[I. Cơ sở lý thuyết 4](#_Toc62673702)

[1. Các dịch vụ trên dns (domain name system) 4](#_Toc62673703)

[Cơ bản về DNS 4](#_Toc62673704)

[DNS hoạt động như thế nào? 4](#_Toc62673705)

[Cách sử dụng 5](#_Toc62673706)

[2. Giới thiệu về DNS server 5](#_Toc62673707)

[Chức năng của DNS Server 5](#_Toc62673708)

[Các loại DNS Server 5](#_Toc62673709)

[Giới thiệu về recursive resolver 5](#_Toc62673710)

[Giới thiệu về Root nameservers 6](#_Toc62673711)

[Giới thiệu về TLD nameservers 6](#_Toc62673712)

[Giới thiệu về authoritative nameserver 6](#_Toc62673713)

[3. Web server 7](#_Toc62673714)

[Khái niệm 7](#_Toc62673715)

[Các loại Web Server phổ biến 8](#_Toc62673716)

[Một số lưu ý khi sử dụng Web Server 8](#_Toc62673717)

[II. Mô hình triển khai 10](#_Toc62673718)

[2. 1. Triển khai mô hình DNS Server 10](#_Toc62673719)

[3. Triển khai mô hình Web Server 11](#_Toc62673720)

[III. Cài Đặt, cấu hình 12](#_Toc62673721)

[1. Bước 1: Đặt Ip và DNS cho các máy 12](#_Toc62673722)

[2. Bước 2: Cài đặt máy ảo 14](#_Toc62673723)

[3. Bước 3: Đặt hostname, ip và dns 14](#_Toc62673724)

[a. Trên máy DNS Server Master (DNS1) 14](#_Toc62673725)

[b. Cấu hình trên máy DNS Server Slave (DNS2) 17](#_Toc62673726)

[c. Trên máy Client 19](#_Toc62673727)

[4. Bước 4: Tắt Ipv6 24](#_Toc62673728)

[a. Trên máy DNS1 24](#_Toc62673729)

[b. Trên máy DNS2 25](#_Toc62673730)

[5. Bước 5: Kết nối ssh từ máy client đến máy server 26](#_Toc62673731)

[a. Kết nối đến máy DNS1 26](#_Toc62673732)

[b. Kết nối với máy DNS2 26](#_Toc62673733)

[6. Bước 6: Cài đặt Bind 27](#_Toc62673734)

[a. Trên máy DNS1 27](#_Toc62673735)

[b. Cài trên máy DNS2 37](#_Toc62673736)

[c. Trên máy Client 42](#_Toc62673737)

[7. Bước 7: Kiểm tra cấu hình DNS 43](#_Toc62673738)

[a. Khi bật cả hai máy 43](#_Toc62673739)

[b. Tắt máy DNS1 44](#_Toc62673740)

[c. Khi bật lại máy DNS1 44](#_Toc62673741)

[8. Bước 8: Tiến hành cấu hình Web server. 45](#_Toc62673742)

[a. Cấu hình Apache trên máy DNS Server Master 45](#_Toc62673743)

[b. Cấu hình Apache trên máy Server Slave 51](#_Toc62673744)

[9. Bước 9: Cài đặt Mysql 54](#_Toc62673745)

[a. Trên máy DNS Server Master 54](#_Toc62673746)

[b. Trên máy DNS Server Slave 60](#_Toc62673747)

[c. Kiểm tra 65](#_Toc62673748)

[10. Bước 10: Cài đặt php 67](#_Toc62673749)

[a. Trên máy Server Master 67](#_Toc62673750)

[b. Trên máy Server Slave 73](#_Toc62673751)

[11. Bước 11: cài đặt phpmyadmin 75](#_Toc62673752)

[a. Trên Server Master 75](#_Toc62673753)

[b. Trên Server Slave 80](#_Toc62673754)

[12. Bước 12: Cài đặt Keeepalived và Haproxy 80](#_Toc62673755)

[a. Trước khi thực hiện cấu hình, ta tìm hiểu keepalived và haproxy là gì? 80](#_Toc62673756)

[b. Trên máy Server Master 81](#_Toc62673757)

[c. Trên máy Server Slave 85](#_Toc62673758)

[d. Kiểm tra 89](#_Toc62673759)

[e. Cấu hình haproxy 90](#_Toc62673760)

[f. Tạo tên miền cho IP 192.168.88.100 92](#_Toc62673761)

[13. Bước 13: Đồng bộ source code của web 95](#_Toc62673762)

[14. Bước 14: Kiểm tra cuối cùng 97](#_Toc62673763)

[IV. Kết quả và kết luận 98](#_Toc62673764)

[1. Kết quả 98](#_Toc62673765)

[2. Kết luận 98](#_Toc62673766)

[V. Tài liệu kham khảo 99](#_Toc62673767)

# Cơ sở lý thuyết

## Các dịch vụ trên dns (domain name system)

### Cơ bản về DNS

* **DNS** là viết tắt của cụm từ **Domain Name System**, mang ý nghĩa đầy đủ là **hệ thống phân giải tên miền**.
* DNS là một phần trung tâm của cơ sở hạ tầng web, nó thực hiện việc chuyển đổi các tên miền mà con người dễ ghi nhớ (dạng kí tự) sang địa chỉ IP dạng số.

Ví dụ: trên trang web daa của trường có địa chỉ IP là 118.69.123.142 và được chuyển sang tên miền là daa.uit.edu.vn

* DNS giống như một “Danh bạ điện thoại”, nó gán tên miền và lập bản đồ tên miền đến cái địa chỉ IP tương ứng
* Thao tác này của DNS giúp liên kết các thiết bị mạng với nhau nhằm mục đích định vị và gán một địa chỉ cụ thể cho các thông tin trên internet.

### DNS hoạt động như thế nào?

- Không gian tên miền bao gồm một cây tên miền, chia thành các vùng.

- Một vùng DNS bao gồm một tập hợp các nút kết nối được lưu trữ bởi máy chủ tên có thẩm quyền. Máy chủ tên miền có thẩm quyền cho các vùng khác nhau chịu trách nhiệm xuất ánh xạ của các tên miền sang địa chỉ IP. Mỗi nút hay lá của cây có từ không đến nhiều bản ghi tài nguyên, giữ thông tin liên quan đến tên miền. Mỗi tên miền đều kết thúc bằng một tên miền cấp cao (TLD) như .com hoặc .tv.

- Để Internet có thể hoạt động và để tránh trùng lặp tên miền, phải có một nơi có thẩm quyền để đăng ký một tên miền.

**Chức năng của DNS**

- Các tài nguyên mạng (những bài viết, phần mềm, ứng dụng, những bài hát, bộ phim…ở trên mạng) được định danh bằng những địa chỉ IP. Mỗi máy tính khi kết nối vào mạng Internet cũng được gán cho 1 địa chỉ IP riêng biệt không trùng lẫn. Để phân biệt những địa chỉ IP mà không phải nhớ những con số khô khan, DNS sẽ có giúp chuyển đổi địa chỉ IP thành những kí tự dễ hiểu hơn.

*Ví dụ*: Khi gõ “www.google.com” vào trình duyệt, máy chủ DNS sẽ lấy địa chỉ IP của máy chủ Google là “74.125.236.37”. Sau đó, bạn sẽ nhìn thấy trang chủ Google tải trang trên trình duyệt. Đó là quá trình phân giải DNS.

- Nhờ DNS, bạn có thể sử dụng nhiều dịch vụ mạng như tìm kiếm thông tin, xem phim, chơi game, đăng nhập các website,… bằng một tên dễ dàng ghi nhớ. Có thể nói nếu không có DNS, con người khó tiếp cận với các trang web nhanh chóng và dễ dàng như ngày nay.

### Cách sử dụng

Các máy DNS có tốc độ biên dịch khác nhau, do đó người sử dụng có thể tự chọn DNS server để sử dụng. Người dùng có thể sử dụng DNS mặc định của nhà cung cấp dịch vụ Internet hoặc sử dụng DNS server khác

## Giới thiệu về DNS server

DNS Server là một loại máy chủ dùng để quản lý và xử lý các tên miền thực hiện các bản ghi liên quan của chúng. DNS Server là thành phần chính thực hiện giao thức DNS và cung cấp các dịch vụ phân giải tên miền cho máy chủ và máy khách web trên mạng dựa trên IP. DNS Server dùng để định vị và phân phối các trang web cho người dùng cuối qua Internet, máy chủ DNS được phát triển trên phần cứng thông thường nhưng chạy phần mềm DNS chuyên dụng. Nó luôn được kết nối với Internet.

### Chức năng của DNS Server

DNS Server dùng để chứa cơ sở dữ liệu bao gồm các địa chỉ IP ứng với tên miền nhất định được gọi là bản ghi DNS. Nó thực hiện chức năng cơ bản nhất là phân giải tên miền thành địa chỉ IP tương ứng. Trong quá trình phân giải tên miền thì kết quả tìm kiếm nếu có trong các bản ghi DNS thì tên miền được trả về ứng với địa chỉ IP của nó. Nếu tên miền không được đăng ký hoặc thêm vào máy chủ DNS đó, truy vấn sẽ được chuyển đến các máy chủ DNS khác cho đến khi tìm thấy bản ghi tên miền.

Phần mềm DNS Server nổi tiếng nhất là BIND hoàn toàn miễn phí và được phân phối với các hệ thống Linux. Đối với các hệ thống sử dụng Microsoft thì Microsoft DNS là một phần mềm phổ biến của Windows Server.

### Các loại DNS Server

Tất cả các máy chủ DNS thuộc một trong 4 loại DNS Server sau:

* Recursive resolver
* Root nameserver
* TLD nameserver
* Authoritative nameserver

Khi chúng ta thực hiện tra cứu DNS thì các máy chủ này sẽ phối hợp với nhau để hoàn thành nhiệm vụ cung cấp IP cho tên miền mà chúng ta tìm kiếm.

### Giới thiệu về recursive resolver

* Một recursive resolver là điểm dừng đầu tiên trong quá trình truy vấn DNS. Nó hoạt động như một cầu nối trung gian giữa máy client và DNS nameserver.
* Sau khi nhận được truy vấn DNS từ một client web thì recursive resolver sẽ phản hồi với dữ liệu được lưu trong bộ nhớ cache hoặc gửi yêu cầu đến root nameservers tiếp theo là đến TLD nameservers và cuối cùng đến authoritative nameserver.
* Sau khi nhận được phản hồi từ authoritative nameserver chứa địa chỉ IP được yêu cầu, recursive resolver sẽ gửi phản hồi cho Client.
* Trong quá trình này, recursive resolver sẽ lưu trữ thông tin nhận được từ authoritative nameserver. Khi một Client yêu cầu địa chỉ IP của một tên miền giống với client trước yêu cầu thì recursive resolver sẽ cung cấp các bản ghi được yêu cầu từ bộ nhớ cache của nó. Bỏ qua quá trình hỏi root nameservers.

### Giới thiệu về Root nameservers

* Các Root nameservers DNS sẽ chứa các bản ghi gồm toàn bộ các thông tin về tên miền cùng với địa chỉ IP ứng với tên miền đó. Đây cũng chính là điểm dừng đầu tiên trong quá trình phân giải DNS.
* Root nameservers chấp nhận truy vấn của recursive resolver gồm một tên miền và thực hiện trả lời bằng cách hướng recursive resolver đến TLD nameservers dựa trên phần mở rộng của tên miền đó (.com, .vn, .net, .org,...). Các Root nameservers này được giảm sác và theo dõi bỡi Internet Corporation for Assigned Names and Numbers (ICANN).

### Giới thiệu về TLD nameservers

TLD nameservers dùng để duy trì thông tin cho tất cả các tên miền có chung một phần mở rộng như chúng ta đã nói bên trên (.com, .vn, .net, .org,...).

*Ví dụ:*Trên trình duyệt web chúng ta thực hiện gõ vào thanh tìm kiếm google.com khi nhận được phản hồi từ Root nameservers thì recursive resolver sẽ gửi truy vấn đến TLD nameservers. Tại đây chúng ta nhận được câu trả lời chỉ cách chúng ta đến máy chủ tên có thẩm quyền cho tên miền có phần mở rộng là .com .

Việc quản lý các máy TLD nameservers bởi Internet Assigned Numbers Authority (IANA), một chi nhánh của ICANN. IANA chia các máy chủ TLD thành hai nhóm chính:

* Các tên miền cấp cao chung: Đây là các tên miền không cụ thể theo quốc gia, một số TLD chung được biết đến nhiều nhất bao gồm .com, .org, .net, .edu và .gov.
* Tên miền cấp cao nhất của mã quốc gia: Chúng bao gồm bất kỳ tên miền nào dành riêng cho một quốc gia hoặc tiểu bang. Ví dụ bao gồm .vn, .uk, .us, .ru và .jp.

### Giới thiệu về authoritative nameserver

* Khi recursive resolver nhận được phản hồi từ TLD nameservers, phản hồi đó sẽ hướng recursive resolver đến authoritative nameserver. Đây chính là bước cuối cùng mà recursive resolver sẽ tìm được tìm địa chỉ IP cần tìm kiếm.
* Authoritative nameserves là nơi chứa thông tin cụ thể của các tên miền như là: google.com và nó có thể cung cấp cho recursive resolver địa chỉ IP của máy chủ đó được tìm thấy trong bản ghi DNS. Nếu như tên miền có bản ghi CNAME (alias) nó sẽ cung cấp cho recursive resolver với một miền alias mà tại đây recursive resolver sẽ phải thực hiện tra cứu DNS hoàn toàn mới để tạo một bản ghi từ authoritative nameserver khác.

## Web server

### Khái niệm

**Web server** là máy tính lớn được kết nối với tập hợp mạng máy tính mở rộng. Máy chủ chứa toàn bộ dữ liệu mà nó được giao quyền quản lý. Mỗi máy chủ có một IP riêng và có thể đọc đa dạng ngôn ngữ như HTML, HTM, File,… Máy chủ có dung lượng lớn và tốc độ rất cao để có thể lưu trữ và vận hành tốt kho dữ liệu trên internet. Thông qua cổng giao tiếp riêng biệt của mỗi máy chủ mà hệ thống máy tính có khả năng hoạt động trơn tru hơn. Máy chủ phải đảm bảo hoạt động liên tục để có thể cung cấp dữ liệu cho mạng lưới máy tính của nó.

[Web server](https://en.wikipedia.org/wiki/Web_server) có thể là phần cứng hoặc phần mềm cũng có thể bao gồm cả hai.

· ***Phần cứng***: Máy chủ web là một máy tính lưu trữ các file ảnh, tài liệu HTML, CSS, file JavaScript của một website và chuyển chúng tới thiết bị của End-user. Máy chủ được kết nối internet và truy cập thông qua một tên miền như Mozilla.org.

· ***Phần mềm***: Web server gồm một số phần điều khiển người dùng truy cập đến file lưu trữ trên một máy chủ HTTP. Máy chủ HTTP là một phần mềm, nó có khả năng hiểu được các địa chỉ website (URL) và giao thức trình duyệt sử dụng để xem các website (HTTP).

Bất cứ khi nào một trình duyệt cần đến file được lưu trữ trên máy chủ, trình duyệt gửi yêu cầu file đó thông qua HTTP. Khi yêu cầu tới đúng máy chủ (phần cứng), HTTP (phần mềm) sẽ gửi tài liệu được yêu cầu trở lại thông qua HTTP

**Các đặt tính của Web Server.**

· **Web server** (máy chủ web) có thể xử lý dữ liệu, cung cấp thông tin đến máy khách qua môi trường internet thông qua giao thức HTTP, giao thức được thiết kế gửi file đến trình duyệt web hay giao thức khác.

Ví dụ: Người dùng truy cập vào website [**bkns.vn**](https://www.bkns.vn/). Khi đó, server sẽ cung cấp tất cả dữ liệu về website đó thông qua lệnh giao tiếp.

· Nếu được cài đặt một chương trình Server Software và kết nối internet thì bất cứ máy tính này cũng có thể trở thành web server. Server Software là phần mềm chuyên dụng để cài đặt và chạy trên bất cứ máy tính nào có thể đáp ứng đủ yêu cầu về bộ nhớ. Nhờ có nó mà người dùng có thể truy cập đến các thông tin của website từ một máy tính khác qua internet.

· Người ta thường thuê các máy chủ nhỏ, máy chủ ảo VPS hay Hosting để lưu trữ dữ liệu cho website của mình.

· Một server có thể cung cấp cả nội dung Static và Dynamic. Static có nghĩa là nội dung nguyên vẹn và dễ dàng để thiết lập. Dynamic là nội dung đã được sever xử lý hoặc tạo mới với dữ liệu từ Database, định dạng, đẩy vàos trong HTTP Template rồi gửi kết quả đến người dùng.

### Các loại Web Server phổ biến

- **Web Server Apache**

**Web server Apache** được phát triển bởi Quỹ Phần mềm Apache và là một trong những web server nổi tiếng trên thế giới. Đây là phần mềm mã nguồn mở, hỗ trợ hầu hết các hệ điều hành như Unix, Linux, Windows, Mac OS X, FreeBSD,… Theo thống kê, khoảng 60% máy tính chạy trên web server Apache.

Web server Apache có các tùy biến được thực hiện dễ dàng bởi nó có cấu trúc dạng Module. Bạn có thể thêm hay sửa đổi các Module vào server theo ý muốn nếu cảm thấy phù hợp. So với bất cứ máy chủ web nào thì Apache cũng ổn định và dễ dàng xử lý khi có vấn đề xảy ra. Các phiên bản mới của web server Apache có khả năng xử lý được nhiều yêu cầu hơn so với phiên bản tiền nhiệm.

* + **Web server IIS**

**Web server IIS** là sản phẩm của Microsoft, nó có rất nhiều tính năng giống như Apache. Tuy nhiên, đây không phải là mã nguồn mở và việc thêm, chỉnh sửa các Module theo ý muốn không hề dễ dàng. Web server IIS có khả năng chạy trên tất cả nền tảng của hệ điều hành của Windows.

* + **Web server Nginx**

**Web server Nginx** là một máy chủ mã nguồn mở miễn phí. Nginx bao gồm máy chủ POP3 và IMAP. Web server Nginx có ưu điểm là ổn định, hiệu suất cao, cấu hình đơn giản và sử dụng tài nguyên thấp. Nginx không dùng các chuỗi (thread) để xử lý các yêu cầu mà sử dụng kiến trúc lập trình theo sự kiện (có khả năng mở rộng). Kiến trúc lập trình này dùng bộ nhớ khi tải nhỏ và dự đoán được. Nginx hiện đang lưu trữ khoảng 7.5 % các tên miền trên toàn thế giới. Trong những năm trở lại đây, phần đông công ty web hosting sử dụng Nginx.

* + **Web server LiteSpeed**

**Web server LiteSpeed** có rất nhiều tính năng giống như Apache. LiteSpeed có khả năng tải các file cấu hình Apache trực tiếp đồng thời hoạt động như một Drop-in Replacement Apache với các Control Panel của Hosting. LiteSpeed có thể được thay thế với web server Apache trong khoảng 15 phút với Downtime bằng 0. Web server LiteSpeed còn có khả năng thay thế tất cả tính năng của **Apache** và đơn giản hóa việc sử dụng.

### Một số lưu ý khi sử dụng Web Server

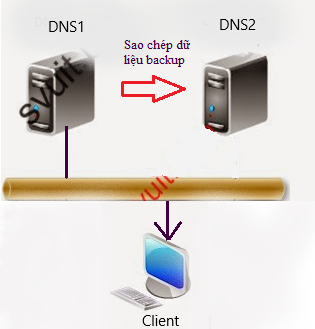
**Thứ nhất**, web server giúp người dùng có thể tìm kiếm thông tin liên quan đến website của mình. Do đó, bạn cần có một máy tính cấu hình cao, lưu trữ được dung lượng lớn dữ liệu và đáp ứng được số lượng lớn người dùng truy cập.

**Thứ hai**, web server cần đảm bảo hoạt động liên tục 24/24 để cung cấp thông tin trực tuyến cho người dùng. Việc lựa chọn máy chủ web đóng vai trò quan trọng trong trong việc lưu chuyển thông tin từ server đến máy tính truy cập. Dịch vụ cho thuê web server ngày càng mở rộng và phát triển không ngừng, nó cho phép tạo ra nhiều gói dịch vụ để doanh nghiệp có thể đưa ra lựa chọn tối ưu nhất. Giữa trăm ngàn lựa chọn nhà cung cấp dịch vụ máy chủ web, bạn cần hiểu biết và tỉnh táo để lựa chọn được địa chỉ uy tín cung cấp sản phẩm chất lượng.

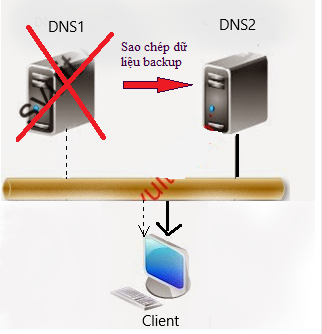
# Mô hình triển khai

## 1. Triển khai mô hình DNS Server

DNS là hệ thống rất quan trọng trong hệ thống mạng, vì vậy, để tránh quá tải cho server DNS chính hoặc DNS server có thể chết, chúng ta cần phải thiệt lập một hệ thống các secondary để chia tải và backup cho Primary. Thay vì các client phải truy vấn trực tiếp vào DNS server chính thì có thể sử dụng DNS server backup



Chúng ta sẽ triển khai mô hình DNS Server dựa trên mô hình Master – Slave với 2 Server và 1 Client bao gồm một máy DNS Server Master (DNS1: Máy chính), một máy DNS Server Slave (DNS2:máy phụ nhận dữ liệu backup được sao chép từ máy DNS1) và một máy Client sẽ truy vấn đến máy chủ để thực hiện phân giải tên



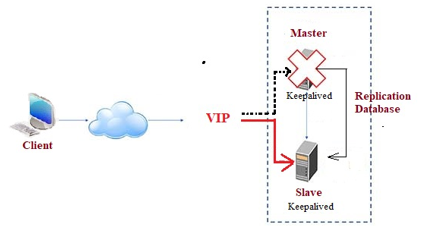
Khi máy DNS1 xảy ra một sự kiện làm thay đổi dữ liệu (thêm, sửa, xóa) thì máy DNS1 sẽ log ra một file, máy DNS2 thì liên tục lắng nghe file log này, và thực hiện việc thay đổi dữ liệu trên máy DNS1 như thay đổi trên DNS1. Khi máy DNS Server Master gặp sự cố, bị hỏng (bị tắt), máy DNS Server Slave vẫn có thể thay máy chính thực hiện dự phòng

## Triển khai mô hình Web Server

Chúng ta sẽ triển khai mô hình gồm 2 máy được cài các thuộc tính của Web Server bao gồm Apache, MySQL và PHP (bao gồm một máy chính Master và một máy phụ Slave hoạt động dựa trên dịch vụ Keepalived) và một máy Client, máy Master sẽ nhân bản cơ sở dữ liệu (Replication Database) và backup mã nguồn (source code) qua máy Slave

Keepalived sẽ khởi tạo một Virtual Server đại diện cho một nhóm thiết bị đó với một Virtual IP (VIP) và một địa chỉ MAC vật lý của máy chủ dịch vụ đang giữ Virtual IP đó. Vào mỗi thời điểm nhất định, chỉ có một server dịch vụ dùng địa chỉ MAC này tương ứng Virtual IP. Khi có ARP request gởi tới virtual IP thì server dịch vụ đó sẽ trả về địa chỉ MAC này.

Các máy chủ dịch vụ sử dụng chung VIP phải liên lạc với nhau bằng địa chỉ *multicast 224.0.0.18* bằng giao thức VRRP. Các máy chủ sẽ có độ ưu tiên (priority) trong khoảng từ 1 – 254, và máy chủ nào có độ ưu tiên cao nhất sẽ thành Master, máy còn lại sẽ thành các Slave/Backup, hoạt động ở chế độ chờ. Cơ chế failover (Giải pháp chịu lỗi) được xử lý bởi giao thức VRRP, khi khởi động dịch vụ, toàn bộ các server cấu hình dùng chung VIP sẽ gia nhập vào một nhóm multicast. Nhóm multicast này dùng để gởi/nhận các gói tin quảng bá VRRP. Các server sẽ quảng bá độ ưu tiên của mình, server với độ ưu tiên cao nhất sẽ được chọn làm Master. Một khi nhóm đã có 1 Master thì Master này sẽ chịu trách nhiệm gởi các gói tin quảng bá VRRP định kỳ cho nhóm multicast.



Nếu vì một sự cố gì đó mà Server backup không nhận được các gói tin quảng bá từ Master trong một khoảng thời gian nhất định (máy Master bị hỏng) thì Server backup lúc này sẽ thành một Master mới. Máy Master mới này sẽ đảm nhiệm và thực hiện tiếp nhiệm vụ. Khi Master cũ hoạt động bình thường trở lại thì server này có thể lại trở thành Master hoặc trở thành Backup tùy theo cấu hình độ ưu tiên của các router.

