



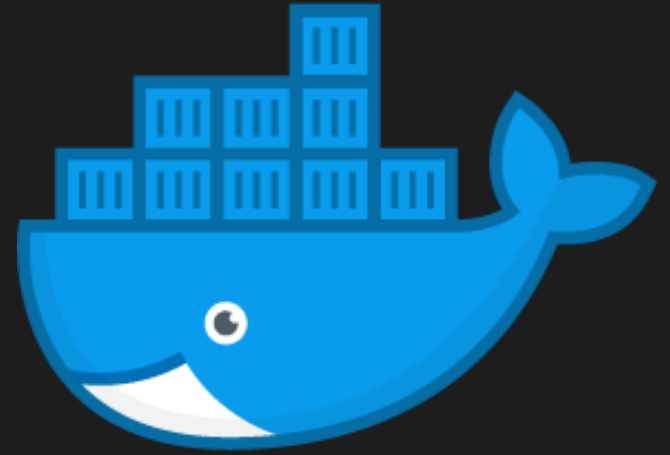
```
[ titangene@stust ~ ] $ cat .profile
```

```
# Author: 郭柏均 / MA690107
```

```
# Modified: 2018-01-26
```

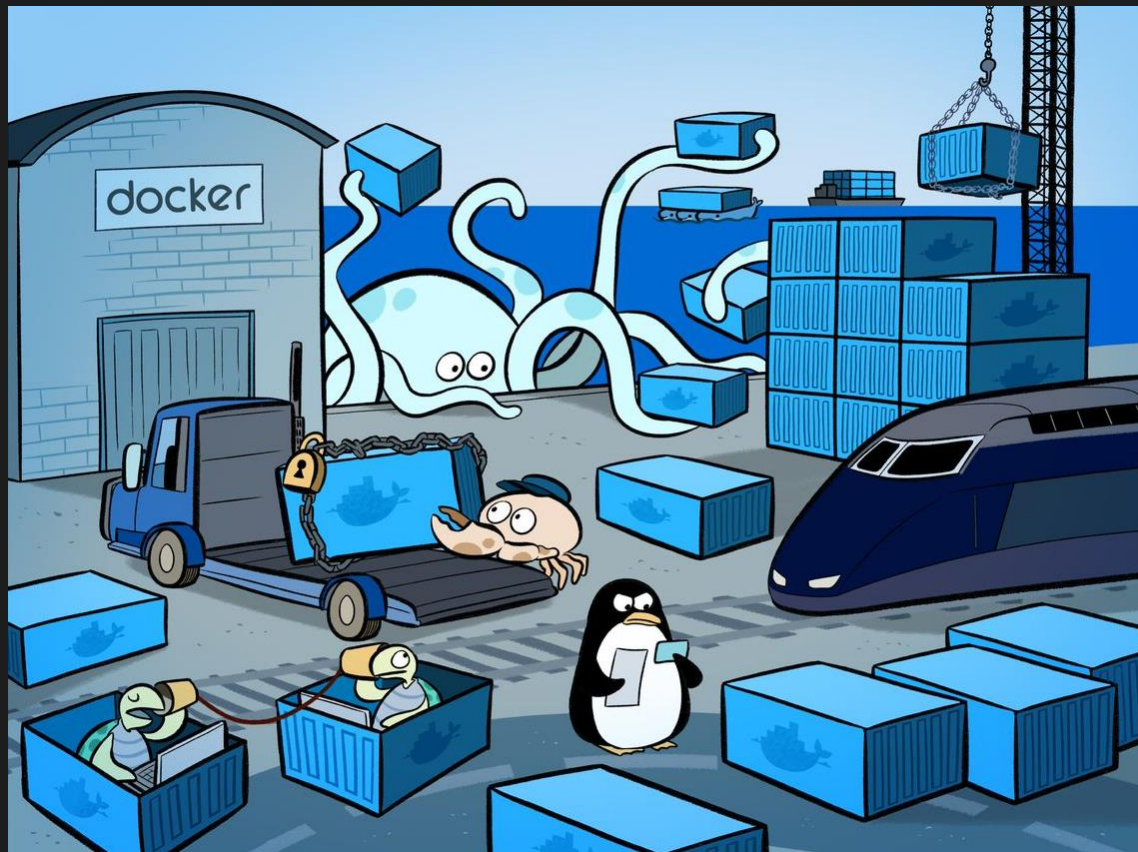
# 大綱

- 什麼是 Docker ?
- Docker 能做什麼 ?
- 為什麼要使用 Docker ?
- Virtualization vs Containerization
- Docker with VM
- Docker 三個基本概念
- Docker 是怎麼運作的 ?
- 安裝 Docker
- Docker 常用指令
- Docker 實戰演練
- 資源



# 什麼是 Docker ?

- Docker 字面上的意思是「碼頭工人」，碼頭上會有打包、運送..等服務，碼頭工人 (Docker) 可以快速的用貨櫃 (Container) 將貨物 (Application) 裝上船。



