# Bài tập cuối khóa Cloud Computing tại Viettel Network

Ngô Quang Dương – 17020191
K62, UET – VNU

Ngày 1 tháng 7 năm 2020

**Tóm tắt nội dung**

Trong báo cáo ngắn này, em sẽ trình bày quá trình chuẩn bị, cấu hình và ý tưởng của em để đạt được kết quả cho bài tập cuối khóa. Báo cáo được gửi kèm các file inventory, playbook, ...

## Mục lục

**1 Chuẩn bị** **2**
- 1.1 Tạo 2 máy ảo với AWS EC2 . . . 2
- 1.2 Key pairs và SSH config . . . 3

**2 Nội dung chính** **3**
- 2.1 File inventory . . . 3
- 2.2 Playbook tasks . . . 4
- 2.3 Mô hình lab . . . 5

**3 Bonus** **5**
- 3.1 Dynamic inventory . . . 5
- 3.2 Makefile . . . 6
- 3.3 Uninstall . . . 6
- 3.4 Ansible Vault . . . 6

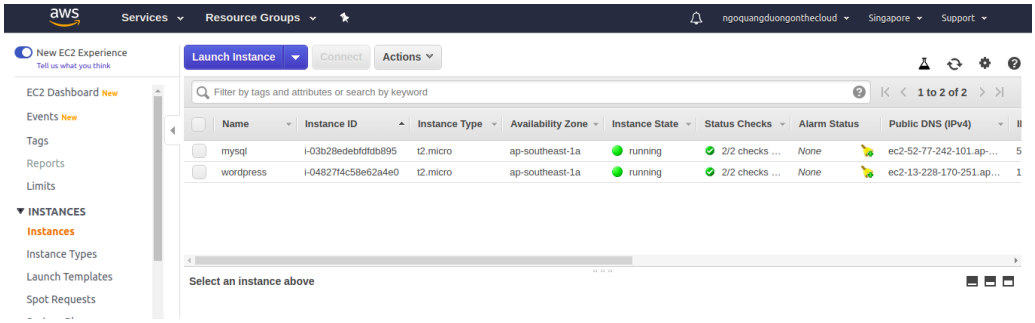
# 1 Chuẩn bị

Ý tưởng của em là sử dụng **AWS EC2** để tạo hai máy ảo làm *managed nodes*. Thông qua **Ansible**, em sẽ cài đặt **Docker** và các *images* lên hai máy đó. https://www.youtube.com/watch?v=2BPx4nPgP

Quá trình chạy *playbook* được ghi lại trong YouTube (lưu ý cần bật subtitle).

Mã nguồn của bài tập được lưu tại GitHub.

## 1.1 Tạo 2 máy ảo với AWS EC2




Hình 1: Hai máy ảo EC2

Thông tin chi tiết hơn về 2 máy này:

- Máy 1
  - Tên: mysql
  - Username mặc định: ubuntu
  - Chạy trên Ubuntu 18.04
  - Dùng để cài mysql image
  - Cho phép kết nối vào cổng 22 (SSH) và 3306 (MYSQL)
- Máy 2
  - Tên: wordpress
  - Username mặc định: ubuntu
  - Chạy trên Ubuntu 18.04
  - Dùng để cài wordpress image
  - Cho phép kết nối vào cổng 22 (SSH) và 80 (HTTP)

### 1.2 Key pairs và SSH config

**AWS EC2** cho phép kết nối đến các máy ảo thông qua **SSH**. Trước khi các máy ảo được khởi tạo, **AWS EC2** tạo ra một file `.pem` – cần tải file này về, đặt vào thư mục `~/.ssh/`.

Để cho đơn giản, cả hai máy trên dùng chung một file `.pem`

SSH config có nội dung như sau

```
Host mysql
    HostName public_ip_of_mysql_instance
    User ubuntu
    Port 22
    IdentitiesOnly yes
    IdentityFile path/to/pem

Host wordpress
    HostName public_ip_of_wordpress_instance
    User ubuntu
    Port 22
    IdentitiesOnly yes
    IdentityFile path/to/pem
```

File sshconfig tuy không được sử dụng bởi **Ansible** nhưng cần thiết để lệnh kết nối được viết ra nhanh chóng.

