

## 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 **webserver** group in your Ansible hosts file:

**ansible -m ping dbserver**

```
[devops@ip-172-31-9-95 ~]$ ansible -m ping devops
[WARNING]: Platform linux on host 172.31.9.135 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.
172.31.9.135 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
```

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

**ansible all --list-hosts**

```
pattern
ansible: error: unrecognized arguments: --all
[devops@ip-172-31-9-95 ~]$ ansible all --list-hosts
 hosts (2):
   172.31.9.135
   172.31.9.140
[devops@ip-172-31-9-95 ~]$ |
```

## 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-172-31-9-95 ~]$ ansible all -m shell -a uptime
The authenticity of host '172.31.9.140 (172.31.9.140)' can't be established.
ECDSA key fingerprint is SHA256:ehesJBMugCdb18jvjSaxCGCBPVBzmgh4q0tSmdZ8wy4.
ECDSA key fingerprint is MD5:0a:03:52:90:e2:eb:bc:aa:3e:ee:1d:84:04:81:f7:5e.
Are you sure you want to continue connecting (yes/no)? [WARNING]: Platform linux on host 172.31.9.135 is using the default
Python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html
172.31.9.135 | CHANGED | rc=0 >>
18:28:51 up 28 min,  2 users,  load average: 0.00, 0.00, 0.00
```

- 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-172-31-9-95 ~]$ ansible all -m shell -a "free -m"
[WARNING]: Platform linux on host 172.31.9.135 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.
172.31.9.135 | CHANGED | rc=0 >>
total      used      free      shared  buff/cache   available
Mem:      952      84      285         0       582       733
Swap:      0         0         0
```

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.