**

2) **Use vault to encrytp/decrypt the file with the password "Linux"**

3) **The File should contain a keys user_pass, dev_pass and their vaules are set to "redhat" and "tekco" respectively**

4) **Store the passwords in secrets.yml file**

Solution:

First we will create a file called secrets.yml in our working directory.

vim secrets.yml

Linux

: wq (save and quit)
Now we will create a file tekco-vault.yml as follows

[tekco@server1 ansible]$ ansible-vault create --vault-password-file secrets.yml tekco-vault.yml

Now add the keys as follows in the file

user_pass: redhat

dev_pass: tekco save and

exit

Now cat the content and it should be encrypted

[tekco@server1 ansible]$ cat tekco-vault.yml
$ANSIBLE_VAULT;1.1;AES256
34343939623031343762643961653135643638343539373536393438646465636
437653839626136
61333462333137393865616535336535626665313630383 70a613466643535323
835303431326334
31626535393436633933343337316435393363356631653162633838663565663 936663762393462

To view the file use the view command

[tekco@server1 ansible]$ ansible-vault view tekco-vault.yml Vault password:

To permanently decrypt a file use the followiing

[tekco@server1 ansible]$ ansible-vault decrypt tekco-vault.yml --vault-passwordfile=secrets.yml
Decryption successful
[tekco@server1 ansible]$ cat tekco-vault.yml user_pass:
redhat dev_pass: tekco

Helpful Commands: ansible-vault –help, ansible-vault create –help

Q-No. 10 Change the key value you created in secrets.yml to dragon, without changing the encryption state of the file.

Solution: issue the following

command:

[tekco@server1 ansible]$ cat tekco-vault.yml
$ANSIBLE_VAULT;1.1;AES256
33313932656238343062653636653661653365663265363533386232663832636
536613365346261
63633939616439303430393832306664646163636239313 00a323664316632656

230653931343964
6337623331303233336333343363653439626238353130643936386137623831 3 834333261313333

We can see it's encrypted.

[tekco@server1 ansible]$ ansible-vault view tekco-vault.yml Vault
password: user_pass: redhat dev_pass: tekco

[tekco@server1 ansible]$ sha256sum tekco-vault.yml
85b27c093e74a209273bf5ca38b327ffc52ea5570e218b8107a51a2e9339b397 tekco-vault.yml

[tekco@server1 ansible]$ ansible-vault rekey tekco-vault.yml --vault-passwordfile=secrets.yml
New Vault password:
Confirm New Vault password:
Rekey successful

[tekco@server1 ansible]$ cat tekco-vault.yml
$ANSIBLE_VAULT;1.1;AES256
6536376639373936346439646162616139656362373434373963313139653338 3
730356231363038
6435623734643433313739643533643634633531316233630a633634326536646
538316234626439
623137633839306164373434653338633138323161393162386462626463633 76 136303333356661
But hash will be different

[tekco@server1 ansible]$ sha256sum tekco-vault.yml
0f245991ce42af2ce8ca79a94e5fafcaf6e95d5acf9a217c6e61b9c866149df0 tekco-vault.yml

This means the rekey is in effect

Let's view the file with our new key value.

[tekco@server1 ansible]$ ansible-vault view tekco-vault.yml Vault
password: user_pass: redhat dev_pass: tekco

Success

Q-No. 11 – Create a Playbook setupusers.yml in "/home/tekco/ansible/ " with the following conditions:

****Note: A file userdata.yml is already provided at example.com, use wget to download it.**

1- **Create users whose uid starts with 2 on dev group – Passwords must be used from the vault file you created earlier**

2- **Create users whose uid starts with 3 on prod group – Passwords must be used from the vault file you created earlier**

3- **Users should be part of additional respective groups such as dev and prod**

4- **The shell should be set to "/bin/bash"**

5- **Password should use SHA512 hash format.**

Solution:

First download the file wget example.com/userdata.yml lets check

the content and save it in our vars directory cat userdata.yml

```
---
users:
-          username: aragon
uid: 2000
 title: developer
-          username: dragon
uid: 2001   title:
developer -  username:
trex   uid: 3001   title:
prod_manager -
username: raptor   uid:
3002
 title: prod_manager
…
```

Now let's create a file (you would need to download this in the exam)

[tekco@server1 ansible]$ mkdir vars

[tekco@server1 vars]$ vim userdata.yml [paste the content from above file] save and quit

Now lets write a playbook setupusers.yml

```
-- name: "User Setup"  hosts:
dev,prod  become: true
vars_files:
-        vars/userdata.yml    - tekco-vault.yml  tasks:
-        name: "Ensure group Developers exists"
group:          name: dev       state: present

-        name: "Creating users for Dev group"     user:
      name: "{{ item.username }}"       uid:
"{{ item.uid }}"       groups: dev       state:
present       shell: /bin/bash
      password: "{{ dev_pass | password_hash('sha256') }}"     when: item.title ==
"developer" and item.uid | string | first == '2'     loop: "{{ users }}"

-        name: "Ensure group Prod exists"     group:
name: prod
      state: present

-        name: "Creating users for Prod group"     user:
      name: "{{ item.username }}"       uid: "{{
item.uid }}"
      groups: prod                                                    state: present
shell: /bin/bash
      password: "{{ user_pass | password_hash('sha256') }}"
    when: item.title == "prod_manager" and item.uid | string | first == '3'     loop: "{{ users }}"
```

Q-No. 12 – Create a Playbook wb-roles.yml in "/home/tekco/ansible/ " with the following conditions:

1- **Run php role on the webservers host group**

2- **Run balancer role on the balancer host group**

3- **Balancer host group via web browser should give the round robin load blancing of webserver nodes which are (node 1 and node 2) with the message "Welcome to <respective node IP Addressess>**

Complete Solution:

This will show complete setup of haproxy with php working template

Let's download the roles (I am not using roles provided by redhat so the url would be different in the exam) ansible-galaxy role install shaneholloman.php

[tekco@server1 ansible]$ ansible-galaxy role install geerlingguy.php -p roles/
Starting galaxy role install process
- downloading role 'php', owned by shaneholloman
- downloading role from
- geerlingguy.php (5.0.1) was installed successfully

Now

ansible-galaxy role install geerlingguy.haproxy

[tekco@server1 ansible]$ ansible-galaxy role instal geerlingguy.haproxy -p roles/
Starting galaxy role install process
- downloading role 'haproxy', owned by geerlingguy.haproxy
- downloading role from
- geerlingguy.haproxy (1.0.5) was installed successfully
Make slight changes to proxy role vim

roles/geerlingguy.haproxy/default/main.yml

List of backend servers. haproxy_backend_servers:
- name: node1
  address: 192.168.0.230:80
- name: node2
  address: 192.168.0.231:80

Also in tempalte section edit haproxy.cfg.jinja2 template make sure to disable cookie e.g **should be like this:**

  cookie SERVERID insert indirect
{ % for backend in haproxy_backend_servers % } server {{ backend.name }} {{ backend.address }} check **instead of**

```
   cookie SERVERID insert indirect
{ % for backend in haproxy_backend_servers % }
   server {{ backend.name }} {{ backend.address }} cookie
{{  backend.name }}  check
```

Creating Jinja 2 Template for PHP to work.

