Building the Inventory

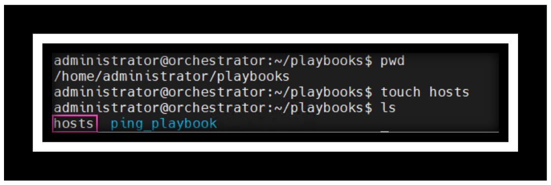
Introduction

In the first lab, we installed Ansible and learned how to use Ad-hoc commands and write a simple playbook to utilize built-in module to test connectivity to target hosts. In this lab, we will build playbook to perform useful tasks.

Exercise 1: Building your Inventory

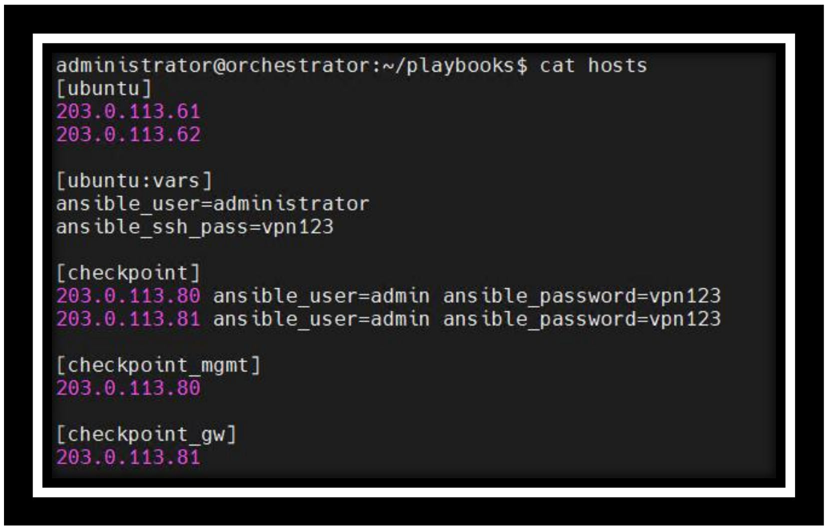
The automation lab we are using consists of multiple Check Point devices, windows VMS and Ubuntu orchestrator and target. We will built a static inventory file that contains all the managed nodes and we will use it for the rest of the tasks.

1. Under the playbooks directory, create a new file called hosts. This will be our inventory.



```
administrator@orchestrator:~/playbooks$ pwd
/home/administrator/playbooks
administrator@orchestrator:~/playbooks$ touch hosts
administrator@orchestrator:~/playbooks$ ls
hosts  ping_playbook
```

2. Add our targets and their variables using the following format.

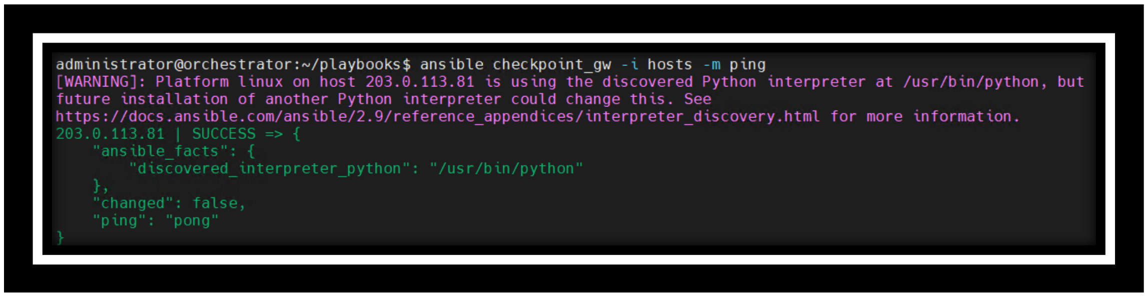


```
administrator@orchestrator:~/playbooks$ cat hosts
[ubuntu]
203.0.113.61
203.0.113.62

[ubuntu:vars]
ansible_user=administrator
ansible_ssh_pass=vpn123

[checkpoint]
203.0.113.80 ansible_user=admin ansible_password=vpn123
203.0.113.81 ansible_user=admin ansible_password=vpn123

[checkpoint_mgmt]
203.0.113.80

[checkpoint_gw]
203.0.113.81
```

- Notice that we have a group called ubuntu and it has variable assigned using the method we learned in lab 1 [ubuntu:vars].

