Lab 3: Writing and Running Playbooks

Introduction:

An Ansible playbook is a file where users write Ansible code, an organized collection of scripts defining the work of a server configuration. They describe a set of steps in a general IT process or a policy for your remote systems to enforce.

Playbooks consist of one or more plays run in a particular order. A play is an ordered set of tasks run against hosts chosen from your inventory. Plays define the work to be done. Each play contains a set of hosts to configure, and a list of tasks to be executed. There are no standardized plays; an administrator must write each play.

Playbooks use YAML, a human-readable data serialization language. YAML is a recursive acronym that stands for "YAML Ain't Markup Language."

Objectives:

- Writing playbook to install httpd service on ansi-node1
- Deploying httpd service on ansi-node1(Host)
- Running Ad-hoc Commands on multiple host

In this Lab, you can able to write a playbook using basic YAML syntax and Ansible Playbook structure and successfully run it with ansible-playbook command.

1. Login into the Control node **ansi-master** as **root** user with password as **linux.**

1.1 Create a Directory by the name playbook and change directory to playbook.

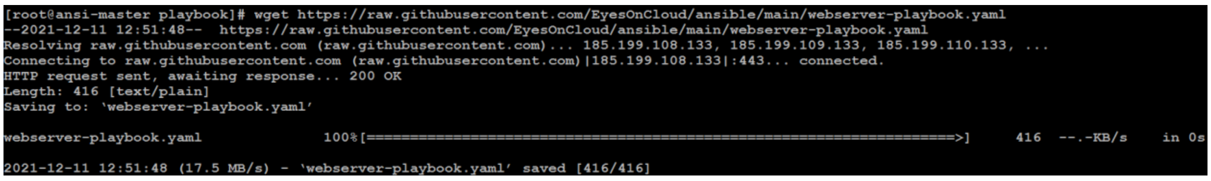
```
# mkdir playbook
# cd playbook
```

1.2 Use a text editor to create a new playbook called webserver-playbook.yaml. Start writing a play that targets the hosts in the web host group.
The playbook should use tasks to ensure that following conditions are met on managed hosts.The httpd package is present, using the yum module
The local **/files/index.html** file is copied to **/var/www/html/index.html** on ansi-node1 managed host using the copy module

Let's download the manifest

```
# wget
https://raw.githubusercontent.com/EyesOnCloud/ansible/main/w
ebserver-playbook.yaml
```

Output:

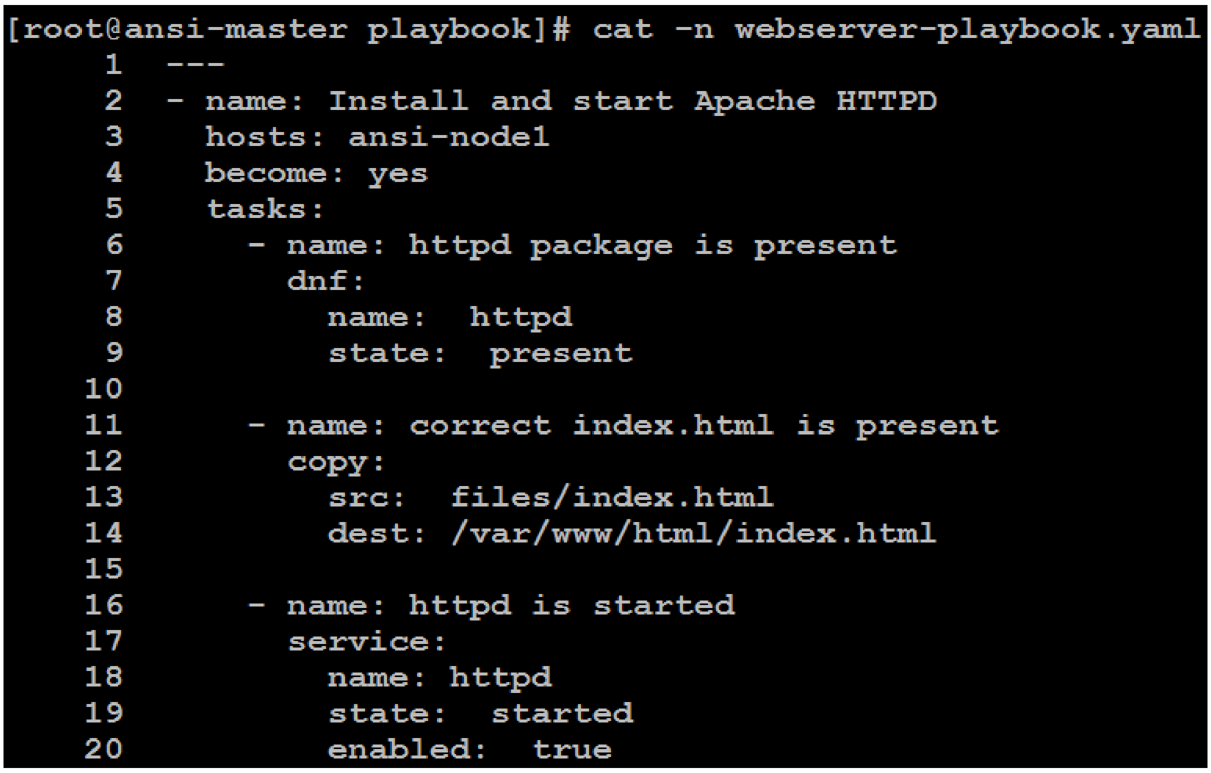


```
[root@ansi-master playbook]# wget https://raw.githubusercontent.com/EyesOnCloud/ansible/main/webserver-playbook.yaml
--2021-12-11 12:51:48--  https://raw.githubusercontent.com/EyesOnCloud/ansible/main/webserver-playbook.yaml
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 416 [text/plain]
Saving to: ‘webserver-playbook.yaml’

webserver-playbook.yaml            100%[===================================================================>]     416  --.-KB/s    in 0s

2021-12-11 12:51:48 (17.5 MB/s) - ‘webserver-playbook.yaml’ saved [416/416]
```

