

Docker

前言

想像你現在是接外包案子的工程師，常常因為顧客需求不同，系統版本不同，常常會在系統上安裝虛擬主機(Virtual Machine)，有了虛擬化的技術，軟體相容性變得不再是問題。

然而，電腦裡有著各式各樣的虛擬作業系統，若要同時執行多個虛擬主機顯然會降低系統效能，因此 Docker 就出現了，其目標是實作輕量級的作業系統虛擬化解決方案。

Docker是甚麼：Build, Ship, and Run Any App, Anywhere.

- Docker 是一個開源專案，提供開發、轉移和執行應用程式。Docker讓應用程式可以獨立於infrastructure中，提高交付的速度。
- 透過作業系統(Linux為其原生)，建立多個虛擬的實例(instances)，有獨立的函式庫及獨立的shell，並彼此隔離。
- Docker將應用程式獨立於Container當中，省去搭建作業系統的成本，使得程式運行更快速。
- 假設在不同的平台上都安裝了Docker的應用程式，就可以將自己開發的程式用Container打包起來，快速移植到另一個平台。

Docker 三要件

1. Image：

是一個可以獨立執行的輕量級套件，其包含所有執行程式所需要的函式庫、環境變數與設定檔等，R / O。

可以把它想成是 VM 的 Guest OS (安裝在虛擬機上的作業系統) 。

2. Container

Image 可以創造出多個不同的 Container，可以被啟動、開始、停止、刪除，並且是互相分離。Container 在啟動的時候會建立一層在最外 (上) 層並且是讀寫模式 (R\W) 。

3. Repository

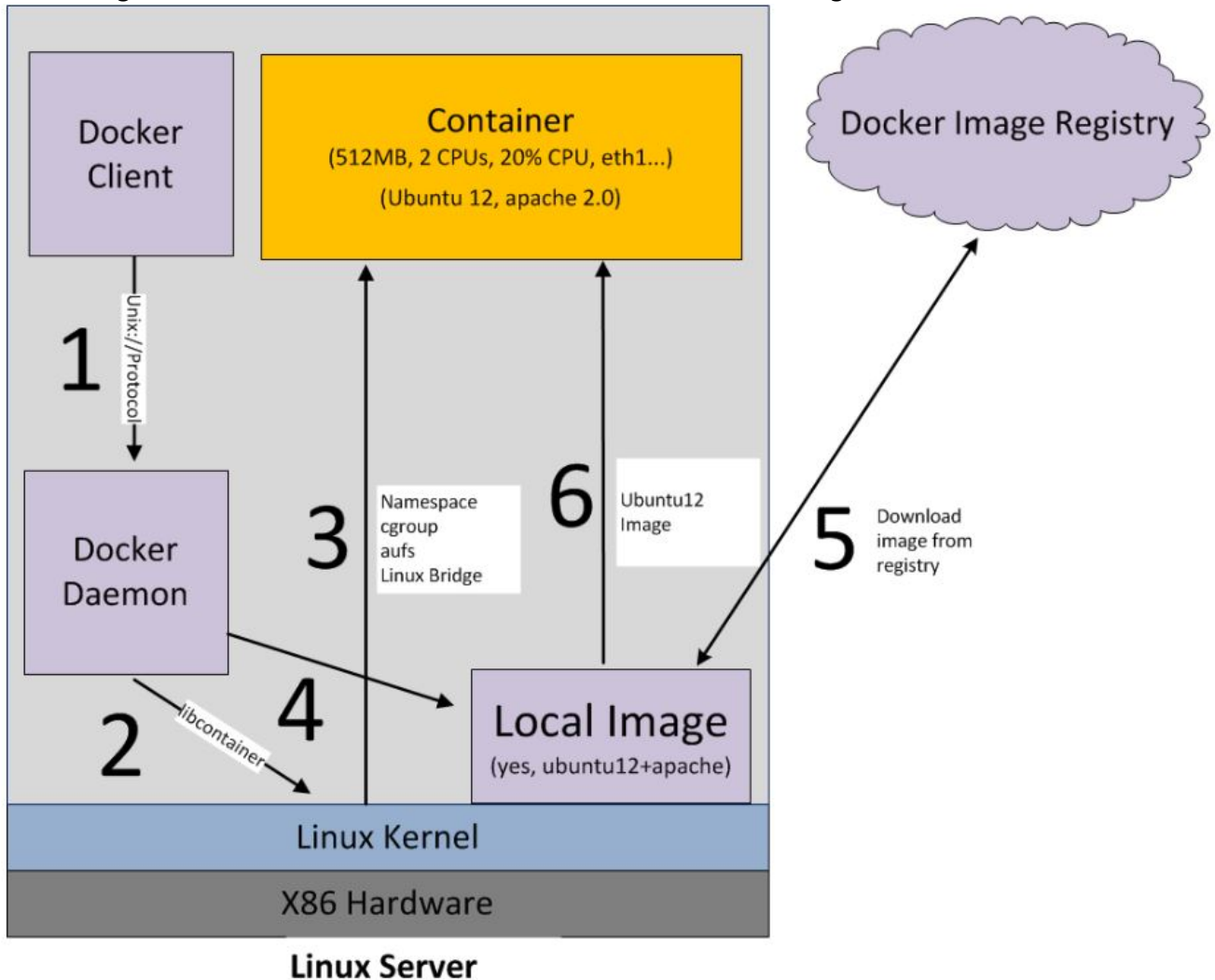
是集中存放 Image 檔案的場所 (Docker Hub)，每個檔有不同的標籤 (tag)。分為公開倉庫 (Public) 和私有倉庫 (Private) 兩種形式。

