LAB WEEK 9

Triển khai hệ thống Load Balancing (HAProxy) cho webserver và database server

NGUYEN CONG LUC

nguyencongluc.82@gmail.com

Luc Nguyen | LinkedIn

0329206845

MỤC LỤC

MỤC TIÊU *3*

CHUẨN BỊ BAN ĐẦU *6*

Phần 1. Triển khai webserver trên WEB01 và WEB02 (Câu 1) *9*

1.1 Cài đặt Nginx **9**
1.2 Đồng bộ source code bằng lsyncd và kiểm tra **9**
1.3 Đồng bộ source code bằng lsyncd và kiểm tra **13**
Phần 2. Triển khai Load Balancing bằng HAProxy (Câu 2) *14*

2.1 Cài HAProxy **14**
2.2 Cấu hình HAProxy **14**
2.3 Kiểm tra **16**
Phần 3. Triển khai MySQL Galera Cluster (Câu 3) *17*

3.1 Cài đặt MariaDB + Galera **17**
3.2 Cài hình MariaDB Galera **17**
3.3 Kiểm tra cluster **19**
3.4 Cấu hình HAProxy cho database **20**
Phần 4. Tạo database và cài WordPress (Câu 4) *21*

4.1 Tạo database **21**
4.2 Cài đặt WordPress **23**

MỤC TIÊU

Lab Week 9: Triển khai hệ thống Load Balancing (HAProxy) cho webserver và database server

1. Yêu cầu môi trường

- **4 máy chủ: Mỗi máy có 2 core CPU, 2GB RAM, 30GB SSD.**
- **Hệ điều hành (OS): Ubuntu 22.04.**
- **IP: Mỗi máy thuộc dải VPS, có 1 IP Public riêng (công khai để truy cập từ internet).**

2. Thời gian

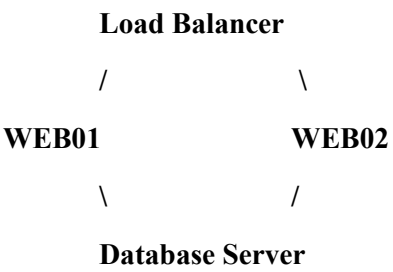
- **8h30 đến 16h00 để hoàn thành bài lab.**
- **Hình thức báo cáo: Báo cáo file kết quả sau khi làm bằng file PDF.**
- **Chương bị môi trường: Sử dụng VươngVD.**
- **Giao diện tài liệu: NghiTT.**
- **Khóa thực tập.**

3. Đề bài

Triển khai hệ thống Load Balancing (HAProxy) cho webserver. Đảm bảo khi có bất kỳ server nào bị lỗi, hệ thống vẫn hoạt động bình thường mà không gây gián đoạn dịch vụ.

4. Mô hình

- **Một Load Balancer (HAProxy) sẽ phân phối tải giữa 2 webserver (WEB01, WEB02) và 1 Database Server.**
- **Mô hình như sau:**



5. Các bước thực hiện (Câu 1-4)

Câu 1: Triển khai hệ thống webserver nginx trên 2 server WEB01 và WEB02. Đảm bảo source code khi upload lên WEB01 thì WEB02 sẽ được đồng bộ bằng `lsyncd`. (3 điểm)

Câu 2: Triển khai Load Balancing bằng HAProxy trên Load Balancer. Cấu hình roundrobin và healthcheck để đảm bảo khi có bất kỳ server nào gặp sự cố thì hệ thống vẫn phục hồi ngay và không gây gián đoạn dịch vụ. (2 điểm)

Câu 3: Triển khai cluster database MySQL Galera trên 3 server: WEB01, WEB02, và Database Server. Sau đó sử dụng Load Balancer để điều phối traffic cho bên ngoài kết nối vào database. (3 điểm)

Câu 4: Tạo database và cài đặt 1 website WordPress đơn giản với database vừa tạo. Phân quyền cho database chỉ cho phép 5 địa chỉ IP sau truy cập:

- localhost

- 125.212.250.9

- 113.161.61.219

- WEB01

- WEB02

(2 điểm)

Triển khai từng bước

Chuẩn bị trước khi bắt đầu

1. **Hiểu vai trò của từng máy**:

- **Load Balancer**: Cài HAProxy để phân phối tải.

- **WEB01, WEB02**: Cài webserver (Apache/Nginx) để chạy website.

- **Database Server**: Cài MySQL để lưu trữ dữ liệu.

2. **IP của các máy**: Bạn cần gán IP cụ thể cho từng máy. Vì đề bài không cung cấp IP thực tế, tôi sẽ giả định:

- Load Balancer: 192.168.1.10

- WEB01: 192.168.1.11

- WEB02: 192.168.1.12

- Database Server: 192.168.1.13

3. **Cài đặt cơ bản**:

- Cài Ubuntu 22.04 trên cả 4 máy.

- Cập nhật hệ thống:

  ```bash

  sudo apt update && sudo apt upgrade -y
  ```

CHUẨN BỊ BAN ĐẦU

IP Public hiện tại

- **Load Balancer (lb): 103.27.61.30/25**
- **WEB01 (web1): 103.27.61.10/25**
- **WEB02 (web2): 45.122.223.43/25**
- **Database Server (db): 45.122.223.47/25**

Yêu cầu Private IP

- **Dải IP: 192.168.x.1 - 192.168.x.4, với Subnet: /24.**
- **x là số cuối của IP Public của Load Balancer.**
- **IP Public của Load Balancer là 103.27.61.30, nên x = 30.**
- **Do đó, dải Private IP sẽ là: 192.168.30.1 - 192.168.30.4/24.**

