Mục lục

[I. Bài toán đặt ra 2](#_Toc28012226)

[II. Ansible là gì 2](#_Toc28012227)

[III. Ansible được dùng khi nào ? 3](#_Toc28012228)

[IV. Ansible – có gì khó? 3](#_Toc28012229)

[V. Ansible phù hợp với ai 4](#_Toc28012230)

[VI. Ansible – nên tiếp cận như thế nào? 4](#_Toc28012231)

[VII. Tại sao cần dùng Ansible 5](#_Toc28012232)

[VIII. THÀNH PHẦN TRONG ANSIBLE 5](#_Toc28012233)

[8.1 YAML 6](#_Toc28012234)

[8.2 PLAYBOOK 7](#_Toc28012235)

[8.2.1 Ví dụ: 7](#_Toc28012236)

[8.2.2 Các phần chính 8](#_Toc28012237)

[8.2.3 Chạy 1 Playbook 8](#_Toc28012238)

[8.3 inventory file 9](#_Toc28012239)

[8.3.1 Ví dụ 9](#_Toc28012240)

[8.3.2 Inventory file cơ bản 10](#_Toc28012241)

[8.3.3 Group trong inventory file 10](#_Toc28012242)

[8.3.4 Regular expression trong inventory file 11](#_Toc28012243)

[8.3.5 Ghi đè tham số cấu hình bằng inventory file 11](#_Toc28012244)

[8.4 Modules 12](#_Toc28012245)

[8.5 Kiến thức bổ trợ 13](#_Toc28012246)

[IX. Mô hình lab 14](#_Toc28012247)

# I. Bài toán đặt ra

Ngày nay, đi đâu chúng ta cũng bắt gặp cụm từ **‘‘tự động hóa ’’**

Trên hệ thống, cụm từ này xuất hiện càng nhiều.

Cách đấy không quá lâu, khoảng gần 10 năm trước, việc cài đặt các gói phần mềm, triển khai các hệ thống từ đơn giản cho đến phức tạp đều do một tay người quản trị hệ thống làm. Và họ làm chủ yếu là thủ công - tự mình cài đặt từng bước một và đôi khi có sự hỗ trợ của script.

Điều này dẫn tới việc nếu hệ thống càng lớn, việc triển khai càng chậm, việc sửa đổi, update hệ thống sau này sẽ dẫn tới vấn đề là thời gian downtime kéo dài, gây ngưng trệ hệ thống từ đó doanh thu công ty, doanh nghiệp bị giảm. Thêm vào đó việc triển khai thủ công rất dễ gặp lỗi vì rất có thể sysadmin sơ xót quên 1 câu lệnh, nhầm thứ tự câu lệnh hay sai version các package.

Vì những lý do đó mà các Automation tool được ra đời và Ansible là một trong số đó.

# II. Ansible là gì

Ansible là một công cụ tự động hóa trong lĩnh vực IT. Nó có thể cấu hình các hệ thống, triển khai phần mềm và điều phối các tác vụ CNTT nâng cao hơn như triển khai liên tục ( continuous deployments) hoặc cập nhật hệ thống với mức downtime nhỏ nhất.

Ansible làm việc thông qua giao thức SSH

[**Tác giả ban đầu**](https://www.google.com/search?rlz=1C1CHBF_viVN869VN869&sxsrf=ACYBGNRa8RsZb5Jz_TIPHycL7OOQgPyCJQ:1577083402600&q=ansible+t%C3%A1c+gi%E1%BA%A3+ban+%C4%91%E1%BA%A7u&sa=X&ved=2ahUKEwjOwPi_lcvmAhWJ4nMBHe6rCzgQ6BMoADAWegQIDBAN)**:**Michael DeHaan

