

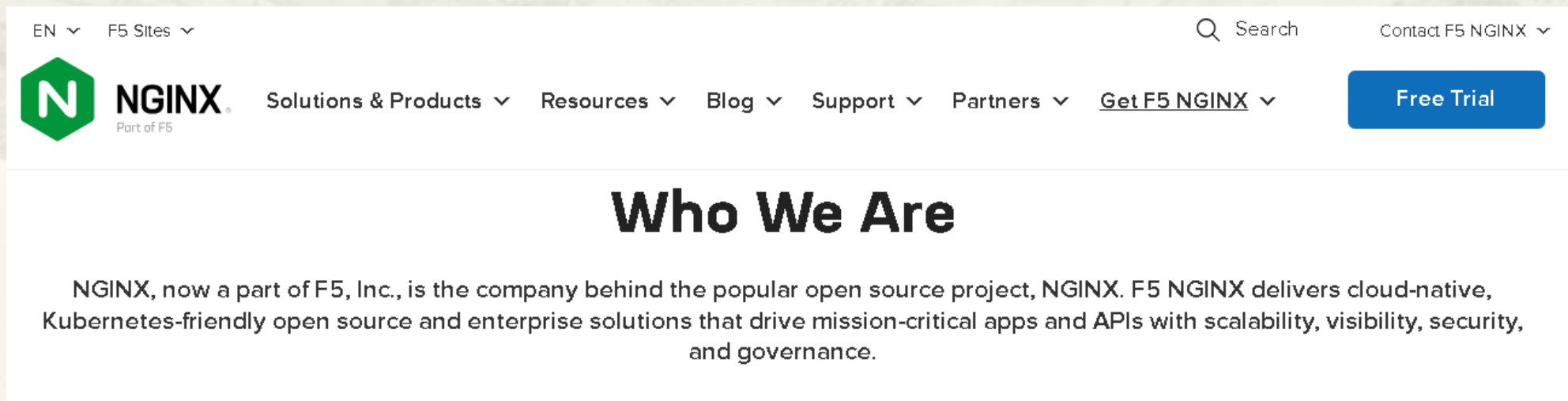
NGINX 簡介

臺大計資中心網路組
游子興

davisyou@ntu.edu.tw
02-33665008

NGINX

- * NGINX pronounced as “ Engine X ”.
- * 常用於 web server, reverse proxy, load balancer, HTTP caching.
- * a business unit of F5 Networks (2019).
 - * <https://www.nginx.com/about/>



The screenshot shows the top navigation bar of the NGINX website. It includes a language selector (EN), a site selector (F5 Sites), a search bar, and a contact link (Contact F5 NGINX). The main navigation menu contains links for Solutions & Products, Resources, Blog, Support, Partners, and a link to Get F5 NGINX. A prominent blue button for 'Free Trial' is also visible. Below the navigation bar, the section 'Who We Are' is displayed, featuring the NGINX logo and a paragraph describing the company's role as a provider of cloud-native, Kubernetes-friendly open source and enterprise solutions.