Docker Architecture

Docker使用client-server architecture。組成包含Docker client、daemon與registries。

- Docker client主要是提供指令操作，透過REST API的方式與Docker daemon溝通。
- Docker daemon負責管理Docker物件，包含images、containers、網路狀況等。

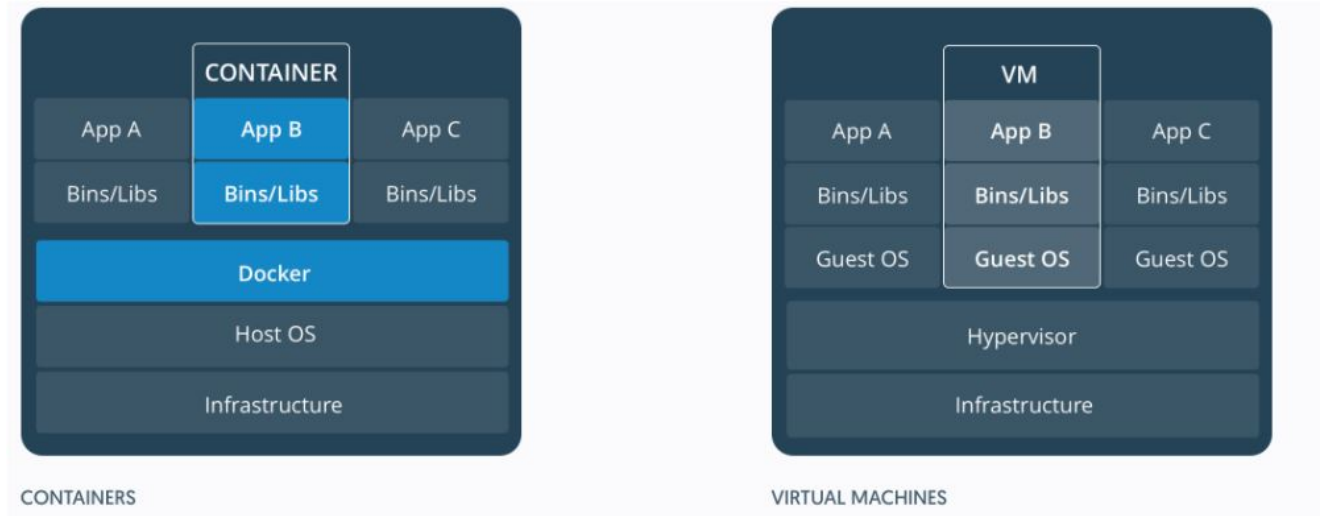
- Docker registries如同上面提到，就是遠端伺服器，必要時會將image載入於本機中。



VM & Docker 差異

1. VM：實作在硬體層之上，為了建立可以執行整套作業系統的sandbox
 - 需要安裝OS
 - VM 開機需要花一點時間
 - 完全的把系統的硬體資源隔離
 - 佔用硬碟的容量較大
2. Docker：實作在作業系統層之上，將程式所需的程式碼、函式庫、環境配置檔都打包
 - 直接從 Docker Hub Pull 作業系統的 Image
 - 不用開機，啟動速度快
 - 底層還是使用作業系統的 Kernel