- We also created a group to represent the Check Point devices. In this case, we assigned the variables directly inside the group.
- In case we would like to apply changes to the management or gateway only, we created separate groups, one for the gateway and one for the management.

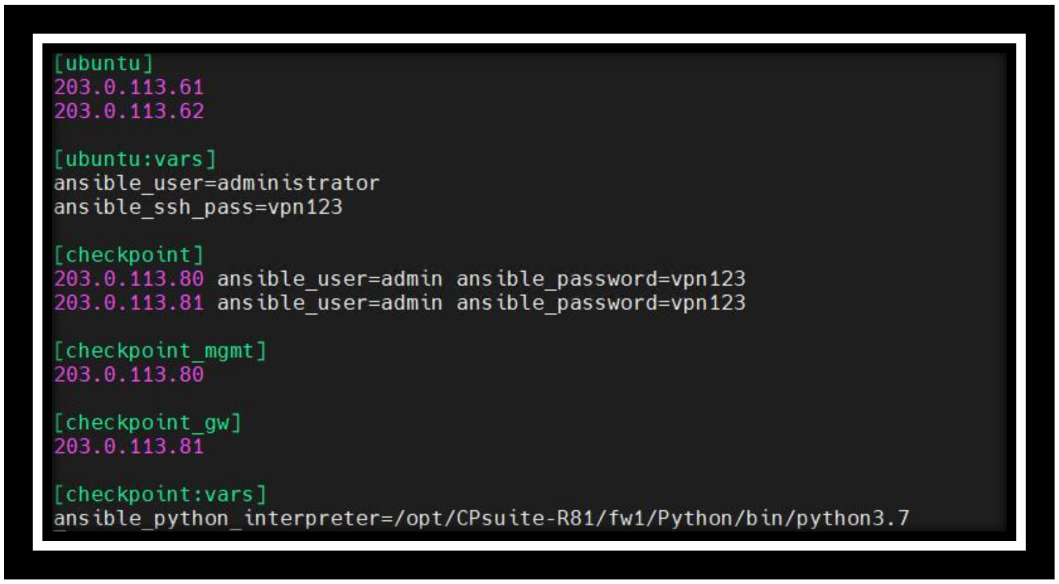
3. Run the ping module as ad-hoc command to test connectivity to the gateway: ansible checkpoint_gw -i hosts -m ping



```
administrator@orchestrator:~/playbooks$ ansible checkpoint_gw -i hosts -m ping
[WARNING]: Platform linux on host 203.0.113.81 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.
203.0.113.81 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
```

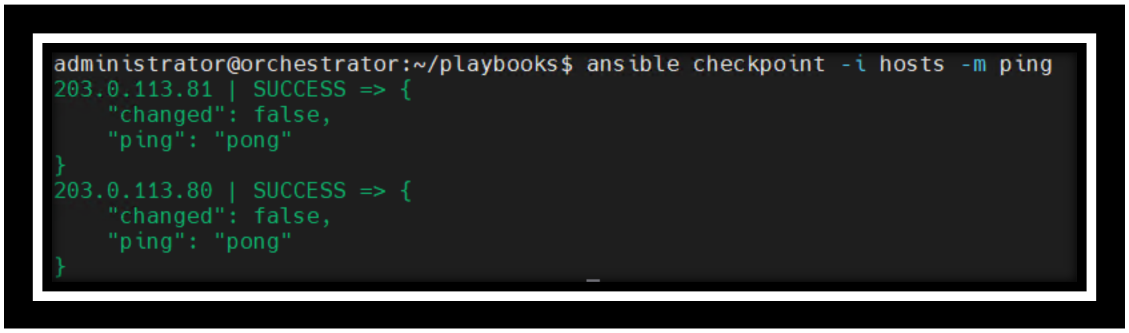
- Note that the connectivity check worked however, there is a warning regarding the python path.
- As this is R81, the python location should be /opt/CPsuite-R81/fw1/Python/bin/python3.7.
- Older Gateways use Python 2.7 under such as /opt/CPsuite-R80.20/fw1/Python/bin/python2.7.
- To specify the python interpreter in Python, use the variable ansible_python_interpreter