EN ▾ F5 Sites ▾

Q Search Contact F5 NGINX ▾

 **NGINX**
Part of F5

Solutions & Products ▾ Resources ▾ Blog ▾ Support ▾ Partners ▾ Get F5 NGINX ▾

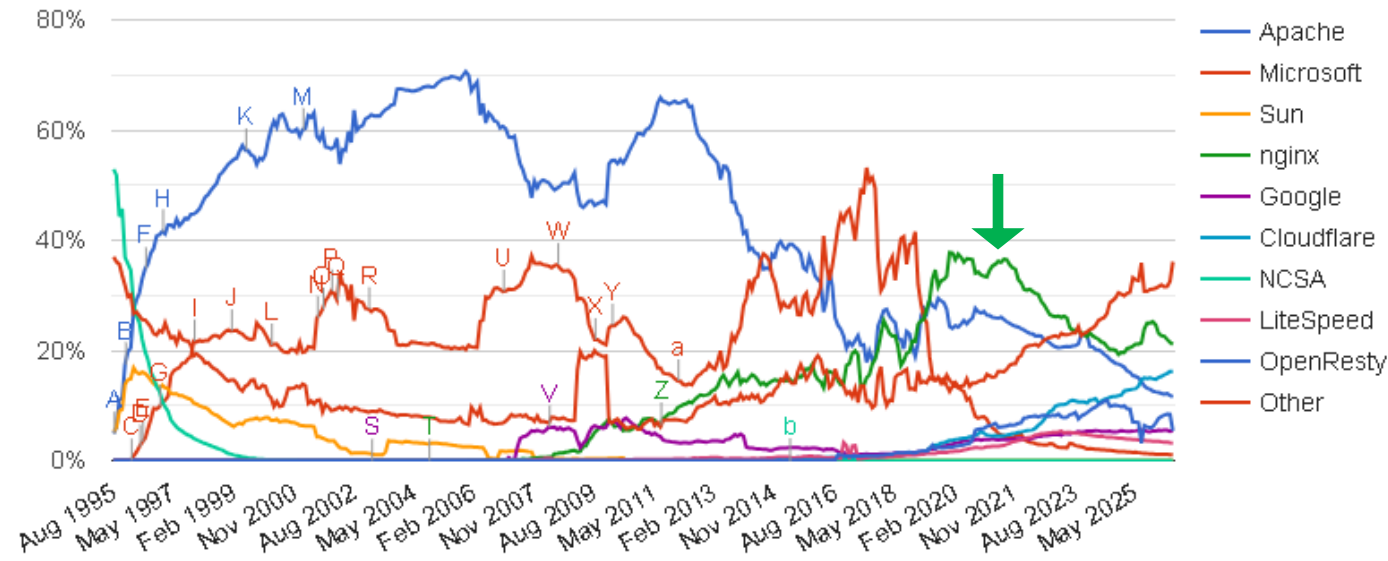
Free Trial

Who We Are

NGINX, now a part of F5, Inc., is the company behind the popular open source project, NGINX. F5 NGINX delivers cloud-native, Kubernetes-friendly open source and enterprise solutions that drive mission-critical apps and APIs with scalability, visibility, security, and governance.

Web Server: Market Share

Web server developers: Market share of all sites



Developer	May 2026	Percent	June 2026	Percent	Change
nginx	315,290,021	21.47%	314,307,194	21.10%	-0.37
Cloudflare	237,122,135	16.15%	241,225,970	16.20%	0.05
Apache	173,358,082	11.81%	172,109,487	11.56%	-0.25
Google	78,187,575	5.33%	80,894,625	5.43%	0.11



VS



- Process Driven Approach.
- Creates a new thread for each request.



BASIC
ARCHITECTURE

- Event-Driven approach.
- Handles multiple requests within one thread.

| STATIC CONTENT

- Serves static content using the file-based method.

| DYNAMIC CONTENT

- Processes dynamic content within the server.



PERFORMANCE

ServerGuy

| STATIC CONTENT

- Nginx serves static resources without using PHP.

| DYNAMIC CONTENT

- It doesn't process dynamic content.

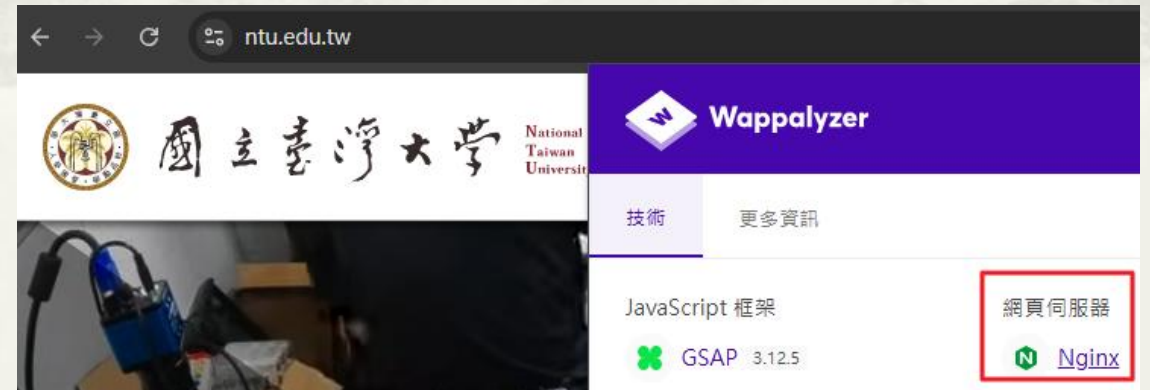
Performance : Static Content

- * Apache
 - * Serves static content using the file-based method.
- * NGINX
 - * Instead of creating new processes for each connection, Nginx uses a single master process and multiple worker processes, managing connections efficiently through an event loop.