[**Ngày phát hành đầu tiên**](https://www.google.com/search?rlz=1C1CHBF_viVN869VN869&sxsrf=ACYBGNRa8RsZb5Jz_TIPHycL7OOQgPyCJQ:1577083402600&q=ansible+ng%C3%A0y+ph%C3%A1t+h%C3%A0nh+%C4%91%E1%BA%A7u+ti%C3%AAn&sa=X&ved=2ahUKEwjOwPi_lcvmAhWJ4nMBHe6rCzgQ6BMoADAXegQIDBAQ)**:**20 tháng 2, 2012; 7 năm trước

[**Bản phát hành ổn định**](https://www.google.com/search?rlz=1C1CHBF_viVN869VN869&sxsrf=ACYBGNRa8RsZb5Jz_TIPHycL7OOQgPyCJQ:1577083402600&q=ansible+b%E1%BA%A3n+ph%C3%A1t+h%C3%A0nh+%E1%BB%95n+%C4%91%E1%BB%8Bnh&sa=X&ved=2ahUKEwjOwPi_lcvmAhWJ4nMBHe6rCzgQ6BMoADAYegQIDBAT)**:**2.9.1 / 14 tháng 11, 2019;

[**Nhà phát triển**](https://www.google.com/search?rlz=1C1CHBF_viVN869VN869&sxsrf=ACYBGNRa8RsZb5Jz_TIPHycL7OOQgPyCJQ:1577083402600&q=ansible+nh%C3%A0+ph%C3%A1t+tri%E1%BB%83n&sa=X&ved=2ahUKEwjOwPi_lcvmAhWJ4nMBHe6rCzgQ6BMoADAZegQIDBAW)**:**Ansible Community / Ansible Inc. / Red Hat Inc.

[**Viết bằng**](https://www.google.com/search?rlz=1C1CHBF_viVN869VN869&sxsrf=ACYBGNRa8RsZb5Jz_TIPHycL7OOQgPyCJQ:1577083402600&q=ansible+vi%E1%BA%BFt+b%E1%BA%B1ng&stick=H4sIAAAAAAAAAOPgE-LSz9U3yDYoq8pK0lLKKLfST87PyUlNLsnMz9Mvzk8rKU8sSrUqL8osKUnNU8jMW8QqmphXnJmUk6pQlvlw1_4ShaSHuzbmpQMAC-ccvkoAAAA&sa=X&ved=2ahUKEwjOwPi_lcvmAhWJ4nMBHe6rCzgQ6BMoADAaegQIDBAZ)**:**[Python](https://www.google.com/search?rlz=1C1CHBF_viVN869VN869&sxsrf=ACYBGNRa8RsZb5Jz_TIPHycL7OOQgPyCJQ:1577083402600&q=Python&stick=H4sIAAAAAAAAAOPgE-LSz9U3yDYoq8pKUuIAsU2rDOO1lDLKrfST83NyUpNLMvPz9Ivz00rKE4tSrcqLMktKUvMUMvMWsbIFVJZk5OcBAOED_wFFAAAA&sa=X&ved=2ahUKEwjOwPi_lcvmAhWJ4nMBHe6rCzgQmxMoATAaegQIDBAa), [PowerShell](https://www.google.com/search?rlz=1C1CHBF_viVN869VN869&sxsrf=ACYBGNRa8RsZb5Jz_TIPHycL7OOQgPyCJQ:1577083402600&q=PowerShell&stick=H4sIAAAAAAAAAOPgE-LSz9U3yDYoq8pKUuIEsY3NLQ3KtJQyyq30k_NzclKTSzLz8_SL89NKyhOLUq3KizJLSlLzFDLzFrFyBeSXpxYFZ6Tm5AAAxZmgg0oAAAA&sa=X&ved=2ahUKEwjOwPi_lcvmAhWJ4nMBHe6rCzgQmxMoAjAaegQIDBAb), [Ruby](https://www.google.com/search?rlz=1C1CHBF_viVN869VN869&sxsrf=ACYBGNRa8RsZb5Jz_TIPHycL7OOQgPyCJQ:1577083402600&q=Ruby&stick=H4sIAAAAAAAAAOPgE-LSz9U3yDYoq8pKUuIAsc3S0ky1lDLKrfST83NyUpNLMvPz9Ivz00rKE4tSrcqLMktKUvMUMvMWsbIElSZVAgDZdx1ZQwAAAA&sa=X&ved=2ahUKEwjOwPi_lcvmAhWJ4nMBHe6rCzgQmxMoAzAaegQIDBAc)