# 什麼是 Docker ?

- 是**容器化技術**（輕量級的虛擬化技術），可以把你的應用程式連同**環境**一起**打包**，部屬的時候不用擔心環境的問題
- 最初是 Dotcloud 公司內部的 Side-Project（2013 年時，公司改名成 Docker Inc）
- 2013/3 將 Docker **開源**
- 2017/4 所有 Docker 的一切程式碼、工程師資源、所有元件，甚至是 Docker 自身的程式碼，都將屬於新的開源專案 Moby
- 原始碼在 **GitHub** 上進行維護，專案名稱為 [Moby](#)
- 使用 **Go 語言**開發
- 不需要用到 **Hypervisor**，直接利用**硬體效能**

# Docker 能做什麼？

- 可以簡化重建環境的過程、降低維運成本，並且讓開發、測試和正式環境無縫接軌

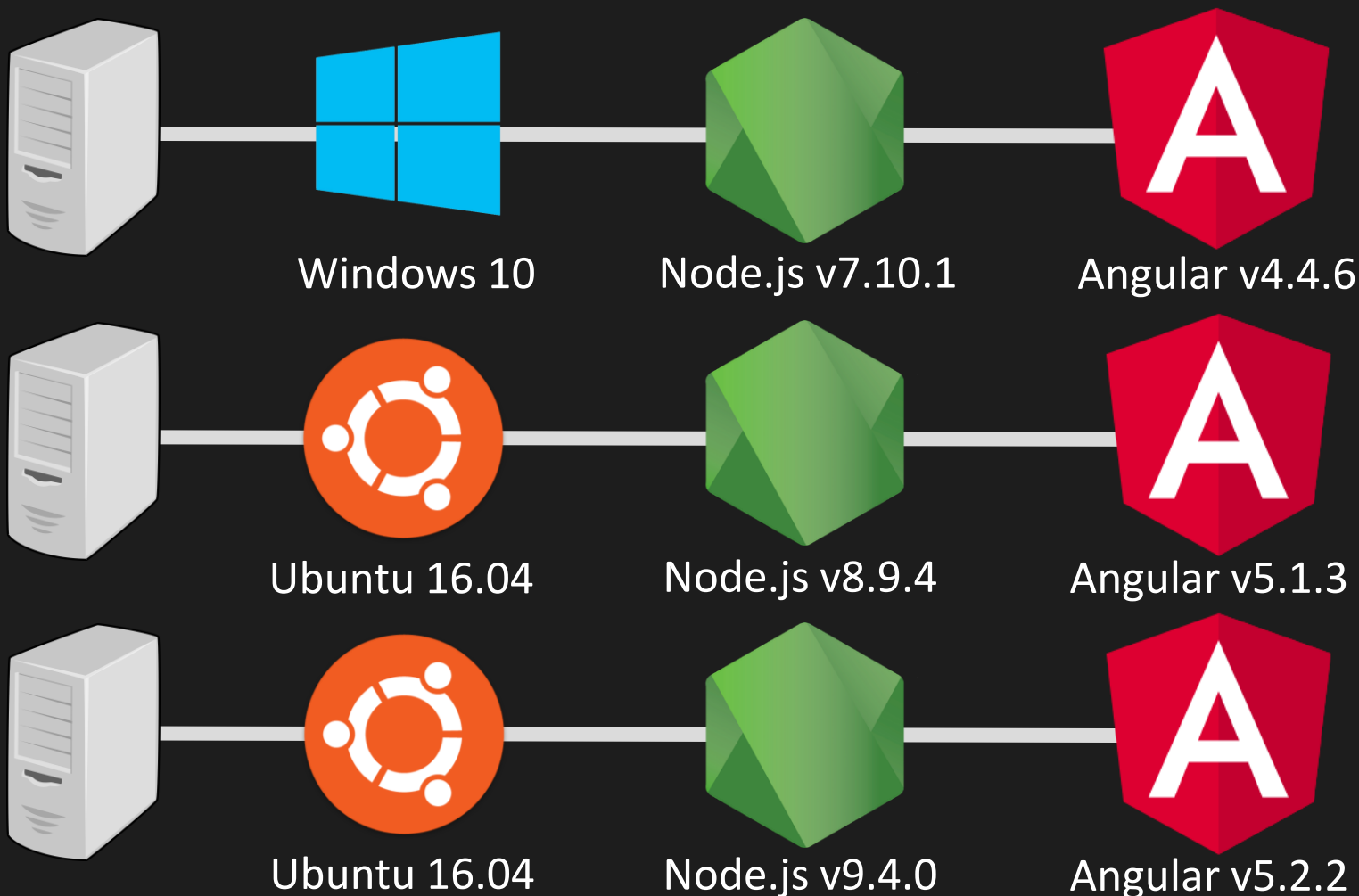


# 為什麼要使用 Docker ?

在說為什麼之前先來講個故事

# 為什麼要使用 Docker ?

- 以 Node.js 為例，每個專案都需要有開發環境



# 傳統架構缺陷

- 維護成本高（機器硬體維修汰換）
- 建置機器複雜（雖然有寫 Shell Script）
- 24 小時 on call（公司隨時都有人加班，有問題就要馬上修）
- 系統擴充困難