APACHE
10770 req/s
@ 512 PARALLEL REQUESTS

NGINX
20232 req/s
@ 512 PARALLEL REQUESTS

<https://zenocloud.io/blog/apache-vs-nginx/>



NGINX Plus vs. NGINX(open source)

商業版 vs. 免費版

NGINX Plus is the closed source,
commercial version of NGINX.

Features	NGINX Open Source	NGINX Plus	NGINXaaS for Azure
Web server and reverse proxy			
Origin server for static content	✓	✓	
Reverse proxy: HTTP, FastCGI, memcached, SCGI, uwsgi	✓	✓	HTTP, FastCGI
HTTP/2 gateway	✓	✓	✓
gRPC proxy	✓	✓	✓
HTTP/2 server push	✓	✓	✓
HTTP/3 over QUIC	✓	✓	

Features	NGINX Open Source	NGINX Plus	NGINXaaS for Azure
Load balancer			
HTTP and TCP/UDP support	✓	✓	✓
Layer 7 request routing	✓	✓	✓
Session persistence	✓	✓	Sticky Cookie and Sticky Route Only
Active health checks		✓	✓
DNS service-discovery integration		✓	✓

<https://docs.nginx.com/nginxaaS/azure/overview/feature-comparison/>

Features	NGINX Open Source	NGINX Plus	NGINXaaS for Azure
----------	-------------------	------------	--------------------

High availability (HA)			
------------------------	--	--	--

Active-active		✓	✓
---------------	--	---	---



Active-passive		✓	
----------------	--	---	--



Configuration synchronization across cluster		✓	✓
---	--	---	---



State sharing: sticky-learn session persistence, rate limiting, key-value stores		✓	✓
--	--	---	---



<https://www.nginx.com/products/nginx/compare-models/>

NGINX Module

* HTTP Basic Module

- * ngx_http_core_module
- * ngx_http_browser_module
- * ngx_http_headers_module
- * ngx_http_auth_basic_module
- * ngx_http_auth_request_module
- * ngx_http_rewrite_module
- * ngx_http_sub_module
- * ngx_http_realip_module

* HTTP Forward Proxy

- * ngx_http_proxy_connect_module

* Reverse Proxy

- * ngx_http_proxy_module: L7 HTTP
- * ngx_mail_proxy_module: L7 Mail
- * ngx_stream_proxy_module: L4

* Load Balance

- * ngx_http_upstream_module: L7 HTTP
- * ngx_stream_upstream_module: L4

NGINX LAB

大綱

- * 環境準備
- * Case1. Virtual Host
- * Case2. Let's Encrypt 免費憑證: Authenticator & Installer
- * Case3. Let's Encrypt 免費憑證: Authenticator Only
- * Case4. HTTP 301/302 Redirect

環境準備

Ubuntu 26.04

- * Install nginx

- * apt install nginx
- * systemctl status nginx
- * systemctl restart nginx

- * Test Config File: /etc/nginx/nginx.conf

- * nginx -t

```
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

/etc/nginx/nginx.conf

設定架構

```
# 註解
user www-data;
error_log /var/log/nginx/error.log;
include /etc/nginx/modules-enabled/*.conf; # main level

events {
    worker_connections 1024; #每個 worker process 最大連線數.
}

mail {
}
```

```
http {
    access_log /var/log/nginx/access.log;
    # Layer7 Reverse proxy
    server {
        listen 80;
        server_name proxy01.buda.idv.tw;
        location / {
            proxy_pass http://172.16.0.7;
        }
    }
    # Layer7 Load Balance
    server {
        listen 80;
        server_name proxy02.buda.idv.tw;
        location / {
            proxy_pass https://backend_https;
        }
    }
    upstream backend_https {
        server www.ntu.edu.tw:443;
        server www.tp1rc.edu.tw:443;
    }
    include /etc/nginx/conf.d/*.conf; # http level
    include /etc/nginx/sites-enabled/*; # http level
}
```

/etc/nginx/nginx.conf

設定架構