# III. Ansible được dùng khi nào ?

* Đảm bảo việc quản lý cấu hình của thiết bị, ứng dụng một cách hiệu quả, tức là quản lý đơn giản, kiểm soát được các cấu hình đúng và đủ hay chưa, chúng chạy có chính xác hay không.
* Tiết kiệm được công sức & thời gian khi phải triển khai đi triển khai lại.
* Tái sử dụng được các bước triển khai trước đó (các bước lặp đi lặp lại khi cài đặt, cấu hình máy chủ, cấu hình ứng dụng)
* Tự động hóa và áp dụng hàng loạt các việc trên hoàng loạt các server, hàng loạt các ứng dụng với thời gian ngắn nhất.
* Ansible sẽ không phù hợp khi mà bạn chưa vững về các ứng dụng sẽ được bạn sử dụng ansible để quản lý. Do vậy, muốn làm được Ansible để tự động hóa việc triển khai, quản lý cấu hình cho ứng dụng nào đó thì hãy nắm chắc chắn về ứng dụng đó trước khi áp dụng.

# IV. Ansible – có gì khó?

* Nếu chưa có tư duy và thói quen tự động hóa (làm tự động các việc lặp đi lặp lại, thậm chí tự động một thứ trong tương lai sẽ làm lần đầu) thì khi tiếp cận độ khó tăng lên gấp đôi. Phải có thói quen tự động hóa trước đã, tức là trước đó không cần dùng ansible bạn đã phải dùng bash script hoặc python hoặc thứ tương tự để tự động các việc bạn phải làm. Thậm chí dùng người khác để tự động việc cho bạn :v.
* Ansible sẽ là khó nếu ngại đọc tài liệu hãng, ngại thực hành.
* Ansible sẽ là khó nếu không có kinh nghiệm về linux. Chủ yếu tiếp cận đầu tiên với ansible là những người làm về linux và phát huy tốt trên môi trường linux. Nếu muốn tìm hiểu tốt về Ansible thì hãy tìm hiểu linux trước.
* Ansible sẽ có nhiều thuật ngữ mới. Ngay cả từ khóa Ansible cũng là mới và trong tài liệu của nó cũng sẽ có những khái niệm mới. Với vấn đề này thì hiện nay lượng tài liệu về Ansible không phải là ít nữa rồi, tuy nhiên hãy tìm hiểu các thuật ngữ mới này như thuật ngữ về inventory, plays, playbooks, var, các module …. trước khi sử dụng.

# V. Ansible phù hợp với ai

* **System admin**: những người quản trị, triển khai các hệ thống ứng dụng trên môi trường Linux (là chủ yếu), Windows cũng có thể. Đặc điểm là công việc phải lặp đi lặp lại các nhiệm vụ triển khai, cấu hình, phân tích, đọc các log về vấn đề gặp phải.
* **Developer**: những nhà phát triển các ứng dụng, hay phải cài cắm mức cơ bản cho môi trường mà các sản phẩm các họ làm ra.
* **DevOps**: là người phải làm cả vai của lập trình viên và cả vai của system admin.

# VI. Ansible – nên tiếp cận như thế nào?

- Cần có kiến thúc nền đối Linux.Tức là phải tàm tạm mức OS của Linux và một số ứng dụng cơ bản được triển khai trên linux cần thành thạo một chút, ví dụ: SSH, SSH Keypair, Apache, Nginx, Database

- Tiếp theo là cần xây dựng một LAB tầm 03 máy chủ với các OS là CentOS7, CentOS6, Ubuntu Server 16.04, Ubuntu Server 18.04… để xây dựng một LAB có đủ môi trường để thực hành.

