Lab: Ansible Roles

Introduction:
Roles provide a framework for fully **independent** or **interdependent** collections of **files**, **tasks**, **templates**, **variables**, and **modules**.

The role is the primary mechanism for **breaking** a playbook into **multiple files**. This simplifies **writing complex** playbooks and makes them **easier to reuse**. The breaking of the playbook allows you to break the playbook into reusable components.

Each role is limited to a **particular functionality** or desired output, with all the necessary steps to provide that result either within the same role itself or in other roles listed as dependencies.

Roles are **not playbooks**. Roles are small **functionality** that can be used within the **playbooks independently**. Roles have **no specific** setting for which **hosts** the role will apply.

Objective:

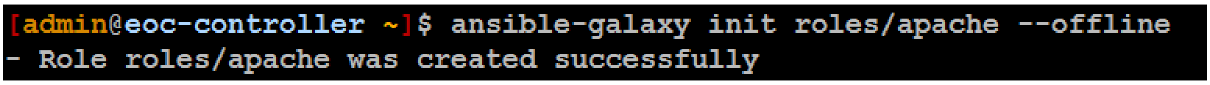
- **Writing role with ansible-galaxy**
- **Installing httpd package**
- **Writing a Play in configuration management**
- **Starting httpd service**
- **Breaking the codes**

1. Writing role with ansible-galaxy

1.1 The first step in creating a role is creating its directory structure. In order to create the base directory structure, we're going to use a tool called **ansible-galaxy**:

```
# ansible-galaxy init roles/apache --offline
```

Output:

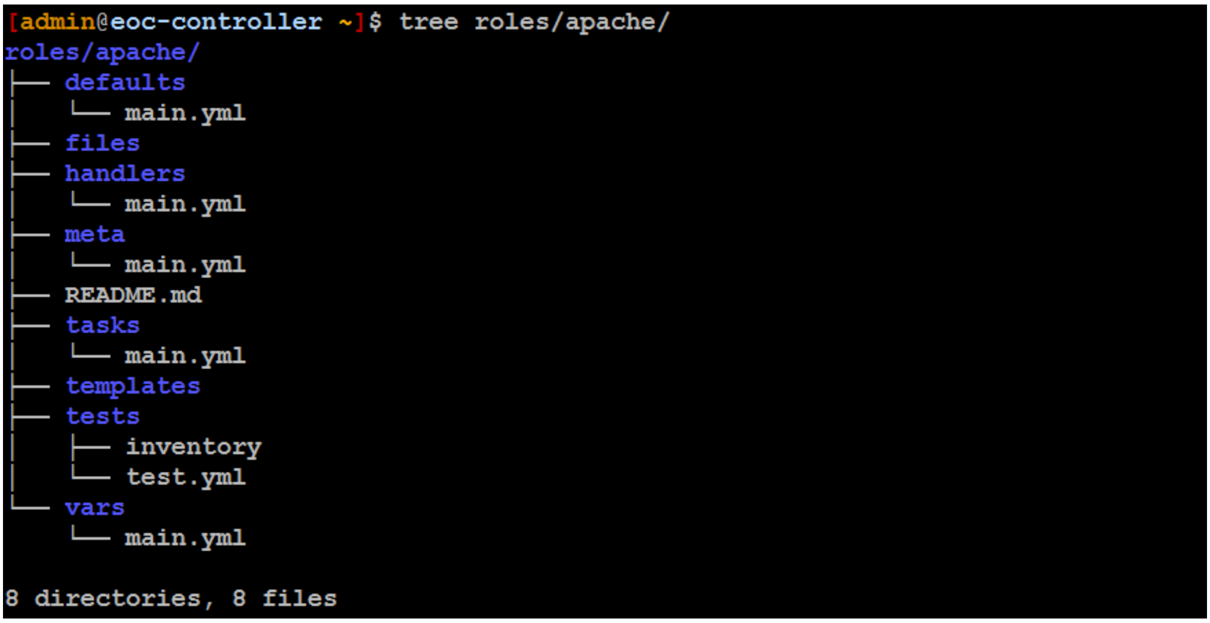


```
[admin@eoc-controller ~]$ ansible-galaxy init roles/apache --offline
- Role roles/apache was created successfully
```

1.2 This command will create an apache directory with the following structure let's use the tree command to view.

```
# tree roles/apache/
```

Output:



```
[admin@eoc-controller ~]$ tree roles/apache/
roles/apache/
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── README.md
├── tasks
│   └── main.yml
├── templates
├── tests
│   ├── inventory
│   └── test.yml
└── vars
    └── main.yml

8 directories, 8 files
```

Note: A role's directory structure consists of defaults, vars, files, handlers, meta, tasks,tests and templates.

