Lab: Controlling playbook execution

Introduction:

FORKS: Maximum number of simultaneous connections Ansible made on each Task.

SERIAL: Decides the number of nodes process in each task in a single run.

Objectives:

Effects of different serial and forks directives on how a play is processed by Ansible.

Log in to **ansi-master** as **root** user and password as **linux.**

1. Change to the **./parallelism** directory.

```
# cd
# mkdir parallelism
# cd parallelism
```

1.1 Examine the contents of the **ansible.cfg file**. Note that the inventory file is set to inventory. Note also that the **forks parameter** is set to **4.**

```
# cat > ansible.cfg <<EOF
[defaults]
inventory = ./inventory
forks=4

[privilege_escalation]
become=true
become_method=sudo
become_user=root
become_ask_pass=false
EOF
```

1.2 Examine the contents of the inventory file. Note that it contains a host group, **webservers**, which contains **three hosts**.

```
 $ cat > inventory << EOF
 [webservers]
 ansi-node1
 ansi-node2
 ansi-node3
EOF
```

1.3 Lets create **playbook.yml** file. The playbook executes on the webservers host group, ensures that the latest **httpd package** is installed and that the httpd service is enabled and started.

```
$ cat > playbook.yml <<EOF
---
- name: Update web server
  hosts: webservers
  become: yes

  tasks:
    - name: Latest httpd package installed
      dnf:
        name: httpd
        state: latest
      notify:
        - Restart httpd

  handlers:
    - name: Restart httpd
      service:
        name: httpd
        enabled: yes
        state: restarted
EOF
```

1.4 Finally examine the contents of the **remove_apache.yml** file. The playbook executes on the webservers host group, ensures that the httpd service is **disabled** and **stopped**, and then ensures that the httpd package is not installed.

```
$ cat > removeapache.yml << EOF
---
- hosts: webservers
  tasks:
    - service:
        name: httpd
        enabled: no
        state: stopped
    - dnf:
        name: httpd
        state: absent
EOF
```

2. Let's Execute the **playbook.yml** playbook using **time** command to determine how long it takes for the playbook to run. Watch the playbook as it runs. Note how ansible performs each task on all three hosts at the same time.

```
$ time ansible-playbook playbook.yml
```

Output:

```
[root@ansi-master parallelism]# time ansible-playbook playbook.yml

PLAY [Update web server] ********************************************************************************************

TASK [Gathering Facts] **********************************************************************************************
ok: [ansi-node3]
ok: [ansi-node2]
ok: [ansi-node1]

TASK [Latest httpd package installed] *******************************************************************************
changed: [ansi-node3]
changed: [ansi-node1]
changed: [ansi-node2]

RUNNING HANDLER [Restart httpd] *************************************************************************************
changed: [ansi-node3]
changed: [ansi-node2]
changed: [ansi-node1]

PLAY RECAP **********************************************************************************************************
ansi-node1                 : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ansi-node2                 : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ansi-node3                 : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0


real    0m6.077s
user    0m1.459s
sys     0m0.556s
```

3. Execute the **remove_apache.yml** playbook to stop and disable the httpd service and to remove the httpd package.

```
# time ansible-playbook removeapache.yml
```

Output:

```
[root@ansi-master parallelism]# time ansible-playbook removeapache.yml

PLAY [webservers] ***************************************************************************************************

TASK [Gathering Facts] **********************************************************************************************
ok: [ansi-node3]
ok: [ansi-node2]
ok: [ansi-node1]

TASK [service] ******************************************************************************************************
changed: [ansi-node2]
changed: [ansi-node1]
changed: [ansi-node3]

TASK [dnf] **********************************************************************************************************
changed: [ansi-node2]
changed: [ansi-node1]
changed: [ansi-node3]

PLAY RECAP **********************************************************************************************************
ansi-node1                 : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ansi-node2                 : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ansi-node3                 : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0


real    0m10.504s
user    0m1.619s
sys     0m0.626s
```

4. Change the value of the **fork parameter** to **1** in **ansible.cfg.**

```
# cat > ansible.cfg <<EOF
[defaults]
inventory = ./inventory
forks=1

[privilege_escalation]
become=true
become_method=sudo
become_user=root
become_ask_pass=false
EOF
```

