

***System Admin***

***Training Assignments***

|  |  |
| --- | --- |
| Program Code |  |
| Issue/Revision | x/y |
| Effective date | 04/Aug /2023 |

**Assignment day 23: Setting Up Full HA WebApp WordPress – Onpremise Infrastructure**

**Mục Lục**

[Kiến trúc hạ tầng 2](#_Toc114605999)

[Setup máy ảo VM theo quy hoạch: 3](#_Toc519460682)

[Setup Server Cân bằng tải LB 3](#_Toc1624649515)

[Cài đặt phần mềm hệ thống Haproxy và Keepalived 4](#_Toc200980105)

[Cấu hình Haproxy 4](#_Toc478411657)

[Cấu hình Keepalived 6](#_Toc298493752)

[Kiểm thử hoạt động 8](#_Toc1097899038)

[Cài đặt Database 9](#_Toc421893373)

[Cấu hình server MySQL Master VM-DB-01 10](#_Toc1631167406)

[Cấu hình server MySQL Slave VM-DB-02 10](#_Toc1691452777)

[Kiểm tra đồng bộ hoạt động 11](#_Toc1867479739)

[Tạo Database cho WordPress hoạt động 12](#_Toc1958100200)

[Cài đặt NFS Storage Server 12](#_Toc1822629988)

[Cài đặt NFS Storage Client 13](#_Toc386028704)

[Cài đặt WebApp 14](#_Toc885942504)

[Cài đặt Apache 14](#_Toc229181247)

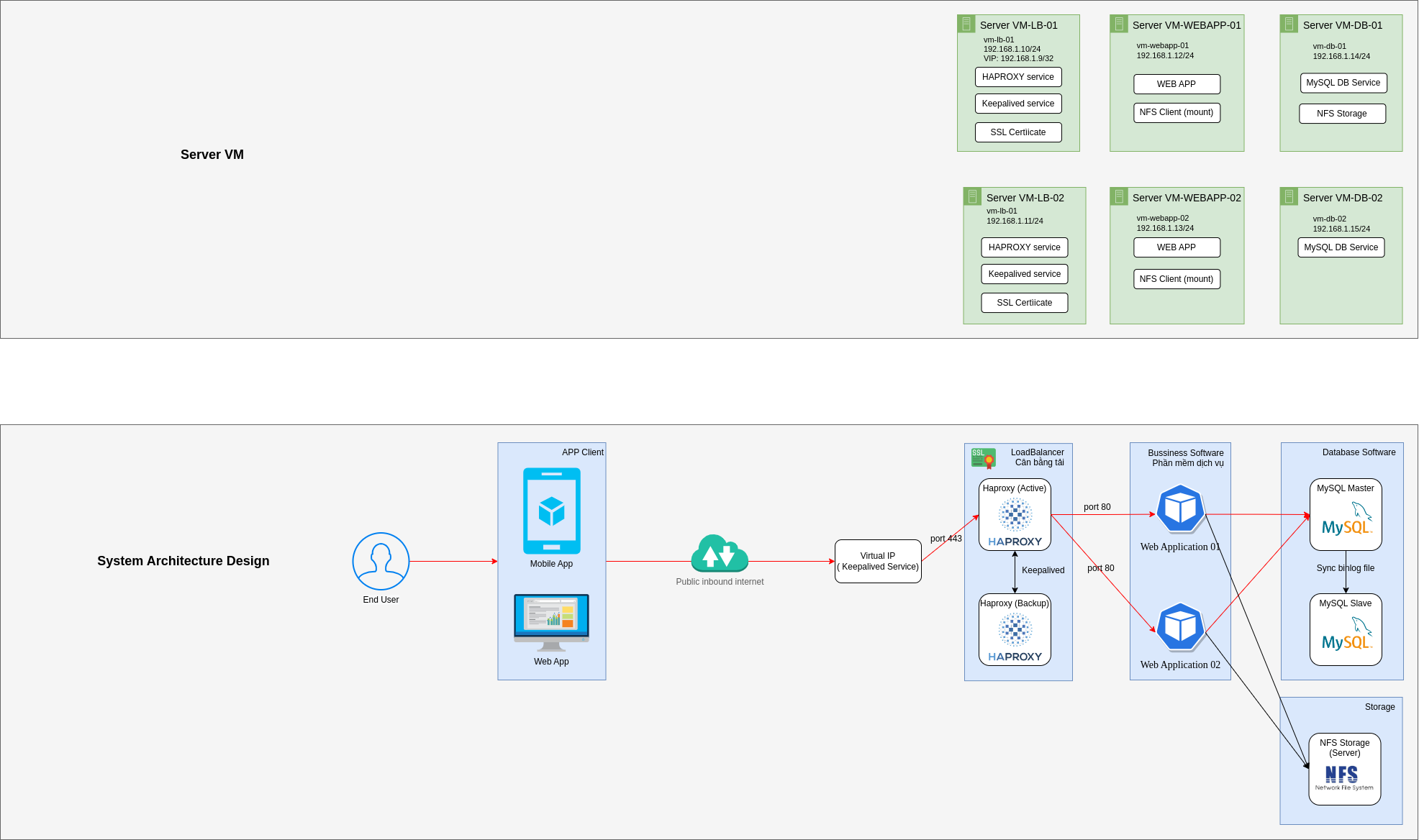
[Cài đặt PHP 7 15](#_Toc204438250)

