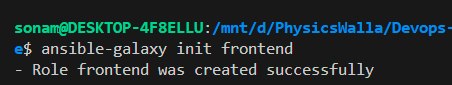
Creating roles in Ansible

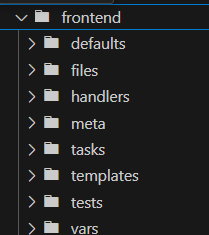
Create a folder for your Project: Ansible\_roles

Move to the folder cd ansible\_roles

Inside that open terminal or wsl where your ansible is installed.

* ansible-galaxy init frontend (runn command)





This Folder structure was generated.

* defaults -- keep default variables
* files -- keep static files to copy
* handlers -- restart services etc..
* meta -- role dependencies
* tasks -- main task
* templates -- jinja2 templates
* tests -- write test cases
* vars -- role specific variables add here

Now Let’s Code the same.

Tasks folder – main.yml file

---

- name: Install Nginx

  apt:

    name: nginx

    state: present

    update\_cache: yes

- name: Copy Custom index.html

  template:

    src: index.html.j2

    dest: /var/www/html/index.html

  notify: restart nginx

handlers – main.yml

---

- name: restart nginx

  service:

    name: nginx

    state: restarted

--defaults – main.yml

---

nginx\_title: "Welcome to My Website"

nginx\_message: "Deployed using Ansible Role"

Templates folder create 🡪 index.html.j2

<html>

    <head>

        <title>

            {{ nginx\_title }}

        </title>

        <body>

            <h1>{{ nginx\_message }}</h1>

        </body>

    </head>

</html>

Create main playbook.yml outside the frontend folder.

---

- hosts: aws1

  become: yes

  roles:

    - frontend

Create inventory:

all:

  children:

    web:

      hosts:

        aws1:

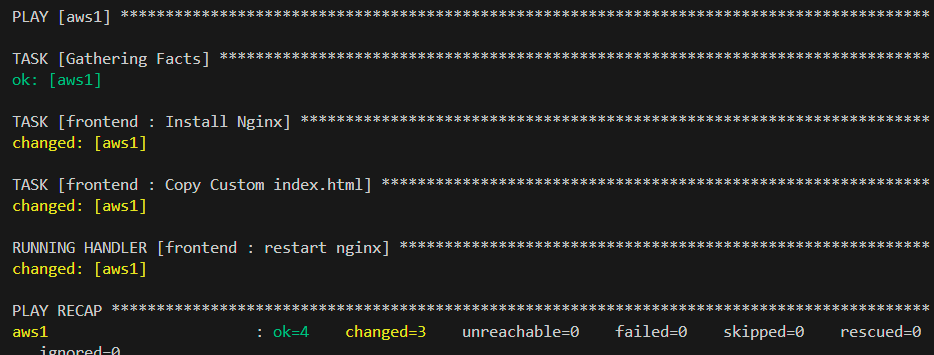
          ansible\_host: ec2-18-234-221-87.compute-1.amazonaws.com

          ansible\_user: ubuntu

          ansible\_ssh\_private\_key\_file: ~/.ssh/testsonam.pem

Give your Public IP or DNS accordingly and run below command.

ansible-playbook -i inventory.yml playbook.yml



Create 1 more Role:

Ansible-galaxy init backend

Opne tasks—main.yml

---

- name: Install Python3 and pip

  apt:

    name:

      - python3

      - python3-pip

      - python3-flask

    state: present

    update\_cache: yes

# - name: Install flask

#   pip:

#     name: flask

- name: upload simple server code

  copy:

    dest: /home/ubuntu/app.py

    content: |

      from flask import Flask

      app = Flask(\_\_name\_\_)

      @app.route('/')

      def hello():

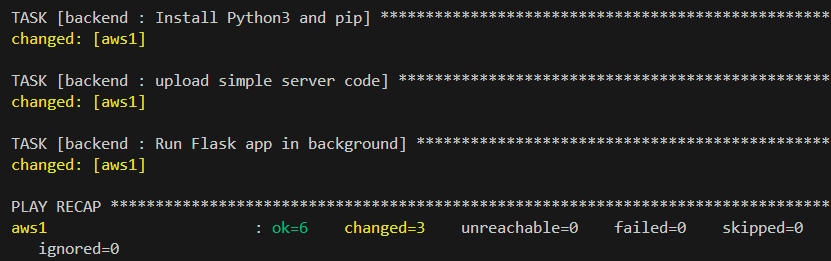
          return "Hello From Backend"

      if \_\_name\_\_ == '\_\_main\_\_':

          app.run(port=5000)

- name: Run Flask app in background

  shell: nohup python3 /home/ubuntu/app.py &



Go to your Security group and Open 5000 port in inbound rules and then check output with

Publicip:5000

It should revert with hello from backend.