[tekco@server1 ansible]$ ansible-galaxy role init roles/php 2

Now go to the roles/php2/templates directory and create index.php.j2 file with the following content:

```
< ?php
  <?php echo "Welcome to " . "{{ ansible_facts['hostname'] }}" . " whose IP is " .
"{{ ansible_facts['default_ipv4']['address'] }}";
?>
```

Then:

Go to roles/php2/tasks/ folder and vim to main.yml and type the following content

```
- name: "Copying Jinja 2 Template"   template:
   src: index.php.j2
   dest: /var/www/html/index.php
```

Now in ansible working directory create a playbook with the following content vim php-ha.yml

```
- name: "For PHP Template Only"   hosts: webservers
  become: True   roles:
- php 2
```

Let's write our playbook vim wb-

roles.yml

Now create a yml file to install the roles wb-roles.yml

```
- name: "Setting up the PhPinfo"   hosts: webservers
  become: True   roles:
- shaneholloman.php

- name: "Setting up the HAProxy Role"   hosts:
  balancers   become: True   roles:
- geerlingguy.haproxy
```

Run and it should be working

Q-No. 13 – Create a Playbook hw-report.yml in "/home/tekco/ansible/ " with the following conditions:

1- Should produce an output file in "/root/hwreport.txt" on all nodes with the below

information: a- Inventory Name (Hostname) b- Memory c- Bios Version d- Size of disk device

e- The file should consist of key: value pair f- If a hardware item is non existing , it should print

NONE Solution:

Some of the useful commands before we write the playbook

```
[ tekco@server1 ansible]$ ansible all -m setup | grep fqdn
      "ansible_fqdn": "node1",
      "ansible_fqdn": "node2",
      "ansible_fqdn": "testnode",
[ tekco@server1 ansible]$ ansible all -m setup | grep hostname
```

```
        "ansible_hostname": "node2",
        "ansible_hostname": "testnode",
        "ansible_hostname": "node1",
[ tekco@server1 ansible]$ ansible all -m setup | grep bios
        "ansible_bios_date": "12/01/2006",
        "ansible_bios_vendor": "innotek GmbH",
        "ansible_bios_version": "VirtualBox",
        "ansible_bios_date": "12/01/2006",
        "ansible_bios_vendor": "innotek GmbH",
        "ansible_bios_version": "VirtualBox",
        "ansible_bios_date": "12/01/2006",
        "ansible_bios_vendor": "innotek GmbH",
        "ansible_bios_version": "VirtualBox",
[ tekco@server1 ansible]$ ansible all -m setup | grep memory
        "ansible_memory_mb": {
        "ansible_memory_mb": {
        "ansible_memory_mb": {

[tekco@server1 ansible]$
```

Now we can write the playbook based on the above

But first we would need to create the Jinja 2 template with the above information

vim hwreport.j2

```
--  Hostname: {{ansible_facts['fqdn' ]}}

--  Total Memory: {{ansible_facts['memtotal_mb' ]}}

--  Bios Version: {{ansible_facts['bios_version' ]}}

--  Block Device: {{ansible_devices['sda']['size'] | default ('None' )}}

--  Block Device: {{ansible_devices['sdb']['size'] | default('None' )}}
```

Now lets create hw-report.yml

```
-          name: "Get the HW Report"
hosts: all   become: True   tasks:
-          name: "Geenrating Report"
template:        src: hw-report.j2
dest: /root/hw-report.txt
```

[tekco@server1 ansible]$ ansible-playbook -i myinventory hw-report.yml

Q-No. 14 – Create a Playbook issue.yml in "/home/tekco/ansible/ " with the following conditions:

a- The playbook runs on all the hosts

b- The playbook will replace the content of "/etc/issue" file with single line of text e.g

- For the Member of Dev group, it would read Development

- For the Member of Prod group, it would read Production - For the Member of

Balancer group, it would read Balancer

Solution:

Let first check the content of our inventory file cat

myinventory

Now lets write our playbook issue.yml

```
-       name: "Replacing the Content of issue file"   hosts:
all   become: True   tasks:
-       name: "Change Content of Dev Group"      copy:
      content: "Developement"      dest:
/etc/issues
    when: "'dev' in group_names"

-       name: "Change Content of Prod  Group"      copy:
      content: "Production"       dest:
/etc/issues
    when: "'prod' in group_names"
-       name: "Change Content of Dev Group"      copy:
      content: "Balancers"       dest:
/etc/issues
    when: "'balancers' in group_names"
```