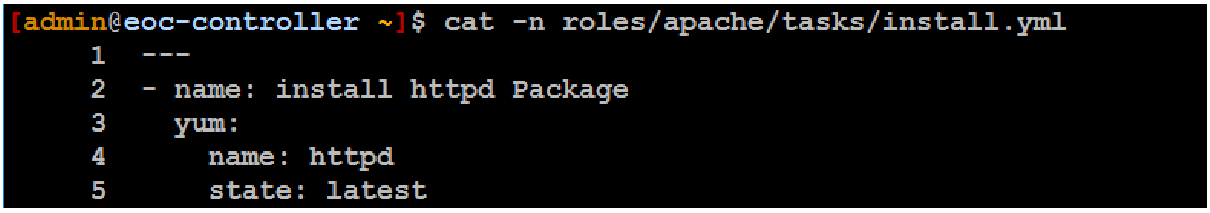
2. Installing httpd package

2.1 Let's create the **install.yml** which will install the httpd package in the task directory

Path: - **roles/apache/tasks/install.yml**

```
# cat -n roles/apache/tasks/install.yml
```

Output:



```
[admin@eoc-controller ~]$ cat -n roles/apache/tasks/install.yml
     1  ---
     2  - name: install httpd Package
     3    yum:
     4      name: httpd
     5      state: latest
```

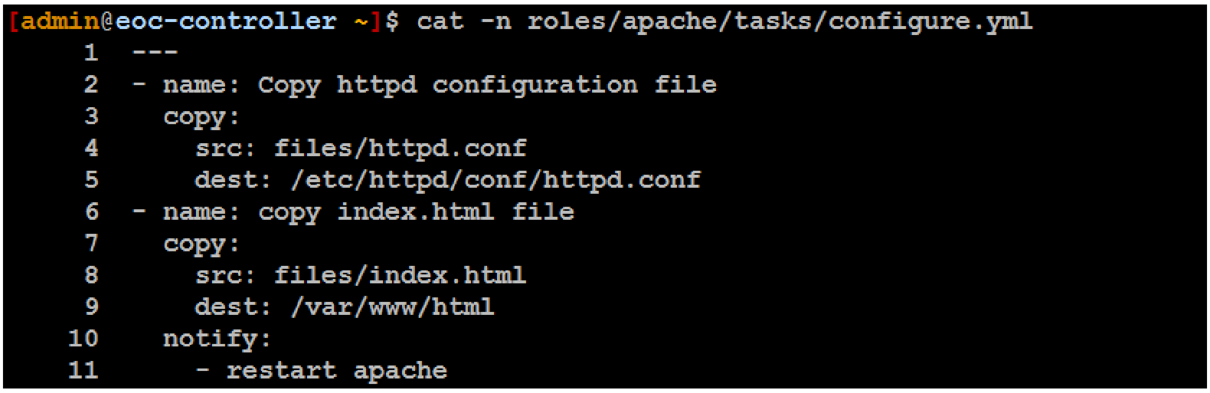
3. Writing a Play in configuration management.

3.1 Let's create the **configaure.yml** which perform configuration management in the task directory

Path: - **roles/apache/tasks/configure.yml**

```
# cat -n roles/apache/tasks/configure.yml
```

Output:



```
[admin@eoc-controller ~]$ cat -n roles/apache/tasks/configure.yml
     1  ---
     2  - name: Copy httpd configuration file
     3    copy:
     4      src: files/httpd.conf
     5      dest: /etc/httpd/conf/httpd.conf
     6  - name: copy index.html file
     7    copy:
     8      src: files/index.html
     9      dest: /var/www/html
    10    notify:
    11      - restart apache
```

