

## **ANSIBLE ROLES**

### **Ansible Roles**

An Ansible role is a reusable, self-contained unit of automation that is used to organize and manage tasks, variables, files, templates, and handlers in a structured way.

Roles help to encapsulate and modularize the logic and configuration needed to manage a particular system or application component.

This modular approach promotes reusability, maintainability, and consistency across different playbooks and environments.

### **Key Components of an Ansible Role**

#### **Tasks**

The main list of actions that the role performs.

#### **Handlers**

Tasks that are triggered by changes in other tasks, typically used for actions like restarting services.

#### **Files**

Static files that need to be transferred to managed hosts.

#### **Templates**

Jinja2 templates that can be rendered and transferred to managed hosts.

#### **Vars**

Variables that are used within the role.

#### **Defaults**

Default variables for the role, which can be overridden.

#### **Meta**

Metadata about the role, including dependencies on other roles.

#### **Library**

Custom modules or plugins used within the role.

#### **Module\_defaults**

Default module parameters for the role.

#### **Lookup\_plugins**

Custom lookup plugins for the role.

### **Directory Structure of an Ansible Role**

An Ansible role follows a specific directory structure:

```
<role_name>/
|— defaults/
|   └─ main.yml
|— files/
|— handlers/
|   └─ main.yml
|— meta/
|   └─ main.yml
|— tasks/
|   └─ main.yml
|— templates/
|— vars/
    └─ main.yml
```

## **Why Use Ansible Roles?**

### **Modularity**

Roles allow you to break down complex playbooks into smaller, reusable components. Each role handles a specific part of the configuration or setup.

### **Reusability**

Once created, roles can be reused across different playbooks and projects. This saves time and effort in writing redundant code.

### **Maintainability**

By organizing related tasks into roles, it becomes easier to manage and maintain the code. Changes can be made in one place and applied consistently wherever the role is used.

### **Readability**

Roles make playbooks cleaner and easier to read by abstracting away the details into logically named roles.

### **Collaboration**

Roles facilitate collaboration among team members by allowing them to work on different parts of the infrastructure independently.

### **Consistency**

Using roles ensures that the same setup and configuration procedures are applied uniformly across multiple environments, reducing the risk of configuration drift.

## Creating a Basic Role

Let's create a simple role named apache2 that installs and starts apache2 and deploy simple app.

Create the Role Structure: You can manually create the directories and files listed above, or use the **ansible-galaxy command** to initialize the role structure:

**ansible-galaxy role init role-name**

```
ubuntu@ip-172-31-15-154:~/ansible$ ansible-galaxy role init webserver
- Role webserver was created successfully
ubuntu@ip-172-31-15-154:~/ansible$ ls
httpd.yml  index.html  webserver
ubuntu@ip-172-31-15-154:~/ansible$ cd webserver/
ubuntu@ip-172-31-15-154:~/ansible/webserver$ ls
README.md  defaults  files  handlers  meta  tasks  templates  tests  vars
ubuntu@ip-172-31-15-154:~/ansible/webserver$
```

- I Created a role named as webserver .The role created below structure.

```
role_name/
├── defaults/           # Default variables for the role
│   └── main.yml
├── vars/              # Other variables for the role
│   └── main.yml
├── tasks/             # Main list of tasks to be executed by the role
│   └── main.yml
├── files/             # Contains files used by the role
├── templates/         # Contains templates used by the role
├── handlers/          # Handlers, which may be used within or outside this ro
│   └── main.yml
└── meta/              # Defines some meta data for this role.
    └── main.yml
```

- Previously I written a playbook To install apache2 webserver and Deploy a simple app on aws.
- Now I convert that playbook into role .

CREATE ROLE.

- Execute below command to create role.

**ansible-galaxy role init role-name**

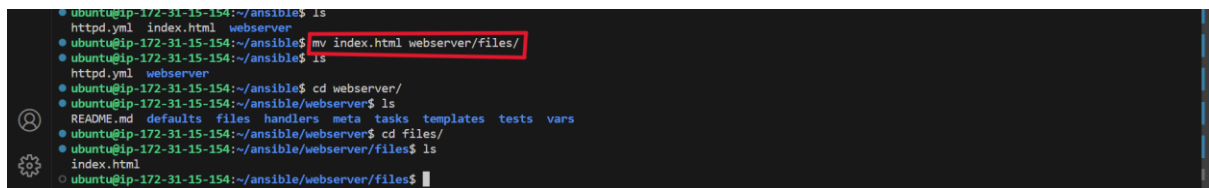
```
ubuntu@ip-172-31-15-154:~/ansible$ ansible-galaxy role init webserver
- Role webserver was created successfully
ubuntu@ip-172-31-15-154:~/ansible$ ls
httpd.yml  index.html  webserver
ubuntu@ip-172-31-15-154:~/ansible$ cd webserver/
ubuntu@ip-172-31-15-154:~/ansible/webserver$ ls
README.md  defaults  files  handlers  meta  tasks  templates  tests  vars
ubuntu@ip-172-31-15-154:~/ansible/webserver$
```

- Place Tasks in tasks/main.yml file.



