Lab: Running ad hoc Commands Using Ansible

Introduction:

An **ad hoc command** is a way of executing a single Ansible task quickly, one that you do not need to save to run again later. They are simple, online operations that can be run without writing a playbook. Ad hoc commands are useful for quick tests and changes. For example, you can use an ad hoc command to make sure that a certain line exists in the /etc/hosts file on a group of servers. You could use another ad hoc command to efficiently restart a service on many different machines or to ensure that a particular software package is up-to-date.

To run an ad hoc command, the command must be framed or have the following syntax.

ansible <host-pattern> [options]

Objectives:

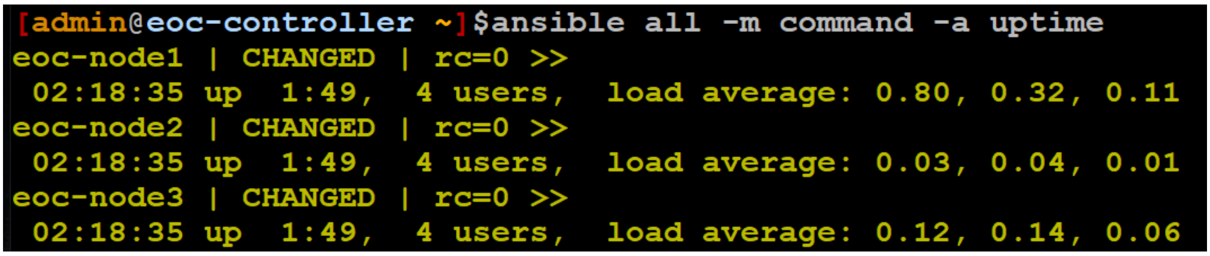
- **Running Ad-hoc Commands**

1. Running Ad-hoc Commands

1.1 In this example, we are going to understand the uptime of the hosts.

```
# ansible all -m command -a uptime
```

Output:

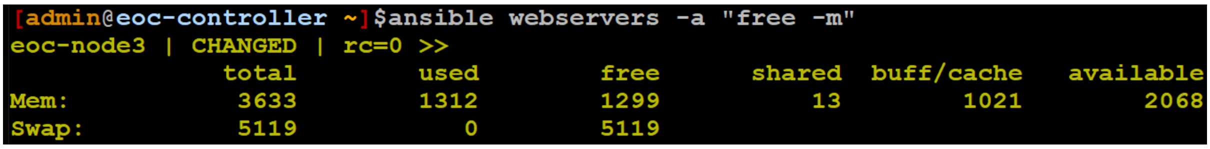


```
[admin@eoc-controller ~]$ansible all -m command -a uptime
eoc-node1 | CHANGED | rc=0 >>
 02:18:35 up  1:49,  4 users,  load average: 0.80, 0.32, 0.11
eoc-node2 | CHANGED | rc=0 >>
 02:18:35 up  1:49,  4 users,  load average: 0.03, 0.04, 0.01
eoc-node3 | CHANGED | rc=0 >>
 02:18:35 up  1:49,  4 users,  load average: 0.12, 0.14, 0.06
```

1.2 The following ansible ad hoc command would help you get the free memory of all the hosts in the host group named webservers.

```
# ansible webservers -a "free -m"
```

Output:

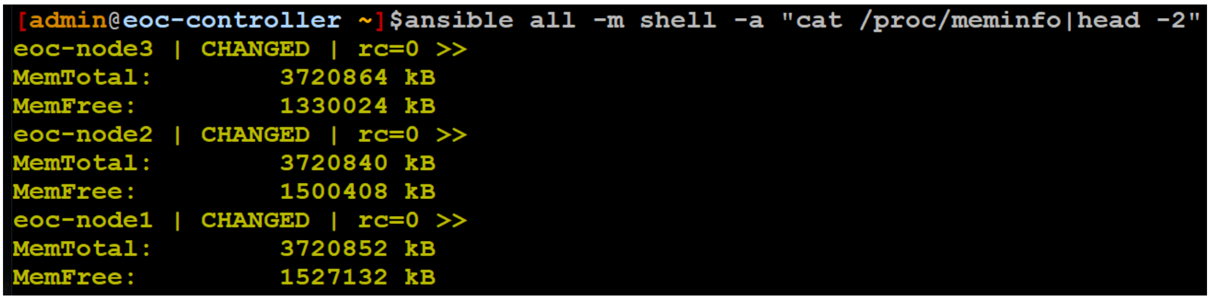


```
[admin@eoc-controller ~]$ansible webservers -a "free -m"
eoc-node3 | CHANGED | rc=0 >>
              total        used        free      shared  buff/cache   available
Mem:           3633        1312        1299          13        1021        2068
Swap:          5119           0        5119
```