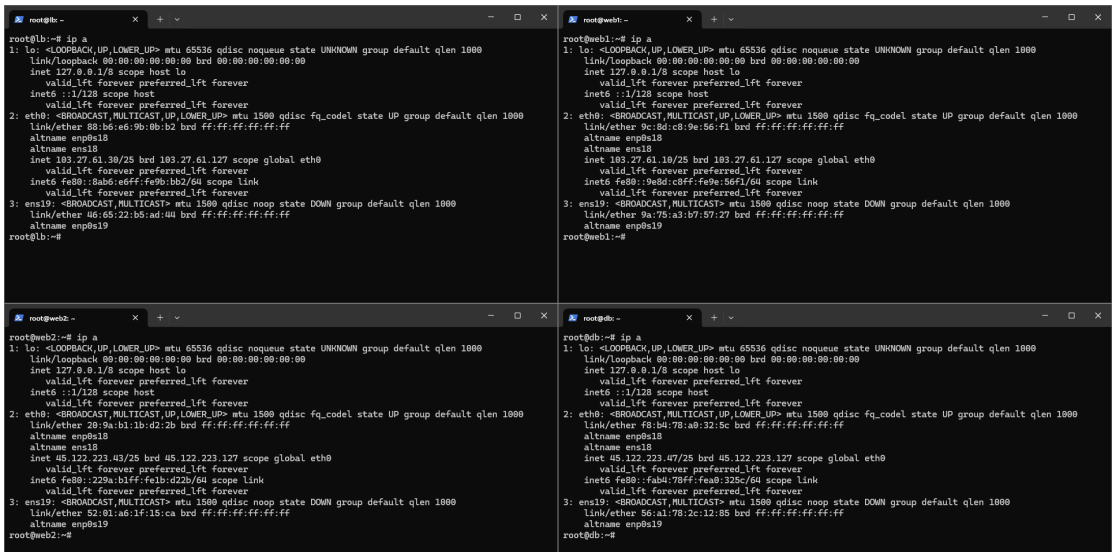
Phân bổ Private IP

- **Load Balancer: 192.168.30.1/24**
- **WEB01: 192.168.30.2/24**
- **WEB02: 192.168.30.3/24**
- **Database Server: 192.168.30.4/24**

Giao diện cần cấu hình

- **Tất cả các máy đều có giao diện ens19 (hiện đang ở trạng thái DOWN).**
- **Chúng ta sẽ kích hoạt ens19 và gán Private IP cho giao diện này.**

- Cấu hình IP hiện tại chưa có Privated IP:



* Thiết lập Privated IP:

- Trên Load Balancer (lb)

- Kết quả sau khi thêm IP và đã ping thông được với nhau:

- Chỉnh sửa lần lượt file cấu hình này trên 4 VM:

nano /etc/netplan/50-cloud-init.yaml

- Thêm cấu hình cho ens19 trong file netplan :

+ Trên Load Balancer (lb)

```
ens19:
  addresses:
  - 192.168.30.1/24
```

+ Trên WEB01 (web1)

```
ens19:
  addresses:
  - 192.168.30.2/24
```

+ Trên WEB02 (web2)

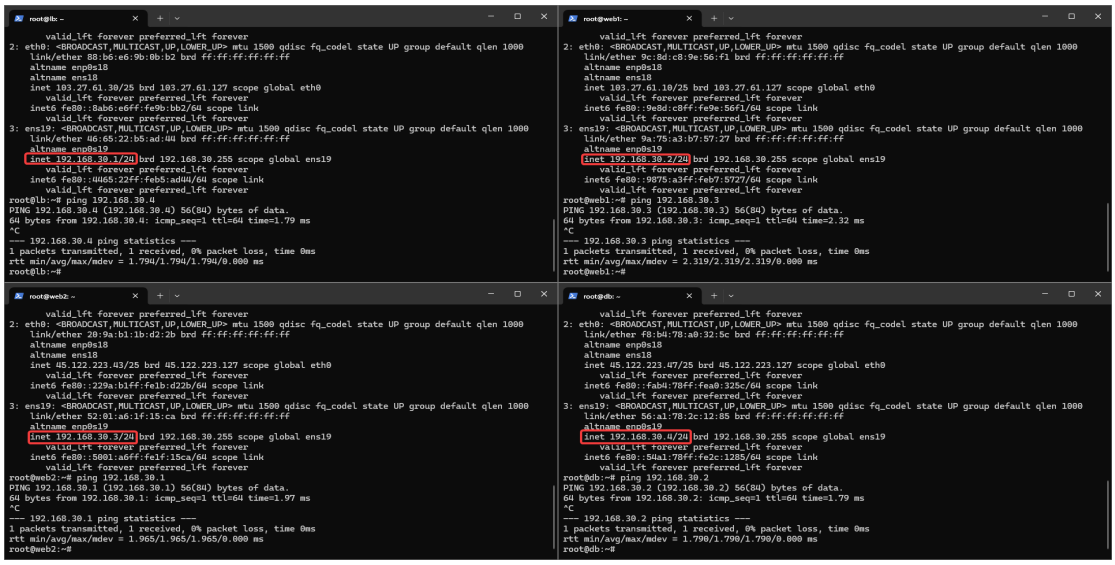
```
ens19:
  addresses:
  - 192.168.30.3/24
```

+ Trên Database Server (db)

```
ens19:
  addresses:
  - 192.168.30.3/24
```

- Chạy lệnh áp dụng cấu hình này trên 4 VM: netplan apply

- Kết quả sau khi set thành công, các IP privated đã ping được với nhau



Phần 1. Triển khai webserver trên WEB01 và WEB02 (Câu 1)

- Triển khai hệ thống webserver nginx trên 2 server WEB01 và WEB02. Đảm bảo source code khi upload lên WEB01 thì WEB02 sẽ được đồng bộ bằng `lsyncd`. (3 điểm)

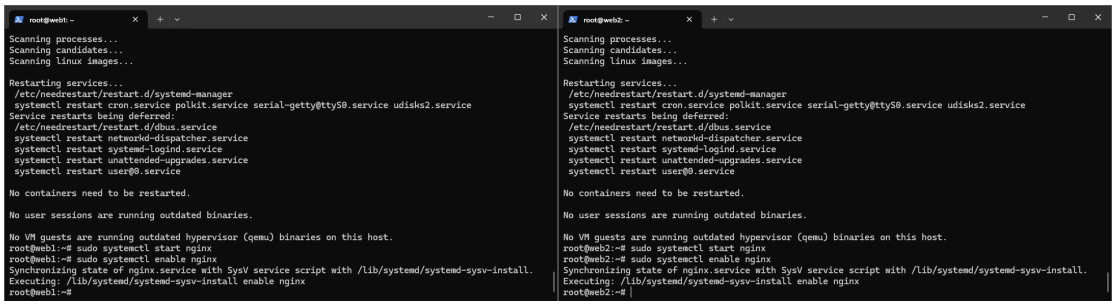
1.1 Cài đặt Nginx

- Cài Nginx trên cả WEB01 và WEB02:

sudo apt install nginx -y

sudo systemctl start nginx (Khởi động)

sudo systemctl enable nginx (Bật tự khởi động lại khi reboot)