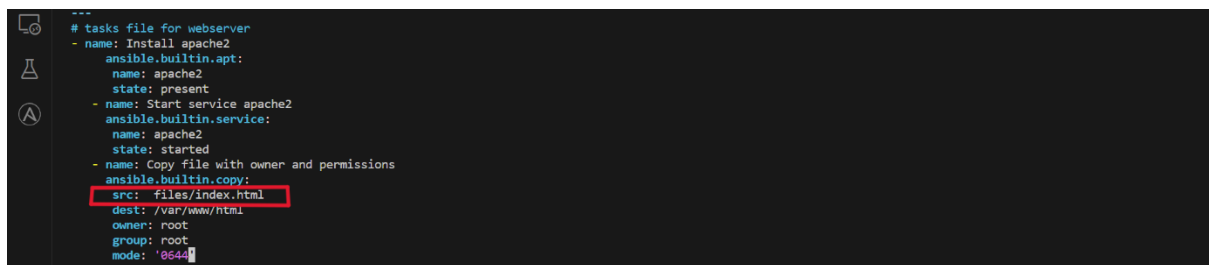
```
---
# tasks file for webserver
- name: Install apache2
  ansible.builtin.apt:
    name: apache2
    state: present
- name: Start service apache2
  ansible.builtin.service:
    name: apache2
    state: started
- name: Copy file with owner and permissions
  ansible.builtin.copy:
    src: files/index.html
    dest: /var/www/html
    owner: root
    group: root
    mode: '0644'
```

- Place a File (index.html) in files directory.



```
ubuntu@ip-172-31-15-154:~/ansible$ ls
httpd.yml  index.html  webserver
ubuntu@ip-172-31-15-154:~/ansible$ mv index.html webserver/files/
ubuntu@ip-172-31-15-154:~/ansible$ ls
httpd.yml  webserver
ubuntu@ip-172-31-15-154:~/ansible$ cd webserver/
ubuntu@ip-172-31-15-154:~/ansible/webserver$ ls
README.md  defaults  files  handlers  meta  tasks  templates  tests  vars
ubuntu@ip-172-31-15-154:~/ansible/webserver$ cd files/
ubuntu@ip-172-31-15-154:~/ansible/webserver/files$ ls
index.html
ubuntu@ip-172-31-15-154:~/ansible/webserver/files$
```

- Need to Mention location of index.html in playbook/



```
---
# tasks file for webserver
- name: Install apache2
  ansible.builtin.apt:
    name: apache2
    state: present
- name: Start service apache2
  ansible.builtin.service:
    name: apache2
    state: started
- name: Copy file with owner and permissions
  ansible.builtin.copy:
    src: files/index.html
    dest: /var/www/html
    owner: root
    group: root
    mode: '0644'
```

- Mention the role in httpd.yml file.



```
---
- hosts: all
  connection: ssh
  become: true
  roles:
    - webserver
```

- Execute the playbook

```
ubuntu@ip-172-31-15-154:~/ansible$ 1
httpd.yml webservers/
ubuntu@ip-172-31-15-154:~/ansible$ sudo ansible-playbook httpd.yml

PLAY [all] *********************************************************************

TASK [Gathering Facts] *********************************************************
[WARNING]: Platform linux on host ubuntu@172.31.0.155 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [ubuntu@172.31.0.155]
[WARNING]: Platform linux on host ubuntu@172.31.2.17 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [ubuntu@172.31.2.17]

TASK [webservers : Install apache2] *********************************************
ok: [ubuntu@172.31.2.17]
ok: [ubuntu@172.31.0.155]

TASK [webservers : Start service apache2] *************************************
ok: [ubuntu@172.31.0.155]
ok: [ubuntu@172.31.2.17]

TASK [webservers : Copy file with owner and permissions] ********************
changed: [ubuntu@172.31.0.155]
changed: [ubuntu@172.31.2.17]

PLAY RECAP *********************************************************************
ubuntu@172.31.0.155      : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu@172.31.2.17      : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ubuntu@ip-172-31-15-154:~/ansible$
```

- Access the App using public\_ip of managed nodes with port 80.

