

***DevOps***

**Training Assignment/Lab**

|  |  |
| --- | --- |
| Document Code | 25e-BM/HR/HDCV/FSOFT |
| Version | 1.1 |
| Effective Date | 00/00/2022 |

**Hanoi, 01/2022**

RECORD OF CHANGES

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Effective Date | Change Description | Reason | Reviewer | Approver |
| 1 | 25/03/2022 | Create a new Assignment/Lab | Create new | ThuongDD7 | VinhNV |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Contents

[Day 2, 3. Unit2:Ansible 3](#_Toc126424240)

[Assignment : 3](#_Toc126424241)

[1. Install Ansible 3](#_Toc126424242)

[2. Ansible Inventory 4](#_Toc126424243)

[3. Ansible playbook 6](#_Toc126424244)

[4. Role 11](#_Toc126424245)

[5. Ansible galaxy 13](#_Toc126424246)

[6. Ansible Tower 14](#_Toc126424247)

|  |  |
| --- | --- |
| D:\FSOFT\SERVER GST\86.WIP\2.Template\6_Logo\Logo FA\Logo FA New-02.png | **CODE : DevOps**  **TYPE : Lab**  **LOC : <Lines of Code>**  **DURATION : 2 day** |

# Day 2, 3. Unit2:Ansible

Assignment :

# Install Ansible

Cài đặt Ansible trên Ubuntu:

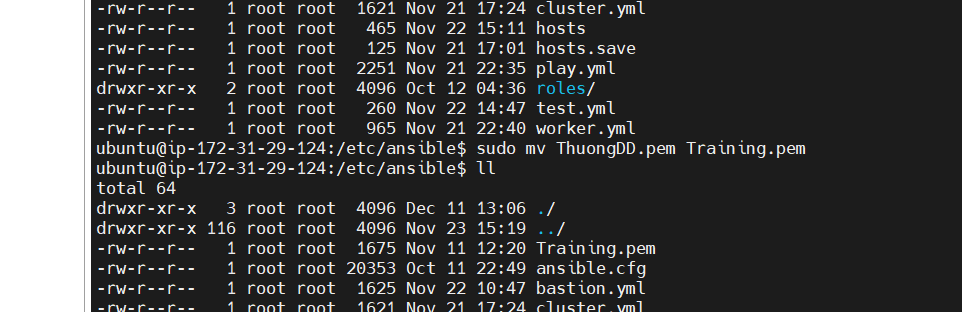
sudo apt update

sudo apt install software-properties-common

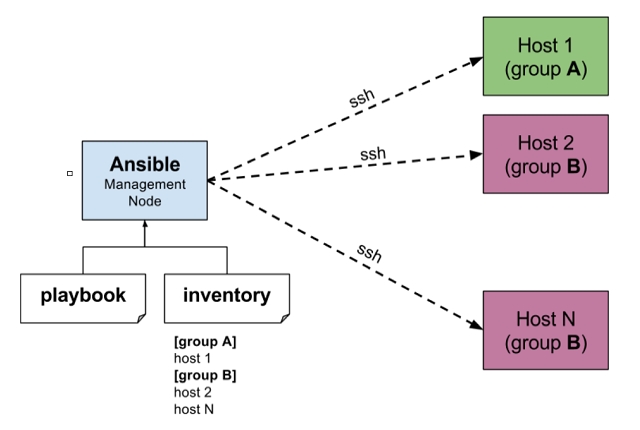
sudo add-apt-repository --yes --update ppa:ansible/ansible

sudo apt install ansible

Có nhiều các để gọi đến các server ta có thể copy public key của máy chạy ansible để máy server, cũng có thể sử dụng user, password để xác thực. Ở đây ta dùng keypair khi tạo server để xác thực, ta copy keypair vào /etc/ansible/ để sử dụng.



Mô hình hoạt động của Ansible:



# Ansible Inventory

Đầu tiên ta cần khai báo host\_group để dễ gọi các nhóm server. Ví dụ ta muốn cài đặt nginx lên nhóm server web\_app, hay cài mysql lên nhóm server database, … Các quy hoạch này của ansible giúp dễ gọi các lệnh về sau.

