**GHI CHÉP HỌC ANSIBLE**

**Trước hết: Cài pip thông qua packages:**

curl "https://bootstrap.pypa.io/get-pip.py" -o "get-pip.py"

python get-pip.py ,

Muốn cài pip3 thì dùng lệnh python3 get-pip.py

**1. Khái niệm**

Ansible là 1 CM tool, bổ trợ cho quá trình automation. Để có thể sử dụng được ansible thì các host phải có ssh và python.

Trong ansible cần nắm rõ:

- các file cấu hình: chú trọng vào file hosts là chính, file này có chức năng khai báo các host hay nhóm host để ansible có thể nhận diện được nó để thực thi các công việc.

- ansible-playbook: Là 1 tập hợp danh sách các công việc được thực hiện trên các client hay nhóm client. Được viết dưới dạng các file yaml (.yml)

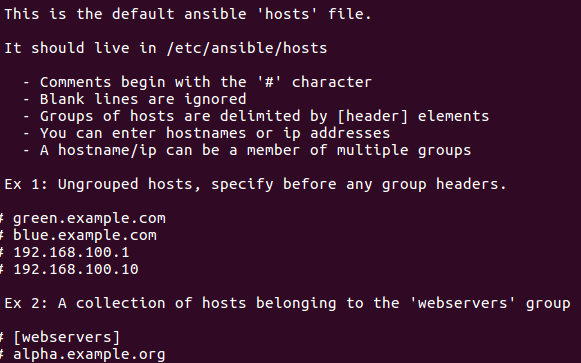
- ansible-roles: Cũng thực hiện tạo các danh sách công việc cần làm trên những client nhất định giống như ansible-playbook nhưng nó được chức năng hóa theo các cây thư mục.

**2. File cấu hình khai báo:**

File hosts:

[<group\_name 1>]

IPs\_of\_group1 or domain name



**CÁC CÁCH THỰC THI ANSIBLE**

Để thực thi các nhiệm vụ trong ansible chúng ta có 3 cách:

+ Cách 1: dùng command

+ Cách 2: Dùng playbook

+ Cách 3: Sử dụng ansible roles

**COMMAND**

Cấu trúc thực hiện ansible dử dụng command như sau:

ansible <name\_of\_group/IP> -m <module\_name> -a <argument>

Ex: ansible webservers -m service -a "name=httpd state=restarted

**3. ANSIBLE\_PLAYBOOK:**

***3.1 Introduction***

Lợi ích:

+ Tách biệt các hệ thống, 1 hệ thống là 1 project riêng biệt

+ Việc tách role giúp dễ dàng quản lý, phát triển

+ Tái sử dụng tốt, chỉ cần sửa thông tin host, vars

Như ta đã biết playbook là 1 danh sách các công việc được thực hiện trên các thực thể cần quản lý cho nên phải có dấu "-" trước tất cả các variable có cùng level.

---

- hosts: webservers

vars:

http\_port: 80

max\_clients: 200

tasks:

- name: ensure apache is at the latest version

yum: pkg=httpd state=latest

- name: write the apache config file

template:

src=/srv/httpd.j2

dest=/etc/httpd.conf

notify:

- restart apache

- name: ensure apache is running (and enable it at boot)

service: name=httpd state=started enabled=yes

handlers:

- name: restart apache

service: name=httpd state=restarted

Giải thích:

+ hosts: xác định đối tượng sẽ thực thi playbook này.

+ vars: các biến dùn trong play, trong ví dụ này các biến sẽ được dùng để cấu hình apache

+ tasks: liệt kê các task cần thực hiện

+ name: tên của task

+ yum, template, service: các module sử dụng

+ notify: giống như trigger, để gọi đến 1 task khác khi task hiện tại thực hiện thành công.

+ handlers: khai báo các task và sẽ được thực thi sau

1 play gồm danh sách các task được thực thi theo thứ tự từ trên xuống. Nếu xảy ra lỗi ở task nào thì host đang thực thi sẽ bị dừng lại mà không ảnh hưởng đến các host khác.

Cấu trúc 1 playbooks chuẩn:

|  |
| --- |
| production # inventory file for production servers  stage # inventory file for stage environment  group\_vars/  group1 # here we assign variables to particular groups  group2 # ""  host\_vars/  hostname1 # if systems need specific variables, put them here  hostname2 # ""  library/ # if any custom modules, put them here (optional)  filter\_plugins/ # if any custom filter plugins, put them here (optional)  site.yml # master playbook  webservers.yml # playbook for webserver tier  dbservers.yml # playbook for dbserver tier  roles/  common/ # this hierarchy represents a "role" tasks/ #  main.yml # <-- tasks file can include smaller files if warranted  handlers/ #  main.yml # <-- handlers file  templates/ # <-- files for use with the template resource  ntp.conf.j2 # <------- templates end in .j2  files/ # bar.txt # <-- files for use with the copy resource  foo.sh # <-- script files for use with the script resource  vars/ # main.yml # <-- variables associated with this role  defaults/ # main.yml # <-- default lower priority variables for this role meta/ main.yml # <-- role dependencies  vars/ #  webtier/ # same kind of structure as "common" was above, done for the webtier role |

Trong đó:

* *production*: giống file /etc/ansible/hosts, liệt kê group, host
* *group\_vars/*\*: đặt các biến chung cho cùng 1 nhóm, ví dụ [webservers] có biến listen\_port: 80
* *host\_vars/*\*: đặt các biến riêng cho từng host
* *roles/*\*: đặt các role, ví dụ các host trong [webservers] gọi đến role webtier

***3.2 Hướng dẫn sử dụng***

<http://www.familug.org/2016/12/ansible-playbook.html>

Playbook là 1 file yml được viết bằng ngôn ngữ yaml. Đối với ansible trong file playbook nhất định phải khai báo thực thi trên host nào thông qua từ khóa "hosts".