1.2 Đồng bộ source code bằng lsyncd và kiểm tra

- Trên WEB01, cấu hình lsyncd để đồng bộ với WEB02 qua Private IP:

sudo apt install lsyncd -y

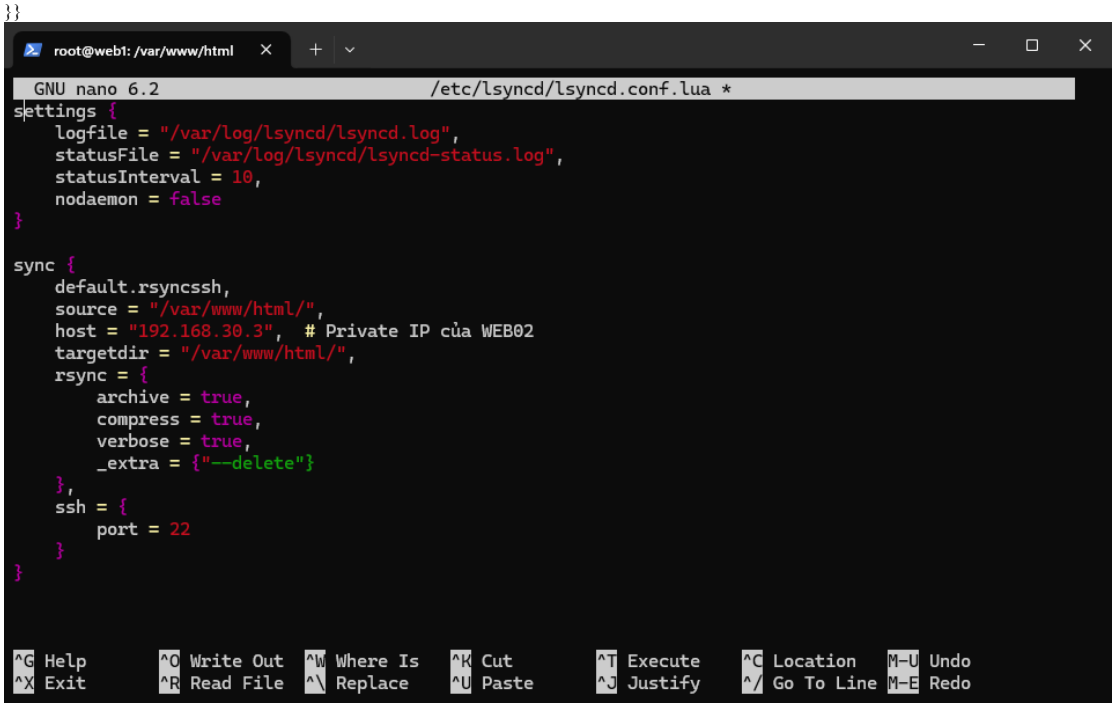
- Trên WEB01, tạo thư mục /etc/lsyncd:

sudo mkdir -p /etc/lsyncd

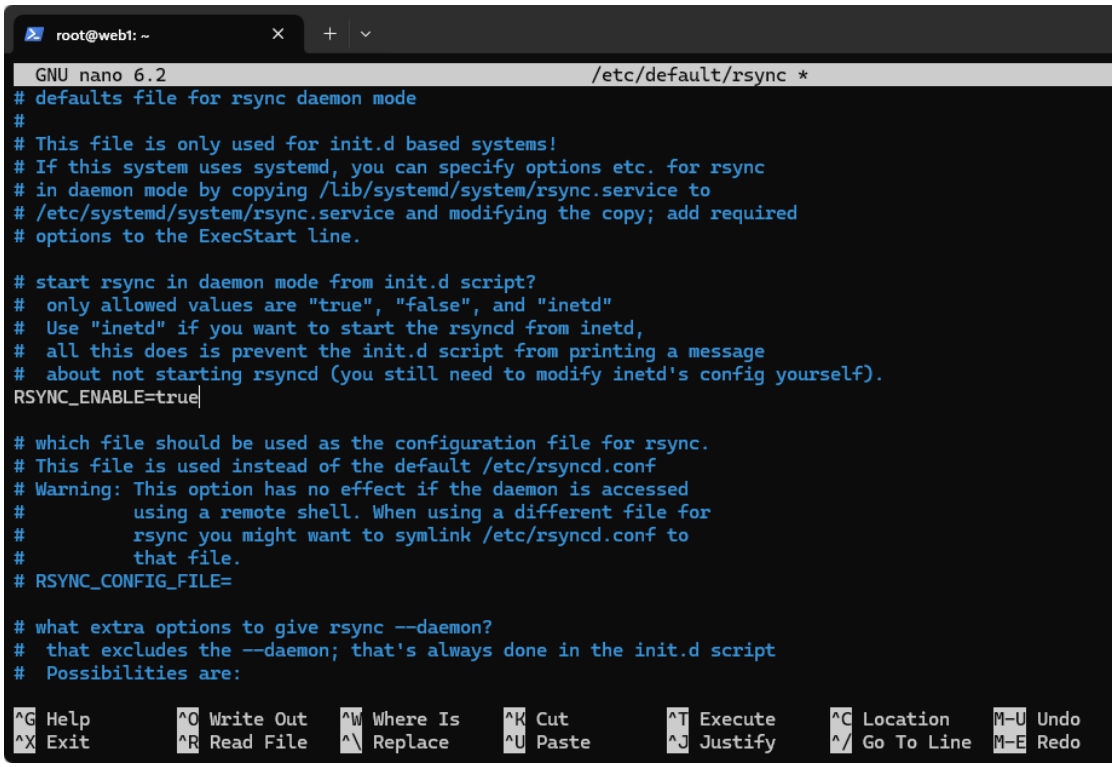
- Cấu hình: sudo nano /etc/lsyncd/lsyncd.conf.lua

```
settings {
    logfile = "/var/log/lsyncd/lsyncd.log",
    statusFile = "/var/log/lsyncd/lsyncd-status.log",
    statusInterval = 10,
    nodaemon = false
}

sync {
    default.rsyncssh,
    source = "/var/www/html/",
    host = "192.168.30.3",  # Private IP của WEB02
    targetdir = "/var/www/html/",
    rsync = {
        archive = true,
        compress = true,
        verbose = true,
        _extra = {"--delete"}
    },
    ssh = {
        port = 22
    }
}}
```



