System Provisioning and

Configuration Management Lab

Submitted To:

Dr. Hitesh Kumar

Sharma

Submitted by:
Saumya Sehgal
500105655
R2142220635
Devops B2(NH)

EXPERIMENT 1

Lab Exercise: Creating Static Host Inventory

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

Tools required: Ubuntu OS

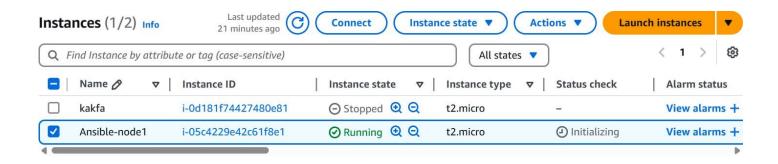**Prerequisites:** You need to have Ansible installed to proceed with this demo

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: Launch EC2 Instance

1. Go to AWS Console → EC2 → Launch instance
2. OS: **Ubuntu 22.04** or similar
3. Instance type: **t2.micro** (Free Tier)
4. Enable **port 22 (SSH)** in Security Group

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
☐	kakfa	i-0d181f74427480e81	⊖ Stopped	t2.micro	–	View alarms +
☑	Ansible-node1	i-05c4229e42c61f8e1	⊘ Running	t2.micro	⏱ Initializing	View alarms +

Instances (1/2) Info — Connect — Instance state ▾ — Actions ▾ — Launch instances ▾

☐ Install Ansible on EC2: Once you're logged into the instance: **sudo apt update**

sudo apt install ansible -y

```
ubuntu@ip-172-31-15-82:~$ sudo apt update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
```

```
ubuntu@ip-172-31-15-82:~$ sudo apt install ansible -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ansible-core python3-argcomplete python3-dnspython python3-kerberos python3-libcloud python3-lockfile python3-ntlm-auth
  python3-passlib python3-requests-ntlm python3-resolvelib python3-selinux python3-simplejson python3-winrm python3-xmltodict
Suggested packages:
  cowsay sshpass python3-trio python3-aioquic python3-h2 python3-httpx python3-httpcore python-lockfile-doc
The following NEW packages will be installed:
  ansible ansible-core python3-argcomplete python3-dnspython python3-kerberos python3-libcloud python3-lockfile python3-ntlm-auth
  python3-passlib python3-requests-ntlm python3-resolvelib python3-selinux python3-simplejson python3-winrm python3-xmltodict
0 upgraded, 15 newly installed, 0 to remove and 68 not upgraded.
Need to get 19.5 MB of archives.
After this operation, 315 MB of additional disk space will be used.
```

```
ubuntu@ip-172-31-15-82:~$ ansible --version
ansible [core 2.16.3]
  config file = None
  configured module search path = ['/home/ubuntu/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/ubuntu/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.12.3 (main, Feb  4 2025, 14:48:35) [GCC 13.3.0] (/usr/bin/python3)
  jinja version = 3.1.2
  libyaml = True
ubuntu@ip-172-31-15-82:~$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_ed25519):
```

Step 2: Generate SSH key pair on the main node

2.1 Use the following command to generate the SSH key on the Ansible server: **ssh-keygen**

```
ubuntu@ip-172-31-15-82:~$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_ed25519
Your public key has been saved in /home/ubuntu/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:MhsVwbyTOP1IfESUOV9iCMbEjWnebGbDzduEwXI8CoY ubuntu@ip-172-31-15-82
The key's randomart image is:
+--[ED25519 256]--+
|      B*Oo*       |
|     E.@.O O .    |
|      B.O X *     |
|     o.O @ = .    |
|     +oSO . +     |
|      =. . . .    |
|      .           |
|                  |
|                  |
+----[SHA256]-----+
```

Step 3: Copy the SSH key to the other two nodes

3.1 Use the following command to copy the public key to a file named **authorized_keys** in localhost: **cat ~/.ssh/id_ed25519.pub >> ~/.ssh/authorized_keys**

```
ubuntu@ip-172-31-15-82:~$ cat ~/.ssh/id_ed25519.pub >> ~/.ssh/authorized_keys
```

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

```
ubuntu@ip-172-31-15-82:~$ cd ~/.ssh/
ubuntu@ip-172-31-15-82:~/.ssh$ ip a
```

3.3 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**

```
ubuntu@ip-172-31-15-82:~/.ssh$ ssh-copy-id ubuntu@172.31.15.82 -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|-x] [-i [identity_file]] [-p port] [-F alternative ssh_config file] [-t target_path] [[-o <ssh
  -o options>] ...] [user@]hostname
        -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
        -x: debug      -- enables -x in this shell, for debugging
        -h|-?: print this help
ubuntu@ip-172-31-15-82:~/.ssh$ ssh-copy-id ubuntu@172.31.15.82
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ubuntu/.ssh/id_ed25519.pub"
The authenticity of host '172.31.15.82 (172.31.15.82)' can't be established.
ED25519 key fingerprint is SHA256:hGGNahApqYNjGRYVqsw7v8ovyOD7weWhCot7NDale24.
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: WARNING: All keys were skipped because they already exist on the remote system.
               (if you think this is a mistake, you may want to use -f option)
```

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

```
ubuntu@ip-172-31-15-82:~/.ssh$ cd
```

Step 4: Update the inventory or host file with the host IP address

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

```
ubuntu@ip-172-31-15-82:~$ sudo mkdir -p /etc/ansible
ubuntu@ip-172-31-15-82:~$ sudo touch /etc/ansible/hosts
ubuntu@ip-172-31-15-82:~$ sudo vi /etc/ansible/hosts
```

4.2 When the file opens, add the three lines of code below to the end of the file:

[dbbservers]
localhost
65.0.183.70
172.31.15.82

```
[dbbservers]
localhost
65.0.183.70
172.31.15.82

~
~
~
~
~
~
~
```

Step 5: Establish connectivity between the hosts specified in the host file and the Ansible server

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

```
ubuntu@ip-172-31-15-82:~$ ansible -m ping dbbservers
172.31.15.82 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
65.0.183.70 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
localhost | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

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

```
ubuntu@ip-172-31-15-82:~$ ansible all --list-hosts
  hosts (3):
    localhost
    65.0.183.70
    172.31.15.82
```

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