Chạy 1 playbook chúng ta dùng lệnh:

ansible-playbook <playbook\_name or path+name> -i <hosts\_name>

# nếu trong playbook đã khai báo hosts rồi thì không cần tùy chọn -i. Mặc định nếu không có -i thì ansible sẽ vào /etc/ansible/hosts để kiếm, chúng ta cũng có thể tạo 1 file hosts ở đường dẫn bất kì và dùng tùy chọn -i để chỉ tới đó.

modules quan trọng trong ansible:

https://github.com/hocchudong/ghichep-ansible/blob/master/Module\_Ansible/Database/Overveiw\_and\_introduce\_about\_postgreSQL.md

Ví dụ 1 playbook đơn giản:

- Trong file hosts có định nghĩa nhóm host như sau:

[ubuntu]

172.17.0.2

192.168.0.5

- Thực thi playbook:

---

- hosts: ubuntu# Có thể thay thế bằng địa chỉ ip (Nhưng chỉ được 1 địa chỉ)

remote\_user: ubuntu # sử dụng user này đỉa thực thi nhiệm vụ.

tasks: # từ khóa bắt đầu để chỉ ra các nhiệm vụ sẽ được thực hiện

- name: Install apache2 # Tên nhiệm vụ thứ nhất

apt: name=apache2 state=present update\_cache=yes

become: true

become\_user: root

# Thực hiện câu lệnh apt-get install apache2 dưới quyền root.

**ANSIBLE ROLES**

Sử dụng câu lệnh sau để tạo nhanh 1 ansible-roles:

Ansible-galaxy init <roles\_name>

Để chạy 1 ansible-roles sử dụng lệnh:

ansible-playbook -i "192.168.0.10" --role "path to role"

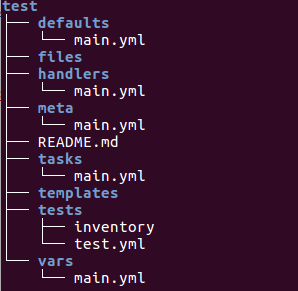
Hoặc có thể khai báo 1 file playbook như sau:

hosts: ubuntu

roles:

- name\_of\_roles

Mô hình của ansible-roles:

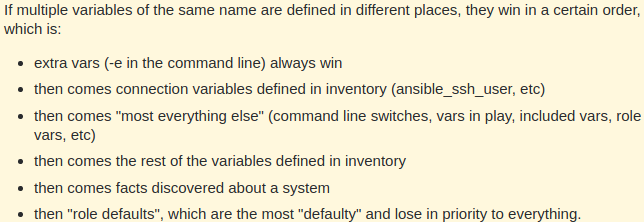


tasks: Bao gồm các nhiệm vụ chính sẽ được thực thi bởi ansible-roles.

handlers: Bao gồm các xử lí. Các xử lí này tương tự như chúng ta khai báo hàm. Và để gọi chúng thì chúng ta sử dụng từ khóa "notify: name\_of\_handlers".

defaults: Các giá trị mặc định cho các giá trị biến trong roles.

vars: Khai báo các giá trị cho các biến trong roles. nếu trong vars và defaults có cùng tên biến thì sẽ ưu tiên lấy trong vars.



files: Bao gồm các file mà chúng ta có thể deploy qua roles.

Gỉa sử chúng ta cần copy 1 file từ server về host mà không muốn chỉ ra đường dẫn của file đó. Chúng ta sẽ copy file đó bỏ vào thư mục này. Khi thực thi tasks thì chỉ cần nêu ra tên file với phần mở rộng của nó thì ansible-roles sẽ vào đây để kiếm.

Inventory:

+ File iventory để giúp Ansible biết các server mà nó cần kết nối sử dụng SSH , thông tin kết nối nó yêu cầu và các tùy chọn biến gắn liền với các server này.

File inventory có định dạng là INI. Trong file inventory, chúng ta có thể chỉ định nhiều hơn một máy chủ và gom chúng thành nhiều nhóm.

meta: .

**MÔ HÌNH THỬ NGHIỆM ANSIBLE**

|  |
| --- |
| ---  - name: Create a new Demo EC2 instance  hosts: local  connection: local  gather\_facts: False    vars:  region: ap-southeast-2  instance\_type: t2.micro  ami: ami-33ab5251 # Ubuntu 16.04 LTS  key\_name: test1 # pem file name    tasks:  - name: Amazon EC2 | Create Key Pair  ec2\_key:  name: "{{ key\_name }}"  region: "{{ region }}"  key\_material: "{{ item }}"  with\_file: /home/tuyet/.ssh/id\_rsa.pub  handlers:  - name: Create an ec2 instance  ec2:  key\_name: "{{ key\_name }}"  group: default # security group name  instance\_type: "{{ instance\_type}}"  image: "{{ ami }}"  wait: true  region: "{{ region }}"  count: 1 # default  count\_tag:  Name: Demo  instance\_tags:  Name: Demo  vpc\_subnet\_id: subnet-6ed44b09  assign\_public\_ip: yes  instance\_tags:  Name: "My-EC2-Instance"  register: ec2      - name: Configuration Instance  hosts: ec2-instance  become\_user:  tasks:  - name: Install apache2  apt: name=apache2 state=present update\_cache=yes  become: true  become\_user: root    - action: ec2\_facts  handlers:  - name: terminating single instance  local\_action:  module: ec2  state: 'absent'  region: ap-southeast-2  instance\_ids: "{{ ansible\_ec2\_instance\_id }}" |