# 如果把架構換成 VM 呢？

雖然較省錢，但每個專案都需維護 VM 檔案，以 VMware 為例，包括 虛擬磁碟 (.vmdk)、VM 的記憶體 (.vmem)、虛擬 BIOS (.nvram)、設定檔 (.vmx)、補充設定檔 (.vmxf)、紀錄檔 (.log)、快照 (.vmsd)、快照狀態資訊 (.vmsn)、紀錄 VM 暫停狀態 (.vmss) ... 等檔案

```
D:\VM\Ubuntu $ ls -al
total 30558838
drwxr-xr-x  1 Titan 197609      0 一月 10 01:42  .
drwxr-xr-x  1 Titan 197609      0 八月 10 11:21  ..
-rw-r--r--  1 Titan 197609    8684 一月 10 01:42  'Ubuntu.nvram'
-rw-r--r--  1 Titan 197609 27544518656 一月 10 01:42  'Ubuntu.vmdk'
-rw-r--r--  1 Titan 197609      0 八月  6 21:37  'Ubuntu.vmsd'
-rw-r--r--  1 Titan 197609    2994 一月 10 01:42  'Ubuntu.vmx'
-rw-r--r--  1 Titan 197609     383 八月  6 23:17  'Ubuntu.vmxf'
-rw-r--r--  1 Titan 197609 3741319168 一月 10 00:59  'Ubuntu-f82919f4.vmem'
-rw-r--r--  1 Titan 197609   6130557 一月 10 01:42  'Ubuntu-f82919f4.vms'
-rw-r--r--  1 Titan 197609   270160 一月 10 01:42  vmware.log
```

# VM 的缺點

- 吃系統記憶體資源
- 切換多重專案浪費時間
- 浪費主機硬碟空間

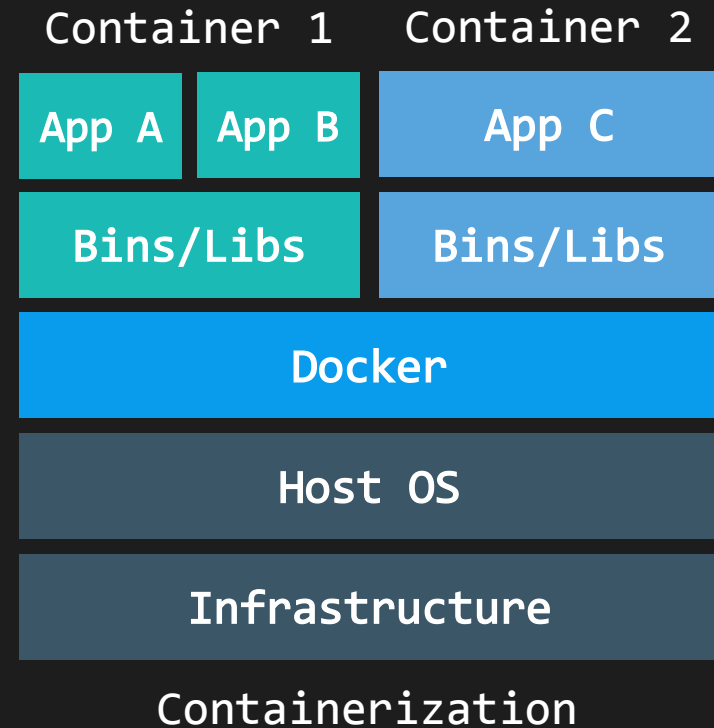
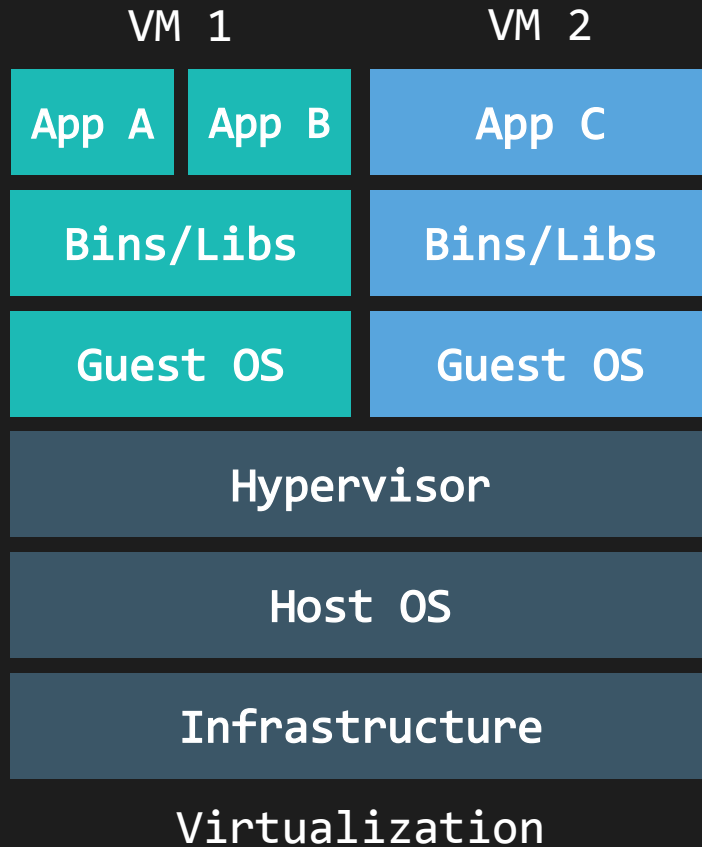
# 改用 Docker 後