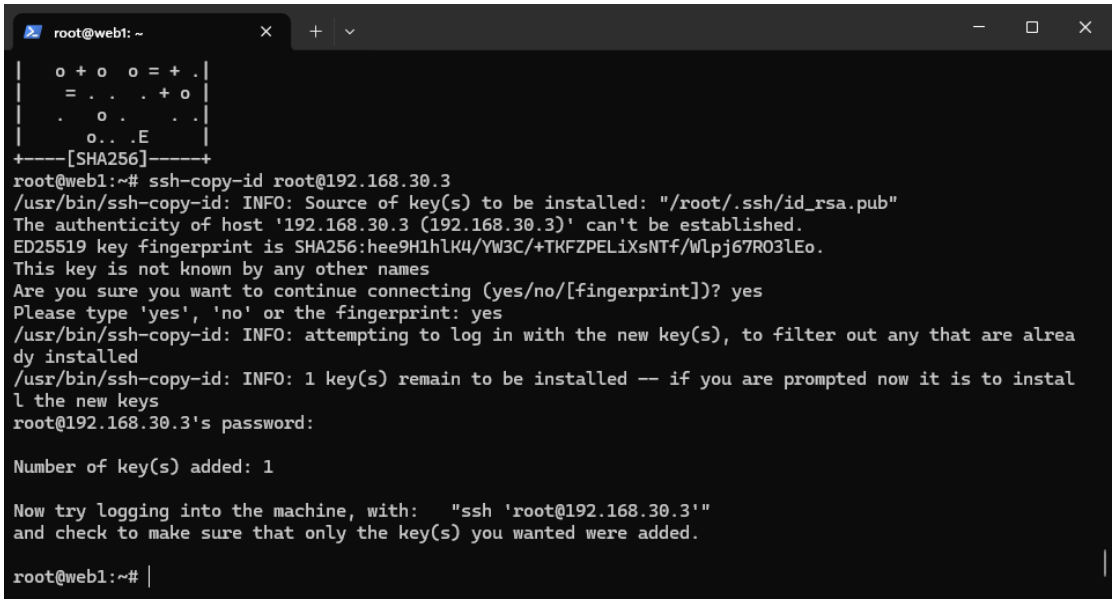
- Bật rsync:


root@web1: ~
GNU nano 6.2 /etc/default/rsync *
defaults file for rsync daemon mode
#
# This file is only used for init.d based systems!
# If this system uses systemd, you can specify options etc. for rsync
# in daemon mode by copying /lib/systemd/system/rsync.service to
# /etc/systemd/system/rsync.service and modifying the copy; add required
# options to the ExecStart line.
# start rsync in daemon mode from init.d script?
# only allowed values are "true", "false", and "inetd"
# Use "inetd" if you want to start the rsyncd from inetd,
# all this does is prevent the init.d script from printing a message
# about not starting rsyncd (you still need to modify inetd's config yourself).
RSYNC_ENABLE=true
# which file should be used as the configuration file for rsync.
# This file is used instead of the default /etc/rsyncd.conf
# Warning: This option has no effect if the daemon is accessed
# using a remote shell. When using a different file for
# rsync you might want to symlink /etc/rsyncd.conf to
# that file.
# RSYNC_CONFIG_FILE=
# what extra options to give rsync --daemon?
# that excludes the --daemon; that's always done in the init.d script
# Possibilities are:
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo

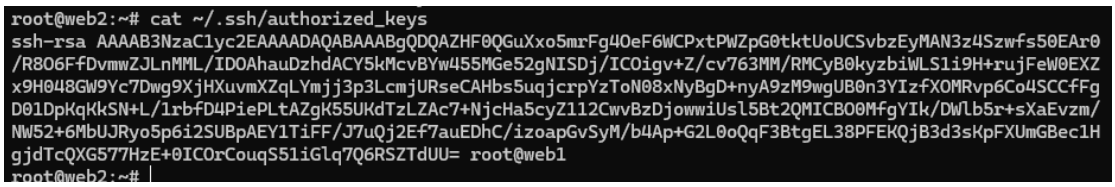
- Cấu hình SSH(vì lsyncd dùng giao thức này để đồng bộ file):

ssh-keygen -t rsa

ssh-copy-id root@192.168.30.3



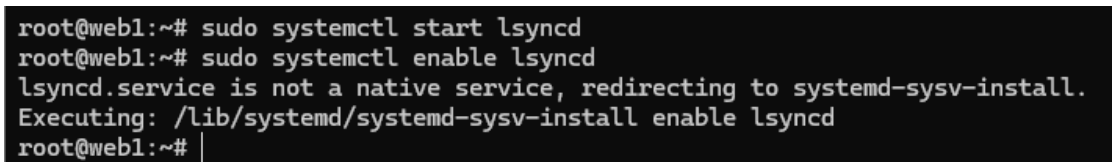
- Kiểm tra đã có tại web2:



- Khởi động:

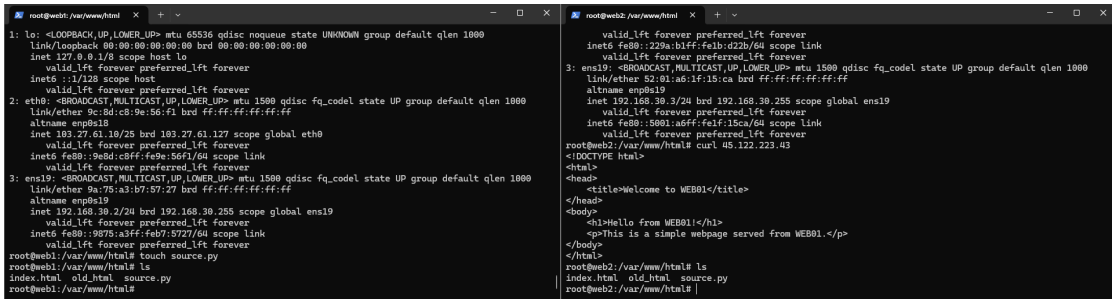
sudo systemctl start lsyncd

sudo systemctl enable lsyncd

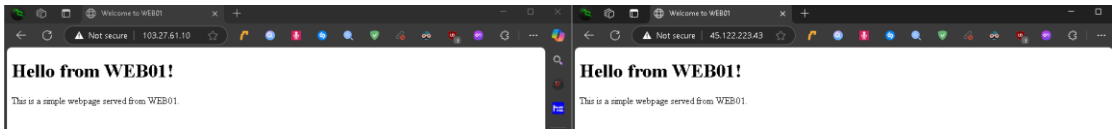


1.3 Đồng bộ source code bằng lsyncd và kiểm tra

- Tạo file source.py ở web 1, và web 2 sẽ được đồng bộ theo



- Cả 2 site đều cập nhật lại web khi có thay đổi:



Phần 2. Triển khai Load Balancing bằng HAProxy (Câu 2)

- Triển khai Load Balancing bằng HAProxy trên Load Balancer. Cấu hình roundrobin và healthcheck để đảm bảo khi có bất kỳ server nào gặp sự cố thì hệ thống vẫn phục hồi ngay và không gây gián đoạn dịch vụ. (2 điểm)

2.1 Cài HAProxy

- Cài đặt trên Load Balancer:

sudo apt install haproxy -y

sudo systemctl start haproxy

sudo systemctl enable haproxy

```
root@lb:~# sudo systemctl start haproxy
root@lb:~# sudo systemctl enable haproxy
Synchronizing state of haproxy.service with SysV service script with /lib/systemd/systemd-sysv-install
.
Executing: /lib/systemd/systemd-sysv-install enable haproxy
```

