

Docker 入門介紹



Daniel

<https://goo.gl/2oXKqD>

Agenda

Docker 簡介

安裝與設定Docker

Docker 基礎操作

Docker 執行容器

Docker 建立映像檔

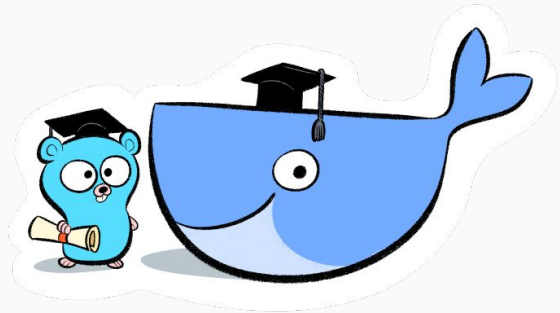
Docker 映像檔管理

Docker 簡介

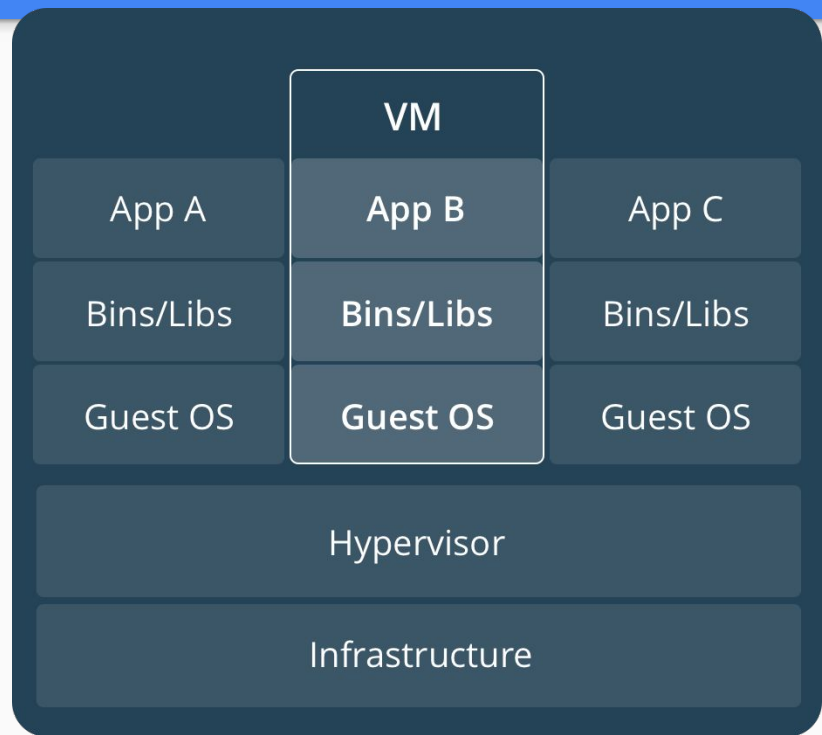
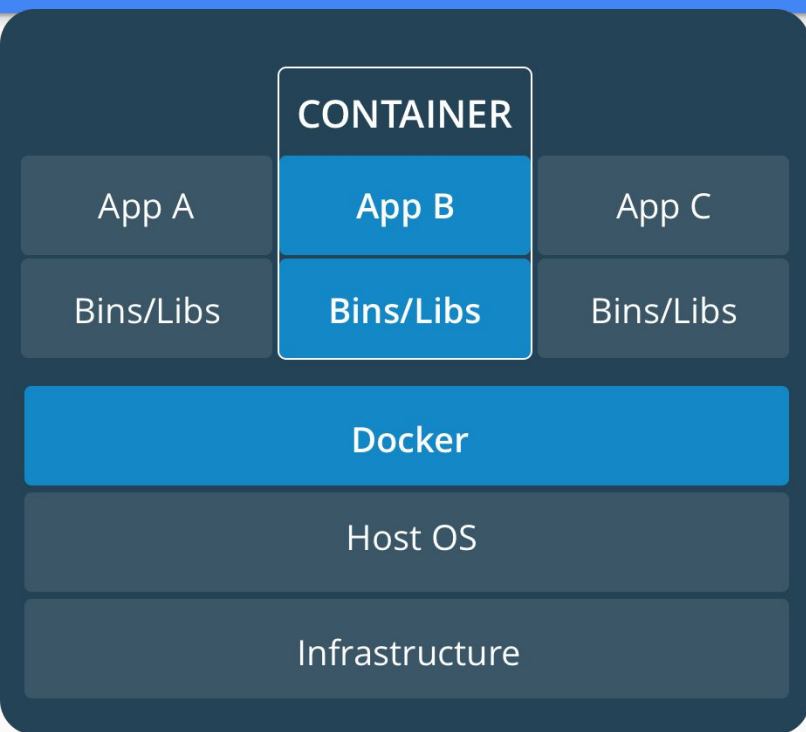
什麼是Docker？