4. Edit the inventory file and add the python interpreter as a variable for all Check Point devices.



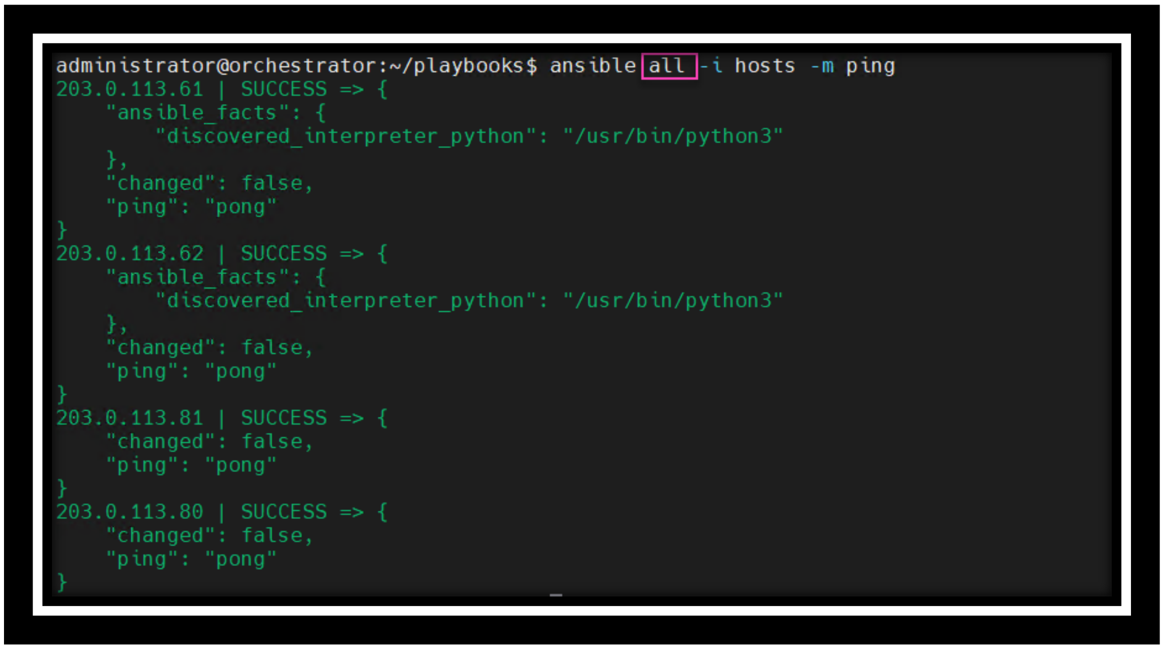
```
[ubuntu]
203.0.113.61
203.0.113.62

[ubuntu:vars]
ansible_user=administrator
ansible_ssh_pass=vpn123

[checkpoint]
203.0.113.80 ansible_user=admin ansible_password=vpn123
203.0.113.81 ansible_user=admin ansible_password=vpn123

[checkpoint_mgmt]
203.0.113.80

[checkpoint_gw]
203.0.113.81

[checkpoint:vars]
ansible_python_interpreter=/opt/CPsuite-R81/fw1/Python/bin/python3.7
```