```
stream {  
    # Layer4 Reverse proxy  
    server {  
        listen 8080;  
        proxy_pass 172.16.0.10:80;;  
    }  
    # Layer4 Load Balance  
    server {  
        listen 8081;  
        proxy_pass backend_tcp_80;  
    }  
    upstream backend_tcp_80 {  
        server 172.16.0.21:80;  
        server 172.16.0.7:80;  
    }  
}
```

Config File

執行順序

- * 1. /etc/nginx/nginx.conf
- * 2. /etc/nginx/modules-enabled/*.conf
- * 3. /etc/nginx/conf.d/*.conf
- * 4. /etc/nginx/sites-enabled/*

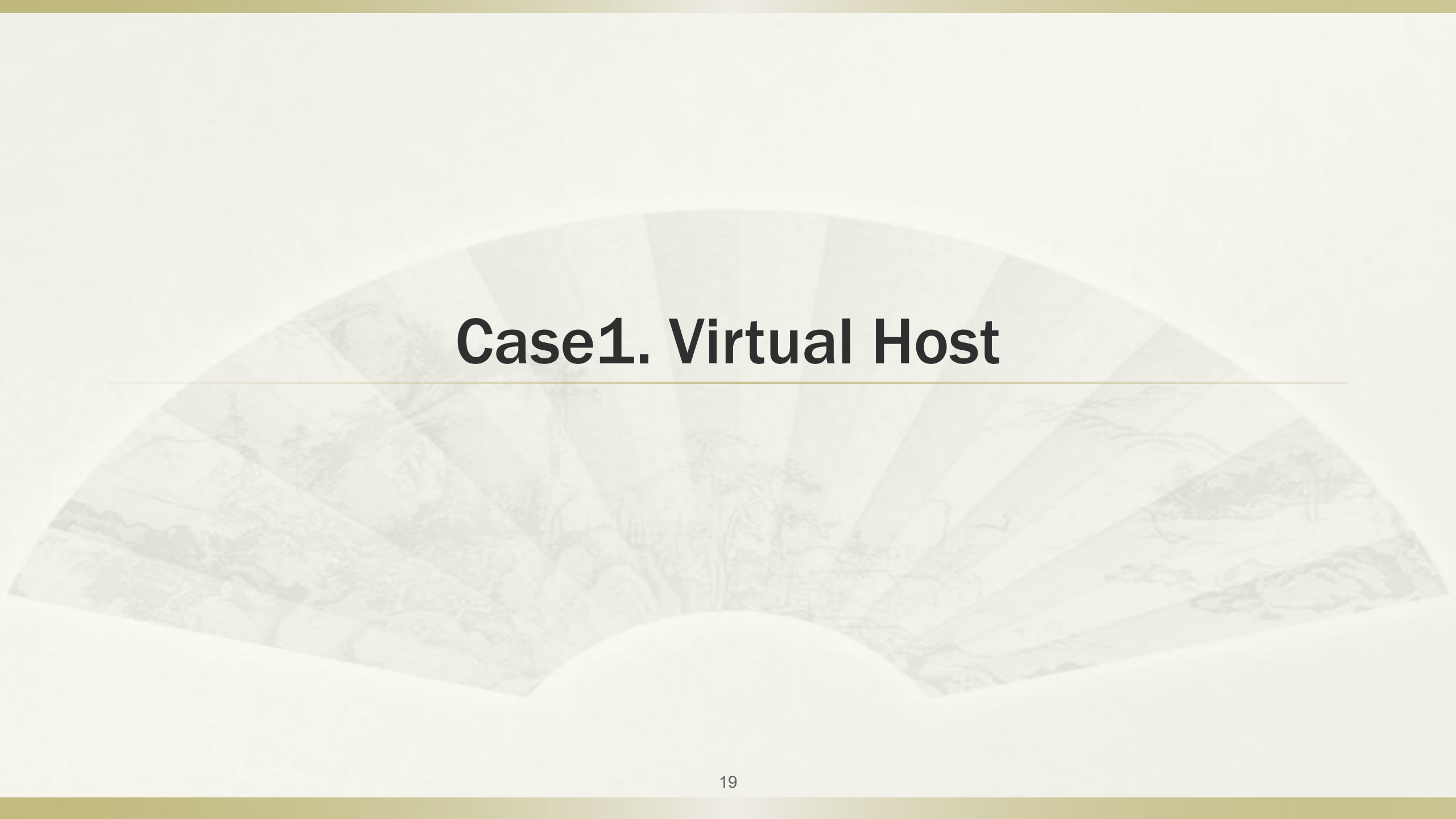
Config File

目錄結構

- * `/etc/nginx/conf.d/` -- 預設無檔案
- * `/etc/nginx/sites-available/default` -- 預設僅此檔案
- * `/etc/nginx/sites-enabled/` -- symbolic link to `/etc/nginx/sites-available/`
 - * `default -> /etc/nginx/sites-available/default` -- 預設僅此檔案
 - * 產生:
 - * `~# ln -s /etc/nginx/sites-available/my.domain.conf /etc/nginx/sites-enabled/`
 - * 移除:
 - * `~# rm /etc/nginx/sites-enabled/my.domain.conf`
- * `/etc/nginx/modules-available/` -- 預設無檔案
- * `/etc/nginx/modules-enabled/` -- symbolic link to `/etc/nginx/modules-available/`

/etc/nginx/sites-available/default