# Cài Đặt, cấu hình

## Bước 1: Đặt Ip và DNS cho các máy

Dựa vào thông tin cấu hình trên VMs, ta đặt cấu hình cho các máy

Vào *Edit Virtual Network Editor…*

Chọn *VMnet8 / NAT Settings…*



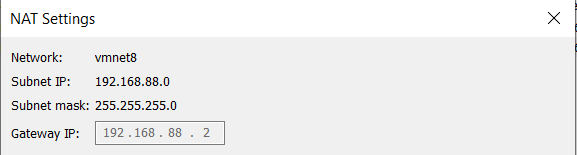
Dưới đây ta thấy các thông số network:

Subnet IP: 192.168.88.0

Subnet mask: 255.255.255.0

Gateway IP: 192.168.88.2

Vậy thông tin cấu hình cần dựa trên các thông số này



Thông tin IP trên 3 máy:

|  |
| --- |
| * **Máy DNS Server Master (DNS1: máy chính)**   Operating system: ubuntu 16.04.6 LTS 64 bit server  Hostname: dns1.nhom5.local  IP Address: 192.168.88.101  Network: 192.168.88.0  Gateway: 192.168.88.2  DNS: 192.168.88.101  DNS: 192.168.88.102  DNS: 8.8.8.8   * **Máy DNS Server Slave (DNS2: máy phụ)**   Operating system: ubuntu 16.04.6 LTS 64 bit server  Hostname: dns2.nhom5.local  IP Address: 192.168.88.102  Network: 192.168.88.0  Gateway: 192.168.88.2  DNS: 192.168.88.101  DNS: 192.168.88.102  DNS: 8.8.8.8   * **Máy Client**   Operating system: ubuntu 16.04.6 64 bit desktop  Hostname: client.nhom5  IP Address: 192.168.88.103  Network: 192.168.88.0  Gateway: 192.168.88.2  DNS: 192.168.88.101 |

Trong đó ta thấy có DNS 8.8.8.8 là DNS Google

Vậy, tại sao phải dùng DNS Google?

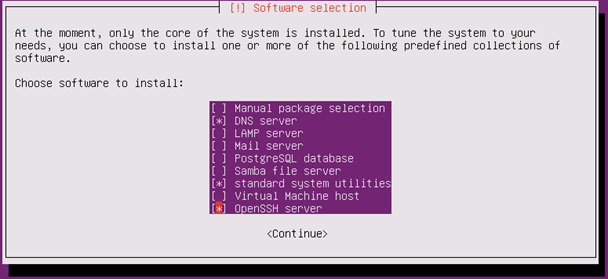
## Bước 2: Cài đặt máy ảo

Ta cần lưu ý ở bước này là ở phần *choose software to install:*

Mặc định ban đầu là có *standard system utilities*

Ta chọn DNS server và OpenSSH

Trong đó OpenSSH là chương trình được sử dụng để đăng nhập từ xa, ở đây dùng SSH để dễ dàng Copy/Paste hơn do giao diện của Server khó Copy/Paste hơn.



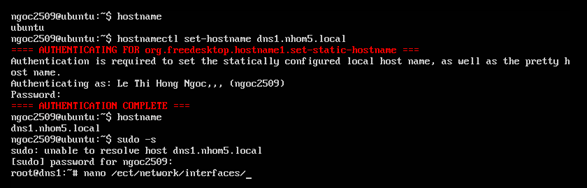
## Bước 3: Đặt hostname, ip và dns

### a. Trên máy DNS Server Master (DNS1)

Để xem hostname dùng lệnh *hostname*

Để đặt lại hostname trên máy DNS1 ta dùng câu lệnh *hostnamectl set-hostname dns1.nhom5.local*

Có thể dùng lệnh *sudo -s* để vào root, không cần gõ lại sudo trước mỗi lệnh

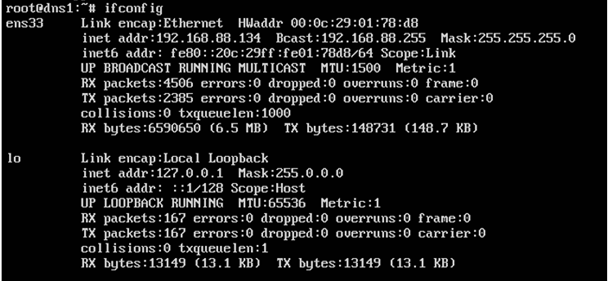


Tiếp theo cấu hình IP và DNS cho máy DNS1

Khi chưa cấu hình, xem thông tin interfaces bằng dòng lệnh *ifconfig*

Trong đó **interface lo**: interface local của máy

**Ens33** là card mạng của máy

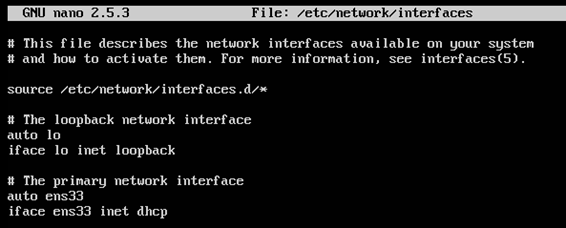


Ta sẽ tiến hành cấu hình trên interface **ens33**

Ta vào file **/ect/network/interfaces** để chỉnh sửa lại thông tin cấu hình bằng dòng lệnh *sudo nano /ect/network/interfaces.* Nhập pasword để vào file



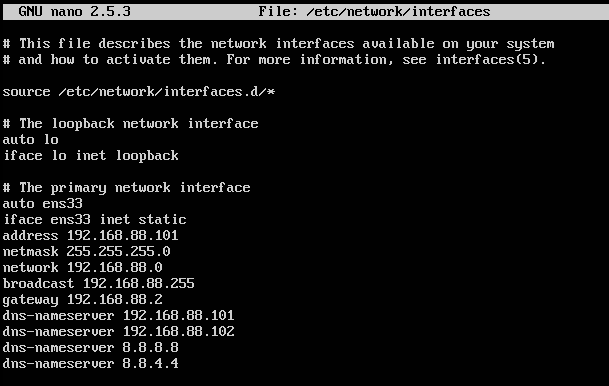
File sẽ đực hiển thị như hình dưới, trên interface ens33 đặt được inet là dhcp



Trên interface ens33 ta xóa dhcp và đặt lại thành static, sau đó cấu hình lại các thông tin IP như bên dưới:

|  |
| --- |
| Address 192.168.88.101  Netmask 255.255.255.0  Network 192.168.88.0  Broadcast 192.168.88.255  Gateway 192.168.88.2  Dns-nameserver 192.168.88.101  Dns-nameserver 192.168.88.102  Dns-nameserver 8.8.8.8  Dns-nameserver 8.8.4.4 #có thể có hoặc không |

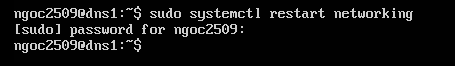
Nguyên nhân cần đặt dns-nameserver 8.8.8.8 vì nếu chỉ cài đặt dns-nameserver 192.168.88.101 hay 192.168.88.102 thì chúng ta sẽ không thể cài đặt được bind vì nó không truy cập internet được, nên nó không phân giải được tên miền nên không thể cài đặt gói đó được.



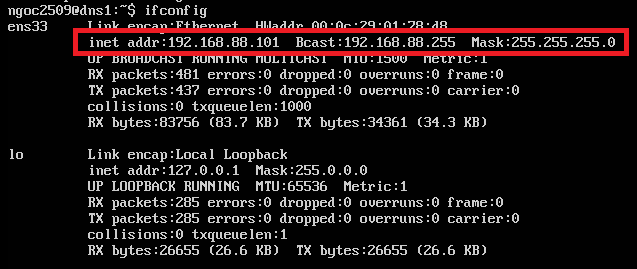
Lưu file và **reboot** lại chương trình



Ta restart network service để khởi động lại dịch vụ mạng

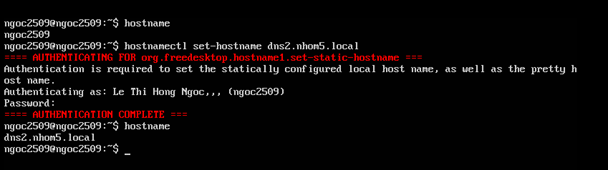


Xem lại thông tin của interface sau khi cấu hình hoàn tất



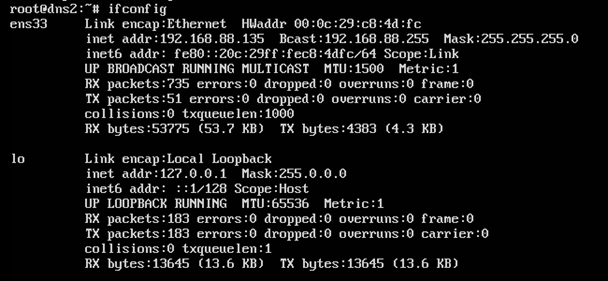
### b. Cấu hình trên máy DNS Server Slave (DNS2)

Đặt lại hostname trên máy DNS2 là dns2.nhom5.local



Cấu hình IP và DNS cho máy DNS2

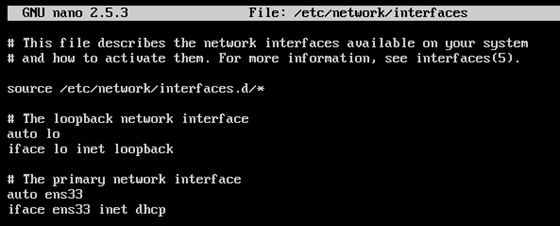
Xem cấu hình của máy DNS2 bằng dòng lệnh *ifconfig*



Ta vào file **/ect/network/interfaces** để thay đổi cấu hình cho interface

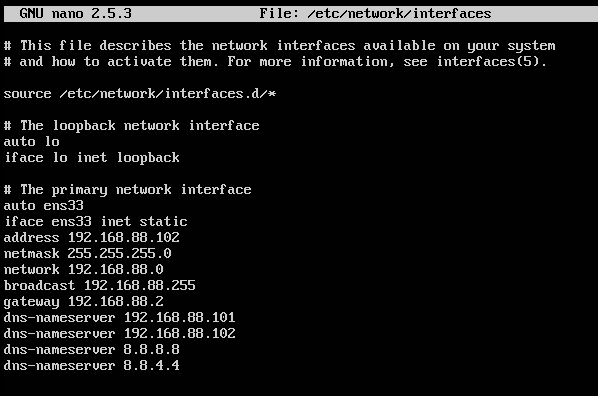


Thông tin file **/ect/network/interfaces**  ban đầu



Ta cũng làm tương tự như máy DNS1, xóa dòng dhcp là viết lại như hình dưới:

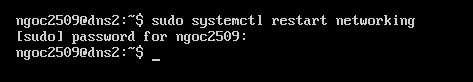
|  |
| --- |
| Address 192.168.88.102  Netmask 255.255.255.0  Network 192.168.88.0  Broadcast 192.168.88.255  Gateway 192.168.88.2  Dns-nameserver 192.168.88.101  Dns-nameserver 192.168.88.102  Dns-nameserver 8.8.8.8  Dns-nameserver 8.8.4.4 #có thể có hoặc không |



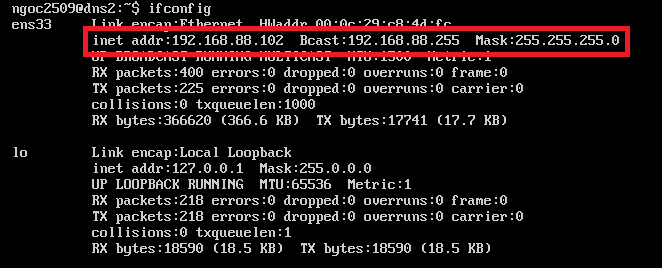
Lưu lại file và reboot lại chương trình



Sau đó restart network service

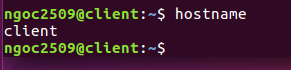


Thông in trên interface khi đã cấu hình thành công



### c. Trên máy Client

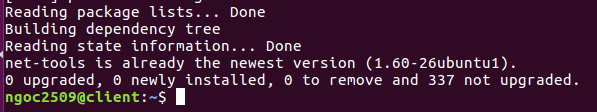
Do lúc cài máy đã đặt sẵn hostname cho client nên sẽ giữa nguyên hostname client



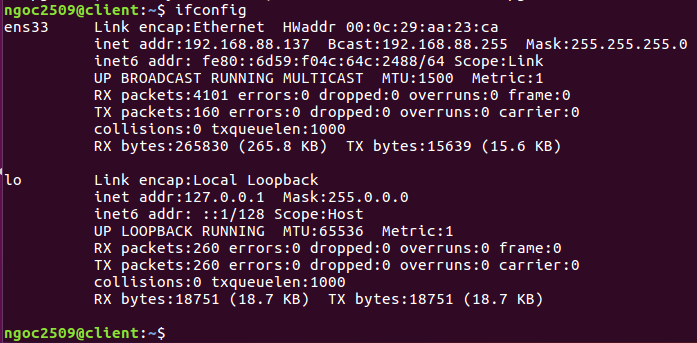
Nhập dòng lệnh sudo apt install net-tools để cài đặt net-tools và nhập mật khẩu



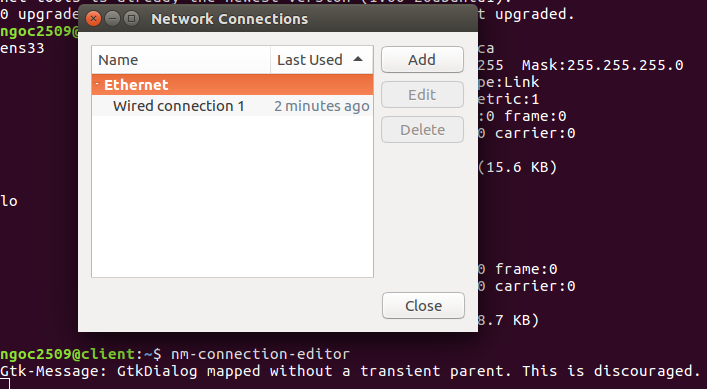
Net-tools cài đặt hoàn tất



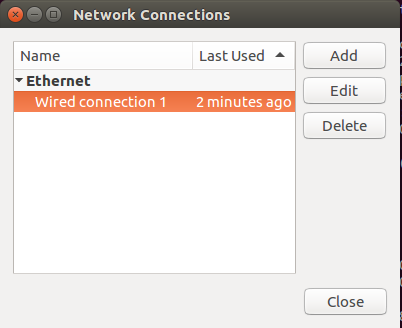
Để xem cấu cấu hình trên máy tính hiện tại, ta gõ *ifconfig*



Tiếp theo chúng ta gõ lệnh *nm-connection-editor*để vào phần truy cập chỉnh sửa đặt địa chỉ IP tĩnh trên Ubuntu, sau đó hộp thoại Network Connections hiện lên.

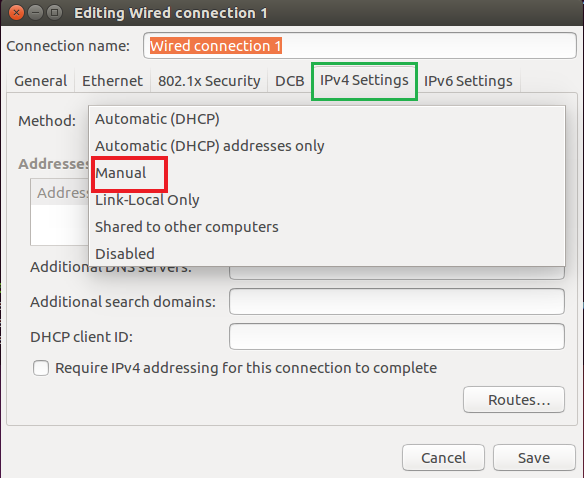


Chọn mạng **Wired connection 1** và nhấn Edit



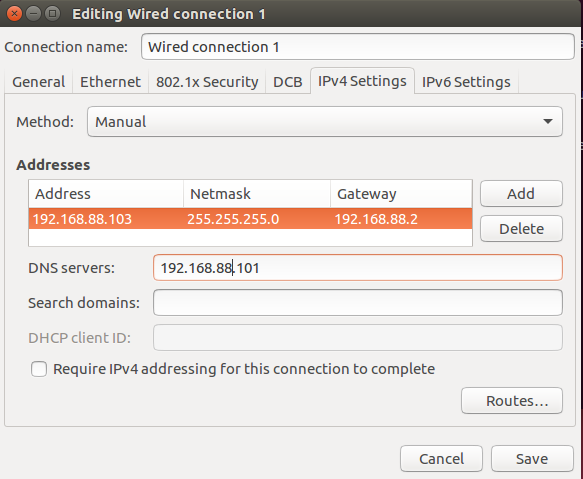
Chọn hộp thoại **IPv4 Settings**

Phần **Method** sửa từ DHCP sang chế độ Manual



Bước tiếp theo, ta nhấn Add để thêm địa chỉ IP, netmask, gateway và DNS như hình dưới

Sau đó nhấn Save để lưu cấu hình



Cuối cùng gõ lệnh **reboot** để khởi động lại với IP đã cài đặt



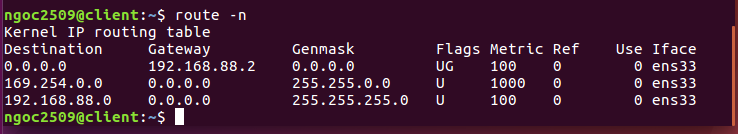
Ta restart network service



Ta kiểm tra lại cấu hình IP bằng lệnh *ifconfig.*



Hoặc có thể dùng lệnh route –n để xem các thông số máy



Vậy ta đã hoàn thành cho việc cấu hình Hostname, IP và DNS cho các máy

## Bước 4: Tắt Ipv6

Để tránh xảy ra các xung đột với địa chỉ IPv4, ta nên tắt IPv6

### Trên máy DNS1

Để xem IPv6 có bật chưa ta dùng lệnh *cat /proc/sys/net/ipv6/conf/all/disable\_ipv6*

Nếu là 0: đang bật IPv6. Nếu là 1: Đang tắt IPv6

Nếu đang bật, thì nên tắt đi

Để tắt IPv6 đi bằng cách nhập lần lượt các dòng lệnh sau:

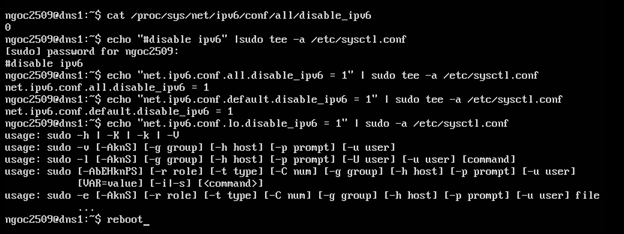
*echo “#disable.ipv6” | sudo tee -a /etc/sysctl.conf*

*echo “net.ipv6.conf.all.disable\_ipv6 = 1” | sudo tee -a /etc/sysctl.conf*

*echo “net.ipv6.conf.default.disable\_ipv6 = 1” | sudo tee -a /etc/sysctl.conf*

*echo “net.ipv6.conf.lo.disable\_ipv6 = 1” | sudo tee -a /etc/sysctl.conf*

Cuối cùng *reboot* lại máy



Xem lại kết quả bằng dòng lệnh *cat /proc/sys/net/ipv6/conf/all/disable\_ipv6*

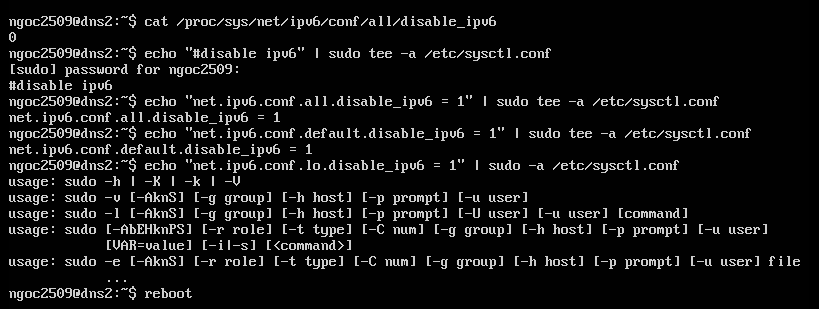
Ta thấy kết quả trả về là 1, tức là IPv6 đã tắt



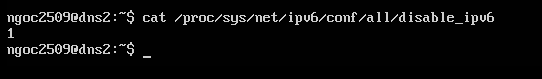
### Trên máy DNS2

Ta làm tương tự từng bước như trên

* Kiểm tra IPv6 đã tắt chưa
* Nếu chưa thì tắt đi
* Reboot lại máy



Kiểm tra lại IPv6, ta thấy với kết quả trả về là 1 tức là Ipv6 trên máy DNS2 đã được tắt



## Bước 5: Kết nối ssh từ máy client đến máy server

Với việc login này giúp ta dễ dàng thực hiện các thao tác hơn là thực hiện trực tiếp trên máy server

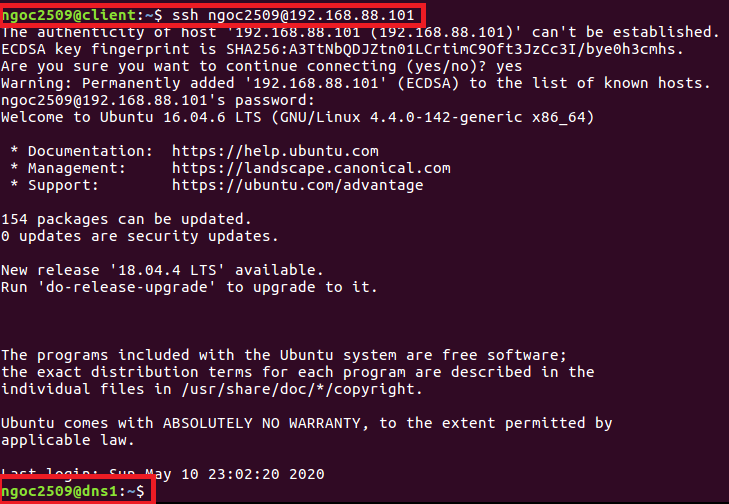
Lưu ý: phải mở cả máy DNS1 và DNS2 cùng hoạt động khi kết nối bằng SSH

### Kết nối đến máy DNS1

Kết nối đến máy DNS1 ta dùng lệnh **ssh <tên user>@<địa chỉ IP của máy đích>**

Ở đây dùng lệnh *ssh* [*ngoc2509@192.168.88.101*](mailto:ngoc2509@192.168.88.101) với IP 192.168.88.101 là IP của máy DNS1

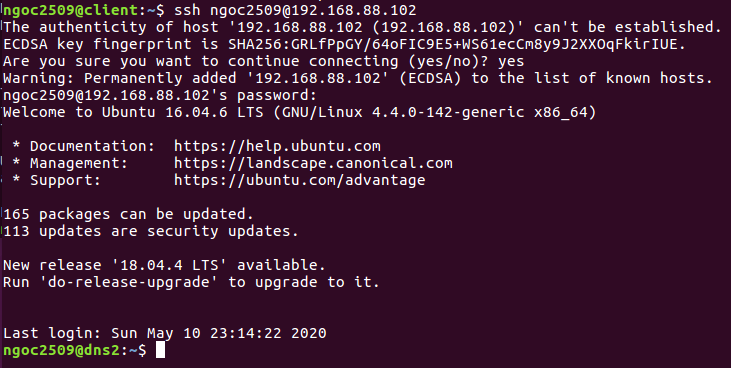
Khi đã kết nối thành công (như hình dưới) ta thấy **ngoc2509@dns1** nghĩa là nó login vào user ngoc2509 tại máy dns1



### Kết nối với máy DNS2

Tương tự với kết nối đến máy DNS2 dùng lệnh *ssh* [*ngoc2509@192.168.88.102*](mailto:ngoc2509@192.168.88.102)với IP 192.168.88.102 là IP của máy DNS1

Khi đã kết nối thành công (như hình dưới) ta thấy **ngoc2509@dns2** nghĩa là nó login vào user ngoc2509 tại máy dns2



Ta nhận thấy các kết nối đều thành công, cũng đồng nghĩa với việc ta đặt các IP và DNS cho các máy là hợp lệ.

## Bước 6: Cài đặt Bind

*Trước tiên ta tìm hiểu gói Bind9 là gì?*

Bind9 là dịch vụ cache server sẽ ghi nhớ những câu trả lời mỗi khi DNS Server xuất hiện truy vấn hostname và trả lời vào những lần sau. Dịch vụ này đặc biệt hữu ích trong những môi trường có kết nối Internet chậm. Bằng cách sử dụng Cache Server, bạn sẽ giảm thiểu được lưu lượng bang thông và độ trễ mỗi khi máy xuất hiện truy vấn.