//nano /etc/ansible/hosts

[localhost]

127.0.0.1

[web\_app]

172.31.13.222

[database]

172.31.13.200

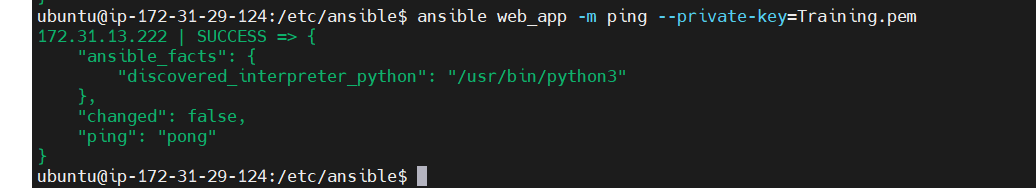
Cấu trúc lệnh ansible như sau:

# ansible [hosts] –m [module] –a [tham số truyền vào] –private-key=[keypair]

Cấu trúc lệnh trên ta sử dụng private key để xác thực kết nối hoặc ta có thể sử dụng **username** và **password** để xác thực với cấu trúc lệnh sau:

#ansible [host] –m [module] –u [username]

Ví dụ, ta kiểm tra kết nối tới server ở nhóm **web\_app** với lệnh sau:



Ở đây ta sử dụng module **ping**  để kiểm tra kết nối đến server nhóm **web\_app** và sử dụng phương thức xác thực là dùng privatekey. Kết quả, kết nối thành công.

Trên đây là cách dùng truyền thống khi khai báo các tham số khi gọi lệnh. Ta có thể khai báo các tham số của server trong file hosts như sau:

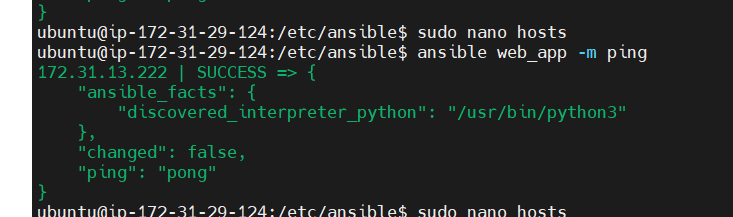
[localhost]

127.0.0.1

[web\_app]

172.31.13.222 ansible\_ssh\_user=ubuntu ansible\_ssh\_private\_key\_file=Training.pem

Ta thử chạy lại lệnh trên mà k cần khai báo lại private-key:



\*# ansible [tên host cần gọi] -m [tên module] -a [tham số truyền vào module]\*

-i : inventory host. Load thư viện host

-m : gọi module của ansible

-a : command\_argument gửi kèm theo module mà ta đang gọi

-u : user

-vvvv : debug option

$$ ansible all -m ping (giải thích: gọi ping toàn bộ các hosts trong /etc/ansible/hosts)

$$ ansible all -m command -a uptime

$$ ansible all -a uptime (Default, ansible sẽ cho module = "command". Nên ta ko cần -m command thêm vào cũng được.)

$$ ansible -m shell -a 'top -bcn1 | head' (giải thích: chạy lệnh shell ở remote client![](https://images.viblo.asia/8335ed0c-7cfe-41b6-b7cf-bd37e05979a7.png)

)

- restart mysql

$$ ansible dbserver -m service -a "name=mysql state=restarted" --key-file=~/.ssh/db.pem -u ubuntu --sudo

Tất cả module của ansible bạn có thể tham khảo ở đây http://docs.ansible.com/ansible/list\_of\_all\_modules.html . Được chia các module chính như: db, file, monitor, network, package, storage, web, cloud....

# Ansible playbook

Ansible rất linh hoạt khi hỗ trợ playbook bằng ngôn ngữ YAML (file.yml). Ansible hỗ trợ với rất nhiều **module** giúp ta chạy các lệnh dễ dàng và giúp ta dễ đọc hiểu. playbook đơn giản có mẫu như sau:

//nano playbook.yml

---

- hosts: web\_app

  become: yes

  tasks:

    - name: Ping host

      ping: ~

    - name: Install nginx

      apt:

        name: nginx

Các module được tùy biến trên nhiều hệ điều hành khác nhau. Ví dụ như trên Ubuntu để cài đặt dịch vụ ta dùng module **apt** còn trên CentOS ta dùng module **yum**.