5. Run the test again but this time run it against all Check Point hosts (use the group checkpoint).



```
administrator@orchestrator:~/playbooks$ ansible checkpoint -i hosts -m ping
203.0.113.81 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
203.0.113.80 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

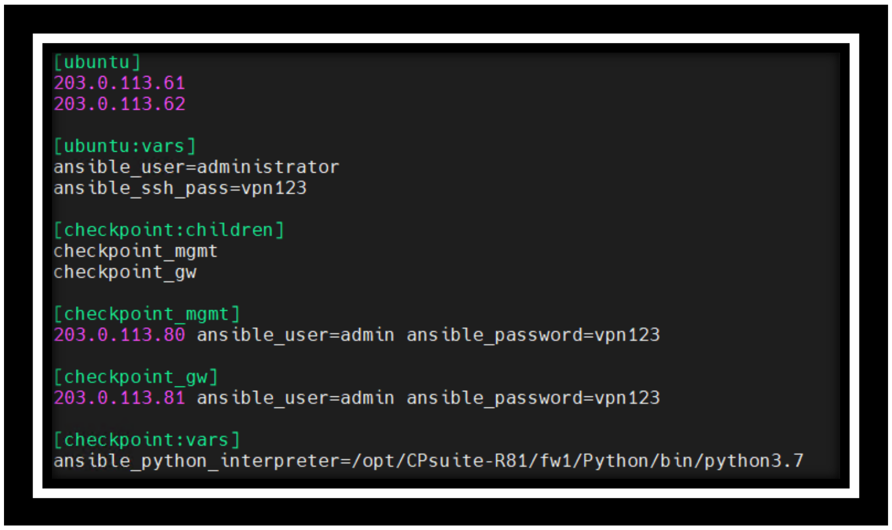
6. Now that we have many groups and hosts in the inventory, try to run a connectivity check against all hosts.



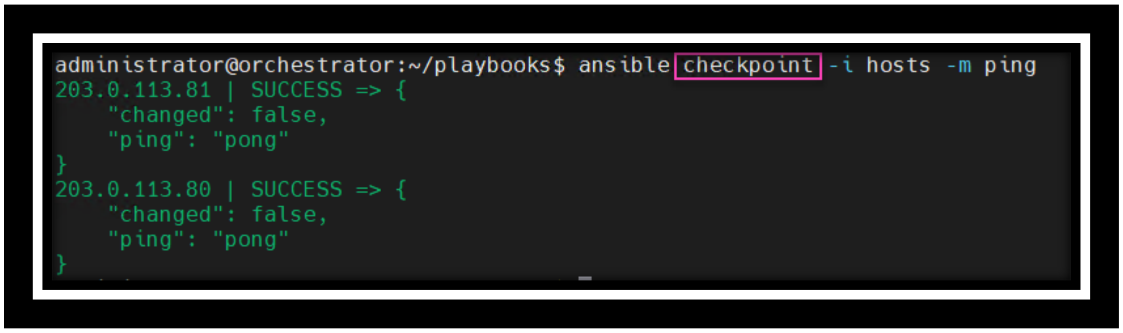
```
administrator@orchestrator:~/playbooks$ ansible all -i hosts -m ping
203.0.113.61 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
203.0.113.62 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
203.0.113.81 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
203.0.113.80 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

- Notice that we typed the addresses for the Check Point devices twice, once in the group checkpoint and later in the separate groups. The better approach is to create subgroups.
- To achieve that, we can use the children keyword.
- The format is [group_name:children]

7. Change the inventory file to form checkpoint group out of the two sub groups.


```
[ubuntu]
203.0.113.61
203.0.113.62

[ubuntu:vars]
ansible_user=administrator
ansible_ssh_pass=vpn123

[checkpoint:children]
checkpoint_mgmt
checkpoint_gw

[checkpoint_mgmt]
203.0.113.80 ansible_user=admin ansible_password=vpn123

[checkpoint_gw]
203.0.113.81 ansible_user=admin ansible_password=vpn123

[checkpoint:vars]
ansible_python_interpreter=/opt/CPsuite-R81/fw1/Python/bin/python3.7
```