- Trước khi tìm hiểu ansible, tốt nhất là nên tự viết được hoặc đọc hiểu được các script bằng bash shell (cao hơn là các script bằng python) cho các ứng dụng cơ bản. Để từ đó ánh xạ sang và chuyển thể sang ngôn ngữ của Ansible

# VII. Tại sao cần dùng Ansible

* Ansible miễn phí và là 1 opensource
* Ansible sử dụng phương thức ssh
* Việc cài đặt không tốn nhiều tài nguyên
* Được phát triển bởi ngôn ngữ python. Nên nếu bạn muốn tạo thêm module thì cũng sử dụng bằng python
* Khá nhẹ và dễ setup
* Các sciprt thường được dùng định dạng YAML
* Và Ansible có một cộng đồng tương tác lớn

# VIII. ****THÀNH PHẦN TRONG ANSIBLE****

Có khá nhiều thành phần trong Ansible nhưng về cơ bản thì có các thành phần chính sau:

* Playbooks - Là nơi bạn sẽ khai báo kịch bản chạy cho server
* Tasks - Là những công việc nhỏ trong cuốn sổ Playbooks trên
* Inventory - Khai báo địa chỉ server cần được setup
* Modules - Những chức năng hỗ trợ cho việc thực thi tasks dễ và đang dạng

Tuy nhiên thứ đầu tiên cần nhắc đến là YAML bởi các playbook của Ansible đều được viết bằng định dạng YAML

## 8.1 YAML

* Ansible Sử dụng YAML để viết Playbook
* Các thành phần cơ bản:
  + Declarations (Tuyên bố/ thông báo)
  + Lists
  + Ascociative arrays (mảng liên kết)
* 1 declaration khá giống với 1 variable trong các ngôn ngữ khác

name: 'This is name'

* Để tạo 1 list, dùng '-'

- 'item1'

- 'item2'

List sẽ chạy theo thứ tự từ trên xuống

* YAML sử dụng thụt đầu dòng để phân chia "cha con" .

item:

name: TheName

location: TheLocation

* Ta có thể trộn chúng với nhau

people:

- name: Albert

number: +1000000000

country: USA

- name: David

number: +44000000000

country: UK

## 8.2 PLAYBOOK

* Là 1 trong những tính năng cốt lõi của Ansible
* Nói cho Ansible biết những gì cần thực hiện
* Giống 1 danh sách (**list**) các việc cần làm (**tasks**)
* Mỗi **task** liên kết đến 1 đoạn code gọi là **module**
* Playbook là các tệp YAML đơn giản, dễ đọc với con người
* Module có thể được viết bằng bất kỳ ngôn ngữ nào, miễn đầu ra ở định dạng JSON.
* 1 playbook có thể bao gồm nhiều task -> được thực hiện tuần tự

--> Hiểu đơn giản: Playbook cho phép ta nhập list các task cần làm -> thực hiện trên 1 hệ thống từ xa

### 8.2.1 Ví dụ:

#### 8.2.1.1 Một task

- name: Ensure the HTTPd package is installed

yum:

name: httpd

state: present

become: True

Mục đính : Cài đặt gói httpd

#### 8.2.1.2 Một Playbook

---

- hosts: centos7

remote\_user: root

tasks:

- name: Ensure the HTTPd package is installed

yum:

name: httpd

state: present

become: True

- name: Ensure the HTTPd service is enabled and running

service:

name: httpd

state: started

enabled: True

become: True

Mục đính : Cài đặt gói httpd và service sẽ được **enabled** và **started**

### 8.2.2 Các phần chính

**hosts**: liệt kê host hoặc group host mà ta muốn chạy task

+ Trường này bắt buộc

+ Ansible sẽ lấy host/group host được cung cấp ở playbook và tìm trong file inventory.

