

Ansible

1.	Install Ansible.....	2
2.	Ansible Inventory	3
3.	Ansible playbook.....	5
4.	Role.....	10
5.	Ansible galaxy	12
6.	Ansible Tower	13

1. Install Ansible

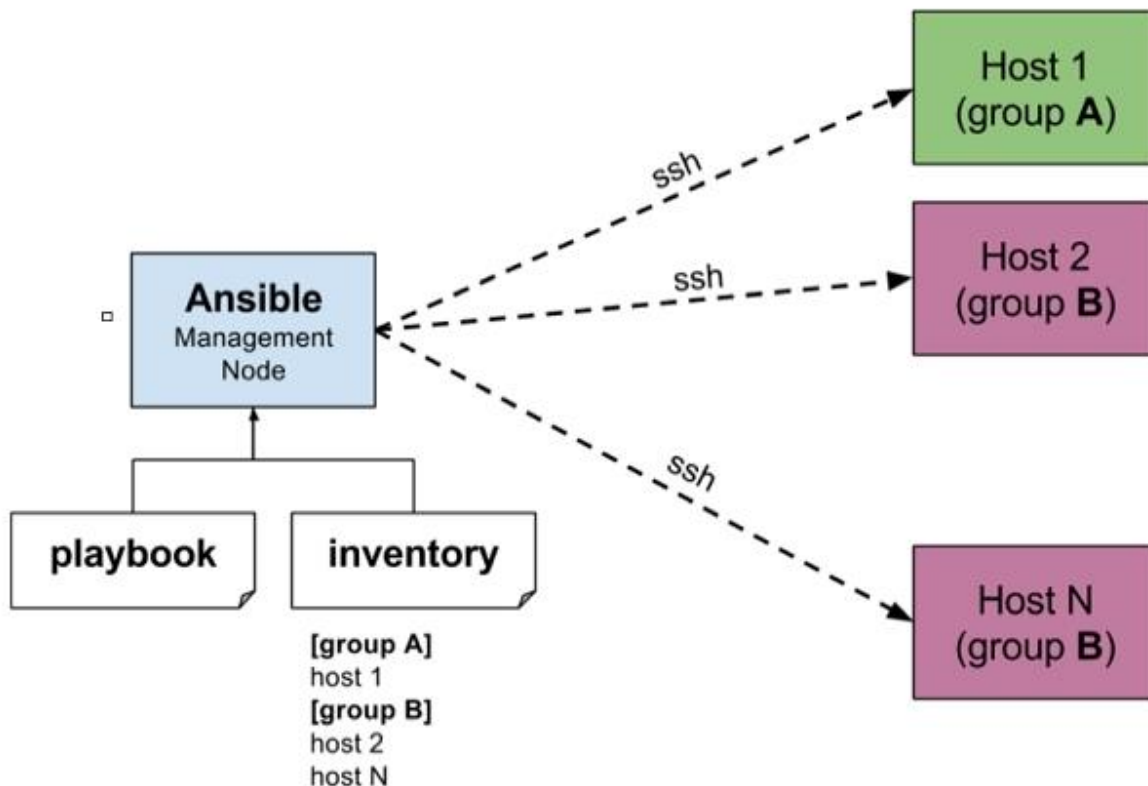
Cài đặt Ansible trên Ubuntu:

```
sudo apt update
sudo apt install software-properties-common
sudo add-apt-repository --yes --update ppa:ansible/ansible
sudo apt install ansible
```

Có nhiều cách 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.

```
ubuntu@ip-172-31-29-124:/etc/ansible$ ll
total 64
drwxr-xr-x  3 root root 4096 Dec 11 13:06 ./
drwxr-xr-x 116 root root 4096 Nov 23 15:19 ../
-rw-r--r--  1 root root 1675 Nov 11 12:20 Training.pem
```

Mô hình hoạt động của Ansible:



2. 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:

```
ubuntu@ip-172-31-29-124:/etc/ansible$ ansible web_app -m ping --private-key=Training.pem
172.31.13.222 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
ubuntu@ip-172-31-29-124:/etc/ansible$
```

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 không cần khai báo lại private-key:

```
ubuntu@ip-172-31-29-124:/etc/ansible$ ansible web_app -m ping
172.31.13.222 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
```

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....

3. 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
ubuntu@ip-172-31-29-124:/etc/ansible$ ansible-playbook  playbook.yml

PLAY [web_app] *****

TASK [Gathering Facts] *****
ok: [172.31.13.222]

TASK [Ping host] *****
ok: [172.31.13.222]

TASK [Install nginx] *****
changed: [172.31.13.222]

PLAY RECAP *****
172.31.13.222      : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0

ubuntu@ip-172-31-29-124:/etc/ansible$
```

Kiểm tra xem nginx đã được cài trên web_app chưa:

```

ubuntu@ip-172-31-29-124:/etc/ansible$ curl 172.31.13.222
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
ubuntu@ip-172-31-29-124:/etc/ansible$

```

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:

```

ubuntu@ip-172-31-29-124:/etc/ansible$ ansible-playbook  playbook.yml

PLAY [web_app] *****

TASK [Gathering Facts] *****
ok: [172.31.13.222]

TASK [Update and upgrade apt packages] *****
ok: [172.31.13.222]

TASK [Copy file] *****
changed: [172.31.13.222]

TASK [Restart nginx] *****
changed: [172.31.13.222]

PLAY RECAP *****
172.31.13.222      : ok=4    changed=2    unreachable=0    failed=0

ubuntu@ip-172-31-29-124:/etc/ansible$ curl 172.31.13.222
<h1>Training DevOps 2022</h1>
ubuntu@ip-172-31-29-124:/etc/ansible$ █

```

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.