Testing:

[tekco@server1 ansible]$ ansible dev -a "cat /etc/issues" node1 | CHANGED
| rc=0 >>

Developement

Q-No. 15 – Create a Playbook web.yml in "/home/tekco/ansible/ " with the following conditions:

a- The playbook runs on dev Group

b- Create Direcotry named "web" with group owner "web" and user owner "apache"

c – Directory will have rwx permission for user and group and rx for others. Also setup the sgid "Set Group ID" on the directory

d - Create a file index.html in the "web" directory with text "Welcome to Tekco.net" e – Create a Symbolic link from "/var/www/html/web to /web

Solution:

```
- name: " Creating some webcontent of Prod"
hosts: prod become: True
tasks:
- name: "Creating Group & Dir" group:
    name: web2000

- name: "Creating Directory" file:
    state: directory path:
/web2000 mode: '2775'
group: web2000 owner:
apache
    setype: httpd_sys_content_t

- name: "Creating Symbolic Link" file:
    src: /web2000 dest:
/var/www/html/web2000
    state: link force: yes
mode: '2775'
    owner: apache

- name: "Creating index.html file in web2000
directory" copy:
    dest: /web2000/index.html mode: '0664'
```

owner: apache content:
"Welcome to Tekco.net" setype:
httpd_sys_content_t

Q-No. 16 – Create a Playbook cronjob.yml in "/home/tekco/ansible/ " with the following conditions:

a- Create a Cronjob for the user tekco on webservers group nodes.

b- Every minute the job will execute logger with the content "EX-294 Exam in Progress"

Solution:

- name: hosts:
webservers become:
True tasks:
- name: "Setting
UP Cron Job" cron:
name: "Cron Logger" user:
tekco
minute: "*/1" job: logger "EX-294
Exam is in Progress" state: present

[tekco@server1 ansible]$ ansible testnode -a 'crontab -lu tekco' testnode | CHANGED |
rc=0 >>
#Ansible: Cron Logger
*/1 * * * * logger "EX-294 Exam is in Progress"

Q-No. 17 – Create a Playbook hosts.yml in "/home/tekco/ansible/ " with the following conditions:

a- The Playbook will use hosts.j2 template and should run on all nodes.

b- Must gather information of all the hosts and copy it to the file "/etc/tekco-hosts of balancers group.

The format would be:

IP address HOSTNAME FQDN

****Note: hosts.j2 will be provided (Most probably we would need to download it with wget in our working directory)**

Solution:

A jinga2 template would be provided with the following content.

```
127.0.0.1  localhost {{ ansible_hostname }} {{ ansible_fqdn  }}
127.0.1.1 localhost
{ % for host in groups['all'] % }
{{  hostvars[host]['ansible_facts']['default_ipv4']['address'] }} {{ hostvars[host ]
[ 'ansible_hostname'] }} {{ hostvars[host]['ansible_fqdn']  }} { % endfor % }
```

Now Create a Playbook.

```
- name: "Copy hosts information to Testnode"   hosts:
  all   become: True   tasks:
- name: "Copy hosts information"      template:
      src: hosts.j2        dest:
/etc/myhosts
    when: "'balancers' in group_names"
```

Q-No. 18 – Create a Playbook repos.yml in "/home/tekco/ansible/ " with the following conditions:

- **The Playbook will configure repositories on all the managed hosts. Webservers Group**

Given Repositories:

1.

a) **Name: RH_Base**
b) **Description: RedHat Core Software**
c) **Url:**
d) **GPG signature check is enabled**
e) **GPG key URL:**
f) **enabled: yes**

2.

a) **Name: RH_AppStream**
b) **Description: RedHat App Stream Software**
c) **Url:**
d) **GPG signature check is enabled**
e) **GPG key URL:**

f) enabled: yes

Solution:

[rhel-9-for-x86_64-appstreamrpms] name = Red Hat Enterprise Linux 9 for x86_64 - AppStream (RPMs) baseurl = https://cdn.redhat.com/content/dist/rhel8/$releasever/x86_64/appstream/os enabled = 1 gpgcheck = 1

gpgkey = file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

Base:

[rhel-8-for-x86_64-baseosrpms]

name = Red Hat Enterprise Linux 9 for x86_64 - BaseOS (RPMs)
baseurl = https://cdn.redhat.com/content/dist/rhel9/$releasever/x86_64/baseos/os
enabled = 1 gpgcheck = 1

gpgkey = file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

[tekco@server1 ansible]$ ansible-doc -l | grep yum
ansible.builtin.yum Manages packages...
ansible.builtin.yum_repository Add or remove YU...
community.general.yum_versionlock Locks / unlocks ... [tekco@server1 ansible]$ ansible-doc yum_repository EXAMPLES:

- name: Add multiple repositories into the same file (1/ 2) ansible.builtin.yum_repository: name: epel
description: EPEL YUM repo file: external_repos
baseurl: https://download.fedoraproject.org/pub/epel/$releasever/$basearch/ gpgcheck: no

vim repo.yml

- name: " Setting Up the Repos in our nodes"
hosts: webservers become: True tasks:
- name: "Setting up Base Repo"
yum_repository: name: "RH_BASE"
description: "Redhat Core Software" baseurl: "

```
https://cdn.redhat.com/content/dist/rhel9/$releasever/x86_64/baseos/os"        enabled:
yes        gpgcheck: 1
       gpgkey: "file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release"

-          name: "Setting up APP STREAM Repo"
yum_repository:
       name: "RH_APP STREAM"
       description: "Redhat APP STREAM Software"        baseurl:
"https://cdn.redhat.com/content/dist/rhel8/$releasever/x86_64/appstream/os"        enabled:
yes        gpgcheck: 1
       gpgkey: " file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release"
```

Q-No. 19 – Create a Playbook timestamp.yml in "/home/tekco/ansible/ " with the following conditions:

a- The Playbook will run on webservers group.

b- Must gather the timestamp information of the file "/var/www/html/index.html" with the help of s script "timestamp.sh" already provided. c- Print the details as standard out on ansible contoller node and save the output to the file as well with their respective nodes.

Solution:

vim timestamp.yml

```
-          name: "Time Stamps of the index.html file"
hosts: webservers   become: True   tasks:
-          name: "Run a Script to Gather information of
index.html file Time stamp"       script:
/home/tekco/ansible/timestamp.sh
/var/www/html/index.html       register:
timestamp_output

-          name: "Print the output of the script"
debug:        var: timestamp_output.stdout_lines

-          name: "Copy the output to the file"
lineinfile:
       path: timestamp.txt
```

line: "{{inventory_hostname}}: {{timestamp_output.stdout_lines}}" delegate_to: localhost

Adhoc:

[tekco@server1 ansible]$ ansible webservers -m script -a "/home/tekco/ansible/timestamp.sh /var/www/html/index.html"

Q-No. 20 – Create a Playbook backup.yml in "/home/tekco/ansible/ " with the following conditions:

a- The Playbook will run on all group.

b- Create a tar file for "/var/www/html/" directory c- Copy the tar file to ansible controller node.

Solution:

```
- name: "Backing up the data from testnode"
  hosts: testnode
become: True   tasks:
- name: "Creating Tar file"      archive:
      path: "/var/www/html"       dest:
"/home/tekco/backup.tar.gz"       owner: tekco
group: tekco
      format: gz

- name: "Copying the tar archive to local host"      fetch:
      src: "/home/tekco/backup.tar.gz"
      dest: "/home/tekco/backup.{{inventory_hostname}}.tar.gz"       flat: yes
```