Khi cài đặt gói Bind9, cấu hình mặc định của dịch vụ sẽ cho phép bạn chạy Cache Server. Tất cả những gì bạn cần làm là thêm vào địa chỉ IP của DNS Server của ISP ( Internet Service Provider – nhà cung cấp dịch vụ Internet ).

### Trên máy DNS1

Ở đây ta sẽ cài đặt trực tiếp trên máy DNS1

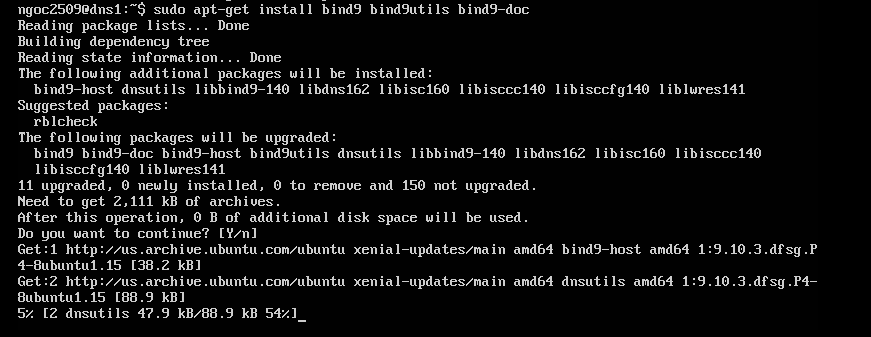
Trước tiên ta cần update lại apt-get lên mới nhất bằng dòng lệnh *sudo apt-get update* để update lên các gói package mới nhất



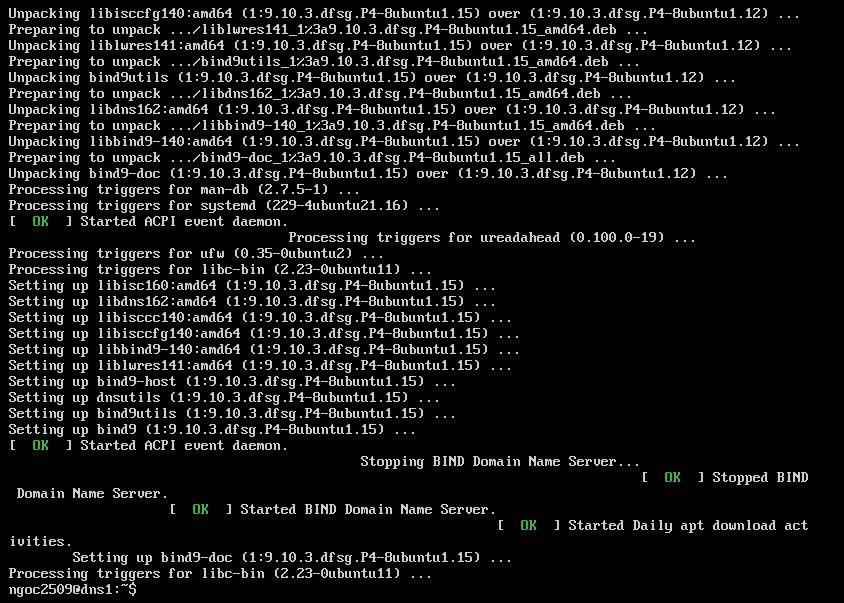
Tiếp theo ta bắt đầu cài gói bind9 bằng cách nhập dòng lệnh *sudo apt-get install bind9 bind9utils bind9-doc*

Trong đó bind9utils là các gói công cụ kèm theo như named, namedconf hay namedconf-zone utils của bind

Và bind9-doc là document của bind



Cài gói bind9 thành công



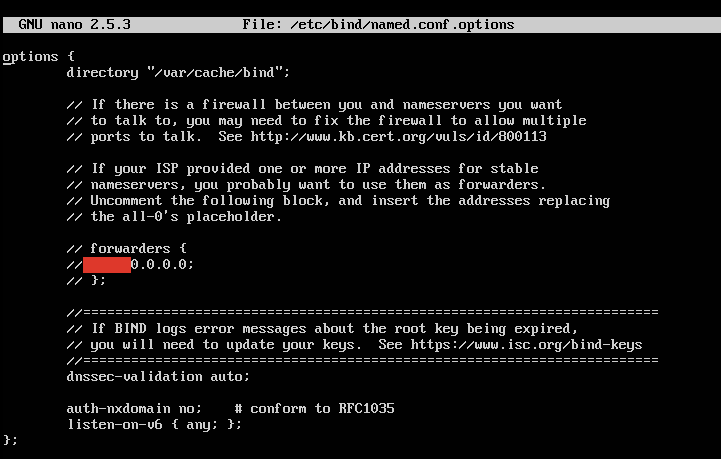
Tiếp theo, ta tiến hành cấu hình Caching DNS

Đầu tiên, edit file **/etc/bind/named.conf.options**

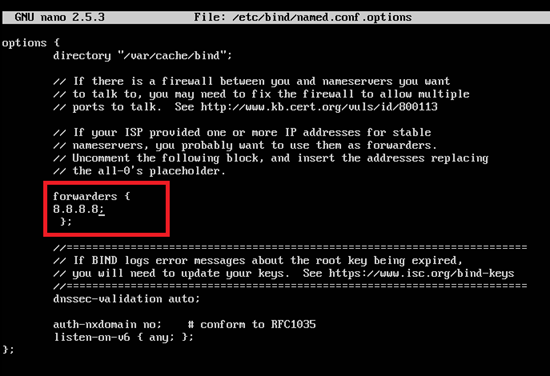
Nhập *sudo nano /etc/bind/named.conf.options* và gõ password



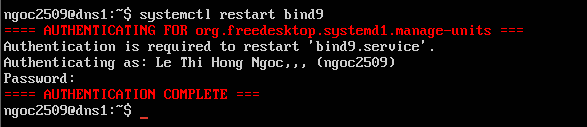
File */etc/bind/named.conf.options* được hiển thị như hình dưới



Ta loại bỏ comment ở 3 dòng forwarders và chuyển thành DNS server của Google là *8.8.8.8*



Khởi động lại dịch vụ Bind9 để thay đổi có hiệu lực: *systemctl restart bind9*



Vậy ta đã cấu hình thành công cho caching dns server

File cấu hình của DNS được đặt tại thư mục **/etc/bind**với địa chỉ chính xác là **/etc/bind/named.conf**

Tiếp theo ta nhập *nano /etc/bind/named.conf* và kiểm tra trong file đã có các dòng lệnh:

*include “/etc/bind/named.conf.options”;*

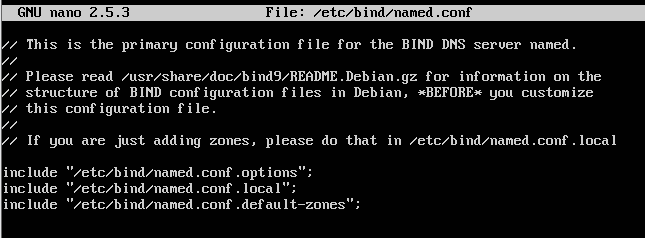
*include “/etc/bind/named.conf.local”;*

*include “/etc/bind/named.conf.default-zones”;*

Nếu chưa có thì thêm vào



Ta kiểm tra thấy đã có đủ các dòng lệnh rồi



Sau đó ta lưu và đóng file lại

Bước tiếp theo ta cần xác định **forward zone** (zone phân giải thuận, chuyển từ tên miền sang IP) và **reverse zone** (zone phân giải nghịch, chuyển từ IP sang tên miền) chỉ ra các file chứa cơ sở dữ liệu phục vụ cho việc phân giải tên

Edit file bằng dòng lệnh *sudo nano /etc/bind/named.conf.local*

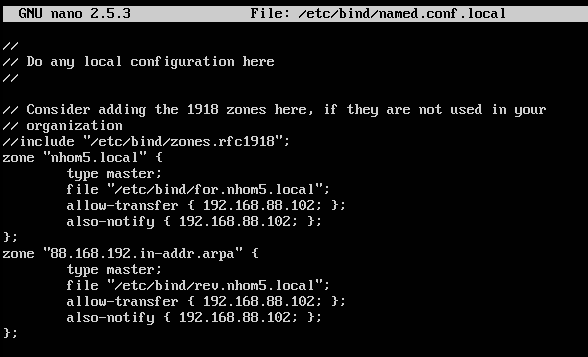


Các thông tin bên trong file



Ta tiến hành thêm các dòng sau vào file và lưu lại file.

|  |
| --- |
| *zone "nhom5.local" { # forward zone*  *type master; #do máy DNS1 nên type của nó là master*  *file "/etc/bind/for.nhom5.local"; # file forward zone*  *allow-transfer { 192.168.88.102; }; #cho phép chuyển các CSDL qua máy chủ có*  *địa chỉ là 192.168.88.102*  *also-notify { 192.168.88.102; }; #Thông báo slave cho thay đổi vùng*  *};*  *zone "88.168.192.in-addr.arpa" { # reverse zone*  *type master;*  *file "/e tc/bind/kê.local "; #file reverse zone*  *allow-transfer { 192.168.88.102; };*  *also-notify { 192.168.88.102; };*  *};* |



Tiếp theo ta tạo file “/etc/bind/for.nhom5.local” (file forward zone) bằng dòng lệnh *nano /etc/bind/for.nhom5.local*

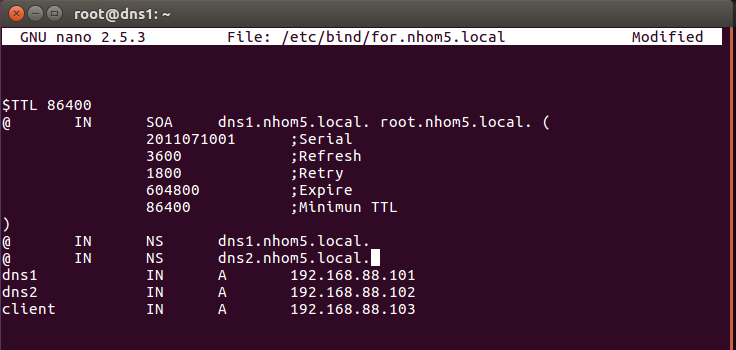


Sau đó ta thêm các dòng sau và lưu file lại

|  |
| --- |
| $TTL 86400  @   IN  SOA     dns1.nhom5.local. root.nhom5.local. (          2011071001  ;Serial          3600        ;Refresh          1800        ;Retry          604800      ;Expire          86400       ;Minimum TTL  )  @       IN  NS          dns1.nhom5.local.  @       IN  NS          dns2.nhom5.local.  dns1       IN  A   192.168.88.101  dns2    IN  A   192.168.88.102  client      IN  A   192.168.88.103 |

Trong đó:

* **SOA**: Chỉ ra rằng máy chủ Name Server là nơi cung cấp thông tin tin cậy từ dữ liệu có trong zone.
* **NS**: Mỗi name server cho zone sẽ có một NS record.
* **A**: Record A (Address) ánh xạ tên máy (hostname) vào địa chỉ IP.



Kế tiếp tạo file /etc/bind/rev.nhom5.local” (file reverse zone) bằng dòng lệnh *nano /etc/bind/rev.nhom5.local*

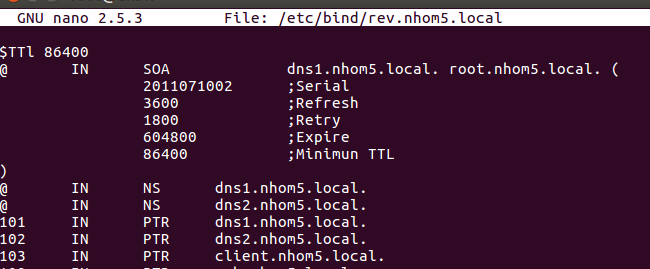


Ta thêm các dòng sau và lưu file lại:

|  |
| --- |
| **$TTL 86400**  **@   IN  SOA     dns1.nhom5.local. root.nhom5.local. (**  **2011071002  ;Serial**  **3600        ;Refresh**  **1800        ;Retry**  **604800      ;Expire**  **86400       ;Minimum TTL**  **)**  **@       IN  NS           dns1.nhom5.local.**  **@ IN NS dns2.nhom5.local.**  **101     IN  PTR         dns1.nhom5.local.**  **102     IN  PTR         dns2.nhom5.local.**  **103     IN  PTR         client.** |

Trong đó:

* **PTR**: Record PTR (pointer) dùng để ánh xạ địa chỉ IP thành hostname.



Cấu hình Permission, Ownership cho dịch vụ Bind bằng lệnh:

***chmod -R 755 /etc/bind***

Trong đó :

+ chomd để thay đổi chỉ số phân quyền

+ 755 có nghĩa là:

owner có quyền r+w+x =  4+2+1 = 7

group có quyền  r+-+x = 4+0+1 = 5

other có quyền  r+-+x = 4+0+1 = 5

+ -R là áp dụng luôn vào các file/folder nằm bên trong folder được phân quyền (chỉ áp dụng cho thư mục)

***chown -R bind:bind /etc/bind****:* file này thuộc quyền sở hữu của user bind và group bind



Kiểm tra lại các file cấu hình DNS và các file zone.

Để kiểm tra các thiết lập về DNS đã đúng chưa ta sử dụng các dòng lệnh sau:

*named-checkconf /etc/bind/named.conf*

*named-checkconf /etc/bind/named.conf.local*

Nếu không báo lỗi, các thiết lập coi như đã đúng

Kiểm tra các thiết lập về zone đã đúng chưa ta sử dụng các dòng lệnh sau:

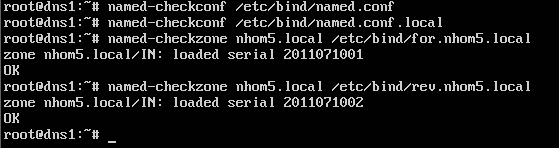
+ Kiểm tra forward zone

*named-checkzone nhom5.local /etc/bind/for.nhom5.local*

*+* Kiểm tra reverse zone

*named-checkzone nhom5.local /etc/bind/rev.nhom5.local*

Nếu có thông báo OK coi như các thiết lập đã đúng.



Sau khi kiểm tra thành công ta khởi động lại bind9 service để áp dụng thay đổi .

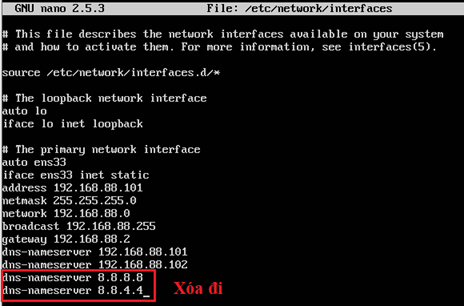
Sử dụng lệnh *systemctl restart bind9*



Sửa cấu hình lại, thêm vào file cấu hình card mạng với thông số của máy DNS1: *sudo nano /etc/network/interfaces*

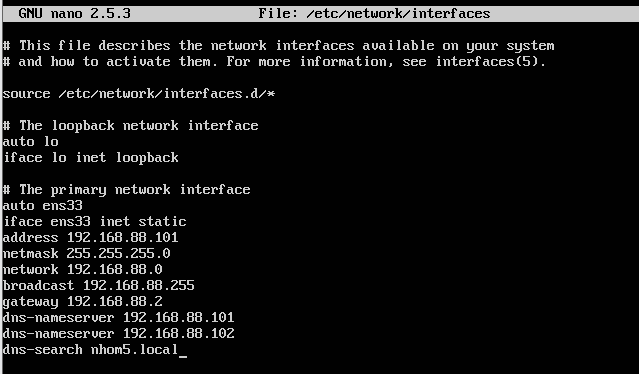


Ta xóa đi dns-name server 8.8.8.8 và 8.8.4.4, vì khi này ta đã cài đặt gói bind thành công.



Thêm địa chỉ IP của máy DNS Server Master và lưu file lại:

*dns-search nhom5.local*



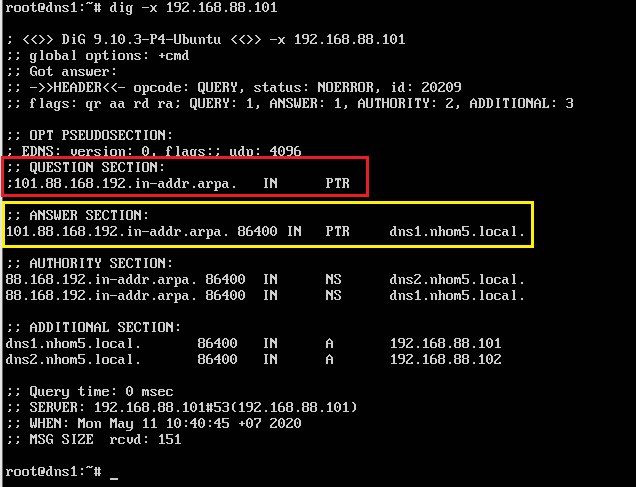
Tiến hành reboot lại máy



Kiểm tra có phân giải được tên miền hay không, dùng lệnh *dig -x 192.168.88.101* như hình dưới

Dòng **Question Selection** là câu hỏi của chúng ta đặt ra địa chỉ **192.168.88.101** có tên miền là gì

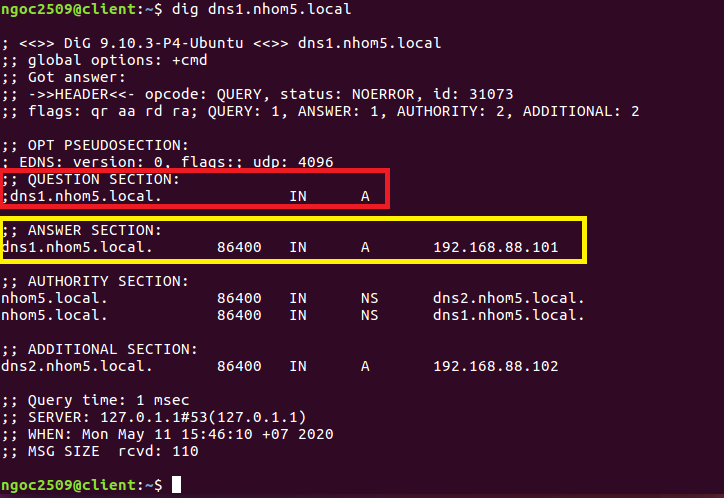
Và dòng **Answer Selection** là cẩu trẩ lời được trả về có tên miền tương ứng là **dns1.nhom5.local**



Một các khác, ta thử phân giải tên miền trên máy Client bằng dòng *lệnh dig dns1.nhom5.local* như hình dưới

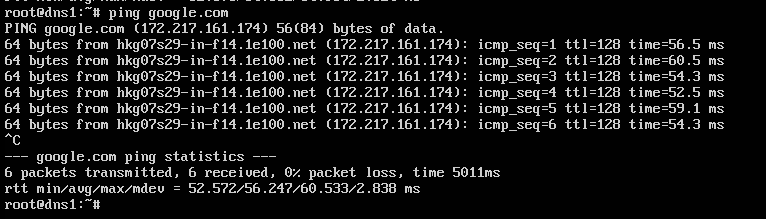
Dòng **Question Selection** là câu hỏi của chúng ta đặt ra địa chỉ **dns1.nhom5.local** có IP là gì

Và dòng **Answer Selection** là cẩu trẩ lời được trả về có địa chỉ Ip tương ứng là **192.168.88.101**



Cuối cùng thử ping đến địa chỉ của google xem có kết nối được ở mạng ngoài không bằng dòng lệnh ping google.com

Theo như hình dưới, có nghĩa là ta đã nhận được các gói tin từ google trả về



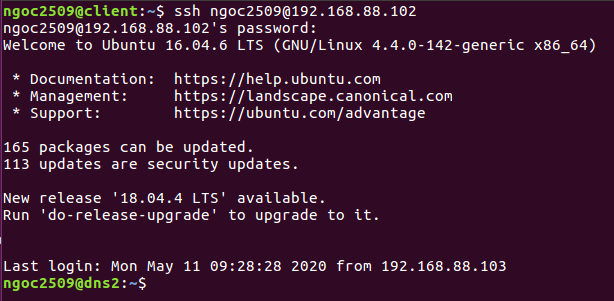
Vậy ta đã cấu hình DNS cho máy DNS Server Master thành công. Như vậy là DNS Master đã có thể hoạt động được bình thường. Chúng ta bắt đầu chuyển sang cấu hình DNS Server Slave.

### Cài trên máy DNS2

Ngoài cách cài trực tiếp trên máy DNS như bước ở trên, ở đây ta còn 1 cách là cài trên máy client bằng SSH kết nối đến máy DNS2, song song ở bước này ta vẫn để cho máy DNS2 hoạt động

Đăng nhập bằng ssh trên máy client: ssh [ngoc2509@192.168.88.102](mailto:ngoc2509@192.168.88.102),

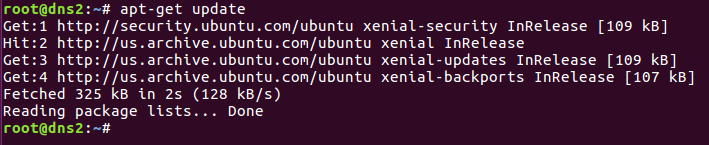
Khi đã kết nối thành công, quyền thao tác sẽ được chuyển sang user ngoc2509 trên máy dns2



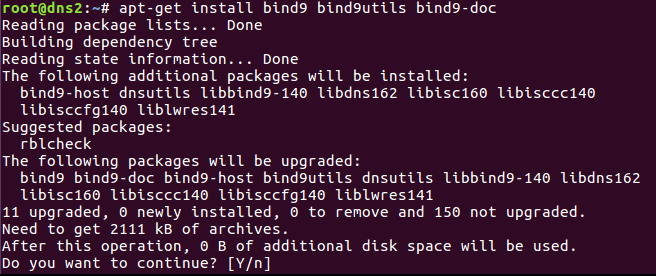
Ta có thể sử dụng lệnh sudo -s để không cần dùng lại sudo và nhập mật khẩu cho các câu lệnh sau:



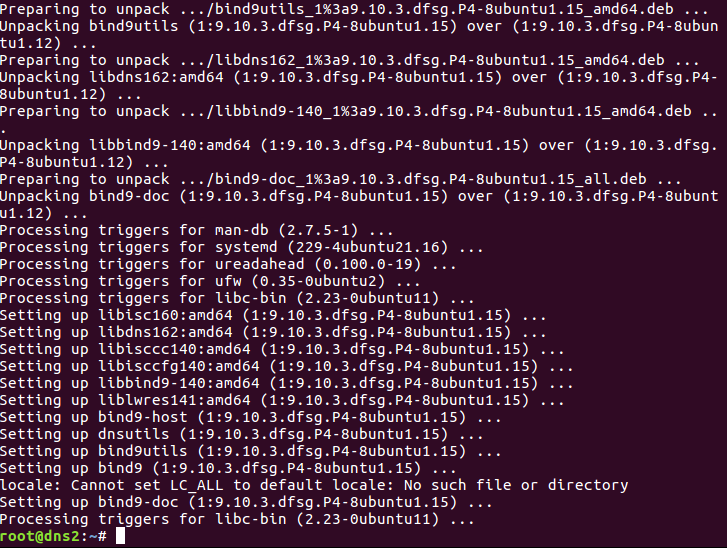
Việc cài đặt gói bind9 ở máy DNS2 cũng gần tương tự như ở máy DNS1, trước tiên ta sẽ update apt-get lên mới nhất: *apt-get update*



Tiếp theo, ta cài gói bind9 vào



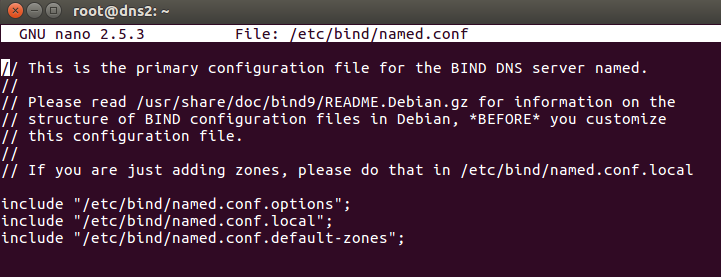
Chờ đến khi cài đặt thành công, nhưu hình dưới



Tiếp theo vào file /etc/bind/named.conf để kiểm tra một số thông tin



Thông thường có sẵn các thông tin như bên dưới, nhưng nếu chưa có thì ta thêm vào



Sau đó tiến hành chỉnh sửa file /etc/bind/named.còn.local bằng cách thêm các zone vào file



Ta thêm thông tin vào file nlưuhư hình dưới và lưu lại file.

