ROLES

❖ ANSIBLE ROLES:

- Ansible roles provide a well-defined framework and structure for setting your tasks, variables, handlers, metadata, templates, and other files. They enable us to reuse and share our Ansible code efficiently.
- It is a self-contained, portable unit of Ansible automation that serves as the preferred method for grouping related tasks and associated variables, files, handlers, and other assets in a known file structure.

BENEFITS:

- Reusability and sharing
- Modularity
- Organization
- Parameterization
- Versioning and dependency management
- Testing and Simplicity

➢ SHARING ANSIBLE ROLES:

- The ease of sharing Ansible Roles allows you to incorporate well-maintained, tested roles into your automation. Roles can be shared using these repositories:

ANSIBLE GALAXY:

- A free repository for sharing roles and other Ansible content with the larger Ansible community.
- Roles can be uploaded to Ansible Galaxy via the command-line, whereas collections can be shared from the UI.

ANSIBLE AUTOMATION HUB:

- It is a central repository for finding, downloading, and sharing Ansible Content Collections.
- Ansible automation hub is hosted by Red Hat and contains both certified and validated content from Red Hat and our independent software vendor (ISV) partners.

PRIVATE AUTOMATION HUB:

- An on-premise repository, private automation hub allows organizations to manage, share, and curate content internally—just for your organization.
- You can share roles and other automation content within your enterprise, allowing teams to simplify workflows and accelerate automation.

EXAMPLE: CREATE A ROLE FOR APACHE HTTP:

SYNTAX: $ansible-galaxy init <your_role_name>

$cd /etc/ansible

$cd roles

$sudo ansible-galaxy init apache-http --offline

$ls

$tree apache-http

$cd tasks

$vim main.yml

```
---
- include: install.yml
- include: configure.yml
- include: check_url_response.yml
...
```

$vim install.yml

```
- name: Install httpd
  yum:
    name: httpd
    state: present
    update_cache: yes
```

$vim configure.yml

```
- name: Copy index.html file
  copy:
    src: index.html
    dest: /var/www/html/index.html
  notify: restart http
```

NOTE: Roles will search the files in the folder of files. so we can create a file index.html in the files folder.

$cd files

$vim index.html

WELCOME TO AWS...!

$cd tasks

$vim check_url_response.yml

```
- name: Check Apache Response on all servers
  uri:
    url: "{{item}}"
    status_code: 200
  with_items:
    - http://node1-ip
    - http://node2-ip
```

$cd handlers

$vim main.yml

```
- name: restart http
  service:
    name: http
    state: restarted
```

NOW CREATE A PLAYBOOK IN ROLES LOCATION:

$cd /etc/ansible/roles

$vim configure_apache.yml

```
---
- name: configure apache using roles
  hosts: webservers
  roles:
    - apache
```

$ansible-playbook configure_apache.yml -b -K