

1. Nội dung và hướng dẫn thực hiện bài thực hành

1.1. Mục đích

- Bài thực hành được xây dựng theo quy trình xây dựng và vận hành một hệ thống website, sử dụng những kiến thức cần thiết như: HAproxy, Ansible, scp.
- Vì vậy bài thực hành giúp sinh viên có những kỹ năng của một system admin.

1.2. Yêu cầu đối với sinh viên

- Người thực hiện cần có khả năng sử dụng Linux dưới mức độ quản trị.
- Có khả năng hiểu cơ chế hoạt động của HAproxy, Ansible ngoài ra còn có khả năng viết script ansible-playbook.

1.3. Nội dung thực hành

- Sinh viên download lab tại:
https://github.com/khainvvp/loadbalancer_nginx/releases/download/stepsof3/module.tar
- Khởi động bài lab
 - Vào terminal và gõ:

labtainer -r ptit-loadbalancer-nginx

<Người thực hiện nhập thông tin email của mình để chấm điểm>

- Trên container **haproxy** cập nhật domain của webserver và proxy bằng cách:

sudo -i

echo "<IP webserver01> web01.ptit.vn" >> /etc/hosts

echo "<IP webserver02> web02.ptit.vn" >> /etc/hosts

Sau khi khởi động xong, 3 terminal sẽ xuất hiện, một terminal đại diện cho máy loadbalancer: **haproxy (cũng được dùng làm ansible)**, một terminal đại diện cho máy webserver01: **webserver01**, một terminal đại diện cho máy webserver02: **webserver02**.

- Trên terminal **haproxy** sử dụng câu lệnh “*systemctl status haproxy*” để xác định tình trạng của haproxy service.

Sinh viên có thể thấy tình trạng của service haproxy là đang *failed*.
Hãy sửa lại file cấu hình của haproxy để sao cho cứ 8 request vào webserver01 thì sẽ có 1 request vào webserver02.

```
sudo vi /etc/haproxy/haproxy.cfg
```

```
sudo systemctl restart haproxy
```

- Sinh viên “cat” ra màn hình kiểm tra sau khi đã cấu hình .

```
cat /etc/haproxy/haproxy.cfg
```

- Để update nội dung các website mà không cần phải vào từng máy update, ta sử dụng Ansible, cụ thể ta sẽ dùng ansible-playbook.
 - Bài Lab này ta kết hợp **ansible** trên máy chủ **haproxy**
 - Đầu tiên cần phải trust-host từ **haproxy** tới **webserver01** và **webserver02** (nghĩa là **haproxy** login tới các **webserver** mà không cần nhập password hay key).
 - Đứng trên terminal “**haproxy**” với user là **ubuntu** gõ:

```
ssh-keygen
```

```
ssh-copy-id -i ~/.ssh/id_rsa.pub ubuntu@<IP  
webserver01>
```

```
ssh-copy-id -i ~/.ssh/id_rsa.pub ubuntu@<IP  
webserver02>
```

<với username/password 2 máy webserver là ubuntu/123 >

- Thêm các địa chỉ IP (**webserver01** và **webserver02**) vào Inventory và đặt tên là “**update_web**”.

```
sudo vi /etc/ansible/hosts
```

- Viết “ansible-playbook” triển khai 2 nhiệm vụ
 - Tạo một file trong /opt có tên insert.yaml
 - Tạo một file index.html với nội dung tùy ý (tại /home/ubuntu).
 - Trong file insert.yaml để xóa trang web cũ:

Để xóa trang web cũ, sử dụng module ***ansible.builtin.file*** với *state* là ***absent***

- Để update trang web mới có thể sử dụng 1 trong 2 cách:

✓ ***shell*** trong *ansible*

✓ Module ***copy*** trong *ansible*

<các trang web được lưu trên webserver tại /var/www/default>

- Cuối cùng, sinh viên chạy lệnh sau để update đồng loạt:

ansible-playbook /opt/insert.yaml -l update_web

- Trên máy “**webserver02**” sinh viên sử dụng câu lệnh để backup file *index.html* có trong */var/www/default* tới máy “**haproxy**” như sau:

scp -r /var/www/default ubuntu@<IP haproxy>:/home/ubuntu/

< với *username/password* máy **haproxy** là *ubuntu/123* >

- Kết thúc bài lab:

Trên terminal đầu tiên sử dụng câu lệnh sau để kết thúc bài lab:

stoplab ptit-loadbalancer-ansible

- Khi bài lab kết thúc, một tệp zip lưu kết quả được tạo và lưu vào một vị trí được hiển thị bên dưới *stoplab*.
- Khởi động lại bài lab:

labtainer -r ptit-loadbalancer-ansible

2. Phân tích, thiết kế bài thực hành

2.1. Phân tích yêu cầu bài thực hành

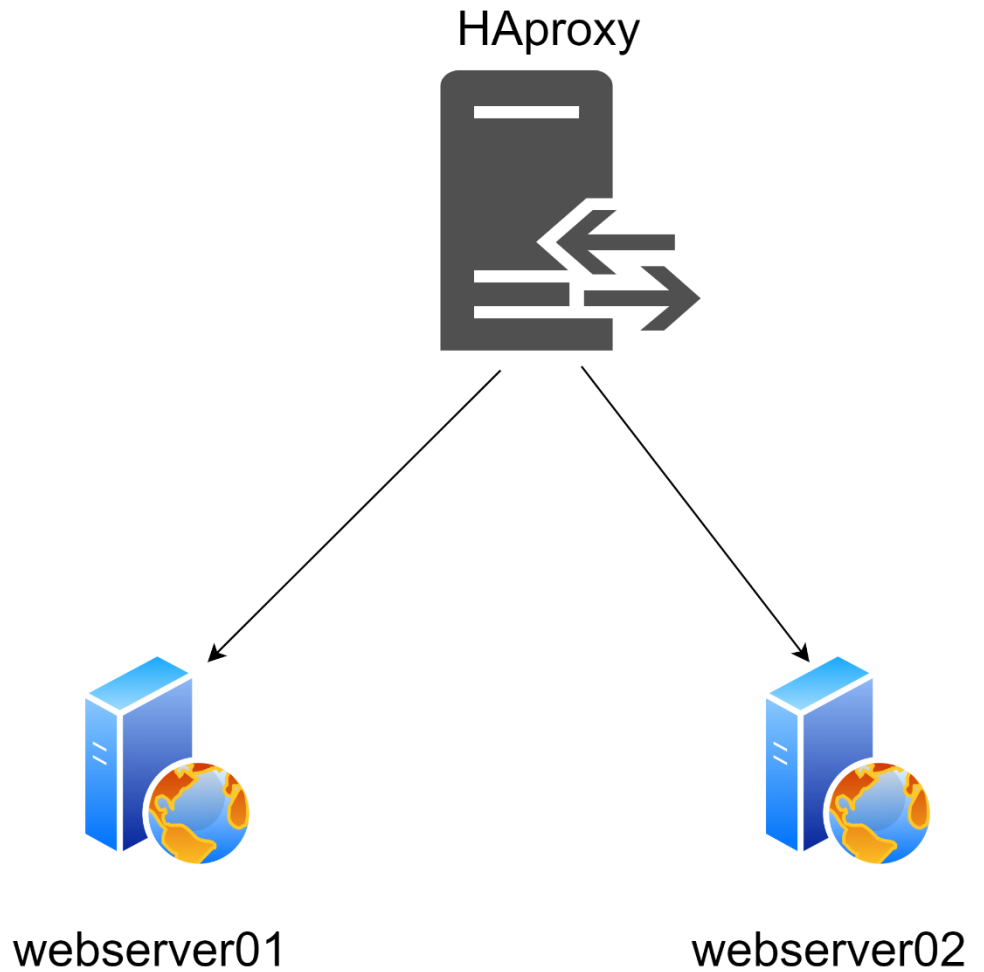
Bài thực hành cần có gồm có 3 máy nằm cùng trong mạng LAN. Trong đó một máy haproxy dùng để điều hướng request của client, hai máy webserver để trả về response của người dùng khi muốn truy cập. Trên máy haproxy, cài đặt một số công cụ hỗ trợ như: haproxy, ansible. Sinh viên cần sửa lại config của haproxy (vì mặc định đang bị lỗi) sao cho cứ 8 request vào webserver01 thì có 1 request vào webserver02 ngoài ra, nếu muốn triển khai update, xóa, tạo mới website thay vì truy cập vào từng webserver thì người ta chỉ cần sử dụng ansible-playbook để triển khai hàng loạt.

