

Owncloud : host 安裝。

主要是修改官方的安裝方式，因為教學文件是用 apache2+mariaDB 但是我想整合到 nginx 配 mysql。

參考官方的 nginx 修改範例先試著在 virtual box 裡面安裝看看。

```
#sudo apt-get update
```

```
#sudo apt-get install php5 nginx mysql-server php5-gd php5-json php5-mysql php5-curl php5-intl php5-mcrypt php5-imagick php5-fpm
```

有私人 SSL 憑證 owncloud 設定

1.增加 nginx 設定：

給 owncloud 一個 virtual host 設定檔：

```
#sudo vim /etc/nginx/sites-available/owncloud
```

※注意 **highlight** 部份的設定之後會用到，設定資訊可依個人需求變更

```
#----- owncloud -----#
upstream php-handler {
    server 127.0.0.1:9000;
    #server unix:/var/run/php5-fpm.sock;
}
#-----#

server {
    listen 443 ssl;
    server_name owncloud.biotrump.com;

    ssl_certificate /etc/ssl/nginx/owncloud.crt;
    ssl_certificate_key /etc/ssl/nginx/owncloud.key;

    # Path to the root of your installation
    root /var/www/owncloud/;
    # set max upload size
    client_max_body_size 10G;
    fastcgi_buffers 64 4K;

    # Disable gzip to avoid the removal of the ETag header
    gzip off;

    # Uncomment if your server is build with the ngx_pagespeed module
    # This module is currently not supported.
    #pagespeed off;

    rewrite ^/caldav(.*)$ /remote.php/caldav$1 redirect;
    rewrite ^/carddav(.*)$ /remote.php/carddav$1 redirect;
    rewrite ^/webdav(.*)$ /remote.php/webdav$1 redirect;
```

```

index index.php;
error_page 403 /core/templates/403.php;
error_page 404 /core/templates/404.php;

location = /robots.txt {
    allow all;
    log_not_found off;
    access_log off;
}

location ~ ^/(?!\.htaccess|data|config|db_structure\.xml|README){
    deny all;
}

location / {
    # The following 2 rules are only needed with webfinger
    rewrite ^/.well-known/host-meta /public.php?service=host-meta last;
    rewrite ^/.well-known/host-meta.json /public.php?service=host-meta-json last;

    rewrite ^/.well-known/carddav /remote.php/carddav/ redirect;
    rewrite ^/.well-known/caldav /remote.php/caldav/ redirect;

    rewrite ^(/core/doc/[^/]+/)$ $1/index.html;

    try_files $uri $uri/ /index.php;
}

location ~ \.php(?:$|/) {
    fastcgi_split_path_info ^(.+\.php)(/.+)$;
    include fastcgi_params;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    fastcgi_param PATH_INFO $fastcgi_path_info;
    fastcgi_param HTTPS on;
    fastcgi_pass php-handler;
}

# Optional: set long EXPIRES header on static assets
location ~* \.(?:jpg|jpeg|gif|bmp|ico|png|css|js|swf)$ {
    expires 30d;
    # Optional: Don't log access to assets
    access_log off;
}
}

```

由於 nginx 的 nginx.conf 設定會 include /etc/nginx/sites-enabled 內的 link，所以我們要把/etc/nginx/sites-available/owncloud 做 link 到/etc/nginx/sites-enabled 去。

```
#cd /etc/nginx/sites-enabled/
```

```
#ln -s /etc/nginx/sites-available/owncloud
```

Reload nginx 設定：

重起 nginx 讓 nginx 使用新設定。

```
#sudo service nginx restart
```

2.產生伺服器私人憑證 key 與 crt:

這邊需要使用 root 身份所以請用 su 切換到 root 或是改用 sudo 來建立私人憑證，過程中需要 key 密碼的部份都不要輸入方便 owncloud 使用。

產生私鑰：

```
#sudo openssl genrsa -out owncloud.key 2048
```

這邊使用 RSA 來建立一個 2048 位元的 key 名稱為 owncloud <=對應到 nginx 的設定檔！！

產生 CSR:

```
#sudo openssl req -new -key owncloud.key -out owncloud.csr
```

產生自我簽署的金鑰：

```
#sudo openssl x509 -req -days 3650 -in owncloud.csr -signkey owncloud.key -out owncloud.crt
```

