

## **Đề bài - Kiểm tra triển khai hạ tầng Web & Database]**

**Câu 1 (3 điểm):** Triển khai hệ thống webserver nginx trên 2 server Web01 và Web02.

Yêu cầu: Đảm bảo source code khi upload lên Web01 thì Web02 sẽ được đồng bộ theo bằng lsyncd.

**Câu 2 (2 điểm):** Triển khai hệ thống load balancing bằng Haproxy (Load Balancer) trên server Load Balancer.

Yêu cầu:

- Cấu hình cơ chế round robin
- Thiết lập healthcheck
- Đảm bảo khi có bất kỳ server nào gặp sự cố thì phát hiện ngay và ngừng chuyển request về webserver đó.

**Câu 3 (3 điểm):** Triển khai cluster database MySQL Galera trên 3 server: Web01, Web02 và Database Server.

Sau đó sử dụng load balancer ở Câu 2 để cấu hình và điều phối traffic cho bên ngoài kết nối vào database.

**Câu 4 (2 điểm):** 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 được truy cập:

1. `localhost`
2. `125.212.250.9`
3. `113.161.61.219`
4. `Web01`
5. `Web02`

## MỤC LỤC

<b>Câu 1.</b>	4
<b>1.1 Cấu hình dịch vụ NGINX trên Web01 và Web02</b>	4
<b>1.2 Cấu hình nginx để phục vụ website Web01 và Web02</b>	4
<b>1.3 Cài đặt Isyncd trên Web01</b>	5
<b>1.4 Cấu hình Isyncd để đồng bộ từ Web01 sang Web02</b>	5
<b>1.5 Kiểm tra đồng bộ Web01 đến Web02</b>	6
<b>Câu 2.</b>	8
<b>2.1 Cài đặt Haproxy trên Load Balancer</b>	8
<b>2.2 Cấu hình Haproxy</b>	8
<b>2.3 Kiểm tra dịch vụ</b>	9
<b>2.3.1 kiểm tra nhật kí</b>	9
<b>2.3.2 kiểm tra stop dịch vụ Web01</b>	10
<b>Câu 3.</b>	12
<b>3.1 Triển khai MySQL Galera Cluster trên Web01, Web02 và Database Server</b>	12
<b>3.2 Cấu hình MariaDB Galera Cluster</b>	12
<b>3.2.1 chỉnh sửa Web01 tại tệp /etc/mysql/mariadb.conf.d/60-galera.cnf:</b>	12
<b>3.2.2 chỉnh sửa Web02 tại tệp /etc/mysql/mariadb.conf.d/60-galera.cnf:</b>	13
<b>3.2.3 chỉnh sửa Database Server tại tệp /etc/mysql/mariadb.conf.d/60-galera.cnf:</b>	13
<b>3.3 Khởi động MariaDB Galera Cluster</b>	14
<b>3.3.1 Trên Web01:</b>	14
<b>3.3.2 Trên Web02 và Database Server</b>	15
<b>3.3.3 Kiểm tra đồng bộ dữ liệu</b>	15
<b>3.4 Cấu hình Haproxy để điều phối traffic đến MariaDB Galera Cluster</b>	16
<b>3.4.1 Cập nhật cấu hình Haproxy trên Load Balancer</b>	16
<b>3.4.2 Tạo user haproxy_check trên MariaDB Cluster</b>	17
<b>3.4.3 Kiểm tra và khởi động Haproxy trên Load Balancer</b>	17
<b>3.4.4 Truy cập database qua haproxy</b>	17
<b>Câu 4.</b>	18
<b>4.1 Tạo database</b>	18

<b>4.2</b>	<b>Tạo và phân quyền user cho database .....</b>	<b>18</b>
<b>4.3</b>	<b>Cài đặt WordPress đơn giản trên Web01 .....</b>	<b>18</b>

## Thực hành

VM	IP
Load Balancer	192.168.22.1
Web01	192.168.22.2
Web02	192.168.22.3
Database Server	192.168.22.4