```
server {  
    # default_server -- to become the default server for the specified address:port  
    # If none of the directives have the default_server parameter then the first server will be the default server.  
    listen 80 default_server;  
    listen [::]:80 default_server;  
  
    root /var/www/html;  
    index index.html index.htm index.nginx-debian.html;  
    server_name _;  
  
    location / {  
        # First attempt to serve request as file, then as directory, then fall back to displaying a 404.  
        try_files $uri $uri/ =404;  
    }  
}
```

The background of the slide features a large, semi-circular fan. The fan's surface is decorated with a traditional East Asian landscape painting, likely a woodblock print or ink wash, depicting mountains, trees, and a body of water. The fan is positioned centrally, with its handle pointing downwards. The overall color scheme is muted, with various shades of beige, cream, and light brown.

Case1. Virtual Host

Virtual Host Config File

/etc/nginx/sites-enabled/01.vhost.conf

```
# Virtual Host: nginx01.buda.idv.tw
server {
    listen 80;
    server_name nginx01.buda.idv.tw;

    root /var/www/html/folder1;
    index index.html index.htm;
}

# Virtual Host: nginx02.buda.idv.tw
server {
    listen 80;
    server_name nginx02.buda.idv.tw;

    root /var/www/html/folder2;
    index index.html index.htm;
}
```

```
# Virtual Host: nginx03.buda.idv.tw
server {
    listen 140.112.237.11:80;
    server_name nginx03.buda.idv.tw;

    root /var/www/html/folder3;
    index index.html index.htm;
}
```

網頁檔案

/var/www/html/folder1/index.html

<H1>nginx01.buda.idv.tw</H1>

/var/www/html/folder2/index.html

<H1>nginx02.buda.idv.tw</H1>

/var/www/html/folder3/index.html

<H1>nginx03.buda.idv.tw</H1>

Testing