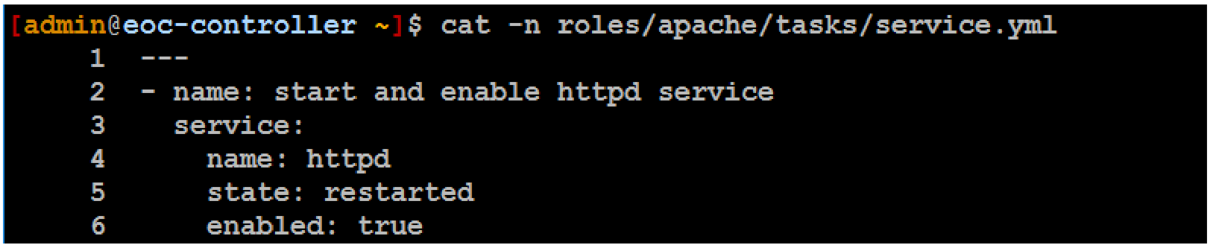
4. Starting httpd service

4.1 Let's create the **service.yml** which enable and start the httpd service in the task directory

Path: - **roles/apache/tasks/service.yml**

```
# cat -n roles/apache/tasks/service.yml
```

Output:



```
[admin@eoc-controller ~]$ cat -n roles/apache/tasks/service.yml
     1  ---
     2  - name: start and enable httpd service
     3    service:
     4      name: httpd
     5      state: restarted
     6      enabled: true
```

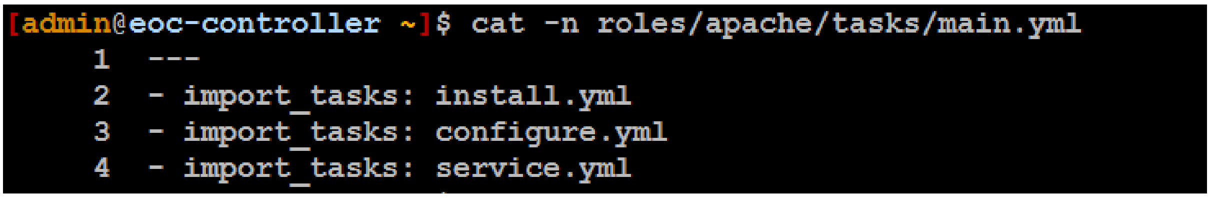
5. Breaking the codes

5.1 Let's break the codes even more as below using "import_tasks" statements

Path: - **roles/apache/tasks/main.yml**

```
# cat -n roles/apache/tasks/main.yml
```

Output:

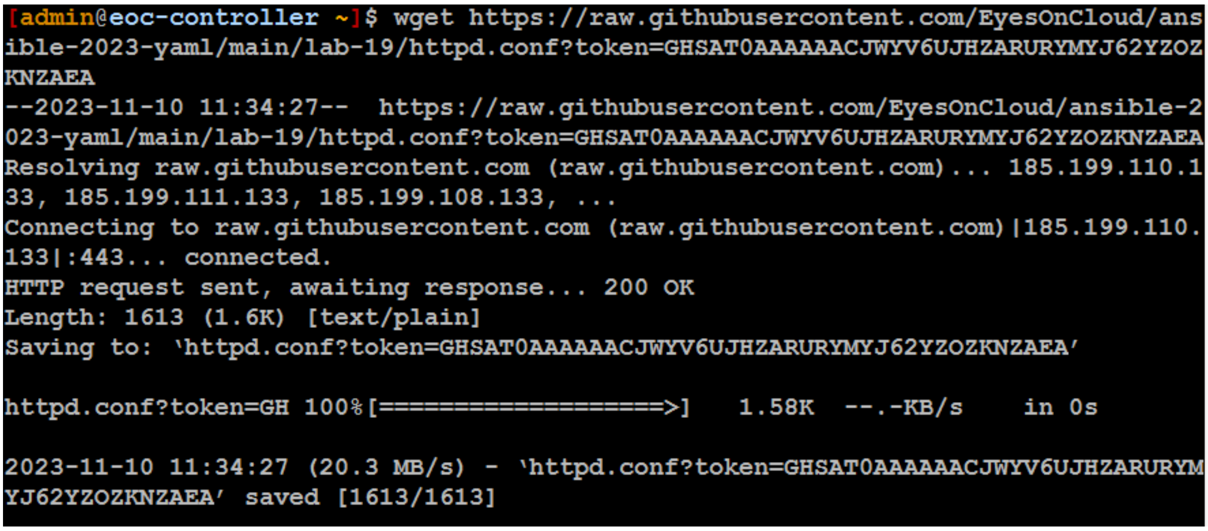


```
[admin@eoc-controller ~]$ cat -n roles/apache/tasks/main.yml
     1  ---
     2  - import_tasks: install.yml
     3  - import_tasks: configure.yml
     4  - import_tasks: service.yml
```