1.3 Ansible ad hoc command to get physical memory allocated to the host.

```
# ansible all -m shell -a "cat /proc/meminfo|head -2"
```

Output:

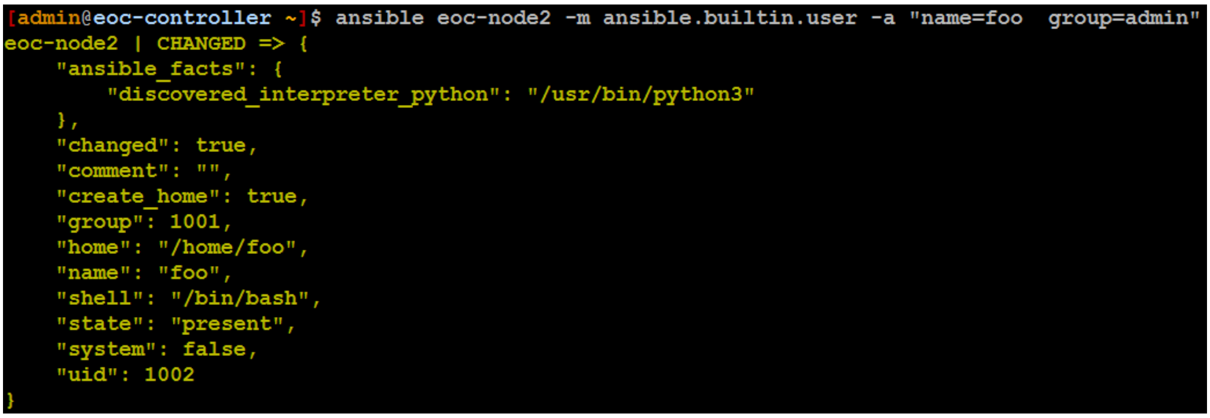


```
[admin@eoc-controller ~]$ansible all -m shell -a "cat /proc/meminfo|head -2"
eoc-node3 | CHANGED | rc=0 >>
MemTotal:        3720864 kB
MemFree:         1330024 kB
eoc-node2 | CHANGED | rc=0 >>
MemTotal:        3720840 kB
MemFree:         1500408 kB
eoc-node1 | CHANGED | rc=0 >>
MemTotal:        3720852 kB
MemFree:         1527132 kB
```

1.4 Creating linux users using ansible Ad-Hoc commands.

```
# ansible eoc-node2 -m ansible.builtin.user -a "name=foo
group=admin"
```

Output:

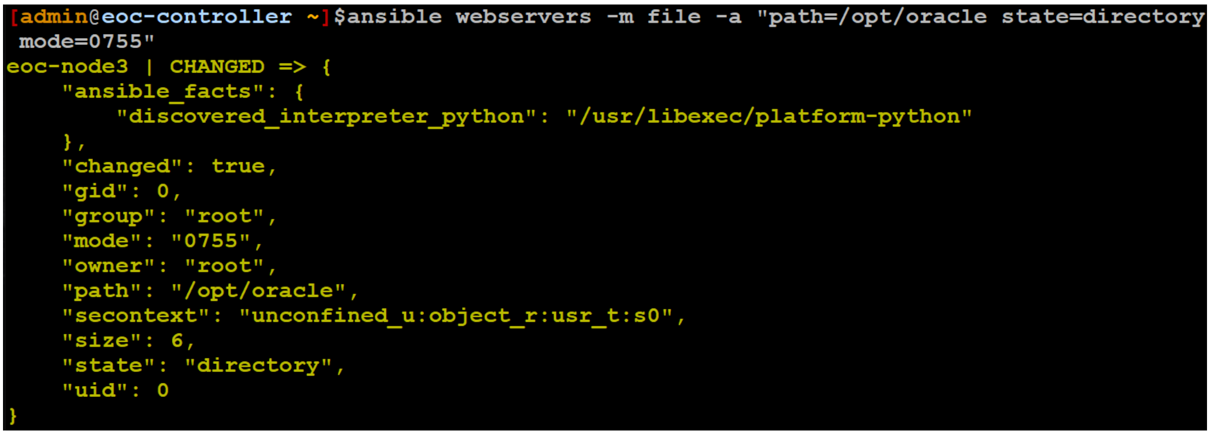


```
[admin@eoc-controller ~]$ ansible eoc-node2 -m ansible.builtin.user -a "name=foo  group=admin"
eoc-node2 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": true,
    "comment": "",
    "create_home": true,
    "group": 1001,
    "home": "/home/foo",
    "name": "foo",
    "shell": "/bin/bash",
    "state": "present",
    "system": false,
    "uid": 1002
}
```

