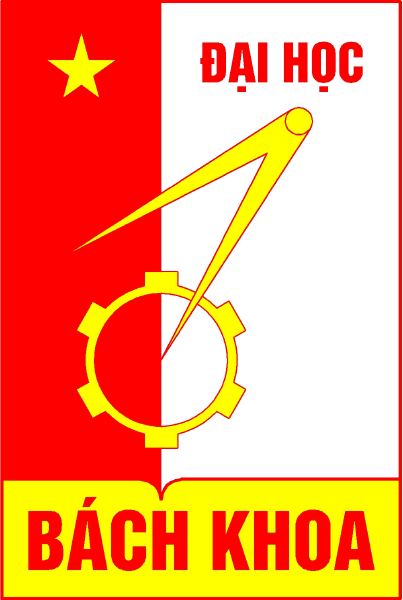
Trường Đại học Bách khoa Hà Nội

Viện Công Nghệ Thông Tin- Truyền Thông



**Báo cáo Linux và Phần mềm mã nguồn mở**

*Đề tài: Tìm hiểu Nginx và MySQL. Tạo trang Web sử dụng Apache 2.0 và MySQL*

**Giảng viên hướng dẫn**: Trần Hải Anh

**Mã học phần**: IT 3110

**Mã lớp**: 101894

**Sinh viên thực hiện:**

* Nguyễn Danh Nam - 20166477
* Lưu Công An - 20165710
* Lê Văn Song - 20153170
* Hoàng Nông Lương - 20156000

**Hà Nội, tháng 5 năm 2018**

*Hà Nội, ngày 30 tháng 04 năm 2018*

Mục Lục

[Lời nói đầu 3](#_Toc513842276)

[Phân công công việc 3](#_Toc513842277)

[Chương 1: Tìm hiểu về Nginx và cách cài đặt 4](#_Toc513842278)

[1.1 Tìm hiểu về Nginx 4](#_Toc513842279)

[1.2.Cài đặt Nginx 6](#_Toc513842280)

[1.3 Giới thiệu module Http trong Nginx 6](#_Toc513842281)

[1.4 Các thông số của Nginx 8](#_Toc513842282)

[1.5 Ẩn phiên bản của Nginx 9](#_Toc513842283)

[1.6 Cấm các truy cập tới các tệp tin ẩn trên Nginx 9](#_Toc513842284)

[1.7 Tính năng của Nginx 9](#_Toc513842285)

[1.8 Cấu hình cho Nginx 10](#_Toc513842286)

[1.9 So sánh giữa apache và Nginx 12](#_Toc513842287)

[Chương 2: Tìm hiểu và cài đặt MySQL 14](#_Toc513842288)

[2.1 MySQL là gì? 14](#_Toc513842289)

[2.2 Ưu điểm của MySQL 15](#_Toc513842290)

[2.3 Cách cài đặt MySQL trên hệ điều hành Linux 15](#_Toc513842291)

[2.4 Chạy demo một số câu lệnh MySQL trên terminal của hệ điều hành Linux 17](#_Toc513842292)

[Chương 3: Lập trình Web với Apache, PHP và MySQL trên Linux 18](#_Toc513842293)

[3.1 Giới thiệu về Apache và PHP 18](#_Toc513842294)

[3.2 Chạy một trang Web đơn giản với Apache, MySQL, PHP 21](#_Toc513842295)

[Kết luận 28](#_Toc513842296)

[Tài liệu tham khảo 29](#_Toc513842297)

# **Lời nói đầu**

Trong bối cảnh nền kinh tế thế giới đang có xu hướng như hiện nay, vấn đề bản quyền là vấn đề nổi cộm. Ở Việt Nam, vấn đề này rất phổ biến về vấn đề sử dụng phần mềm không có bản quyền. Chính vì thế, phần mềm mã nguồn mở được xem như là một giải pháp hữu hiệu cho bài toán bản quyền ở nước ta hiện nay. Phần mềm mã nguồn mở một mặt có chi phí rẻ so với các phần mềm truyền thống, dễ nâng cấp, cải tiến. Chính vì thế, phát triển phần mềm mã nguồn mở chúng ta có thể tận dụng được những công nghệ tiên tiến có sẵn trên thế giới, cải tiến cho phù hợp với người Việt Nam, tiết kiệm được nhiều công sức và Linux là một hệ điều hành mã nguồn mở như vậy.

Nginx và Apache là 2 open source web server phổ biến hiện nay. Chúng phục vụ hơn 50% lượng traffic trên mạng Internet. Đồng thời MySQL là hệ quản trị cơ sở dữ liệu tự do mã nguồn mở phổ biến nhất trên thế giới và được các nhà phát triển ưa chuộng trong quá trình phát triển ứng dụng bởi những tính năng ưu việt của nó như: tốc độ cao, ổn định, dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành. Chính vì thế nhóm em đã chọn đề tài : Tìm hiểu Nginx và MySQL. Tạo trang Web sử dụng Apache 2.0 và MySQL” để hiểu rõ hơn về Nginx, MySQL, cũng như cách tạo một trang Web sử dụng Apache và MySQL như thế nào.

# **Phân công công việc**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Họ tên** | **MSSV** | **Công việc đóng góp** | **Mức độ hoàn thành** |
| 1 | Nguyễn Danh Nam | 20166477 | * Cài đặt Apache+ PHP+ MySQL để chạy thử nghiệm web * Lập trình web * Chỉnh sửa Slide | 99% |
| 2 | Lưu Công An | 20165710 | * Tìm hiểu Nginx * Giới thiệu qua và cài đặt Apache, PHP | 99% |
| 3 | Hoàng Nông Lương | 20156000 | * Tìm hiểu Nginx và cài đặt Nginx | 99% |
| 4 | Lê Văn Song | 20153170 | * Tìm hiểu MySQL * Hỗ trợ phần lập trình web * Tổng hợp và chỉnh sửa báo cáo, slide | 99% |

# **Chương 1: Tìm hiểu về Nginx và cách cài đặt**

## **1.1 Tìm hiểu về Nginx**

#### 1.1.1 Nginx là gì?

Nginx (được đọc là “Engine-X”)  là phần mềm mã nguồn mở cho web server và là 1 máy chủ reverse proxy mã nguồn mở cho các giao thức HTTP, HTTPS, SMTP, POP3 và IMAP, cũng như là 1 máy chủ cân bằng tải (load balancer), HTTP cache và web. Nhằm nâng cao hiệu xuất xử lý khi sử dung lượng RAM thấp. Dự án Nginx được bắt đầu với việc tập trung vào tính đồng thời cao, hiệu năng cao và sử dụng tài nguyên thấp và được phát triển bởi Igor Sysoev vào nằm 2002, được phân phối ra công chúng lần đầu vào nằm 2004 và chạy trên nên tảng Unix , Linux  và các biến thểBSD, Mac OS X, Solaris, AIX, HP-UX và Microsoft Windows.

#### 1.1.2 Đặc điểm của Nginx

Nginx có thể triển khai nội dung của các trang web động bằng cách sử lý FastCGI, SCGI  cho các scripts . Và có thể sử dụng như là một server cân bằng tải . Sau đó vấn đề C10K xuất hiện  nói cách khác để cho phép mỗi máy chủ web phải có khả năng xử lý 10.000 khách hàng cùng một lúc.  Cần phải phát triển một mạng lưới  I / O tốt hơn và công nghệ quản lý chủ đề đã được xuất hiện. Sự xuất hiện của NGinx không phải là kết quả của một nỗ lực để giải quyết vấn đề C10K (như là một vấn đề phổ biến) nhưng “vấn đề C10K” đã thành công trong việc đưa ra các  nỗ lực để nâng cao hiệu suất phát triển mạng máy chủ.