### Câu 1.

#### 1.1 Cấu hình dịch vụ NGINX trên Web01 và Web02

- Dùng lệnh:

```
# sudo apt update
```

```
# sudo apt install nginx -y
```

- Khởi động và kích hoạt nginx:

```
# sudo systemctl start nginx
```

```
# sudo systemctl enable nginx
```

```
root@lb:~# curl -I 192.168.22.2
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Sat, 12 Apr 2025 02:24:36 GMT
Content-Type: text/html
Content-Length: 612
Last-Modified: Sat, 12 Apr 2025 02:23:28 GMT
Connection: keep-alive
ETag: "67f9cea0-264"
Accept-Ranges: bytes
```

```
root@lb:~# curl -I 192.168.22.3
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Sat, 12 Apr 2025 02:24:46 GMT
Content-Type: text/html
Content-Length: 612
Last-Modified: Sat, 12 Apr 2025 02:23:41 GMT
Connection: keep-alive
ETag: "67f9cead-264"
Accept-Ranges: bytes
```

#### 1.2 Cấu hình nginx để phục vụ website Web01 và Web02

- Cấu hình file nginx mặc định (ví dụ: /etc/nginx/sites-available/default) trên cả hai server:

```
server {  
    listen 80;  
    server_name _;  
    root /var/www/html;  
    index index.html index.htm;  
    location / {  
        try_files $uri $uri/ =404;  
    }  
}
```

- Kiểm tra cấu hình và khởi động lại nginx:

```
# sudo nginx -t
```

```
# sudo systemctl restart nginx
```

### 1.3 Cài đặt lsyncd trên Web01

- Cài đặt dịch vụ

```
# sudo apt install lsyncd -y
```

### 1.4 Cấu hình lsyncd để đồng bộ từ Web01 sang Web02

- Tạo file cấu hình lsyncd (ví dụ: /etc/lsyncd/lsyncd.conf.lua):

```
settings {  
    logfile = "/var/log/lsyncd.log",  
    statusFile = "/var/log/lsyncd.status",  
}  
sync {  
    default.rsync,  
    source = "/var/www/html",  
    target = "root@192.168.22.3:/var/www/html",  
    rsync = {  
        archive = true,  
        compress = true,  
        _extra = { "--rsh=sshpass -p 'YOUR_PASSWORD' ssh -o StrictHostKeyChecking=no" }  
    }  
}
```

- Khởi động và enable lsyncd

```
# sudo systemctl enable lsyncd
```

```
# sudo systemctl start lsyncd
```

```
root@web1: /etc
root@web1:/etc# sudo service lsyncd enable
sudo service lsyncd start
Usage: /etc/init.d/lsyncd {start|stop|restart|force-reload}
root@web1:/etc# sudo service lsyncd status
● lsyncd.service - LSB: lsyncd daemon init script
   Loaded: loaded (/etc/init.d/lsyncd; generated)
   Active: active (exited) since Sat 2025-04-12 02:34:07 UTC; 9min ago
     Docs: man:systemd-sysv-generator(8)
    CPU: 3ms

Apr 12 02:34:07 web1 systemd[1]: Starting LSB: lsyncd daemon init script...
Apr 12 02:34:07 web1 systemd[1]: Started LSB: lsyncd daemon init script.
root@web1:/etc#
```

## 1.5 Kiểm tra đồng bộ Web01 đến Web02

- Trên Web01, tạo một file thử nghiệm trong /var/www/html/test.html:

```
# echo "<h1>Test sync from Web01</h1>" | sudo tee /var/www/html/test.html
```

- Kiểm tra cú pháp file cấu hình.

```
# sudo lsyncd -nodaemon /etc/lsyncd/lsyncd.conf.lua
```

```
root@web1: /var/www/html
root@web1:/var/www/html# echo "<h1>Test sync from Web01</h1>" | sudo tee /va
r/www/html/test.html
<h1>Test sync from Web01</h1>
root@web1:/var/www/html# sudo lsyncd -nodaemon /etc/lsyncd/lsyncd.conf.lua
02:57:55 Normal: --- Startup ---
02:57:55 Normal: recursive startup rsync: /var/www/html/ -> root@192.168.22.
3: /var/www/html/
02:57:56 Normal: Startup of /var/www/html/ -> root@192.168.22.3:/var/www/htm
l/ finished.
```