複製到 nginx 設定檔內容指定位置：

以下是我們剛剛設定 nginx 時 ssl 憑證位置。

```
ssl_certificate /etc/ssl/nginx/owncloud.crt;  
ssl_certificate_key /etc/ssl/nginx/owncloud.key;
```

首先建立資料夾：

```
#sudo mkdir /etc/ssl/nginx
```

複製到/etc/ssl/nginx 下：

```
#sudo cp owncloud.key owncloud.crt /etc/ssl/nginx/
```

3.PHP-fpm 設定：

開啟 cgi.fix_pathinfo 功能：

(5 年前這邊有漏洞，現在應該修正掉了吧！尚未查證)

```
#sudo vim /etc/php5/fpm/php.ini
```

```
; cgi.fix_pathinfo provides *real* PATH_INFO/PATH_TRANSLATED support for CGI. PHP's  
; previous behaviour was to set PATH_TRANSLATED to SCRIPT_FILENAME, and to not grok  
; what PATH_INFO is. For more information on PATH_INFO, see the cgi specs. Setting  
; this to 1 will cause PHP CGI to fix its paths to conform to the spec. A setting  
; of zero causes PHP to behave as before. Default is 1. You should fix your scripts  
; to use SCRIPT_FILENAME rather than PATH_TRANSLATED.  
; http://php.net/cgi.fix-pathinfo  
cgi.fix_pathinfo=1
```

使用 TCP/IP 做入口：

由於我們在 nginx 設定使用 tcp/ip 作為入口所以 php-fpm 這邊的 listen 也要改成聽 tcp/ip

以下是我們剛剛 nginx 設定內容

```
upstream php-handler {  
    server 127.0.0.1:9000;  
    #server unix:/var/run/php5-fpm.sock;  
}
```

修改 pool.conf:

```
#sudo vim /etc/php5/fpm/pool.d/www.conf
```

```
; The address on which to accept FastCGI requests.  
; Valid syntaxes are:  
; 'ip.add.re.ss:port' - to listen on a TCP socket to a specific address on  
;                      a specific port;  
; 'port'              - to listen on a TCP socket to all addresses on a  
;                      specific port;  
; '/path/to/unix/socket' - to listen on a unix socket.  
; Note: This value is mandatory.  
listen = 127.0.0.1:9000
```

Reload php-fpm:

```
#sudo /etc/init.d/php5-fpm reload
```

```
#sudo service php5-fpm restart
```

4. 解開 owncloud:

我們將下載的 owncloud 壓縮檔解開到 nginx 我們設定的位置。

```
# Path to the root of your installation  
root /var/www/owncloud/;
```

由於 owncloud 解開時本身會產生一個名為『owncloud』的資料夾所以只要指定解壓縮到/var/www 就可以。

在這邊我是在官網下 zip 檔(<https://owncloud.org/install/>)。

```
#sudo unzip owncloud-8.0.4.zip -d /var/www/
```

就應該會在/var/www 下看到一個名為 owncloud 的資料夾裡面放了 owncloud 的 web 檔案。

變更資料夾擁有者：

將 owncloud 資料夾改成為 www-data 使用者與群組

```
#sudo chown -R www-data:www-data /var/www/owncloud
```

5.測試 php-fpm 是否正常運作：


由於我們尚未設定 mysql 帳號所以先測試剛剛 php 設定是否有錯誤！

我們在/var/www/owncloud 下建立一個簡單的 test.php 檔案並寫入下列內容：

#sudo vim /var/www/owncloud/test.php

```
<?php
    phpinfo();
?>
```

在用瀏覽器開開看，php 正常運作的話應該可以看到下列畫面：

PHP Version 5.5.9-1ubuntu4.9	
	
System	Linux cloverVG 3.13.0-55-generic #92-Ubuntu SMP Sun Jun 14 18:32:20 UTC 2015 x86_64
Build Date	Apr 17 2015 11:43:17
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/fpm
Loaded Configuration File	/etc/php5/fpm/php.ini
Scan this dir for additional .ini files	/etc/php5/fpm/conf.d
Additional .ini files parsed	/etc/php5/fpm/conf.d/05-opcache.ini, /etc/php5/fpm/conf.d/10-pdo.ini, /etc/php5/fpm/conf.d/20-curl.ini, /etc/php5/fpm/conf.d/20-gd.ini, /etc/php5/fpm/conf.d/20-imagick.ini, /etc/php5/fpm/conf.d/20-intl.ini, /etc/php5/fpm/conf.d/20-json.ini, /etc/php5/fpm/conf.d/20-mysql.ini, /etc/php5/fpm/conf.d/20-mysqli.ini, /etc/php5/fpm/conf.d/20-pdo_mysql.ini
PHP API	20121113
PHP Extension	20121212
Zend Extension	220121212
Zend Extension Build	API220121212,NTS
PHP Extension Build	API20121212,NTS
Debug Build	no
Thread Safety	disabled