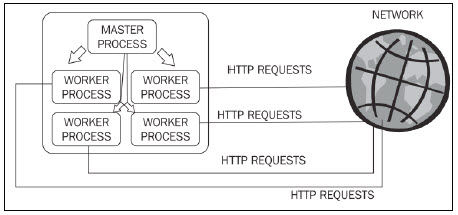
Không giống với các máy chủ web truyền thống, Nginx không dựa trên luồn (thread) để xử lý yêu cầu. Thay vào đó, nó sử dụng 1 kiến trúc bất đồng bộ hướng sự kiện linh hoạt . Kiến trúc này sử dụng ít, nhưng quan trọng hợn, là lượng bộ nhớ có thể dự đoán khi hoạt động. Đây chính là điểm mấu chốt khiến Nginx là 1 trong số ít những máy chủ được viết để giải quyết vấn đề C10K

Igor Sysoev phát triển nginx từ cách đây hơn 9 năm. Vào tháng 10/2004, phiên bản 0.1.0 được phát hành rộng rãi theo giấy phép BSD. Công dụng của nginx ngoài máy chủ web, còn có thể làm proxy nghịch cho Web và làm proxy email (SMTP/POP3/IMAP). Theo thống kê của Netcraft, trong số 1 triệu website lớn nhất thế giới, có 6,52% sử dụng nginx. Tại Nga, quê hương của nginx, có đến 46,9% sử dụng máy chủ này. Nginx chỉ đứng sau Apache và IIS (của Microsoft).

Nginx cung cấp gần như tất cả các chức năng máy chủ web.

Hiện nay, có khoảng 14,72 % (hơn 137 triệu) các website trên Internet đang sử dụng Nginx là máy chủ web.

#### 1.1.3 Kiến trúc của Nginx

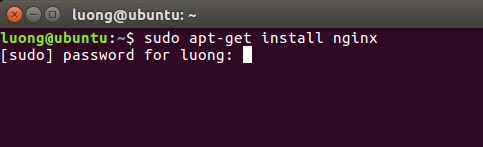
 Khi được khởi chạy service, nginx khởi tạo mọt tiến trình chủ và cũng là tiến trình duy nhất tồn tại trong bộ nhớ Master Process. Tiến trình này không chịu trách nhiệm tự xử lý bất kỳ request nào từ phía client mà thay vào đó nó sinh ra các tiến trình con gọi là Worker Process để xử lý các request này.

Để định nghĩa cho các Worker Process này, chúng ta cần sử dụng tệp tin cấu hình để xác định số tiến trình, số lượng kết nối , tài khoản và nhóm tài khoản mà mỗi Worker Process chạy.

## **1.2.Cài đặt Nginx**

Để cài đặt Nginx bạn cần chạy câu lệnh:

*sudo apt-get install nginx*



Sau khi gõ câu lệnh trên thì các package sẽ được cài:

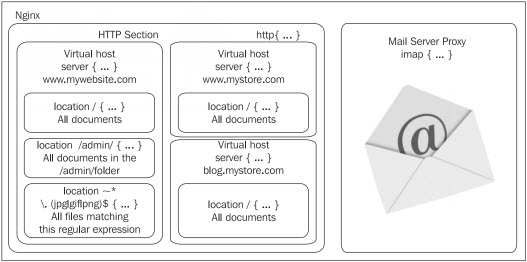


## **1.3 Giới thiệu module Http trong Nginx**

Module HTTP Core là thành phần chứa tất cả các khối, chỉ thị và các biến cơ bản của máy chủ HTTP. Mặc định module này được cài đặt trong khi biên dịch, nhưng không được bật lên khi Nginx chạy, việc sử dụng module này là không bắt buộc.

Module này là 1 trong những module tiêu chuẩn lớn nhất của Nginx – nó cung cấp 1 số lượng lớn các chỉ thị và biến. Để có thể hiểu được tất cả các yếu tố này và vai trò của chúng, chúng ta sẽ bắt tay vào tìm hiểu 3 khối chỉ thị chính – http, server và location.

* **http** : được khai báo ở phần đầu của tập tin cấu hình. Nó cho phép chúng ta định nghĩa các chỉ thị và các khối từ tất cả các module liên quan đến khía cạnh HTTP của Nginx. Khối chỉ thị này có thể được khai báo nhiều lần trong tập tin cấu hình, và các giá trị chỉ thị được chèn trong khối http sau sẽ ghi đè lên các chỉ thị nằm trong khối http trước đó
* **server** : khối này cho phép chúng ta khai báo 1 website. Nói cách khác, 1 website cụ thể (được nhận diện bởi 1 hoặc nhiều hostname) được thừa nhận bới Nginx và nhận cấu hình của chính nó. Khối này chỉ có thể được dùng bên trong khối http.
* **location** : cho phép chúng ta định nghĩa 1 nhóm các thiết lập được áp dụng cho 1 vị trí cụ thể trên website (thể hiện qua URL của website đó). Khối location có thể được dùng bên trong 1 khối server hoặc nằm chồng bên trong 1 khối location khác.



Trong biểu đồ trên, khu vực HTTP, được định nghĩa bởi khối http, bao quanh toàn bộ các cấu hình liên quan đến web. Nó cũng chứa 1 hoặc nhiều khối server, định nghĩa các tên miền của các website mà chúng ta có. Với mỗi website này, chúng ta có thể định nghĩa nhiều khối location mà cho phép chúng ta áp dụng các thiết lập bổ sung đến 1 URI yêu cầu cụ thể của website hoặc các URI yêu cầu khớp 1 mẫu nào đó.

## **1.4 Các thông số của Nginx**

#### 1.4.1 worker\_process

Với cấu hình mặc định, Nginx sẽ sử dụng một CPU để xử lý các tác vụ của mình. Tùy theo mức độ hoạt động của web server mà chúng ta có thể thay đổi lại thiết lập này. Ví dụ với các web server hay sử dụng về SSL, gzip thì ta nên đặt chỉ số của **worker\_processes** này lên cao hơn. Nếu website của bạn có số lượng các tệp tin tĩnh nhiều, và dung lượng của chúng lớn hơn bộ nhớ RAM thì việc tăng **worker\_processes** sẽ tối ưu băng thông đĩa của hệ thống.

#### 1.4.2 worker\_connections

**Worker\_connections** sẽ cho biết số lượng connection mà CPU sẽ xử lý. Mặc định, số lượng connection này được thiết lập là 1024. Để xem về mức giới hạn sử dụng của hệ thống bạn có thể dụng lệnh ulimit. Con số thiết lập của worker\_connections nên nhỏ hơn hoặc bằng giới hạn này! Nếu bạn đã điều chỉnh lại giá trị worker\_processes giúp Nginx sử dụng nhiều cores để xử lý các tác vụ hơn thì có thể thêm dòng cấu hình sau để tăng số lượng clients lên cao nhất.

#### 1.4.3 buffers

Một trong những cấu hình quan trọng để tối ưu Nginx là thiết đặt các giá trị buffer. Nếu bạn thiết lập bộ nhớ buffer quá nhỏ thì sẽ dễ dẫn tới tình trạng “thắt cỗ chai” khi web server của chúng ta tiếp nhận một lượng traffic lớn. Để thay đổi các giá trị buffer này, chúng ta có thể thêm vào các dòng cấu hình ở thẻ http của file cấu hình chính nginx.conf

Trong đó:

* **client\_body\_buffer\_size**: Thiết đặt giá trị kích thước của body mà client yêu cầu. Nếu kích thước được yêu cầu lớn hơn giá trị buffer thì sẽ được lưu vào temporary file.
* **client\_header\_buffer\_size**: Thiết đặt giá trị kích thước của header mà client yêu cầu. Thông thường thì kích thước này 1K là đủ.
* **client\_max\_body\_size**: Thiết đặt giá trị kích thước tối đa của body mà client có thể yêu cầu được, xác định bởi dòng Conent-Length trong header. Nếu kích thước body yêu cầu vượt giới hạn nãy thì client sẽ nhận được thông báo lỗi “Request Entity Too Large” (413).
* **large\_client\_header\_buffers**: Thiết đặt giá trị kích về số lượng và kích thước lớn nhất của buffer dùng để đọc các headers có kích thước lớn từ các request của client. Nếu client gửi một header quá lớn Nginx sẽ trả về lỗi “Request URL too large” (414)hoặc “Bad request” (400)nếu header của request quá dài.