- 不需要額外的機器及人力維護成本 (提早下班)
- 各專案可透過 Dockerfile 將環境進行版本控制
- 任何環境都可執行 (不管是實體機、虛擬機、雲端，不管系統是 Windows、Linux 都可以)

# 所以為什麼要使用 Docker ?

- 容易部署
- 快速建置環境，可自動建立開發環境
- 開發與正式環境一致
- 環境 (資源) 隔離
- 資源有效利用 (直接使用系統資源)
- 降低維運成本 (讓機器做，可以提早下班)
- 比虛擬化技術更輕量 (容器化技術，後面會介紹)

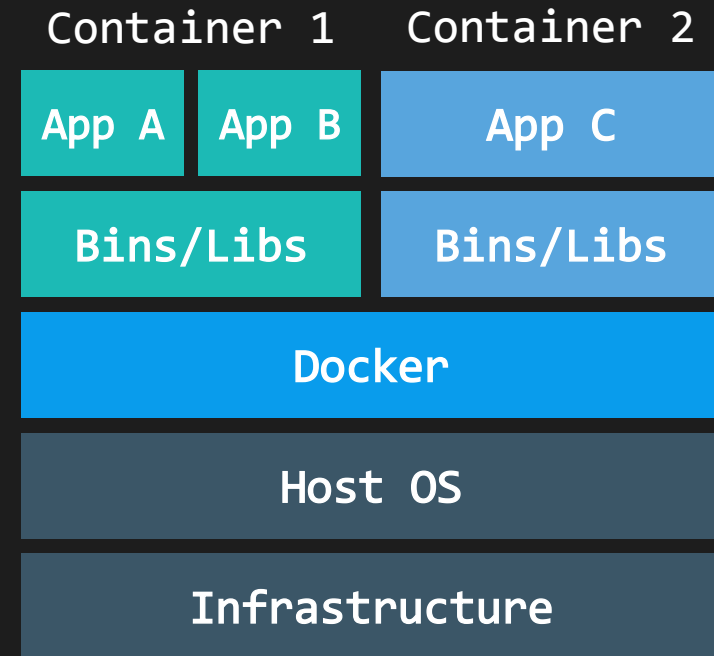
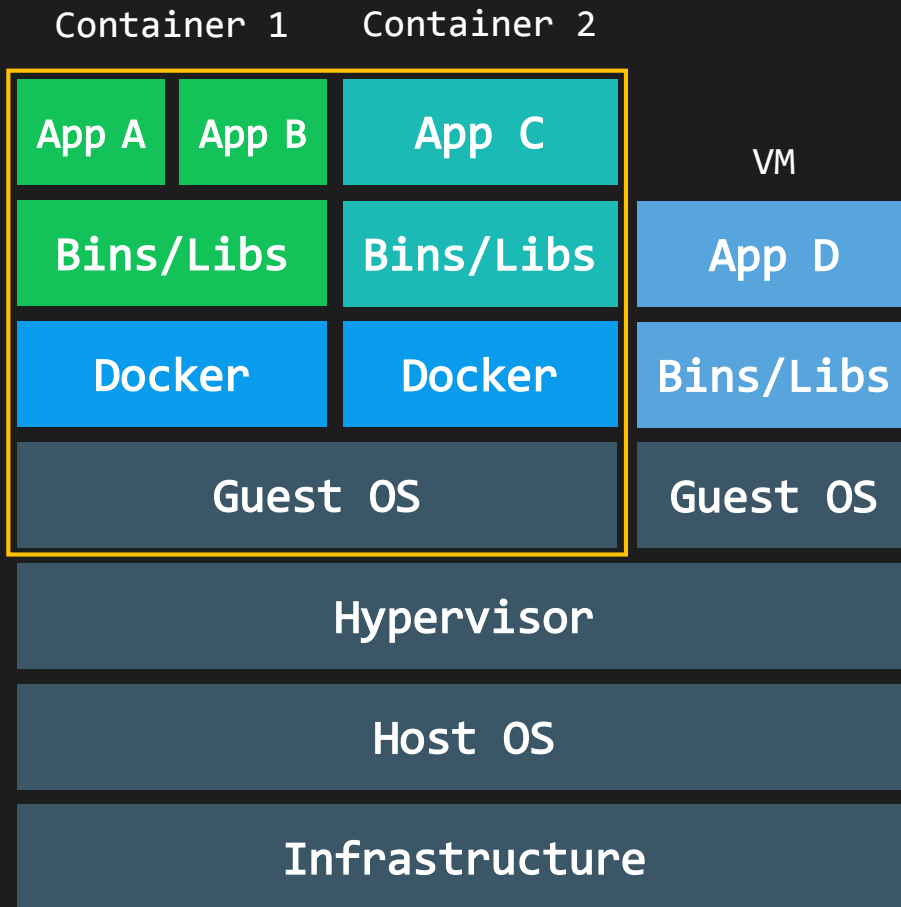
# Virtualization vs Containerization



# Virtualization vs Containerization

	Virtualization	Containerization
啟動	慢 (分鐘)	快 (秒開)
容量	大 (GB)	小 (MB)
效能	慢	快
host 可支撐數量	數個 ~ 數十個	數個 ~ 數百個
複製相同環境	超慢	快

