Lab for Ansible

1) Create 3 Ec2 machine name them as : Ansibleserver

Node1 and node2

2) Go to software Mobaxterm and there login into all the three machines

3) **In Ansible machine/ansible server only :**

sudo su

yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm

yum update -y

yum install git python python-level python-pip openssl ansible -y

ansible --version

vi /etc/ansible/hosts

press i

## Add the below content in the hosts file…make sure you copy paste the correct ip

[demo]

<<private ip of the machine1>>

<<private ip of the machine2>>

press esc :wq

vi /etc/ansible/ansible.cfg

press i

Remove # from inventory (line 14)

Remove # from sudo-user (line 22)

esc and :wq

4) >>>>>>>do this step in all the nodes and ansible machine <<<<<<<

adduser ansible

# create a user with the name ansible

passwd ansible

#setting up the password for the user ansible

(now set the password)

## Now we need to provide the authentication rights to the ansible user. So that the ansible user can perform sudo commands …

visudo

press i

scroll down and find #All root to run any command ....

now there add

ansible ALL=(ALL) NOPASSWD: ALL (Add this below root)

(line number: 101)

5) **Do this step in ansible server only:**

su - ansible

*## su ansible means that you want to login as ansible user*

sudo yum install httpd -y

*# we are checking that are we able to install a software httpd via ansible user*

sudo yum remove httpd -y

exit

#*exit will exit from the ansible user*

6) **Do this step in ansible and node servers**

vi /etc/ssh/sshd\_config

press i

in authentication,

uncomment Permit root login yes

(remove #) (line number 38)

scroll down and uncomment password authentication yes(by removing #) (line61)

and comment password authentication no

(by putting #)(line 63)

7) **do this for nodes and ansible server**

service sshd restart

8)**In ansible server**:

##*Login as a ansible user*

su - ansible

#*# Trying to connect with node1 . you can copy paste the private ip of node1 from aws management console*

ssh <<private ip of node1>>

touch fileinnode1 file1innode1

exit

**Now go to node 1**

su - ansible

ls

check if fileinnode1 and file1innode1 is there

Again come back to ansible server

su - ansible

ssh <<private ip of node2>>

touch fileinnode2 file1innode2

exit

**Now go to node 2**

su - ansible

ls

check if fileinnode2 and file1innode2 is there

###########################REMOVE PASSWORD AUTHENTICATION USING TRUST RELATIONSHIP#########################

9) Go to ansible server (you need to ensure that you are logged in as ansible user(su - ansible))

ssh-keygen (press enter - enter - enter)

ls -a

cd .ssh/

ls

ssh-copy-id ansible@<<private ip of node1>>

password : last time it will ask for pass

ssh-copy-id ansible@<<private ip of node2>>

password : last time it will ask for pass

cd ..

ssh <<private ip of node1>>

exit

ssh <<private ip of node 2>>

exit

##########################################HOST PATTERN###########################################

10) Do this in ansible machine

ansible all --list-hosts

ansible demo --list-hosts

#show me the hosts within the demo group

################################################ADHOC COMMANDS ########################################

11) Go to ansible server

ansible demo -a "ls"

ansible all -a "touch fileinallnodes"

ansible demo -a "sudo yum install httpd -y"

12) Go to Nodes

Which httpd

<<Make sure httpd is present>>

13) Go to ansible server

ansible demo -a”touch file1”

ansible demo -a”touch file2”

14) Go to Nodes

su – ansible

ls

<<Make sure file1 and file2 present in Node servers>>

15) Go to ansible server

ansible demo -b -m yum -a “pkg=httpd state=present”

touch subbaramfile

16) Go to Nodes

su – ansible

ls

<<Make sure subbaramfile present in Node servers>>

17) Go to ansible server

ansible demo -b -m copy -a “src=subbaramfile dest=<copy the **path** from node dic by running “pwd”>”

ansible demo -b -m copy -a “src=subbaramfile dest=/home/ansible”

IF you do ls in node will see copy of subbaramfile

18) Go to ansible server

ansible demo -m command -a “ls -lh /home/ansible/subbaramfile”

Ansibleplay book 🡪 Written with YAML

Ansible playbooks always start with “---”

YAML 🡪Yet another language its a Data serialization language used to write configuration file

19) Go to ansible server

vi target.yml

It will start with ---

Write yaml script

-host: demo 🡪 the group which is host machine are present

user: ansible 🡪Ansible user who is performing the task

become: yes 🡪whatever action would be performed in node1 & node2 have sudo permission

connection: ssh 🡪How I’m going to connect

press i

save and quit :wq

20) Go to ansible server

ansible-playbook target.yml

vi target.yml

- hosts: demo

user: ansible

become: yes

connection: ssh

tasks:

- name: install apache server in node

action: yum name=httpd state=present

- name: install docker in node

action: yum name=docker state=present

21) Go to nodes

Which httpd

Which docker

<< [ansible@ip-172-31-44-93 ~]$ which docker

/usr/bin/docker

[ansible@ip-172-31-44-93 ~]$>>

Service docker status

<< [ansible@ip-172-31-44-93 ~]$ service docker status

Redirecting to /bin/systemctl status docker.service

● docker.service - Docker Application Container Engine

Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)

Active: **inactive** (dead)

Docs: https://docs.docker.com

[ansible@ip-172-31-44-93 ~]$ >>

23) Go to ansible server 🡪 to start the application

Vi targert1.yml

Press i

- hosts: demo

user: ansible

become: yes

connection: ssh

tasks:

- name: install apache server in node

action: yum name=httpd state=present

- name: install docker in node

action: yum name=docker state=present

- name: start the docker

action: service name=docker state=started

- name: start the apache

action: service name=httpd state=started

- name: copy the file from master to node

action: copy src=subbaramfile dest=/home/ansible

24) Go to node 🡪 to check whether application is started or not

service httpd status

service docker status

<<Make sure the application are in active running>>

25)Go to ansible server

cat >file4.txt 🡪 content of the file4.txt press enter for writing the content

hello world 🡪 after write the content and press ctrl + d to come out of the content

26) Go to node

ls

<make sure all the created file present in node>

cat file4.txt 🡪will see the content of txt file “Hello world”

27) Go to AWS 🡪 Lunching the application in nodes

Take public IP of the machine and lunch in browser it will open apache test page

Default location of apache server is “/var/www/html/

Create html file 🡪 cat >firstApplication.html

I'm the first application hosted by AWS press ctrl+d

**Ansible vault**

1. Go to ansible server 🡪 To create a vault file

<<Ansible-vault create subbaramVault.yml>>

It will ask to setup new password and confirm password

Yaml file will be opened

Write yaml script

- hosts: demo

user: ansible

become: yes

connection: ssh

tasks:

- name: install apache server in node

action: yum name=httpd state=present

- name: install docker in node

action: yum name=docker state=present

- name: start the docker

action: service name=docker state=started

- name: start the apache

action: service name=httpd state=started

- name: copy the file from master to node

action: copy src=subbaramfile dest=/home/ansible

Make sure yml file present in directory “ls”

1. Go to ansible server 🡪 To edit a vault file

<<ansible-vault edit subbaramVault.yml>>

It will ask a password to edit, enter it yml file will be opened we can edit

1. Go to ansible server 🡪 To change the password a vault file

<<ansible-vault rekey subbaramVault.yml>>

1. Go to ansible server 🡪 To run the vault file

<<ansible-playbook subbaramVault.yml --ask-vault-pass>>