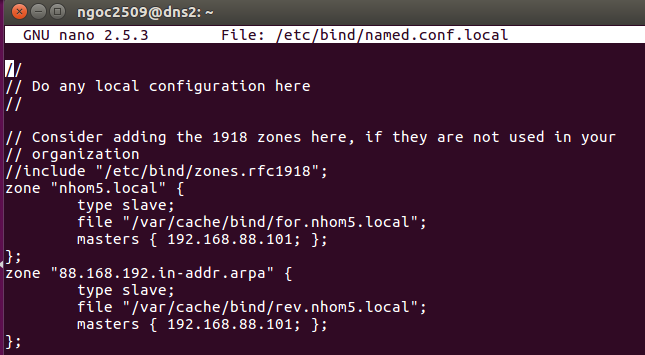
Trong đó:

Zone là nhom5.local

Sử dụng lạo slave

File được đặc ở thư mục /var/cache/bind/for.nhom5.local và /var/cache/bind/rev.nhom5.local bởi vì tường lửa App Amor của ubuntu chỉ chấp nhận cho file được ghi đè trong thư mục /var/cache/bind/ thôi.

Với master là địa chỉ IP của máy DNS1



Cấu hình Permission, Ownership cho máy chủ Bind :

Chạy lại lần lượt các dòng lệnh, ý nghĩa của các dòng lệnh cũng tương tự như được giải thích khi cấu hình ở máy DNS1

*sudo chmod -R 755 /etc/bind*

*sudo chown -R bind:bind /etc/bind*



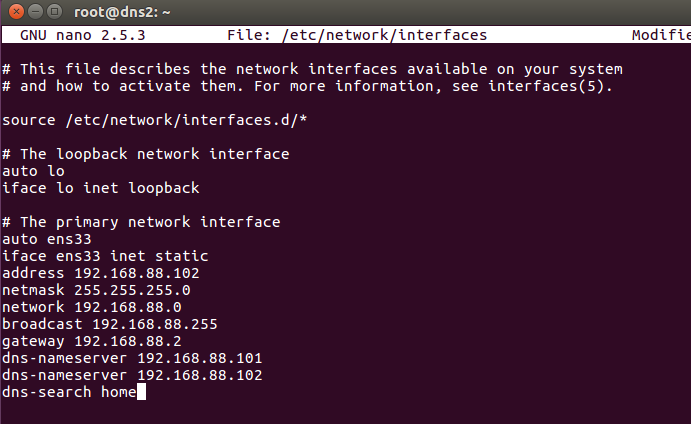
Ta restart lại cấu hình của bind9



Sau đó ta thay đổi một số thông số địa chỉ Ip cho card mạng ở file **/etc/network/interfaces**

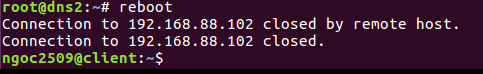


Ta xóa 2 dòng **dns-nameserver 8.8.8.8** và **8.8.4.4** và thêm **dns-search home**

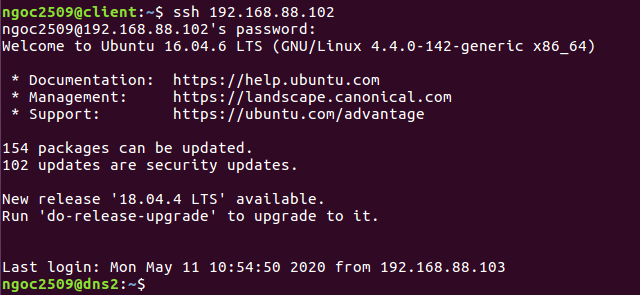


Cuối cùng, ta **reboot** lại chương trình để cập nhật các thay đổi

Với 2 dòng thông báo **Connection to 192.168.88.102 closed by remote host** và **Connection to 192.168.88.102 closed** tức là ở máy DNS2 đang reboot lại và thoát trên máy client



Sau khi đợi máy DNS2 reboot xong, ta tiến hành kết nối lại để thực hiện một số thao tác kiểm tra

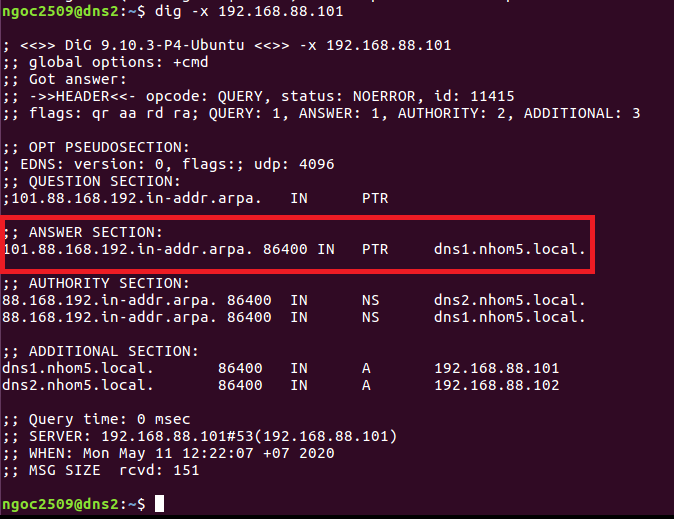


Ta sử dụng lệnh *dig -x 192.168.88.101*

Lện đó chỉ ra rằng ta đang hỏi địa chỉ 192.168.88.101 là địa chỉ của máy nào

Kết quả trả về cho chúng ta là dns1.nhom5.local

Điều này chứng tỏ máy DNS2 đã có thể phân giải được tên miền



### Trên máy Client

**Lưu ý:** Mỗi khi reboot lại máy client,1 vào file /etc/resolv.conf và edit lại như sau:



## Bước 7: Kiểm tra cấu hình DNS

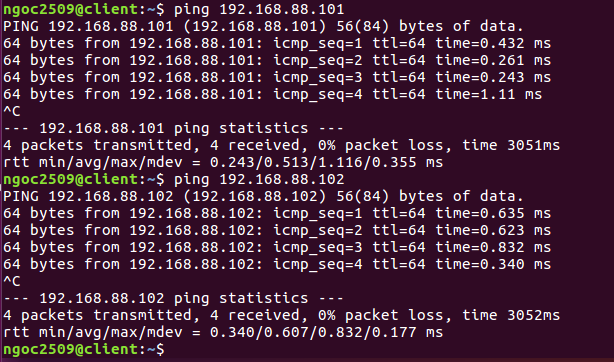
Kiểm tra trên máy client:

### Khi bật cả hai máy

Dùng lệnh ping 192.168.88.101 để ping đến máy DNS1

Và ping 192.168.88.102 để ping đến máy DNS2

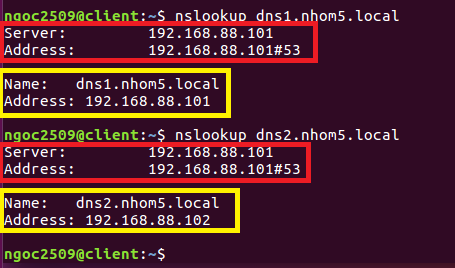
Ta thấy rằng các gói tin được trả về cho client là thành công (như hình dưới), vậy ta có thể nhận xét, khi bật cả 2 máy, các máy vẫn hoạt động bình thường.



Hoặc sử dụng lệnh nslookup dns1.nhom5.local để kiểm tra đến máy dns1 và nslookup dns1.nhom5.local để kiểm tra đến máy dns2

Ta thấy với **dns1.nhom5.local**: trước tiên sẽ hỏi máy server (192.168.88.101) là máy dns1.nhom5.local có IP là gì. Sau đó ta nhận được dòng trả lời lại là máy dns1.nhom5.local có IP là 192.168.88.101

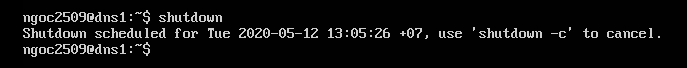
Với **dns2.nhom5.local**: trước tiên sẽ hỏi máy server (192.168.88.101) là máy dns2.nhom5.local có IP là gì. Sau đó ta nhận được dòng trả lời lại là máy dns2.nhom5.local có IP là 192.168.88.102



### Tắt máy DNS1

Gõ lệnh *shutdown* để tắt máy DNS1, một khoảng thời gian 7s sau thì máy sẽ tự tắt, nếu không muốn tắt để chỉnh sửa gì đó thì sử dụng *shutdown -c*

Ngoài ra ta còn có thể dùng lệnh *shutdown -h now* để shutdown ngay lập tức



Ta dùng lệnh nslookup dns2.nhom5.local để kiểm tra xem máy còn có thể phân giải được IP hay không

+ Khi chưa tắt máy DNS1

Trước tiên sẽ hỏi máy server (192.168.88.101) là máy dns2.nhom5.local có IP là gì. Sau đó ta nhận được dòng trả lời lại là máy dns2.nhom5.local có IP là 192.168.88.102

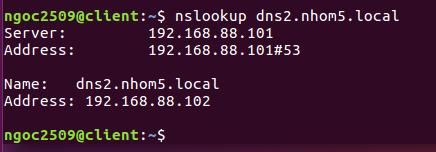
+ Khi đã tắt máy DNS1

Thông thường sẽ hỏi máy server (192.168.88.101), nhưng ở trường hợp này khi máy DNS1 đã tắt (bị hỏng) thì sẽ chuyển sang hỏi máy DNS2 (máy backup) là máy dns2.nhom5.local có IP là gì. Sau đó ta cũng nhận được dòng trả lời lại là máy dns2.nhom5.local có IP là 192.168.88.102



### Khi bật lại máy DNS1

Ta thấy sẽ hỏi máy server (192.168.88.101) là máy dns2.nhom5.local có IP là gì. Sau đó ta nhận được dòng trả lời lại là máy dns2.nhom5.local có IP là 192.168.88.102



Vậy dựa vào việc kiểm tra, ta nhận thấy rằng việc cấu hình cho các máy đã thành công, các máy hoạt động bình thường và khi máy DNS Server Master bị hỏng, máy DNS Server Slave vẫn có thể thực hiện được backup.

## Bước 8: Tiến hành cấu hình Web server.

Các thành phần để cài Web Server trên Linux gồm có Apache2 Web Server, hệ quản trị cơ sở dữ liệu MySQL (hoặc MariaDB, ở đây dùng MySQL) và PHP5 (hoặc Java, ở đây dùng PHP). Các dữ liệu trang web được lưu trữ trong cơ sở dữ liệu MySQL và nội dung được tự động xử lý bởi PHP.

Tất cả các thành phân nêu trên được cài vào Linux được gọi chung là LAMP Stack – một nhóm phần mềm mã nguồn mở. Và khi cài đặt các phần mềm này, ta cần sử dụng user root hoặc user có quyền sudo.

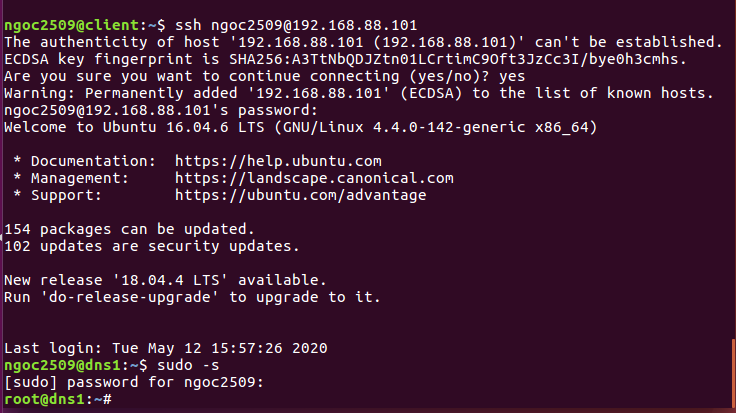
Ở đây, ta cần sử dụng trên 2 máy Server, một máy Server Master để triển khai các dịch vụ web và một máy Server Slave để backup một số trường hợp mà máy Master gặp sự cố.

Ta cần kiểm tra xem đã update lên package mới nhất chưa, nếu chưa thì dùng dòng lệnh: *sudo apt-get update*

Ở đây do đã update trong bước cài đặt DNS ở trên rồi nên không cần update lại.

### Cấu hình Apache trên máy DNS Server Master

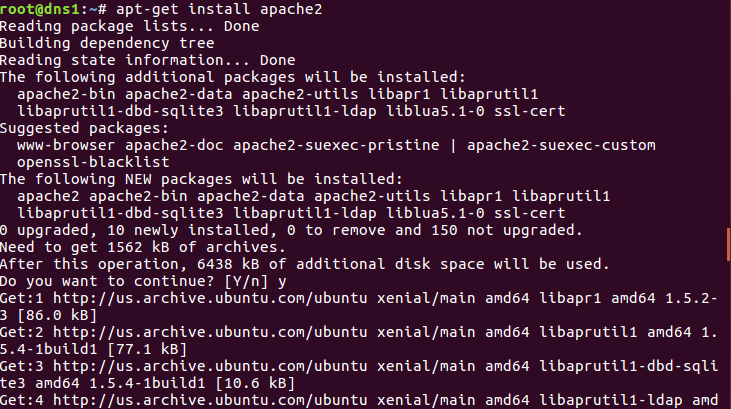
Để thuận tiện trong việc cài đặt, ta nên kết nối SSH từ máy client đến máy chủ DNS bằng dòng lệnh: *ssh* [*ngoc2509@192.168.88.101*](mailto:ngoc2509@192.168.88.101)*.* Sử dụng sudo -s để không cần nhập lại sudo ở các dòng lệnh sau nữa.



Tiếp theo, ta bắt đầu quá trình cài đặt các công cụ của web server, mà trước tiên là cài Apache, máy chủ web Apache là một trong những máy chủ phổ biến nhất trên thế giới, nó ghi chép tốt và được sử dụng rộng rãi cho lịch sử của trang web. Nó là một lựa chọn mặc định tốt cho việc lưu trữ các website.

Ta cài đặt gói Apache bằng dòng lệnh *apt-get install apache2.*

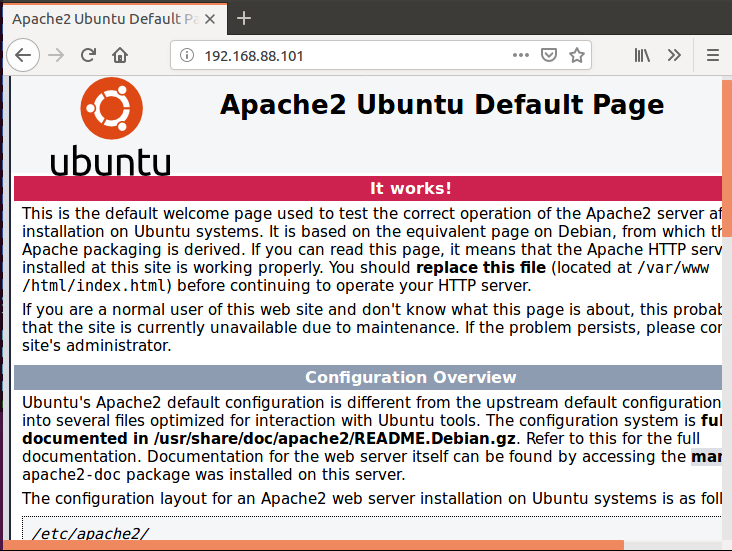
Khi đến dòng hỏi “**Do you want to continue? [Y/n]**” chit cần nhấn *y* để tiếp tục.



Kiểm tra lại gói Apache đã được cài đặt chưa bằng cách mở một trình duyệt bất kì như chorme, coccoc,… và nhập URL với cú pháp như sau http://<dia-chi-ip-hoặc-tên-miền>

Ta nhập URL là: <http://192.168.88.101>

Nếu hiển thị như hình dưới tức là đẫ cài đặt gói Apache thành công.



Đặt Global ServerName để loại bổ một số loại cú pháp

Ta sẽ thực hiện việc thêm một dòng duy nhất vào file /etc/apache2/apache2.còn để chặn thông điệp cảnh báo. Mặc dù vô hại, nếu không đặt ServerName toàn cầu, bạn sẽ nhận các thông điệp cảnh báo khi kiểm tra cấu hình Apache của bạn với một số lỗi cú pháp.

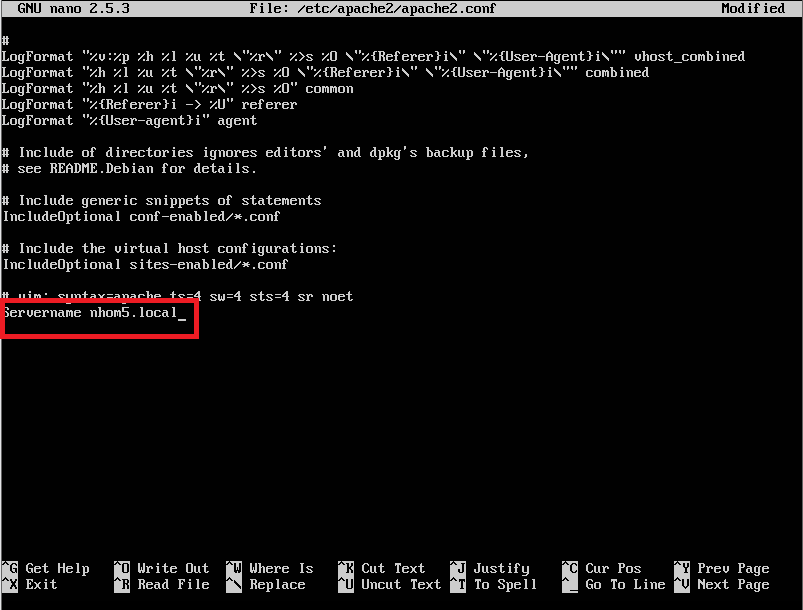
Mở file cấu hình và thực hiện chỉnh sửa: *nano /etc/apache/apache2.conf*



Ở dưới cùng của tệp, thêm một lệnh ServerName, chỉ vào chính tên miền của máy.

Nếu không có tên miền liên kết với máy chủ, có thể sử dụng chính địa chỉ IP public của mình.

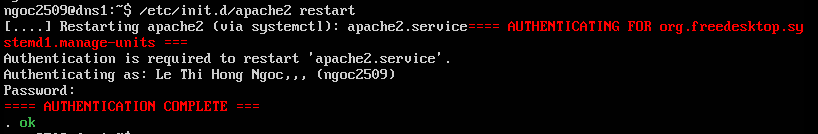
Ta thêm *ServerName nhom5.com* vào cuối file như hình dưới và lưu file lại.



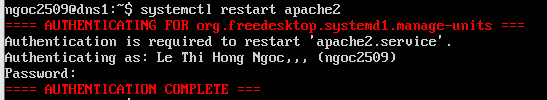
Kiểm tra lại lỗi cú pháp bằng dòng lệnh *apache2ctl configtest*. Nếu trả về là **Systax OK** là thành công



Tiến hành khởi động lại để chấp nhận các sự thay đổi bằng dòng lệnh */etc/init.d/apache2 restart*

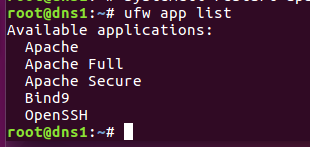


Khởi động lại Apache : *systemctl restart apache2*

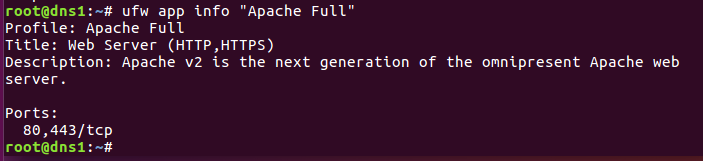


Phải chắc chắn rằng tường lửa của bạn cho phép HTTP và HTTPS bằng cách xem nội dùn ufw có nội dung Apache, Apache Full avf Apache Secure hay không, nếu không có, ta thiết lập lại tường lửa

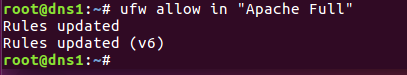
Xem nội dung bằng dòng lệnh *ufw app list.* Ta thấy rằng 3 nội dung đó đã có sẵn trong profile



Nếu nhìn vào thông tin file **Apache Full**, nó sẽ cho thấy rằng nó cho phép lưu lượng truy cập đến các cổng 80 vad 443 bằng giao thức TCP



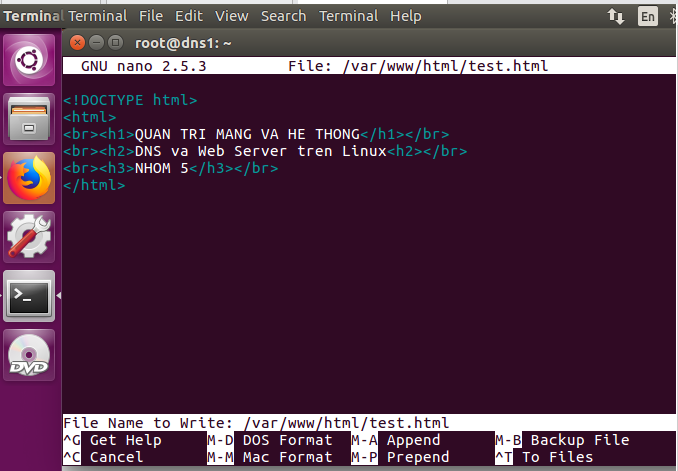
Thực hiện dòng lệnh *ufw allow in “Apache Full”* để cho phép các truy cập đến file này



Tiếp đến để test lại Apache chúng ta vừa tạo đẫ có thể hoạt động như một trang web chưa bằng cách ta tạo một file test.html



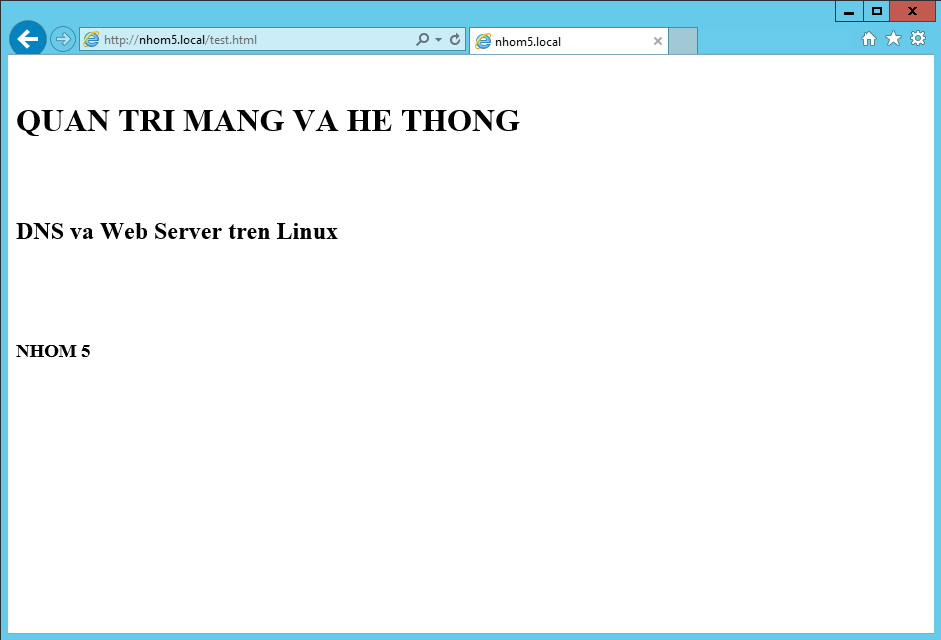
Thêm một nội dung bằng mã nguồn của html bất kì bạn muốn và lưu tệp lại



Kiểm tra xem ta có thể truy cập trang html vừa tạo được trên internet chưa bằng cách mở lên một trình duyệt bất kì, nhập trên thanh URL là <http://192.168.88.101/test.html> hoặc <http://nhom5.local/test.html> như 2 hình dưới.