- File test.html đã đồng bộ đến Web02

```
root@web2: /var/www/html  ×  +  ∨  
root@web2:/var/www/html# ls  
index.nginx-debian.html  test.html  
root@web2:/var/www/html#  
root@web2:/var/www/html#
```

## Câu 2.

### 2.1 Cài đặt Haproxy trên Load Balancer

- Cài đặt dịch vụ.

```
# sudo apt update
```

```
# sudo apt install haproxy -y
```

### 2.2 Cấu hình Haproxy

- Sao lưu file cấu hình mặc định (để đề phòng):

```
# sudo cp /etc/haproxy/haproxy.cfg /etc/haproxy/haproxy.cfg.bak
```

- Chỉnh sửa file cấu hình Haproxy:
- Giữ nguyên phần global.
- Giữ nguyên phần defaults
- Thêm mới phần frontend http\_front theo VM
  - bind \*:80: Lắng nghe trên port 80 cho mọi IP vào Load Balancer (192.168.22.1).
  - mode http: Xác định chế độ HTTP.
  - option forwardfor: Thêm header X-Forwarded-For để backend (Web01, Web02) biết IP của client.
  - default\_backend http\_back: Chuyển mọi request đến backend http\_back.
- Thêm mới phần backend http\_back theo VM
  - mode http: Đảm bảo backend hoạt động ở chế độ HTTP.
  - balance roundrobin: Sử dụng thuật toán round-robin để phân phối request lần lượt giữa Web01 và Web02.
  - option httpchk GET / HTTP/1.1\r\nHost:\ localhost: Thiết lập health check bằng cách gửi yêu cầu GET đến / trên mỗi server. Nếu server trả về mã trạng thái 2xx hoặc 3xx, nó được coi là "healthy"; nếu không, Haproxy sẽ ngừng gửi request đến server đó.
  - server web01 192.168.22.2:80 check: Định nghĩa Web01 với health check.
  - server web02 192.168.22.3:80 check: Định nghĩa Web02 với health check.

```
# sudo nano /etc/haproxy/haproxy.cfg
```

```
frontend http_front

    bind *:80

    mode http

    option forwardfor
```



```
default_backend http_back

backend http_back

    mode http

    balance roundrobin

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

    server web01 192.168.22.2:80 check

    server web02 192.168.22.3:80 check
```

- Khởi động lại Haproxy

```
# sudo systemctl restart haproxy
```

```
# sudo systemctl enable haproxy
```

```
# sudo systemctl status haproxy
```

```
root@lb:~# sudo systemctl status haproxy
● haproxy.service - HAProxy Load Balancer
   Loaded: loaded (/lib/systemd/system/haproxy.service; enabled; vendor p
   Active: active (running) since Sat 2025-04-12 03:02:46 UTC; 11min ago
     Docs: man:haproxy(1)
           file:/usr/share/doc/haproxy/configuration.txt.gz
   Main PID: 29232 (haproxy)
     Tasks: 3 (limit: 2324)
    Memory: 69.3M
       CPU: 124ms
    CGroup: /system.slice/haproxy.service
            └─29232 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -p />
            └─29234 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -p />

Apr 12 03:02:45 lb systemd[1]: Starting HAProxy Load Balancer...
Apr 12 03:02:46 lb haproxy[29232]: [NOTICE] (29232) : New worker #1 (2923
Apr 12 03:02:46 lb systemd[1]: Started HAProxy Load Balancer.
lines 1-16/16 (END)
```

## 2.3 Kiểm tra dịch vụ

### 2.3.1 kiểm tra nhật kí

- Để xem Haproxy phân phối request thế nào, bật logging và chỉnh sửa `/etc/rsyslog.d/haproxy.conf``:
- Khởi động lại rsyslog:

```
# sudo systemctl restart rsyslog
```

