Lab Exercise 01

Creating Static Host Inventory

By: Ansh Tyagi

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

Tools required: Ubuntu OS

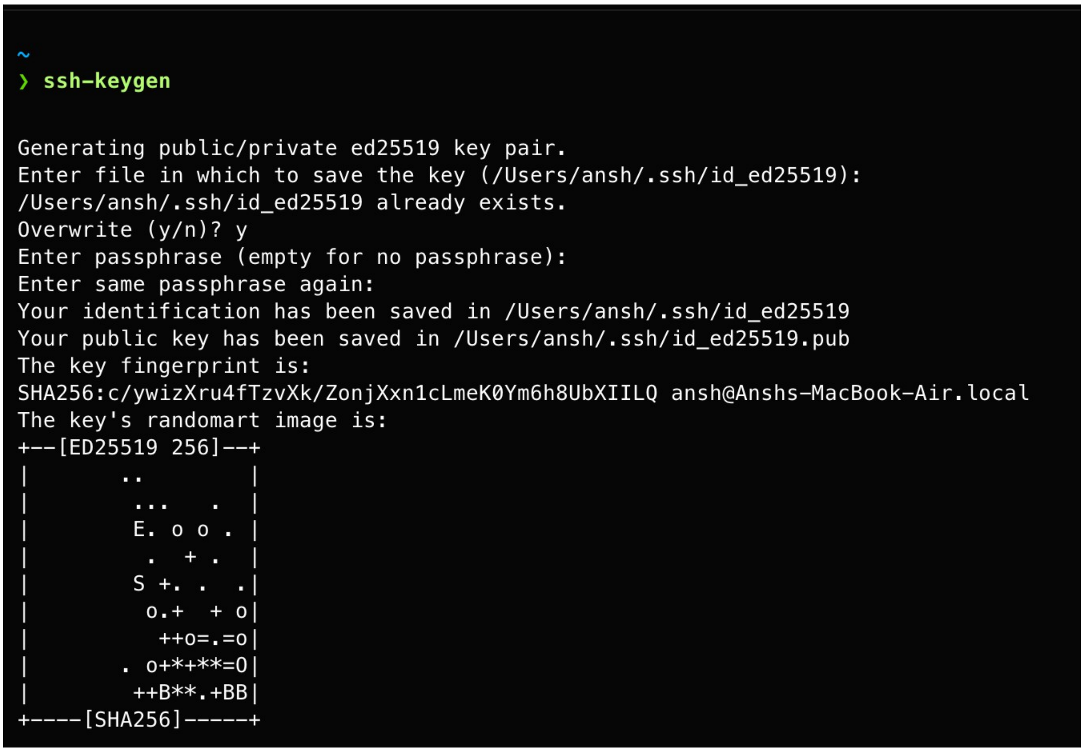
Prerequisites: You need to have Ansible installed to proceed with this demo

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

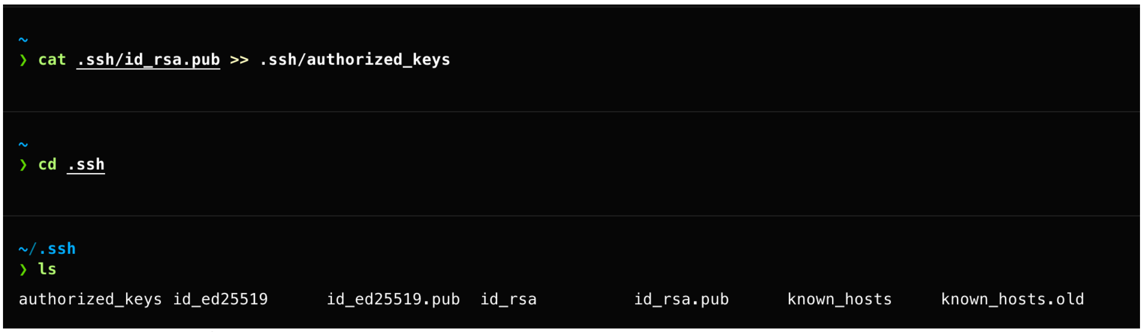


```
~
❯ ssh-keygen

Generating public/private ed25519 key pair.
Enter file in which to save the key (/Users/ansh/.ssh/id_ed25519):
/Users/ansh/.ssh/id_ed25519 already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/ansh/.ssh/id_ed25519
Your public key has been saved in /Users/ansh/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:c/ywizXru4fTzvXk/ZonjXxn1cLmeK0Ym6h8UbXIILQ ansh@Anshs-MacBook-Air.local
The key's randomart image is:
+--[ED25519 256]--+
|       ..        |
|        ...   .  |
|        E. o o . |
|         .  + .  |
|        S +. .  .|
|         o.+  + o|
|          ++o=.=o|
|       . o+*+**=O|
|        ++B**.+BB|
+----[SHA256]-----+
```