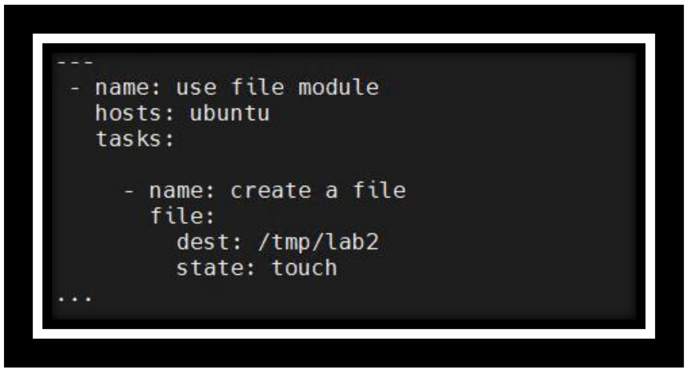
8. Run the previous command to verify that the changes were applied to the sub groups.


```
administrator@orchestrator:~/playbooks$ ansible checkpoint -i hosts -m ping
203.0.113.81 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
203.0.113.80 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

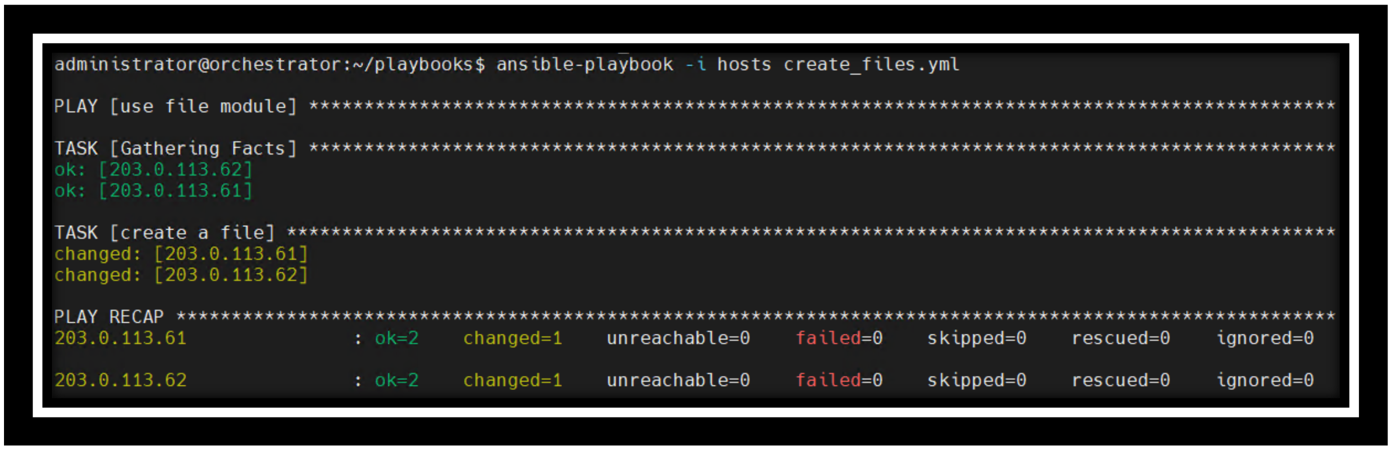
Exercise 2: Using Custom Inventory in Playbooks

We will now use ansible playbooks to run operations on the target hosts and apply the changes to specific hosts in our inventory.

1. Create a simple playbook to create a simple file on the Ubuntu machines.

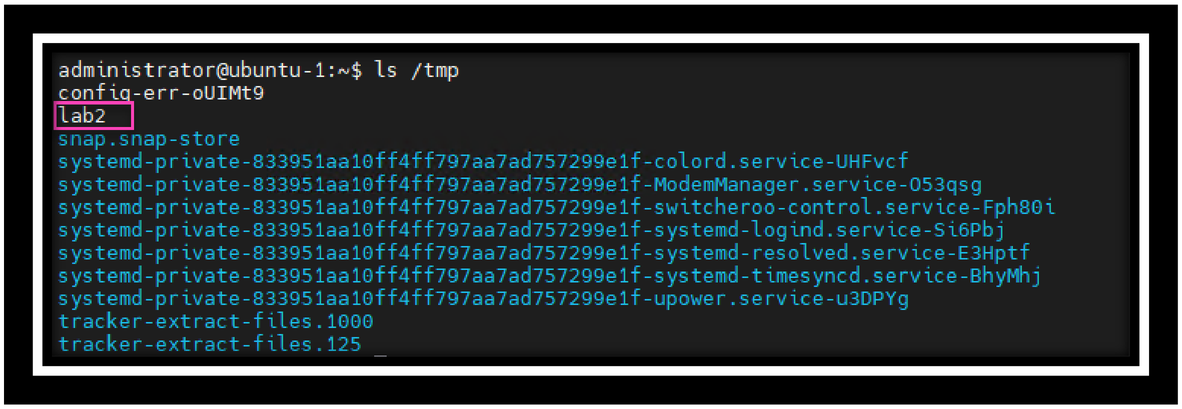

```
---
 - name: use file module
   hosts: ubuntu
   tasks:

     - name: create a file
       file:
         dest: /tmp/lab2
         state: touch
...
```

2. Run the playbook using the command ansible-playbook –i hosts create_file.yml.



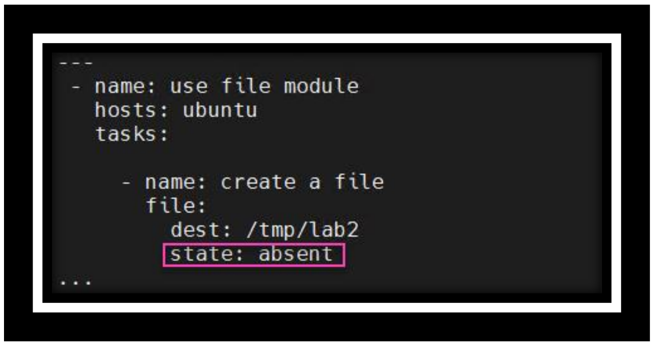
```
administrator@orchestrator:~/playbooks$ ansible-playbook -i hosts create_files.yml

PLAY [use file module] *********************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [203.0.113.62]
ok: [203.0.113.61]

TASK [create a file] ***********************************************************************************************
changed: [203.0.113.61]
changed: [203.0.113.62]

PLAY RECAP *********************************************************************************************************
203.0.113.61               : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

203.0.113.62               : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

3. Connect to the Ubuntu nodes and verify that the file was created.



```
administrator@ubuntu-1:~$ ls /tmp
config-err-oUIMt9
lab2
snap.snap-store
systemd-private-833951aa10ff4ff797aa7ad757299e1f-colord.service-UHFvcf
systemd-private-833951aa10ff4ff797aa7ad757299e1f-ModemManager.service-053qsg
systemd-private-833951aa10ff4ff797aa7ad757299e1f-switcheroo-control.service-Fph80i
systemd-private-833951aa10ff4ff797aa7ad757299e1f-systemd-logind.service-Si6Pbj
systemd-private-833951aa10ff4ff797aa7ad757299e1f-systemd-resolved.service-E3Hptf
systemd-private-833951aa10ff4ff797aa7ad757299e1f-systemd-timesyncd.service-BhyMhj
systemd-private-833951aa10ff4ff797aa7ad757299e1f-upower.service-u3DPYg
tracker-extract-files.1000
tracker-extract-files.125 _
```

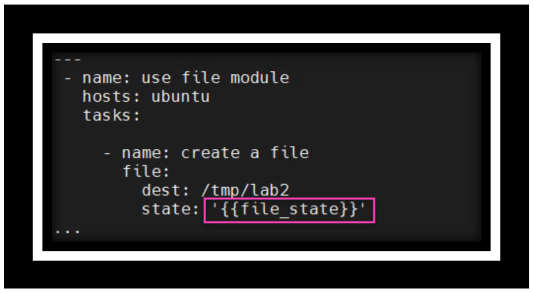
4. To delte the file, we need to change the state from touch to absent. Refer to the module documentations for more details on how to control the module options. https://docs.ansible.com/ansible/latest/collections/ansible/builtin/file_module.html



```
---
 - name: use file module
   hosts: ubuntu
   tasks:

     - name: create a file
       file:
         dest: /tmp/lab2
         state: absent
...
```

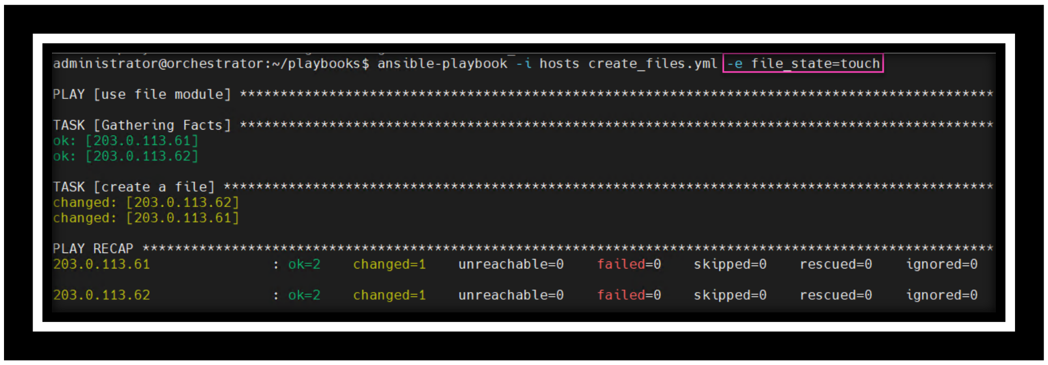
5. Run the playbook again and make sure the files were deleted from both Ubuntu nodes.