5.2 Let's download the required files (**httpd.conf**).

```
# wget
https://raw.githubusercontent.com/EyesOnCloud/ansible-2023-yaml/main/lab-19/httpd.conf?token=GHSAT0AAAAAACJWYV6UJHZARURYMYJ62YZOZKNZAEA
```

Output:



```
[admin@eoc-controller ~]$ wget https://raw.githubusercontent.com/EyesOnCloud/ansible-2023-yaml/main/lab-19/httpd.conf?token=GHSAT0AAAAAACJWYV6UJHZARURYMYJ62YZOZKNZAEA
--2023-11-10 11:34:27--  https://raw.githubusercontent.com/EyesOnCloud/ansible-2023-yaml/main/lab-19/httpd.conf?token=GHSAT0AAAAAACJWYV6UJHZARURYMYJ62YZOZKNZAEA
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.111.133, 185.199.108.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1613 (1.6K) [text/plain]
Saving to: 'httpd.conf?token=GHSAT0AAAAAACJWYV6UJHZARURYMYJ62YZOZKNZAEA'

httpd.conf?token=GH 100%[===================>]   1.58K  --.-KB/s    in 0s

2023-11-10 11:34:27 (20.3 MB/s) - 'httpd.conf?token=GHSAT0AAAAAACJWYV6UJHZARURYMYJ62YZOZKNZAEA' saved [1613/1613]
```

5.3 Let’s copy the file to the **roles/apache/files/** directory.

```
# cp
'httpd.conf?token=GHSAT0AAAAAACJWYV6UJHZARURYMYJ62YZOZKNZAEA
' roles/apache/files/httpd.conf
```

5.4 Let’s copy the required files **index.html** to the files directory

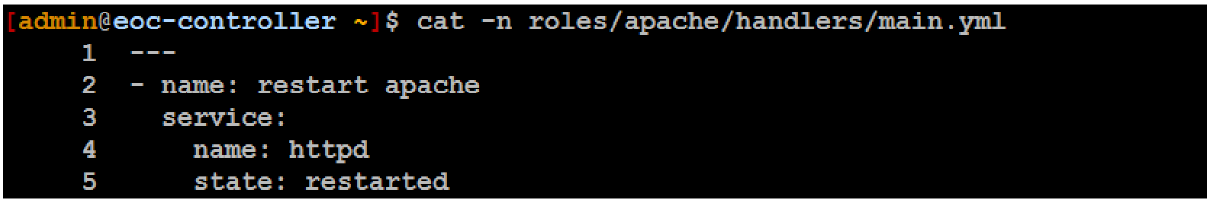
```
# cat > roles/apache/files/index.html << EOF
“Hello and Welcome to Ansible Class”
EOF
```

5.5 Let’s edit the **main.yaml** to restart the server when there is a change. Because we have already defined it in the task with notify option. Use the same name “restart apache”

Path: - roles/apache/handlers/main.yml

```
# cat -n roles/apache/handlers/main.yml
```

Output:

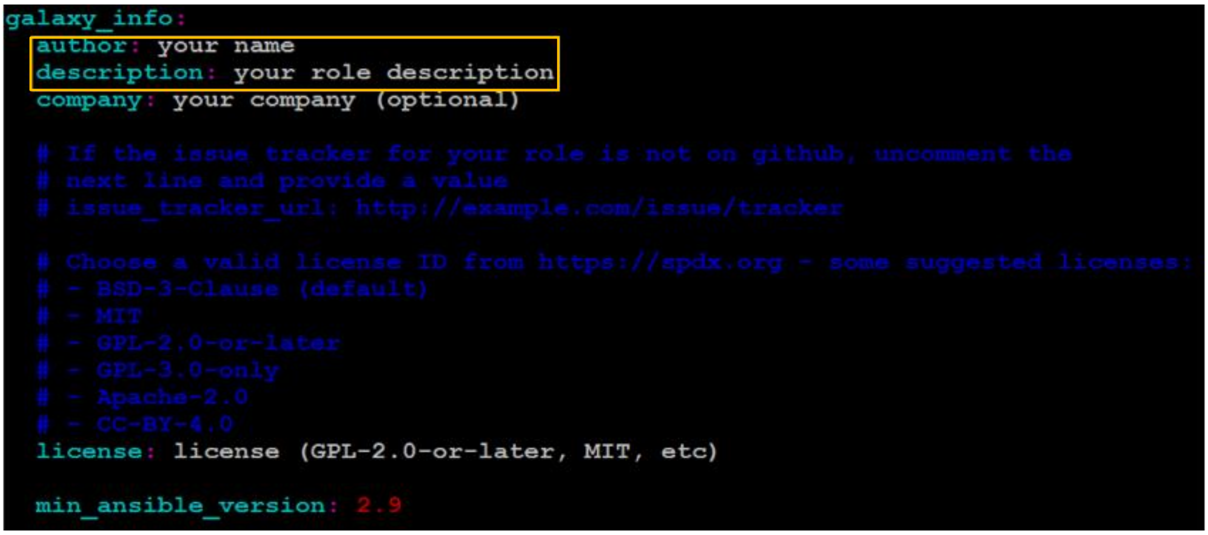


```
[admin@eoc-controller ~]$ cat -n roles/apache/handlers/main.yml
     1  ---
     2  - name: restart apache
     3    service:
     4      name: httpd
     5      state: restarted
```

5.6 Let’s edit the meta **main.yaml** to add the information about the roles like author, description, license, platform supported.

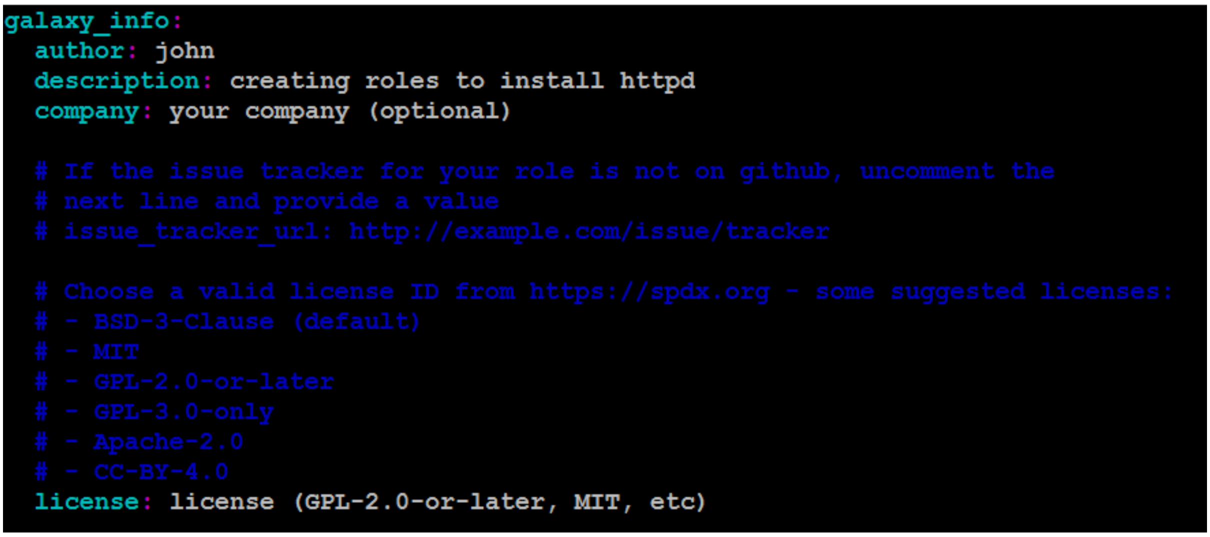
```
# vim roles/apache/meta/main.yml
```

Output:



```
galaxy_info:
  author: your name
  description: your role description
  company: your company (optional)

  # If the issue tracker for your role is not on github, uncomment the
  # next line and provide a value
  # issue_tracker_url: http://example.com/issue/tracker

  # Choose a valid license ID from https://spdx.org - some suggested licenses:
  # - BSD-3-Clause (default)
  # - MIT
  # - GPL-2.0-or-later
  # - GPL-3.0-only
  # - Apache-2.0
  # - CC-BY-4.0
  license: license (GPL-2.0-or-later, MIT, etc)

  min_ansible_version: 2.9
```

Change the above reuired lines in the file.