Ta thấy đã có thể truy cập vào trang web vừa được tạo thành công

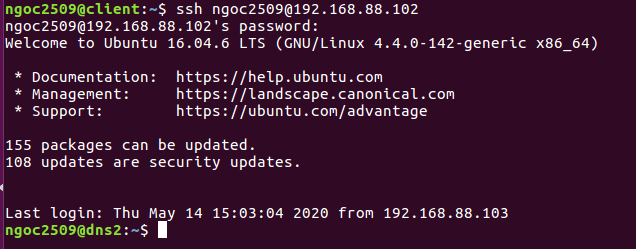




### Cấu hình Apache trên máy Server Slave

Trước tiên, để dễ check ta nên tắt máy Server Master trước

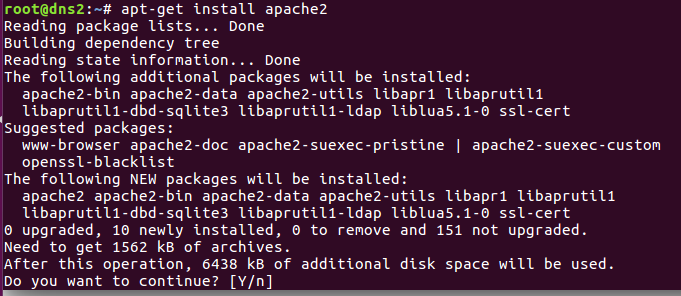
Đăng nhập SSH từ máy client vào máy Server Slave:



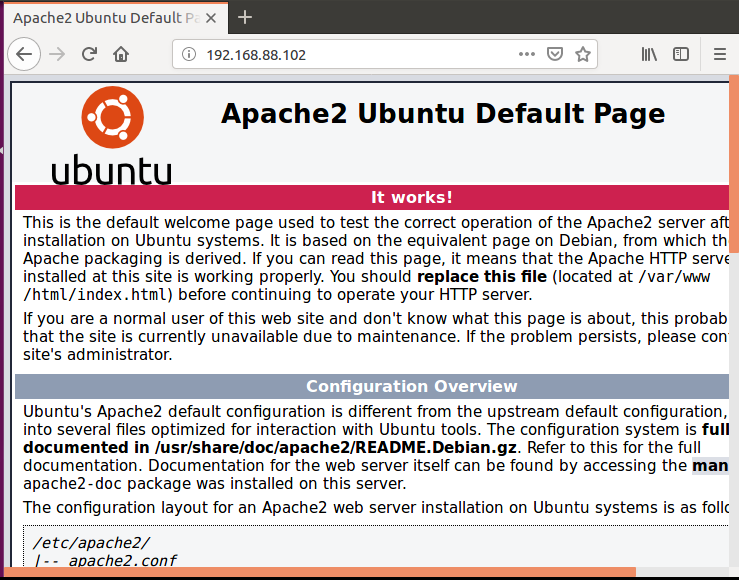
Và dùng lệnh sudo -s để khỏi nhập sudo ở các dòng lệnh sau:



Bước cài đặt gói apache này cũng tương tự như ở máy Server Master, ta dùng lệnh: *apt-get install apache2*



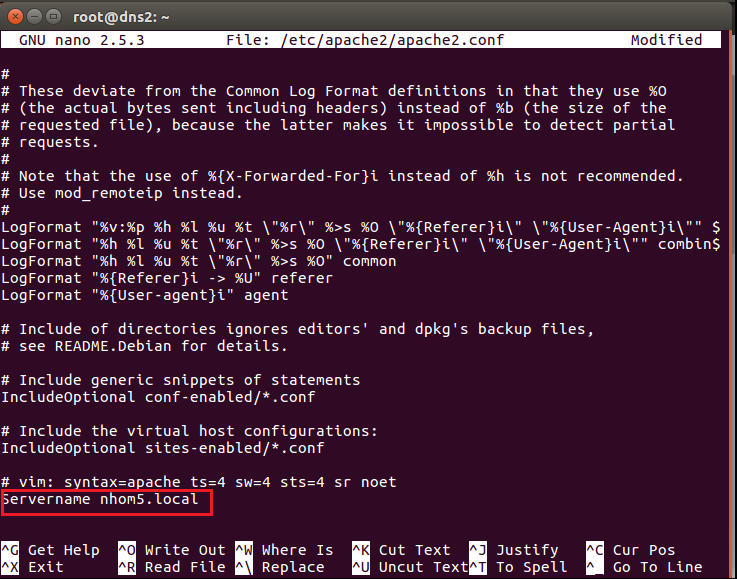
Sau khi cài đặt hoàn tất, kiểm tra lại trên trình duyệt web với URL là: <http://192.168.88.102>, nếu kết quả như hình dưới tức là ta đã cài đặt thành công



Tiếp đến, ta chỉnh sửa thông tin của file /etc/apache2/apache2.conf để thêm tên miền vào, ta dùng lệnh: *nano /etc/apache2/apache2.conf*



Kéo đến cuối file và chèn thêm dòng **Servername nhom5.local** để thêm domain vào apache



Kiểm tra lại lỗi cú pháp bằng dòng lệnh *apache2ctl configtest*. Nếu trả về là **Systax OK** là thành công



Tiến hành khởi động lại để chấp nhận các sự thay đổi bằng dòng lệnh */etc/init.d/apache2 restart*



Restart lại apache để áp dụng tahy đổi trên apache



Dùng lệnh **ufw allow in “Apache Full”** để chấp nhận quyền truy cập đến Apache Full



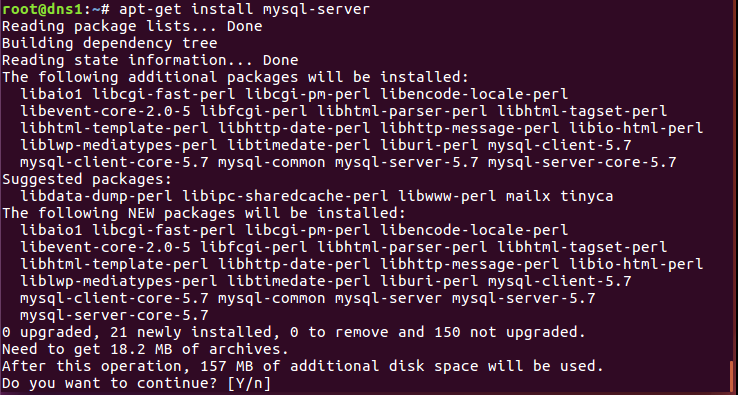
## Bước 9: Cài đặt Mysql

### Trên máy DNS Server Master

Bây giờ máy chủ web đã hoạt động, đã đến lúc cài đặt MySQL. MySQL là một hệ thống quản lý cơ sở dữ liệu. Về cơ bản, nó sẽ tổ chức và cung cấp quyền kết nối đến cơ sở dữ liệu cuãng là nơi mà chúng ta có thể lưu trữ thông tin.

Một lần nữa, chúng ta sử dụng apt để có quyền và cài đặt phần mềm. Lúc này, chúng ta cũng sẽ cài đặt một số gói “heaper” khác mà nó sẽ hỗ trợ chúng ta trong quá trình liên lạc lẫn nhau:

Ở đây là sẽ dùng dòng lệnh *apt-get install mysql-server*

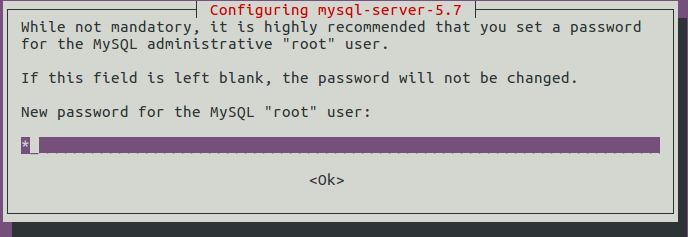


Ở đây sẽ hiển thị một danh sách các gói tin đẫ được cài đặt, cùng với dung lượng ổ đĩa mà chúng chiếm. Nhập **Y** để tiếp tục.

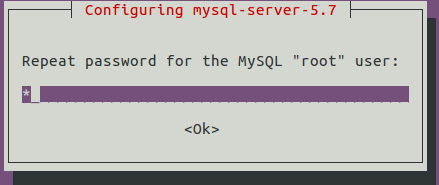
Lưu ý: Trong trường hợp này không cần phải chạy lệnh *sudo apt-get update* trước đó dó đã được sử dụng ở các bước trên. Chỉ mục gói tỉn trên máy tính đẫ được cập nhật.

Trong quá trình cài đặt, máy chủ sẽ yêu cầu xác nhận mật khẩu cho người dùng gốc MySQL. Đây là một tài khoảng quản trị trong MySQL đã tăng đặc quyền. Nó cũng giống như tài khoảng root cho chính máy chủ (tuy nhiên tài khoảng đang cấu hình là một tài khoảng dành riêng cho MySQL). Ở đây không bắt buộc một mật khẩu mạnh nhưng nếu để sử dụng cho một server đích thực ta nên đặt một mật khẩu đủ mạnh để bảo mật tốt hơn (trong trường hợp này do nhóm sử dụng để làm đề tài nên đặt mật khẩu hơi đơn giản).

Dù không yêu cầu mật khẩu mạnh như cũng không được để trống.



Ở bước này để nhập lại mật khẩu vừa rồi để xác minh, và nhấn **Ok**



Khi quá trình cài đặt hoàn tất, ta sẽ chạy một số tập lệnh đơn giản để loại bỏ một số mặc định nguy hiểm và khóa quyền truy cập vào hệ thống cơ sở dữ liệu. Bắt đầu tập lệnh tương tác bằng cách chạy lệnh: *mysql\_secure\_installation*

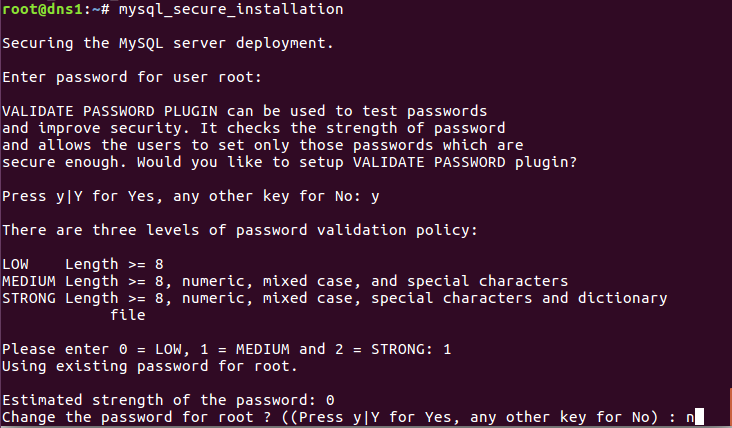
Ở đây sẽ được yêu cầu nhập mật khẩu đã đặt cho tài khoản MySQL bên trên. Tiếp theo sẽ được hỏi nếu bạn muốn cấu hình VALIDATE PASWORD PLUGIN.

**Lưu ý:** Kích hoạt tính năng này là mọt cái gì đó cho một cuộc gọi phán xét. Nếu được bật, mật khẩu không phù hợp với tiêu chí đã chỉ định sẽ bị MySQL từ chối do bị lỗi. Điều mày sẽ gây ra sự cố nếu bạn sử dụng một mật khẩu yếu kết hợp với phần mềm tự động định cấu hình thông tin đăng nhập của người dùng MySQL, chẳng hạn như các gói Ubuntu cho phép phpMyAdmin. Sẽ an toàn khi vô hiệu háo xác thực, nhưng bạn vẫn pahir luốn sử dụng một mật khẩu mạnh, suy nhất cho thông tin cơ sở dữ liệu

Có thể trả lời y hoặc bất kì cái khác để tiếp tục mà không kích hoạt.

Tiếp theo, Bạn sẽ được yêu cầu chọn một mức độ xác nhận mật khẩu. Nếu nhập 2, ở mức mạnh nhất, có thể sẽ gặp lỗi khi cố gắng đặt bất kỳ mật khẩu không chứa số, chữ in hoa và chữ thường và ký tự đặc biệt hoặc dựa trên các từ điển thông dụng. (Ở đây cho làm đề tài trên lớp nên có thể sử dụng mức 1: trung bình hoặc mức 0: thấp để dễ dàng theo tác hơn)

Nếu đã bật xác thực mật khẩu, bạn sẽ được hiển thị cường độ mật khẩu cho mật khẩu gốc hiện có và hỏi bạn có muốn thay đổi mật khẩu đó không (ở đây cường độ mật khẩu của nhóm là 0). Nếu đã hài lòng với mật khẩu hiện tại của mình, nhập **n** cho không có dấu nhắc tại dấu nhắc:



Đối với các câu hỏi còn lại, bạn nên nhấn **Y** và nhấn phím **Enter** tại mỗi dấu nhắc. Điều này sẽ xóa một số người dùng ẩn danh và cơ sở dữ liệu thử nghiệm, vô hiệu hóa đăng nhập gốc từ xa và tải các quy tắc mới này để MySQL ngay lập tức tôn trọng các thay đổi chúng tôi đã thực hiện.

Khi xuất hiện dòng bên dưới tức là hệ thống cơ sở dữ liệu của bạn hiện đã được thiết lập và có thể tiếp tục.

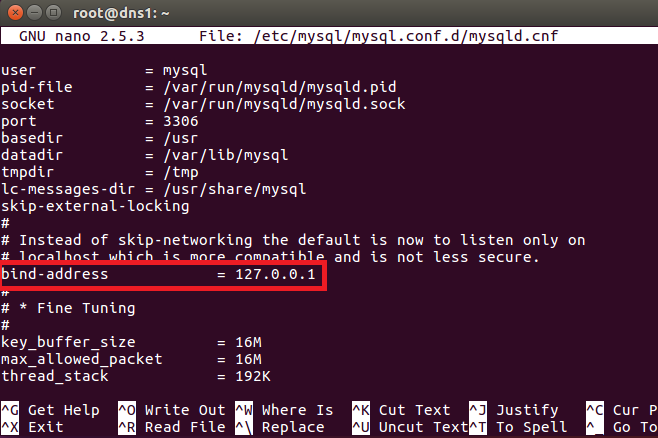


Tiếp theo, tiến hành việc cấu hình bản sao của Mysql, để khi máy Master có bị hỏng cũng vẫn có thể lưu trữ lại CSDL mà không bị mất.

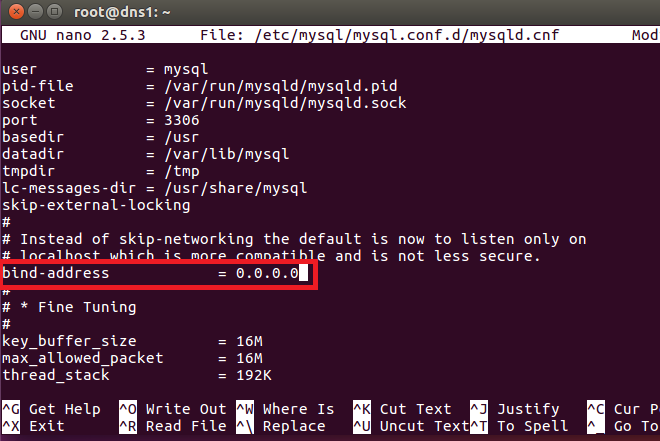
Bước đầu tiên, ta chỉnh sửa tệp cấu hình MySQL để máy chủ MySQL cho phép truy cập từ xa bằng dòng lệnh *nano /etc/mysql/mysql.conf.d/mysqld.cnf*



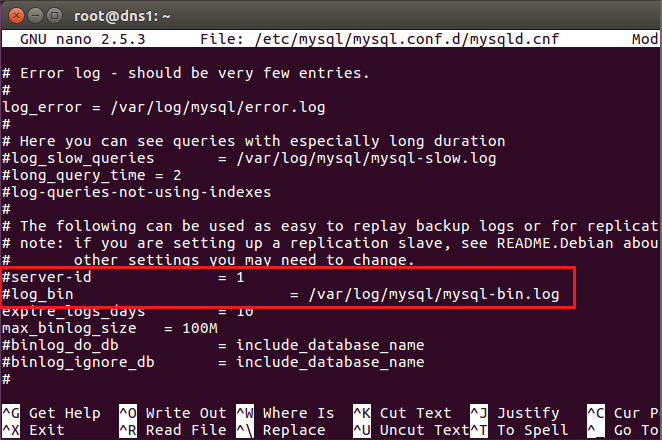
Bên dưới là file **/etc/mysql/mysql.conf.d/mysqld.cnf**, ta tìm đến vị trí dòng **bind-address =127.0.0.1**



Thay đổi thành *bind-address = 0.0.0.0*



Tiếp theo, tìm đến dòng khung đỏ bên hình dưới và tiến hành edit, điều này sẽ khiến MySQL cho phép kết nối trong nước.



Thêm các dòng sau đây vào tệp cấu hình đó và lưu lại tệp.

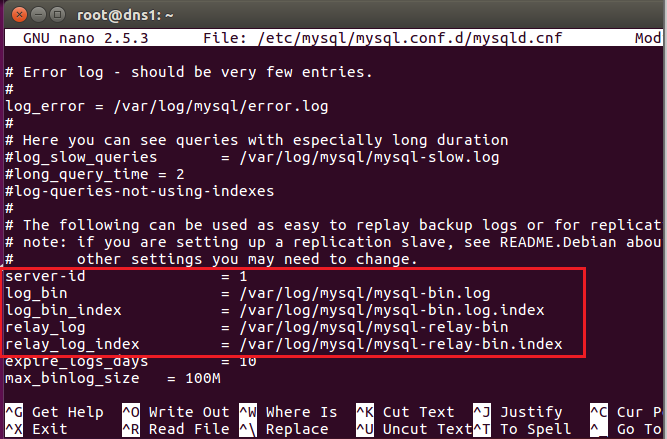
*server-id = 1*

*log\_bin = /var/log/mysql/mysql-bind.log*

*log\_bin\_index = /var/log/mysql/mysql-bind.log.index*

*relay\_log = /var/log/mysql/mysql-relay-bind*

*relay\_log\_index = /var/log/mysql/mysql-relay-bind.index*

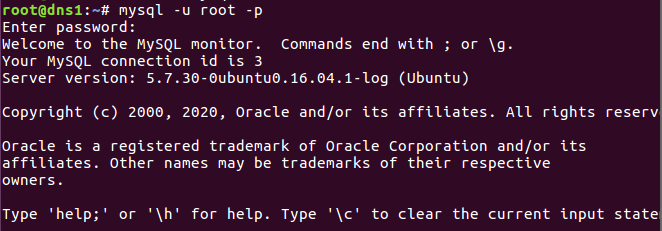


Khởi động lại máy chủ MySQL để thay đổi có hiệu lực:



Tiếp theo, cần tạo một người dùng mới sẽ xử lý các dịch vụ sao chép trên máy chủ Master. Vì vậy, ta đăng nhập vào MySQL của bạn trên máy chủ Master.

Dùng lệnh *mysql -u root -p* để đăng nhập, mật khẩu là mật khẩu tạo MySQL ở trên và nhấn **enter** để tiếp tục



Trên dấu nhắc MySQL xuất hiện, nhập các truy vấn SQL bên dưới để tạo người dùng bằng các lệnh:

>*create user ‘replication\_user’@’%’ identified by ‘Qtm&htNhom5’;*

Trong đó:

replication\_user là username

Qtm&htNhom5 là password, lưu ý có thể thay thế một password khác bằng một giá trị mạnh nhầm mục đích bảo mật, ở đây nếu nhập một mật khẩu yếu sẽ bị báo lỗi.

Yêu cầu có ; cuối dòng thì mới hiểu là kết thúc một câu lệnh

>*GRANT REPLICATION SLAVE ON \*.\* TO ‘replication\_user’@’%’;*

Cấp quyền sao chép cho user *replication\_user*

*>FLUSH PRIVILEGES;*

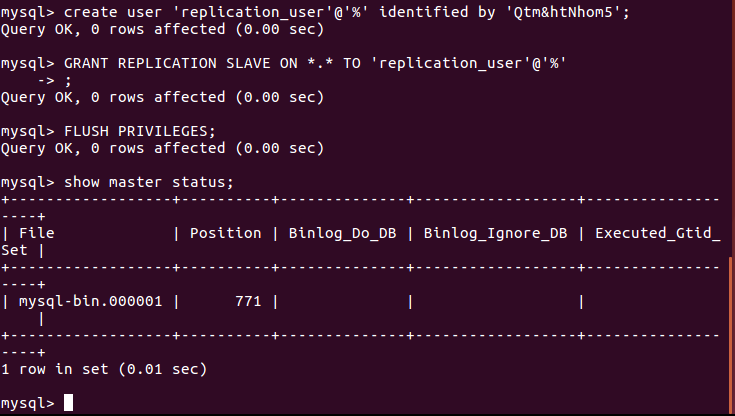
Trước khi thoát khỏi dấu nhắc MySQL, nhập lệnh bên dưới để xem thông tin về vị trí và tệp của máy Master.

>*show master status;*

Tại đây, bạn cần lưu ý tên tệp, ' **mysql-bin.000001** ' và Vị trí ' **771** '. Ta sẽ cần những chi tiết này khi cấu hình **Server Slave** .Lưu ý, những giá trị này có thể khác trên những máy chủ khác.

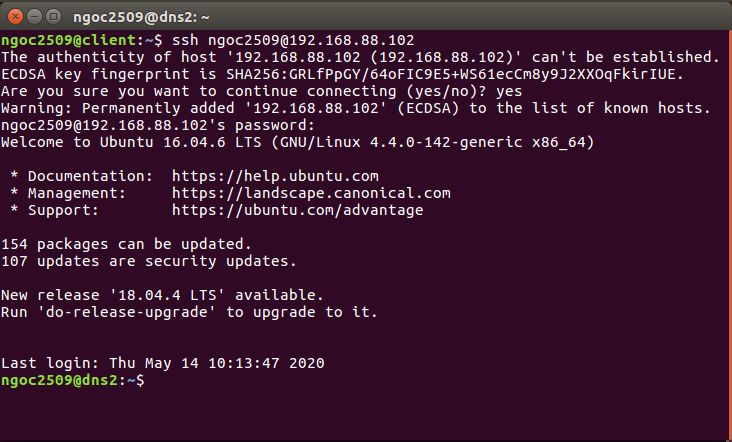
Để thoát khỏi dấu nhắc của MySQL ta dùng lệnh quit

>Quit;



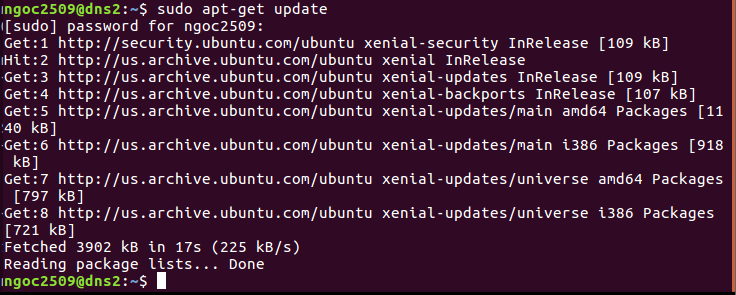
### Trên máy DNS Server Slave

Trước tiên, để tiện cấu hình ta đăng nhập SSH từ máy client kết nối đến server slave



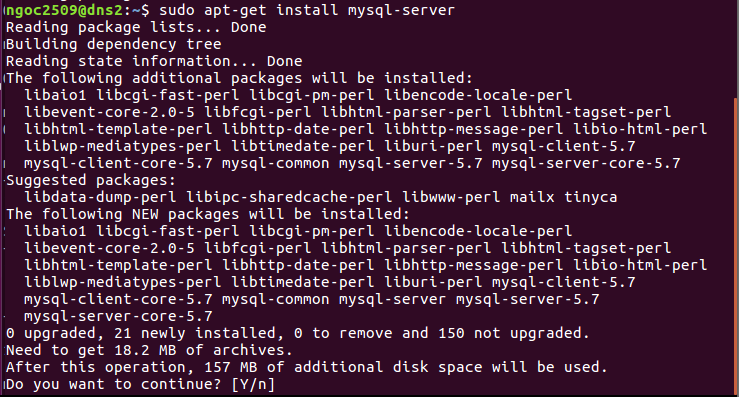
Giống như đã làm trong máy chủ đầu tiên, ta cần cài đặt máy chủ MySQL trên máy chủ thứ hai.

Ta cần update các packeage lên mới nhất bằng lệnh *sudo apt-get update*

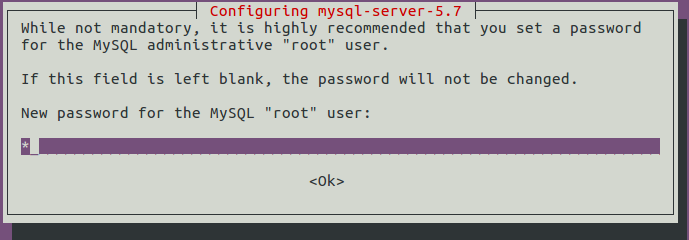


Ta cài đặt gói mysql-server lên **Server Slave** bằng lệnh *sudo apt-get install mysql-server*

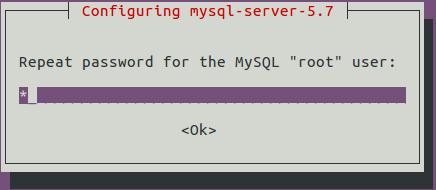
Nhấn **Y** và nhấn **Enter** khi được nhắc để xác nhận cài đặt.



Ngoài ra, sẽ cần cung cấp mật khẩu gốc cho máy chủ MySQL của mình. Sau đó, nhấn **Enter** trên màn hình bật lên. Lặp lại mật khẩu một lần nữa trên màn hình tiếp theo để xác nhận.