2.2 Cấu hình HAProxy

- Chỉnh sửa file cấu hình:

sudo nano /etc/haproxy/haproxy.cfg

theo cấu hình sau:

frontend http_front

    bind 0.0.0.0:80

    mode http

    option http-server-close

    option forwardfor

    default_backend http_back

backend http_back

    mode http

    balance roundrobin

    option httpchk GET / HTTP/1.1\r\nHost:\ localhost

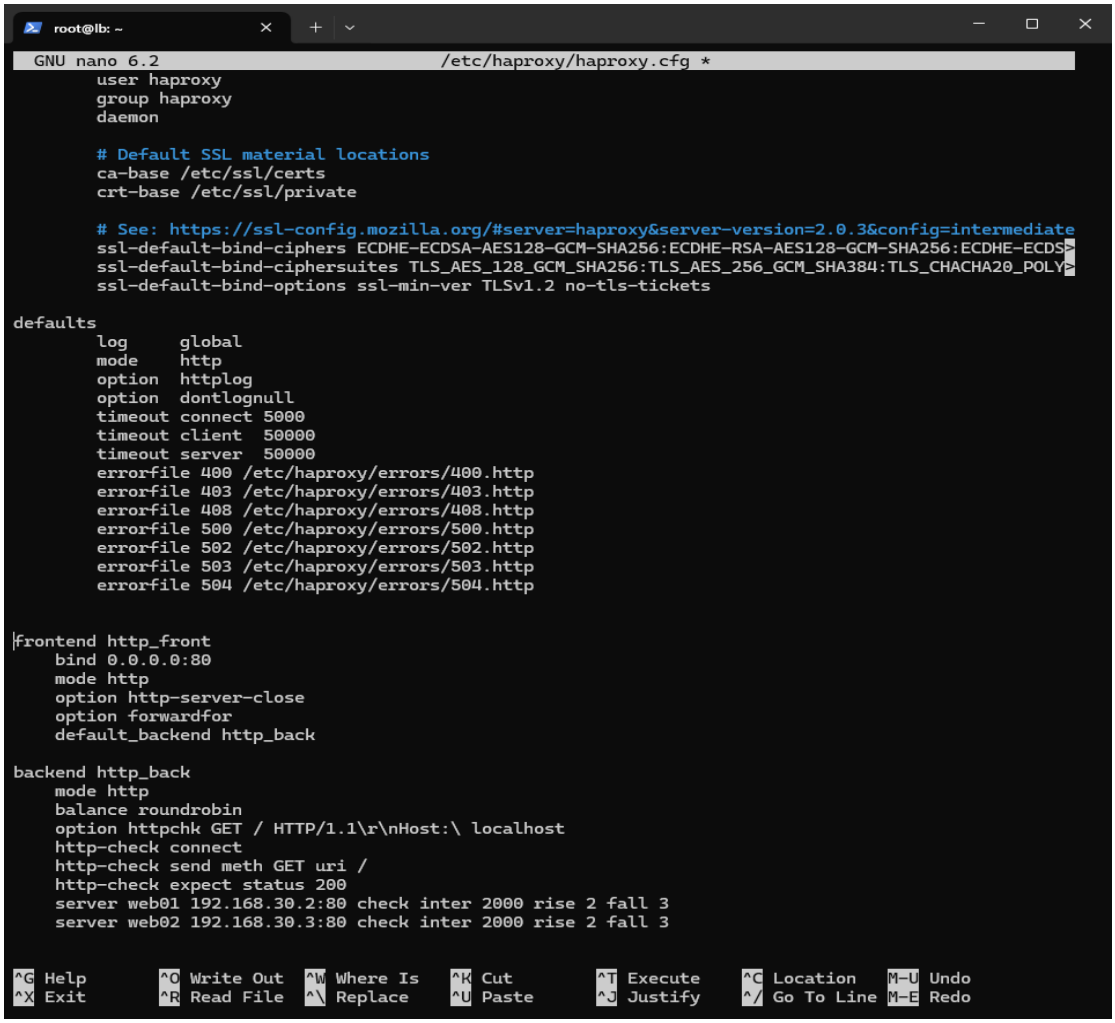
    http-check connect

    http-check send meth GET uri /

    http-check expect status 200

    server web01 192.168.30.2:80 check inter 2000 rise 2 fall 3

    server web02 192.168.30.3:80 check inter 2000 rise 2 fall 3



- Kiểm tra cú pháp:

sudo haproxy -c -f /etc/haproxy/haproxy.cfg

- Khởi động:

root@lb:~# sudo systemctl restart haproxy

root@lb:~# sudo systemctl enable haproxy

```
root@lb:~# sudo nano /etc/haproxy/haproxy.cfg
root@lb:~#
root@lb:~# sudo haproxy -c -f /etc/haproxy/haproxy.cfg
[WARNING]  (29435) : parsing [/etc/haproxy/haproxy.cfg:47]: 'option httpchk' : hiding headers or body
at the end of the version string is deprecated. Please, consider to use 'http-check send' directive in
stead.
Warnings were found.
Configuration file is valid
root@lb:~# sudo systemctl restart haproxy
root@lb:~# sudo systemctl enable haproxy
Synchronizing state of haproxy.service with SysV service script with /lib/systemd/systemd-sysv-install
.
Executing: /lib/systemd/systemd-sysv-install enable haproxy
```

2.3 Kiểm tra

- Truy cập IP Public của Load Balancer (103.27.61.30) trên trình duyệt và liên tục reload. Sẽ thấy nội dung từ WEB01 hoặc WEB02 luân phiên.