[Tải wordpress và giải nén 15](#_Toc1576130338)

[Cấu hình wordpress kết nối đến DB 16](#_Toc603776836)

[Tiến hành cài đặt wordpress 16](#_Toc2063444278)

## Kiến trúc hạ tầng



## Setup máy ảo VM theo quy hoạch:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STT | OS Hostname | CPU (core) | RAM (GB) | DISK (GB) | OS | IP | Virtual IP |
| 1 | VM-LB-01 | 1 | 1 | 20 | CentOS 7.9 64bit | 192.168.1.10 | 192.168.1.9 |
| 2 | VM-LB-02 | 1 | 1 | 20 | CentOS 7.9 64bit | 192.168.1.11 |
| 3 | VM-WebApp-01 | 1 | 1 | 20 | CentOS 7.9 64bit | 192.168.1.12 |  |
| 4 | VM-WebApp-02 | 1 | 1 | 20 | CentOS 7.9 64bit | 192.168.1.13 |  |
| 5 | VM-DB-01 | 1 | 1 | 20 | CentOS 7.9 64bit | 192.168.1.14 |  |
| 6 | VM-DB-02 | 1 | 1 | 20 | CentOS 7.9 64bit | 192.168.1.15 |  |

## Setup Server Cân bằng tải LB

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STT | OS Hostname | CPU (core) | RAM (GB) | DISK (GB) | OS | IP | Virtual IP |
| 1 | VM-LB-01 | 1 | 1 | 20 | CentOS 7.9 64bit | 192.168.1.10 | 192.168.1.9 |
| 2 | VM-LB-02 | 1 | 1 | 20 | CentOS 7.9 64bit | 192.168.1.11 |

### Cài đặt phần mềm hệ thống Haproxy và Keepalived

Mục đích:

Haproxy là nhiệm vụ cân bằng tải, Terminate SSL

Keepalived: Kiểm tra heatbeat giữa 2 server haproxy để auto chuyển VIP

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | OS Hostname | System Software | Location File | Start Service |
| 1 | VM-LB-01 | haproxy | /etc/haproxy/haproxy.cfg | sudo systemctl restart haproxy |
| 2 | VM-LB-01 | keepalived | /etc/keepalived/keepalived.conf | sudo systemctl restart keepalived |
| 3 | VM-LB-02 | haproxy | /etc/haproxy/haproxy.cfg | sudo systemctl restart haproxy |
| 4 | VM-LB-02 | keepalived | /etc/keepalived/keepalived.conf | sudo systemctl restart keepalived |

Cài đặt in Server VM-LB-01 và VM-LB-02 ( giống hệt nhau)

|  |
| --- |
| $ sudo yum install haproxy keepalived |

### Cấu hình Haproxy

Config file Haproxy in Server VM-LB-01 và VM-LB-02 ( giống hệt nhau)

|  |
| --- |
| $ sudo vim /etc/haproxy/haproxy.cfg |

Option 1:

Cấu hình không có cert SSL:

|  |
| --- |
| global  maxconn 4096  user haproxy  group haproxy  daemon   defaults  log global  mode http  option httplog  option dontlognull  retries 3  option redispatch  maxconn 2000  timeout connect 5000  timeout client 50000  timeout server 50000   listen webservice-frontend domain-example.com:80  mode http  stats enable  stats auth username:password  balance roundrobin  option httpchk HEAD /healthcheck HTTP/1.0  option forwardfor  option httpclose  cookie SRVNAME insert indirect nocache  server VM-Webapp-01 192.168.1.12:80 cookie p1 weight 1 check inter 2000  server VM-Webapp-02 192.168.1.13:80 cookie p2 weight 1 check inter 2000 |

Option2:

Cấu hình có cert SSL

|  |
| --- |
| $ sudo vim /etc/haproxy/haproxy.cfg |

|  |
| --- |
| global  maxconn 4096  user haproxy  group haproxy  tune.ssl.default-dh-param 2048  daemon   defaults  log global  mode http  option httplog  option dontlognull  retries 3  option redispatch  maxconn 2000  timeout connect 5000  timeout client 50000  timeout server 50000  stats enable  stats auth username:password  stats uri /stats   frontend webservice-frontend domain-example.com:443  bind \*:443 ssl crt /etc/haproxy/ssl/cert.pem  mode http  option httpclose  option forwardfor  reqadd X-Forwarded-Proto:\ https  default\_backend webservice-backend   backend webservice-backend  mode http  balance roundrobin  option httpchk HEAD /healthcheck HTTP/1.0  cookie SRVNAME insert indirect nocache  server VM-Webapp-01 192.168.1.12:80 cookie p1 weight 1 check inter 2000  server VM-Webapp-02 192.168.1.13:80 cookie p2 weight 1 check inter 2000 |

