**DO374 Exam**

\* Read the Instruction.

1. you will get a total 10 vms.

control.lab17.example.com

node1.lab17.example.com

node2.lab17.example.com

node3.lab17.example.com

node4.lab17.example.com

node5.lab17.example.com

node6.lab17.example.com

controller.lab17.example.com

hub.lab17.example.com (private hub)

git.lab17.example.com(Git repos)

2. Total 17 questions,Time 4.00 hr.

3. ssh keybased configuration already done with all machines using 'matthew' user.

4. 'root' user password is 'mishings' .

5. all ques solved by matthew user in control mc

####################################################################

Q1. configure git in the control machine user name is 'matthew', user email id 'matthew.lab.example.com' push default method is 'simple'.

Ans -:

git config --global user.name 'matthew'

git config --global user.email 'matthew.lab.example.com'

git config --global push.default simple

git config -–global -l

Q2. create a project create user from http://git.lab.example.com:8081/git/create\_user.git .

playbook created their name create\_user.yml with some task.users will create from a file user\_list.yml where some user names are there with a group.

Add bob user to a group developer.add node3 to a group dev inside the inventory and run the playbook.

after done push to git remote repo.

Ans -:

git clone <git url>

cd /home/matthew/create\_user/

Podman login -u admin(given) -p mishings(given) hub.lab17.example.com

vim inventory

[dev]

Servera

Serverc(add)

[prod]

serverb

vim user\_list.yml

---

developer:

- stave

- matt

- vision

- bob

- sally

ansible-navigator run create\_user.yml -m stdout --eei ee-supported-rhel8:latest

Vim .gitignore

.ssh/

ansible-navigator.log

\*-arifact-\*

git status

o/p modified inventory

modified user\_list,yml

.gitignore

git add .

git commit -m 'invetory and user updated'

git push

Cd ..

Q3. install alias httpd install.clone the project name web\_install from http://git.lab.example.com:8081/git/alias\_httpd.git . there is a file alias.conf with some data . playbook was created with copy task.

if the task is executing httpd service should restart if not httpd service should not start.

After exicute playbook push to git server.

Ans -:

git clone

cd /home/matthew/web\_install

vim apache-setup.yml

* hosts: dev

become: yes

Gather\_facts: false

- name: Generate a basic homepage from jinja2 template

ansible.builtin.copy:

src: alias.conf

dest: /etc/httpd/conf/

notify: restart apache

handlers:

- name: restart apache

ansible.builtin.service:

name: httpd

state: restarted

Podman images

ansible-navigator run apache-setup.yml -m stdout --eei ee-supported-rhel8:latest --pp missing

cp ~/create\_user/.gitignore .

git add --all

git commit -m 'notify triggered for action'

git push

Q4. create a tag based playbook.

playbook name tags.yml pull from http://git.lab.example.com:8081/git/tag-project.git.

Add 'que, sara sara' to /var/www/html/index.html in 'dev' host group with tag 'alpha'.

Add 'what ever, will be will be' to /var/www/html/index.html in same 'dev' host group with tag 'beta'.

if any tag not given playbook should not execute any task.

After done everything push to git server.

Ans -:

cd /home/matthew

git clone <git url>

cd tag-project

vim tags.yml

---

- name: Ensure selected task executed

hosts: dev

become: yes

- name: first tag task

ansible.builtin.copy:

content: "que, sara sara"

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

tags: [ never, alpha ]

- name: second tag task

ansible.builtin.copy:

content: "what ever, will be will be"

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

tags: [ never, beta ]

ansible-navigator run tags.yml -t alpha/beta --eei ee-supported-rhel8:latest --pp missing -m stdout

ansible-navigator run tags.yml --eei ee-supported-rhel8:latest --pp missing -m stdout

git add .

git commit -m 'selected task executed'

git push

cd ..

Q5. Tune ansible configuration file.

pull git project from http://git.lab.example.com:8081/git/performance-tune.git

modify ansible.cfg file with parallelization size is 30 and gather facts no.

push to git server.

Ans -:

git clone <git repo>

cd performance-tune

vim ansible.cfg

[defaults]

inventory=inventory

remote\_user=matthew

forks=30

gathering=explicit

[privilege\_escalation]

become=True

become\_method=sudo

become\_user=root

become\_ask\_pass=False

git add ansible.cfg

git commit -m 'tuned profile with parallelism'

git push

cd ..