- 佔用硬碟的容量較小



簡單範例

1. 先確定環境是否可執行

`docker version`

```
root@ubuntu: /
root@ubuntu: /#
root@ubuntu: /# docker version
Client version: 1.5.0
Client API version: 1.17
Go version (client): go1.4.1
Git commit (client): a8a31ef
OS/Arch (client): linux/amd64
Server version: 1.5.0
Server API version: 1.17
Go version (server): go1.4.1
Git commit (server): a8a31ef
root@ubuntu: /# lsb_release -r
Release: 14.10
root@ubuntu: /# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0c:29:8b:e4:58
          inet addr:192.168.1.103  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe8b:e458/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:10422 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3629 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2519934 (2.5 MB)  TX bytes:409113 (409.1 KB)

root@ubuntu: /#
```

2. 輸入指令，呼叫daemon

`docker run -d --name web -m 512m -p 8080:80 joshhu/webdemo`

```

root@ubuntu: /
root@ubuntu: /#
root@ubuntu: /#
root@ubuntu: /# docker run -d --name web -m 512m -p 8080:80 joshhu/webdemo

```

docker client command → `docker run`
 name of the container → `--name web`
 setting system resource(512M memory) → `-m 512m`
 the image to fill up container → `joshhu/webdemo`

unix://var/run/unix.sock

```

root@ubuntu: /
root@ubuntu: /#
root@ubuntu: /# docker images

```

REPOSITORY	TAG	IMAGE ID	CREATED	VIRTUAL SIZE
joshhu/webdemo	latest	a3574f323972	42 hours ago	243.2 MB

```

root@ubuntu: /# docker run -d --name web -m 512m -p 8080:80 joshhu/webdemo
6cfc8e566e59f1f2e1eacb8f76c4183b63d06ac5216e4588394578b1aee8196c
root@ubuntu: /#

```

show local images → `docker images`
 image already exists → `joshhu/webdemo`
 container id → `6cfc8e566e59f1f2e1eacb8f76c4183b63d06ac5216e4588394578b1aee8196c`
 image name → `joshhu/webdemo`

3. 如果這個joshhu/webdemo影像檔沒有在本機，就會先去下載，下載回來後，再填入空的Container web中。

```

root@ubuntu: /
root@ubuntu: /# docker images

```

REPOSITORY	TAG	IMAGE ID	CREATED	VIRTUAL SIZE

```

root@ubuntu: /# docker run -d --name web -m 512m -p 8080:80 joshhu/webdemo
Unable to find image 'joshhu/webdemo:latest' locally
Pulling repository joshhu/webdemo
a3574f323972: Pulling image (latest) from joshhu/webdemo, endpoint: https://reg
a3574f323972: Download complete
511136ea3c5a: Download complete
f3c84ac3a053: Download complete
a1a958a24818: Download complete
9fec74352904: Download complete
d0955f21bf24: Download complete
1214be61bcaa: Download complete
45ad00454734: Download complete
8be3dadcb43a: Download complete
2d4173730925: Download complete
e8f612e3238f: Download complete
d11f03cd837d: Download complete
8abfc77aac61: Download complete
d1dfd80f5ed9: Download complete
Status: Downloaded newer image for joshhu/webdemo:latest
9738c5c00c07ef0534267e657e609460bf360147bf73970799c374e28bcb7441
root@ubuntu: /#

```

no local image download from registry → `Unable to find image 'joshhu/webdemo:latest' locally`

4. 注意 - Container的ID及名稱

在Docker執行時，如果你沒有使用`-name <名稱>`的參數，Docker會主動幫這個Container取一個名稱。而不管有沒有命名，一定會產生一個全世界獨一無二的Container id。

```
root@ubuntu: /
root@ubuntu: /#
root@ubuntu: /#
root@ubuntu: /# docker run -d joshhu/webdemo
69a0c2851abdd49aaa019c6f2d8074d08c665a75e25251eed3bd4d73e66be331
root@ubuntu: /# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
69a0c2851abd        joshhu/webdemo:latest  "/start.sh"        3 seconds ago       Up 2 seconds
80/tcp              stoic_yalow
```

No name assigned

Container id

System generated name