Lab Exercise 01 Creating Static HostInventory

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

```
[ec2-user@ip-172-31-8-236 ~]$ sudo su
[root@ip-172-31-8-236 ec2-user]# visudo
[root@ip-172-31-8-236 ec2-user]# vi /etc/ssh/sshd_config
[root@ip-172-31-8-236 ec2-user]# service sshd restart
Redirecting to /bin/systemctl restart sshd.service
[root@ip-172-31-8-236 ec2-user]# su -do
su: invalid option -- 'd'
Try 'su --help' for more information.
[root@ip-172-31-8-236 ec2-user]# su - do
[do@ip-172-31-8-236 ~]$ ssh 172.31.6.104
The authenticity of host '172.31.6.104 (172.31.6.104)' can't be established.
ECDSA key fingerprint is SHA256:snAWMWHvK75T0j2sDd4LTMb/i6vtwxJ9uxf6WIL8rqA.
ECDSA key fingerprint is MD5:dd:02:76:74:99:31:3f:8c:ce:67:57:7a:61:3a:1c:8e.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.31.6.104' (ECDSA) to the list of known hosts.
do@172.31.6.104's password:
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2026-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/
```

```
[do@ip-172-31-8-236 ~]$ ssh 172.31.5.230
The authenticity of host '172.31.5.230 (172.31.5.230)' can't be established.
ECDSA key fingerprint is SHA256:xFKb5WRxVgHBp3tj1FHjaHocF4HVpVazmu5lQyponYM.
ECDSA key fingerprint is MD5:9b:a6:83:2a:8d:cb:5f:ec:ee:31:92:6b:6d:00:0b:c0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.31.5.230' (ECDSA) to the list of known hosts.
Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[do@ip-172-31-8-236 ~]$ ssh 172.31.5.230
do@172.31.5.230's password:
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2026-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/
```

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

```
[do@ip-172-31-8-236 ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/do/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/do/.ssh/id_rsa.
Your public key has been saved in /home/do/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:L3ct5mbFqPxK8F+YohNtqfSd9CoFZsTYywBBN5lQPHs do@ip-172-31-8-236.ap-south-1.compute.internal
The key's randomart image is:
+---[RSA 2048]----+
|     .=*+*       |
|       .B.+      |
|         * .     |
|        . E      |
|        S= o o   |
|        o++ ++o  |
|       ..**=*+o  |
|        o==*=o.  |
|         ...*=.  |
+----[SHA256]-----+
```

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

```
[do@ip-172-31-8-236 ~]$ cd .ssh
[do@ip-172-31-8-236 .ssh]$ ls
id_rsa  id_rsa.pub  known_hosts
```

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

```
[do@ip-172-31-8-236 .ssh]$ ssh-copy-id do@172.31.6.104
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/do/.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
do@172.31.6.104's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'do@172.31.6.104'"
and check to make sure that only the key(s) you wanted were added.
```

> **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

```
[do@ip-172-31-8-236 .ssh]$ su - do
Password:
Last login: Thu Apr 10 09:24:38 UTC 2025 on pts/0
[do@ip-172-31-8-236 ~]$ sudo su
[root@ip-172-31-8-236 do]# sudo vi /etc/ansible/hosts
```

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

```
[root@ip-172-31-8-236 do]# sudo vi /etc/ansible/hosts
```

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

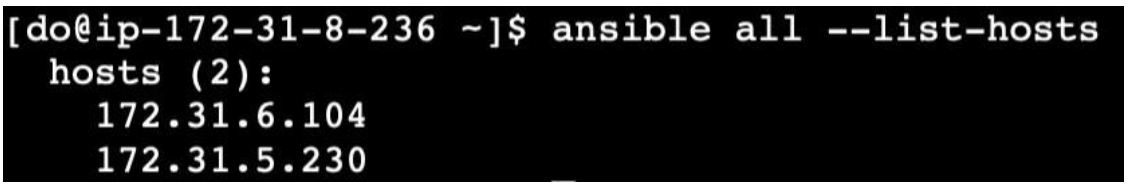
```
## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110



# 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
[dbservers]
172.31.6.104
172.31.5.230
# Here's another example of host ranges, this time there are no
# leading 0s:

## db-[99:101]-node.example.com

"/etc/ansible/hosts" 48L, 1055B
```

> **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 dbservers

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.