Q6. Create users.clone the project from <git server>

already having a user\_list.yaml file with content

users:

- name: bobby

first: john

middle: micheal

last: bobby

uid: 1003

- name: smith

first: steven

middle: franklin

last: smith

uid: 1007

- name: warner

first: david

middle: simon

last: warner

uid: 1009

create a playbook create\_user.yml which will create the users with the following details.

a. users will be created with the name and proper uid.

b. in the GECOS part it will be 'firstname middlename lastname' and the first letter should be capital.(eg John Micheal Bobby).

c. for every user it should create a 6 digit random password with SHA512 encryption. the salt value of the password will be stored in a file password-<username> in the directory on which the playbook will be executed in the control node

After done all push to git server.

Ans -:

git clone <git repo>

#cat create\_user.yml

---

- name: Create remote users

hosts: all

become: yes

vars\_files:

- user\_list.yml

tasks:

- name: Create users

vars:

password: "{{ lookup('password', 'filename-' + item.name + ' length=6 chars=digits') }}"

user:

name: "{{ item.name }}"

uid: "{{ item.uid }}"

state: present

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

update\_password: on\_create

comment: "{{ item.first | capitalize }} {{ item.middle | capitalize }} {{ item.last | capitalize }}"

loop: "{{ users }}"

git add .

git commit -m 'user account managed'

git push

cd ..

Q7. Install an example.collection from hub.lab17.example.com into /home/matthew/mycollections/.

Ans -:

a. Navigate to https://hub.lab17.example.com and then log in with student as your username and redhat123 as your password.

b. Browse the Ansible Content Collections that are available on the Collections page accessible from Collections > Collections.

c. #mkdir /home/student/mycollections

d. From the private automation hub's web UI, navigate to Collections > Repository Management. This page has the parameters that you need for configuring the ansible.cfg file.

e. #Cat ansible.cfg

[defaults]

**collections\_paths = ./mycollections:/usr/share/ansible/collections**

**[galaxy]**

**server\_list = published\_repo**

**[galaxy\_server.published\_repo]**

**url=https://hub.lab17.example.com/api/galaxy/content/published/**

**token=<put your token here>**

f. #ansible-galaxy collection install example.collection -p mycollections

g. #**ansible-galaxy collection list**

(guided exercise = 83)

Q8. Create a custom collection.

Clone the url <>

a. Collection name rhel.user

b. Create a role on it name newuser.

c. Copy a yml task file to newuser/tasks/main.yml file.

d. Copy a user.conf file to newuser/files/ .

E.publish your custom collection to the automation hub

Ans -:

a. #cd /home/matthew/customcollection

ansible.cfg inventory user.conf tasks\_main.yml

b. #ansible-galaxy collection init rhel.user

c. #cd /home/matthew/customcollections/rhel/user/

d. #mkdir meta

e. #cat meta/runtime.yml

---

requires\_ansible: '>=2.9.10'

f.cd roles/

g. ansible-galaxy init newuser

h. cp ~/customcollection/tasks\_main.yml newuser/tasks/main.yml

i. cp ~/customcollection/user.conf ~/customcollection/rhel/user/roles/newuser/files/

j. cd /home/student/customcollections/rhel/user/

rm -vr docs roles/newuser/{defaults,templates,handlers,tests,vars}

k. ansible-galaxy collection build

l. Copy generated tar file to /home/matthew/customcollection/ and publish to automation hub.

or

Cli command to publish: ansible-galaxy collection publish -v rhel-user-1.0.0.tar.gz

m. Approve after publish.

Q9. Create a custom execution environment name ee-ansible-user:v1.0

a. Base image -:ee-supported-rhel8:latest

b. Build image -:ansible-builder-rhel8:latest

c. Use rhel.user custom collection create in previous question.

D. publish to the automation hub

For certificate check /etc/ssl/tls/certs

Ans -: Guided Exercise – chapter -9

mkdir customcollection/ee-build

version: 1

build\_arg\_defaults:

EE\_BASE\_IMAGE: 'hub.lab17.example.com/ee-supported-rhel8:latest' EE\_BUILDER\_IMAGE: 'hub.lab17.example.com/ansible-builder-rhel8:latest'

dependencies:

galaxy: requirements.yml

Cat requirements.yml

collections:

- name: /build/rhel.user.1.0.0.tar.gz

type: file

Ansible-builder create

cp rhel-user-1.0.0.tar.gz context/\_build

ansible-builder build -t ee-ansible-

podman tag localhost/ hub.lab.example.com/

podman push hub.lab17.example.com/ww-ansible-rhel8:v1.0

Q10. Create a custom execution environment ee-dynamic-rhel8:v1.0

with Python package.

a. python36 and python3-ldap must be in that eei.

b. Base image -:ee-supported-rhel8:latest

c. Build image :-ansible-builder-rhel8:latest

d. Publish to private hub.