Nếu không có kết quả khớp -> bỏ qua các task cho host/group host này

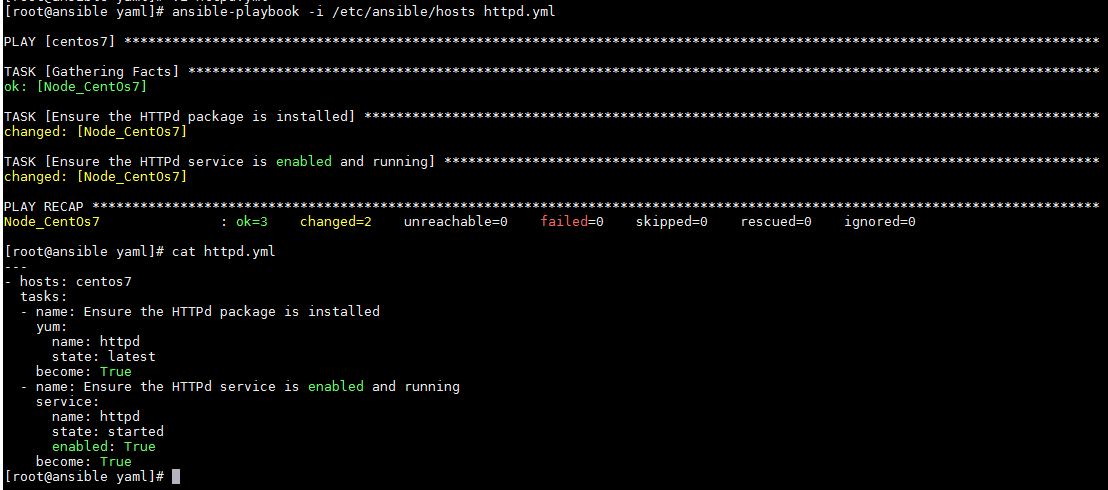
**remote\_user**: chỉ định user khi đăng nhập vào hệ thống

**tasks**: là 1 danh sách các hành động muốn thực hiện.

+ 1 task chứa tên task , 1 module, các đối số yêu cầu cho module

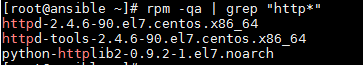
### 8.2.3 Chạy 1 Playbook

$ ansible-playbook -i path\_inventoryfile ten\_file.yml

[](https://camo.githubusercontent.com/e255c7252e9fada007df8f17c5b9499b5f40e2a5/68747470733a2f2f692e696d6775722e636f6d2f75764141754b792e706e67)

**changed** nghĩa là các thay đổi đã được thực hiện

Dòng cuối khi chạy playbook -> thống kê kết quả playbook đã thực thi Check trên node

[](https://camo.githubusercontent.com/5054d2c7fcb2c79c755b40fb95c42909eeadb1fa/68747470733a2f2f692e696d6775722e636f6d2f4b4d30424d77522e706e67)

Các task nếu đã changed rồi, trên client không có thay đổi gì thì nếu chạy lại playbook, task đó sẽ được bỏ qua (nguyên tắc Idempotency)

Nên giữ, viết các task đơn giản nhất có thể -> dể maintain, dễ đọc hiểu lại sau thời gian dài

## 8.3 inventory file

* inventory file tuân theo định dạng INI
* Nói cho Ansible biết danh sách các remote host mà nó có thể thử "động tới"
* Truyền inventory file cho ansible -> ansible search trong inventory các host/group sẽ được thực hiện task (ta khai báo trong Playbook) - > Chạy task
* Thêm tùy chọn **-i** hoặc **--inventory-file** theo sau là đường dẫn đến inventory file để truyền inventory file cho ansible trên cmdline