# Docker with VM

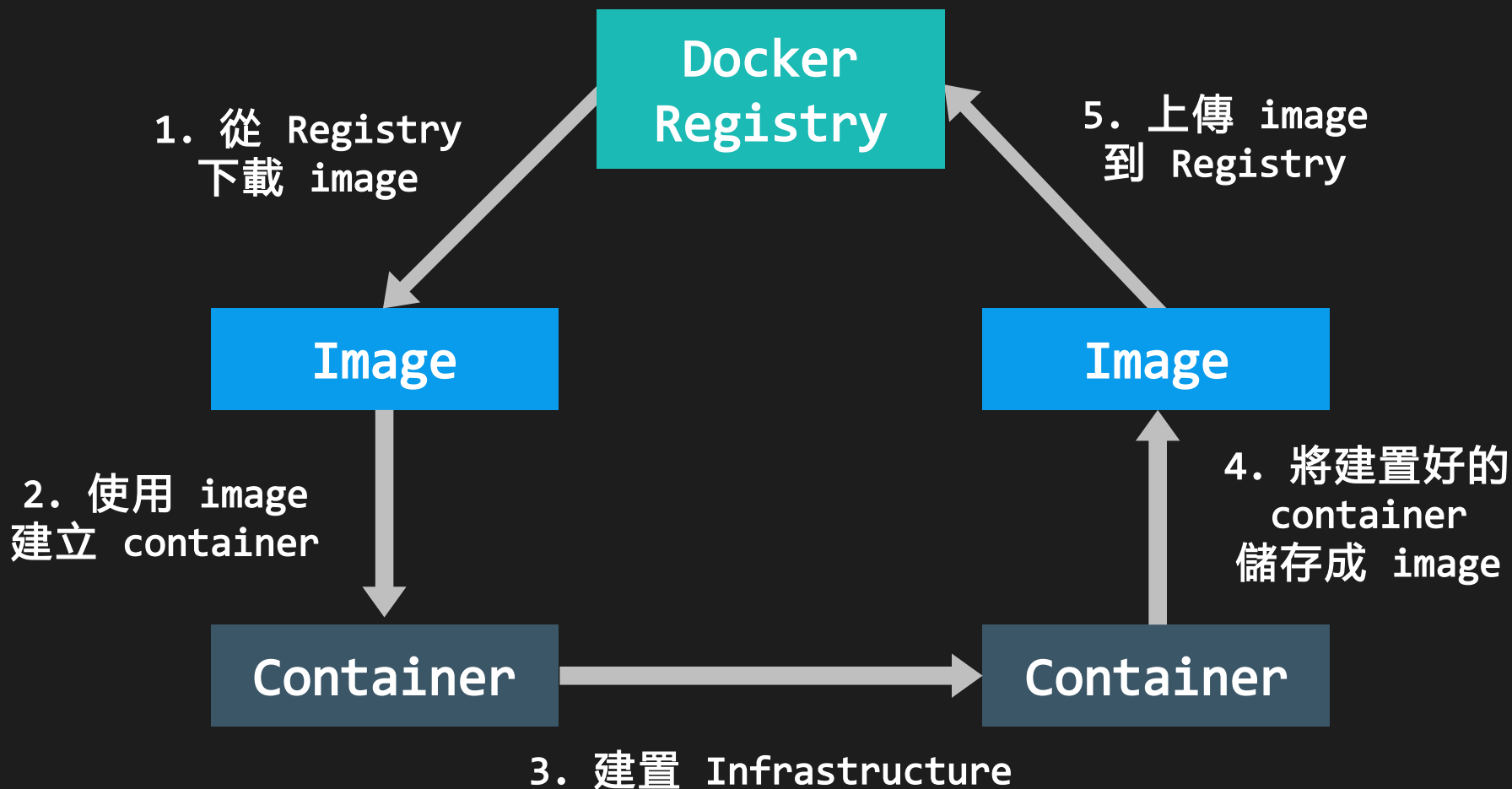


# Docker 三個基本概念

- Image (映像檔)
  - 唯讀模式 (R\O)
  - 用來建立 Container 的模板
  - 可從 Repository 下載到別人做好的 image 來直接使用
- Container (容器)
  - 讀寫模式 (R\W)
  - 也是貨櫃的意思
  - 可將軟體執行所需的所有資源打包到一個隔離的容器中
  - 可解決環境不一致的問題 (e.g. 雲端、OS、套件版本...等環境)
  - 一個 image 可以創造出多個不同的 container
  - Docker 利用 Container 來執行應用
- Repository (倉庫)
  - 集中存放 image 的地方
  - Repository vs Registry (倉庫註冊伺服器)
  - Registry 分為公開倉庫 (Public) 和私有倉庫 (Private) 兩種形式
    - 最大的公開倉庫是 Docker Hub , 存放了大量的 image 供使用者下載
    - 使用者也可以在本地網路內建立一個私有倉庫