## **1.5 Ẩn phiên bản của Nginx**

Việc ẩn đi phiên bản của Nginx từ Server Header sẽ giúp hệ thống webserver của chúng ta được bảo mật tốt hơn. Để thực hiện điều này, trong thẻ http của của tệp tin cấu hình chính nginx.conf ta thêm vào dòng sau *server\_tokens off;*

## **1.6 Cấm các truy cập tới các tệp tin ẩn trên Nginx**

Đôi khi trên các thư mục web chúng ta có lưu những tệp tin ẩn (bắt đầu với dấu chấm “.”) như .svn, .htaccess. Đây là các tệp tin không mang tính public đối với người dùng.

Để ngăn chặn các truy xuất tới các tệp tin ẩn này ta có thể thêm vào đoạn cấu hình sau

***location ~ /\. {***

***access\_log off;***

***log\_not\_found off;***

***deny all;***

***}***

## **1.7 Tính năng của Nginx**

**Những tính năng của máy chủ HTTP Nginx**

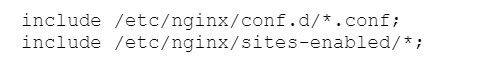
* Có khả năng xử lý hơn 10.000 kết nối cùng lúc với bộ nhớ thấp
* Phục vụ tập tin tĩnh (static files) và [**lập chỉ mục**](http://blog-xtraffic.pep.vn/thiet-ke-phat-trien-website-thien-voi-cong-cu-tim-kiem-lap-chi-muc/) tập tinTăng tốc proxy ngược bằng bộ nhớ đệm (cache); cân bằng tải đơn giản và khả năng chịu lỗi
* Hỗ trợ tăng tốc với bộ nhớ đệm của FastCGI, uwsgi, SCGI, và các máy chủ memcached
* Kiến trúc modular; tăng tốc độ nạp trang bằng nén gzip tự động
* Hỗ trợ mã hoá SSL và TLS
* Cấu hình linh hoạt; lưu lại nhật ký truy vấn
* Chuyển hướng lỗi 3XX-5XX
* Rewrite URL (URL rewriting) dùng regular expressions
* Hạn chế tỷ lệ đáp ứng truy vấn
* Giới hạn số kết nối đồng thời hoặc truy vấn từ 1 địa chỉ
* Khả năng nhúng mã PERL
* Hỗ trợ và tương thích với IPv6
* Hỗ trợ WebSockets
* Hỗ trợ truyền tải file FLV và MP4

**Những tính năng máy chủ mail proxy của Nginx:**

* Các phương pháp xác thực :
  + POP3: USER/PASS, APOP, AUTH LOGIN/PLAIN/CRAM-MD5;
  + IMAP: LOGIN, AUTH LOGIN/PLAIN/CRAM-MD5;SMTP: AUTH LOGIN/PLAIN/CRAM-MD5;
  + Hỗ trợ SSL, STARTTLS và STLS

## **1.8 Cấu hình cho Nginx**

Tất cả các file cấu hình của nginx nằm trong thư mục /etc/nginx. Trong đó tập tin cấu hình chính là /etc/nginx/nginx.conf. File cấu hình này gồm các biến như: user, worker\_processes, pid, event và HTTP như đã trình bày ở phần trên. Trong khối HTTP là khối cấu hình chung giúp Nginx xử lý lưu lượng web HTTP, ta quan tâm đến chỉ thị **include:**

****

**Include** có vai trò thêm nội dung cấu hình từ file khác vào file cấu hình chính,do đó nó có thể cho phép ta thêm vào cấu hình của các tên miền ảo: **include /etc/nginx/site-enabled**(chứa liên kết biểu tượng tới các file cấu hình trong thư mục **sites-available**), để cấu hình các khối máy chủ ta cấu hình các file của thư mục **sites-available** (mỗi file trong thư mục này là một tên miền khác nhau):

A screenshot of a social media post

Description generated with very high confidence

Khối server là nơi người dùng nginx cấu hình các chỉ thị cho từng domain của họ gồm các cấu hình chính:

*1. Cổng lắng nghe:*

Cho biết hostname hoặc IP của nginx và cổng TCP lắng nghe (mà ta có thể thay đổi) các kết nối HTTP, mặc định là 80:



*2. Thư mục root :*

Ta có thể cấu hình thư mục lưu các tài liệu gốc:

A picture containing object

Description generated with very high confidence

Như ở ví dụ này thư mục gốc là /usr/share/nginx/html

*3. Location (cấu hình file và thư mục):*

Cho biết nginx làm thế nào để xử lý yêu cầu cho tên miền(vd: http://example.com) với các tập tin và tư mục cụ thể (vd: <http://example.com/blog/>)

A screenshot of a cell phone

Description generated with very high confidence

## **1.9 So sánh giữa apache và Nginx**

#### 1.7.1 Máy chủ phục vụ nội dung tĩnh và động

**Apache:**

Apache xử lý yêu cầu nội dung tĩnh từ client bằng phương pháp dựa trên file thông thường.

Apache cũng có thể xử lý nội dung động bằng cách nhúng một bộ xử lý của ngôn ngữ trong yêu vào mỗi tiến trình worker của mình. Điều này cho phép nó có thể tự xử lý các yêu cầu nội dung động mà không cần phụ thuộc vào một thành phần nào bên ngoài. Bộ xử lý nội dung động này được bật lên thông qua việc sử dụng các module tự động nạp. Các modules có thể dễ dàng được tháo ra nếu như nội dung yêu cầu thay đổi.

**Nginx:**

Nginx thì không có khả năng xử lý các yêu cầu nội dung dynamic một cách tự nhiên mà phải gửi cho một bộ xử lý bên ngoài thực thi,và đợi việc xử lý nội dung xong rồi mới trả về cho client. Đối với người adminstrator thì điều này có nghĩa là anh ta phải config giữa Nginx và bộ xử lý bên ngoài thông qua một trong những protocol mà Nginx có thể giao tiếp như FastCGI, SCGI, uWSGI, Memcache. Tuy nhiên do bộ biên dịch nội dung động không được nhúng trong các tiến trình worker nên nội dung tĩnh sẽ được phục vụ nhanh chóng còn và việc biên dịch cho các yêu cầu nội dung động chỉ được gọi khi cần thiết.

#### 1.9.2 Cấu hình phân bố và cấu hình tập trung

**Apache:**

Apache cho phép thêm cấu hình trên mỗi thư mục bằng việc kiểm tra, biên dịch các chỉ thị trong tệp ẩn .htaccess chứa trong thư mục đó. Khi tiếp nhận một request, Apache sẽ kiểm tra mỗi thành phần của đường dẫn của request để tìm tệp .htaccess và áp dụng các chỉ thị trong tệp cho request đó. Tệp .htaccess giúp cho các nhà cung cấp dịnh vụ hosting duy trì kiểm soát cấu hình chính trong khi cung cấp cho khách hàng quyền kiểm soát các thư mục của họ.

**Nginx:**

Nginx không có thêm cấu hình .htaccess như Apache, cũng như là không cung cấp cơ chế nào để đánh giá cấu hình từng thư mục như apache. Điều này dẫn tới sự không linh hoạt bằng Apache nhưng Nginx có những lợi thế riêng của mình. Nginx xử lý request nhanh hơn Apache do không phải xử lý file .htaccess.

Về vấn đề bảo mật, đối với Apache việc phân mảnh cấu hình cấp thư mục tỏ ra nguy hiểm với người dùng có khả năng cấu hình không tốt, có nhiều mối đe dọa hơn vì trách nhiệm về bảo mật phụ thuộc vào từng cá nhân riêng biệt.