|  |
| --- |
| $sudo systemctl restart haproxy |

### Cấu hình Keepalived

Cấu hình Keepalived in Server VM-LB-01

|  |
| --- |
| $sudo vim /etc/keepalived/keepalived.conf |

|  |
| --- |
| vrrp\_script haproxy-check {  script "killall -0 haproxy"  interval 2  weight 10  }   vrrp\_instance webservice-frontend {  state MASTER  priority 101 # do uu tien master 101; backup 100  interface eth0 # interface tao Vitural IP (VIP)  virtual\_router\_id 61  advert\_int 2    virtual\_ipaddress {  192.168.1.9  }    track\_script {  haproxy-check  }  } |

Restart/start service sau khi cấu hình xong

Cấu hình Keepalived in Server VM-LB-02

|  |
| --- |
| $sudo vim /etc/keepalived/keepalived.conf |

|  |
| --- |
| vrrp\_script haproxy-check {  script "killall -0 haproxy"  interval 2  weight 10  }   vrrp\_instance webservice-frontend {  state BACKUP  priority 100 # do uu tien master 101; backup 100  interface eth0 # interface tao Vitural IP (VIP)  virtual\_router\_id 61  advert\_int 2    virtual\_ipaddress {  192.168.1.9  }    track\_script {  haproxy-check  }  } |

Restart/start service sau khi cấu hình xong

|  |
| --- |
| sudo systemctl restart keepalived |

### Kiểm thử hoạt động

Ảnh có chứa văn bản, biểu đồ, ảnh chụp màn hình

Mô tả được tự động tạo

Tắt card mạng Haproxy trên Server VM-LB-01 . Sau đó xác nhận rằng VIP 192.168.1.9 auto chuyển sang server Haproxy Backup.

Kiểm tra bằng lệnh “ifconfig” or “ip a”

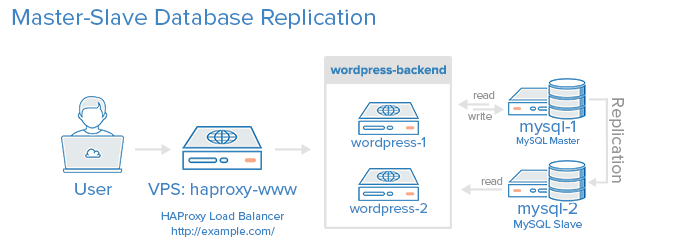
Bật lại card mạng trên Server VM-LB-01. Sau đó xác nhận rằng VIP 192.168.1.9 auto chuyển sang server Haproxy Active.

Hoàn tất cài đặt haproxy và keepalived.

## Cài đặt Database

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STT | OS Hostname | CPU (core) | RAM (GB) | DISK (GB) | OS | IP | Virtual IP |
| 5 | VM-DB-01 | 1 | 1 | 20 | CentOS 7.9 64bit | 192.168.1.14 |  |
| 6 | VM-DB-02 | 1 | 1 | 20 | CentOS 7.9 64bit | 192.168.1.15 |  |

Cài đặt Database MySQL theo mô hình 01 Master và 01 Slave



Cài đặt Mysql on Centos 7 trên cả 2 server VM-DB-01 và VM-DB-02

|  |
| --- |
| $sudo yum localinstall <https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm>$sudo yum install mysql-community-server |

Start/auto start dịch vụ

|  |
| --- |
| $sudo systemctl enable mysqld $sudo systemctl start mysqld |

đặt password cho user root

|  |
| --- |
| $mysql\_secure\_installation |

### Cấu hình server MySQL Master VM-DB-01

sudo vim /etc/my.cnf

sửa lại cấu hình như sau:

|  |
| --- |
| bind-address = 192.168.1.14  server-id = 1  log\_bin = mysql-bin |

* **bind-address**: địa chỉ IP MySQL sẽ nghe. Thường là private IP address của *VM-DB-01*
* **server-id**: ID riêng của server. Ở đây ta sẽ đặt giá trị cho nó bằng 1 rồi uncomment( xóa dấu # để dòng lệnh được chạy) dòng này.
* **log\_bin**: Vị trí của file log nhị phân. Uncomment dòng này.

