Lab: Building Custom Ansible Inventory

Introduction:

An inventory defines a collection of hosts that Ansible will manage. These hosts can also be assigned to groups, which can be managed collectively. Groups can contain child groups, and host can be members of multiple groups.

Host inventories can be defined in two different ways.

- A static host inventory can be defined by a text file.
- A dynamic host inventory can be generated by a script or other program as needed, using external information providers

Objectives:

- **Creating Static Inventory**
- **Verifying Static Inventory**
- **Verifying Communication**

1. Creating Static Inventory

1.1 Let's create a custom static inventory file named **org-infra** in the home directory.

Information about four managed hosts is listed in the below manifest. You will assign each host to multiple groups for management purposes.

```
# cat > ~/org-infra << EOF
[dev]
eoc-node1

[test]
eoc-node2

[prod]
eoc-node3
eoc-node2

[balancer]
eoc-node1

[webservers]
eoc-node3
EOF
```

Note: Inventory can be grouped in different ways as shown above.

2. Verifying Static Inventory

2.1 Let's list the customized hosts.

```
# ansible all --list-hosts
```

Output:

```
[admin@eoc-controller ~]$ ansible all --list-hosts
  hosts (3):
    eoc-node1
    eoc-node2
    eoc-node3
```

2.2 Once we setup the inventory file, we can use the ansible-inventory command to validate and obtain information about your Ansible inventory by using -i option which specify the inventory path.

```
# ansible-inventory -i org-infra --list
```

Note: (-i) option is to specify the inventory path

Output:

```
[admin@eoc-controller ~]$ ansible-inventory -i org-infra --list
{
    "_meta": {
        "hostvars": {}
    },
    "all": {
        "children": [
            "ungrouped",
            "dev",
            "test",
            "prod",
            "balancer",
            "webservers"
        ]
    },
    "balancer": {
        "hosts": [
            "eoc-node1"
        ]
    },
    "dev": {
        "hosts": [
            "eoc-node1"
        ]
    },
    "prod": {
        "hosts": [
            "eoc-node3",
            "eoc-node2"
        ]
    },
    "test": {
        "hosts": [
            "eoc-node2"
        ]
    },
    "webservers": {
        "hosts": [
            "eoc-node3"
        ]
    }
}
```

2.3 Let's list all managed hosts listed in the inventory file but are not part of a group.

```
# ansible ungrouped --list-hosts
```

Output:

```
[admin@eoc-controller ~]$ ansible ungrouped --list-hosts
[WARNING]: No hosts matched, nothing to do
  hosts (0):
```

Note: There are no ungrouped managed hosts in the inventory file.

2.4 Let's List all managed hosts listed in the **test** group.

```
# ansible test --list-hosts
```

Output:

```
[admin@eoc-controller ~]$ ansible test --list-hosts
  hosts (1):
    eoc-node2
```

2.5 Let's list all managed hosts listed in the **balancer** group.

```
# ansible balancer --list-hosts
```

Output:

```
[admin@eoc-controller ~]$ ansible balancer --list-hosts
  hosts (1):
    eoc-node1
```

2.6 Let's list all managed hosts listed in the **webserver** group.

```
# ansible webservers --list-hosts
```

Output:

```
[admin@eoc-controller ~]$ ansible webservers --list-hosts
  hosts (1):
    eoc-node3
```

3. Verifying Communication

3.1 Let's verify the communication between the manged host by running the below command and using the module called ping.

Note: Ansible modules are discrete units of code which can be used from the command line.

```
# ansible all -m ping
```

Output:

```
[admin@eoc-controller ~]$ansible all -m ping
eoc-node1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": false,
    "ping": "pong"
}
eoc-node3 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": false,
    "ping": "pong"
}
eoc-node2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": false,
    "ping": "pong"
}
```