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

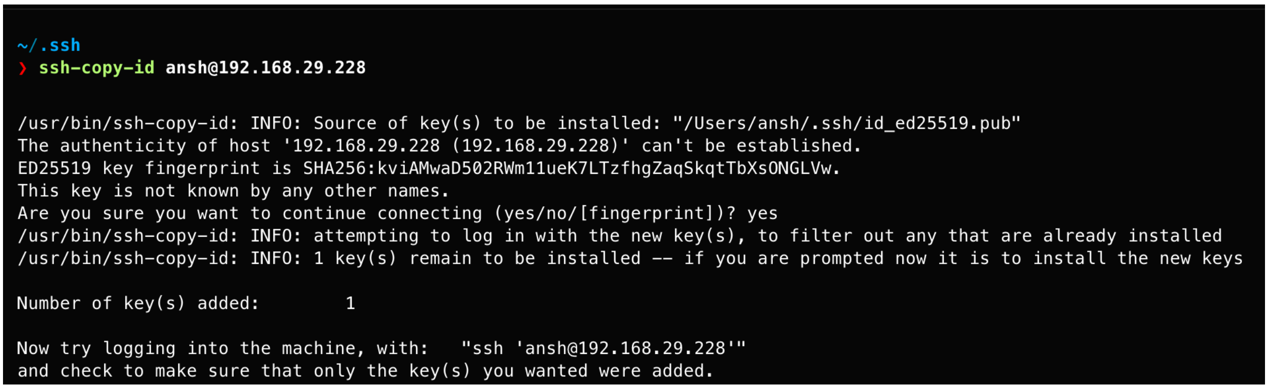
1. Use the following command to copy the public key to a file named **authorized_keys** in localhost:
   cat .ssh/id_rsa.pub >> .ssh/authorized_keys

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



```
~
❯ cat .ssh/id_rsa.pub >> .ssh/authorized_keys

~
❯ cd .ssh

~/.ssh
❯ ls
authorized_keys id_ed25519      id_ed25519.pub  id_rsa          id_rsa.pub      known_hosts     known_hosts.old
```

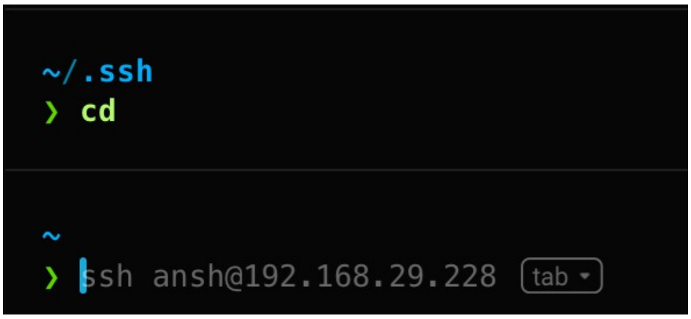
3. Run the following command to copy the public key to another node that will connect to the Ansible server:
   ssh-copy-id username@ip -p 22



```
~/.ssh
❯ ssh-copy-id ansh@192.168.29.228

/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/Users/ansh/.ssh/id_ed25519.pub"
The authenticity of host '192.168.29.228 (192.168.29.228)' can't be established.
ED25519 key fingerprint is SHA256:kviAMwaD502RWm11ueK7LTzfhgZaqSkqtTbXsONGLVw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/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

Number of key(s) added:        1

Now try logging into the machine, with:   "ssh 'ansh@192.168.29.228'"
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.

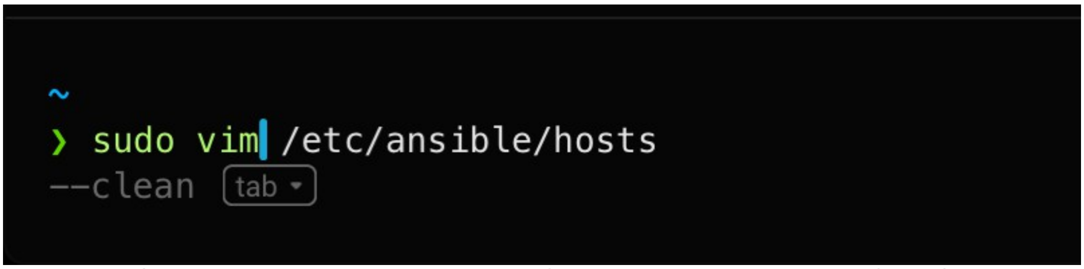
4. Execute the following command to exit the **.ssh** directory of the Ansible server:
   cd