- Kiểm tra log:

```
# sudo tail -f /var/log/haproxy.log
```

- Tại Load Balance dùng lệnh `tail -f /var/log/haproxy.log` để xem log hiện tại
- Dùng Web01 hoặc Web02 yêu cầu `curl 192.168.22.1` kiểm tra.

```
root@lb: /var/log
s /usr/sbin/haproxy
Apr 12 03:41:34 lb haproxy[29232]: [ALERT] (29232) : Current worker #1 (29234) exited with code 143 (Terminated)
Apr 12 03:41:34 lb haproxy[29232]: [WARNING] (29232) : All workers exited. Exiting... (0)
Apr 12 03:41:34 lb haproxy[29668]: [WARNING] (29668) : parsing [/etc/haproxy/haproxy.cfg:45]: 'option httpchk' : hiding headers or body at the end of the version string is deprecated. Please, consider to use 'http-check send' directive instead.
Apr 12 03:41:34 lb haproxy[29668]: [NOTICE] (29668) : New worker #1 (29670) forked
Apr 12 03:41:55 lb haproxy[29670]: 192.168.22.2:47884 [12/Apr/2025:03:41:55.394] http_front http_back/web01 0/0/0/1/1 200 822 - - ---- 1/1/0/0/0 0/0 "GET / HTTP/1.1"
Apr 12 03:42:13 lb haproxy[29670]: 192.168.22.3:47084 [12/Apr/2025:03:42:13.907] http_front http_back/web02 0/0/0/1/1 200 822 - - ---- 1/1/0/0/0 0/0 "GET / HTTP/1.1"
Apr 12 03:43:31 lb haproxy[29670]: 154.81.156.54:34120 [12/Apr/2025:03:43:31.780] http_front http_back/web01 0/0/0/1/1 200 606 - - ---- 1/1/0/0/0 0/0 "GET / HTTP/1.1"
```

### 2.3.2 kiểm tra stop dịch vụ Web01

- Dùng lệnh `'systemctl stop nginx'` để dừng dịch vụ

```
root@web1: /var/www/html
root@web1:/var/www/html# sudo systemctl stop nginx
root@web1:/var/www/html# sudo systemctl start nginx
root@web1:/var/www/html# sudo systemctl stop nginx
root@web1:/var/www/html# curl 192.168.22.1
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
```

- Kiểm tra nhật kí ta thấy thông báo trạng trạng Web01 đã bị DOWN và và chuyển request sang Web02 khi yêu c

```
root@lb: /var/log
Apr 12 03:48:09 lb haproxy[29670]: Server http_back/web01 is DOWN, reason: Layer4 connection problem, info: "Connection refused", check duration: 0ms. 1 active and 0 backup servers left. 0 sessions active, 0 requeued, 0 remaining in queue.
Apr 12 03:48:09 lb haproxy[29670]: [WARNING] (29670) : Server http_back/web01 is DOWN, reason: Layer4 connection problem, info: "Connection refused", check duration: 0ms. 1 active and 0 backup servers left. 0 sessions active, 0 requeued, 0 remaining in queue.
```

## Câu 3.

### 3.1 Triển khai MySQL Galera Cluster trên Web01, Web02 và Database Server.

- Cài đặt và kiểm tra trạng thái dịch vụ:

```
# sudo apt update
```

```
# sudo apt install mariadb-server galera-4 -ysudo systemctl status mysql
```

- Dừng dịch vụ MariaDB trên cả 3 server để chuẩn bị cấu hình:

```
# sudo systemctl stop mariadb
```

### 3.2 Cấu hình MariaDB Galera Cluster

- Giải thích:
  - `wsrep_on=ON`: Kích hoạt Galera.
  - `wsrep_provider`: Đường dẫn đến thư viện Galera.
  - `wsrep_cluster_name`: Tên cluster để các node nhận diện.
  - `wsrep_cluster_address`: Danh sách IP của các node trong cluster.
  - `wsrep_node_address`: IP của node hiện tại.
  - `wsrep_node_name`: Tên node để dễ nhận diện.
  - `wsrep_sst_method=rsync`: Phương thức đồng bộ trạng thái.
  - `bind-address=0.0.0.0`: Cho phép kết nối từ mọi IP.