Docker專案誕生於2013年，基於Go語言實作，目標是實作輕量級的作業系統虛擬化解決方案。Docker的基礎是Linux容器(LXC)等技術。By --- [Docker---入門到實踐](#)

Docker是一個開放原始碼軟體專案，讓應用程式佈署在軟體容器下的工作可以自動化進行，藉此在Linux作業系統上，提供一個額外的軟體抽象層，以及[作業系統層虛擬化](#)的自動管理機制。By --- [維基百科](#)



Docker vs VM



Docker vs VM

特性	容器	虛擬機器
啟動速度	秒級	分鐘級
硬碟使用	一般為 MB	一般為 GB
性能	接近原生	較差
系統支援量	單機支持上千個容器	一般幾十個
隔離性	安全隔離	完全隔離

Docker 特性

輕量

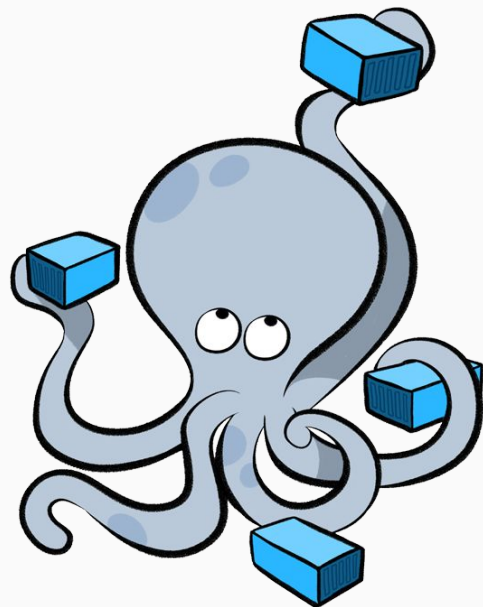
- Container(容器)不需要消耗太多記憶體, 只需要程序所需的用量。

快速

- 相較於VM, 啟動Container就像啟動一般程序一樣快, 只需要幾秒鐘的時間。

易於重現環境

- 透過Dockerfile所建構出來的 Docker image, 可以確保每次 Container的環境的完整性。



Docker Basic



映像檔(Image)

- 用來建立容器的唯讀範本。

容器(Container)