```
C:\Users\user>nslookup nginx.buda.idv.tw
伺服器: dns.google
Address: 8.8.8.8

未經授權的回答:
名稱: nginx.buda.idv.tw
Address: 140.112.237.10

C:\Users\user>nslookup nginx01.buda.idv.tw
伺服器: dns.google
Address: 8.8.8.8

未經授權的回答:
名稱: nginx01.buda.idv.tw
Address: 140.112.237.10

C:\Users\user>nslookup nginx02.buda.idv.tw
伺服器: dns.google
Address: 8.8.8.8

未經授權的回答:
名稱: nginx02.buda.idv.tw
Address: 140.112.237.10
```

- * <http://nginx.buda.idv.tw/>
- * <http://nginx01.buda.idv.tw/>
- * <http://nginx02.buda.idv.tw/>
- * <http://nginx03.buda.idv.tw/>
- * <http://140.112.237.10/>
- * ???

Case2. Let's Encrypt 免費憑證

Authenticator & Installer

Let's Encrypt

- * Let's Encrypt 是免費、自動化和開放的憑證頒發機構，由非營利組織網路安全研究小組 (Internet Security Research Group, ISRG) 營運
- * 旨在以自動化流程消除手動建立和安裝憑證的複雜流程，並推廣使全球資訊網伺服器加密服務，為安全網站提供免費的SSL/TLS憑證
- * Limitations
 - * Hostname on certificate can contain up to 10 levels of subdomains.
 - * Duplicate certificate limit of 5 certificates ↗ per week.
- * <https://letsencrypt.org/sponsors/>
- * 參考資料
 - * <https://letsencrypt.org/getting-started/>



Automatic Certificate Management Environment (ACME) Protocol

- * Designed by the Internet Security Research Group (ISRG) for their Let's Encrypt service.
- * Use ACME protocol to verify that you control a given domain name and to issue you a certificate.
- * To get a Let's Encrypt certificate, you'll need to choose a ACME Client software.
 - * <https://letsencrypt.org/docs/client-options/>
 - * Linux: Certbot (官方推薦)
 - * Windows: win-acme (個人推薦)

HTTPS 免費憑證安全性評估

方法	Generate Certificate	Create Private Key	Web Server Config	Auto Renew	安全性
Create From CSR	Web(zerossl.com) or ACME Client	User	User	Not Support (Auto/Manual Run per 90 days)	★★★
Authenticator	Web(zerossl.com) or ACME Client	Web(zerossl.com) or ACME Client	User	Support	★★
Authenticator & Installer	ACME Client	ACME Client	ACME Client	Support	★

- * Will Certbot generate or store the private keys for my certificates on Let's Encrypt's servers?
 - * **No. Never.**
 - * The private key is always **generated and managed on your own servers**, not by the Let's Encrypt certificate authority.

Certbot Plugins 分類

- * Authenticator

- * automatically perform the steps to prove that you control the domain to get the certificate.

- * Installer

- * automatically modify web server's configuration to install the certificate.

- * Some plugins are both authenticator and installer.

Domain Control Validation (DCV)

- * The process for CA to prove they have control over that domain.
- * HTTP based validation
 - * Support Non-wildcard certificates
- * DNS TXT-based validation
 - * Support Non-wildcard and Wildcard certificates

Certbot Plugins (Official)

Plugin	Auth	Inst	Notes	Challenge types (port)
Apache	Y	Y	For Apache.	http (80)
Nginx	Y	Y	For Nginx.	http (80)
Webroot	Y	N	Get certificate by writing to the webroot directory of an already running webserver .	http (80)
Standalone	Y	N	For systems without webserver. (Certbot 即是 Webserver)	http (80)
DNS plugins	Y	N	Get certificate by modifying DNS records to prove you have control over a domain. ✂Domain validation is the only way to get wildcard certificates.	dns (53)
Manual	Y	N		http (80) or dns (53)

<https://eff-certbot.readthedocs.io/en/stable/using.html#getting-certificates-and-choosing-plugins>

Install Certbot & Plugin

- * Install certbot
 - * apt install certbot
- * Install Plugin: NGINX
 - * apt install python3-certbot-nginx

Certbot Plugin: NGINX

* ~# certbot --nginx

```
Which names would you like to activate HTTPS for?
We recommend selecting either all domains, or all domains in a VirtualHost/server block.
-----
1: nginx01.buda.idv.tw
2: nginx02.buda.idv.tw
-----
Select the appropriate numbers separated by commas and/or spaces, or leave input
blank to select all options shown (Enter 'c' to cancel):
Requesting a certificate for nginx01.buda.idv.tw and nginx02.buda.idv.tw

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/nginx01.buda.idv.tw/fullchain.pem
Key is saved at: /etc/letsencrypt/live/nginx01.buda.idv.tw/privkey.pem
This certificate expires on 2026-10-30.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.

Deploying certificate
Successfully deployed certificate for nginx01.buda.idv.tw to /etc/nginx/sites-enabled/vhost.conf
Successfully deployed certificate for nginx02.buda.idv.tw to /etc/nginx/sites-enabled/vhost.conf
Congratulations! You have successfully enabled HTTPS on https://nginx01.buda.idv.tw and https://nginx02.buda.idv.tw

-----
If you like Certbot, please consider supporting our work by:
* Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
* Donating to EFF: https://eff.org/donate-le
```

Certbot 自動修改 01.vhost.conf

```
server {  
    server_name nginx01.buda.idv.tw;
```

```
    root /var/www/html/folder1;  
    index index.html index.htm;
```

```
    listen 443 ssl; # managed by Certbot  
    ssl_certificate /etc/letsencrypt/live/nginx01.buda.idv.tw/fullchain.pem; # managed by Certbot  
    ssl_certificate_key /etc/letsencrypt/live/nginx01.buda.idv.tw/privkey.pem; # managed by Certbot  
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot  
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
```

```
server {  
    if ($host = nginx01.buda.idv.tw) {  
        return 301 https://$host$request_uri;  
    } # managed by Certbot
```

http redirect to https