6. It is recommended to make the file state as a variable as this entry will change. Change the playbook to have the state entered as a variable.



```
---
 - name: use file module
   hosts: ubuntu
   tasks:

     - name: create a file
       file:
         dest: /tmp/lab2
         state: '{{file_state}}'
...
```

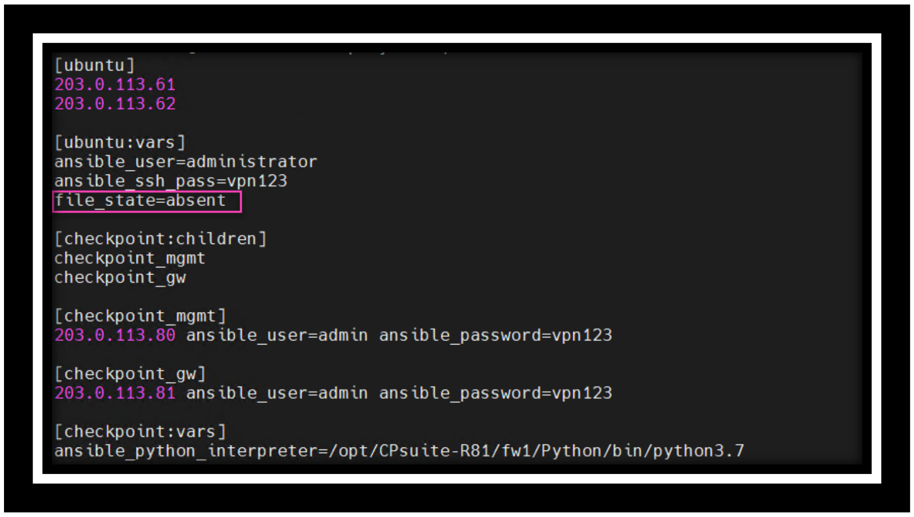
7. Now that the state can be provided as a variable, we can provide it in the hosts file or directly when we run the playbook from CLI. Use the following command to create the files again:
ansible-playbook -i hosts create_files.yml -e file_state=touch



```
administrator@orchestrator:~/playbooks$ ansible-playbook -i hosts create_files.yml -e file_state=touch

PLAY [use file module] ************************************************************************************************

TASK [Gathering Facts] ************************************************************************************************
ok: [203.0.113.61]
ok: [203.0.113.62]

TASK [create a file] **************************************************************************************************
changed: [203.0.113.62]
changed: [203.0.113.61]

PLAY RECAP ************************************************************************************************************
203.0.113.61               : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

203.0.113.62               : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

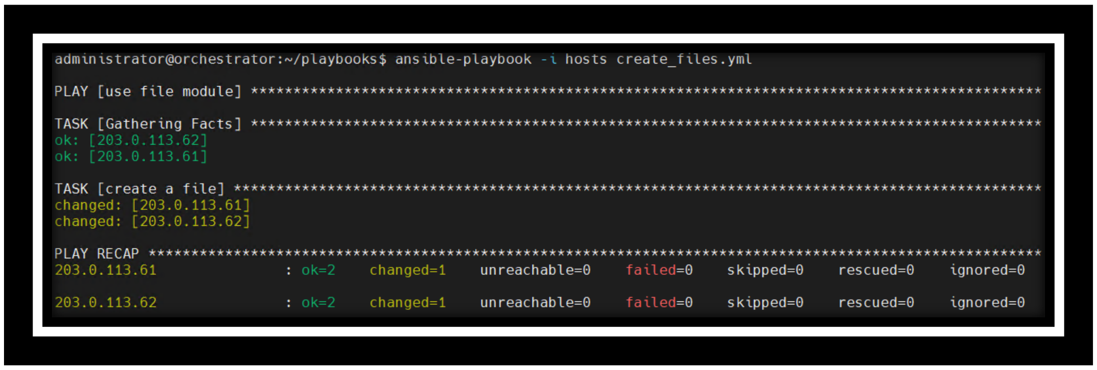
- Notice that we provided the variable value with –e flag.

8. Try the other method and provide the variable value ion the inventory file.



