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

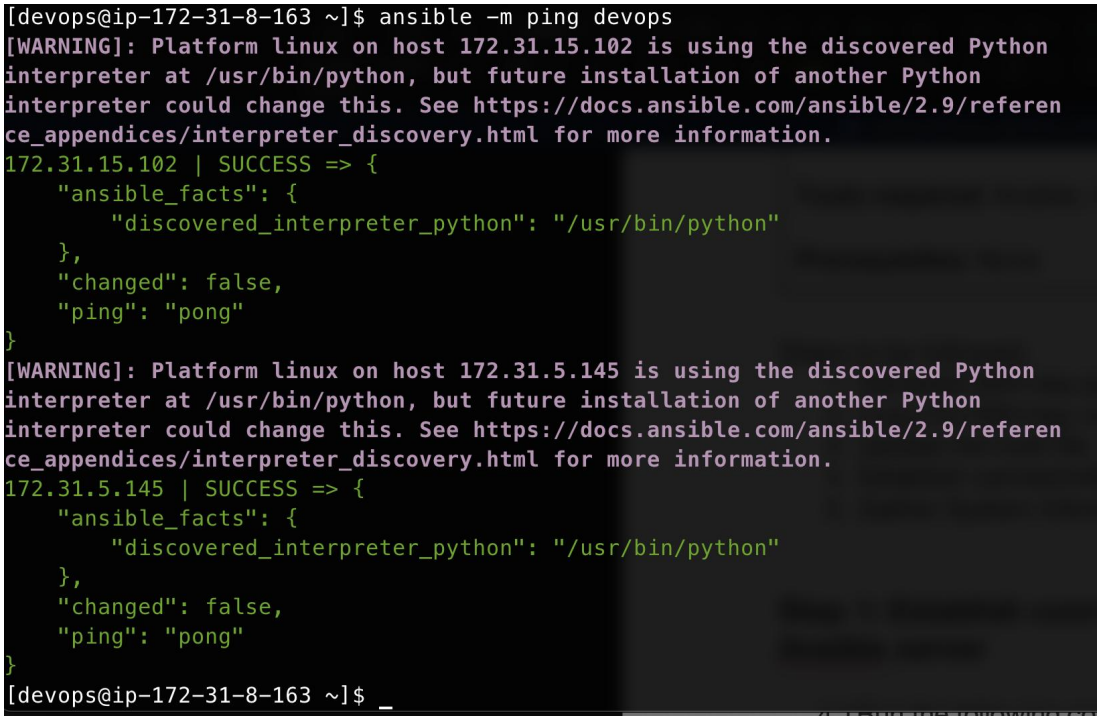
Steps to be followed:

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



4.2 Use the following command to check the number of hosts in the host file:

ansible all --list-hosts

```
[devops@ip-172-31-8-163 ~]$ ansible all --list-hosts
  hosts (2):
    172.31.15.102
    172.31.5.145
[devops@ip-172-31-8-163 ~]$ _
```

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

```
[devops@ip-172-31-8-163 ~]$ ansible all -m shell -a uptime
[WARNING]: Platform linux on host 172.31.5.145 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.
172.31.5.145 | CHANGED | rc=0 >>
 05:53:04 up  1:35,  2 users,  load average: 0.00, 0.00, 0.00
[WARNING]: Platform linux on host 172.31.15.102 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.
172.31.15.102 | CHANGED | rc=0 >>
 05:53:04 up  1:35,  3 users,  load average: 0.00, 0.00, 0.00
[devops@ip-172-31-8-163 ~]$
```

5.2 Similarly, execute the below command to obtain detailed information about memory usage on all hosts:

ansible all -m shell -a "free -m"

```
[devops@ip-172-31-8-163 ~]$ ansible all -m shell -a "free -m"
[WARNING]: Platform linux on host 172.31.15.102 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.
172.31.15.102 | CHANGED | rc=0 >>
              total        used        free      shared  buff/cache   available
Mem:            952          99         222           0         630         711
Swap:             0           0           0
[WARNING]: Platform linux on host 172.31.5.145 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.
172.31.5.145 | CHANGED | rc=0 >>
              total        used        free      shared  buff/cache   available
Mem:            952          97         220           0         635         713
Swap:             0           0           0
[devops@ip-172-31-8-163 ~]$ _
```

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.