```
    listen 80;  
    server_name nginx01.buda.idv.tw;  
    return 404; # managed by Certbot
```

一張憑證 or 兩張憑證

憑證檢視者 : nginx01.buda.idv.tw

一般(G) 詳細資訊(D)

核發對象

一般名稱 (CN)	nginx01.buda.idv.tw
組織 (O)	<不是憑證的一部分>
組織單位 (OU)	<不是憑證的一部分>

發行者

一般名稱 (CN)	YE2
組織 (O)	Let's Encrypt
組織單位 (OU)	<不是憑證的一部分>

有效期間

發行日期	2026年8月2日 星期日 晚上7:53:54
到期日	2026年10月31日 星期六 晚上7:53:53

憑證檢視者 : nginx01.buda.idv.tw

一般(G) 詳細資訊(D)

憑證階層

- ▼ Root YE
 - ▼ YE2
 - nginx01.buda.idv.tw

憑證欄位

- 憑證金鑰用途
- 擴充金鑰使用方法
- 憑證基本限制
- 憑證主體金鑰識別碼
- 憑證授權單位金鑰識別碼
- 授權單位資訊存取
- 憑證主體替代名稱**

欄位值

非重要

- DNS 名稱 : nginx01.buda.idv.tw
- DNS 名稱 : nginx02.buda.idv.tw

How to get 憑證 of nginx.buda.idv.tw?

- * nano /etc/nginx/sites-enabled/default

```
#server_name _;
```

```
server_name nginx.buda.idv.tw;
```

- * certbot --nginx

```
We recommend selecting either all domains, or all domains in a VirtualHost/server block.
```

```
- - - - -  
1: nginx.buda.idv.tw  
2: nginx01.buda.idv.tw  
3: nginx02.buda.idv.tw  
- - - - -
```

完成

憑證檢視者：nginx.buda.idv.tw

一般(G)

詳細資訊(D)

憑證階層

▼ ISRG Root X2

▼ Root YE

▼ YE1

憑證欄位

憑證主體金鑰識別碼

憑證授權單位金鑰識別碼

授權單位資訊存取

憑證主體替代名稱

憑證原則

CRL 發布點

欄位值

非正式

DNS 名稱：nginx.buda.idv.tw

DNS 名稱：nginx01.buda.idv.tw

DNS 名稱：nginx02.buda.idv.tw

Certbot Plugin: NGINX

總結

- * 優點: 方便快捷
- * 缺點: Certbot 自動修改 .conf

Case3. Let's Encrypt 免費憑證 Authenticator Only

Plugin: Webroot

Plugin: Webroot

- * Already have web server running(port 80)
 - * 優點: 支援任何 Web Server
- * 需提供目前 Web Server Webroot 路徑
- * 憑證到期會自動 renewal，但需自行重啟 Web Server

Plugin: Webroot

- * Interactive

- * certbot certonly --webroot

```
Please enter the domain name(s) you would like on your certificate (comma and/or  
space separated) (Enter 'c' to cancel): nginx.buda.idv.tw 需人工輸入
```

```
Requesting a certificate for nginx.buda.idv.tw
```

```
Input the webroot for nginx.buda.idv.tw: (Enter 'c' to cancel): /var/www/html/
```

```
Successfully received certificate.
```

```
Certificate is saved at: /etc/letsencrypt/live/nginx.buda.idv.tw/fullchain.pem
```