Lets view the manifest

```
# cat -n webserver-playbook.yaml
```

Output:



```
[root@ansi-master playbook]# cat -n webserver-playbook.yaml
     1  ---
     2  - name: Install and start Apache HTTPD
     3    hosts: ansi-node1
     4    become: yes
     5    tasks:
     6      - name: httpd package is present
     7        dnf:
     8          name:  httpd
     9          state:  present
    10
    11      - name: correct index.html is present
    12        copy:
    13          src:  files/index.html
    14          dest: /var/www/html/index.html
    15
    16      - name: httpd is started
    17        service:
    18          name: httpd
    19          state:  started
    20          enabled:  true
```

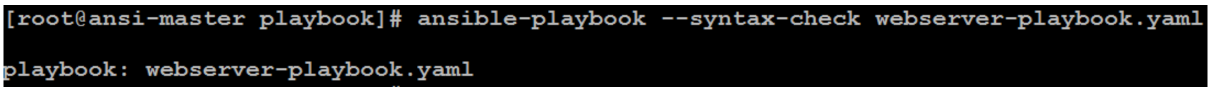
1.3 Create The local **files/index.html** file.

```
# mkdir files
# cd files
# cat > index.html <<EOF
Welcome to Ansible Class
EOF
# cd ..
```

1.4 Before running your playbook run the ansible-playbook –system-check site.yml command to verify that its syntax is correct.

```
# ansible-playbook --syntax-check webserver-playbook.yaml
```

Output:

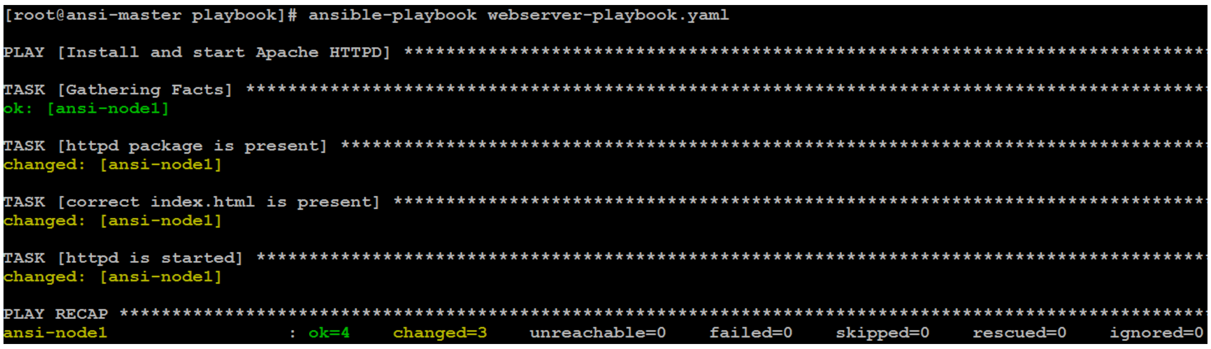


```
[root@ansi-master playbook]# ansible-playbook --syntax-check webserver-playbook.yaml

playbook: webserver-playbook.yaml
```

1.5 Run your playbook and check the output generated to ensure that all tasks completed successfully.

```
# ansible-playbook webserver-playbook.yaml
```

Output:

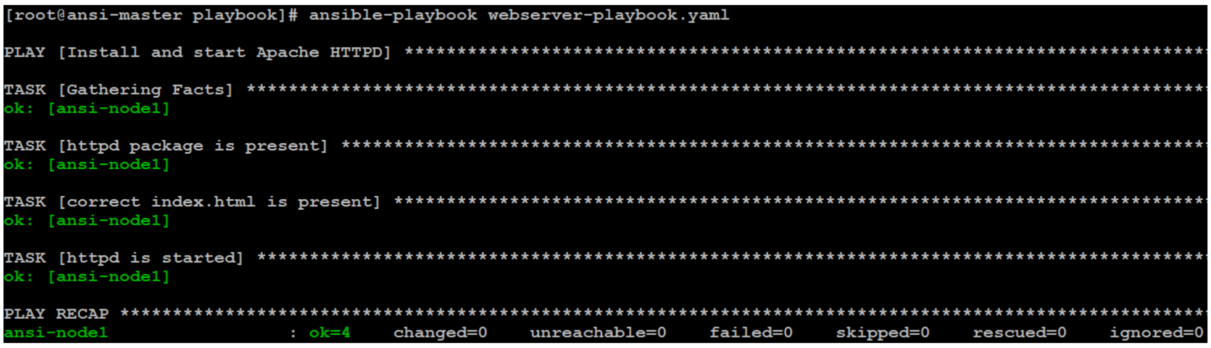


```
[root@ansi-master playbook]# ansible-playbook webserver-playbook.yaml

PLAY [Install and start Apache HTTPD] ************************************************************************

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

TASK [httpd package is present] ******************************************************************************
changed: [ansi-node1]

TASK [correct index.html is present] *************************************************************************
changed: [ansi-node1]

TASK [httpd is started] **************************************************************************************
changed: [ansi-node1]

PLAY RECAP ***************************************************************************************************
ansi-node1                 : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

1.6 If all went well, you should be able to run the playbooks a second time and see all tasks complete with no changes to the managed hosts.

```
# ansible-playbook webserver-playbook.yaml
```

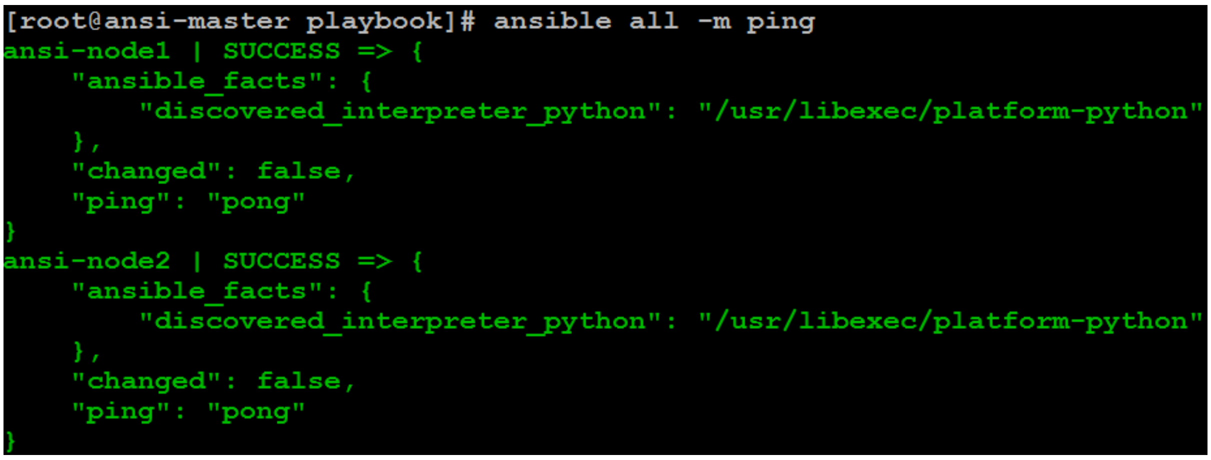
Output:



```
[root@ansi-master playbook]# ansible-playbook webserver-playbook.yaml

PLAY [Install and start Apache HTTPD] ************************************************************************

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

TASK [httpd package is present] ******************************************************************************
ok: [ansi-node1]

TASK [correct index.html is present] *************************************************************************
ok: [ansi-node1]

TASK [httpd is started] **************************************************************************************
ok: [ansi-node1]

PLAY RECAP ***************************************************************************************************
ansi-node1                 : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

1.7 To Check the Connectiviy to all the managed Nodes:

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

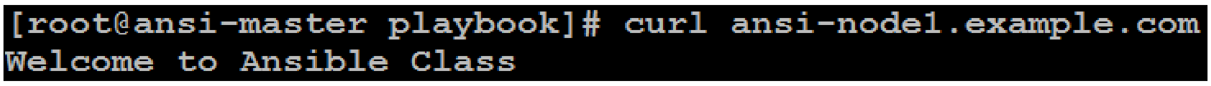


```
[root@ansi-master playbook]# ansible all -m ping
ansi-node1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": false,
    "ping": "pong"
}
ansi-node2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": false,
    "ping": "pong"
}
```

1.8 Use the curl command to verify that ansi-node1 is configured as an HTTPD server

```
# curl ansi-node1.example.com
```

Output:



```
[root@ansi-master playbook]# curl ansi-node1.example.com
Welcome to Ansible Class
```