Xác nhận lại mật khẩu ở trên:



Ta có thể sử dụng sudo -s để không cần nhập sudo cho các lệnh phía sau:



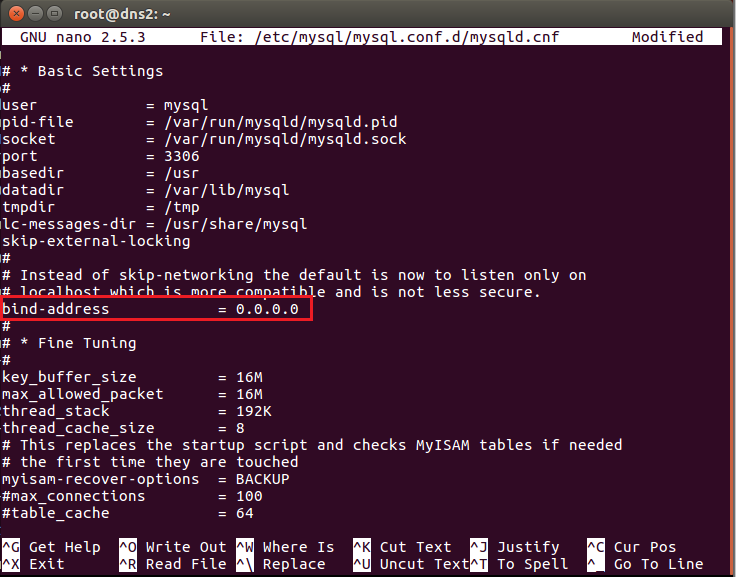
Tiếp theo, chỉnh sửa tệp cấu hình MySQL trên máy chủ slave để cho phép truy cập từ xa ta dùng lệnh sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf



Ý nghĩa ở bước này là để truy cập đến CSDL từ máy Master, ở máy Slave ngoài việc có CSDL giống như máy Master, có còn là chức năng backup nếu lỡ khi máy chính bị hỏng thì cơ sở dữ liệu vẫn không bị mất.

Điều quan trong khi làm một trang web là cần có backup nếu hệ thống bị lỗi, mà cái cần backup nhất chính là cơ sở dữ liệu trong đó.

Ta tìm đến dòng **bind-address = 127.0.0.1** và thay đổi thành *bind-address = 0.0.0.0* như ở máy Server Master.



Tiếp theo ta tìm đến dòng **#server-id = 1** ở gần cuối file và tiến hành edit

*server-id = 2*

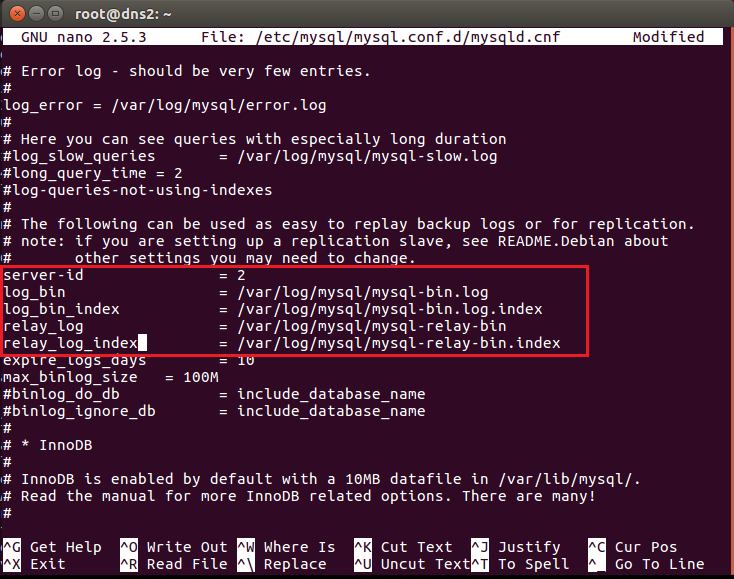
*log\_bin = /var/log/mysql/mysql-bind.log*

*log\_bin\_index = /var/log/mysql/mysql-bind.log.index*

*relay\_log = /var/log/mysql/mysql-relay-bind*

*relay\_log\_index = /var/log/mysql/mysql-relay-bind.index*

**Lưu ý**: server-id của máy chủ slave là **2**

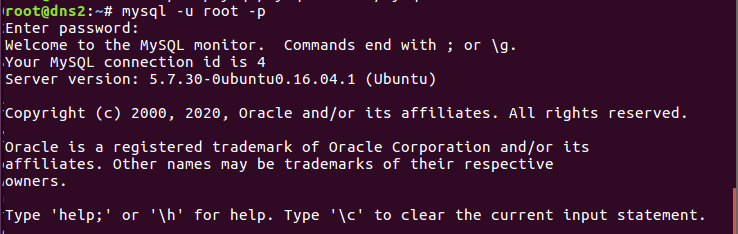


Khởi động lại MySQL trên máy chủ 2 để các thay đổi có hiệu lực:



Tiếp theo, đăng nhập vào MySQL trên máy chủ slave bằng dòng lệnh *mysql -u root -p*

Nhập mật khẩu MySQL ở trên và nhấn **Enter** để tiếp tục.



Lần lượt dùng các lệnh sau:

>stop slave;

>CHANGE MASTER TO MASTER\_HOST = ‘192.168.88.101’, MASTER\_USER = ‘replication\_user’, MASTER\_PASSWORD = ‘Qtm&htNhom5’, MASTER\_LOG\_FILE = ‘mysql-bin.000001’, MASTER\_LOG\_POS = 771;

Trong đó:

Trên lời nhắc của MySQL xuất hiện, nhập các dòng lệnh bên dưới và đặt địa chỉ IP là địa chỉ kết nối đến máy DNS Server Master.

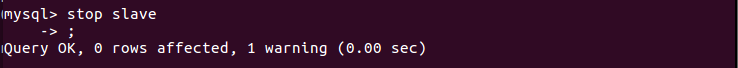
Ngoài ra, thay thế ' **MASTER\_LOG\_FILE** ' và ' **MASTER\_LOG\_POS** ' bằng thông tin có được từ máy chủ Master (**mysql-bin.000001** và **771)**

Tên người dùng và mật khẩu MySQL cũng phải khớp với các giá trị đã tạo trên máy chủ Master.

>start slave;

Dùng lệnh quit để thoát khỏi lời nhắc trên MySQL

>Quit;

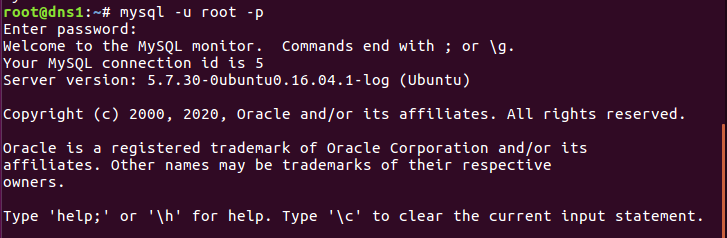




### Kiểm tra

Bây giờ ta bắt đầu kiểm tra việc cấu hình và sao chép tài liệu MySQL có như mong đợi không. Để xác nhận điều này, ta làm theo các bước dưới đây

* Bước 1: Đăng nhập MySQL trên máy Master

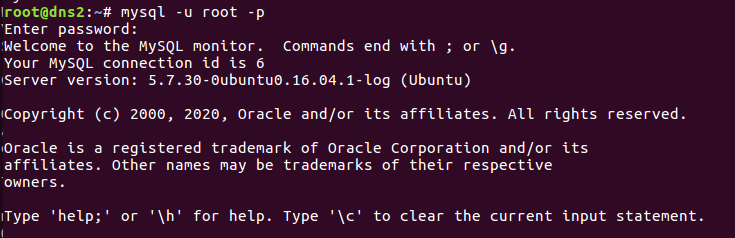


* Bước 2: Tạo CSDL với tên nhom5\_demo

Trên dấu nhắc của MySQL ta dùng lệnh *create database nhom5\_demo;*

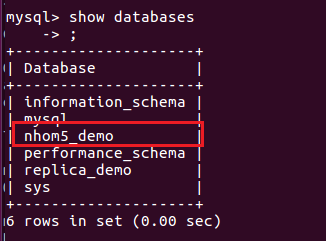


* Bước 3: Đăng nhập MySQL trên máy Slave



* Bước 4: Trên dấu nhắc của MySQL, nhập lệnh *show databases;* để liệt kê tất cả cơ sở dữ liệu có sẵn

Đầu ra ở trên là dành cho máy chủ Slave và nó xác nhận rõ ràng rằng bản sao MySQL Master-Slave đang hoạt động như mong đợi vì cơ sở dữ liệu ‘nhom5\_demo’ đã được sao chép trên máy chủ Slave.



Chúng ta có thể kiểm tra sâu sơn bằng cách thêm một bảng ‘sinh viên’ trên database nhom5\_demo ở máy chủ Master

* Bước 5: Dùng lệnh use nhom5\_demo ở máy Master để thêm 1 bảng vào database.



* Bước 6: Sau đó tạo bảng ‘sinh vien’ như câu lệnh dưới

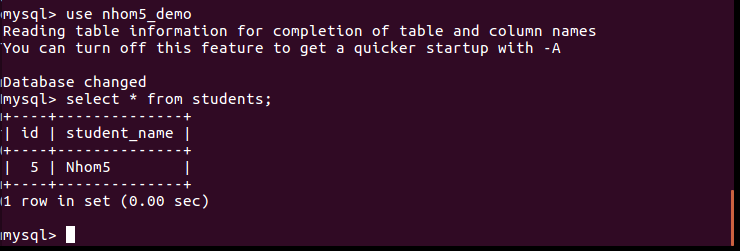


* Bước 7: Sau đó thêm giá trị về thông tin sinh viên vào bảng



* Bước 8: Trên máy chủ Slave, ta dùng lệnh *select \* from students;* để liệt kê bảng ghi trên bảng sinh viên vừa được tạo

Ta thấy. bảng sinh viên vừa được tạo trên Server Master và danh sách sinh viên đã được sao chép thành công trên máy Server Salve.



## Bước 10: Cài đặt php

### Trên máy Server Master

PHP là thành phần thiết lập xử lý mã để hiển thị nội dung. Nó có thể chạy các tập lệnh, kết nối với CSDL MySQL để lấy thông tin và trao đổi nội dung được xử lý cho máy chủ web của chúng ta để hiển thị lên.

Chúng ta có thể một lần nữa tận dụng apt - hệ thống để cài đặt của ubuntu. Nó cũng sẽ bao gồm một số gói trợ giúp, để mã PHP có thể chạy trong máy chủ Apache và giao tiếp với cơ sở dữ liệu MySQL của chúng ta:

Để cài đặt ta dùng lệnh: *apt-get install php libapache2-mod-php php-mcrypt php-mysql*

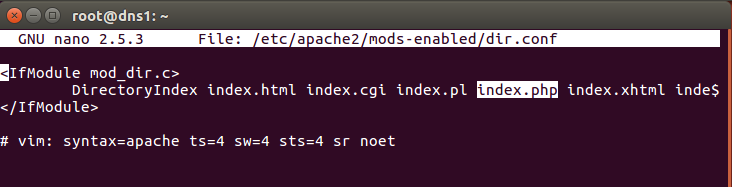


Trong hầu hết các trường hợp, chúng ta muốn sửa đổi cách Apache phục vụ các tệp khi thư mục được yêu cầu. Hiện tại, nếu người dùng yêu cầu một thư mục từ máy chủ, trước tiên Apache sẽ tìm kiếm một tệp có tên index.html. Nếu muốn nói với máy chủ web rằng chúng ma muốn chọn các tệp PHP, vì vậy chúng ta sẽ làm cho Apache tìm kiếm một tệp index.php trước tiên.

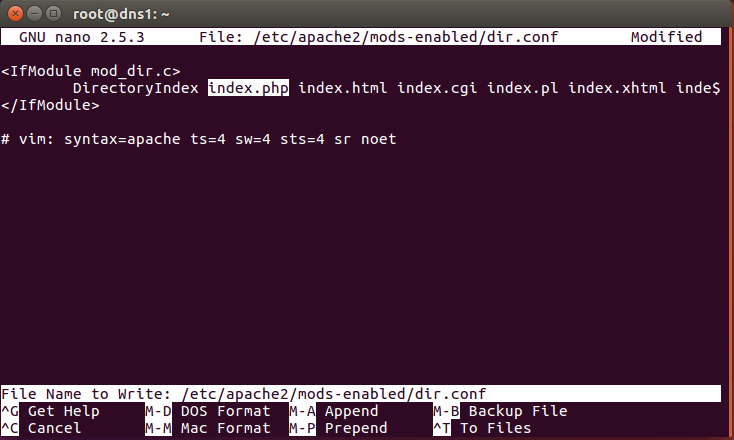
Ta dùng lệnh *nano /etc/apache2/mods-enabled/dir.conf* để chỉnh sửa tệp



Hình bên dưới ta tệp sau khi được mở, ta thấy rằng thứ tự tìm kiếm của Apache lần lượt là .html, .cgi, .pl, .php, …



Ta chỉnh sửa đưa index.php lên đầu tiên và sau DirectoryIndex và lưu lại file

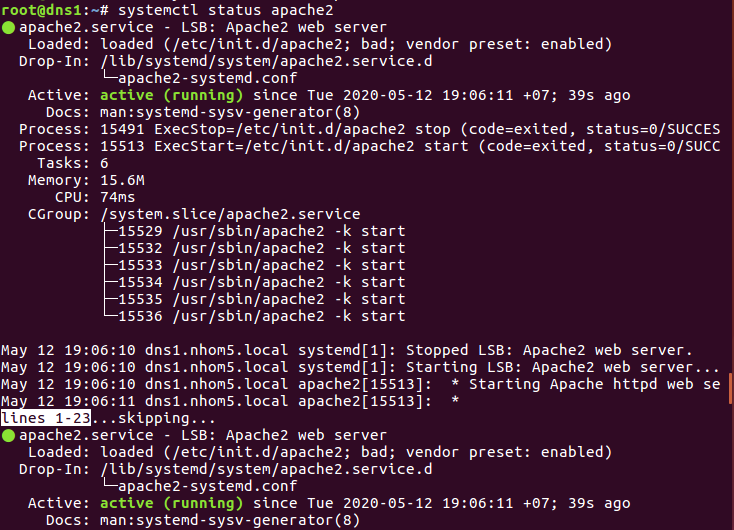


Sau đó, ta tiến hành khởi động lại máy chủ web Apache để các thay đổi được công nhận.

Ta dùng lệnh: *systemctl restart apache2*



Ta có thể kiểm tra trạng thái của dịch vụ Apache bằng cachs sử dụng lệnh: *systemctl status apache2*



Cài đặt các modules php

Để tăng cường chức năng của PHP, chúng ta có thể tùy chọn cài đặt một số mô-đun bổ sung.

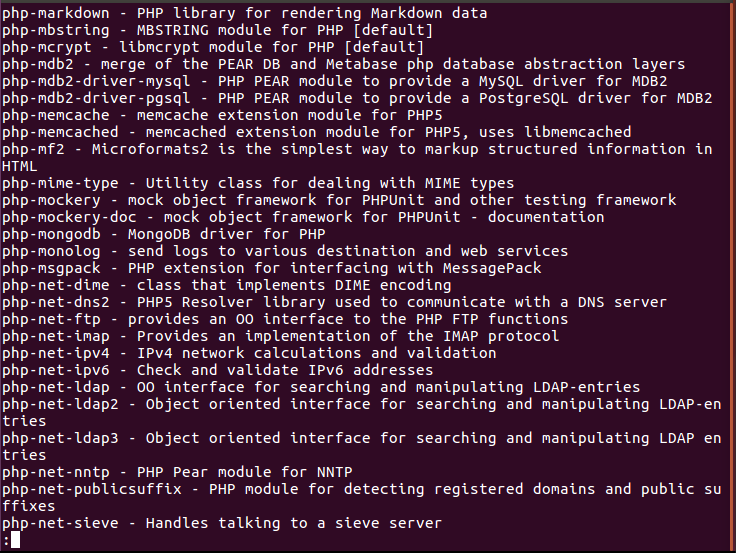
Để xem các tùy chọn có sẵn cho các mô-đun và thư viện PHP, ta có thể dẫn các kết quả apt-cache serach vào less, nó cho phép bạn cuộn qua đầu ra của các lệnh khác:

Ta sử dụng dòng lệnh: *apt-cache search php- | less*



Dùng các phím mũi tên để cuộn lên xuống và **q** để thoát

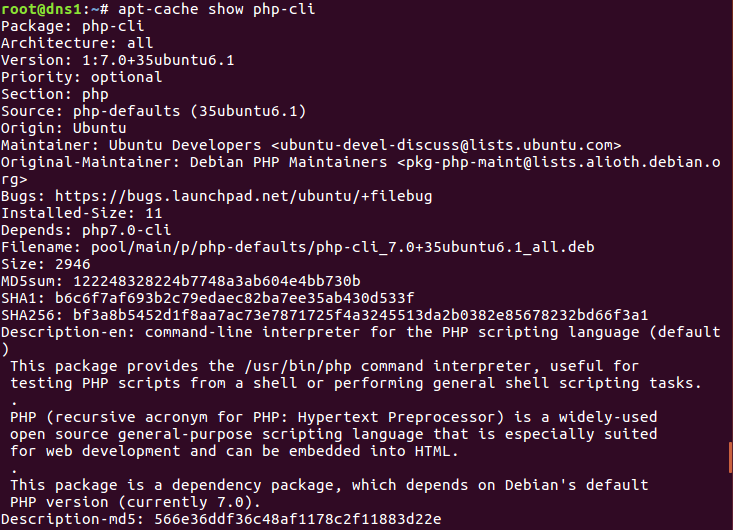
Kết quả là tất cả các thành phần tùy chọn mà bạn có thể cài đặt. Nó sẽ cung cấp cho bạn một mô tả ngắn cho mỗi gói:



Để có thêm thông tin về những gì mỗi mô-đun làm, bạn có thể tìm kiếm trên internet hoặc bạn có thể xem mô tả dài của gói bằng cách nhập:

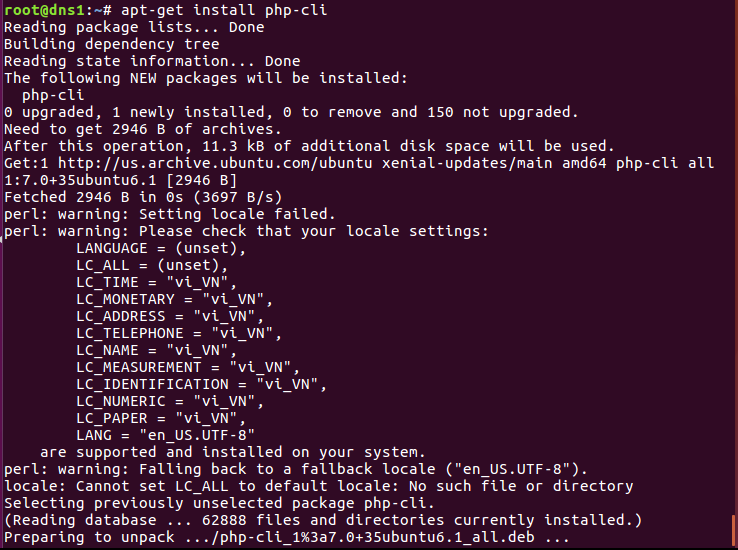
**apt-cache show <package-name>**

ví dụ: để tìm hiểu về php-cli ta dùng lệnh: *apt-cache show php-cli*



Nếu, sau khi nghiên cứu, bạn quyết định bạn muốn cài đặt một gói, bạn có thể dùng lệnh bằng cách sử dụng **apt-get install** đã làm cho phần mềm khác.

Nếu quyết định sẽ cài php-cli là thứ chúng ta cần, sau đó sẽ dùng lệnh: **apt-get install php-cli** để cài đặt nó vào



Nếu bạn muốn cài đặt nhiều hơn một mô-đun, bạn có thể làm điều đó bằng cách liệt kê từng mô-đun, cách nhau bởi một khoảng trắng, theo lệnh **apt-get install**, như sau:

*apt-get install package1 package2 package3 …*

Tại thời điểm này, ngăn xếp LAMP được cài đặt và định cấu hình. Chúng ta vẫn nên kiểm tra lại PHP.

Để kiểm tra xem hệ thống của có được cấu hình đúng cho PHP không, chúng ta có thể tạo một tập lệnh PHP rất cơ bản.

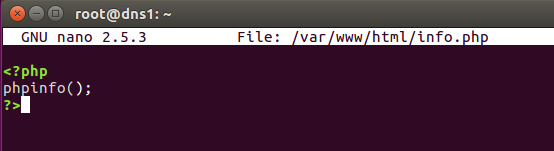
Chúng ta sẽ tạo một file info.php. Để Apache tìm thấy tệp và phục vụ nó một cách chính xác, nó phải được lưu vào một thư mục rất cụ thể, được gọi là web web trực tuyến.

Trong Ubuntu 16.04, thư mục này được đặt tại /var/www/html. Chúng ta có thể tạo tệp tại vị trí đó bằng cách nhập: *nano /var/www/html/info.php*



Đặt một mã php vào đó, ở đây sẽ đặt một dòng lệnh nhỏ để kiểm tra.

Với một mã php hợp lệ sẽ được đặt vào trong <?php ?>

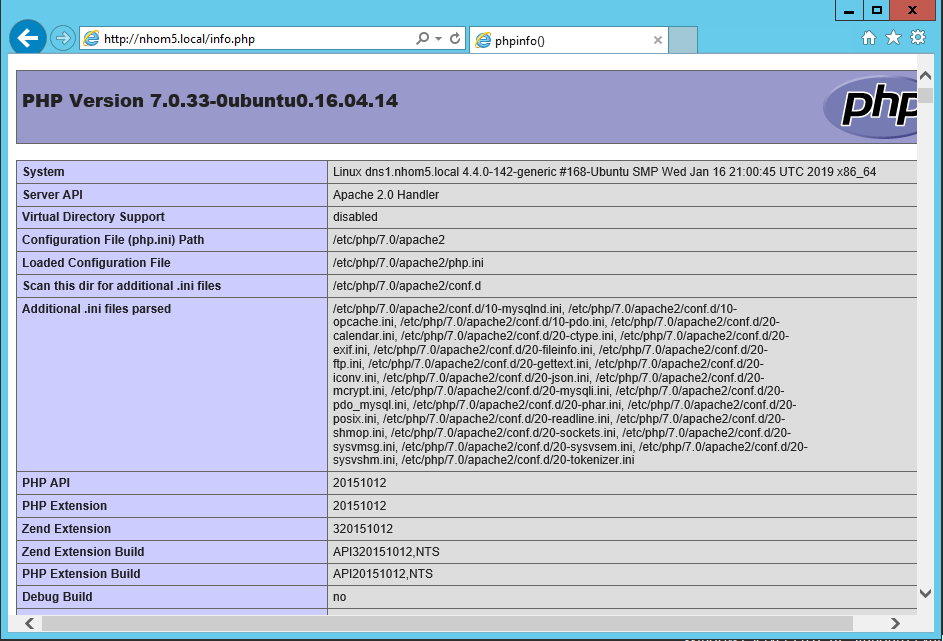


Khi bạn hoàn thành, lưu và đóng tệp.

Bây giờ chúng ta có thể kiểm tra xem máy chủ web của chúng ta có thể hiển thị chính xác nội dung được tạo bởi tập lệnh PHP hay không. Để thử điều này, chúng ta chỉ cần truy cập trang này trong trình duyệt web bất kì. Bạn sẽ cần địa chỉ IP hoặc một tên miền của máy chủ của bạn một lần nữa.

Trên đường dẫn URL dùng dùng lệnh: <http://192.168.88.101/info.php> hoặc http://nhom5.local/info.php

Nếu kết quả được hiển thị như hình dưới tức là quá trình cài đặt PHP đã hoàn tất.



Trang này về cơ bản cung cấp cho bạn thông tin về máy chủ của bạn từ quan điểm của PHP. Nó rất hữu ích để gỡ lỗi và để đảm bảo rằng các cài đặt của bạn đang được áp dụng chính xác.

Nếu điều này thành công, thì PHP của bạn sẽ hoạt động như mong đợi.