Ans -: Guided Exercise – chapter -9

version: 1

build\_arg\_defaults:

EE\_BASE\_IMAGE: 'hub.lab17.example.com/ee-supported-rhel8:2.0' EE\_BUILDER\_IMAGE: 'hub.lab17.example.com/ansible-builder-rhel8:2.0' dependencies:

python: requirements.txt

system: bindep.txt

Vim requirements.txt

python3-ldap

Vim bindep.txt

Python36

ansible-builder build -t ee-dynamic-rhel8:v1.0

Podman tag

Podman push

Q11. Run playbook in an execution environment (created in 10 ques)by script main.sh.

a. Git project need to clone.

b. A config file and dynamic inventory script (ldap-free-ipa.py) there.

c. Create playbook with deploy content task and run using above eei in web hostgroup.

d. Push to git.

Ans :-

a. Clone the git project in /home/matthew directory.

b. Provide execute permission to dynamic inventory.

Chmod a+x ldap-free-ipa.py

ansible-navigator inventory -m stdout -i inventory/ldap-free-ipa.py --graph web

c. Create playbook main.yml and add copy task on it.

d. Create a script file named main.sh

e. Add below command inside the script file and execute,

ansible-navigator run main.yml –i inventory –eei ee-dynamic-rhel8:v1.0 -–pp missing –m stdout

f. Push to git server.

Q12. Run a variable playbook.

a. Clone project from git url.

b. Config file dynamic inventory script available.

c. Create a playbook which copy content from CONTENT variable to a FILE and file is inside DIRECTORY variable

d.execute on testing groups, file not created in any other host

Ans :-

a. Clone git project in /home/matthew/ directory.

b. Provide execute permission to dynamic inventory script. Chmod a+x inventorya.py

inventory/inventorya.py --list

0/p testing node6.lab17.example.com

c. Create playbook add copy task with given variable.

vim master\_playbook.yml

* hosts: node6.lab17.example.com

become: yes

garther\_facts: false

Vars:

content: “sun comes up goes down”

file: banner

directory: /etc/motd.d/

tasks:

* name: copy content to index file

Copy:

Content: “{{ content }}”

Dest: “{{ directory}} / {{ file }}”

d. Push to git.

Q13. Use custom collection role.

a. Clone git project from url.

b. Config file and dynamic inventory.

c. Use custom role from rhel.user collection in a playbook.

Ans :-

a. Clone project from git to /home/matthew directory.

b. Provide execute permission to dynamic inventory script.

c. Check custom execution-environment defined in ansible-navigator.yml file.

---

ansible-navigator:

ansible:

config: ./ansible.cfg

execution-environment:

image: ee-ansible-user:v1.0

pull-policy: missing

d. Create playbook with below role defined

---

- name: Test newuser role from rhel.user `collection

hosts: all

roles:

- role: rhel.user.newuser

e. Push to git server.

Configure projects

14 Create the following Ansible automation controller projects

- Name: EX374 copy file project

- Organization: Default

- Credential type: Git

- URL: http://git.realm10.example.com/git/master\_playbook

- Name: EX374 user project

- Organization: Default

- Credential type: Git

- URL: http://git.realm10.example.com/git/master\_user

15 Configure inventory

Configure the following Ansible automation controller inventories:

- EX374 static inventory contains the following host groups

- Host group development which contains the host node4.realm10.example.com

- Host group testing which contains the host

- node6.realm10.example.com

- EX374 dynamic inventory contains the inventory source EX374 custom source which contains the inventory script inventory.py from project EX374 copy file project

Note that the inventory source is automatically updated before each launch

Do not create any other resources other than those mentioned above.

16 Configure Ansible automation controller execution environment

Create an Ansible automation controller execution environment named EX374 custom user execution environment that uses the hub.realm10.example.com/ee-user-supported:2.0 execution environment. The execution environment should download the image if not present before running.

17 Configure templates

Create the following Ansible automation controller job template:

- Template EX374 static copy project template

- When launched the template runs the playbook master\_playbook.yml in project EX374 copy file project against hosts in EX374 static inventory.

- Set the following variables in template EX374 static copy project template.

—

Directory: “/etc/motd.d”

File: “todays\_message”

Content: “The sun goes down, and then the moon comes up”

- Template EX374 dynamic copy project template

- When launched the template runs the playbook master\_playbook.yml in project EX374 copy file project against hosts in EX374 dynamic inventory.

- Set the following variables in template EX374 dynamic copy project template

—

Directory: “etc/issue.d”

File: “todays\_issue”

Content: “After the moon goes down, the sun comes up”

- Template EX374 user project template

- When launched the template runs the playbook main.yml in project EX374 user project against hosts in inventory EX374 static inventory

- The template uses the EX374 custom user execution environment.