#### 3.2.1 chỉnh sửa Web01 tại tệp `/etc/mysql/mariadb.conf.d/60-galera.cnf`:

```
[galera]
# Mandatory settings
wsrep_on = ON
wsrep_provider = /usr/lib/galera/libgalera_smm.so
wsrep_cluster_name = "galera_cluster"
wsrep_cluster_address = "gcomm://192.168.22.2,192.168.22.3,192.168.22.4"
wsrep_node_address = "192.168.22.2"
wsrep_node_name = "web01"
binlog_format = row
default_storage_engine = InnoDB
innodb_autoinc_lock_mode = 2
wsrep_sst_method = rsync
```

```
# Allow server to accept connections on all interfaces
bind-address      = 0.0.0.0
```

### 3.2.2 chỉnh sửa Web02 tại tệp /etc/mysql/mariadb.conf.d/60-galera.cnf:

```
[galera]
# Mandatory settings
wsrep_on          = ON
wsrep_provider     = /usr/lib/galera/libgalera_smm.so
wsrep_cluster_name = "galera_cluster"
wsrep_cluster_address = "gcomm://192.168.22.2,192.168.22.3,192.168.22.4"
wsrep_node_address = "192.168.22.3"
wsrep_node_name    = "web02"
binlog_format      = row
default_storage_engine = InnoDB
innodb_autoinc_lock_mode = 2
wsrep_sst_method    = rsync

# Allow server to accept connections on all interfaces
bind-address      = 0.0.0.0
```

### 3.2.3 chỉnh sửa Database Server tại tệp /etc/mysql/mariadb.conf.d/60-galera.cnf:

```
[galera]
# Mandatory settings
wsrep_on          = ON
wsrep_provider     = /usr/lib/galera/libgalera_smm.so
wsrep_cluster_name = "galera_cluster"
wsrep_cluster_address = "gcomm://192.168.22.2,192.168.22.3,192.168.22.4"
wsrep_node_address = "192.168.22.4"
wsrep_node_name    = "db01"
binlog_format      = row
default_storage_engine = InnoDB
innodb_autoinc_lock_mode = 2
wsrep_sst_method    = rsync

# Allow server to accept connections on all interfaces
bind-address      = 0.0.0.0
```

## 3.3 Khởi động MariaDB Galera Cluster

### 3.3.1 Trên Web01:

```
# sudo galera_new_cluster
```

- Kiểm tra trạng thái:

```
# sudo systemctl status mariadb
```

```
root@web1:/etc/mysql/mariadb.conf.d# sudo galera_new_cluster
root@web1:/etc/mysql/mariadb.conf.d# sudo systemctl status mariadb
● mariadb.service - MariaDB 10.6.21 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor p>
   Active: active (running) since Sat 2025-04-12 04:19:29 UTC; 12s ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
   Process: 32351 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d>
   Process: 32352 ExecStartPre=/bin/sh -c systemctl unset-environment _WSR>
   Process: 32354 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery >
   Process: 32448 ExecStartPost=/bin/sh -c systemctl unset-environment _WS>
   Process: 32450 ExecStartPost=/etc/mysql/debian-start (code=exited, stat>
   Main PID: 32430 (mariabdb)
   Status: "Taking your SQL requests now..."
     Tasks: 18 (limit: 15340)
    Memory: 87.7M
           CPU: 1.341s
    CGroup: /system.slice/mariadb.service
            └─32430 /usr/sbin/mariabdb --wsrep-new-cluster --wsrep_start_p>
```

- Đăng nhập vào MariaDB để kiểm tra:

```
# mariadb -u root -p
```

- Chạy lệnh:

```
# SHOW STATUS LIKE 'wsrep_cluster_size';
```

```
MariaDB [(none)]> SHOW STATUS LIKE 'wsrep_cluster_size';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| wsrep_cluster_size | 1     |
+-----+-----+
1 row in set (0.002 sec)

MariaDB [(none)]>
```

- wsrep\_cluster\_size = 1, node đầu tiên đã hoạt động.