#### 1.9.3 Modules

Apache và Nginx đều hỗ trợ mở rộng bằng các modules.

**Apache** cho phép ta có thể tự động load hoặc unload modules trong quá trình running. Lõi của Apache luôn hiện diện, còn các modules có thể bật hoặc tắt, thêm bớt tính năng gắn vào main server. Apache sử dụng module trong rất nhiều tác vụ của nó. Các module có thể được sử dụng để thay thế một số chức năng lõi của server, ví dụ như là mod\_php, module này tích hợp trong trình biên dịch PHP của mỗi tiến trình worker. Các module không bị giới hạn khi xử lý các nội dung động, để mở rộng chức năng lõi của web server.

**Nginx** cũng áp dụng hệ thống modules nhưng khác với Apache, modules trong Nginx không tự động load mà nó phải được chọn và được biên dịch vào hệ thống lõi của Nginx. Module của Nginx cũng có chức năng như Apache như caching, proxy, access limit, deny IP…Với Nginx thì cách thức lựa chọn module như trên, ta có thể thêm module đúng theo những gì mình cần, nâng cao tính bảo mật, sự thuận tiện, dễ bảo trì hơn so với Apache. Nhưng ta cần phải biết chính xác những module cần tích hợp vào Nginx.

#### 1.9.4 Sử dụng Nginx chung với apache

Sử dụng nginx làm reverse proxy đứng trước Apache. Nginx sẽ handle tất cả request từ client. Với tốc độ xử lý nhanh của Nginx, có thể handle được nhiều conection đồng thời. Đối với nội dung tĩnh, Nginx sẽ handle nhanh hơn và trả kết quả trực tiếp đến client. Còn đối với nội dung động, Nginx sẽ proxy chuyển đến Apache, Apache sẽ xử lý kết quả, trả về tương ứng. Sau đó Nginx sẽ trả kết quả từ Apache về phía client. Với cách thiết kế này, Nginx đóng vai trò như một máy phân loại request, giảm bớt request phía Apache, handle nhanh hơn, giảm số lượng process và thread của Apache.

# **Chương 2: Tìm hiểu và cài đặt MySQL**

## **2.1 MySQL là gì?**

- MySQL được phát triển, được công bố, được hỗ trợ bởi MySQL AB, là một công ty của Thụy Điển.

- MySQL là một RDBMS nhanh và dễ dàng để sử dụng.

- MySQL là hệ quản trị cơ sở dữ liệu tự do mã nguồn mở phổ biến nhấ thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là cơ sở dữ liệu tốc đô cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tinh bảo mật cao, MySQL rất thích hợp với các ứng dụng có truy cập CSDL trên Internet. Người dùng có thể tải về MYSQL miễn phí từ trang chủ. MySQL có nhiều pheien bản cho các hệ điều hành khác nhau: Phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix,…

- MySQL là một trong những ví dụ rất cơ bản về Hệ quản trị Cơ sở dữ liệu sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL).

- MySQL được sử dụng cho việc bổ trợ PHP, Peri và nhiều ngôn ngữ khác, làm nơi lưu trữ thông tin trên các trang web viết bằng PHP hay Peri….

## **2.2 Ưu điểm của MySQL**

- MySQL là mã nguồn mở. Vì thế, để sử dụng nó, bạn không mất khoản phí nào. MySQL là một chương trình rất mạnh mẽ, sử dụng một Form chuẩn của ngôn ngữ dữ liệu nổi tiếng là SQL. Việc trên nhiều hệ điều hành và với nhiều ngôn ngữ PHP, PERL, C, C++, Java… là một điều kiện thuận lợi cho các nhà phát triển ứng dụng, đồng thời làm cho MySQL ngày càng trở nên phổ biến và rộng rãi.

- MySQl làm việc nhanh và khỏe ngay với các tập dữ liệu lớn, hỗ trợ các cơ sở dữ liệu lớn, lên tới 50 triệu hàng hoặc nhiều hơn nữa trong một bảng. Kích cỡ file mặc định được giới hạn cho một bảng là 4GB, nhưng có thể tăng kích cỡ này để đạt tới giới hạn lý thuyết là 8TB.

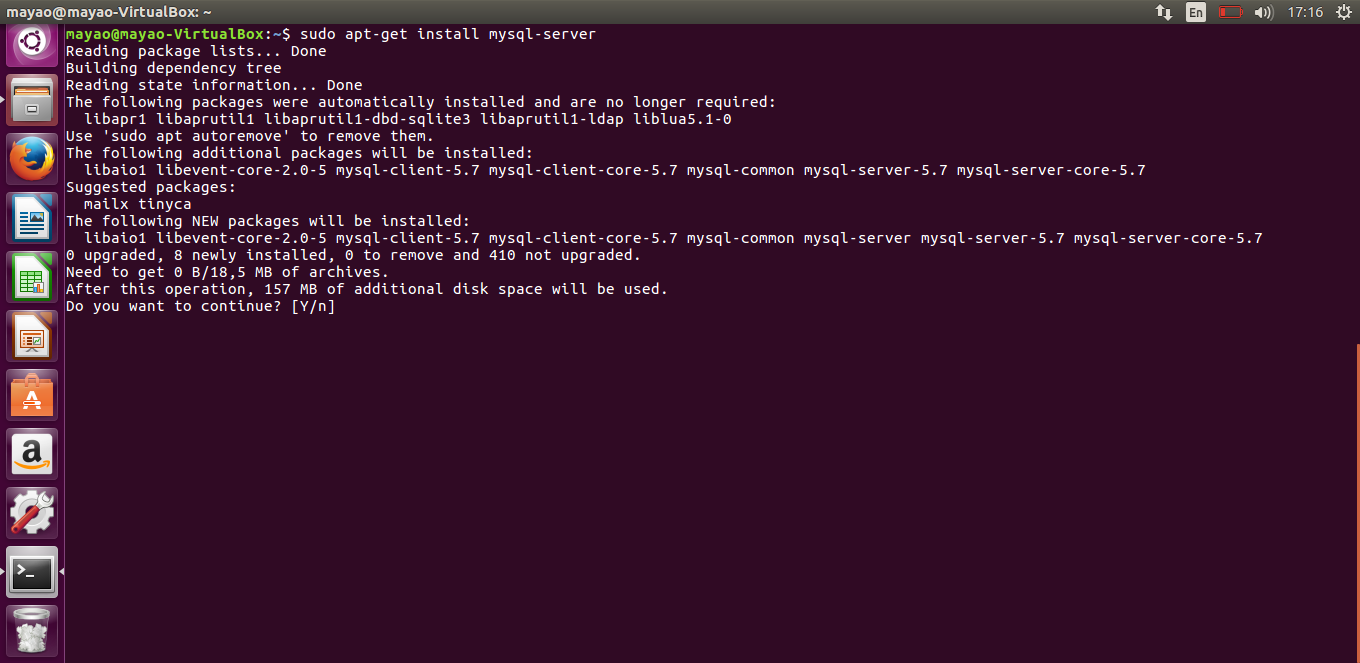
- MySQL là có thể điều chỉnh. Giấy phép GPL mã nguồn mở cho phép lập trình viên sửa đổi phần mềm MySQL để phù hợp với môi trường cụ thể của họ.