## 2 Nội dung chính

GitHub repo không lưu file inventory.

Inventory được tạo ra từ file `inventory_generator.py` (do public IP của máy ảo không cố định)

### 2.1 File inventory

```
[homework]

[mysql]
public_ip_of_mysql_instance

[wordpress]
public_ip_of_wordpress_instance

[homework:children]
mysql
wordpress

[homework:vars]
```

```
ansible_python_interpreter=/usr/bin/python
ansible_ssh_user=ubuntu
ansible_ssh_private_key_file=~/.ssh/homework.pem
```

Trong đó, `homework` là group chứa cả hai *managed nodes*.

Hai *managed nodes* được truy cập đến qua **SSH** với file key được chỉ định trong biến `ansible_ssh_private_key_file`.

`homework` chứa hai group con `mysql`, `wordpress`. Mỗi group này lần lượt gồm host của *managed nodes* được dùng để cài đặt **MySQL** và **Wordpress** image. IP được cung cấp trong hai group này là public IP v4 của hai máy ảo.

Mục đích của việc tạo group con là để IP được cô lập trong file *inventory* và khi sử dụng *playbook* thì chỉ cần đến tên group chứa IP đó.

## 2.2 Playbook tasks

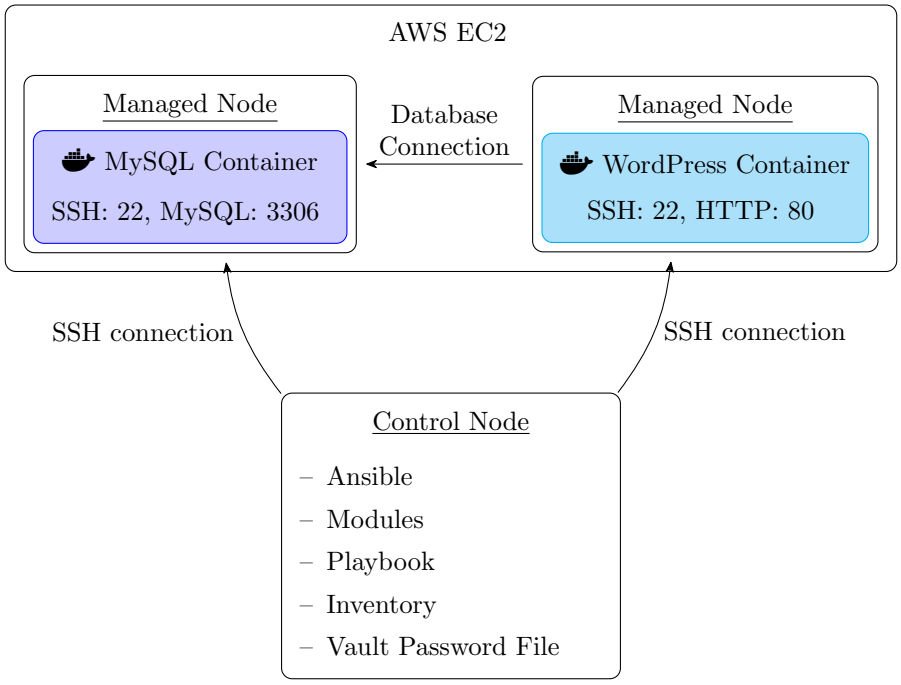
File playbook là `homework.yaml`

Việc cài đặt **Docker** được chia làm các tasks sau (thực hiện trên cả 2 nodes):

- Cài các packages: `apt-transport-https`, `ca-certificates`, `curl`, `gnupg-agent`, `software-properties-common`
- Thêm GPG key của **Docker**
- Thêm **Docker** repository
- `apt update`
- `dpkg -configure -a`
- `apt upgrade`
- Cài các packages: `docker-ce`, `docker-ce-cli`, `containerd.io`
- Tạo group docker
- Thêm user ubuntu và group docker
- Khởi động lại cả 2 nodes

Sau khi các tasks chung trên hoàn tất, *playbook* thực hiện các task riêng cho từng node. Cụ thể là pull image, sau đó chạy container.