### 3.3.2 Trên Web02 và Database Server

- Khởi động lại dịch vụ:

```
# sudo systemctl start mariadb
```

- Quay lại Web01, kiểm tra lại:

```
# mariadb -u root -p
```

```
# SHOW STATUS LIKE 'wsrep_cluster_size';
```

```
MariaDB [(none)]> SHOW STATUS LIKE 'wsrep_cluster_size';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| wsrep_cluster_size | 3     |
+-----+-----+
1 row in set (0.003 sec)

MariaDB [(none)]>
```

- wsrep\_cluster\_size = 3, cả ba node đã tham gia cluster thành công.

### 3.3.3 Kiểm tra đồng bộ dữ liệu

- Trên Web01, tạo một database thử:

```
# CREATE DATABASE test_db;
```

- Trên Web02 và Database Server, kiểm tra:

```
# mariadb -u root -p
```

```
# SHOW DATABASES;
```

- Kiểm tra thấy test\_db, cluster đang đồng bộ đúng.

The image contains four terminal screenshots arranged in a 2x2 grid. The top-left terminal shows the creation of a database named 'test\_db' and a list of databases including 'test\_db'. The top-right terminal shows a successful connection to MariaDB and a list of databases including 'test\_db'. The bottom-left terminal shows a log of HTTP requests being handled by haproxy. The bottom-right terminal shows another successful connection to MariaDB and a list of databases including 'test\_db'.

### 3.4 Cấu hình Haproxy để điều phối traffic đến MariaDB Galera Cluster

#### 3.4.1 Cập nhật cấu hình Haproxy trên Load Balancer

- Chỉnh sửa file cấu hình Haproxy:

```
# sudo nano /etc/haproxy/haproxy.cfg
```

- Thêm vào:

```
listen mariadb_cluster
    bind *:3306
    mode tcp
    balance roundrobin
    # option mysql-check user haproxy_check
    server web01 192.168.22.2:3306 check
    server web02 192.168.22.3:3306 check
    server db01 192.168.22.4:3306 check
```

- Giải thích:

- bind \*:3306: Lắng nghe trên port 3306.
- mode tcp: Dùng TCP cho MariaDB.



- balance roundrobin: Phân phối kết nối luân phiên.
- option mysql-check user haproxy\_check: Health check bằng user haproxy\_check.
- server ... check: Định nghĩa 3 node với health check.

### 3.4.2 Tạo user haproxy\_check trên MariaDB Cluster

- Trên Web01
- Truy cập vào Mariadb tạo User haproxy\_check:

```
MariaDB [(none)]> CREATE USER 'haproxy_check'@'%' IDENTIFIED BY 'vhost@54321';
IVILEGES;Query OK, 0 rows affected (0.033 sec)

MariaDB [(none)]> GRANT USAGE ON *.* TO 'haproxy_check'@'%';
Query OK, 0 rows affected (0.024 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
```

### 3.4.3 Kiểm tra và khởi động Haproxy trên Load Balancer

- Kiểm tra trạng thái

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

```
# sudo systemctl restart haproxy
```

```
# sudo systemctl status haproxy
```

### 3.4.4 Truy cập database qua haproxy

Tại VM Web01 kết nối database thông qua haproxy 192.168.22.1

```
root@web1: /
root@web1:/# mysql -u haproxy_check -p -h 192.168.22.1
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 4254
Server version: 10.6.21-MariaDB-0ubuntu0.22.04.2 Ubuntu 22.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

- Cho thấy kết nối thành công

## Câu 4.

### 4.1 Tạo database

- Tạo Database Wordpress

```
MariaDB [(none)]> CREATE DATABASE wp_database;  
Query OK, 1 row affected (0.118 sec)  
  
MariaDB [(none)]> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| sys |  
| test_db |  
| wp_database |  
+-----+  
6 rows in set (0.002 sec)  
  
MariaDB [(none)]>
```

### 4.2 Tạo và phân quyền user cho database

- Tạo user WordPress và cấp quyền cho 5 địa chỉ IP