Khởi động lại mysql sau khi sửa cấu hình

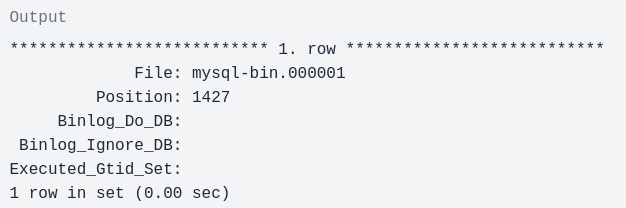
|  |
| --- |
| sudo systemctl restart mysqld |

Tạo user “replica” cho phép DB master và slave đồng bộ nhau:

|  |
| --- |
| $mysql -uroot -p mysql> CREATE USER '[replica'@'192.168.121.14](mailto:replica'@'192.168.121.14)' IDENTIFIED BY 'strong\_password'; mysql> GRANT REPLICATION SLAVE ON \*.\* TO '[replica'@'192.168.121.14](mailto:replica'@'192.168.121.14)'; |

kiểm tra trạng thái master

|  |
| --- |
| mysql -uroot -p  mysql> SHOW MASTER STATUS\G |



### Cấu hình server MySQL Slave VM-DB-02

sudo vim /etc/my.cnf

|  |
| --- |
| bind-address = 192.168.1.15  server-id = 2  log\_bin = mysql-bin |

* **bind-address**: địa chỉ IP MySQL sẽ nghe. Thường là private IP address của *VM-DB-02*
* **server-id**: ID riêng của server. Ở đây ta sẽ đặt giá trị cho nó bằng 2 rồi uncomment( xóa dấu # để dòng lệnh được chạy) dòng này.
* **log\_bin**: Vị trí của file log nhị phân. Uncomment dòng này.

Khởi động lại mysql sau khi sửa cấu hình

|  |
| --- |
| sudo systemctl restart mysqld |

Cấu hình kết nối đồng bộ logbin giữa master và slave

|  |
| --- |
| mysql -uroot -p  mysql> STOP SLAVE;  mysql> CHANGE MASTER TO  mysql> MASTER\_HOST='192.168.121.59',  mysql> MASTER\_USER='replica',  mysql> MASTER\_PASSWORD='strong\_password',  mysql> MASTER\_LOG\_FILE='mysql-bin.000001',  mysql> MASTER\_LOG\_POS=1427; |

Start database SLAVE

|  |
| --- |
| mysql -uroot -p  mysql> START SLAVE; |

### Kiểm tra đồng bộ hoạt động

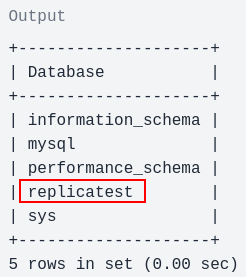
Tạo DB “replicatest” trên server Master. Sau đó vào Server Slave. Xác nhận DB đó được tạo

Trên server Master:

|  |
| --- |
| mysql -uroot -p  CREATE DATABASE replicatest; |

Trên server Slave

|  |
| --- |
| mysql -uroot -p  SHOW DATABASES; |



Done.

### Tạo Database cho WordPress hoạt động

You will need a database to store, collect, and retrieve website information. Now we need to create a MySQL database. First login to MySQL:

$mysql -u root –p

Enter the password, and create a database name:

mysql>

CREATE DATABASE wordpress;

After, grant privileges on the database name, create a database user, and set password:

mysql>

GRANT ALL PRIVILEGES on wordpress.\* to 'wordpress\_user'@'localhost' identified by 'wordpress\_pw';

Flush privileges to make changes.

mysql>FLUSH PRIVILEGES;

And log out from MySQL:

mysql> exit

You have successfully created a database for your WordPress site. Now, change the WordPress configuration file with the database that you created.

## **Cài đặt NFS Storage Server**

chạy chung dịch vụ NFS trên server VM-DB-01. Tận dụng phân vùng lưu trữ tại server VM-DB-01. Mục dịch toàn bộ file của wordpress sẽ được lưu trữ tại đây. Và share ổ cứng NFS cho các client để chạy dịch vụ.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STT | OS Hostname | CPU (core) | RAM (GB) | DISK (GB) | OS | IP | Virtual IP |
| 5 | VM-DB-01 | 1 | 1 | 20 | CentOS 7.9 64bit | 192.168.1.14 |  |

1. Kiểm tra package **portmap** và **nfs-utils** đã được cài đặt trên NFS Server hay chưa. Nếu chưa cần tiến hành cài đặt các package này qua **yum** hoặc **rpm.**

**[root@VM-DB-01]# rpm –qa | grep –e portmap –e nfs**

**[root@VM-DB-01]# yum install portmap nfs-utils**

**[root@VM-DB-01]# rpm –qa | grep –e portmap –e nfs**

portmap-4.0-65.2.2.1

nfs-utils-1.0.9-44.el5

nfs-utils-lib-1.0.8-7.6.el5

1. Cấu hình quản lý truy cập thông qua TCP Wrapper để chỉ cho phép các client ở dải 192.168.1.0/24 được phép truy cập vào dịch vụ NFS trên NFS Server

**[root@VM-DB-01]# vi /etc/hosts.deny**

portmap: ALL

**[root@VM-DB-01]# vi /etc/hosts.allow**

portmap: 192.168.1.

