UPES

UNIVERSITY OF TOMORROW

System Provisioning and Configuration Management LAB

SUBMITTED TO
Dr. Hitesh Kumar Sharma

SUBMITTED BY
Siddharth Agarwal
500107594
R2142220663
Btech CSE DevOps B1

Lab Exercise 03

Executing Ad Hoc Commands

Objective: To demonstrate ad hoc commands for quickly executing tasks on remote servers without writing full playbooks

Tools required: Ansible, Ubuntu OS

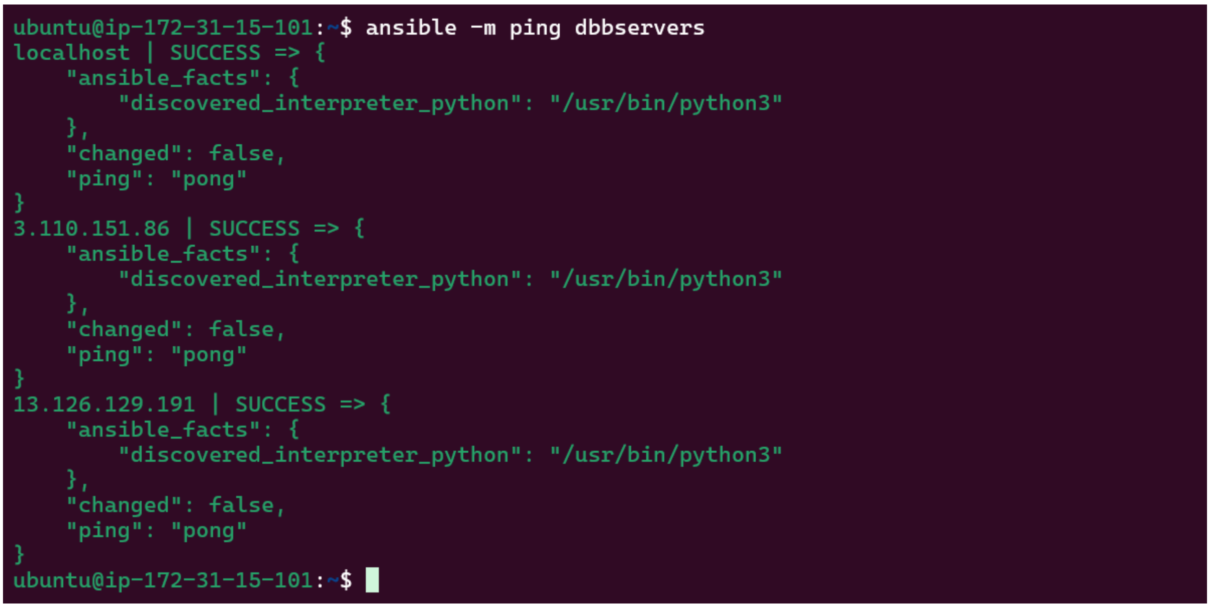
Prerequisites: None

Steps to be followed:

1. Generate SSH key pair on the main node
2. Copy the SSH key on the other two nodes
3. Update the host file with the host IP address
4. Establish connectivity between specified hosts and the Ansible server
5. Gather System Information Using Ad-Hoc Commands

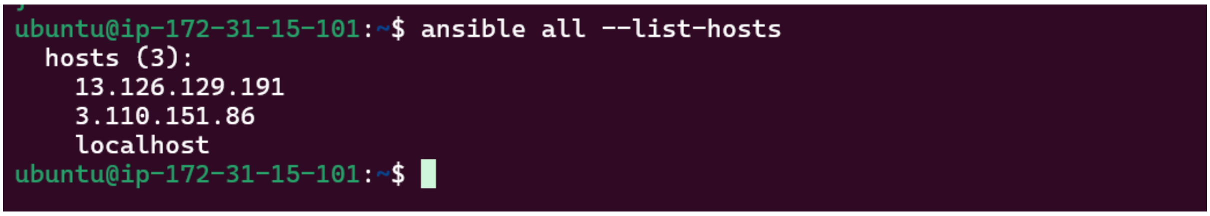
Step 1: Establish connectivity between specified hosts and the Ansible server

4.1 Run the following command to verify connectivity to all servers listed under the **webservers** group in your Ansible hosts file:
ansible -m ping dbservers



```
ubuntu@ip-172-31-15-101:~$ ansible -m ping dbbservers
localhost | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
3.110.151.86 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
13.126.129.191 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-15-101:~$ 
```