# Docker 是怎麼運作的？



# 安裝 Docker

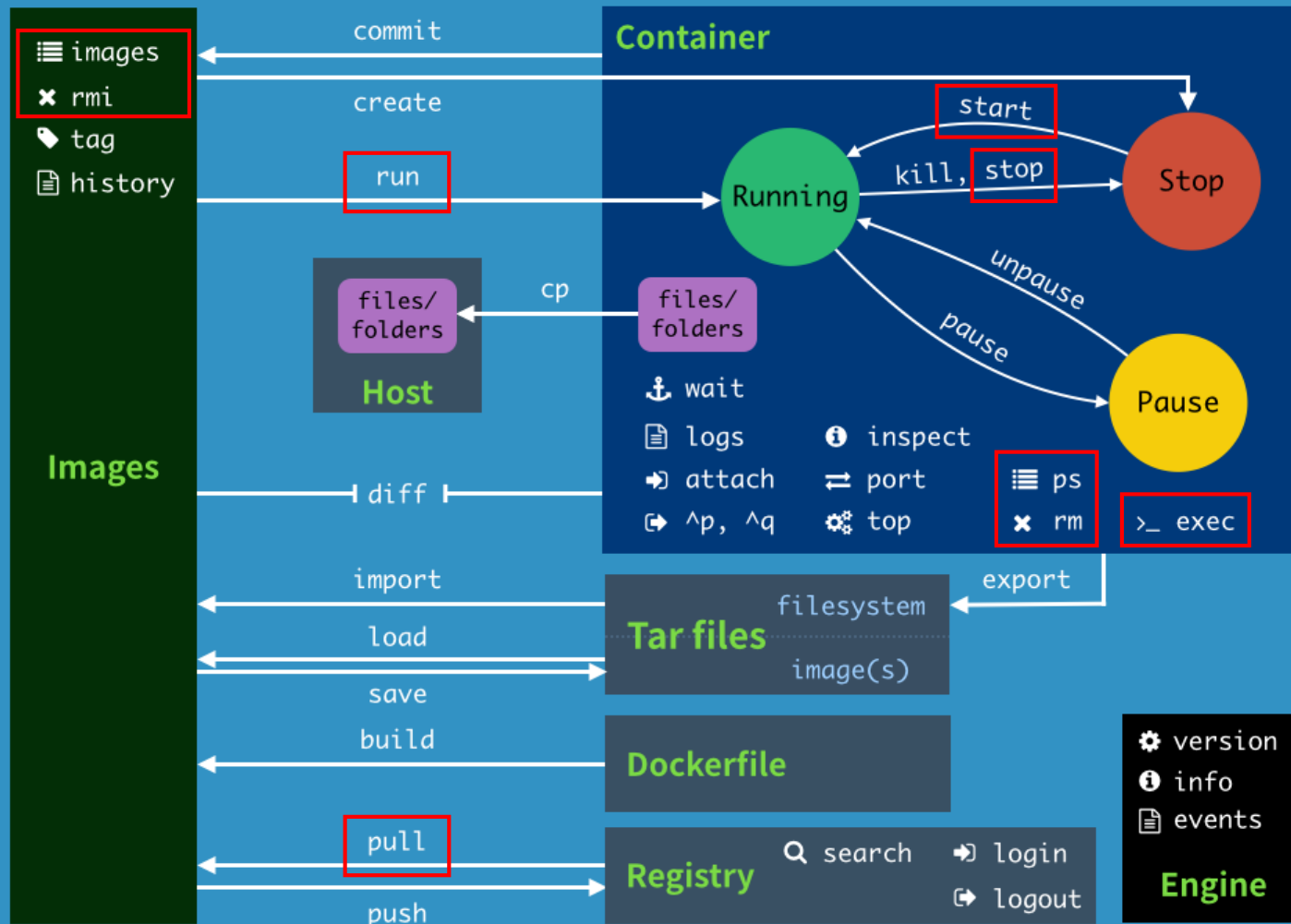
- 安裝：<https://docs.docker.com/engine/installation/>
- 檢查：

```
$ docker -v
```

```
Docker version 17.12.0-ce, build c97c6d6
```

