

在 Ubuntu 16.04 下安裝 Apache 2.4 + PHP 5.6 + Laravel 5

Laravel is a very popular open source PHP framework aimed at easy development of applications. The following guide will allow you to run Laravel 5 on a Ubuntu 16.04 based Apache HTTP server with PHP 5.6.

目錄：

1. 系統準備（選用）
2. 安裝 Apache HTTP Server
3. 安裝 PHP 5.6 和 PHP extensions
4. 啟用模組
5. 安裝 Composer
6. 安裝 Laravel 和建立新專案
7. 設定 Apache 虛擬主機組態（以 Laravel 為例）

1. 系統準備：

A. 安裝基本套件：

```
$ sudo apt-get install ssh # 或 openssh-server
$ sudo apt-get install openssl
$ sudo apt-get install zip unzip
$ sudo apt-get install git
```

其中：

- Secure Shell (SSH) 模組是為了提供遠端使用者在電腦上的 Shell (殼層) 提供安全的傳輸和使用環境。
- Secure Sockets Layer (SSL) 是一種安全協定，目的是為網際網路通訊，提供安全及資料完整性保障。
- zip 和 unzip 是為了提供 zip 檔案的壓縮和解壓縮使用。
- git 套件是為了讓 Ubuntu 可以提供作為 Git 的本區儲存庫或別人的遠端儲存庫使用。

B. 將系統更新到最新狀態。

```
$ sudo apt-get update  
$ sudo apt-get upgrade
```

C. 防火牆設定：

a. 安裝 ufw

```
$ sudo apt-get install ufw
```

b. 允許 SSH 和 HTTP 服務通過防火牆：

```
$ sudo ufw allow openssh-server  
$ sudo ufw allow http
```

c. 啟用 ufw

```
$ sudo ufw enable
```

d. 確認 ufw 的狀態

```
$ sudo ufw status verbose
```

2. 安裝 Apache HTTP Server

A. 安裝 Apache2 套件：

```
$ sudo apt-get update
```

```
$ sudo apt-get install apache2
```

B. Apache 的安全設定

B-1. 隱藏 Apache 的版本

編輯 `/etc/apache2/conf-available/security.conf` 檔案，將 "ServerTokens" 設定為 "Prod"，並將 "ServerSignature" 設定為 "Off"。

```
$ sudo nano /etc/apache2/conf-available/security.conf
```

在 27 約第行可以找到 "ServerTokens OS"，在前面加 "#" 註解後，新增一行 "ServerTokens Prod"。

在 37 約第行可以找到 "ServerSignature On"，在前面加 "#" 註解後，新增一行 "ServerSignature Off"。

重啟 Apache2 服務。

```
$ sudo service apache2 restart
```

也可以使用下列指令來檢查組態設定是否成功。

```
$ sudo curl -I http://127.0.0.1
```

B-2. 限制 /var/www/html 目錄的瀏覽權限：

編輯 /etc/apache2/apache2.conf 檔案，將下列文字中的 Indexes 移除。

```
<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

重啟 Apache 服務。

```
$ sudo service apache2 restart
```

C. 設定 /var/www 目錄的讀取、寫入和執行權限：

一般來說，在安裝 Apache 軟體時會使用 sudo 指令，這會使得 /var/www 目錄的用戶和群組權限均屬於 root 擁有，這對於後續要增加新的網站目錄相當的不方便，而且 root 帳號一但被盜用就會危及全系統的安全，所以需要修改 /var/www 目錄的用戶及群組權限，讓特定的用戶或群組可以讀、寫和執行網站檔案。執行指令如下：

```
$ sudo chown -R www-data:www-data /var/www
$ sudo chmod -R 775 /var/www
```

其中，www-data 用戶及群組是 Ubuntu 的 Apache 帳號。

D. 設定全域伺服器名稱或 IP：

接下來，可能會碰到一種情形，那就是啟動 Apache 時發現雖然 Apache 正常啟動了，但卻出現一條警告訊息如下：

AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1. Set the 'ServerName' directive globally to suppress this message

這個訊息是說還沒替 Apache 設定 ServerName。要解決這個問題，需要手動在 /etc/apache2/apache2.conf 檔案末端中加入 "ServerName 127.0.0.1" 的設定。然後再測試 Apache 的組態設定語法，指令如下，如果組態設定正確則會出現 "Syntax OK" 訊息。

```
$ sudo nano /etc/apache2/apache2.conf
```

在檔案末端加入 "ServerName 127.0.0.1"，儲存後離開。