### 8.3.1 Ví dụ

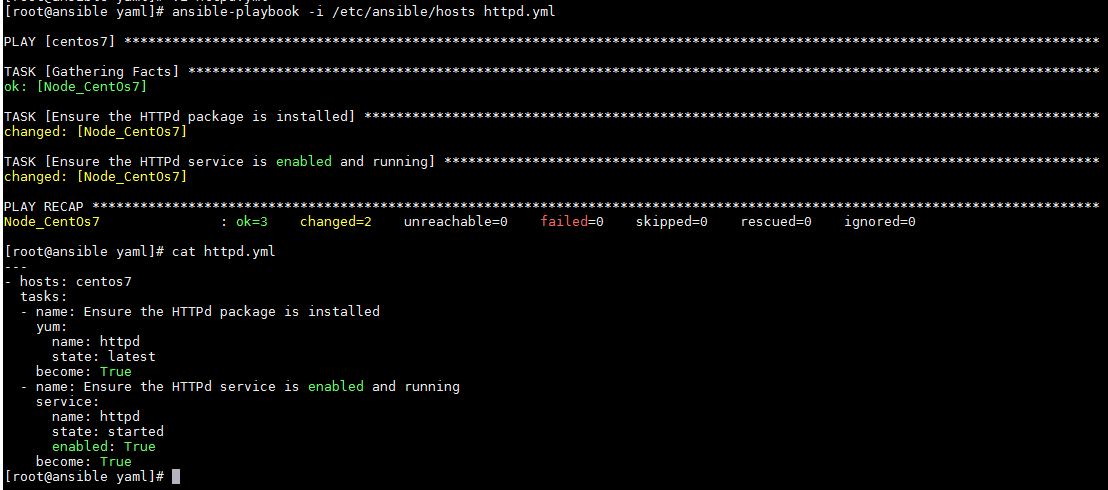
[centos7\_webserver]

Node\_CentOs7\_1 ansible\_host=192.168.91.171 ansible\_port=22 ansible\_user=root

Node\_CentOs7\_2 ansible\_host=192.168.91.172 ansible\_port=22 ansible\_user=root

[ubuntu16]

Node\_Ubuntu16 ansible\_host=192.168.91.173 ansible\_port=22 ansible\_user=khanh

[](https://camo.githubusercontent.com/e255c7252e9fada007df8f17c5b9499b5f40e2a5/68747470733a2f2f692e696d6775722e636f6d2f75764141754b792e706e67)

### 8.3.2 Inventory file cơ bản

**Mặc định inventory file**: /etc/ansible/hosts

* Có thể thêm FQDN hay IP

[centos7\_webserver]

Node\_CentOs7\_1 ansible\_host=192.168.91.171 ansible\_port=22 ansible\_user=root

Node\_CentOs7\_2 ansible\_host=192.168.91.172 ansible\_port=22 ansible\_user=root

[ubuntu16]

Node\_Ubuntu16 ansible\_host=192.168.91.173 ansible\_port=22 ansible\_user=khanh

### 8.3.3 Group trong inventory file

Ansible cho phép ta nhóm các máy chủ dựa trên vai trò (cùng chạy 1 hay nhiều task giống nhau) vào thành 1 nhóm

**Ví dụ:**

[centos7\_webserver]

Node\_CentOs7\_1 ansible\_host=192.168.91.171 ansible\_port=22 ansible\_user=root

Node\_CentOs7\_2 ansible\_host=192.168.91.172 ansible\_port=22 ansible\_user=root

[ubuntu16]

Node\_Ubuntu16 ansible\_host=192.168.91.173 ansible\_port=22 ansible\_user=khanh

Bình thường file yaml ta viết

---

- hosts: all

Giờ ta có thể viết lại thành

---

- hosts: centos7\_webserver

Để chỉ định rõ ràng group sẽ chạy tasks

### 8.3.4 Regular expression trong inventory file

Nêu số lượng máy chủ trong 1 group trở nên lớn -> việc viết ra sẽ dài dòng, phức tạp, khó kiểm soát

Ta có thể dùng cách sau:

[centos7\_webserver]

ws[01:20].root.io

[database]

db01.fale.io

### 8.3.5 Ghi đè tham số cấu hình bằng inventory file

Có thể ghi đè 1 số tham số cấu hình của Ansible qua inventory file. (các tham số cấu hình đặt trong ansible.cfg)