- 用映像檔所創造的執行實例 (Instance), 使用docker run來執行的程序或是服務就稱為 Container。

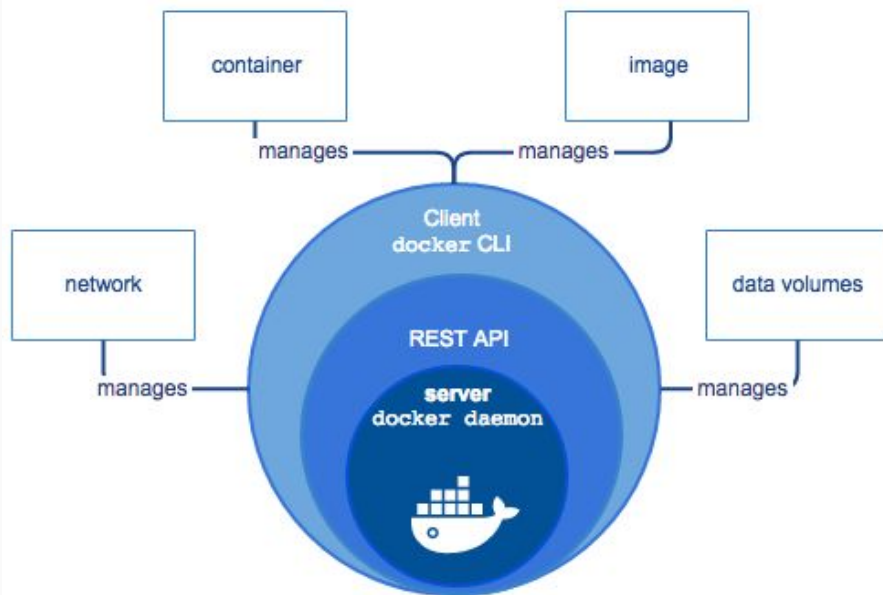
倉庫(Registry)

倉庫是docker image的集合場所。

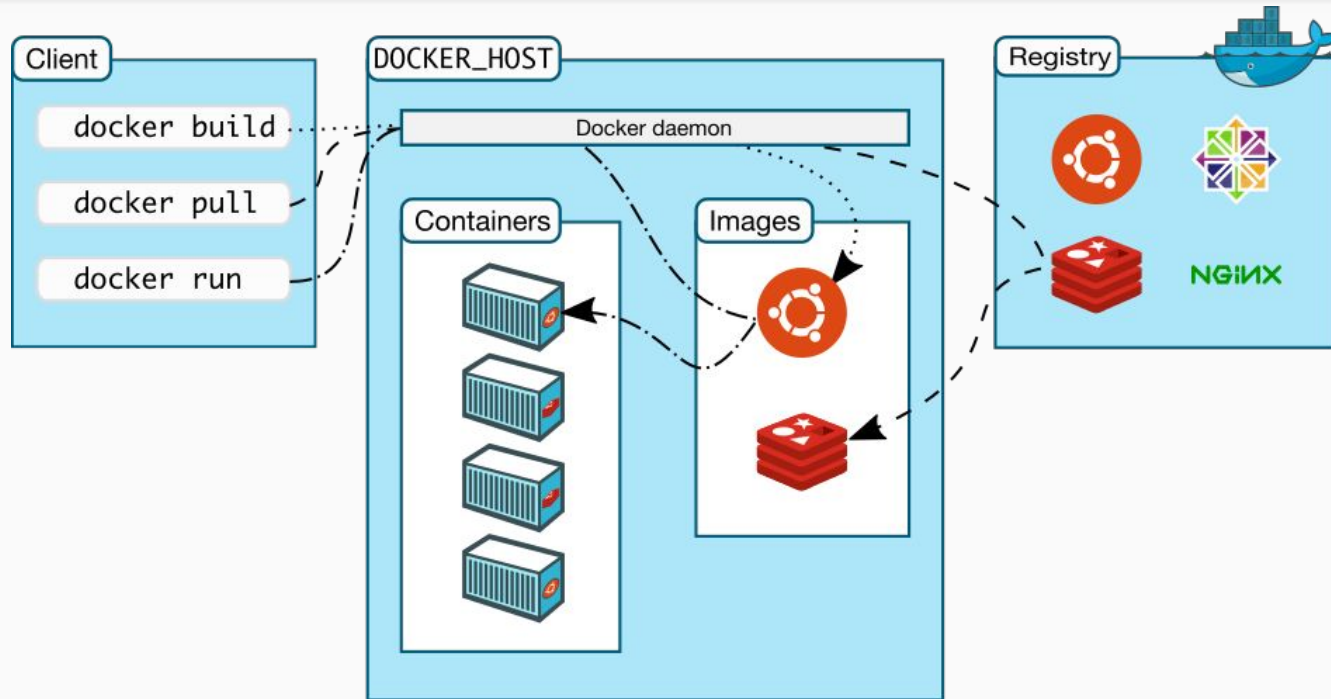
- 公開倉庫 --- [DockerHub](https://hub.docker.com/)
- 私有倉庫 --- [Docker Registry](https://docs.docker.com/registry/)