```
Key is saved at: /etc/letsencrypt/live/nginx.buda.idv.tw/privkey.pem
```

```
This certificate expires on 2026-10-30.
```

```
These files will be updated when the certificate renews.
```

```
Certbot has set up a scheduled task to automatically renew this certificate in the background.
```

Plugin: Webroot

- * Non-Interactive

- * `certbot certonly --webroot --webroot-path /var/www/html/ --agree-tos --email davisyou@ntu.edu.tw --noninteractive -d nginx.buda.idv.tw`

Webroot

NGINX Access Log

- * tail /var/log/nginx/access.log
 - * 23.178.112.108 - - [01/Aug/2026:16:15:48 +0800] "GET /.well-known/acme-challenge/hEEtYSZKzWI2cKongDVaQujA08pgbNzGKMU6Q7a55Nw HTTP/1.1" 200 87 "-" "Mozilla/5.0 (compatible; Let's Encrypt validation server; +https://www.letsencrypt.org)"
 - * 35.85.44.91 - - [01/Aug/2026:16:15:49 +0800] "GET /.well-known/acme-challenge/hEEtYSZKzWI2cKongDVaQujA08pgbNzGKMU6Q7a55Nw HTTP/1.1" 200 87 "-" "Mozilla/5.0 (compatible; Let's Encrypt validation server; +https://www.letsencrypt.org)"
 - * 47.128.224.138 - - [01/Aug/2026:16:15:49 +0800] "GET /.well-known/acme-challenge/hEEtYSZKzWI2cKongDVaQujA08pgbNzGKMU6Q7a55Nw HTTP/1.1" 200 87 "-" "Mozilla/5.0 (compatible; Let's Encrypt validation server; +https://www.letsencrypt.org)"
 - * 3.148.102.240 - - [01/Aug/2026:16:15:49 +0800] "GET /.well-known/acme-challenge/hEEtYSZKzWI2cKongDVaQujA08pgbNzGKMU6Q7a55Nw HTTP/1.1" 200 87 "-" "Mozilla/5.0 (compatible; Let's Encrypt validation server; +https://www.letsencrypt.org)"
 - * 16.16.253.91 - - [01/Aug/2026:16:15:50 +0800] "GET /.well-known/acme-challenge/hEEtYSZKzWI2cKongDVaQujA08pgbNzGKMU6Q7a55Nw HTTP/1.1" 200 87 "-" "Mozilla/5.0 (compatible; Let's Encrypt validation server; +https://www.letsencrypt.org)"

憑證更新 自動

- * 每 12小時執行一次

- * /etc/cron.d/certbot

SHELL=/bin/sh

PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

0 */12 * * * root test -x /usr/bin/certbot -a \! -d
/run/systemd/system && perl -e 'sleep int(rand(43200))' && certbot
-q renew --no-random-sleep-on-renew

憑證更新後

How to automatic restart Web Server

- * `/etc/letsencrypt/renewal-hooks/`
 - * `./deploy/` – run for each Certificate successful renewal.
 - * `./pre/` – run before renewal.
 - * `./post/` – run after renewal(successful or not), only once per certbot renew command invocation.
- * 範例

```
~# cd /etc/letsencrypt/renewal-hooks/post/
~# nano 00-reload-nginx.sh
#!/bin/bash
systemctl reload nginx
~# chmod +x 00-reload-nginx.sh
```

Case4. HTTP 301/302 Redirect

Chrome Browser

- * Chrome 特殊功能: 省略 www
- * "www.ntu.edu.tw" vs. "ntu.edu.tw"
- * "www.che.ntu.edu.tw" vs. "che.ntu.edu.tw"

台大化工系首頁 Domain

- * 過去台大化工系首頁: www.che.ntu.edu.tw
- * 某年某日改成: che.ntu.edu.tw
- * 許多舊連結還是使用 www.che.ntu.edu.tw
 - * google 搜尋 "www.che.ntu.edu.tw"
 - * <https://www.google.com/search?q=%22www.che.ntu.edu.tw%22>
 - * 何時可以全部修正 ??

解決方法

- * 再買一張憑證: www.che.ntu.edu.tw
 - * 沒錢
- * 使用 NGINX + Lets'Encrypt 產生HTTPS 免費憑證
 - * 將 www.che.ntu.edu.tw DNS A 記錄指向 NGINX Server 163.28.16.13
 - * NGINX: redirect www.che.ntu.edu.tw to che.ntu.edu.tw

```
C:\Users\user>ping www.che.ntu.edu.tw
```

```
Ping www.che.ntu.edu.tw [163.28.16.13] (使用 32 位元組的資料):  
回覆自 163.28.16.13: 位元組=32 時間=1ms TTL=59
```

```
C:\Users\user>ping che.ntu.edu.tw
```

```
Ping che.ntu.edu.tw [140.112.23.234] (使用 32 位元組的資料):  
回覆自 140.112.23.234: 位元組=32 時間=1ms TTL=60
```

HTTP 301/302 Redirect

```
server {  
    listen 80;  
    listen 443 ssl;  
    server_name www.che.ntu.edu.tw;  
  
    ssl_certificate /etc/letsencrypt/live/www.che.ntu.edu.tw/fullchain.pem;  
    ssl_certificate_key /etc/letsencrypt/live/www.che.ntu.edu.tw/privkey.pem;  
  
    ## Redirect  
    location / {  
        return 301 https://che.ntu.edu.tw$request_uri;  
    }  
}
```



簡報完畢
謝謝