Phần 3. Triển khai MySQL Galera Cluster (Câu 3)

- Triển khai cluster database MySQL Galera trên 3 server: WEB01, WEB02, và Database Server. Sau đó sử dụng Load Balancer để điều phối traffic cho bên ngoài kết nối vào database. (3 điểm)

3.1 Tải về MariaDB + Galera

- MySQL Galera trên Ubuntu 24.04 yêu cầu thêm repository chính thức. Thực hiện trên WEB01, WEB02, và Database Server:

sudo apt install mariadb-server galera-4 -y

3.2 Cấu hình MariaDB Galera

- Trên **WEB01**:

sudo nano /etc/mysql/mariadb.conf.d/60-galera.cnf

- Nội dung:

[galera]

Mandatory settings

wsrep_on=ON

wsrep_provider=/usr/lib/galera/libgalera_smm.so

wsrep_cluster_name="MariaDB Galera Cluster"

wsrep_cluster_address="gcomm://192.168.30.2,192.168.30.3,192.168.30.4"

wsrep_node_name="web01"

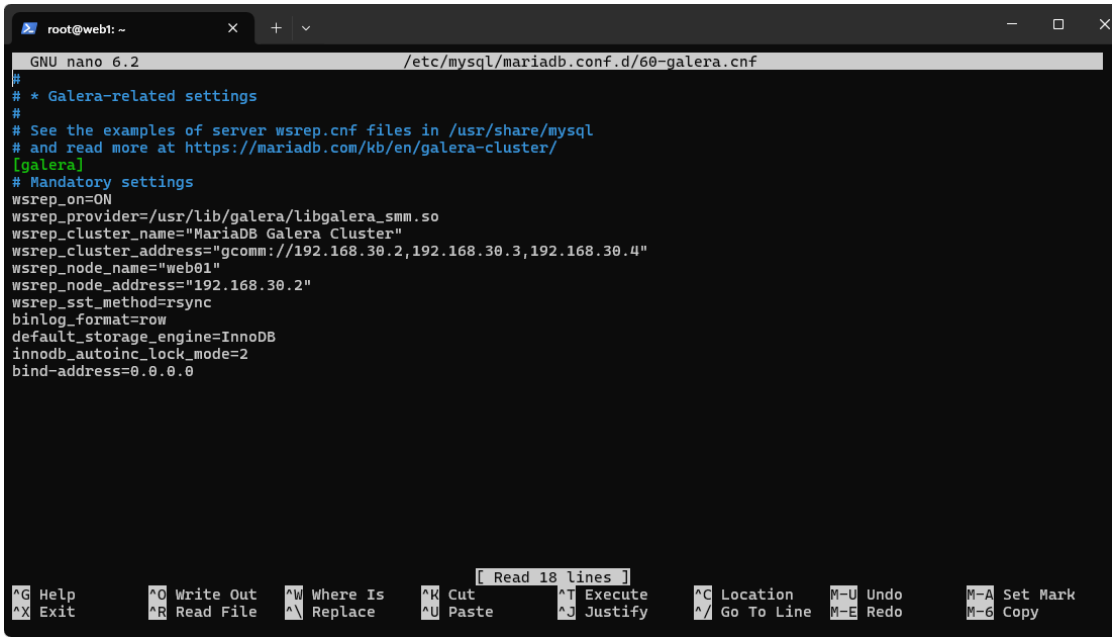
wsrep_node_address="192.168.30.2"

wsrep_sst_method=rsync

binlog_format=row

default_storage_engine=InnoDB

innodb_autoinc_lock_mode=2

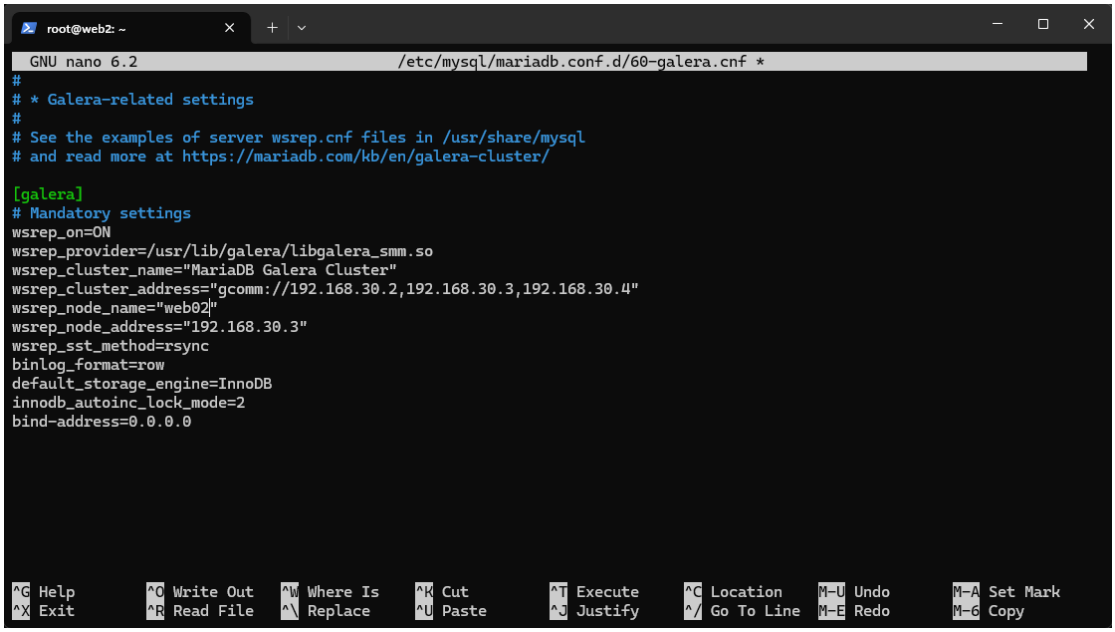


sudo systemctl stop mariadb

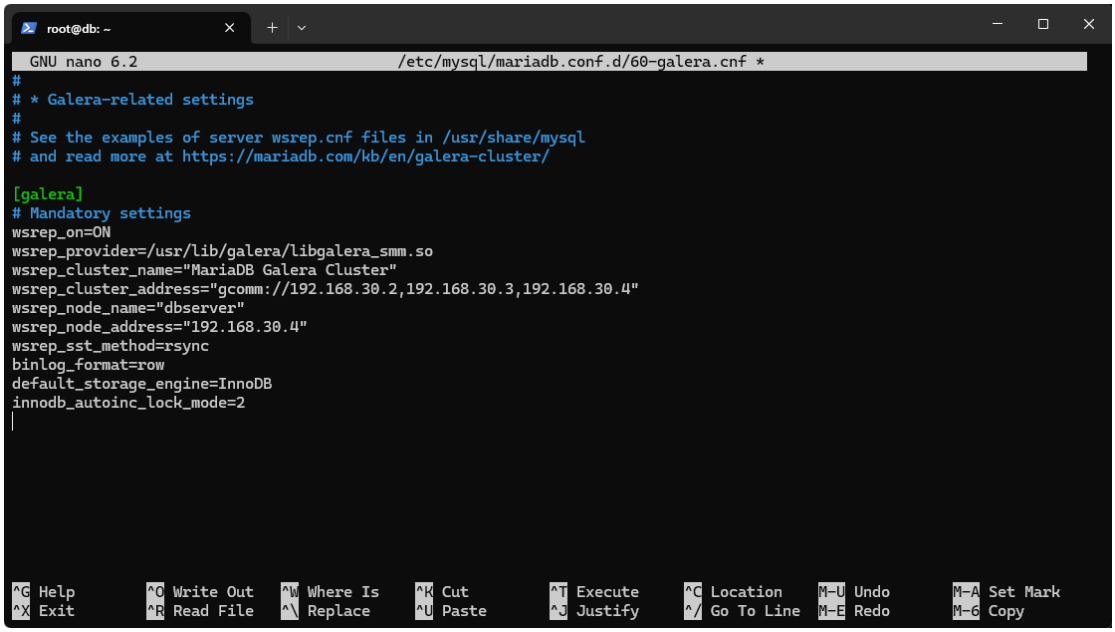
galera_new_cluster (Chỉ chạy ở web1)

systemctl start mariadb

-Trên WEB02 và Database Server: Tương tự, chỉ thay đổi wsrep_node_address và wsrep_node_name.