[centos7]

Node\_CentOs7 ansible\_host=192.168.91.171 ansible\_port=22 ansible\_user=root

[ubuntu16]

Node\_Ubuntu16 ansible\_host=192.168.91.172 ansible\_port=22 ansible\_user=khanh

Danh sách các tham số ta có thể ghi đè thông qua inventory file:

• ansible\_user : Tham số này được sử dụng để ghi đè người dùng được sử dụng để liên lạc với máy chủ từ xa. Đôi khi, một máy nhất định cần một người dùng khác, trong những trường hợp đó, biến này sẽ giúp bạn.

• ansible\_port : Tham số này sẽ ghi đè cổng SSH mặc định bằng cổng do người dùng chỉ định . Đôi khi sysadmin chọn chạy SSH trên một cổng không chuẩn. Trong trường hợp này, bạn sẽ cần hướng dẫn cho Ansible về sự thay đổi.

• ansible\_host : Tham số này được sử dụng để ghi đè máy chủ cho bí danh.

• ansible\_connection : Điều này chỉ định loại kết nối đến máy chủ từ xa. Các giá trị là SSH, Paramiko hoặc cục bộ

• ansible\_private\_key\_file : Tham số này sẽ ghi đè khóa riêng được sử dụng cho SSH; điều này sẽ hữu ích nếu bạn muốn sử dụng các khóa cụ thể cho một máy chủ cụ thể. Trường hợp sử dụng phổ biến là nếu bạn có máy chủ trải rộng trên nhiều trung tâm dữ liệu, nhiều vùng AWS hoặc các loại ứng dụng khác nhau. Khóa riêng có khả năng khác nhau trong các tình huống như vậy.

• ansible\_\_type : Theo mặc định, Ansible sử dụng sh shell ; bạn có thể ghi đè lên điều này bằng tham số ansible\_shell\_type . Thay đổi điều này thành csh , ksh , v.v. sẽ khiến Ansible sử dụng các lệnh của shell đó.

## 8.4 Modules

Ansible cung cấp rất nhiều module. Một vài module phổ biến thường dùng cho những thao tác đơn giản.

* System: Bao gồm các module như User, Group, Hostname, Systemd, Service, v.v...
* Commands: Thường có module con như Command, Expect, Raw, Script, Shell, v.v...
* Files: Các module làm việc với file như Copy, Find, Lineinfile, Replace, v.v...
* Database: Ansbile cũng support mạnh mẽ những module làm việc với DB như Mongodb, Mssql, Mysql, Postgresql, Proxysql, v.v...
* Cloud: Ansible cũng không quên kết hợp với các dịch vụ clound nổi tiếng như Amazon, Google, Docker, Linode, VMware, Digital Ocean, v.v...
* Windows: Mạnh mẽ với những module như win\_copy, win\_command, win\_domain, win\_file, win\_shell

Và còn hàng trăm module khác đã được ansible cung cấp sẵn.

Ta có thể tham khảo thêm tại doc của ansible:

<https://docs.ansible.com/ansible/latest/modules/modules_by_category.html>

## 8.5 Kiến thức bổ trợ

* Ansible hỗ trợ rất nhiều module để mình có thể viết playbooks. Chỉ cần search từ khóa + ansible là có thể tìm ra.
* Ansible Galaxy, một kho tàng các roles mà mọi người trên toàn thế giới có thể đóng góp. Bạn muốn triển khai một cái gì đó, hãy thử search trên Ansible Galaxy xem đã có chưa :))). “Ansible Galaxy” can either refer to a website for sharing and downloading Ansible roles, or a command line tool for managing and creating roles Chi tiết tại <https://galaxy.ansible.com/>
* Cấu trúc một playbook chuẩn

production # inventory file for production servers

staging # inventory file for staging 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

webtier/ # same kind of structure as "common" was above, done for the webtier role

monitoring/ # ""

fooapp/ # ""