```
CREATE USER 'wp_user'@'localhost' IDENTIFIED BY 'vhost@54321';  
CREATE USER 'wp_user'@'192.168.22.2' IDENTIFIED BY 'vhost@54321';  
CREATE USER 'wp_user'@'192.168.22.3' IDENTIFIED BY 'vhost@54321';  
CREATE USER 'wp_user'@'125.212.250.9' IDENTIFIED BY 'vhost@54321';  
CREATE USER 'wp_user'@'113.161.61.219' IDENTIFIED BY 'vhost@54321';  
  
GRANT ALL PRIVILEGES ON wp_database.* TO 'wp_user'@'localhost';  
GRANT ALL PRIVILEGES ON wp_database.* TO 'wp_user'@'192.168.22.2';  
GRANT ALL PRIVILEGES ON wp_database.* TO 'wp_user'@'192.168.22.3';  
GRANT ALL PRIVILEGES ON wp_database.* TO 'wp_user'@'125.212.250.9';  
GRANT ALL PRIVILEGES ON wp_database.* TO 'wp_user'@'113.161.61.219';  
  
FLUSH PRIVILEGES;
```

### 4.3 Cài đặt WordPress đơn giản trên Web01

- Tải và giải nén Wordpress

```
# cd /var/www/html
```

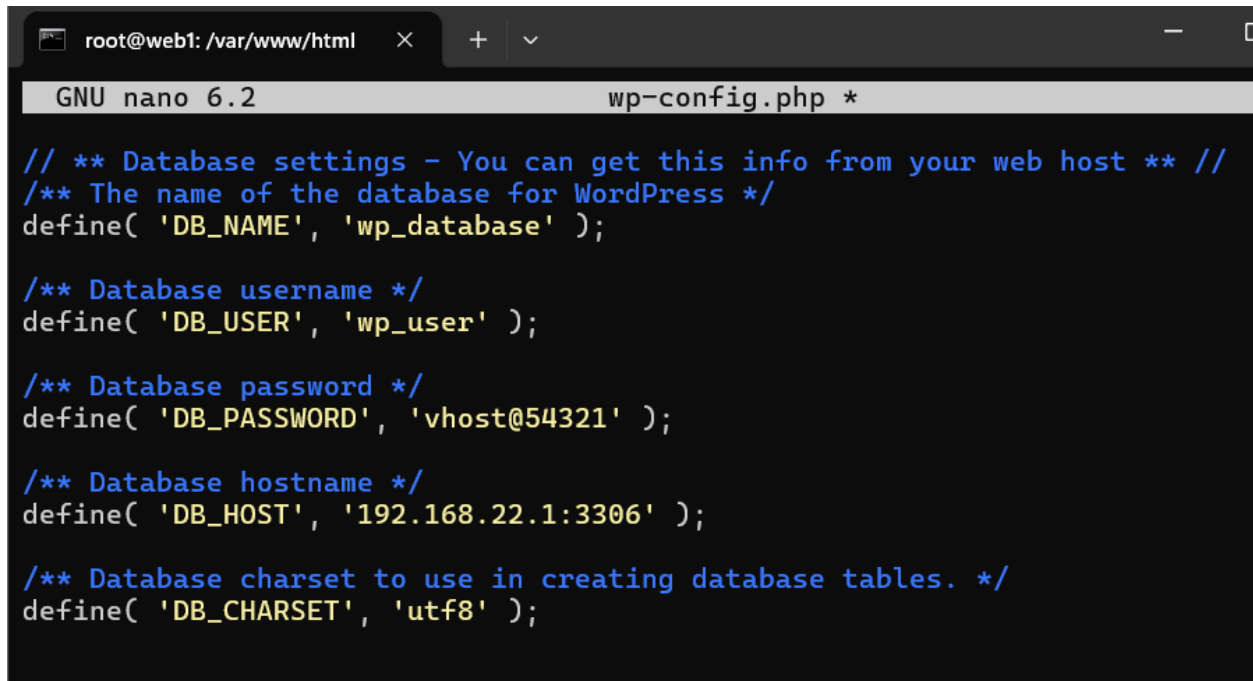
```
# sudo wget https://wordpress.org/latest.tar.gz
```