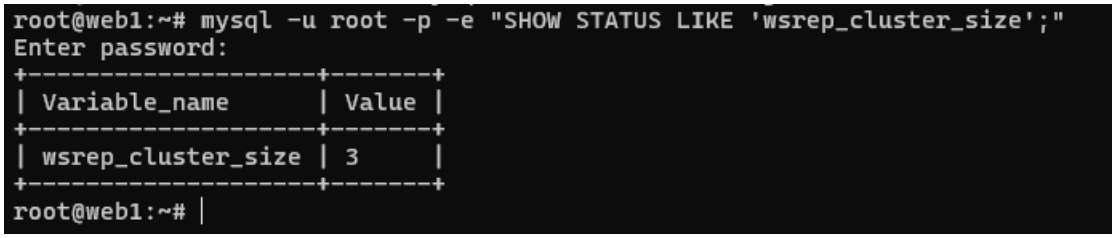
systemctl start mariadb



systemctl start mariadb

3.3 Kiểm tra cluster

Ở web 1: chạy mysql -u root -p -e "SHOW STATUS LIKE 'wsrep_cluster_size';"



3.4 Cấu hình HAProxy cho database

- Trên Load Balancer, chỉnh sửa haproxy.cfg:

sudo nano /etc/haproxy/haproxy.cfg

frontend mysql_front

bind *:3306

mode tcp

default_backend mysql_back

backend mysql_back

mode tcp

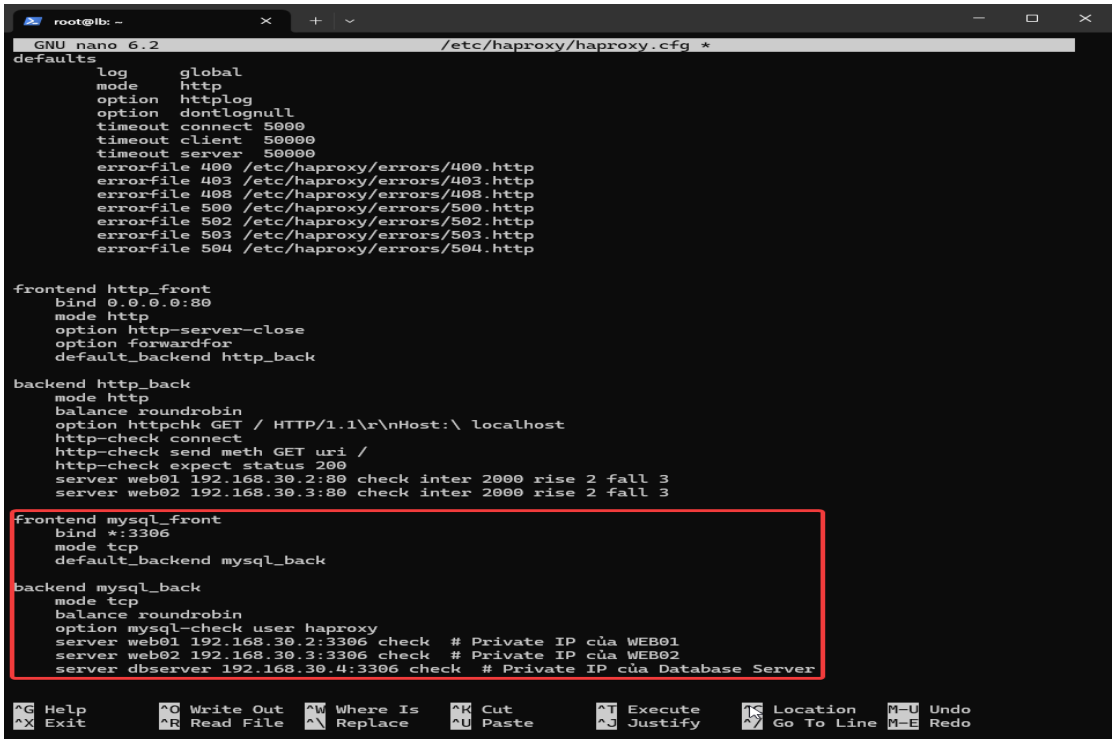
balance roundrobin

option mysql-check user haproxy

server web01 192.168.30.2:3306 check # Private IP của WEB01

server web02 192.168.30.3:3306 check # Private IP của WEB02

server dbserver 192.168.30.4:3306 check # Private IP của Database Server



Khởi động lại:

sudo systemctl restart haproxy

Phần 4. Tạo database và cài WordPress (Câu 4)

Tạo database và cài đặt 1 website WordPress đơn giản với database vừa tạo. Phân quyền cho database chỉ cho phép 5 địa chỉ IP sau truy cập:

- localhost

- 125.212.250.9

- 113.161.61.219

- WEB01

- WEB02

4.1 Tạo database

- Truy cập MySQL trên WEB01:

```
mysql -u root -p
```

- Tạo database và user:

```
CREATE DATABASE wordpress;

CREATE USER 'wp_user'@'localhost' IDENTIFIED BY 'password';

CREATE USER 'wp_user'@'125.212.250.9' IDENTIFIED BY 'password';

CREATE USER 'wp_user'@'113.161.61.219' IDENTIFIED BY 'password';

CREATE USER 'wp_user'@'192.168.30.2' IDENTIFIED BY 'password';  # Private IP của WEB01

CREATE USER 'wp_user'@'192.168.30.3' IDENTIFIED BY 'password';  # Private IP của WEB02

GRANT ALL PRIVILEGES ON wordpress.* TO 'wp_user'@'localhost';

GRANT ALL PRIVILEGES ON wordpress.* TO 'wp_user'@'125.212.250.9';

GRANT ALL PRIVILEGES ON wordpress.* TO 'wp_user'@'113.161.61.219';

GRANT ALL PRIVILEGES ON wordpress.* TO 'wp_user'@'192.168.30.2';

GRANT ALL PRIVILEGES ON wordpress.* TO 'wp_user'@'192.168.30.3';

FLUSH PRIVILEGES;

EXIT;
```

```
MariaDB [(none)]> CREATE DATABASE wordpress;
ER 'wp_user'@'125.212.250.9' IDENTIFIED BY 'password';
CREATE USER 'wp_user'@'113.161.61.219' IDENTIFIED BY 'password';
CREATE USER 'wp_user'@'192.168.30.2' IDENTIFIED BY 'password';  # Private IP của WEB01
CREATE USER 'wp_user'@'192.168.30.3' IDENTIFIED BY 'password';  # Private IP của WEB02
GRANT ALL PRIVILEGES ON wordpress.* TO 'wp_user'@'localhost';
GRANT ALL PRIVILEGES ON wordpress.* TO 'wp_user'@'125.212.250.9';
GRANT ALL PRIVILEGES ON wordpress.* TO 'wp_user'@'113.161.61.219';
GRANT ALL PRIVILEGES ON wordpress.* TO 'wp_user'@'192.168.30.2';
GRANT ALL PRIVILEGES ON wordpress.* TO 'wp_user'@'192.168.30.3';
FLUSH PRIVILEGES;
EXIT;Query OK, 1 row affected (0.075 sec)

MariaDB [(none)]> CREATE USER 'wp_user'@'localhost' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.096 sec)

MariaDB [(none)]> CREATE USER 'wp_user'@'125.212.250.9' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.103 sec)

MariaDB [(none)]> CREATE USER 'wp_user'@'113.161.61.219' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.083 sec)

MariaDB [(none)]> CREATE USER 'wp_user'@'192.168.30.2' IDENTIFIED BY 'password';  # Private IP của WEB01
Query OK, 0 rows affected (0.042 sec)

MariaDB [(none)]> CREATE USER 'wp_user'@'192.168.30.3' IDENTIFIED BY 'password';  # Private IP của WEB02
Query OK, 0 rows affected (0.062 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON wordpress.* TO 'wp_user'@'localhost';
Query OK, 0 rows affected (0.055 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON wordpress.* TO 'wp_user'@'125.212.250.9';
Query OK, 0 rows affected (0.084 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON wordpress.* TO 'wp_user'@'113.161.61.219';
Query OK, 0 rows affected (0.091 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON wordpress.* TO 'wp_user'@'192.168.30.2';
Query OK, 0 rows affected (0.061 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON wordpress.* TO 'wp_user'@'192.168.30.3';
Query OK, 0 rows affected (0.071 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.047 sec)

MariaDB [(none)]> EXIT;
```

4.2 Cài đặt WordPress

- Tải và cài đặt trên WEB01:

cd /var/www/html

sudo wget https://wordpress.org/latest.tar.gz

sudo tar -xvzf latest.tar.gz

sudo mv wordpress/* .

sudo rm -rf wordpress latest.tar.gz

sudo chown -R www-data:www-data /var/www/html

sudo chmod -R 755 /var/www/html

- Cấu hình wp-config.php:

sudo cp wp-config-sample.php wp-config.php

sudo nano wp-config.php

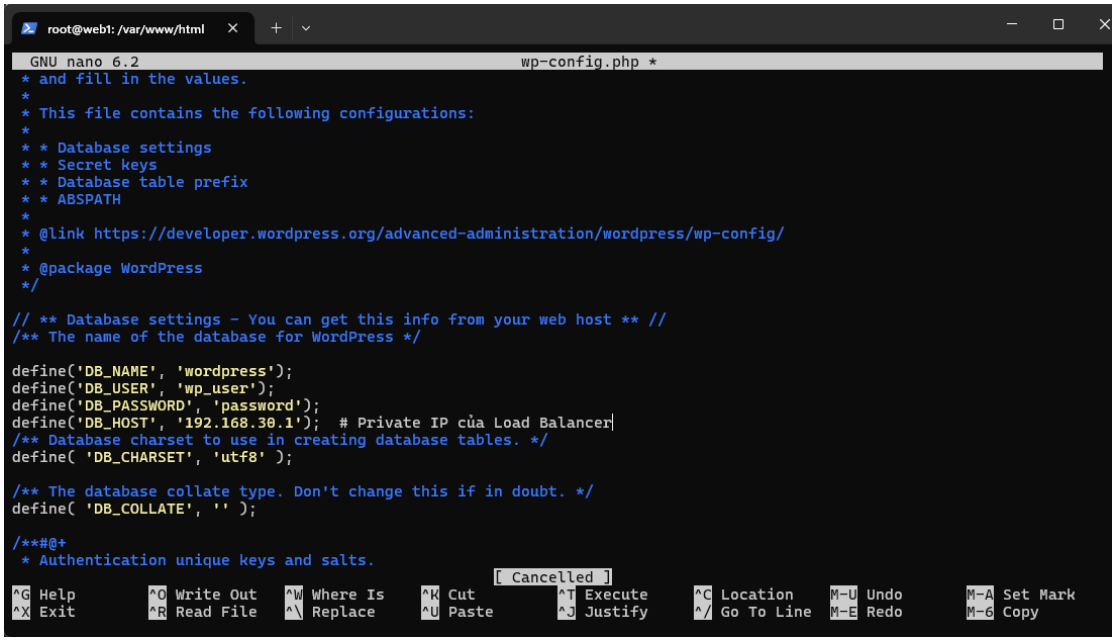
Sửa =>

define('DB_NAME', 'wordpress');

define('DB_USER', 'wp_user');

define('DB_PASSWORD', 'password');

define('DB_HOST', '192.168.30.1'); # Private IP của Load Balancer



- Kiểm tra truy cập:

```
root@web1:/var/www/html# curl http://192.168.30.2/wp-admin/admin.php
<?php
/**
 * WordPress Administration Bootstrap
 *
 * @package WordPress
 * @subpackage Administration
 */

/**
 * In WordPress Administration Screens
 *
 * @since 2.3.2
 */
if ( ! defined( 'WP_ADMIN' ) ) {
        define( 'WP_ADMIN', true );
}

if ( ! defined( 'WP_NETWORK_ADMIN' ) ) {
        define( 'WP_NETWORK_ADMIN', false );
}

if ( ! defined( 'WP_USER_ADMIN' ) ) {
        define( 'WP_USER_ADMIN', false );
}

if ( ! WP_NETWORK_ADMIN && ! WP_USER_ADMIN ) {
        define( 'WP_BLOG_ADMIN', true );
}

if ( isset( $_GET['import'] ) && ! defined( 'WP_LOAD_IMPORTERS' ) ) {
        define( 'WP_LOAD_IMPORTERS', true );
}
```