Docker Engine

- Docker是採用Client-Server架構。
- Server會起一個daemon叫**dockerd**。
- REST API接口會跟daemon溝通，並下達指示。
- Docker CLI的指令叫**docker**。



Docker Architecture



openSUSE Leap 42.2 Yast 簡介

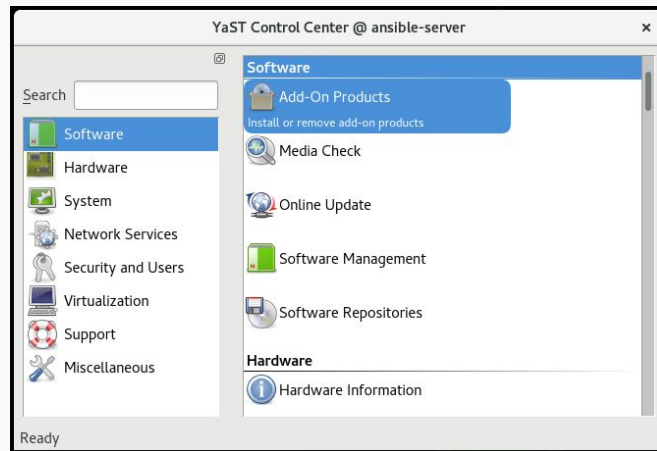
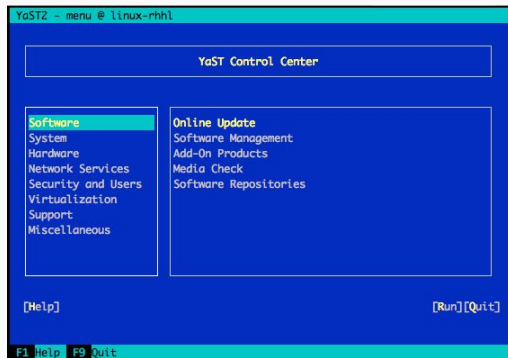
YaST

- SUSE提供Yast(Yet another Setup Tool), 實現方便的系統管理, 幾乎可以配置系統各各方面, 如: 硬體管理、軟體管理、網路設定、系統服務、安全設置等諸多功能。這些功能的引入, 大大簡化配置系統的過程。



指令

- `# yast` --- 文字介面
- `#yast2` --- 圖形介面
- `# yast --list` --- 列出yast模組



Activities → Show Applications → YaST

Lab

安裝openSUSE Docker Repo

更新軟體

```
# zypper update -y
```

匯入openSUSE官方的Docker Repo

```
# zypper --gpg-auto-import-keys addrepo -f
```

```
http://download.opensuse.org/repositories/Virtualization:/containers/openSUSE\_Leap\_42.2/Virtualization:containers.repo
```