```
$ sudo apache2ctl configtest
```

E. 啟用重寫規則 (mod_rewrite)

Rewrite 的主要功能是實現 URL 的跳轉，其目的是當網站在更新或維護期間，例如主機維護、移轉伺服器、重新組織目錄結構、變換 URL 或更改功能變數名稱等行為，讓使用者不會因此受到影響的機制。

Rewrite 的執行層級有兩種，分別是伺服器級 (HTTPd.conf) 和目錄級 (.htaccess)。伺服器級的執行方法是直接利用 RewriteEngine on 來打開

rewrite 功能。目錄級的執行方法則是必須打開此目錄的 FollowSymLinks 屬性且在 .htaccess 裡要聲明 RewriteEngine on。

在 Apache 中啟用 rewrite 模組。

```
$ sudo a2enmod rewrite
```

如果想知道 Apache 已經啟用了那些模組，可以到 /etc/apache2/mods-enabled 目錄下查詢。

F. 重新啟動 Apache：

```
$ sudo systemctl restart apache2
```

or

```
$ sudo service apache2 restart
```

3. 安裝 PHP 5.6 和 Extension：

A. Ubuntu 16.04 的 PHP 預設安裝為 7.1 版，指令如下：

```
$ sudo apt-get install php
```

B. 如果想要使用 PHP 5.6 時需要指定安裝版本，程序如下：

```
$ sudo apt-get install python-software-properties
```

```
$ sudo add-apt-repository ppa:ondrej/php
```

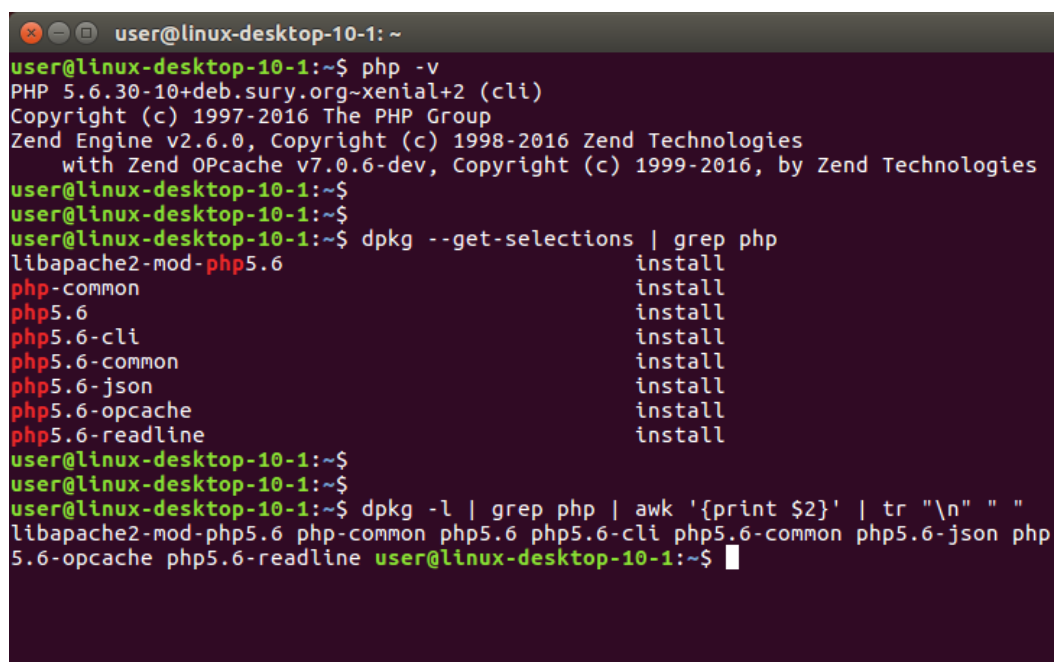
```
$ sudo apt-get update
```

```
$ sudo apt-get upgrade
```

```
$ sudo apt-get install php5.6 # for PHP 5.6
```

C. 檢查 PHP 版本：其執行結果如圖 1。（三種方法擇一即可）

```
$ php -v
$ dpkg --get-selections | grep php
$ dpkg -l | grep php | awk '{print $2}' | tr "\n" " "
```

A terminal window titled 'user@linux-desktop-10-1: ~' showing the execution of three commands to check the PHP version. The first command 'php -v' displays the PHP version 5.6.30-10+deb.sury.org~xenial+2 (cli) and its dependencies. The second command 'dpkg --get-selections | grep php' lists installed PHP-related packages and their status as 'install'. The third command 'dpkg -l | grep php | awk '{print \$2}' | tr "\n" " "' lists the same packages separated by spaces.