## **2.3 Cách cài đặt MySQL trên hệ điều hành Linux**

- Đầu tiên mở terminal lên và gõ vào dòng lệnh

*sudo apt-get install mysql-server*

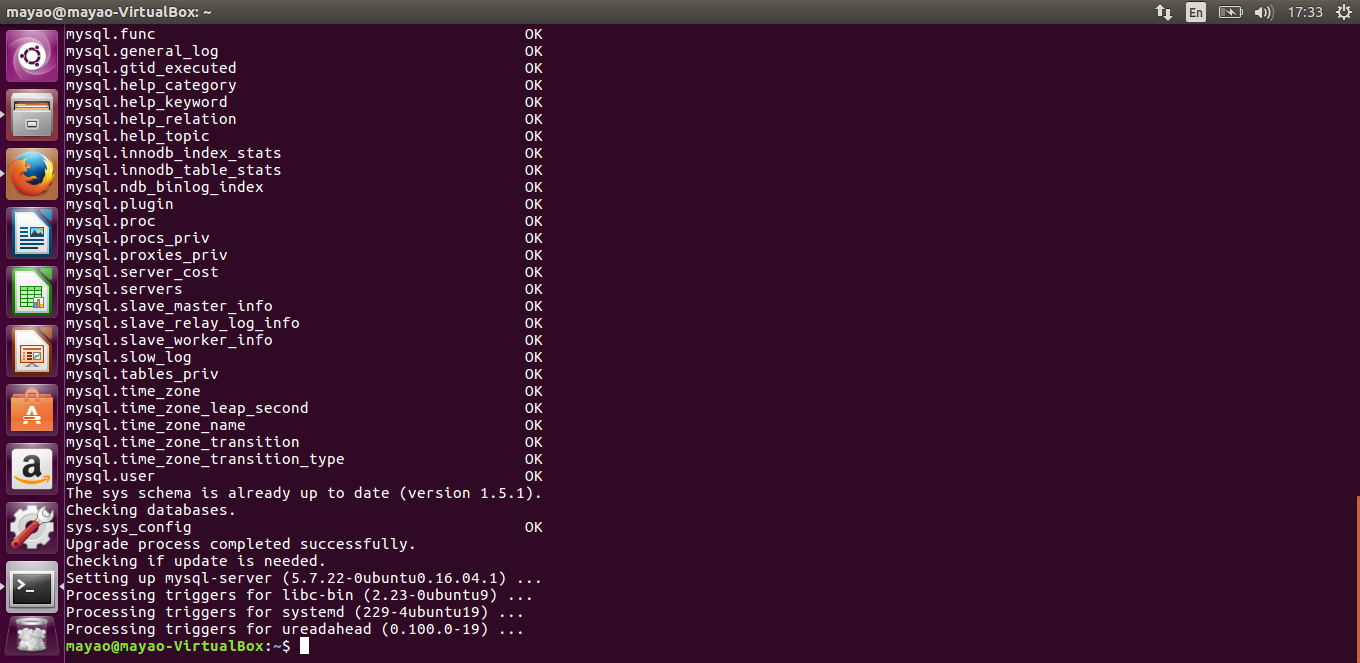
- Sau đó màn hình sẽ hiển thị như sau:



- Gõ Y và ấn Enter để tiếp tục.

- Sau một thời gian màn hình sẽ hiển thị giao diện để nhập password và xác nhận password cho database. Sau khi nhập và xác nhận xong bạn ấn Enter.

- Sau đó màn hình hiển thị:



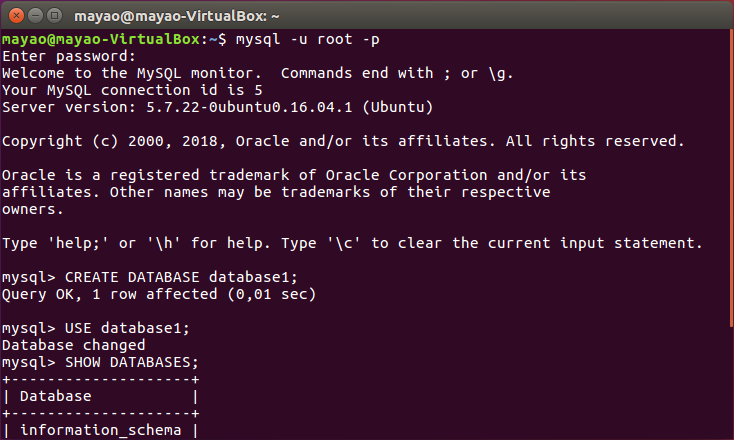
- Màn hình hiển thị dòng: ***upgrade process completed successfully*** nghĩa là quá trình cài đặt MySQL của bạn đã thành công.

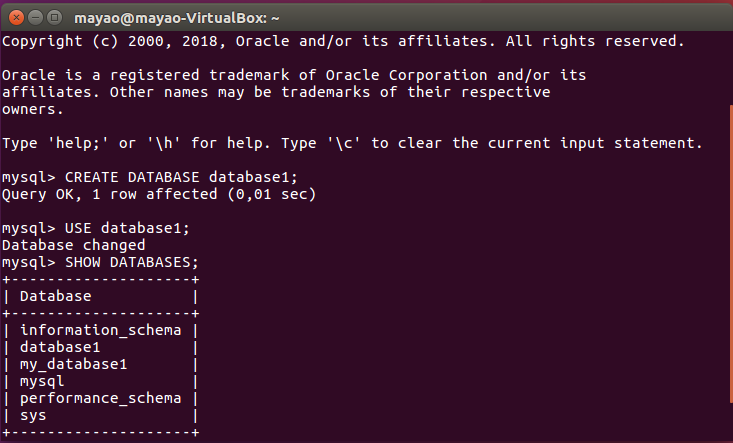
## **2.4 Chạy demo một số câu lệnh của MySQL trên terminal của hệ điều hành Linux**

Để chạy chương trình MySQL trên Linux ta dùng câu lệnh:

*mysql -u root -p*

Chương trình dưới đây demo các câu lệnh để tạo một database và hiển thị các trường của database đó.





# **Chương 3: Lập trình Web với Apache, PHP và MySQL trên Linux**

## **3.1 Giới thiệu về Apache và PHP**

#### 3.1.1 Apache và cách cài đặt

Apache là một chương trình máy chủ HTTP nó giúp cho các máy chủ đối thoại với nhau qua giao thức HTTP. Apache chạy trên các hệ điều hành khác nhau như Unix, Window, Novell Netware,…Nó đóng vai trò quan trọng trong quá trình phát triển hệ thống mạng các trang web trên thế giới(world wide web). Khi được phát hành lần đầu, Apache là chương trình máy chủ mã nguồn mở duy nhất có khả năng cạnh tranh với các chương trình máy chủ tương tự của Netscape Communication Corporacation mà ngày nay được phát triển qua bản thương mại Sun Java System Web Server. Từ đó trở đi Apache đã không ngừng tiến triển và trở thành một phần mền có sức cạnh tranh với các chương trình máy chủ khác về mặt hiệu suất và tính năng phong phú. Apache được phát triển và duy trì bởi một cộng đồng mã nguồn mở dưới sự bảo trợ của Apache Software Foundation và được phát hành với giấy phép Apache license.

**Cài đặt Apache 2.4 ( trên phiên bản ubuntu 16.04 LTS)**

*Bước 1:* mở một Terminal mới và và thực hiện các dòng lệnh.

*Bước 2*: lần lượt gõ các lệnh:

1- *$ sudo apt-get update*

2- *$ sudo apt-get install apache2*

-các gói cài đặt của Apache sẽ được tải về và cài đặt.

*Bước 3:* để kiểm tra cài đặt ta sử dụng các lệnh

+ *$ dpkg --get-selections | grep apache*(kiểm tra các gói cài đặt của Apache)

*+ $ dpkg --status apache2*(kiểm tra chi tiết gói apache2: số phiên bản,…)

A screenshot of a cell phone

Description generated with very high confidence

#### 3.1.2 PHP

PHP là ngôn ngữ kịch bản hay mã lệnh chủ yếu được dùng để phát triển ứng dụng viết cho máy chủ, dùng cho mục đích tổng quát, rất thích hợp với các web và có thể dễ dàng nhúng vào các trang HTML. Nó cho phép các nhà phát triển tạo một trang web động một cách nhanh chóng. PHP được tạo ra bởi Rasmus Lerdorf vào năm 1994, ban đầu được xem là một tập con đơn giản của ngôn ngữ kịch bản perl để kết nối cơ sở dữ liệu và từ đó theo dõi tình hình truy cập bản sơ yếu lý lịch của ông trên mạng. Ông đã đặt tên cho nó là “Personal Home Page/Forms interpreter” hay viết tắt là PHP/FI và ngày nay thường được biết đến với “PHP: Hypertext Preprocessor”. PHP hiện đang được duy trì bởi “the PHP group”.