1. Tạo thư mục chia sẻ **/nfsshare** trên NFS Server

**[root@VM-DB-01]# mkdir /nfsshare**

1. Cấu hình chia sẻ thư mục **/nfsshare** trên NFS Server

**[root@VM-DB-01]# vi /etc/exports**

/nfsshare 192.168.1.0/24(rw,sync)

1. Khởi động dịch vụ **portmap** và **nfs** trên NFS Server

**[root@VM-DB-01]# service portmap start**

Starting portmap: [ OK ]

**[root@VM-DB-01]# service nfs start**

Starting NFS services: [ OK ]

Starting NFS quotas: [ OK ]

Starting NFS daemon: [ OK ]

Starting NFS mountd: [ OK ]

## **Cài đặt NFS Storage Client**

Thực hiện cài đặt phần mềm nfs client trên các server sau:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STT | OS Hostname | CPU (core) | RAM (GB) | DISK (GB) | OS | IP | Virtual IP |
| 3 | VM-WebApp-01 | 1 | 1 | 20 | CentOS 7.9 64bit | 192.168.1.12 |  |
| 4 | VM-WebApp-02 | 1 | 1 | 20 | CentOS 7.9 64bit | 192.168.1.13 |  |

Làm lần lượt trên cả 2 server VM-WebApp-01 và VM-WebApp-02

Toàn bộ source code của wordpress sẽ được lưu tại thư mục này: **/var/www/html/nfs-storage**

1. Trên NFS Client, tạo mountpoint và thực hiện việc mount thư mục chia sẻ (sử dụng account root). Kiểm tra việc mount.

**[root@VM-WebApp-01]# mkdir /var/www/html/nfs-storage**

**[root@VM-WebApp-01]# mount –t nfs 192.168.1.14:/nfsshare /var/www/html/nfs-storage**

**[root@VM-WebApp-01]# mount**

…

192.168.1.14:/nfsshare on /var/www/html/nfs-storage type nfs (rw,addr=192.168.1.14)

**[root@VM-WebApp-01]# ls /var/www/html/nfs-storage**

1. Trên NFS Client, thử tạo file mới bên trong thư mục chia sẻ

**[root@VM-WebApp-01]# cd /var/www/html/nfs-storage**

**[root@VM-WebApp-01]# mkdir test**

mkdir: cannot create directory `test': Permission denied

1. Trên NFS Server, thay đổi thông tin về thư mục chia sẻ để cho phép user **root** có toàn quyền truy cập. Sau khi thay đổi, reload lại cấu hình cho **nfs**

**[root@NFSServer]# vi /etc/exports**

*/nfsshare* *192.168.1.0/24(***no\_root\_squash***,rw,sync)*

**[root@NFSServer]# exportfs -r**

1. Trên NFS Client, thử tạo lại file mới bên trong thư mục chia sẻ

**[root@VM-WebApp-01]# cd /var/www/html/nfs-storage**

**[root@VM-WebApp-01]# mkdir test**

**[root@VM-WebApp-01]# ls**

test

1. Trên NFS Server, kiểm tra trạng thái của dịch vụ NFS

**[root@NFSServer]# nfsstat**

**[root@NFSServer]# showmount –-all localhost**

1. (Tùy chọn) Trên NFS Client, cấu hình **/etc/fstab** để tự động mount thư mục chia sẻ mỗi khi khởi động hệ thống

**[root@VM-WebApp-01]# vi /etc/fstab**

…

192.168.1.14:/nfsshare /var/www/html/nfs-storage nfs default 0 0

## Cài đặt WebApp

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STT | OS Hostname | CPU (core) | RAM (GB) | DISK (GB) | OS | IP | Virtual IP |
| 3 | VM-WebApp-01 | 1 | 1 | 20 | CentOS 7.9 64bit | 192.168.1.12 |  |
| 4 | VM-WebApp-02 | 1 | 1 | 20 | CentOS 7.9 64bit | 192.168.1.13 |  |

Làm lần lượt các bước dưới đây trên cả 2 server VM-WebApp-01 VM-WebApp-02

### Cài đặt Apache

yum install httpd

systemctl start httpd.service

systemctl enable httpd.service

Thử truy cập web: http://192.168.1.12 và <http://192.168.1.13>



Chỉnh sửa lại thư mục gốc của apache /var/www/html/

**sudo vim /etc/httpd/conf/httpd.conf**

Change the AllowOverride None line in the above section to read AllowOverride *All*.

Hoặc sửa trực tiếp từ /var/www/html/ thành /var/www/html/nfs-storage/ để website có thể hoạt động với thư mục gốc mới.

### Cài đặt PHP 7

yum install epel-release yum-utils -y

yum install <http://rpms.remirepo.net/enterprise/remi-release-7.rpm>

yum-config-manager --enable remi-php70

yum install php php-mysql

systemctl restart httpd

### Tải wordpress và giải nén

Lưu ý: website sẽ chạy trong thư mục /var/www/html/nfs-storage. là thư mục mount từ NFS server. Cho phép cả 2 server webapp đều có thể đọc ghi từ 1 nguồn lưu trữ duy nhất.