```
user@linux-desktop-10-1:~$ php -v
PHP 5.6.30-10+deb.sury.org~xenial+2 (cli)
Copyright (c) 1997-2016 The PHP Group
Zend Engine v2.6.0, Copyright (c) 1998-2016 Zend Technologies
    with Zend OPcache v7.0.6-dev, Copyright (c) 1999-2016, by Zend Technologies
user@linux-desktop-10-1:~$
user@linux-desktop-10-1:~$ dpkg --get-selections | grep php
libapache2-mod-php5.6                                install
php-common                                           install
php5.6                                               install
php5.6-cli                                           install
php5.6-common                                        install
php5.6-json                                          install
php5.6-opcache                                       install
php5.6-readline                                      install
user@linux-desktop-10-1:~$
user@linux-desktop-10-1:~$ dpkg -l | grep php | awk '{print $2}' | tr "\n" " "
libapache2-mod-php5.6 php-common php5.6 php5.6-cli php5.6-common php5.6-json php
5.6-opcache php5.6-readline user@linux-desktop-10-1:~$
```

圖 1 PHP 版本檢查

D. 安裝 PHP extensions

由於網路上不同教學文件所安裝的 PHP extension 並不相同，所以將所有出現過的 extensions 的安裝程序列出，請依自己的需求增減。

```
$ sudo apt-get install libapache2-mod-php5
$ sudo apt-get install php5.6-cli
$ sudo apt-get install php5.6-curl
$ sudo apt-get install php5.6-gd
$ sudo apt-get install php5.6-json
```

```
$ sudo apt-get install php5.6-mbstring
$ sudo apt-get install php5.6-mcrypt
$ sudo apt-get install php5.6-gettext
$ sudo apt-get install php5.6-zip
```

其中：

- cURL 模組是為了解讓 PHP 可以使用不同的通訊協定去連結不同種類的伺服器。目前支援的通訊協定包含了 http, https, ftp, gopher, telnet, dict, file, and ldap。
- mcrypt 模組是為了解讓 PHP 進行加密解密操作，它提供了內含 DES 算法在內的 35 種處理資料用的函數。
- mbstring 模組是為了解讓 PHP 可以處理多位元組的文字，例如 UTF-8。
- gettext 模組是為了解讓 PHP 可以實做多國語系支援使用的。
- gd 模組是為了解讓 PHP 可以方便地處理圖形，讓設計者可以很方便地生成驗證碼、圖片加水印，以及生成略縮圖等。
- php-json 模組是為了解讓 PHP 可以處理 JSON 資料格式。
- php-zip 模組則是為了解讓 PHP 可以壓縮或讀取 zip 文件。
- libapache2-mod-php5 是讓 Apache 支援 PHP 的模組。
- php-cli 是指命令列界面 (Command Line Interface; CLI)。

4. 啟用套件：

至此，使用安裝的套件並未被啟用，所以不會正常的發揮作用。為了解讓已安裝的 php extensions 可以正常作用，Ubuntu 提供了 phpenmod 套件（在

安裝 php5.6-common 套件時因逃鍵相依性會被自動安裝）來啟用其他的套件，例如 mcrypt 和 mbstring 模組，指令如下：

```
$ sudo phpenmod mcrypt
$ sudo phpenmod mbstring
$ sudo a2enmod php-5.6
$ sudo systemctl restart apache2
```

5. 安裝 Composer：

Composer 是用來管理 PHP 套件相依性的工具。其安裝方式有兩種，第一種是本地安裝 (Locally)，指安裝完畢後僅能在一個專案目錄中執行，也就是安裝 Composer 的目錄。第二種是全域安裝 (Globally)，指安裝完畢後將安裝路徑加到 PATH 變數中，或將它放在 /usr/local/bin 目錄中，成為全域指令，供全部專案使用。

A. 本地安裝：

A-1. 使用 curl 指令下載並安裝：

```
$ sudo apt-get install curl
$ cd ~/
$ sudo curl -sS https://getcomposer.org/installer | php
```

A-2. 使用 PHP 指令安裝：

```
$ php composer-setup.php --install-dir=bin --filename=composer
```

其中，`--install-dir` 為安裝路徑，而 `-filename` 為安裝檔案名稱。執行時需輸入 `php bin/composer`。

A-3. 下載最新版的 Composer：

```
$ php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');  
$ php -r "if (hash_file('SHA384', 'composer-setup.php') ===  
    '55d6ead61b29c7bdee5cccfb50076874187bd9f21f65d8991d46ec5cc90  
    518f447387fb9f76eabelfbbacf329e583e30') { echo 'Installer  
    verified'; } else { echo 'Installer corrupt';  
    unlink('composer-setup.php'); } echo PHP_EOL;"  
$ php composer-setup.php  
$ php -r "unlink('composer-setup.php');"
```

B. 將 Composer 從本地使用改為全域使用：

```
$ sudo mv composer.phar /usr/local/bin/composer  
$ sudo chmod +x /usr/local/bin/composer
```

6. 安裝及使用 Laravel：

在第一次建立 Laravel 專案前需要使用 Composer 來下載 Laravel installer。

A. 使用 Composer 下載 Laravel 安裝包：

```
$ sudo composer global require "laravel/installer"
```

並將 `$HOME/.composer/vendor/bin` 路徑加到 `~/.bashrc` 檔案中，再執行 `source` 指令：