PHP khá hữu dụng cho các dự án “one-man-operated”, bởi với một kĩ thuật thành thạo có thể nhanh chóng tạo ra một nguyên mẫu làm việc cho một webapp. Tuy nhiên lại khó để sử dụng PHP để viết một MVC (Model-View-Control) webapp tốt, do các phần: dữ liệu (model), sự thể hiện(view) và logic (control) được phân chia cho các nhóm khác nhau xử lý.

PHP có 2 chế độ sử dụng:

*1-Command-line Interface (CLI):* ta có thể chạy một đoạn mã PHP tương tự như các chương trình được viết bằng Perl/Python hoặc C/C++…

*2-Các mã kịch bản nhúng trong HTML ở phía server***:** ở phía server mã kịch bản phải được chạy bởi ứng dụng web server có tương thích PHP như: Apache HTTP Server,…

**Cài đặt trên ubuntu:**

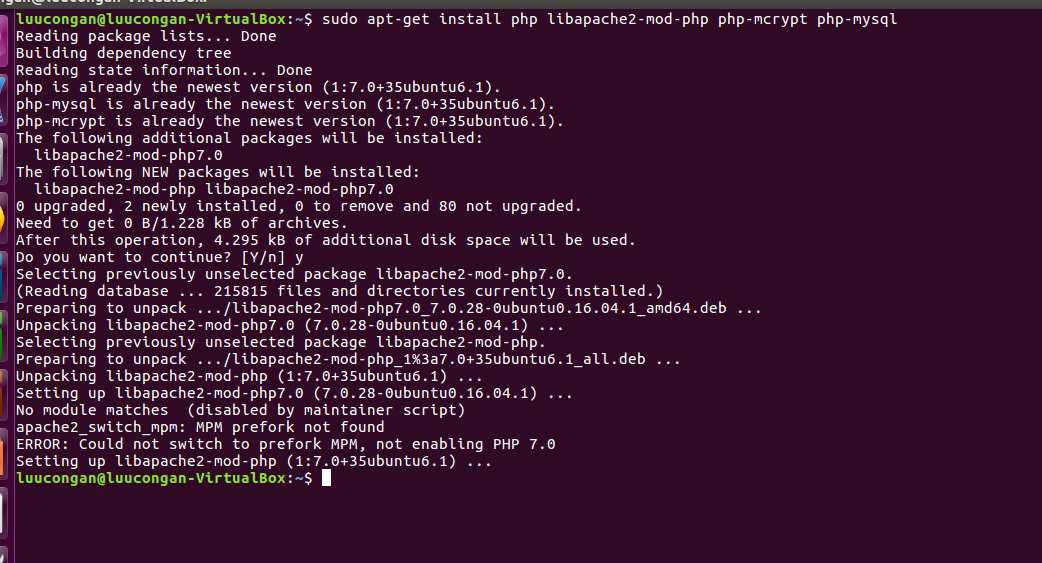
Để cài đặt PHP 7 trên ubuntu ta thực hiện các bước sau:

*Bước 1:* chạy lệnh: *$ sudo apt-get install php libapache2-mod-php php-mcrypt php-mysql* , để tải về các gói cài đặt của PHP và chờ cho đến khi cài đặt thành công.

*Bước 2*: để kiểm tất cả các gói đã cài đặt có liên quan đến PHP ta sử dụng lệnh

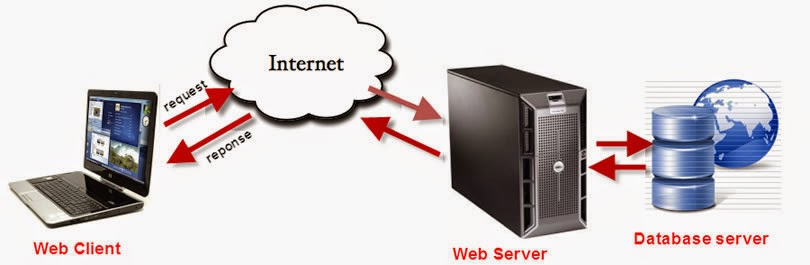
*$ dpkg --get-selections | grep php*

Bước cài đặt như trên hình:



## **3.2 Chạy một trang Web đơn giản với Apache, MySQL, PHP**

#### 3.2.1 Cách thức hoạt động



Mô tả đơn giản hoạt động của 1 trang web động:

Web server có chức năng lưu trữ các file mã nguồn như HTML, CSS, JavaScript, PHP,…. ảnh, video… và phải luôn luôn trong trạng thái hoạt động.

Khi gõ 1 tên miền lên trình duyệt, tên miền này được DNS phân giải và trả về địa chỉ IP của Web server lưu trữ dữ liệu của trang web đó. Phía client sẽ gửi 1 thông điệp HTTP request tới máy chủ ở trên, sau đó máy chủ sẽ gửi lại thông điệp HTTP response( có thể là thông điệp báo lỗi hoặc file html), nếu không có lỗi trình duyệt sẽ hiển thị nội dung của trang web đó.

Bên cạnh đó web server còn có 1 chức năng là complie mã nguồn và xử lí chúng.

#### 3.2.2 Cấu hình

Khi thử nghiệm web trên máy tính cá nhân, chúng ta thường sử dụng localhost, không cần đến internet, dùng chính máy tính đó làm web server, cách này có ưu điểm là dễ dàng kiểm tra và sửa đổi.

Sau đã cài đặt đầy đủ Apache, PHP, MySQL để localhost hoạt động thì ta phải kết nối chúng lại với nhau*( Sử dụng Ubuntu 16.04, PHP7, MySQL 5.7)*

Nếu cài đặt thành công trong thư mục /etc sẽ xuất hiện thêm thư mục apache2 chứa các file cấu hình của phiên bản Apache đã cài đặt. Tệp cấu hình chính là tệp */etc/apache2/apache2.conf* định nghĩa các cấu hình cơ bản của httpd, tiến trình phục cho dịch vụ web.

Thư mục gốc của web server nằm trong thư mục /var/www/, để chạy 1 web khác ta có thể thay đổi đường dẫn trong file apache2.conf hoặc đưa thẳng file chứa mã nguồn của trang web vào /var/www/ .

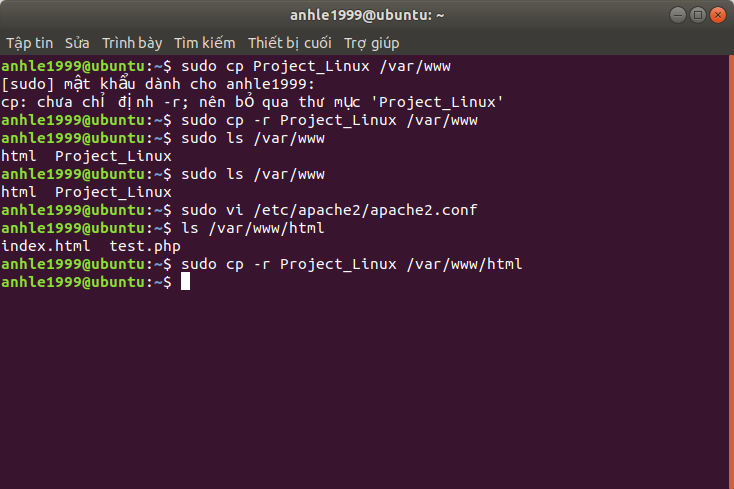
**Thay đổi đường dẫn thư mục gốc web server:**

Đầu tiên chúng ta dùng Vim hoặc Nano để mở file config của Apache lên, vì đây là file cấu hình hệ thống nên phải thực hiện dưới quyền root thì mới có quyền chỉnh sửa.