Bạn có thể muốn xóa tệp này sau khi kiểm tra vì nó thực sự có thể cung cấp thông tin về máy chủ của bạn cho người dùng trái phép. Để làm điều này, bạn có thể gõ lệnh: *rm /var/www/html/info.php*



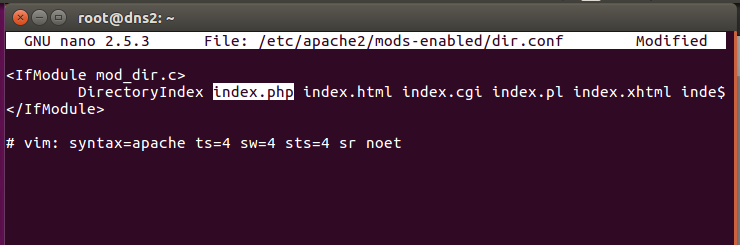
### Trên máy Server Slave

Trên máy này, ta làm tương tự như ở máy Server Master

Trước tiên ta cần cài gói PHP



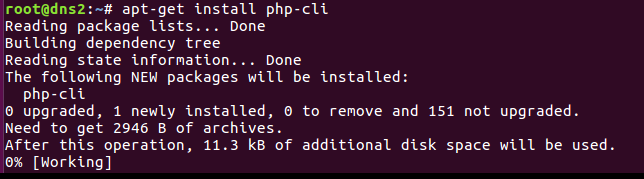
Tiếp theo chỉnh sửa file **/etc/apache2/mods-enabled/dir.conf** bằng cách chuyển index.php lên đầu sau **DirectoryIndex**



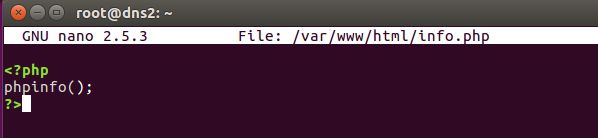
Ta khởi động lại apache để áp dụng các thay đổi:



Kế tiếp cài đặt gói php-cli

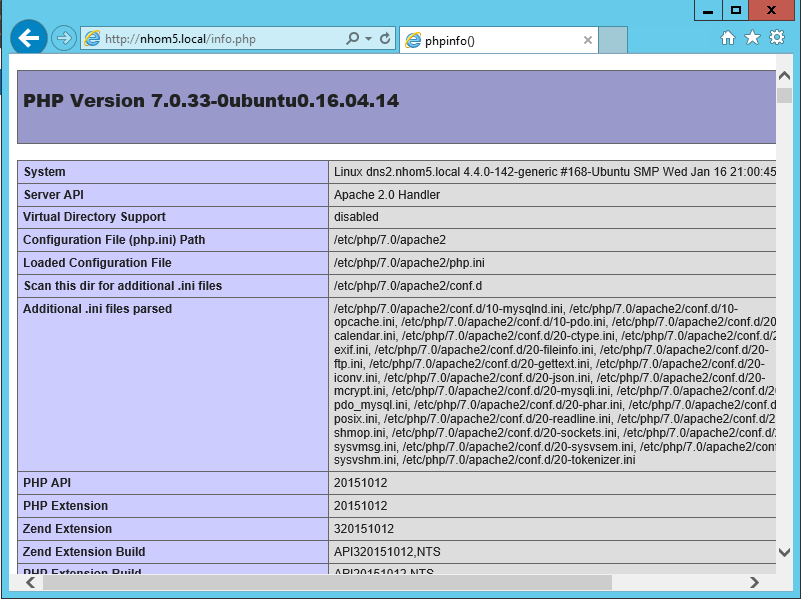


Cuối cùng tạo một file mã nguồn php để kiểm tra, ta đặt nó trong file **/var/www/html/info.php**



Mở trên một trình duyệt bất kì, nếu kết quả như hình dưới tức là ta đã cài đặt php thành công

Gõ trên thanh URL là <http://nhom5.local/info.php>

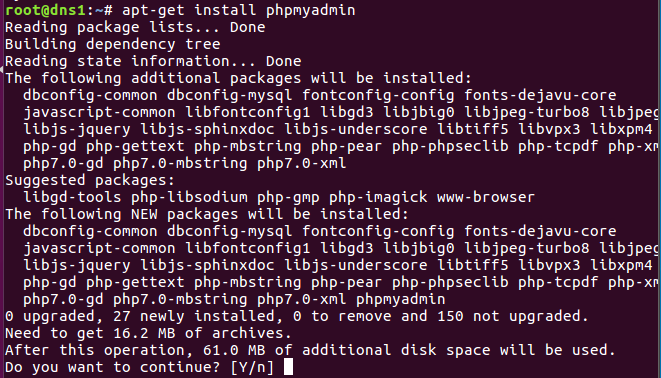


## Bước 11: cài đặt phpmyadmin

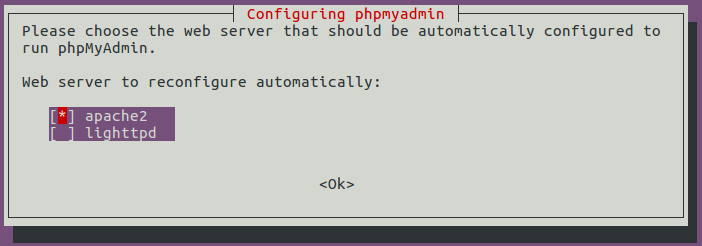
### Trên Server Master

Cài đặt PHPMyAdmin để giúp quản lý cơ sở dữ liệu MySQL của bạn từ trình duyệt web

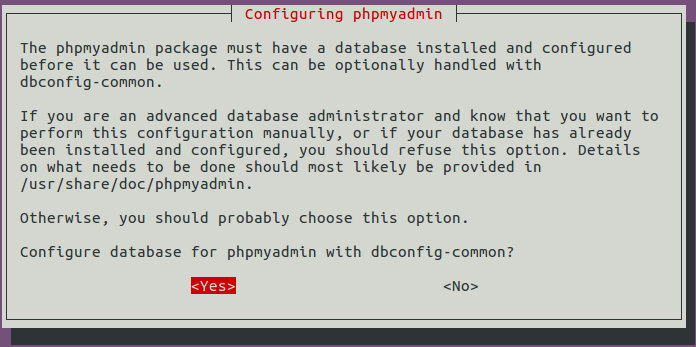
Việc cài đặt PHPMyadmin thông qua dòng lệnh: *sudo apt-get install phpmyadmin*



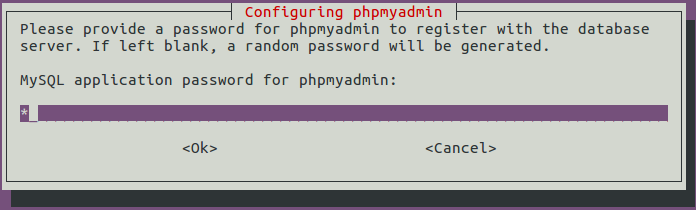
Bạn sẽ được yêu cầu chọn loại máy chủ web. Sự lựa chọn mặc định sẽ là apache2. Nhấn phím cách để chọn. Bây giờ nhấn phím Tab để tô sáng nút Ok và nhấn phím **enter**. Nhấn phím này rất quan trọng nếu không apache2 sẽ không được chọn. Khi apache2 được chọn sẽ có ký hiệu \* bên cạnh, hãy xem hình ảnh thứ hai trông như thế nào.



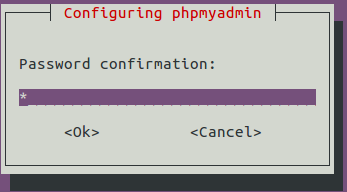
Sau đó, bạn sẽ được hỏi có nên định cấu hình cơ sở dữ liệu cho phpmyadmin với dbconfig-common không. Chọn Có và bấm vào.



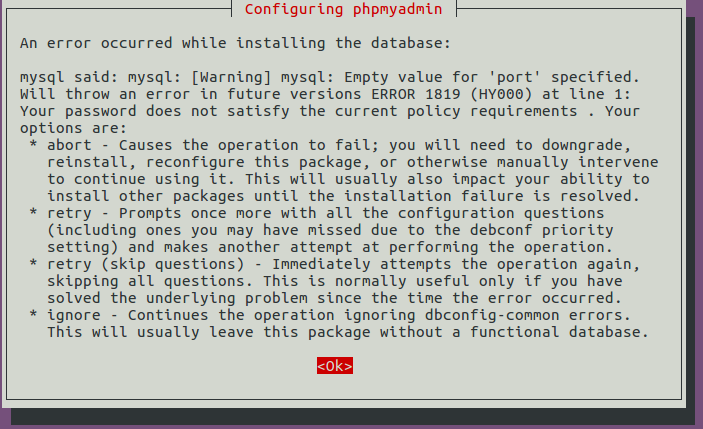
Sau đó, bạn sẽ được yêu cầu nhập mật khẩu của tài khoản gốc MySql. Nhập mật khẩu mà bạn đã nhập trong khi cài đặt MySql. Sử dụng phím Tab để tô sáng nút Ok và nhấn enter.



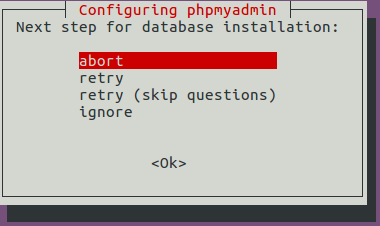
Làm tương tự khi yêu cầu xác nhận mật khẩu.



Ở bước này cứ nhấn enter để qua

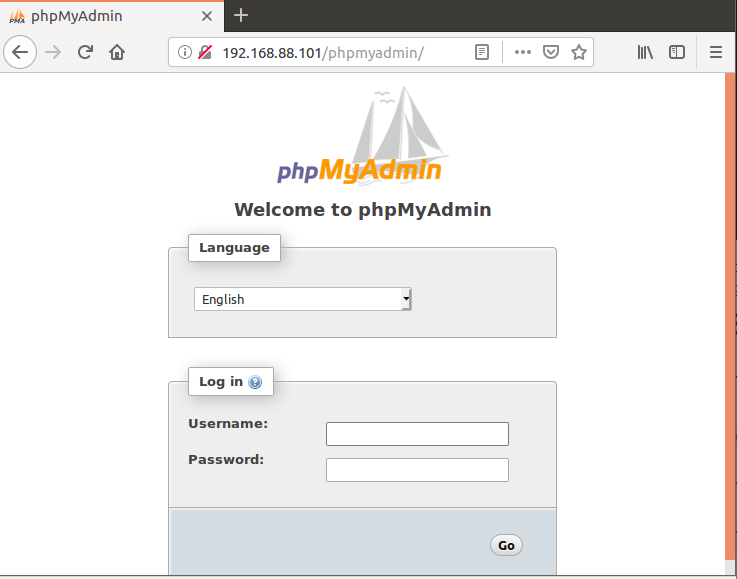


Ta chọn **abort** và **enter**

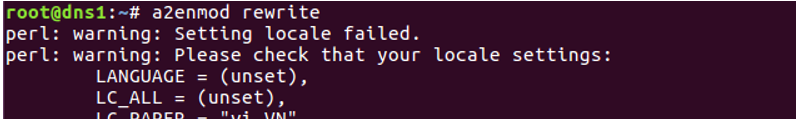


Cuối cùng, ta kiểm tra lại

Truy cập url <http://nhom5.local/phpmyadmin> trong trình duyệt của bạn. Nếu phpMyAdmin được cài đặt thành công, bạn sẽ thấy trang đăng nhập phpMyAdmin tiêu chuẩn. Nhập root làm tên người dùng và mật khẩu MySql bạn đã chọn trước đó để truy cập cơ sở dữ liệu.



Kích hoạt Mô-đun Viết lại trong Apache, để ứng dụng của bạn có thể sử dụng các URL đẹp thân thiện với người dùng hơn , chẳng hạn như http://website.com/posts/12/post-on-ub Ubuntu / (thay vì trang web http: //. com? post\_id = 12). Rất có thể bạn sẽ cần các URL đẹp trong tương lai, tốt hơn hãy kích hoạt nó ngay bây giờ.



Bây giờ ta khởi động lại Apache để xác áp dụng các thay đổi



Vậy, ta đã hoàn thành việc cài đặt cho ngăn xếp LAMP đầy đủ chức năng, bây giờ ta đã có thể viết một trang web riêng của chúng ta.

### Trên Server Slave

Ta làm các bước tương tự như máy Server Master, sau đó mở trình duyệt web gõ trên thanh URL như sau: <http://192.168.88.102/phpmyadmin>

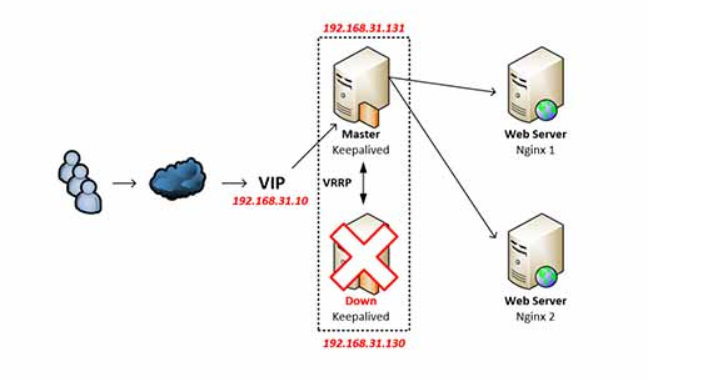
Nếu kết quả giống như hình dưới tức là ta đã cài đặt PHPMyAdmin thành công



## Bước 12: Cài đặt Keeepalived và Haproxy

### Trước khi thực hiện cấu hình, ta tìm hiểu keepalived và haproxy là gì?

Đối với những mô hình dịch vụ cần đảm bảo tính sẵn sàng cao (High Availability – HA), thì việc hệ thống bị down là không thể chấp nhận được. Hiện có rất nhiều phần mềm, giải pháp để đảm bảo tính HA cho các hệ thống nhưng ở đây sẽ sử dụng phần mềm đơn giản nhất làKeepalived với tính năng tự động chuyển đổi Virtual IP (VIP) giữa các máy chủ theo mô hình Active/Passive.



Keepalived là một chương trình dịch vụ trên Linux cung cấp khả năng tạo độ sẵn sàng cao (High Availability) cho hệ thống dịch vụ và khả năng cân bằng tải (Load Balancing) đơn giản. Với sự gọn nhẹ, tối ưu trong dịch vụ HA củaKeepalivedmang đến cho quản trị viên một giải pháp Active-Backup dịch vụ rất tốt.

Keepalived sẽ gom nhóm các máy chủ dịch vụ nào tham gia cụm HA, khởi tạo một Virtual Server đại diện cho một nhóm thiết bị đó với một Virtual IP (VIP) và một địa chỉ MAC vật lý của máy chủ dịch vụ đang giữ Virtual IP đó. Vào mỗi thời điểm nhất định, chỉ có một server dịch vụ dùng địa chỉ MAC này tương ứng Virtual IP. Khi có ARP request gởi tới virtual IP thì server dịch vụ đó sẽ trả về địa chỉ MAC này.

Các máy chủ dịch vụ sử dụng chung VIP phải liên lạc với nhau bằng địa chỉ *multicast 224.0.0.18* bằng giao thức VRRP. Các máy chủ sẽ có độ ưu tiên (priority) trong khoảng từ 1 – 254, và máy chủ nào có độ ưu tiên cao nhất sẽ thành Master, các máy chủ còn lại sẽ thành các Slave/Backup, hoạt động ở chế độ chờ.

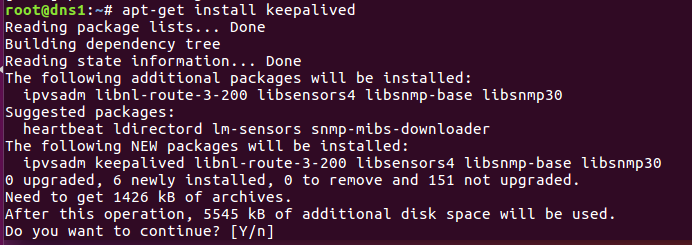
Nếu vì một sự cố gì đó mà các server BACKUP không nhận được các gói tin quảng bá từ MASTER trong một khoảng thời gian nhất định thì cả nhóm sẽ bầu ra một MASTER mới. MASTER mới này sẽ tiếp quản địa chỉ VIP của nhóm và gởi các gói tin ARP báo là nó đang giữ địa chỉ VIPnày. Khi MASTERcũ hoạt động bình thường trở lại thì server này có thể lại trở thành MASTER hoặc trở thành BACKUP tùy theo cấu hình độ ưu tiên của các router.

* Ở đây ta có thể hiểu nôm na, keepalived sẽ sử dụng một IP ảo để đặt trên interface, mấy nào được người quản trị đặt ưu tiên cao hơn sẽ trở thành Master và giữ IP ảo đí, khi máy Master hỏng, IP ảo sẽ được giao cho Backup.

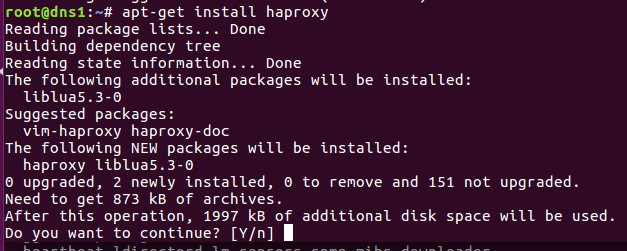
### Trên máy Server Master

Trước tiên ta cài đặt keepalived và haproxy

Sử dụng lệnh: *apt-get install keepalived* để cài đặt keepalived



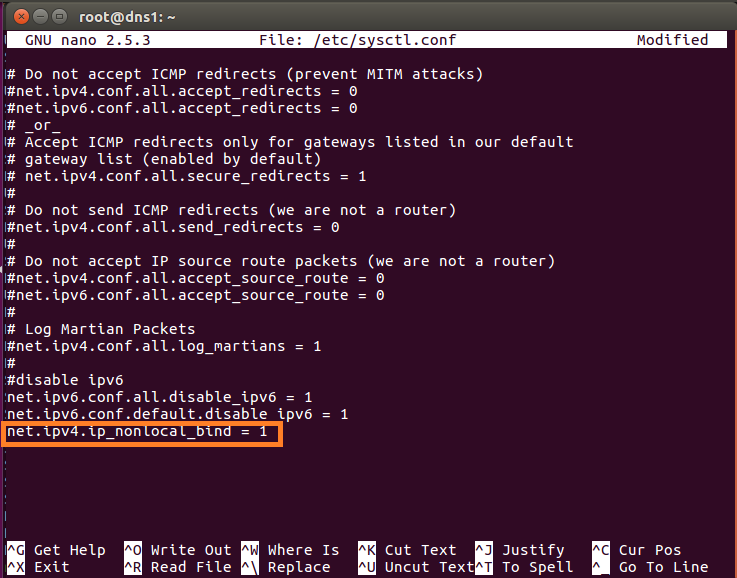
Và *apt-get install haproxy* để cài đặt haproxy



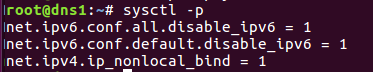
Dịch vụ Keepalived sẽ giúp chúng ta tạo 1 Virtual IP để dùng cho máy chủ, nói một cách nôm na là máy chủ sẽ sử dụng IP do chúng ta tự định nghĩa bằng Keepalived chứ không phải dùng IP trên interface của máy chủ (được cấp bởi 1 DHCP nào đó hay do chúng ta tự gán.). Để làm việc này, chúng ta cần vào file **/etc/sysctl.conf** để edit:



Thêm dòng **net.ipv4.ip\_nonlocal\_bind=1** vào cuối file và lưu lại



Sau đó các bạn lưu lại và thoát ra. Tiếp tục thực thi câu lệnh được gán vào bằng cách restart server hoặc chạy câu lệnh sau:

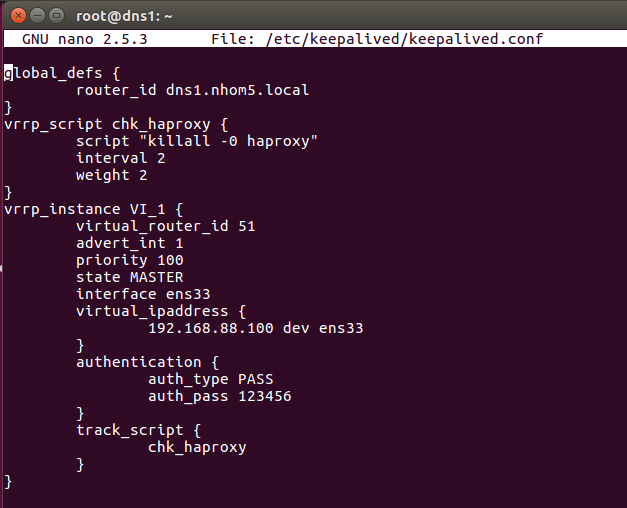


File config của Keepalived sẽ được lưu ở **/etc/keepalived/keepalived.conf**. Lưu ý nếu file chưa có sẵn thì tạo file mớ. Ta có thể dùng bất cứ trình biên soạn text nào trên Linux đều được. Ở đây mình sẽ dùng nano.



Điền nội dung sau vào file và lưu lại:

|  |
| --- |
| global\_defs {  router\_id dns1.nhom5.local #khai báo router\_id của keepalived  }  vrrp\_script chk\_haproxy {  script “killall -0 haproxy”  interval 2  weight 2  }  vrrp\_instance VI\_1 {  virtuak\_router\_id 51  advert\_int 1  prioriry 100  state MASTER  interface ens33 #thông tin trên interface của server  virtual\_ipaddress {  192.168.88.100 dev ens33 #khai báo virtual IP cho interface tương ứng  }  authentication {  auth\_type PASS  auth\_pas 123456 #passwork phải khai báo giống nhau giữa các keepalived  }  track\_script {  chk\_haproxy  }  } |



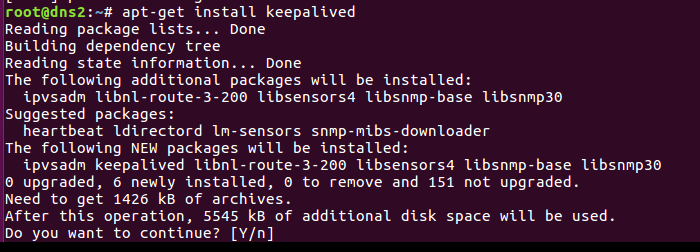
Sau đó dùng lệnh *service keepalived restart* để áp dụng các cài đặt:



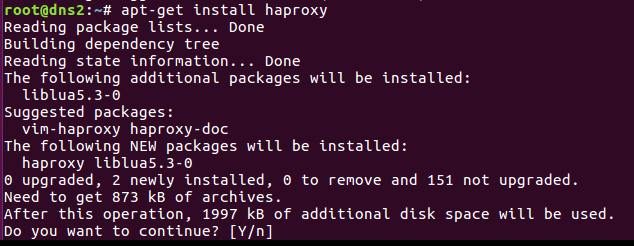
### Trên máy Server Slave

Tương tự trên Master, ta cũng cài keepalive và haproxy trên máy Slave:

Cài keepalived:



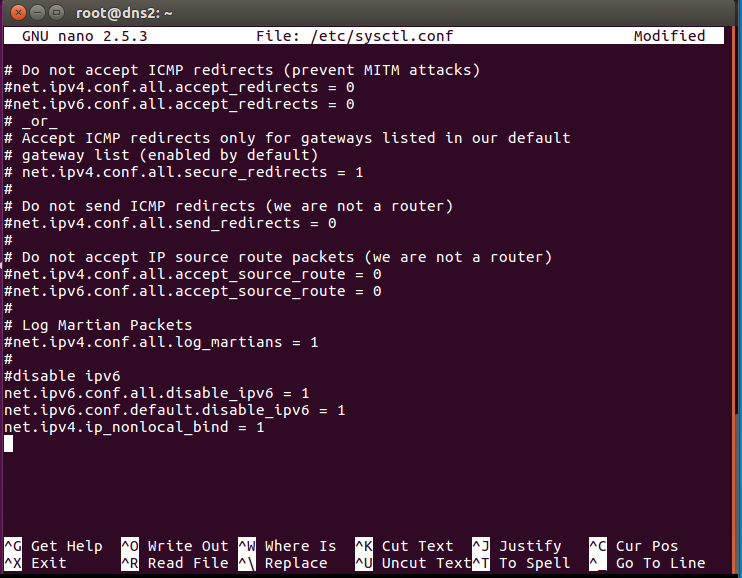
Cài haproxy:



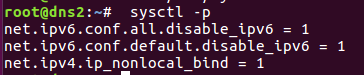
Tiếp theo cũng vào file /etc/sysctl.conf chỉnh sửa:



Thêm dòng **net.ipv4.ip\_nonlocal\_bind=1** vào cuối file và lưu lại file:



Tiếp tục thực thi câu lệnh được gán vào bằng cách restart server hoặc chạy câu lệnh sau:

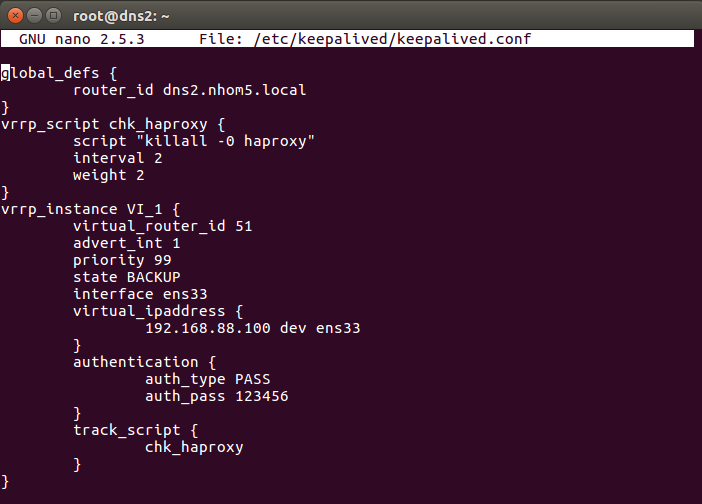


Tiếp theo ta config lại file keepalived bằng cáo vào file **/etc/keepalived/keepalived.conf**



Thêm các dòng sau vào file và lưu lại:

|  |
| --- |
| global\_defs {  router\_id dns2.nhom5.local #khai báo router\_id của keepalived  }  vrrp\_script chk\_haproxy {  script “killall -0 haproxy” #Check pid của dịch vụ haproxy có tồn tại không  interval 2 #Thời gian lặp đoạn Script đơn vị là second  weight 2 #Trognj số khấu trừ priority 2  }  vrrp\_instance VI\_1 {  virtuak\_router\_id 51  advert\_int 1  prioriry 99  state BACKUP  interface ens33 #thông tin trên interface của server  virtual\_ipaddress {  192.168.88.100 dev ens33 #khia báo virtual IP cho interface tương ứng  }  authentication {  auth\_type PASS  auth\_pas 123456 #passwork phải khai báo giống nhau giữa các keepalived  }  track\_script {  chk\_haproxy #Khai báo tên đoạn sript  }  } |



* Lưu ý ở 2 khai báo trên

|  |  |
| --- | --- |
| **Máy Master** | **Máy Slave** |
| vrrp\_instance VI\_1 {  virtuak\_router\_id 51  advert\_int 1  prioriry 100  state MASTER  interface ens33 #thông tin trên interface của server  virtual\_ipaddress {  192.168.88.100 dev ens33 #khia báo virtual IP cho interface tương ứng  } | vrrp\_instance VI\_1 {  virtuak\_router\_id 51  advert\_int 1  prioriry 99  state BACKUP  interface ens33 #thông tin trên interface của server  virtual\_ipaddress {  192.168.88.100 dev ens33 #khia báo virtual IP cho interface tương ứng  } |

Ta thấy khai báo với test1 state là MASTER , priority 100

Khác với test2 state BACKUP và priority 99

Khai báo trên có ý nghĩa khởi tạo ban đầu máy test1 sẽ là MASTER và được quyền giữ VIP 172.17.3.222 được gán vào interface ens33.