//nano playbook.yml

---

- hosts: web\_app

  become: yes

  tasks:

    - name: Ping host

      ping: ~

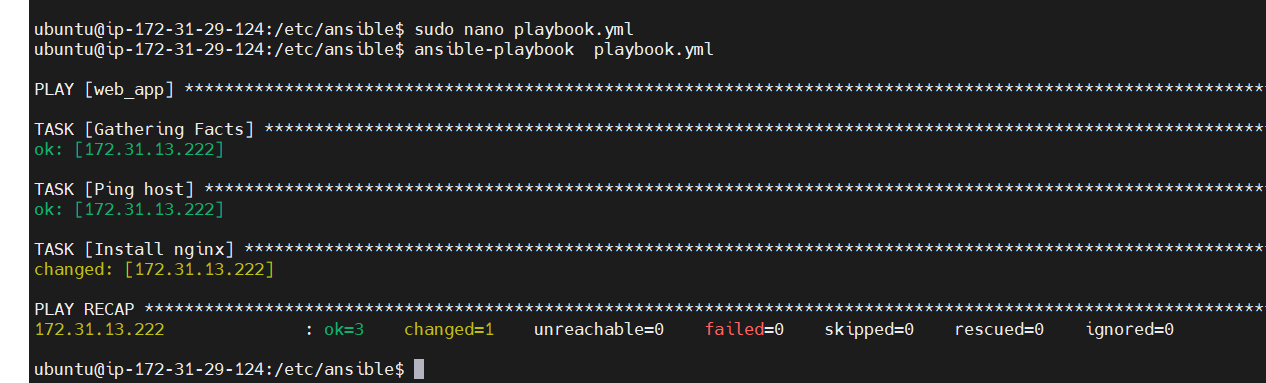
    - name: Install nginx

      yum:

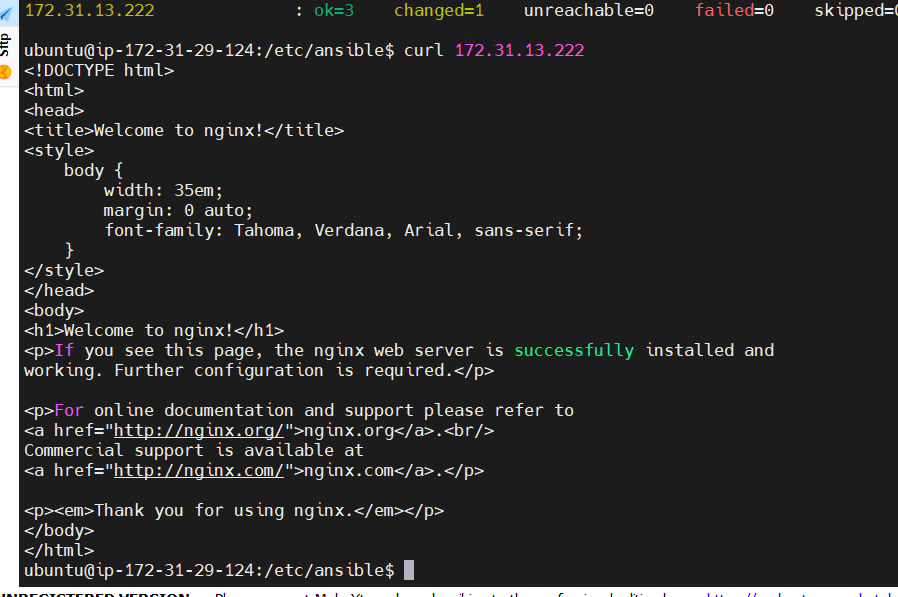
        name: nginx

Để chạy playbook ta sử dụng lệnh sau:

ansible-playbook playbook.yml



Kiểm tra xem nginx đã được cài trên web\_app chưa:



Dịch vụ nginx cài đặt thành công, playbook chạy thành công.