### 2.3 Mô hình lab




Hình 2: Mô hình lab

## 3 Bonus

### 3.1 Dynamic inventory

Các máy ảo **EC2** có một đặc điểm là public IP sẽ được reset mỗi khi tắt đi – bật lại (nhưng restart thì không). Điều này dẫn tới một điều bất tiện là phải copy paste public ip mới vào file inventory mỗi khi thực hiện việc tắt đi bật lại.

Để khắc phục điều này, em sử dụng **Python** và **AWS SDK** dành cho **Python** để tự động lấy public IP về. Em đã chuẩn bị:

- Cài AWS SDK cho Python – Boto3
- Thêm region của máy ảo vào file `~/.aws/config`

```
[default]
region = ap-southeast-1
```

- Thêm access key id và access key secret vào file `~/.aws/credentials`

```
[default]
aws_access_key_id=ACCESS_KEY_ID
aws_secret_access_key=SECRET_KEY
```

- Viết Python script tạo inventory.

Trước khi chạy *playbook*, file `inventory_generator.py` sẽ được chạy trước để tạo ra file *inventory*.

## 3.2 Makefile

Một vấn đề khác em thấy khi dùng **Ansible** là lệnh dài. Điều này có thể được khắc phục hoàn toàn bằng cách ghi lại lệnh vào **Makefile**, một tiện ích có sẵn trên **Linux**.

Với **Makefile**, em có thể áp đặt phải chạy file `inventory_generator.py` trước khi chạy `playbook`.

```
all:

FORCE:

homework.ini: FORCE
    python3 ./inventory_generator.py > homework.ini

ping: homework.ini
    ansible homework -i ./homework.ini -m ping

play: homework.ini
    ansible-playbook -i ./homework.ini --vault-password-file
    ↪  ~/.ansible/default_vault_password ./homework.yaml

open:
    $(eval IP = $(shell python3 ./wordpress_address.py))
    xdg-open http://$(IP)

uninstall: homework.ini
    ansible-playbook -i ./homework.ini ./uninstall.yaml
```

## 3.3 Uninstall

Trong quá trình làm bài tập, em cần cài, rồi gỡ rất nhiều lần.

Do đó, em có viết thêm một *playbook* dành riêng cho việc hủy các container, image và gỡ cài đặt.

*Playbook* dành cho việc gỡ được đặt trong file `uninstall.yaml`

## 3.4 Ansible Vault

Với bài tập này, có những thông tin sau được truyền vào container:

- `mysql_root_password`
- `mysql_database`
- `mysql_user`
- `mysql_password`

Những thông tin này không nên được lưu dưới dạng plain text hay viết trực tiếp vào *playbook*.

Qua tìm hiểu trên trang tài liệu của **Ansible**, em sử dụng *vault*. Nhưng thay vì dùng *vault* để mã hóa toàn bộ *playbook*, em chỉ mã hóa từng chuỗi (4 chuỗi). Em làm như sau

Em dùng lệnh:

```
ansible-vault encrypt-string 'str to encrypt' --vault-id
↪  mysql@~/.ansible/default_vault_password --name variable
```

để mã hóa một chuỗi, em lấy output của lệnh này và đặt vào playbook, lưu thành một biến, cụ thể là:

```
  vars:
    secret_mysql_password: !vault |
        $ANSIBLE_VAULT;1.2;AES256;mysql
        ...
        ...
        ...
```

Khi chạy playbook, em truyền thêm option `--vault-id mysql@path/to/file`. Playbook sẽ dùng nội dung trong file được chỉ định để phá mã.

## Tài liệu


[1] Ansible Docs
https://docs.ansible.com/ansible/latest/user_guide/index.html

[2] Boto3 Docs
https://boto3.amazonaws.com/v1/documentation/api/latest/index.html