*<Dicrectory /var/www/**>*  đường dẫn tới thư mục chứa mã nguồn của web.

**Copy thư mục vào thư mục gốc:**

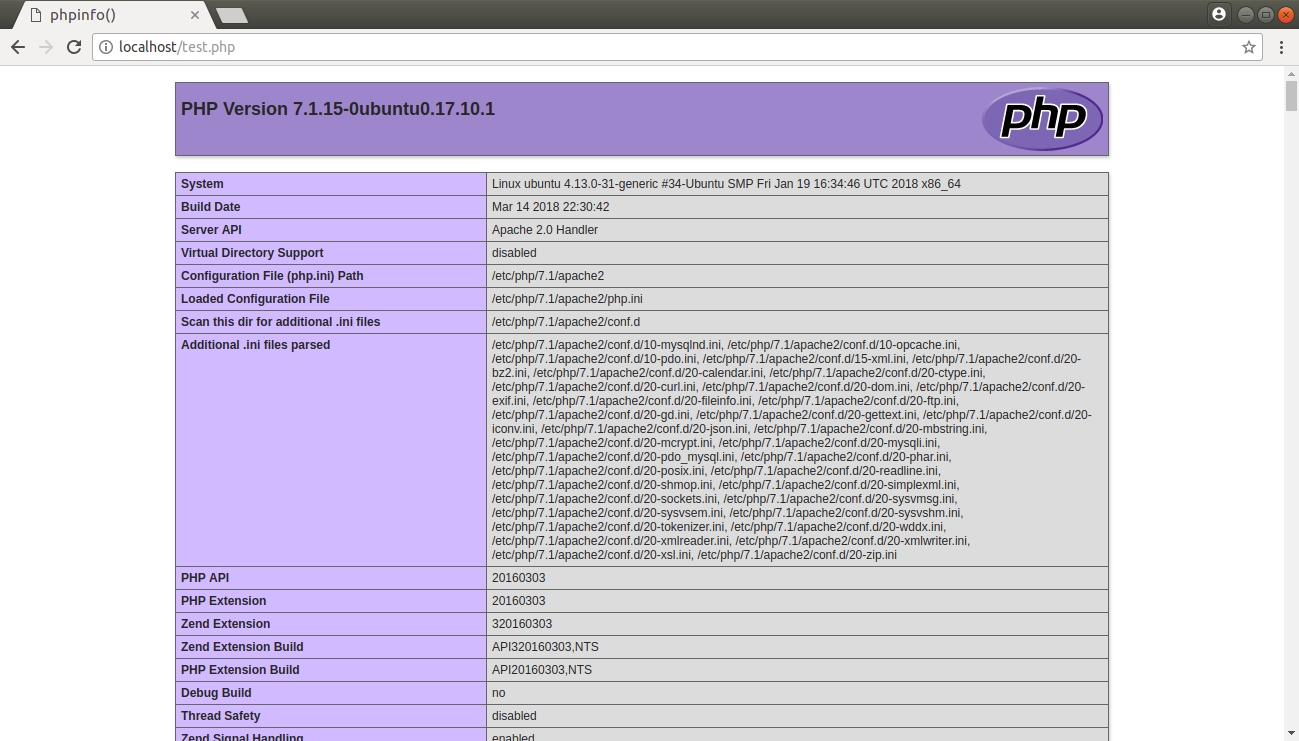


Thư mục Web đã tồn tại trong máy chứa mã nguồn của trang web ta định thử nghiệm, lệnh **cp ...ở trên**  sẽ copy thư mục Project\_Linux tới /var/www/html

Để chạy thử trang web trong Web chúng ta gõ địa chỉ tới *127.0.0.1/Project\_Linux/Web hoặc localhost/Project\_Linux/Web.*

**Kiểm tra xem PHP có hoạt động với Apache:**

Để kiểm tra hoạt động của Apache+ PHP ta chỉ cần tạo 1 file **test.php** vào trong /var/www/html, rồi gõ địa chỉ 127.0.0.1/test.php

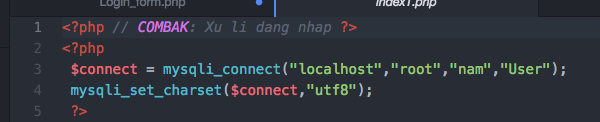


Nếu như trình duyệt hiển thị các thông số của các service mà trang test.php đang sử dụng thì việc cài đặt Apache+ PHP đã thành công.

**MySQL:**

Với 1 trang web động thì cơ sở dữ liệu là không thể thiếu, PHP và MySQL là hai công cụ khá nổi tiếng thường đi kèm với nhau.

Trong phiên bản mới của MySQL( MySQLi), để kết nối PHP với MySQL có những thay đổi nhỏ so với các phiên bản cũ.



Trong đó mysqli() là phương thức dùng để kết nối với MySQL do PHP định nghĩa sẵn, “”……

**phpMyAdmin:**

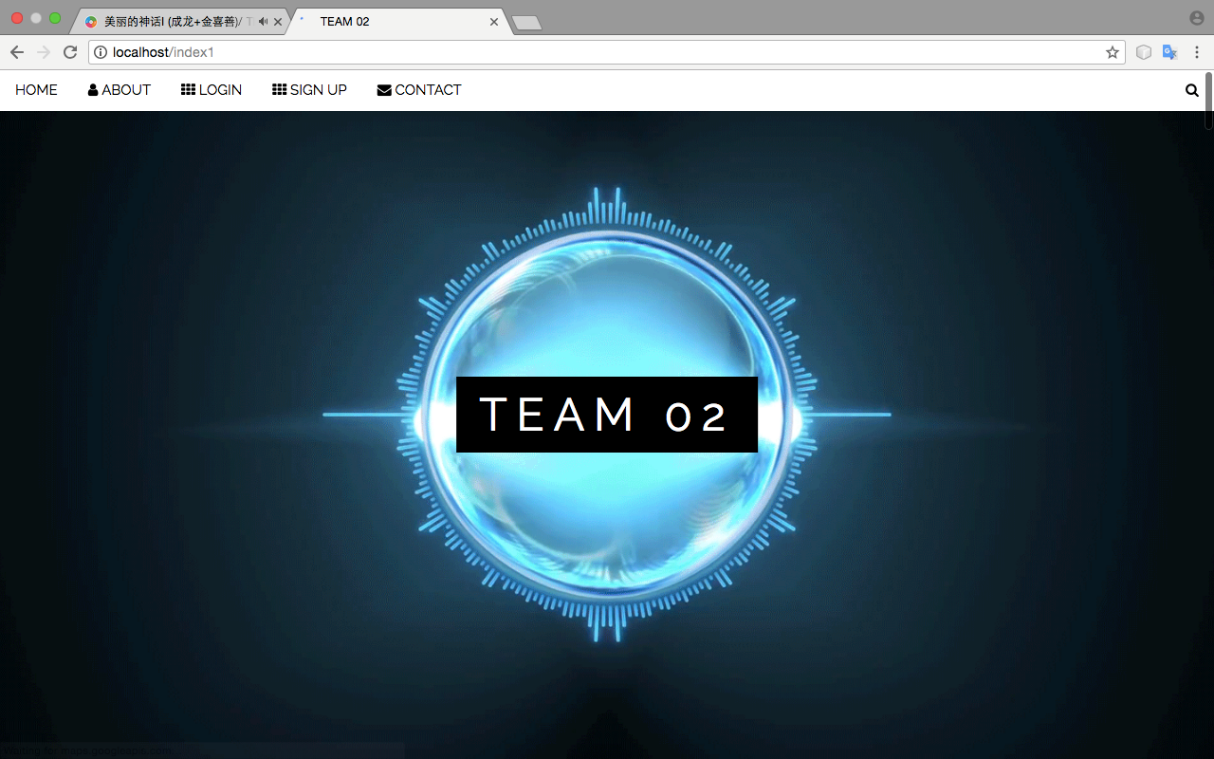
Để quản lí các database, bảng.. trong MySQL 1 cách dễ dàng hơn, chúng ta có thể sử dụng phpMyAdmin. phpMyAdmin giúp người dùng quản lí MySQL thông qua giao diện web. Cài đặt:

***$ sudo apt-get install phpmyadmin***

Trong Linux, sau khi cài đặt phpMyAdmin thì thư mục phpMyAdmin sẽ thêm vào trong /etc . Muốn mở phpMyAdmin lên, ta chỉ cần gõ địa chỉ *127.0.0.1/phpmyadmin* hoặc *localhost/phpmyadmin* giao diện của nó sẽ hiện ra.