Tương tự ta có thể viết playbook để update máy server và copy file index lên server web\_app. Tạo file với nội dung sau:

//nano playbook.yml

---

- hosts: web\_app

  become: yes

  tasks:

    - name: Update and upgrade apt packages

      apt:

        upgrade: yes

        update\_cache: yes

    - name: Copy file

      copy:

        src: index.html

        dest: /var/www/html/

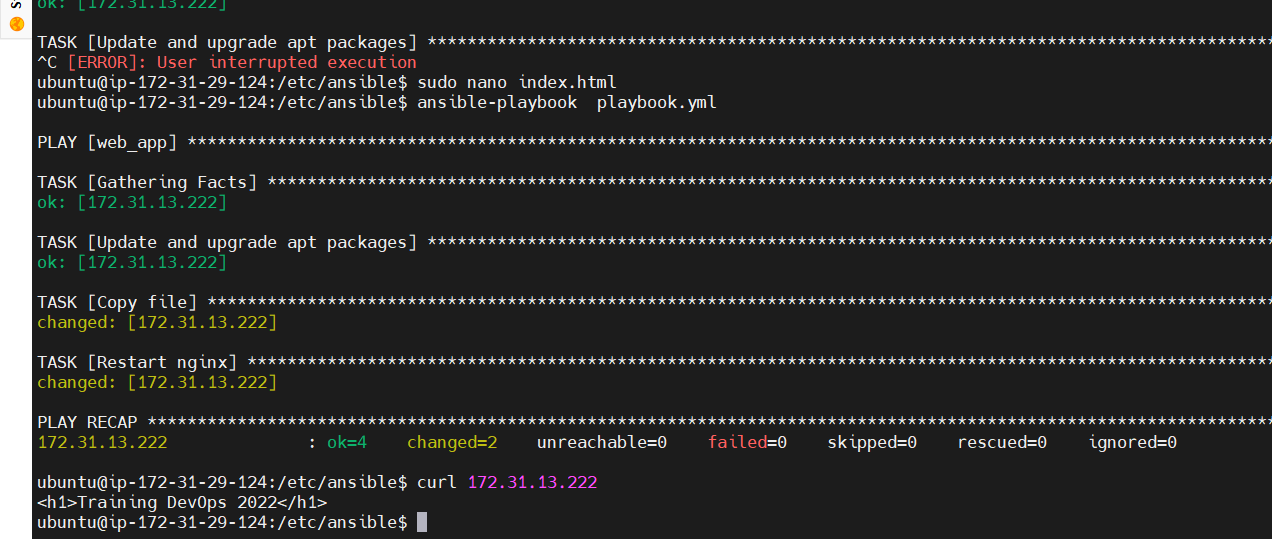
    - name: Restart nginx

      service:

        name: nginx

        state: restarted

 Tạo file index.html với nội dung bất kì lưu ở đường dẫn /etc/ansible/ và chạy playbook rồi kiểm tra kết quả:



Ta thấy nội dung file index đã được thay đổi. playbook chạy thành công.

Tương tự các module khác ta có thể search google.

* Thay vì viết module apt cho từng gói cài đặt ta có thể nhóm vào **item** để chạy 1 lần và sử dụng **handlers** để thực hiện các hành động cần thực thi nhiều lần (notify).

//playbook.yml

---

- hosts: web\_app

  become: yes

  tasks:

    - name: Install Apache.

      apt:

        name: "{{ item }}"

        state: present

      with\_items:

        - apache2

        - mysql-server

    - name: deploy html file

      template:

        src: /tmp/index.html

        dest: /var/www/html/index.html

      notify: restart web

  handlers:

    - name: restart web

      service:

        name: "{{ item }}"

