Lab 4 – Inventory

Introduction

Ansible can work against multiple devices in your infrastructure. This could be switches, routers, firewalls, servers and more all managed at the same time. It is able to do this by using a file called an Inventory and allowing selection of certain hosts or groups of hosts.

Please refer to the **Ansible-Pod-Info.docx** file for information on connecting to your Ansible host.

Don't forget to change the XX in your inventory file based on your Pod information.

1. Inventory Structure

1.1 So the inventory file can be in a few different formats, the most popular are INI or YAML. Most of the time you will see the inventory file in INI format. See the 2 examples below to compare.

```
cd ~/ansible_labs/lab4-inventory
```

INI Inventory Example

```
localhost

[servers]

localhost

[switches]

switch1.localdomain.com

switch2.localdomain.com
```

```
[routers]
router1.localdomain.com
router2.localdomain.com
```

Notice the names in the [ ] that is the group name, you can give these any name you want to group together hosts. This way in playbooks you can specify switches as your host value to only apply it to the switches group for example.

YAML Inventory Example

```
all:
  hosts:
    localhost:
  children:
    servers:
      hosts:
        localhost:
    switches:
      hosts:
        switch1.localdomain.com:
        switch2.localdomain.com:
    routers:
      hosts:
        router1.localdomain.com:
        router2.localdomain.com:
```

Note: Systems can be in more than 1 group.

1.2 Build your own inventory file using the list of devices below, make sure the devices go under their respective groups of server, eos, nxos, router, and junos. If you get stuck ask the instructor for help.

The working directory will be **lab4-inventory**, a blank inventory file is already in the directory.

Your Device List - These are also in your reference sheet given to you.

```
server - localhost

eos - veos-pod-XX.localdomain

nxos - n9k-standalone-XX.localdomain

router - csr1000v-pod-XX.localdomain

junos - vsrx-pod-XX.localdomain
```

Now we should have a complete inventory file, if you have issues please check with the instructor for help. We will be using this inventory file throughout the rest of this lab.

1.3 Default Groups

In ansible there are 2 default groups by default **all** and **ungrouped**. **all** contains every host and **ungrouped** contains all hosts that don't have another group other than **all**

These 2 groups might not always appear in group listings but are assumed implicit and always there.

1.4 Dynamic Inventory Knowledge Check

What command could we run that we learned previously to see the inventory information and these 2 groups of all and ungrouped?

Try running this command against your new inventory file and see what you get. You may run into an error, please debug this issue and resolve it so that you end up getting the command to work.

```
{

    "_meta": {

        "hostvars": {}

    },

    "all": {

        "children": [

            "ungrouped"

        ]

    },

    "ungrouped": {}

}
```

Why aren’t we seeing our inventory that we created?

Fix the issue and get the data to show our new inventory file.

See the instructor if you get stuck or are unsure what command we could use here. Please reference Lab 1 for the command in question.

We should see something similar to the below output once you have the command working properly:

```
{
    "_meta": {
        "hostvars": {
            "csr1000v-pod-00.localdomain": {},
            "localhost": {},
            "n9k-standalone-01.localdomain": {},
            "veos-pod-00.localdomain": {},
            "vsrx-pod-00.localdomain": {}
        }
    },
    "all": {
        "children": [
            "eos",
            "junos",
            "nxos",
            "router",
            "server",
            "ungrouped"
        ]
    },
    "eos": {
        "hosts": [
            "veos-pod-00.localdomain"
        ]
    },
    "junos": {
        "hosts": [
            "vsrx-pod-00.localdomain"
        ]
    },
    "nxos": {
        "hosts": [
            "n9k-standalone-01.localdomain"
        ]
    },
    "router": {
        "hosts": [
            "csr1000v-pod-00.localdomain"
        ]
    },
    "server": {
        "hosts": [
            "localhost"
        ]
    },
    "ungrouped": {}
}
```

Version Control Commit

So now let’s commit our inventory file to our git repo. You might also have an ansible.cfg file which you can commit the same way.

```
git add inventory
git commit -m "adding our inventory file"
git push origin master
```

1.5 Multiple Inventory Files

You can also specify multiple inventory files by using a folder that can contain multiple subfolders of inventories and stuff.

On your own test creating a few folders under a top level folder along with separate inventory files in each and call the **ansible-inventory -i yourdir --list** and see the results. This could be useful depending on your setup and how you have your inventories structured.