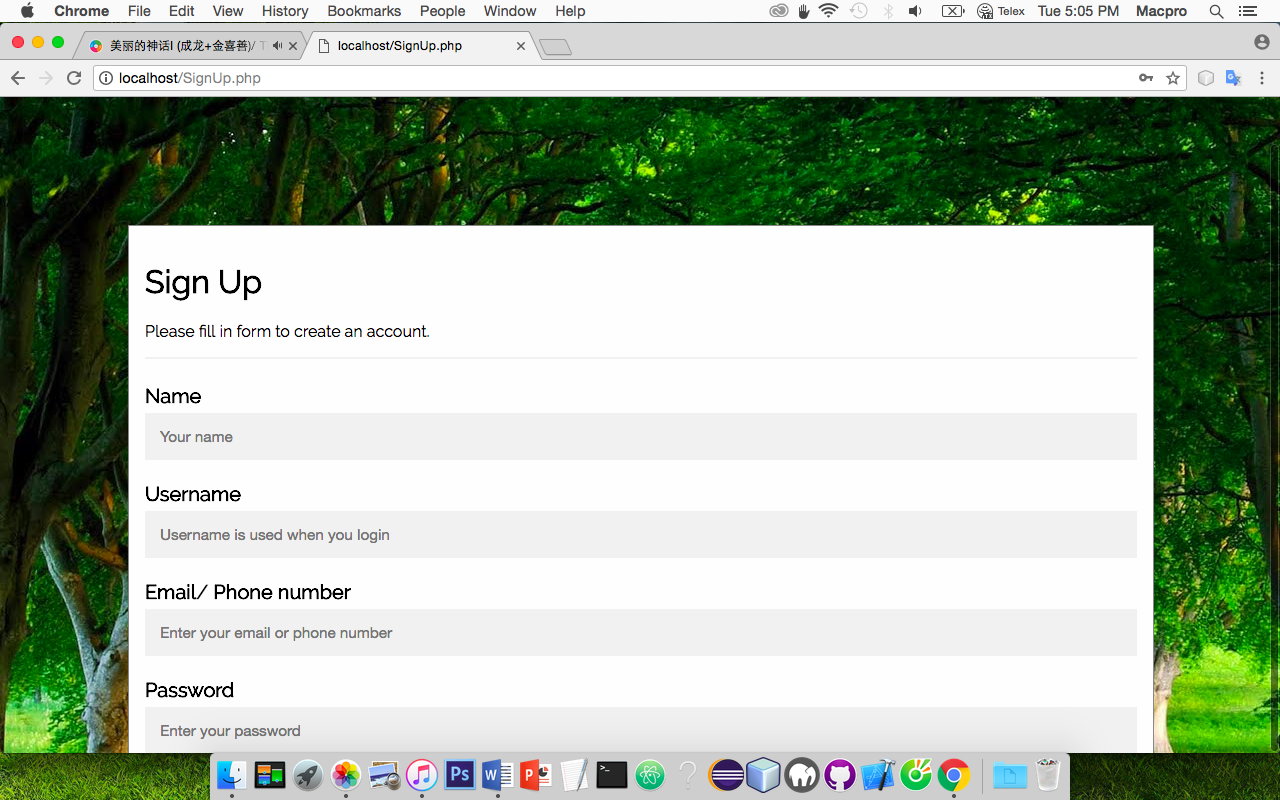
#### 3.2.3 Web demo

Trang web sử dụng Apache 2 , PHP 7, MySQLi các thư viện W3.CSS, jQuery

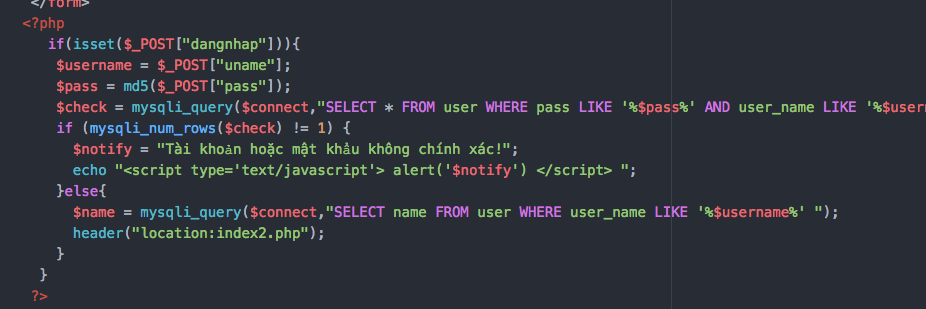


Home page trạng thái người dùng chưa đăng nhập hiển thị nội dung giới thiệu các thành viên trong team.

**Chức năng đăng ký:**

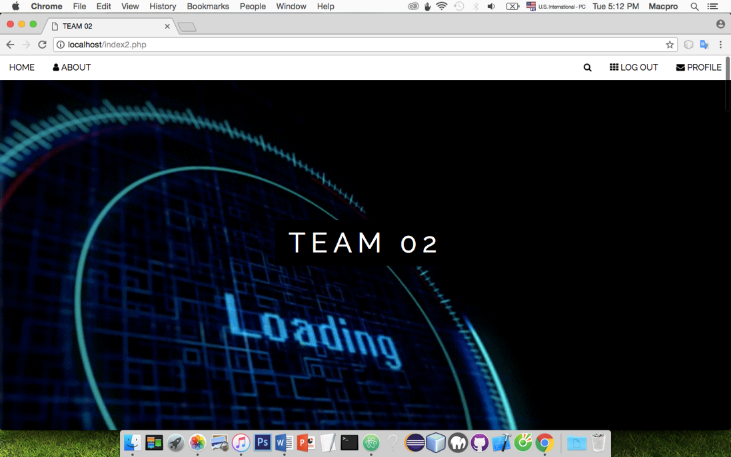


Dữ liệu người dùng nhập vào sẽ được insert vào trong database tên là **User**, nếu trùng username thì sẽ không thêm, bắt người dùng đổi tên khác.



**Chức năng đăng nhập:**

User name và pass người dùng nhập vào sẽ được kiểm tra xem có tồn tại trong bảng **user**, nếu đăng nhập thành công sẽ chuyển sang giao diện đăng nhập.



# **Kết luận**

Phần mềm mã nguồn mở là rất hữu ích cho việc phát triển, dễ dàng sửa đổi mà không lo lắng về vấn đề bản quyền. Chính vì thế, hệ điều hành Linux ngày càng có tầm quan trọng hơn trong thế giới công nghệ 4.0 đang rất phát triển hiện nay. Thông qua đề tài này, nhóm được hiểu rõ hơn về Nginx, MySQL, Apache, hiểu rõ hơn về cách thức thực hiện cũng như vai trò của nó, hiểu được cách tạo một trang web bằng ngôn ngữ PHP sử dụng Apache như thế nào.

Trong quá trình tìm hiểu sẽ không tránh khỏi những sai sót. Nhóm em rất mong được sự góp ý bổ sung từ thầy.

# **Tài liệu tham khảo**

1. <https://vi.wikipedia.org/wiki/MySQL>
2. <https://www.digitalocean.com/community/tutorials/apache-vs-nginx-practical-considerations>
3. <https://mangmaytinh.net/threads/cau-hinh-nginx.49/>
4. <https://hocvps.com/nginx/>