# Docker 指令

## Docker Commands Diagram



@fntsrlike

# Docker 常用指令

# 從遠端取得 php image

```
$ docker pull php:7.1-apache
```

# 查看 local 已下載哪些 image

```
$ docker images
```

# 查看正在執行的 container 狀態

```
$ docker ps
```

# 查看所有 container 狀態

```
$ docker ps -a
```

# 刪除 container , 如果容器還在運作中 , 需要將他先 stop

```
$ docker rm <container>
```

# 刪除 image

```
$ docker rmi <image>
```

# Docker 常用指令

# 新建並啟動容器

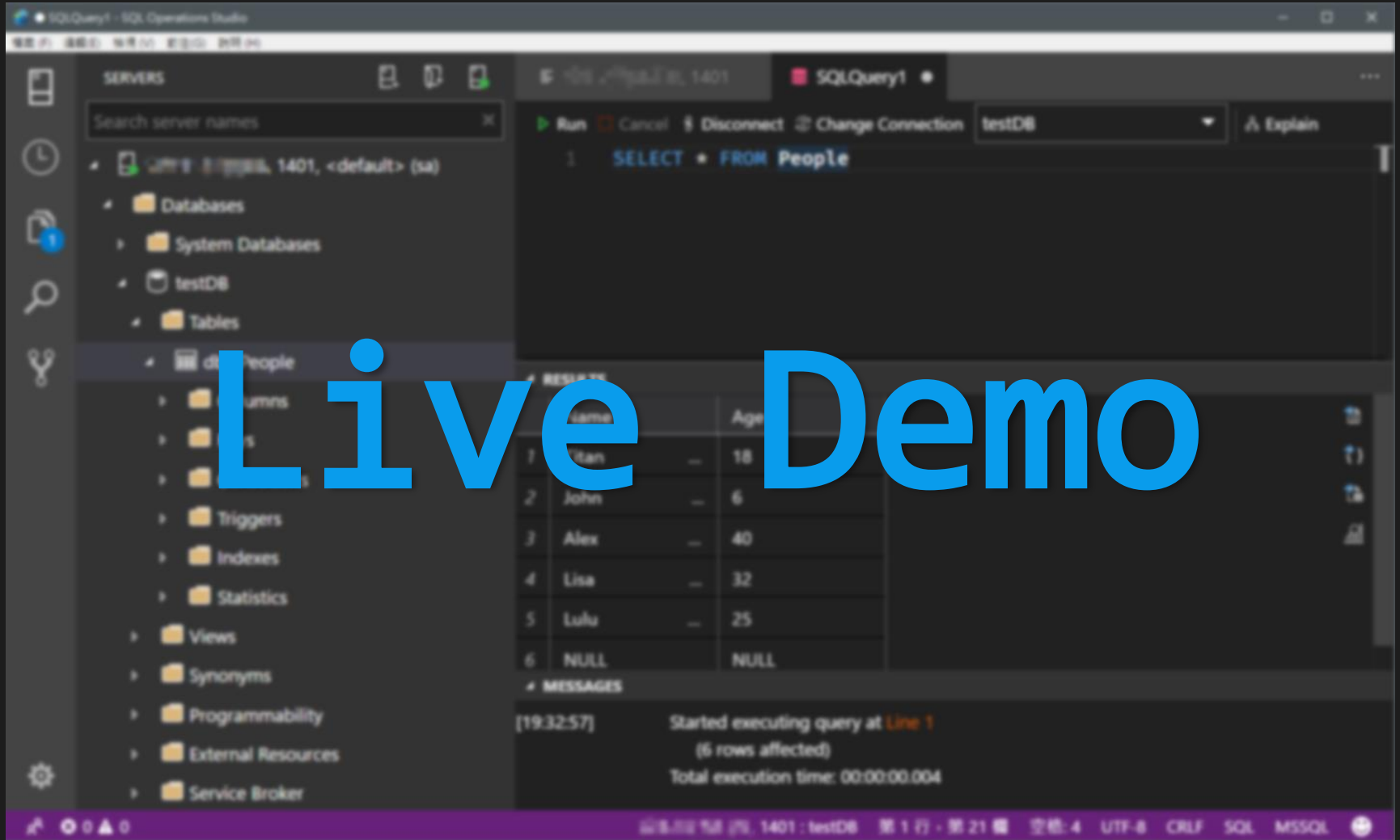
- `--name php-apache` : 容器名稱為 php-apache
- `-d` : 在背景執行
- `-p 8080:80` : 指定一個 port , host 對外開 8080 port , 容器內開 80 port
- `-v `pwd` /php:/var/www/html/` : 將 host 目前目錄掛載在容器的指定目錄上

# 將主機上的目前目錄掛載到容器的指定目錄 , 並且是背景執行

```
$ docker run --name phpApache -d -p 8080:80 -v  
`pwd`/php:/var/www/html/ php:7.1-apache
```