```
ubuntu@ip-172-31-29-124:/etc/ansible$ ansible-playbook  playbook.yml

PLAY [web_app] *****

TASK [Gathering Facts] *****
ok: [172.31.13.222]

TASK [Install Apache.] *****
ok: [172.31.13.222] => (item=apache2)
ok: [172.31.13.222] => (item=mysql-server)

TASK [deploy html file] *****
ok: [172.31.13.222]

PLAY RECAP *****
172.31.13.222      : ok=3    changed=0    unreachable=0    failed=0

ubuntu@ip-172-31-29-124:/etc/ansible$ curl 172.31.13.222
<h1>Training DevOps 2022</h1>
ubuntu@ip-172-31-29-124:/etc/ansible$
```

- 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:

```
ubuntu@ip-172-31-29-124:/etc/ansible$ cat index.html
<h1>Training DevOps 2022</h1>
<h2>{{variable}}</h2>
ubuntu@ip-172-31-29-124:/etc/ansible$
```

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

```
ubuntu@ip-172-31-29-124:/etc/ansible$ ansible-playbook  playbook.yml

PLAY [web_app] *****

TASK [Gathering Facts] *****
ok: [172.31.13.222]

TASK [Install Apache.] *****
ok: [172.31.13.222] => ( item=apache2)
ok: [172.31.13.222] => ( item=mysql-server)

TASK [deploy html file] *****
changed: [172.31.13.222]

ubuntu@ip-172-31-29-124:/etc/ansible$ curl 172.31.13.222
<h1>Training DevOps 2022</h1>
<h2>Test var Ansible</h2>
ubuntu@ip-172-31-29-124:/etc/ansible$
```

4. 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:

```
ubuntu@ip-172-31-29-124:/etc/ansible$ tree
.
├── Training.pem
├── ansible.cfg
├── hosts
├── index.html
├── playbook.yml
├── roles
│   └── web
│       ├── handlers
│       │   └── main.yml
│       └── tasks
│           └── main.yml
4 directories, 7 files
```

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.

```
ubuntu@ip-172-31-29-124:/etc/ansible$ ansible-playbook playbook.yml

PLAY [web_app] *****

TASK [Gathering Facts] *****
ok: [172.31.13.222]

TASK [include web role] *****

TASK [web : Install Apache.] *****
ok: [172.31.13.222] => ( item=apache2)
ok: [172.31.13.222] => ( item=mysql-server)

TASK [web : deploy html file] *****
ok: [172.31.13.222]

PLAY RECAP *****
172.31.13.222      : ok=3    changed=0    unreachable=0    failed=0    skipped=0

ubuntu@ip-172-31-29-124:/etc/ansible$ curl 172.31.13.222
<h1>Training DevOps 2022</h1>
```

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

5. 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
```

```
ubuntu@ip-172-31-29-124:~$ ansible-galaxy install geerlingguy.nginx
Starting galaxy role install process
- downloading role 'nginx', owned by geerlingguy
- downloading role from https://github.com/geerlingguy/ansible-role-nginx/archive/3.1.0.tar.gz
- extracting geerlingguy.nginx to /home/ubuntu/.ansible/roles/geerlingguy.nginx
- geerlingguy.nginx (3.1.0) was installed successfully
ubuntu@ip-172-31-29-124:~$
```

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

```
ansible-galaxy list
```

```
ubuntu@ip-172-31-29-124:~$ ansible-galaxy list
# /home/ubuntu/.ansible/roles
- geerlingguy.nginx, 3.1.0
# /etc/ansible/roles
```

- Search for roles

```
ansible-galaxy search apache --author author_name
```

```
ubuntu@ip-172-31-29-124:~$ ansible-galaxy search apache --author nginxinc

Found 20 roles matching your search:

Name                                     Description
----                                     -
mesaguy.prometheus                     install and manage
nginxinc.ansible_role_nginx_controller_api_definition_import A role to import
nginxinc.ansible_role_nginx_controller_publish_api           A role to upsert
nginxinc.nginx                                           Official Ansible
nginxinc.nginx_app_protect                               Official Ansible
nginxinc.nginx_config                                   Official Ansible
nginxinc.nginx_controller_agent                         A role to install
nginxinc.nginx_controller_application                   A role to define
nginxinc.nginx_controller_certificate                   A role to upsert
nginxinc.nginx_controller_component                     A role to define
nginxinc.nginx_controller_environment                   A role to define
nginxinc.nginx_controller_gateway                       A role to upsert
```

- Create role

ansible-galaxy init role_name

```
ubuntu@ip-172-31-29-124:~/ansible-galaxy$ ansible-galaxy init demo
- Role demo was created successfully
ubuntu@ip-172-31-29-124:~/ansible-galaxy$ ll
total 12
drwxrwxr-x  3 ubuntu ubuntu 4096 Dec 11 21:37 ./
drwxr-xr-x 13 ubuntu ubuntu 4096 Dec 11 21:11 ../
drwxrwxr-x 10 ubuntu ubuntu 4096 Dec 11 21:37 demo/
ubuntu@ip-172-31-29-124:~/ansible-galaxy$ tree
.
├── demo
│   ├── README.md
│   ├── defaults
│   │   └── main.yml
│   ├── files
│   ├── handlers
│   │   └── main.yml
│   ├── meta
│   │   └── main.yml
│   ├── tasks
│   │   └── main.yml
│   ├── templates
│   ├── tests
│   │   ├── inventory
│   │   └── test.yml
│   └── vars
│       └── main.yml
```

- Remove role

ansible-galaxy remove username.role_name

```
ubuntu@ip-172-31-29-124:~$ ansible-galaxy remove geerlingguy.nginx
- successfully removed geerlingguy.nginx
ubuntu@ip-172-31-29-124:~$ █
```

6. Ansible Tower

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