Để đáp ứng bài thực hành, cần cung cấp máy ảo chứa docker trong đó có 3 containers, mỗi container là một máy tính ảo chạy hệ điều hành Linux là máy haproxy, webserver01 và webserver02. Để có thể thực hiện được các tác vụ kể trên thì hệ thống cần cung cấp khả năng thiết lập một mạng LAN giữa các máy với nhau, đồng thời cung cấp các thư viện cần cho các máy để sử dụng được dịch vụ (ví dụ với webserver thì cần cung cấp các dịch vụ như Apache2, open-ssh, với haproxy cần cung cấp HAproxy, open-ssh).

Để bắt đầu bài thực hành, sinh viên cần phải sử dụng các câu lệnh khởi tạo (labtainer -r <tên bài lab>) và câu lệnh kết thúc (stoplab <tên bài lab>) để hệ thống chạy bài lab cũng như lưu lại kết quả.

2.2. Thiết kế bài thực hành

Trên môi trường máy ảo Ubuntu được cung cấp, sử dụng docker tạo ra 3 container: 1 container mang tên “haproxy” đóng vai trò là proxy và 2 container mang tên “webserver01”, “webserver02” đóng vai trò máy chủ webserver, cả 3 máy đều được mở các cổng cần thiết.



Hình 1: Sơ đồ thiết kế bài thực hành bài lab ptit-loadbalancer-ansible

- Tạo mạng LAN “NETWORKING” có cấu hình: 172.10.0.0/24 và gateway: 172.10.0.1
- Cấu hình docker gồm có:
 - haproxy: lưu cấu hình cho máy gồm có:
 - Tên máy: haproxy
 - Địa chỉ IP trong mạng: 172.10.0.10
 - Gateway: 172.10.0.1
 - Webserver01: lưu cấu hình máy gồm có:
 - Tên máy: webserver01
 - Địa chỉ IP trong mạng: 172.10.0.11

- Gateway: 172.10.0.1
 - Webserver02: lưu cấu hình máy gồm có:
 - Tên máy: webserver02
 - Địa chỉ IP trong mạng: 172.10.0.12
 - Gateway: 172.10.0.1
 - Dockerfile
 - HAproxy: Sử dụng các thư viện mặc định hệ thống cùng với cấu hình NETWORK ngoài ra cần sử dụng các package như haproxy, ansible nên sẽ được cài đặt vào Dockerfile.
 - Webserver01 và Webserver02: Sử dụng các thư viện mặc định hệ thống cùng với cấu hình NETWORK ngoài ra cần sử dụng các package như Apache2, open-ssh nên sẽ được cài đặt vào Dockerfile.
 - Docs: Lưu phần mô tả hướng dẫn bài thực hành cho sinh viên.
- Các nhiệm vụ cần được thực hiện thành công:
- ✓ Sử dụng lệnh “cat” hiển thị file haproxy.cfg sau khi cấu hình trên máy “haproxy” .
 - ✓ Vẫn trên máy “haproxy”, chạy câu lệnh “*ansible-playbook /opt/insert.yaml -l update_web*” để update lại trang web
 - ✓ Từ máy “webserver02”, backup file index.html trong (/var/www/default/) tới máy “haproxy”.
- Để đánh giá người thực hiện có hoàn thành được bài thực hành hay không, cần chia bài thực hành ra các nhiệm vụ nhỏ, mỗi nhiệm vụ cần phải chỉ rõ kết quả để có thể dựa vào đó đánh giá, chấm điểm. Do vậy, trong bài thực hành này hệ thống cần ghi nhận các thao tác, sự kiện được mô tả và cấu hình như bảng.