更新套件庫

```
# zypper refresh
```



安裝Docker

搜尋套件庫中docker版本

```
# zypper search -s docker
```

安裝docker

```
# zypper -n install docker
```

確認docker資訊

```
# zypper info docker
```

開機啟動docker服務

```
# systemctl enable docker.service
```

啟動docker

```
# systemctl start docker
```

觀察docker版本

```
# docker --version
```

觀察跟上一個指令差異

```
# docker version
```



Docker 基礎操作

觀察 docker image, 顯示本機目前下載的images

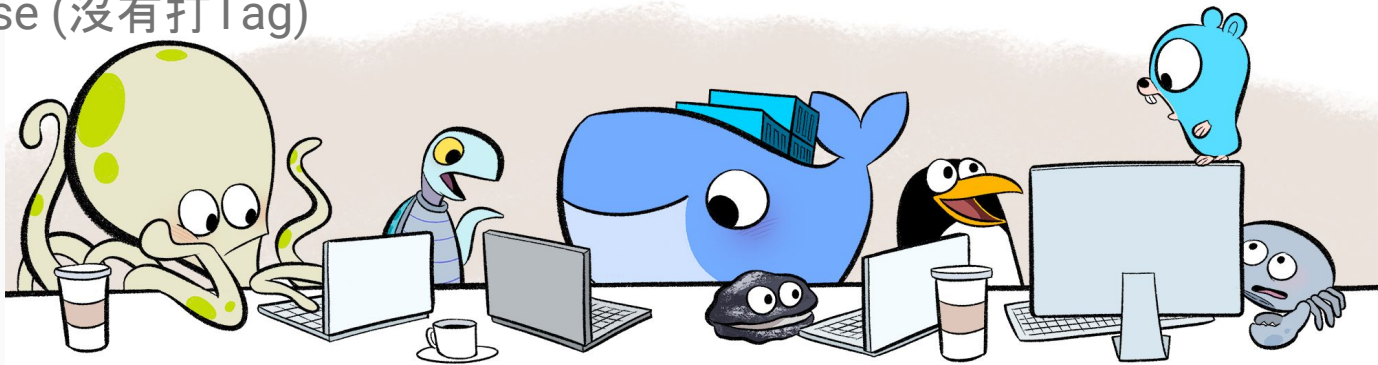
docker images

搜尋[Docker Hub](https://hub.docker.com/)上公開的image, 最多只會顯示**25個**

docker search opensuse

如果沒有指定Tag, **預設是latest**

docker pull opensuse (沒有打Tag)



Docker 執行容器

列出所有Container

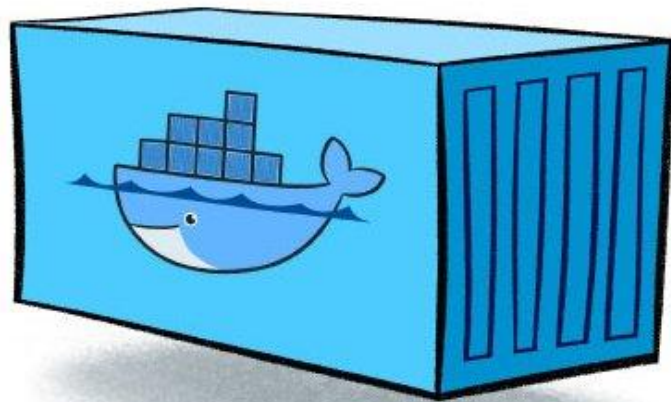
```
# docker ps -a
```

- -a 顯示所有Containers

執行Container

```
# docker run --rm -it opensuse /bin/bash
```

- --rm 退出Container就自動刪除
- -i 保持互動模式 STDIN/STDOUT
- -t 模擬終端機



Docker 建立映像檔

修改現在映像檔

```
# docker run -it opensuse /bin/bash
```

```
|----- bash-4.3# zypper -n install openssh
```

```
# docker commit -m "Add openssh" -a "daniel" d2398694b7ef testrepo/openssh:v1
```

- -m commit訊息
- -a 作者
- d2398694b7ef 原始Container的ID
- testrepo/openssh:v1 Repository名稱/映像檔名稱:Tag

```
# docker commit -m "Add openssh" -a "daniel" $(docker ps -ql)
```

```
testrepo/openssh:v2
```

- -q 安靜模式
- -l 顯示最新的Container, 包含所有狀態

Dockerfile 建立映像檔

基礎映像檔來源

FROM <image> or <image>:<tag>

維護者

MAINTAINER <name>

加入檔案(Dockerfile所在目錄的相對路徑、目錄或檔案, 也支援網址及自動解壓縮)