```
$ echo 'export PATH="$PATH:$HOME/.composer/vendor/bin"' >>
~/.bashrc
$ source ~/.bashrc
```

B. 在 `/var/www` 目錄下建立 laravel 網站：

建立 laravel 網站的方法有兩種，第一種是使用 `laravel new` 指令建立。第二種是使用 `composer` 指令建立。第一種方法需要較複雜的目錄權限設定程序，例如建立新網站的用戶帳號必須要有執行 `laravel` 指令的權限，必須要有在 `/var/www` 建立新目錄的權限，且不能使用 `sudo` 和 `laravel` 的組合指令，因為 `sudo` 會暫時改變環境變數而找不到 `laravel` 致使無法執行的問題。雖然不是無法解決，但程序較為複雜，所以記錄在其他文件中，在此暫不討論。所以，底下僅就第二種方法，使用 `composer` 指令來建立 laravel 網站來做說明。指令執行程序如下：

B-1. 下載並安裝一份 Laravel 副本，存放在 "your-project-name"（以 `mylaravel.com` 為例）的目錄下。

```
$ sudo composer create-project laravel/laravel your-project-
name --prefer-dist
```

B-2. 改變網站目錄的擁有者和群組：

在 B-1 步驟執行完畢後，mylaravel.com 目錄的權限是屬於 root 群組和帳號的，但我希望 mylaravel.com 目錄的權限可以開放給 www-data 群組讀、寫和執行。同時，隸屬於 www-data 群組的用戶，例如 user 帳號，可以有 rwx 的權限，因此需要執行下列程序。

首先，將 user 帳號加入 www-data 群組，並將 user 帳號的群組順序的第一個切換為 www-data 群組。

```
$ sudo adduser user www-data
$ newgrp - www-data
```

其次，將 mylaravel.com 目錄的權限開放給 www-data 群組，但擁有者維持為 root 帳號。

```
$ sudo chown -R :www-data /var/www/mylaravel.com
```

接著，開放 rwx 權限給現在的用戶和群組：

```
$ sudo chmod -R 775 /var/www/mylaravel.com
```

最後，開放 storage 和 bootstrap/cache 的 rwx 權限，這是 laravel 官網的安裝程序提到要開放的。

```
$ sudo chmod -R 777 /var/www/mylaravel.com/storage
$ sudo chmod -R 775 /var/www/mylaravel.com/bootstrap
```

B-3. 重新計算加密金鑰 (Set Encryption Key)：

```
$ cd /var/www/mylaravel.com
$ php artisan key:generate
$ php artisan cache:clear
```

```
$ php artisan clear-compiled
```

B-4. Laravel 出現 HTTP ERROR 500 訊息：

如果在執行將 laravel 網站佈署到 Apache 虛擬主機的過程中出現未知的錯誤，例如 HTTP ERROR 500，這個問題通常是 laravel 目錄內容被改變，但沒有重新計算加解密金鑰造成的，所以重新執行 B-3 步驟就可以了。

7. 建立虛擬主機 (Virtual Host)：

此範例是為 mylaravel.com 目錄建立 www.mylaravel.com 虛擬主機。

- A. 到 /etc/apache2/sites-available 目錄下，編輯 mylaravel.com.conf 組態設定檔，其內容如下：

```
<VirtualHost *:80>
    ServerName mylaravel.com
    ServerAlias www.mylaravel.com
    DocumentRoot /var/www/mylaravel.com/public
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
    <Directory /var/www/mylaravel.com>
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

其中：

- /var/www/mylaravel.com/public 為網頁放置的根目錄。

- ServerName 為該虛擬主機所傾聽的網域。
- <Directory /var/www/mylaravel.com> 為定義網頁權限的路徑。

B. 將 mylaravel.com.conf 存檔後，編輯 /etc/hosts 檔案，加上
"127.0.0.1 mylaravel.com"。

```
$ sudo nano /etc/hosts  
127.0.0.1      mylaravel.com
```

C. 啟用新的虛擬主機：

```
$ sudo a2ensite mylaravel.com  
$ sudo service apache2 reload  
$ sudo service apache2 restart
```

最後，使用瀏覽器在網址列輸入 <http://mylaravel.com> 應該就可以看到

D. 從其他電腦連線到 Apache 虛擬主機：

如果想要從其他電腦使用瀏覽器瀏覽 Apache 的虛擬主機，瀏覽用的電腦需要建立 hosts 檔案。一般來說，linux 作業系統在 /etc/hosts 位置，而 Windows 7 在 [C:\Windows\System32\drivers\etc\hosts](#) 位置。只要在 hosts 檔案內加上 Apache 主機的實體 IP 和虛擬主機名稱就可以了，例如：

```
10.10.1.104     mylaravel.com
```