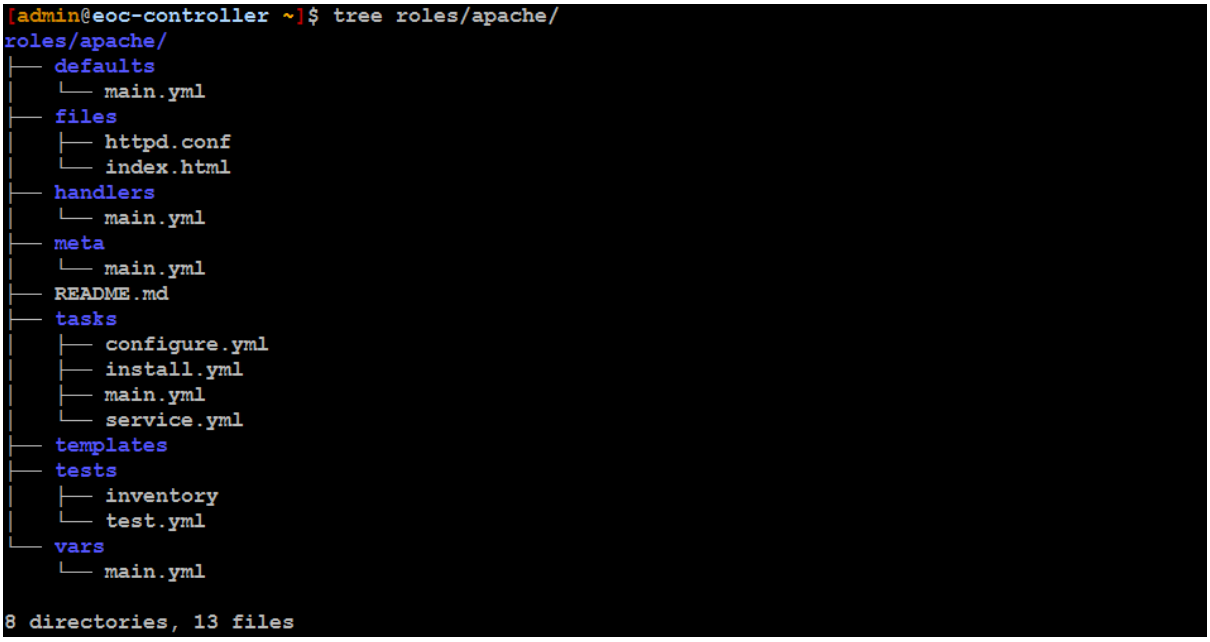


```
galaxy_info:
  author: john
  description: creating roles to install httpd
  company: your company (optional)

  # If the issue tracker for your role is not on github, uncomment the
  # next line and provide a value
  # issue_tracker_url: http://example.com/issue/tracker

  # Choose a valid license ID from https://spdx.org - some suggested licenses:
  # - BSD-3-Clause (default)
  # - MIT
  # - GPL-2.0-or-later
  # - GPL-3.0-only
  # - Apache-2.0
  # - CC-BY-4.0
  license: license (GPL-2.0-or-later, MIT, etc)
```

5.7 Let’s view all the files using tree command

```
# tree roles/apache/
```

Output:



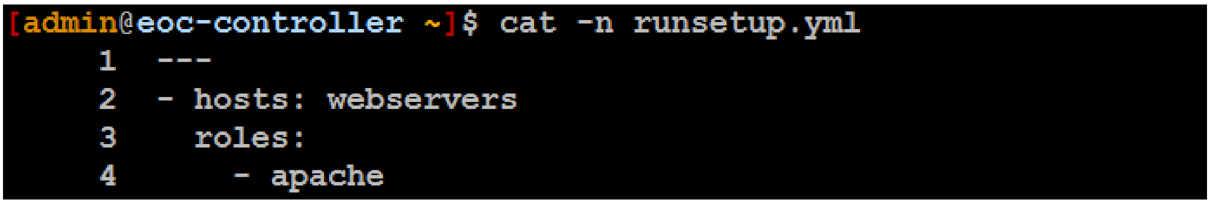
```
[admin@eoc-controller ~]$ tree roles/apache/
roles/apache/
├── defaults
│   └── main.yml
├── files
│   ├── httpd.conf
│   └── index.html
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── README.md
├── tasks
│   ├── configure.yml
│   ├── install.yml
│   ├── main.yml
│   └── service.yml
├── templates
├── tests
│   ├── inventory
│   └── test.yml
└── vars
    └── main.yml

8 directories, 13 files
```