若出現 504 錯誤訊息，表示在步驟 3php-fpm 設定 /etc/php5/fpm/pool.d/www.conf 內的 listen 與步驟 1 設定 nginx 的 php-handler 沒相對應！

```
upstream php-handler {
    server 127.0.0.1:9000;
    #server unix:/var/run/php5-fpm.sock;
}
```

步驟 1

6.替 owncloud 建立一個 MySQL 的帳號與專屬資料庫：

首先先登入 MySQL：

```
#mysql -u root -p
```

輸入密碼登入 mysql command line。

建立資料庫：

```
mysql> CREATE DATABASE owncloud;
```

確認資料庫。

```
mysql> SHOW DATABASES;
```

看看看看 owncloud 這個 databases 有沒有在上面。

建立使用者：

建立一個使用者為 owncloud、密碼為 12345678 並管理名為 owncloud 的資料庫。

```
mysql> GRANT ALL PRIVILEGES ON owncloud.* TO owncloud@localhost IDENTIFIED BY '12345678';
```

立即更新：

```
mysql> FLUSH PRIVILEGES;
```

接著我們離開資料庫用 owncloud 這個帳號登入看看

```
mysql> exit
```

```
#mysql -u owncloud -p
```

7.開啟 owncloud 首頁並且設定：

開啓瀏覽器輸入 <https://owncloud.biotrump.com> 應該可以看到憑證警告，由於我們是使用私人憑證，所以把這個憑證加到信任區，這邊照個瀏覽器上面的提示設定就可以了。

※如果你是用 virtual machine 安裝 owncloud，請在 host 端的/etc/hosts 寫入相對應的 ip 與 hostname。EX：192.168.2.3 owncloud.biotrump.com

接著會看到第一次登入 owncloud 的畫面：

The screenshot shows the ownCloud installation wizard interface. At the top is the ownCloud logo. Below it, the text '建立一個管理員帳號' (Create an administrator account) is displayed. The form has several sections:

- Username:** A field containing 'admin'.
- Password:** A field with masked characters (dots) and a toggle icon. A red arrow points to this field with the annotation '建立一個管理員帳號' (Create an administrator account).
- Storage space and database:** A section with a dropdown menu for '儲存空間和資料庫' (Storage space and database).
- Database storage location:** A field containing '/var/www/owncloud/data'.
- Database settings:** A section with the text '設定資料庫' (Set database) and '剩下 MySQL/MariaDB 可使用.' (MySQL/MariaDB can be used).
- Database name:** A field containing 'owncloud'. A red arrow points to this field with the annotation '輸入之前設定的資料庫帳號與資料庫名稱' (Enter the database account and name set before).
- Database user:** A field containing 'owncloud'.
- Database host:** A field containing 'localhost'.
- Completion button:** A blue button labeled '完成設定' (Finish setup).
- Footer:** The text 'ownCloud - 由您控制的網路服務' (ownCloud - Network service controlled by you).

8.讓 nginx 自動將 http://owncloud.biotrump.com 轉到 https :

由於這個方法使用者必須在網址開頭打上 https，否則無法正確進到 owncloud，因此我們在 nginx 設定多一個轉換動作。

讓 nginx 讀到封包內的 host 為 owncloud.biotrump.com 並且為 80 port 的就自動轉到 443 https 去。

建立 redirect 設定檔：

#sudo vim /etc/nginx/sites-available/redirect-owncloud

輸入下列內容：

```
server{  
    listen 80;  
    server_name owncloud.biotrump.com  
  
    root /var/www/owncloud;  
  
    rewrite ^(.*)$ https://$host$1 permanent;  
  
    error_page 404 https://owncloud.biotrump.com;  
}
```

存檔後一樣必須在/etc/nginx/sites-enabled/ 下建立 link:

#ln -s /etc/nginx/sites-available/redirect-owncloud /etc/nginx/sites-enabled/

重起 nginx:

#sudo service nginx restart

接著我們在瀏覽器輸入 **http://owncloud.biotrump.com** 應該就會自動轉到 **https** 的網域去。