Tiếp theo với các khai báo vrrp\_script trên cả 2 máy:

|  |
| --- |
| vrrp\_script chk\_haproxy {  script “killall -0 haproxy” #Check pid của dịch vụ haproxy có tồn tại không  interval 2 #Thời gian lặp đoạn Script đơn vị là second  weight 2 #Trognj số khấu trừ priority 2  }  track\_script {  chk\_haproxy #Khai báo tên đoạn sript  } |

Trong file cấu hình Keepalived trên của 2 máy dns1 và dns2 đều có track\_script - có nghĩa là cả 2 server đều sẽ tiến hành chạy đoạn script check trạng thái của dịch vụ process ID (PID) được khai báo trong script ( ở đây khai báo check service haproxy). Giả sử service haproxy trên dns1 (được khởi tạo ban đầu với state Master và priority=100 ) vì một lý do nào đó không hoạt động, Keepalived sẽ trừ trọng số (priority 100-2=98) trên máy dns1. Lúc này priority của dns1 sẽ là 98 và nhỏ hơn priority=99 được khai báo ban đầu ở dns2, do đó Keepalived sẽ chuyển trạng thái của dns2 từ BACKUP thành MASTER và dns2 sẽ được giữ VIP đã khai báo.

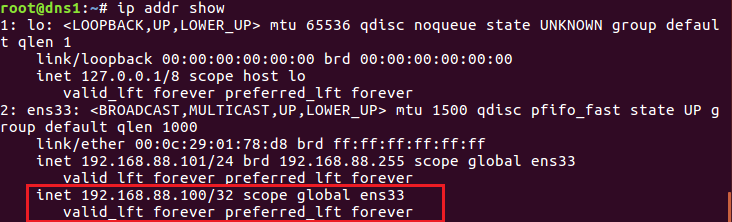
Tiếp đến ta dùng lệnh *service keepalived start* để áp dụng các thay đổi:



### Kiểm tra

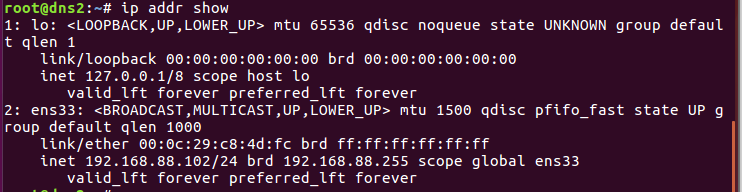
Check lại interface bằng cách gõ lệnh: *ip addr show*

Trên máy Master ta thấy dòng inet 192.168.88.100/32 scope global ens33 hiện lên tức là đã cài đặt thành công trên máy Master.



Còn với máy Slave, nó sẽ không có dòng trên vì đang không giữ VIP.

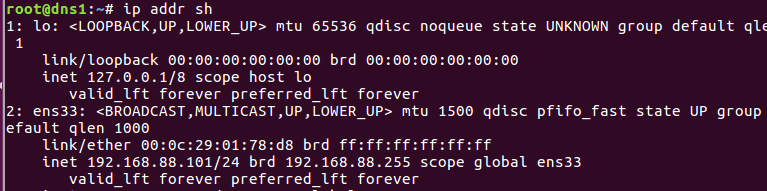
Virtual IP chỉ được gán vào interface ens33 của máy dns1 chứ không được gán vào máy dns2 dù ta đã khai báo trên cả 2 máy. Vì hiện tại khởi tạo ban đầu máy dns1 giữ vai trò Master.



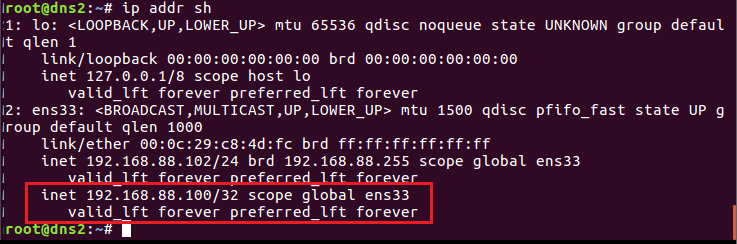
Kiểm tra xem khi dừng haproxy trên máy Master thì sẽ chuyển VIP sang máy Slave hay không bằng cách tắt haproxy trên máy Master: *service haproxy stop*



Sau đó dùng lệnh lại trên máy Master: *ip addr sh* và ta thấy trên máy Master không còn giữ VIP nữa.



Lúc này ta thấy trên máy Slave đã giữ VIP



Vậy việc cấu hình keepalived đã thành công.

### Cấu hình haproxy

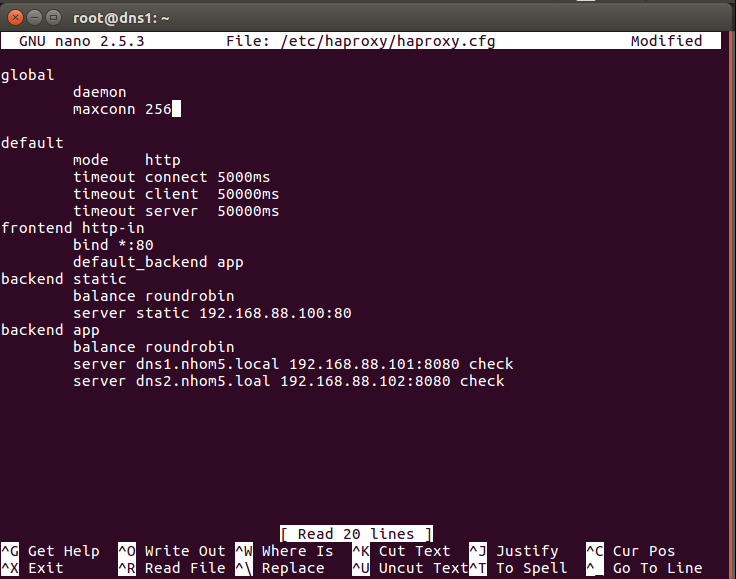
Ở đây ta thực hiện song song cho 2 máy:

Ta tiến hành chỉnh sửa file config của haproxy bằng lệnh *nano /etc/haproxy/haproxy.cfg*

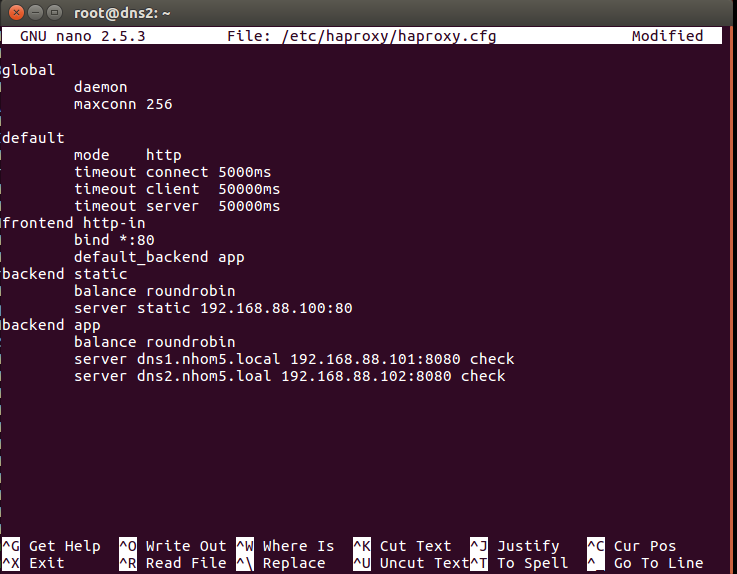


Ta thực hiện chỉnh sửa file như hình dưới:

Lưu ý: Chuyển listen port của apache2 thành port khác. Vì việc cài tất cả các dịch vụ haproxy và http đều sử dụng port 80 trên cùng 1 máy nên sẽ bị conflict. Cụ thể ở đây mình sẽ chuyển default listen port của apache thành 8080



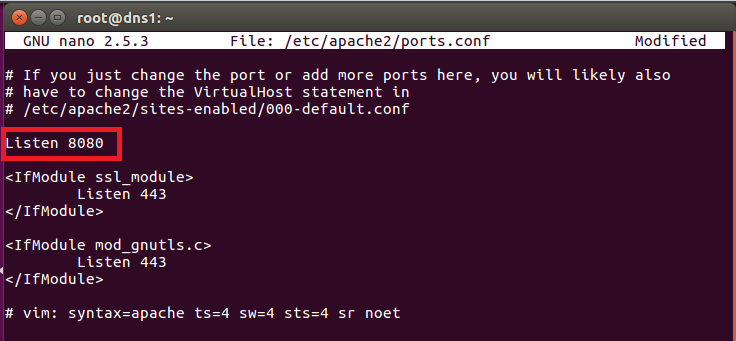
Làm tương tự trên máy Slave, vào file */etc/haproxy/haproxy.cfg* và chỉnh sửa như hình dưới:



Chỉnh sửa lần lượt trên cả 2 server . Edit cả 2 file sau trên mỗi server **sudo nano /etc/apache2/ports.conf** và **sudo nano /etc/apache2/sites-available/000-default.conf**

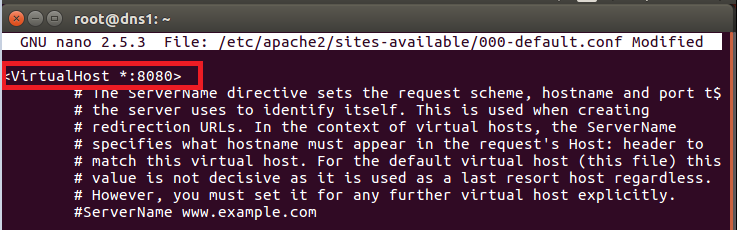


Chỉnh sửa Listen thành 8080





Và chỉnh VirtualHost thành 8080



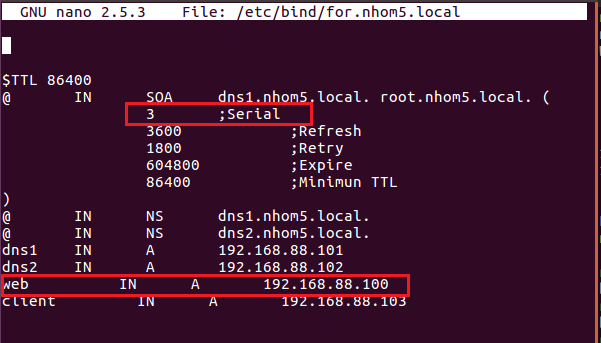
Việc này cũng thực hiện tương tự trên máy Slave

### Tạo tên miền cho IP 192.168.88.100

Trước tiên ta vào file /etc/bind/for.nhom5.local ở máy Master để chỉnh sửa forward zone



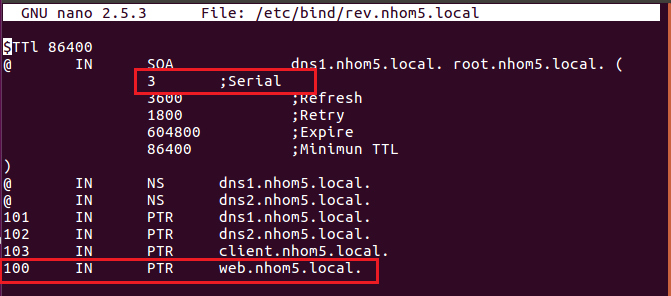
Ta thêm *web IN A 192.168.88.100* vào file và lưu ý chỉnh sửa Serial thay đổi, vì máy Slave sẽ kiểm tra nếu Serial không đổi sẽ không nhận thêm dữ liệu.



Tiếp theo vào file */etc/bind/rev.nhom5.local* để chỉnh sửa thêm **reverse zone**



Ta thêm dòng *100 IN PTR web.nhom5.local. và chỉnh sửa lại Serial cho phù hợp:*



Sau đó ta lần lượt restart lại bind9 trên máy Master và Slave:





Kiểm tra xem tên miền đã có thể sử dụng chưa, ta dùng lệnh *nslookup web.nhom5.local* trên máy **client**

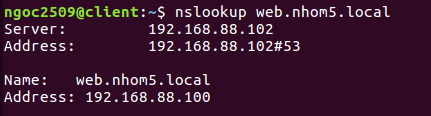
Khi mở cả 2 máy Master và Slave:

Ta thấy, máy Master sẽ có nhiệm vụ phân giải tên miền *web.nhom5.local* đúng với IP 192.168.88.100 đã đặt trước



Khi tắt máy Master đi:

Ta thấy lúc này máy Slave có nhiệm vụ phân giải tên miền *web.nhom5.local* đúng với IP 192.168.88.100 đã đặt trước



Vậy chứng tỏ với tên miền cho IP ảo 192.168.88.100 là web.nhom5.local đã có thể sử dụng được.

Ở bước tạo Keepalived và haproxy coi thư đã hoàn thành, sau đây ta cần thực hiện việc backup cuối cùng là backup sữ liệu (source code) của một trang web, điều này cũng tương đối quan trong để vận hành một trang web vẫn cung cấp bình thường cho người sử dụng nếu có bị hỏng ở bên một máy Master.

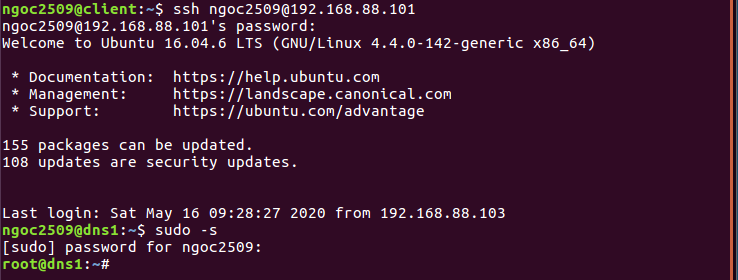
## Bước 13: Đồng bộ source code của web

Đây cũng là một dạng backup source code của một trang web, khác với việc backup CSDL cần tự động do CSDL có thể thường xuyên thay đổi.

Với việc backup một source code của một trang web ta chỉ việc làm thủ công do source code ít khi nào bị thay đổi, trong một số trường hợp người quản trị có thể thêm sửa một số chi tiết nhỏ và sử dụng được trong thời gian dài, do vậy việc backup source code ta cũng có thể làm đơn giản hơn, nó gần giống như copy một trang web sang máy khác và khi cần chỉnh sửa thì chỉnh sửa lại trên tất cả máy, tuy nhiên sử dụng cách dưới đây sẽ có thêm một số tính năng cho việc backup.

Do vậy việc backup source code ta thường không dùng tự động backup theo định kì, chỉ chỉnh sửa khi nào người quản trị muốn thay đổi.

Lưu ý là đăng nhập ssh kết nối đến Server Master ở user root:

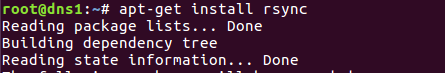


Điều qua trọng là cần cung cấp quyền thực thi trên cả 2 file nguồn và đích, ở đây nhóm chọn file nguồn là /var/ww/html/ ở máy Master và file đích là /var/www/ ở máy Slave.





Trước tiên ta cài đặt rsync trên cả 2 máy Master và Slave:





Trước tiên ta nên xóa một số file mặc đinh trong máy Slave để tranh xung đột dữ liệu



Cú pháp chung của rsync là: #rsync option source destination

**Source**: thư mục chứa dữ liệu gốc muốn đồng bộ, nơi truyền dữ liệu.

**destination**: nơi sẽ chứa dữ liệu đồng bộ đến, nơi nhận dữ liệu.

**option**: các tham số để tùy biến rsync khi đồng bộ dữ liệu.

Trên máy Master dùng dòng lệnh để đồng bộ file sang máy Slave:

*Srync -azvhPe ssh –delete /var/www/html ngoc2509@192.168.88.102:/var/www*

**Trong đó**:

-a option này sẽ bảo toàn user, group, permission,symbolic link của dữ liệu

-v: show trạng thái truyền tải file ra màn hình để bạn theo dõi.

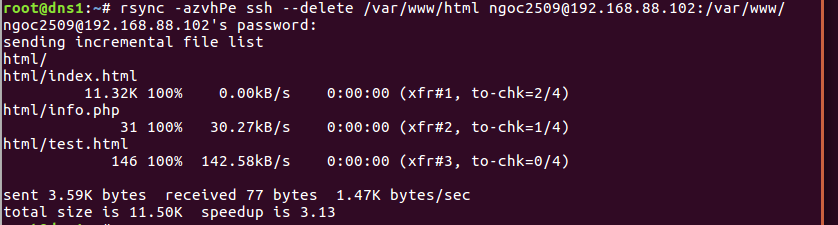
-h: kết hợp với -v để định dạng dữ liệu show ra dễ nhìn hơn.

-z: nén dữ liệu trước khi truyền đi giúp tăng tốc quá trình đồng bộ file.

-e: sử dụng giao thức SSH để mã hóa dữ liệu.

-P: Option này dùng khi đường truyền không ổn định, nó sẽ gửi tiếp các file chưa được gửi đi khi có kết nối trở lại.

--delete: xóa dữ liệu ở destination nếu source không tồn tại dữ liệu đó.



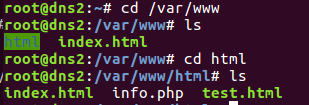
Cuối cùng ta kiểm tra lại việc backup đã thành công chưa:

Vào folder **/var/www/html** trên Master và xem các file trong đó



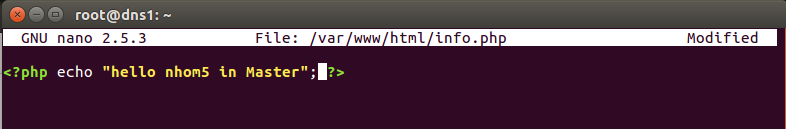
Tiếp theo trên máy Slave, vào folder **/var/www**, ta thấy đã có hiển thị thêm folder **html**

Tiếp tục dùng lệnh cd để vào folder html và ls nội dung trong đó, ta thấy có đúng các file trên máy Master, vậy chứng tỏ việc backup dữ liệu đã thành công.

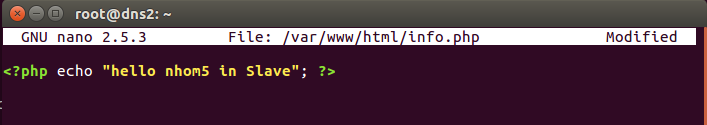


## Bước 14: Kiểm tra cuối cùng

Ta có thông tin file info.php với đường dẫn /var/www/html/info.php như trên máy Master như sau:



Và trên máy Slave như sau:



Ta mở trên một trình duyệt bất kì và nhập đường dẫn **web.nhom5.local/info.php**.

Khi mở cùng 2 máy:

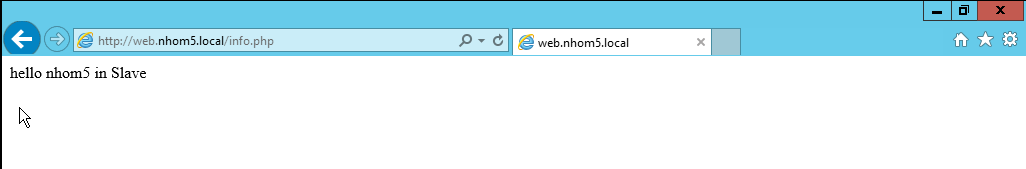
Lúc này VIP đang ở máy Master nên nó sẽ lấy thông tin trong file info.php của máy Master.



Khi tắt máy Master: VIP lúc này thuộc về máy Slave nên nó sẽ lấy thông tin trong file info.php của Slave, và máy Slave đã chuyển thành Master.



Khi bật lại máy Master: VIP lúc này vẫn thuộc về Slave, nên nó sẽ lấy thông tin trong file info.php của Slave



Thông qua việc kiểm tra ta thấy việc cấu hình keepalived và haproxy đã thành công.

# Kết quả và kết luận

## Kết quả

Link video kết quả: <https://drive.google.com/file/d/1PChvY50MzwTm3c-3lJIqmGhJLIT2gAqG/view?usp=sharing>

## Kết luận

Về việc cấu hình **DNS Server** ta cần lưu ý nhất là Forward Zone và Reverse Zone

Trong đó Forward Zone có chức năng chỉ từ tên miền sang IP, và ngược lại thì Reverse Zone chỉ từ IP sang tên miền.

Hai file đó được viết ở máy Master, nếu thay đổi hoặc thêm một tên miền nào đó, ta cần thay đổi luôn số Serial để bên các máy Slave biết được có thông tin cần cập nhật.

Đối với **Web Server**, cần cài LAMP Stack (Apache, MySQL, PHP) được hỗ trợ cho web trên ubuntu (Linux). Khi tạo một trang web cho một doanh nghiệp hay một tổ chức nào đó, ta cần lưu ý về việc backup trang web trường hợp máy Server bị hỏng.

Việc backup trước tiên quan trong nhất chính là cơ sở dữ liệu cho một trang web, do CSDL có thể thường xuyên thay đổi nên cần backup dạng tự động để khi mỗi lần chỉnh sửa thì nó cũng sẽ tự động cập nhật.

Kế tiếp việc backup source code của một trang web cũng khá quan trọng, tuy nhiên source code thì ít thay đổi, do đó nếu muốn thay đổi hay cập nhật ta cũng có thể làm các bước thủ công.

Tiếp đến là việc duy trì một trang web vẫn truy cập được nếu có lỗi là dựa vào việc sử dụng một IP ảo, và trỏ vào máy Master, nếu không may máy bị hỏng IP đó sẽ được trỏ về các máy Slave, đồng thời trên các máy Slave cũng có trang web tương tự như Master. Do đó nếu bị lỗi cũng không lo lắng về việc mất truy cập đến trang web của chúng ta.

Về **sự ảnh hưởng của DNS Server đối với Web Server,** DNS giúp việc tìm kiếm một trang web được tại ra dễ dàng và dễ nhớ hơn. Khi một trang web vừa được tạo, việc truy cập đến nó chỉ có thể bằng một địa chỉ IP í thân thiên. Do vậy DNS cũng rất quan trọng và cần thiết trong việc tạo lập và phát triển trang một trang web.

# Tài liệu kham khảo

[1] <https://www.thegeekstuff.com/2014/01/install-dns-server/>

[2] <https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-16-04>

[3] <https://www.alibabacloud.com/blog/how-to-configure-mysql-master-slave-replication-on-ubuntu-16-04_593982>

[4] <https://www.digitalocean.com/community/tutorials/how-to-set-up-highly-available-haproxy-servers-with-keepalived-and-floating-ips-on-ubuntu-14-04>

[5] <https://www.server-world.info/en/note?os=Ubuntu_16.04&p=rsync>