**Nội dung và hướng dẫn thực hiện bài thực hành**

**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 mội system admin.

**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.

**Nội dung thực hành**

* imodule *https://github.com/ParadoxParadise/labattt-2025/raw/refs/heads/main/ptit-loadbalancer-ansible/imodule.tar*
* Khởi động bài lab
  + Vào terminal và gõ:

*labtainer -r loadbalancer\_ansible*

<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 statte 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/khainv>*

* + *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/khainv*tới máy “haproxy” như sau:

*scp -r /var/www/khainv ubuntu@172.10.0.10:/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 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 loadbalancer\_ansible*