5. **Re-execute** the **playbook.yml** playbook using time command to determine how long it takes for the playbook to run.

```
# time ansible-playbook playbook.yml
```

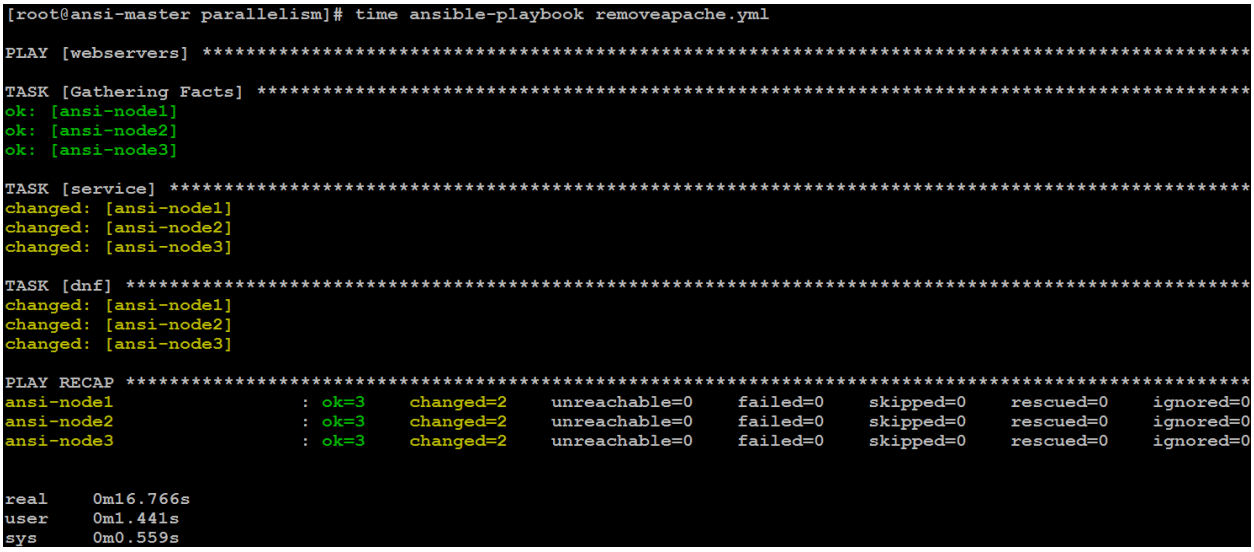
Output:

```
[root@ansi-master parallelism]# time ansible-playbook playbook.yml

PLAY [Update web server] ********************************************************************************************

TASK [Gathering Facts] **********************************************************************************************
ok: [ansi-node1]
ok: [ansi-node2]
ok: [ansi-node3]

TASK [Latest httpd package installed] *******************************************************************************
changed: [ansi-node1]
changed: [ansi-node2]
changed: [ansi-node3]

RUNNING HANDLER [Restart httpd] *************************************************************************************
changed: [ansi-node1]
changed: [ansi-node2]
changed: [ansi-node3]

PLAY RECAP **********************************************************************************************************
ansi-node1                 : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ansi-node2                 : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ansi-node3                 : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0


real    0m16.108s
user    0m1.463s
sys     0m0.527s
```

6. Execute the **remove_apache.yml** playbook to stop and disable the httpd service and to remove the httpd package.

```
# time ansible-playbook removeapache.yml
```

Output:



7. Set the value of the **forks parameter** to **2** in **ansible.cfg.**

```
# cat > ansible.cfg <<EOF
[defaults]
inventory = ./inventory
forks=2

[privilege_escalation]
become=true
become_method=sudo
become_user=root
become_ask_pass=false
EOF
```

8. Add the following serial parameter to the play in the playbook.yml playbook so that the play only executes on two hosts at a time.

```
# cat > playbook.yml <<EOF
---
- name: Update web server
  hosts: webservers
  serial: 1
  tasks:
    - name: Latest httpd package installed
      dnf:
        name: httpd
        state: latest
      notify:
        - Restart httpd

  handlers:
    - name: Restart httpd
      service:
        name: httpd
        enabled: yes
        state: restarted
EOF
```