Note: Before further proceeding verify the files created

5.8 We have got the required files for Apache roles. Let's apply this role into the ansible playbook "**runsetup.yml**" as below to deploy it on the client nodes.

```
# cat -n runsetup.yml
```

Output:

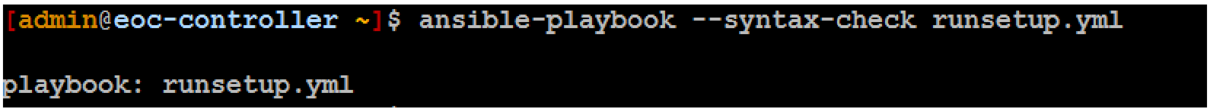


```
[admin@eoc-controller ~]$ cat -n runsetup.yml
     1  ---
     2  - hosts: webservers
     3    roles:
     4      - apache
```

5.9 Let's verify if there are any syntax errors.

```
# ansible-playbook --syntax-check runsetup.yml
```

Output:

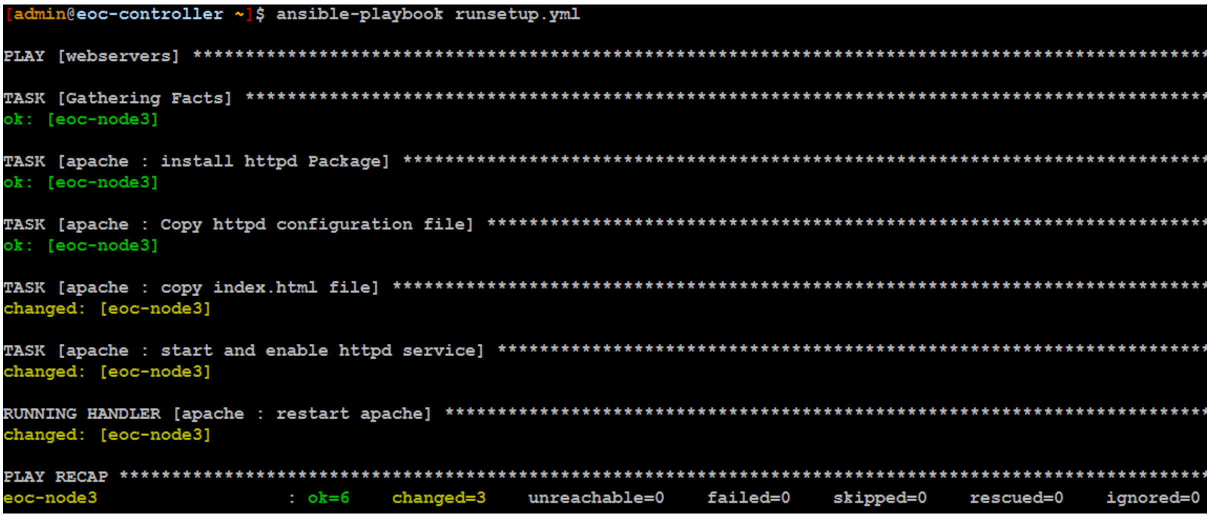


```
[admin@eoc-controller ~]$ ansible-playbook --syntax-check runsetup.yml

playbook: runsetup.yml
```

5.10 Let's deploy the roles.

```
# ansible-playbook runsetup.yml
```

Output:

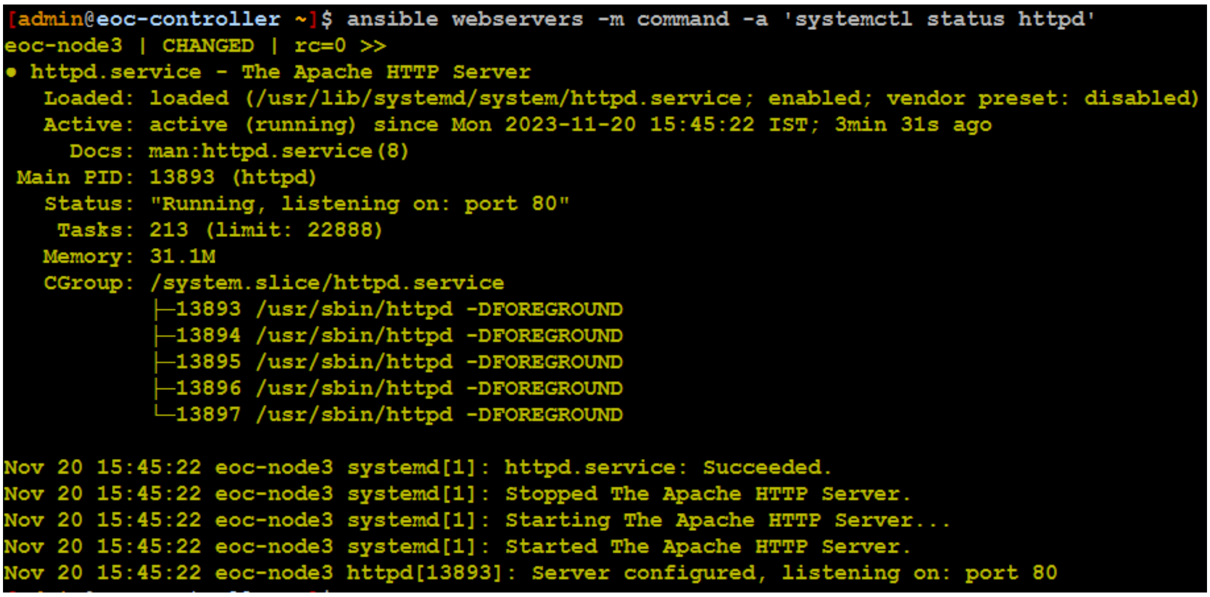


```
[admin@eoc-controller ~]$ ansible-playbook runsetup.yml

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

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

TASK [apache : install httpd Package] **********************************************************
ok: [eoc-node3]

TASK [apache : Copy httpd configuration file] **************************************************
ok: [eoc-node3]

TASK [apache : copy index.html file] ***********************************************************
changed: [eoc-node3]

TASK [apache : start and enable httpd service] *************************************************
changed: [eoc-node3]

RUNNING HANDLER [apache : restart apache] ******************************************************
changed: [eoc-node3]

PLAY RECAP *************************************************************************************
eoc-node3                  : ok=6    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

5.11 Let's verify the status httpd on the managed hosts

```
# ansible webservers -m command -a 'systemctl status httpd'
```

Output:

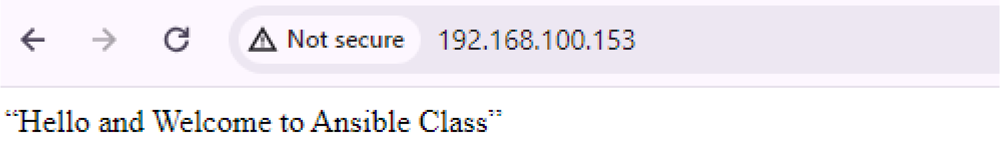


```
[admin@eoc-controller ~]$ ansible webservers -m command -a 'systemctl status httpd'
eoc-node3 | CHANGED | rc=0 >>
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2023-11-20 15:45:22 IST; 3min 31s ago
     Docs: man:httpd.service(8)
 Main PID: 13893 (httpd)
   Status: "Running, listening on: port 80"
    Tasks: 213 (limit: 22888)
   Memory: 31.1M
   CGroup: /system.slice/httpd.service
           ├─13893 /usr/sbin/httpd -DFOREGROUND
           ├─13894 /usr/sbin/httpd -DFOREGROUND
           ├─13895 /usr/sbin/httpd -DFOREGROUND
           ├─13896 /usr/sbin/httpd -DFOREGROUND
           └─13897 /usr/sbin/httpd -DFOREGROUND

Nov 20 15:45:22 eoc-node3 systemd[1]: httpd.service: Succeeded.
Nov 20 15:45:22 eoc-node3 systemd[1]: Stopped The Apache HTTP Server.
Nov 20 15:45:22 eoc-node3 systemd[1]: Starting The Apache HTTP Server...
Nov 20 15:45:22 eoc-node3 systemd[1]: Started The Apache HTTP Server.
Nov 20 15:45:22 eoc-node3 httpd[13893]: Server configured, listening on: port 80
```

5.12 Let's verify the status httpd on client host (eoc-node3)

Let's access the service through browser

Output:



192.168.100.153

"Hello and Welcome to Ansible Class"