```
# sudo tar -xvzf latest.tar.gz
```

```
# sudo mv wordpress/* .
```

```
# sudo rm -rf wordpress latest.tar.gz
```

- Cấu hình file wp-config.php



```
GNU nano 6.2 wp-config.php *

// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wp_database' );

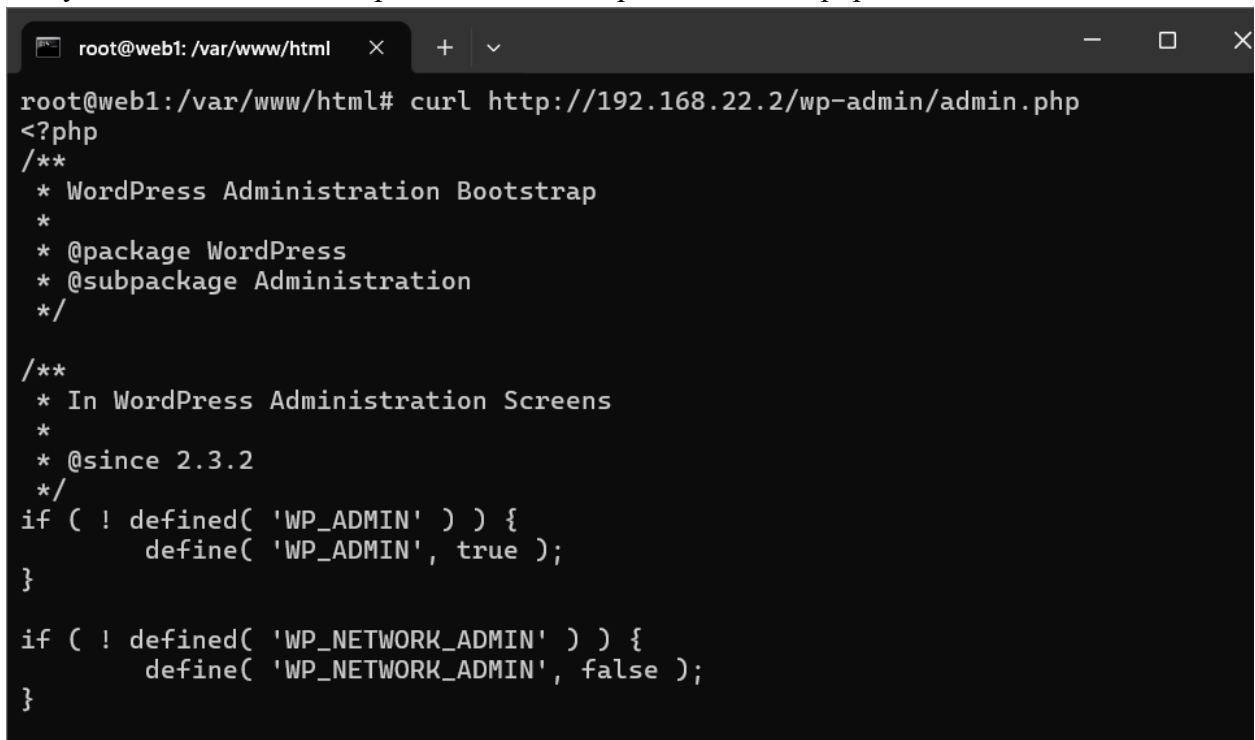
/** Database username */
define( 'DB_USER', 'wp_user' );

/** Database password */
define( 'DB_PASSWORD', 'vhost@54321' );

/** Database hostname */
define( 'DB_HOST', '192.168.22.1:3306' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );
```

- Gửi yêu cầu kết nối : curl http://192.168.22.2/wp-admin/admin.php



```
root@web1: /var/www/html# curl http://192.168.22.2/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 );
}
```

- Wordpress đã chạy thành công.
- Tương tự Web02.

---- Hết ----