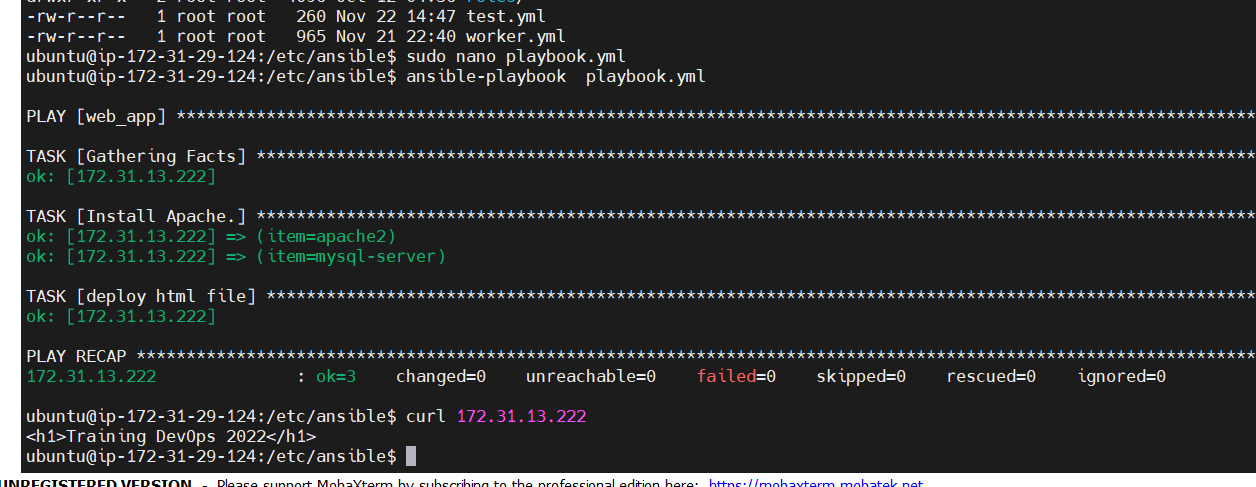
        state:  running

      with\_items:

          - apache2

          - mysql

Run playbook và kiểm tra kết quả playbook chạy thành công.



* Ta có thể khai báo biến trong playbook như sau:

//nano playbook.yml

---

- hosts: web\_app

  become: yes

  vars:

    - index\_file: "index.html"

    - variable: "Test var Ansible"

  tasks:

    - name: Install Apache.

      apt:

        name: "{{ item }}"

        state: present

      with\_items:

        - apache2

        - mysql-server

    - name: deploy html file

      template:

        src: "{{index\_file}}"

        dest: /var/www/html/index.html

      notify: restart web

  handlers:

    - name: restart web

      service:

        name: "{{ item }}"

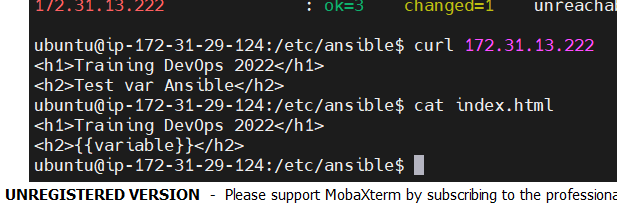
        state:  running

      with\_items:

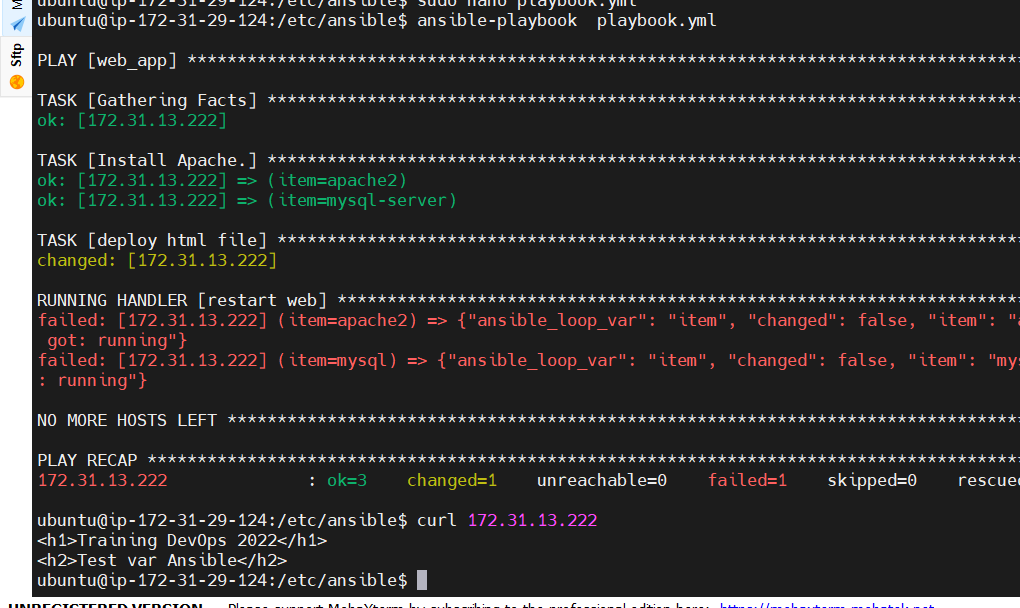
          - apache2

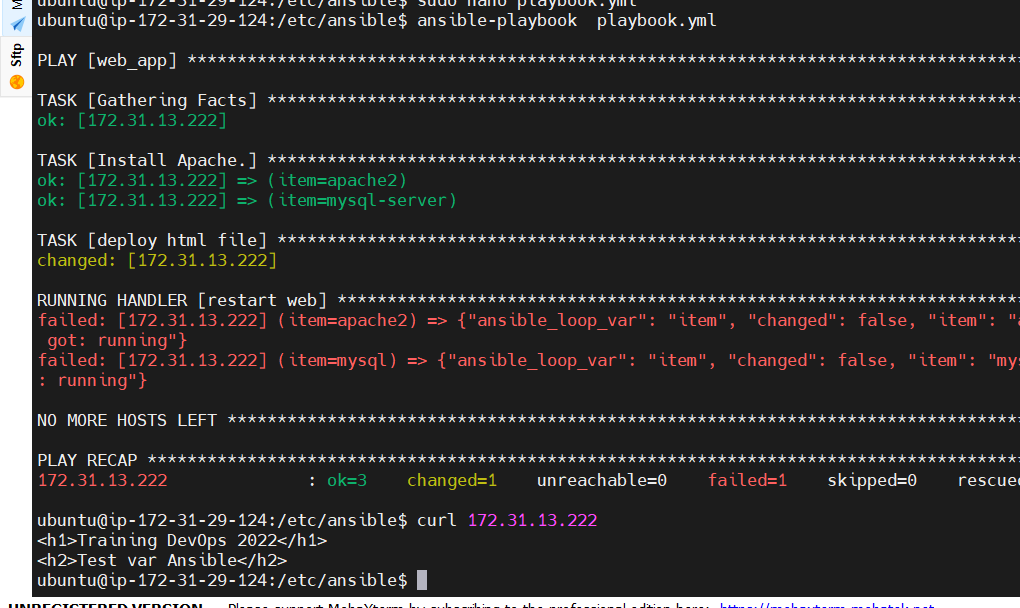
          - mysql

Thay đổi nội dung file index.html như sau:



Chạy và kiểm tra kết quả ta đã lấy được biến khai báo trong playbook.

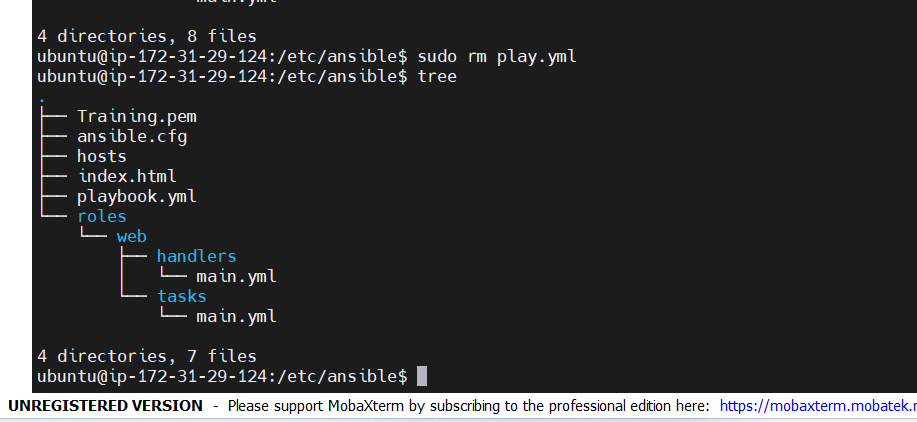




# Role

Trong Ansible, **Role** là cơ chế tách 1 playbook ra thành nhiều file. Việc này nhằm đơn giản hóa việc viết các playbook phức tạp có thể tái sử dụng nhiều lần. Mỗi role là một thành phần độc lập, bao gồm nhiều **variables, tasks, files, templates** và **modules** bên dưới.