After you have successfully installed LAMP Stack based on the article above, change current directory to web directory:

$ cd /var/www/html/nfs-storage

Now download the latest WordPress file by running this command:

$ wget <https://wordpress.org/latest.tar.gz>

Next, extract the WordPress file in the current directory:

$ tar xvzf latest.tar.gz

After, you need to move files from **wordpress** directory into web directory:

$ mv /var/www/html//nfs-storage/wordpress/\* /var/www/html/nfs-storage

### Cấu hình wordpress kết nối đến DB

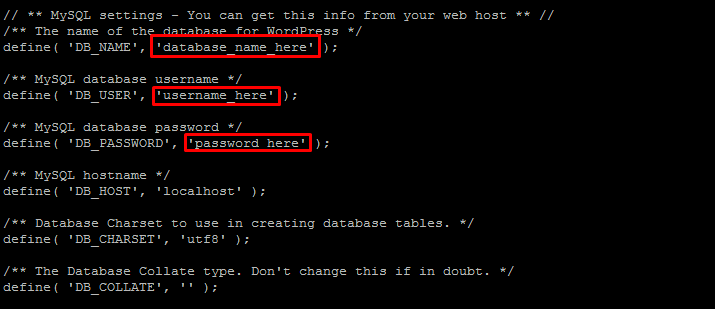
First, rename wp-config-sample.php to wp-config.php:

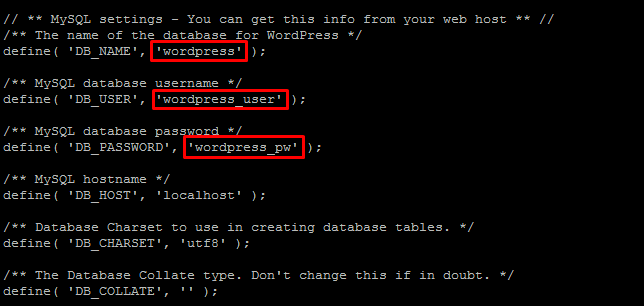
$ mv wp-config-sample.php wp-config.php

Open wp-config.php file:

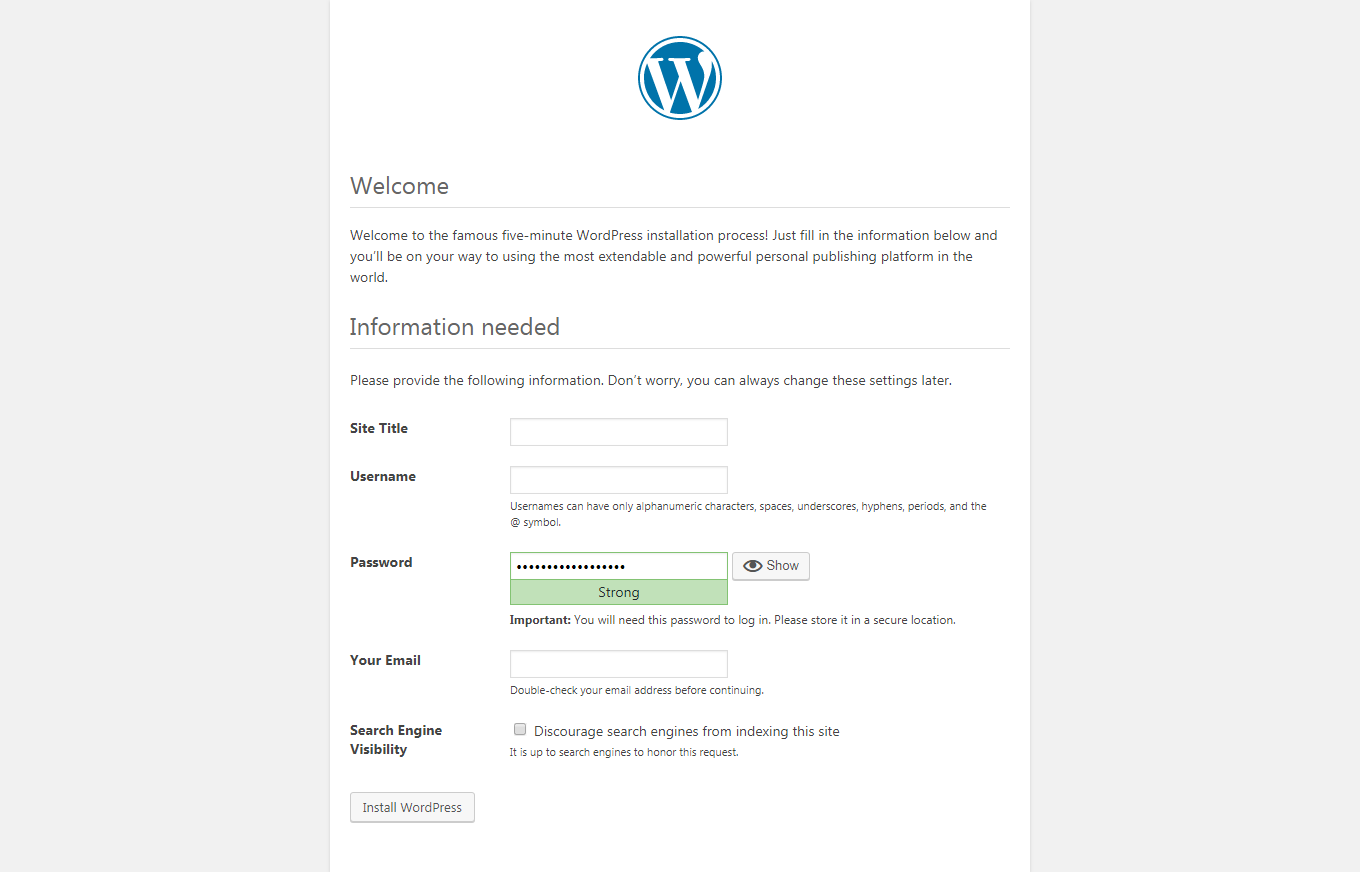
$ vim wp-config.php

Change lines in red:

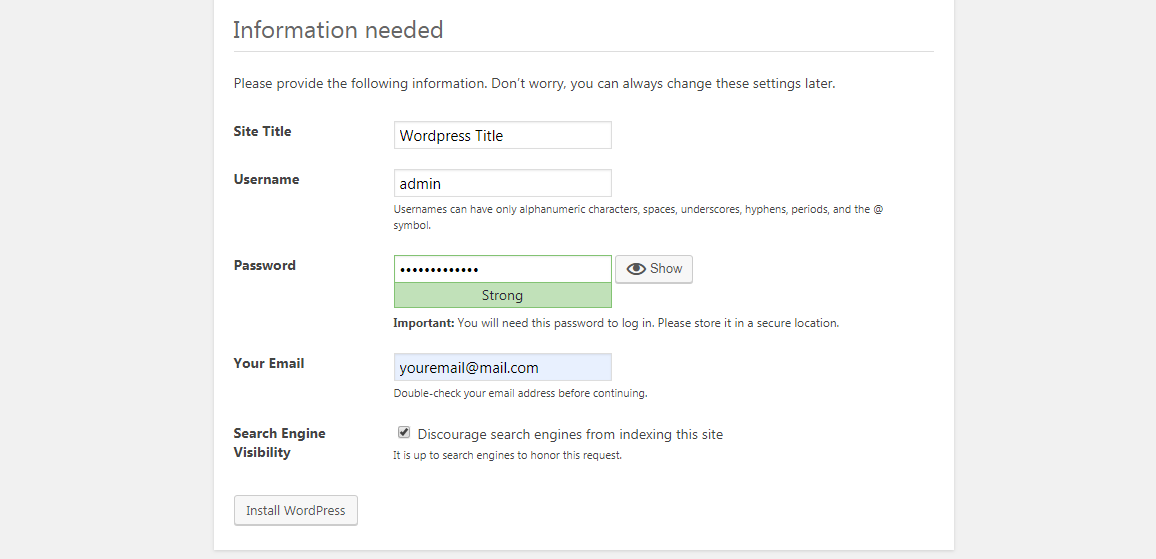


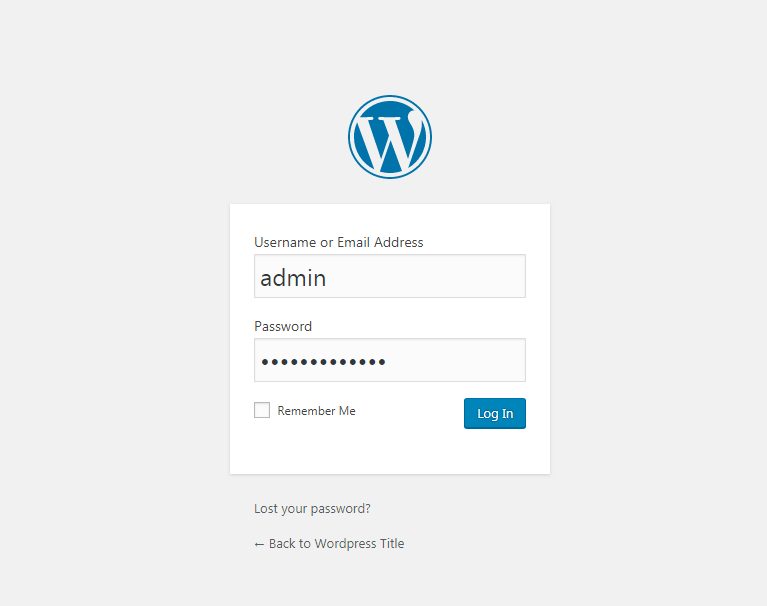
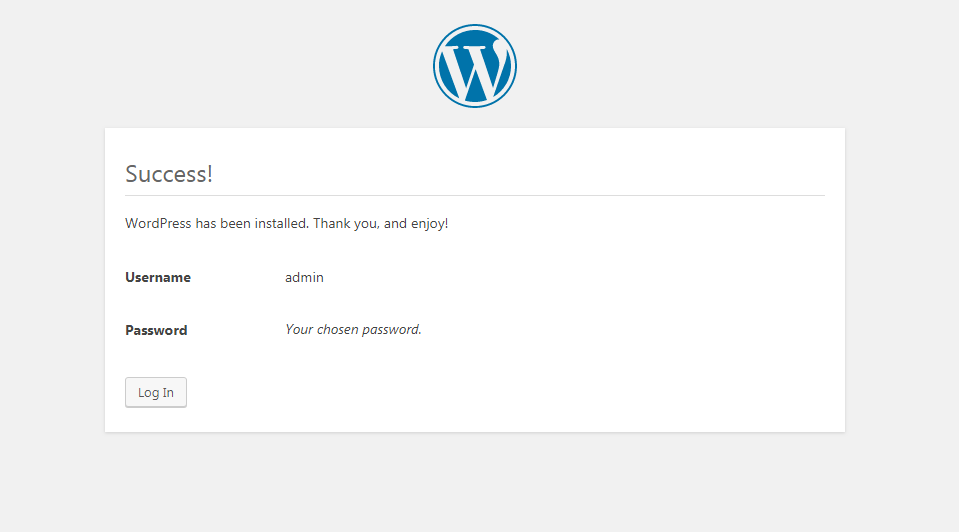
With the database that you created above: 

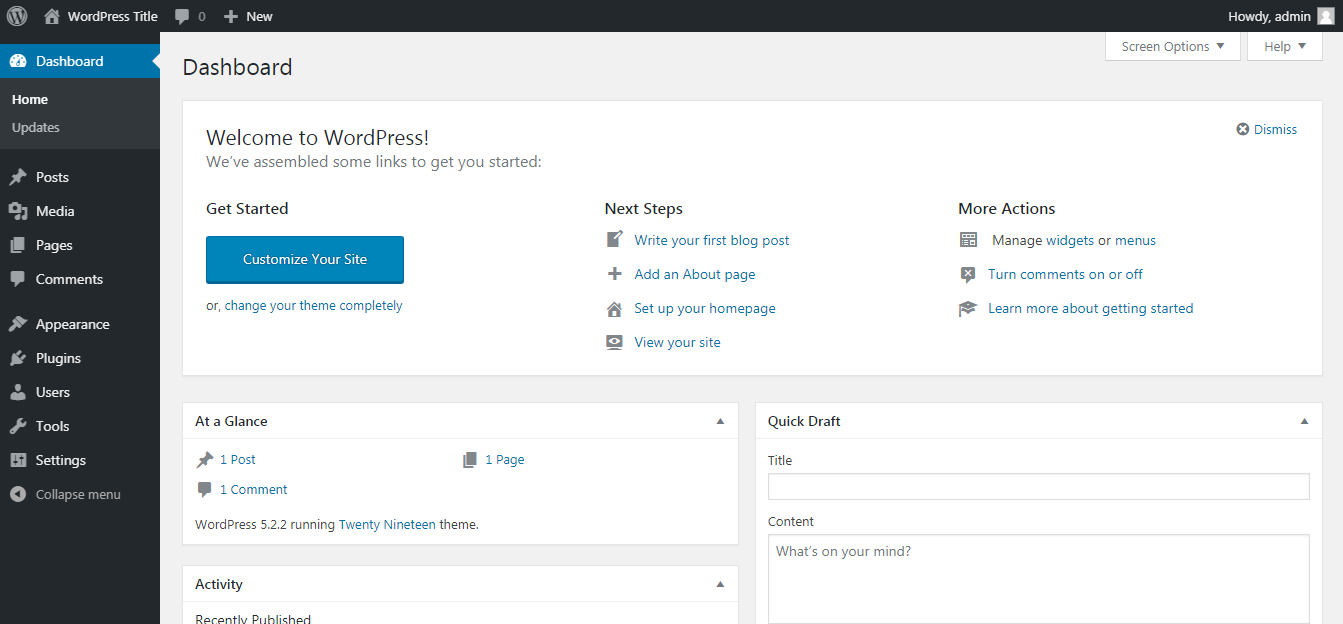
### Tiến hành cài đặt wordpress

Ơpen browser and go to http://92.168.1.12 or http://192.168.1.13 After you have gone to your site, you will get the famous five-minute installation process: 

Provide the information below by your preferences and click on the Install WordPress button to finish the installation process:



Now, log in to your WP Admin Panel: 

And manage your WordPress site: 

Done.