ADD <src> <dest>

複製檔案或目錄(相對路徑)

COPY <src> <dest>

執行的指令

RUN <command>

容器啟動時執行的指令, 只能有一條CMD命令, 如果指定多個CMD, 只有最後一條會生效。

CMD ["param1", "param2"]

可接受連線埠 (需搭配 -p 參數使用)

EXPOSE <port_number>

容器啟動時執行的程序

ENTRYPOINT ["executable", "param1", "param2"]

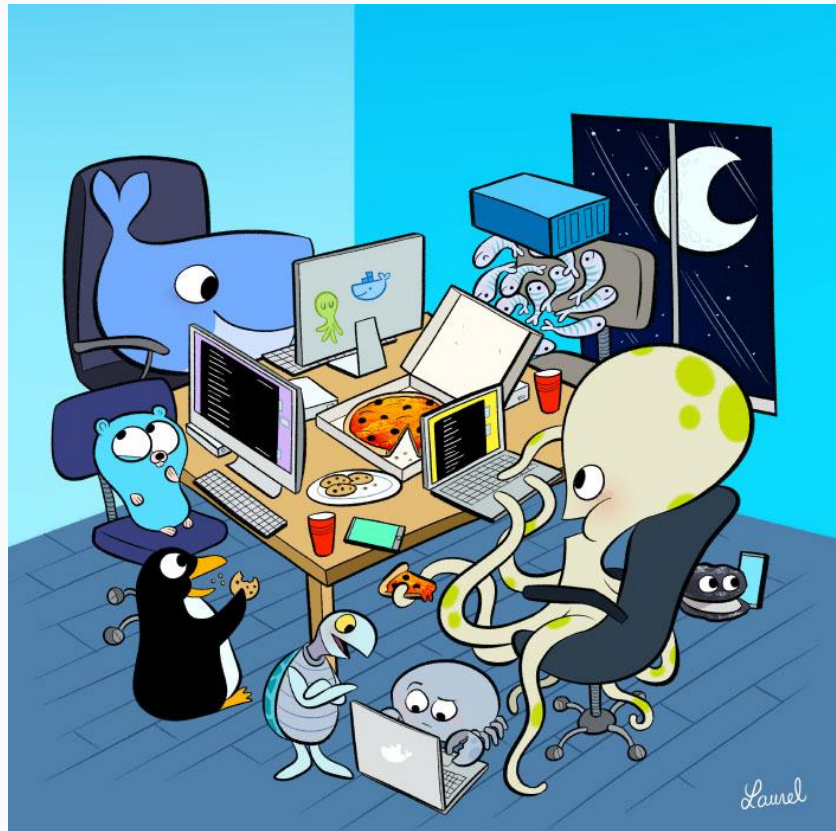
Dockerfile 建立映像檔

建立映像檔指令

```
# docker build -f Dockerfile -t testrepo/demo:v1 .
```

-f 指定Dockerfile

-t Repository名稱/Container名稱:Tag



Docker 映像檔管理

映像檔管理

docker tag SOURCE_IMAGE[:TAG] TARGET_IMAGE[:TAG]

- -f 指定Dockerfile
- -t Repository名稱/Container名稱:Tag

docker rmi -f <image_id> 刪除image

- -f 強制執行

docker rm -f <container_id>

- -f 強制執行



Events

openSUSE X StudyArea

StudyArea

Facebook: <https://www.facebook.com/sataiwan/>

openSUSE Taiwan

Facebook: <https://www.facebook.com/groups/opensuse.zh/>

KKTIX: <https://opensuse-tw.kktix.cc/>

去年9月開始舉辦一連串的workshop

- Ansible
- Ceph
- OBS / openQA
- Docker
- openStack

Reference

[Docker Documentation](#)

[O'relly Docker Cookbook](#)

[Docker for beginners](#)

[Docker入門與實戰](#)

Q & A