Lab Exercise 01

Creating Static Host Inventory

Aditya Tomar

500106015

R2142221060

Batch-2(DevOps)

Objective: To create a static host inventory for managing and automating infrastructure tasks efficiently across multiple servers using Ansible

Tools required: Ubuntu OS

Prerequisites: You need to have Ansible installed to proceed with this demo

Steps to be followed:

1. Generate SSH key pair on the main node
2. Copy the SSH key to the two other nodes
3. Update the inventory or host file with the host IP address
4. Establish connectivity between the hosts specified in the host file and the Ansible server

Step 1: Generate SSH key pair on the main node

1. Use the following command to generate the SSH key on the Ansible server:
   ssh-keygen

```
[ec2-user@ip-172-31-18-141 ~]$ ansible --version
ansible 2.9.23
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/home/ec2-user/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/site-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.7.18 (default, Mar 13 2025, 19:20:03) [GCC 7.3.1 20180712 (Red Hat 7.3.1-17)]
[ec2-user@ip-172-31-18-141 ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ec2-user/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ec2-user/.ssh/id_rsa.
Your public key has been saved in /home/ec2-user/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:geJhQRYKxiGs+yxxvxmtJ2g0p/oZQaAwd/zWAASz0tU ec2-user@ip-172-31-18-141.ec2.internal
The key's randomart image is:
+---[RSA 2048]----+
|Oo==X+           |
|=O B..E.         |
|+ = +..o.        |
|.o o oo ..       |
| .. ..  S        |
|o +...           |
| *.*. .          |
|. *oo+.          |
|.=o ++           |
+----[SHA256]-----+
[ec2-user@ip-172-31-18-141 ~]$
```

Step 2: Copy the SSH key to the other two nodes

1. Use the following command to copy the public key to a file named **authorized_keys** in localhost:
   cat .ssh/id_rsa.pub >> .ssh/authorized_keys

2. Run the following command to go to the **.ssh** directory of the Ansible server:
   cd .ssh

```
syedsharozsimpl@ip-172-31-44-85:~$ cd  .ssh
syedsharozsimpl@ip-172-31-44-85:~/.ssh$
```

3. Run the following command to copy the public key to another node that will connect to the Ansible server:
   ssh-copy-id username@ip -p 22

```
syedsharozsimpl@ip-172-31-44-85:~/.ssh$ ssh-copy-id syedsharozsimpl@ip-172.17.0.1 -p 22
/usr/bin/ssh-copy-id: ERROR: Too many arguments.  Expecting a target hostname, got:

Usage: /usr/bin/ssh-copy-id [-h|-?|-f|-n|-s] [-i [identity_file]] [-p port] [-F alternative ssh_config file] [[-o <ssh -o options>] ...] [user@]hostn
ame
        -f: force mode -- copy keys without trying to check if they are already installed
        -n: dry run    -- no keys are actually copied
        -s: use sftp   -- use sftp instead of executing remote-commands. Can be useful if the remote only allows sftp
        -h|-?: print this help
syedsharozsimpl@ip-172-31-44-85:~/.ssh$
```

> **Note**: You must use a **username@ip** with your node and IP username, which are provided in the lab credential.

4. Execute the following command to exit the **.ssh** directory of the Ansible server:
   cd

```
syedsharozsimpl@ip-172-31-44-85:~/.ssh$ cd
syedsharozsimpl@ip-172-31-44-85:~$
```

Step 3: Update the inventory or host file with the host IP address

1. Use the following command to open the Ansible inventory file and add the host localhost to it:
   sudo vi /etc/ansible/hosts

```
syedsharozsimpl@ip-172-31-44-85:~$ sudo vi /etc/ansible/hosts
syedsharozsimpl@ip-172-31-44-85:~$
```

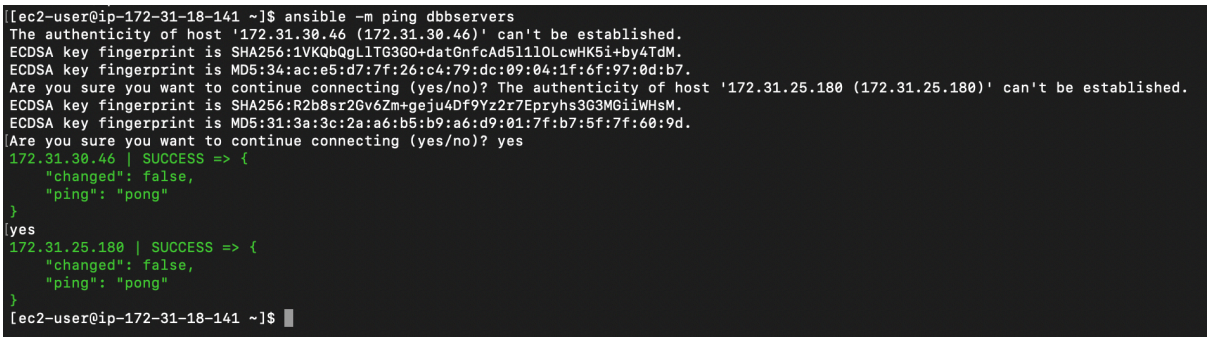
2. When the file opens, add the three lines of code below to the end of the file:
   [dbbservers]
   localhost:22
   172.31.5.76:22

```
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
#   - Comments begin with the '#' character
#   - Blank lines are ignored
#   - Groups of hosts are delimited by [header] elements
#   - You can enter hostnames or ip addresses
#   - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrouped hosts, specify before any group headers.

## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10

# Ex 2: A collection of hosts belonging to the 'webservers' group

## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110

# If you have multiple hosts following a pattern you can specify
# them like this:

## www[001:006].example.com

# Ex 3: A collection of database servers in the 'dbservers' group

## [dbservers]
##
## db01.intranet.mydomain.net
## db02.intranet.mydomain.net
## 10.25.1.56
## 10.25.1.57

# Here's another example of host ranges, this time there are no
# leading 0s:

## db-[99:101]-node.example.com
[dbbservers]
localhost
172.31.30.46 ansible_user=ec2-user ansible_python_interpreter=/usr/bin/python3
172.31.25.180 ansible_user=ec2-user ansible_python_interpreter=/usr/bin/python3
~
~
~
```

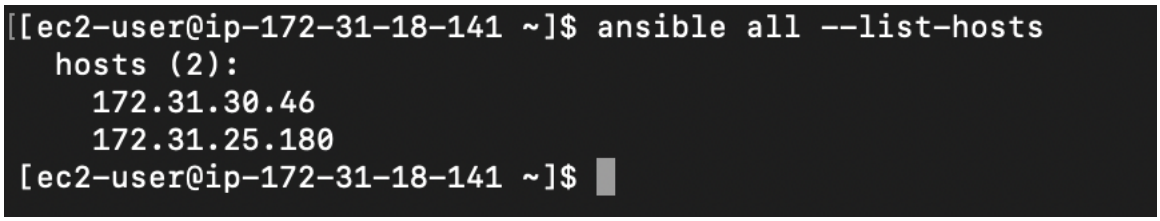
Note: Press **esc**, then write **:wq** and press **enter** to save the file.

Step 4: Establish connectivity between the hosts specified in the host file and the Ansible server

1. Run the following command to copy the public key to another node that will connect to the Ansible server:
   ansible -m ping dbbservers



2. Use the following command to check the number of hosts in the host file:
   ansible all —list-hosts



By following these steps, you have successfully created a static host inventory for managing and automating infrastructure tasks efficiently across multiple servers using Ansible.