# Docker 實戰演練

# 在 Docker 下建立並使用 MSSQL Server for Linux



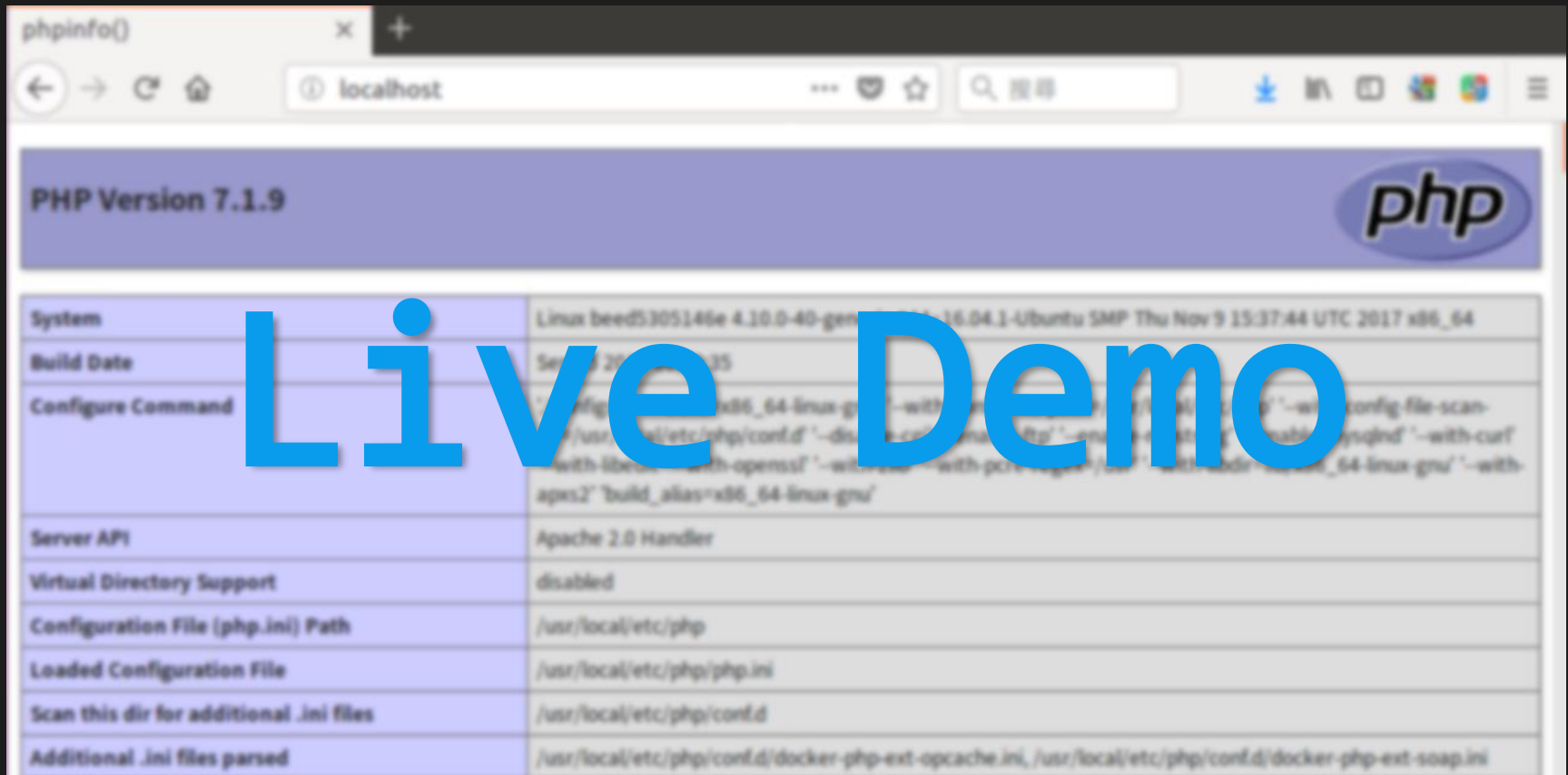
The screenshot displays the SQL Operations Studio interface. On the left, the 'SERVERS' pane shows a tree view with 'Databases' expanded to 'testDB' and 'Tables' expanded to 'People'. The central query editor contains the SQL statement: `SELECT * FROM People`. The 'RESULTS' pane shows the following data:

	Name	Age
1	Tan	18
2	John	6
3	Alex	40
4	Lisa	32
5	Lulu	25
6	NULL	NULL

The 'MESSAGES' pane shows the execution log: `[19:32:57] Started executing query at Line 1 (6 rows affected) Total execution time: 00:00:00.004`. The status bar at the bottom indicates the connection is to '1401: testDB' and the query is on 'Line 1'.

# Live Demo

# 利用 Dockerfile 自製 PHP Apache Server 的 Docker image



phpinfo() localhost

PHP Version 7.1.9

System	Linux beed5305146e 4.10.0-40-generic #36-Ubuntu SMP Thu Nov 9 15:37:44 UTC 2017 x86_64
Build Date	Sep 26 2016 13:55:55
Configure Command	./configure '--prefix=/usr' '--enable-mbstring' '--enable-sockets' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--disable-zend-debug' '--enable-gd' '--with-curl' '--with-openssl' '--with-pcre-glib' '--with-xml' '--with-xmlrpc' '--with-zlib' '--with-apxs2' 'build_alias=x86_64-linux-gnu'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	/usr/local/etc/php/php.ini
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	/usr/local/etc/php/conf.d/docker-php-ext-opcache.ini, /usr/local/etc/php/conf.d/docker-php-ext-soap.ini

**Live Demo**



# 利用 Docker Compose 建立 LAMP

Docker Compose 是一個工具，用來定義與執行多個 container 組成的 Docker Applications



# 資源

## • 官方

- [Home](#)
- [Facebook](#)
- [Twitter](#)
- [Docker Hub](#)

## • 論壇

- [DockOne.io](#)
- [docker 中文](#)

## • 社群

- [Docker Taipei - Meetup](#)
- [Docker Taipei - Facebook group](#)
- [線上 docker 讀書會 - Facebook group](#)
- [Docker & DevOps - Facebook group](#)

## • 電子書

- [Gitbook - 《Docker — 從入門到實踐》  
正體中文版](#)
- [Gitbook - 最完整的Docker聖經 - Docker  
原理圖解及全環境安裝](#)

# 參考文獻

- [Gitbook - 《Docker — 從入門到實踐》正體中文版](#)
- [Gitbook - 最完整的Docker聖經 - Docker原理圖解及全環境安裝](#)
- [SlideShare - 前端工程師一定要知道的 Docker 虛擬化容器技巧](#)
- [iT 邦幫忙 - 用30天來介紹和使用 Docker系列 - Day 1 介紹為何要使用 Docker](#)
- [Microsoft - Run the SQL Server 2017 container image with Docker](#)
- [Microsoft - Configure SQL Server 2017 container images on Docker](#)
- [Microsoft - Backup and restore SQL Server databases on Linux](#)
- [KKBruce - 一次就愛上 MSSQL Server for Linux](#)
- [Christian Lee - SQL Operation Studio for DevOps](#)