Để dễ hình dung ta sẽ làm lab với role **web** có **task** và **handlers** như sau:



Nội dung các file như sau:

playbook.yml

---

- hosts: web\_app

  become: yes

  tasks:

    - name: include web role

      include\_role:

        name: web

      tags:

        - deployments

/roles/web/tasks/main.yml

---

- name: Install Apache.

  apt:

    name: "{{ item }}"

    state: present

  with\_items:

    - apache2

    - mysql-server

- name: deploy html file

  template:

    src: index.html

    dest: /var/www/html/index.html

  notify: restart web

/roles/web/handlers/main.yml

---

- name: restart web

  service:

    name: "{{ item }}"

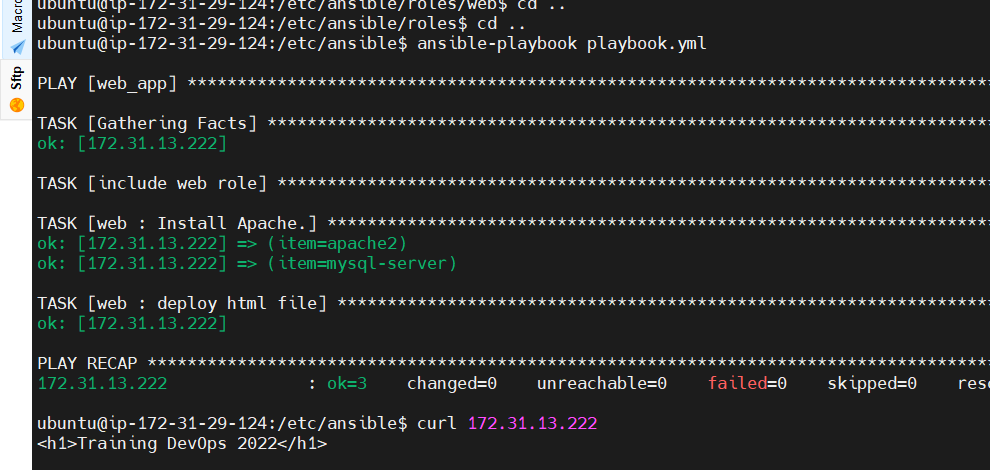
    state:  restarted

  with\_items:

      - apache2

      - mysql

Chạy playbook và kiểm tra kết quả play chạy thành công.



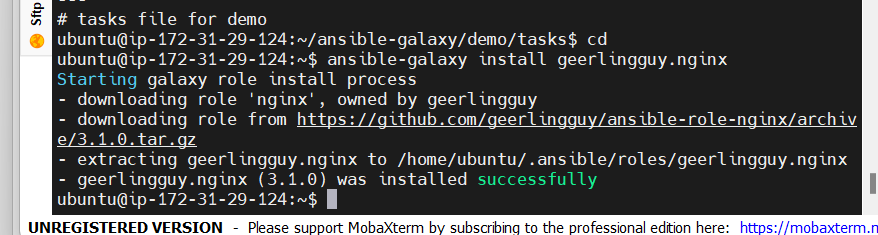
Thông thường các playbook có rất nhiều thành phần nên việc phân chia ra các role giúp việc quản lý dễ dang hơn. Trên đây chỉ là ví dụ nhỏ để ta hình dung ra cấu trúc của roles.

# Ansible galaxy

Ansible galaxy là một trang web miễn phí để tìm kiếm, tải xuống, xếp hạng và xem xét tất cả các tính chất được cộng đồng Ansible phát triển. Trong lab này ta sẽ đi qua một số command sử dụng trong ansible galaxy.

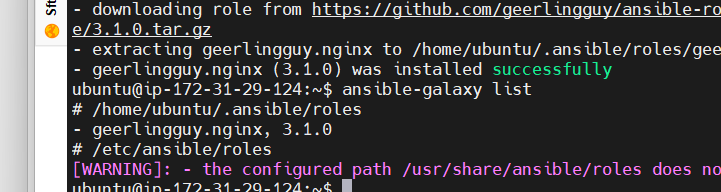
* Install Role:

ansible-galaxy install geerlingguy.nginx



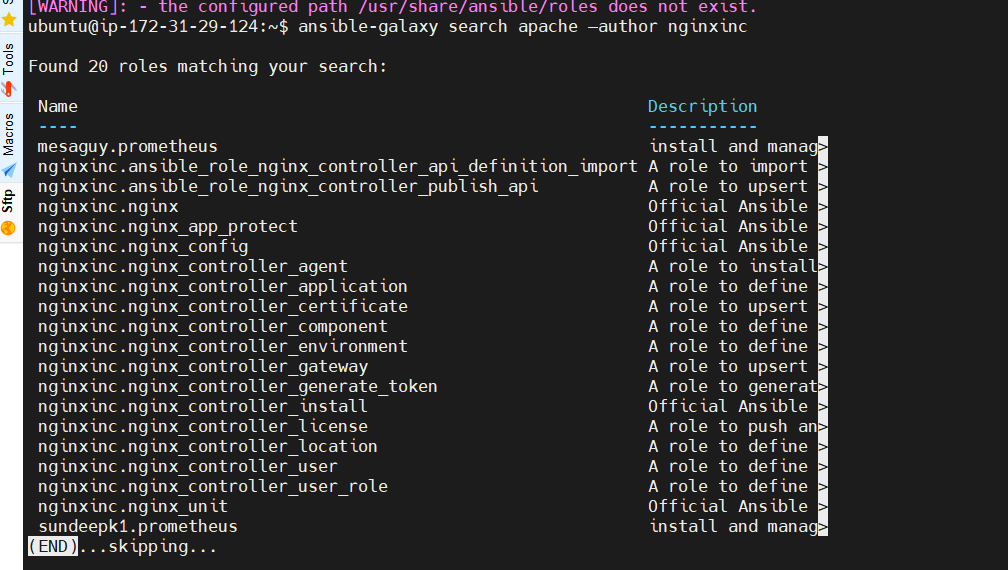
Để xem danh sách các role được install thì ta dùng lệnh sau:

ansible-galaxy list



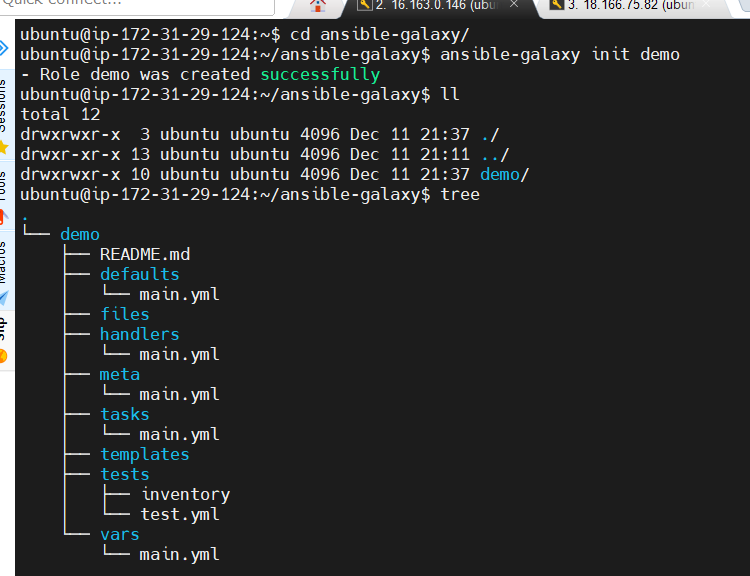
* Search for roles

ansible-galaxy search apache –author author\_name



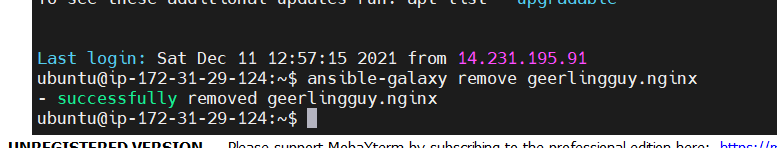
* Create role

ansible-galaxy init role\_name



* Remove role

ansible-galaxy remove username.role\_name



# Ansible Tower

Refer: <https://www.linuxtechi.com/install-ansible-awx-on-ubuntu/>

Objectives:

* Know Ansible, ansible inventory, ansible-playbook, ansible-galaxy, ansible tower.

Problem Descriptions:

Assumptions:

Technical Requirements:

Questions to answer:

Estimated Time to complete: 700 mins

**-- THE END --**