```
[ubuntu]
203.0.113.61
203.0.113.62

[ubuntu:vars]
ansible_user=administrator
ansible_ssh_pass=vpn123
file_state=absent

[checkpoint:children]
checkpoint_mgmt
checkpoint_gw

[checkpoint_mgmt]
203.0.113.80 ansible_user=admin ansible_password=vpn123

[checkpoint_gw]
203.0.113.81 ansible_user=admin ansible_password=vpn123

[checkpoint:vars]
ansible_python_interpreter=/opt/CPsuite-R81/fw1/Python/bin/python3.7
```

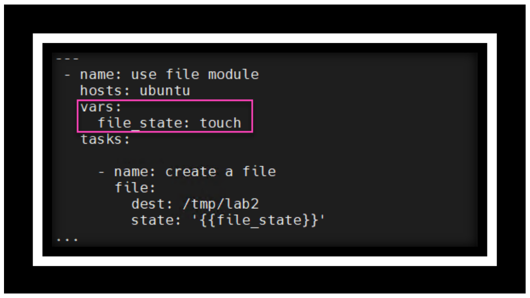
9. Run the playbook and make sure the files were deleted.



```
administrator@orchestrator:~/playbooks$ ansible-playbook -i hosts create_files.yml

PLAY [use file module] *********************************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [203.0.113.62]
ok: [203.0.113.61]

TASK [create a file] ***********************************************************************************
changed: [203.0.113.61]
changed: [203.0.113.62]

PLAY RECAP *********************************************************************************************
203.0.113.61               : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

203.0.113.62               : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

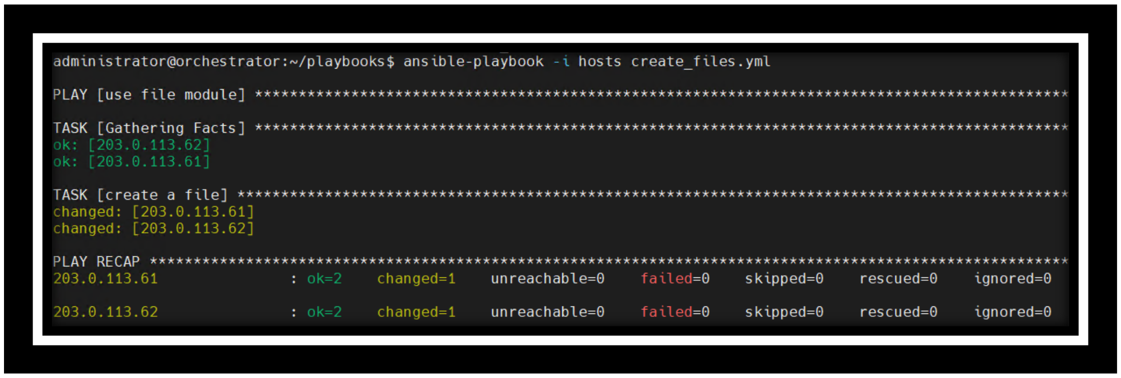
- Notice that we did not provide any variables in the command itself however, ansible discovered the value from the inventory file.

10. Remove the previous variable from the inventory file. Now create the variable inside the playbook using the format below.



```
---
- name: use file module
  hosts: ubuntu
  vars:
    file_state: touch
  tasks:

    - name: create a file
      file:
        dest: /tmp/lab2
        state: '{{file_state}}'
...
```

11. Run the playbook and confirm the file was created successfully.



```
administrator@orchestrator:~/playbooks$ ansible-playbook -i hosts create_files.yml

PLAY [use file module] *********************************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [203.0.113.62]
ok: [203.0.113.61]

TASK [create a file] ***********************************************************************************
changed: [203.0.113.61]
changed: [203.0.113.62]

PLAY RECAP *********************************************************************************************
203.0.113.61               : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

203.0.113.62               : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

12. Change the state to absent and run the playbook to clean up your environment.

End of Lab 2