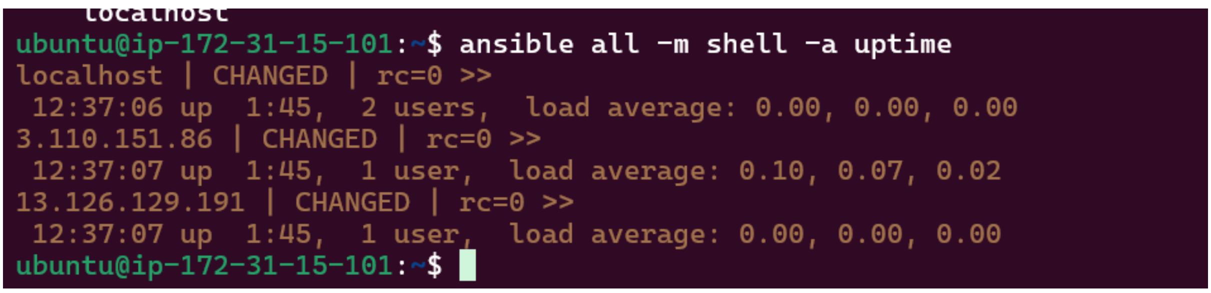
4.2 Use the following command to check the number of hosts in the host file:
ansible all --list-hosts



```
ubuntu@ip-172-31-15-101:~$ ansible all --list-hosts
  hosts (3):
    13.126.129.191
    3.110.151.86
    localhost
ubuntu@ip-172-31-15-101:~$
```

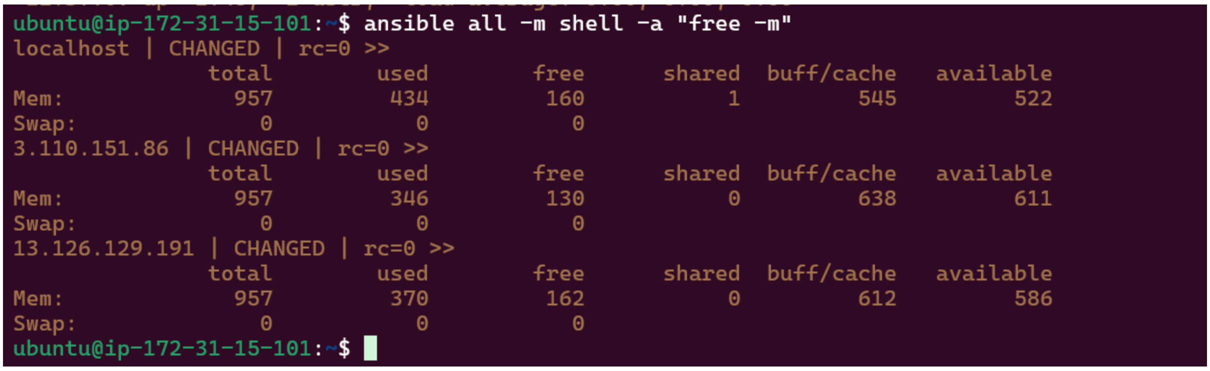
Step 2: Gather System Information Using Ad Hoc Commands

5.1 Run the following command to obtain the uptime from all managed hosts using an ad hoc command:
ansible all -m shell -a uptime



```
    localhost
ubuntu@ip-172-31-15-101:~$ ansible all -m shell -a uptime
localhost | CHANGED | rc=0 >>
 12:37:06 up  1:45,  2 users,  load average: 0.00, 0.00, 0.00
3.110.151.86 | CHANGED | rc=0 >>
 12:37:07 up  1:45,  1 user,  load average: 0.10, 0.07, 0.02
13.126.129.191 | CHANGED | rc=0 >>
 12:37:07 up  1:45,  1 user,  load average: 0.00, 0.00, 0.00
ubuntu@ip-172-31-15-101:~$
```

5.2 Similarly, execute the below command to obtain detailed information about memory usage on all hosts:
ansible all -m shell -a "free -m"



```
ubuntu@ip-172-31-15-101:~$ ansible all -m shell -a "free -m"
localhost | CHANGED | rc=0 >>
               total        used        free      shared  buff/cache   available
Mem:             957         434         160           1         545         522
Swap:              0           0           0
3.110.151.86 | CHANGED | rc=0 >>
               total        used        free      shared  buff/cache   available
Mem:             957         346         130           0         638         611
Swap:              0           0           0
13.126.129.191 | CHANGED | rc=0 >>
               total        used        free      shared  buff/cache   available
Mem:             957         370         162           0         612         586
Swap:              0           0           0
ubuntu@ip-172-31-15-101:~$
```

You will see that Ansible logs in to each machine in turn and runs the uptime command, returning the current uptime output.

By following these steps, you have successfully demonstrated how to use ad hoc commands for quickly executing tasks on remote servers without the need for full playbooks.