```
~/.ssh
❯ cd

~
❯ ssh ansh@192.168.29.228
```

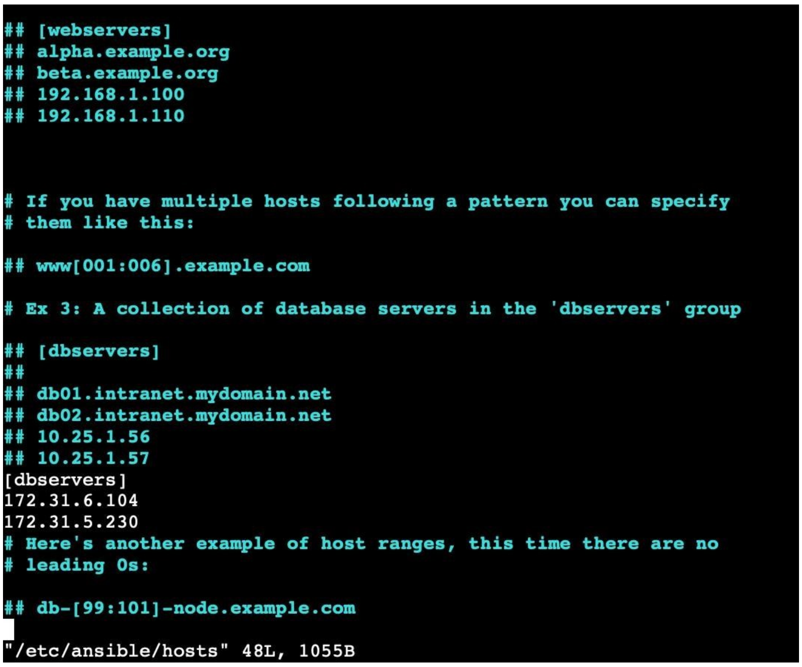
Step 3: Update the inventory or host file with the host IP address

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



```
~
❯ sudo vim /etc/ansible/hosts
--clean
```

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

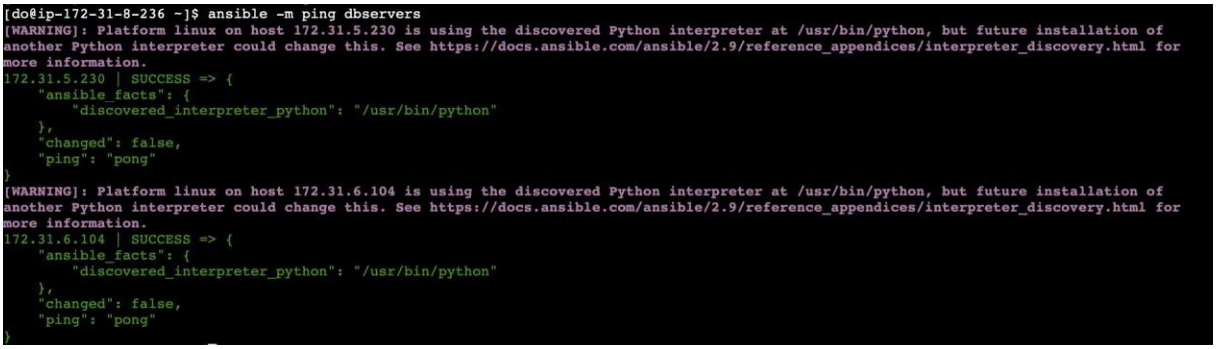


```
## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110



# 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.

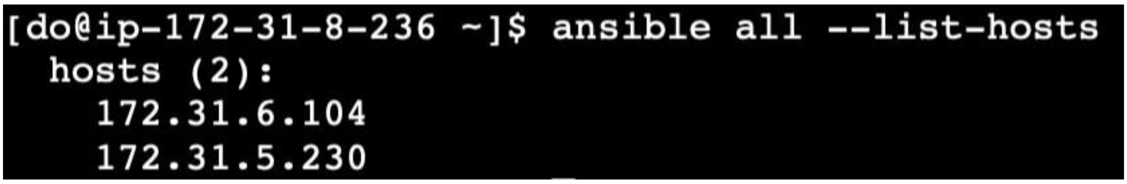
Step 4: Establish connectivity between the hosts specified in the host file and the Ansible server

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



```
[do@ip-172-31-8-236 ~]$ ansible -m ping dbservers
[WARNING]: Platform linux on host 172.31.5.230 is using the discovered Python interpreter at /usr/bin/python, but future installation of
another Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for
more information.
172.31.5.230 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host 172.31.6.104 is using the discovered Python interpreter at /usr/bin/python, but future installation of
another Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for
more information.
172.31.6.104 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
```

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



```
[do@ip-172-31-8-236 ~]$ ansible all --list-hosts
  hosts (2):
    172.31.6.104
    172.31.5.230
```

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