# IX. Mô hình lab

**Node server**

* Centos 7
* IP: 192.168.91.170/24
* RAM: 2G

**Node client:**

* Centos 7
* IP: 192.168.91.171/24
* RAM: 1G

**Server đã trao đổi ssh public key với client**

**File inventory**

**[centos7]**

**Node\_CentOs7 ansible\_host=192.168.91.171 ansible\_port=22 ansible\_user=root**

**[ubuntu16]**

**Node\_Ubuntu16 ansible\_host=192.168.91.172 ansible\_port=22 ansible\_user=khanh**

**File playbook**

**---**

**- hosts: centos7**

**remote\_user: root**

**tasks:**

**- name: Install LAMP**

**yum:**

**name: '{{item}}'**

**state: present**

**with\_items:**

**- httpd**

**- mariadb-server**

**- mariadb**

**- php**

**- php-mysql**

**- php-fpm**

**- name: Ensure service enabled and started**

**service:**

**name: '{{item}}'**

**state: started**

**enabled: True**

**with\_items:**

**- mariadb**

**- httpd**

**- name: Ensure HTTP and HTTPS can pass the firewall**

**firewalld:**

**service: '{{item}}'**

**state: enabled**

**permanent: True**

**immediate: True**

**become: True**

**with\_items:**

**- http**

**- https**

**- name: Install php-gd, wget,rsync**

**yum:**

**name: '{{item}}'**

**state: present**

**with\_items:**

**- php-gd**

**- wget**

**- rsync**

**- name: Restart httpd**

**service:**

**name: httpd**

**state: restarted**

**- name: Down wordpress**

**get\_url:**

**url: http://wordpress.org/wordpress-5.3.2.tar.gz**

**dest: /root**

**- name: extract wordpress**

**unarchive:**

**src: /root/wordpress-5.3.2.tar.gz**

**dest: /root**

**remote\_src: yes**

**- name: rsync wordpress**

**shell: rsync -avP /root/wordpress/ /var/www/html/**

**- name: Create folder uploads**

**shell: mkdir /var/www/html/wp-content/uploads**

**- name: Set user:group**

**shell: chown -R apache:apache /var/www/html/\***

**- name: Install MySQL-python**

**yum:**

**name: MySQL-python**

**state: present**

**- name: Create database wordpress**

**mysql\_db:**

**name: wordpress**

**state: present**

**- name: Create user wordpressuser**

**mysql\_user:**

**name: wordpressuser**

**host: localhost**

**password: wordpresspassword**

**priv: 'wordpress.\*:ALL'**

**state: present**

**- name: Backup file config wp**

**shell: cp /var/www/html/wp-config-sample.php /var/www/html/wp-config.php**

**- name: Config WP1**

**replace:**

**path: /var/www/html/wp-config.php**

**regexp: 'database\_name\_here'**

**replace: 'wordpress'**

**- name: Config WP2**

**replace:**

**path: /var/www/html/wp-config.php**

**regexp: 'username\_here'**

**replace: 'wordpressuser'**

**- name: Config WP3**

**replace:**

**path: /var/www/html/wp-config.php**

**regexp: 'password\_here'**

**replace: 'wordpresspassword'**

**- name: Config version PHP1**

**yum:**

**name: '{{item}}'**

**state: present**

**with\_items:**

**- epel-release**

**- yum-utils**

**- http://rpms.remirepo.net/enterprise/remi-release-7.rpm**

**- name: Config version PHP2**

**shell: yum-config-manager --enable remi-php72**

**- name: Config version PHP3**

**yum:**

**name: '{{item}}'**

**state: present**

**with\_items:**

**- php**

**- php-common**

**- php-opcache**

**- php-mcrypt**

**- php-cli**

**- php-gd**

**- php-curl**

**- php-mysqlnd**

**- name: Restart Apache**

**service:**

**name: httpd**

**state: restarted**