9. Re-execute the playbook.yml playbook. Watch the playbook as it runs. Note how Ansible executes the entire play on just two hosts before re-executing the play on the two remaining hosts.

```
# time ansible-playbook playbook.yml
```

Output:

```
[root@ansi-master parallelism]# time ansible-playbook playbook.yml

PLAY [Update web server] ********************************************************************************************

TASK [Gathering Facts] **********************************************************************************************
ok: [ansi-node1]

TASK [Latest httpd package installed] *******************************************************************************
changed: [ansi-node1]

RUNNING HANDLER [Restart httpd] *************************************************************************************
changed: [ansi-node1]

PLAY [Update web server] ********************************************************************************************

TASK [Gathering Facts] **********************************************************************************************
ok: [ansi-node2]

TASK [Latest httpd package installed] *******************************************************************************
changed: [ansi-node2]

RUNNING HANDLER [Restart httpd] *************************************************************************************
changed: [ansi-node2]

PLAY [Update web server] ********************************************************************************************

TASK [Gathering Facts] **********************************************************************************************
ok: [ansi-node3]

TASK [Latest httpd package installed] *******************************************************************************
changed: [ansi-node3]

RUNNING HANDLER [Restart httpd] *************************************************************************************
changed: [ansi-node3]

PLAY RECAP **********************************************************************************************************
ansi-node1                 : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ansi-node2                 : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ansi-node3                 : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

10. Execute the remove_apache.yml playbook to stop and disable the httpd service and to remove the httpd package.

```
$ time ansible-playbook removeapache.yml -i inventory
```

Output:

```
[root@ansi-master parallelism]# time ansible-playbook removeapache.yml -i inventory

PLAY [webservers] ***************************************************************************************************

TASK [Gathering Facts] **********************************************************************************************
ok: [ansi-node2]
ok: [ansi-node1]
ok: [ansi-node3]

TASK [service] ******************************************************************************************************
changed: [ansi-node2]
changed: [ansi-node1]
changed: [ansi-node3]

TASK [dnf] **********************************************************************************************************
changed: [ansi-node2]
changed: [ansi-node1]
changed: [ansi-node3]

PLAY RECAP **********************************************************************************************************
ansi-node1                 : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ansi-node2                 : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ansi-node3                 : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0


real    0m11.019s
user    0m1.393s
sys     0m0.534s
```

11. Set the value of the **forks** parameter back to **4** in **ansible.cfg.**

```
$ cat > ansible.cfg <<EOF
[defaults]
inventory = ./inventory
forks=4

[privilege_escalation]
become=true
become_method=sudo
become_user=root
become_ask_pass=false
EOF
```

12. Set the serial parameter in the playbook.yml playbook to 3.

```
# cat > playbook.yml <<EOF
---
- name: Update web server
  hosts: webservers
  serial: 3
  tasks:
    - name: Latest httpd package installed
      dnf:
        name: httpd
        state: latest
      notify:
        - Restart httpd

  handlers:
    - name: Restart httpd
      service:
        name: httpd
        enabled: yes
        state: restarted
EOF
```

13. Re-execute the playbook.yml playbook. Ansible executes the entire play on just three hosts and then re-executes the play on the one remaining host.

```
$ time ansible-playbook playbook.yml -i inventory
```

Output:

```
[root@ansi-master parallelism]# time ansible-playbook playbook.yml -i inventory

PLAY [Update web server] ********************************************************************************************

TASK [Gathering Facts] **********************************************************************************************
ok: [ansi-node1]
ok: [ansi-node3]
ok: [ansi-node2]

TASK [Latest httpd package installed] *******************************************************************************
ok: [ansi-node2]
ok: [ansi-node1]
ok: [ansi-node3]

PLAY RECAP **********************************************************************************************************
ansi-node1                 : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ansi-node2                 : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ansi-node3                 : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0


real    0m4.233s
user    0m1.024s
sys     0m0.403s
```