1.5 Create a directory with 755 permissions using ansible ad hoc command.

```
# ansible webservers -m file -a "path=/opt/oracle
state=directory mode=0755"
```

Outptu:



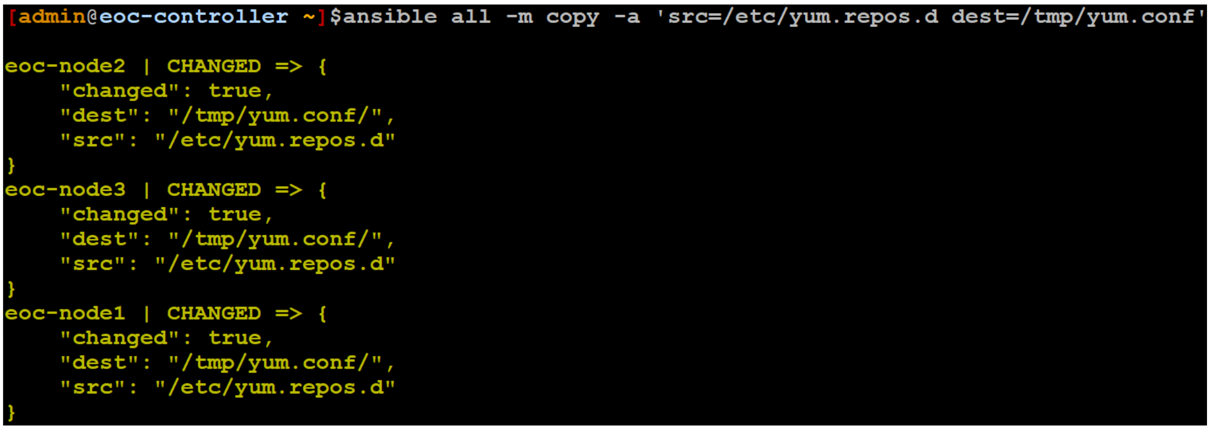
```
[admin@eoc-controller ~]$ansible webservers -m file -a "path=/opt/oracle state=directory
 mode=0755"
eoc-node3 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": true,
    "gid": 0,
    "group": "root",
    "mode": "0755",
    "owner": "root",
    "path": "/opt/oracle",
    "secontext": "unconfined_u:object_r:usr_t:s0",
    "size": 6,
    "state": "directory",
    "uid": 0
}
```

1.6 File Transfer

we can use ad-hoc commands for doing **SCP** (secure copy protocol) which means lots of files in parallel on multiple machines or servers.

```
# ansible all -m copy -a 'src=/etc/yum.repos.d
dest=/tmp/yum.conf'
```

Output:



```
[admin@eoc-controller ~]$ansible all -m copy -a 'src=/etc/yum.repos.d dest=/tmp/yum.conf'

eoc-node2 | CHANGED => {
    "changed": true,
    "dest": "/tmp/yum.conf/",
    "src": "/etc/yum.repos.d"
}
eoc-node3 | CHANGED => {
    "changed": true,
    "dest": "/tmp/yum.conf/",
    "src": "/etc/yum.repos.d"
}
eoc-node1 | CHANGED => {
    "changed": true,
    "dest": "/tmp/yum.conf/",
    "src": "/etc/yum.repos.d"
}
```