



# Docker WorkShop

投程科 DevOps 組

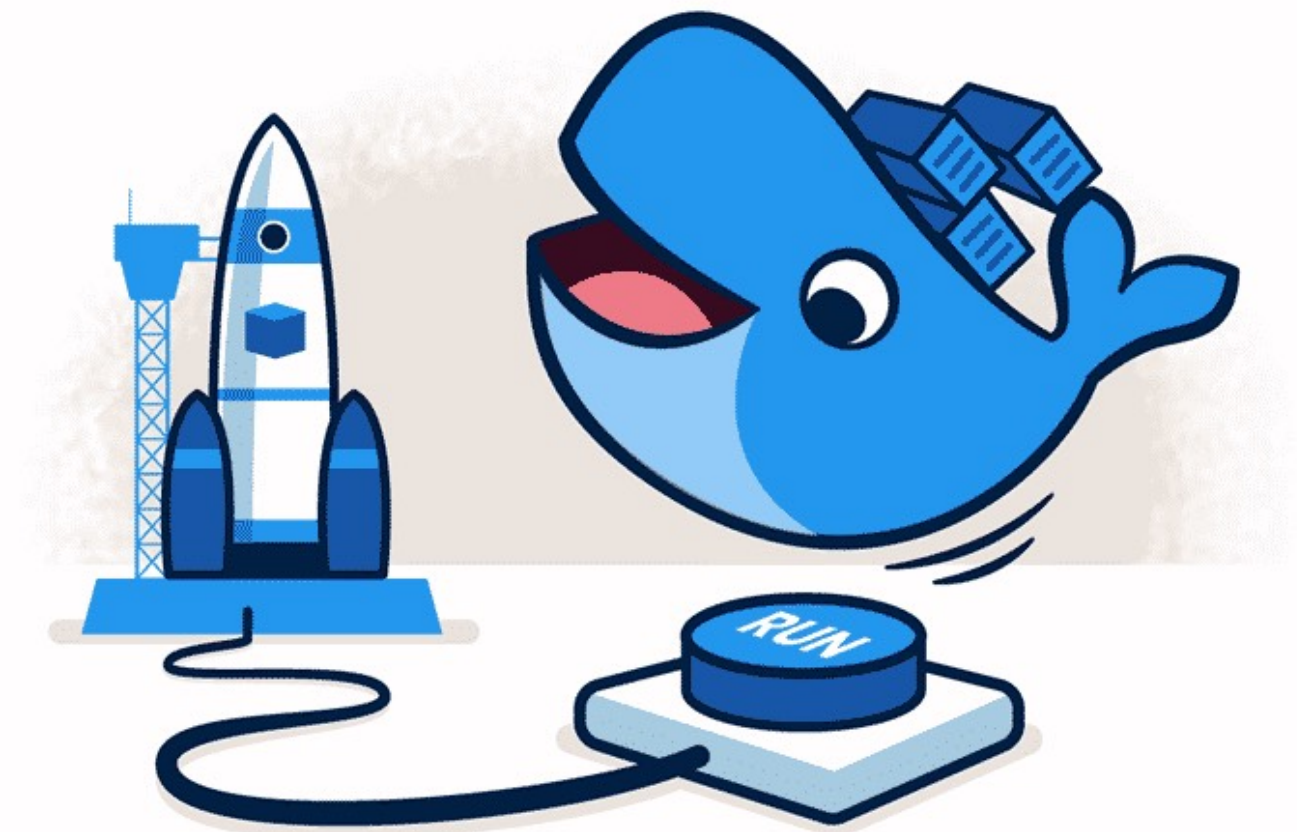
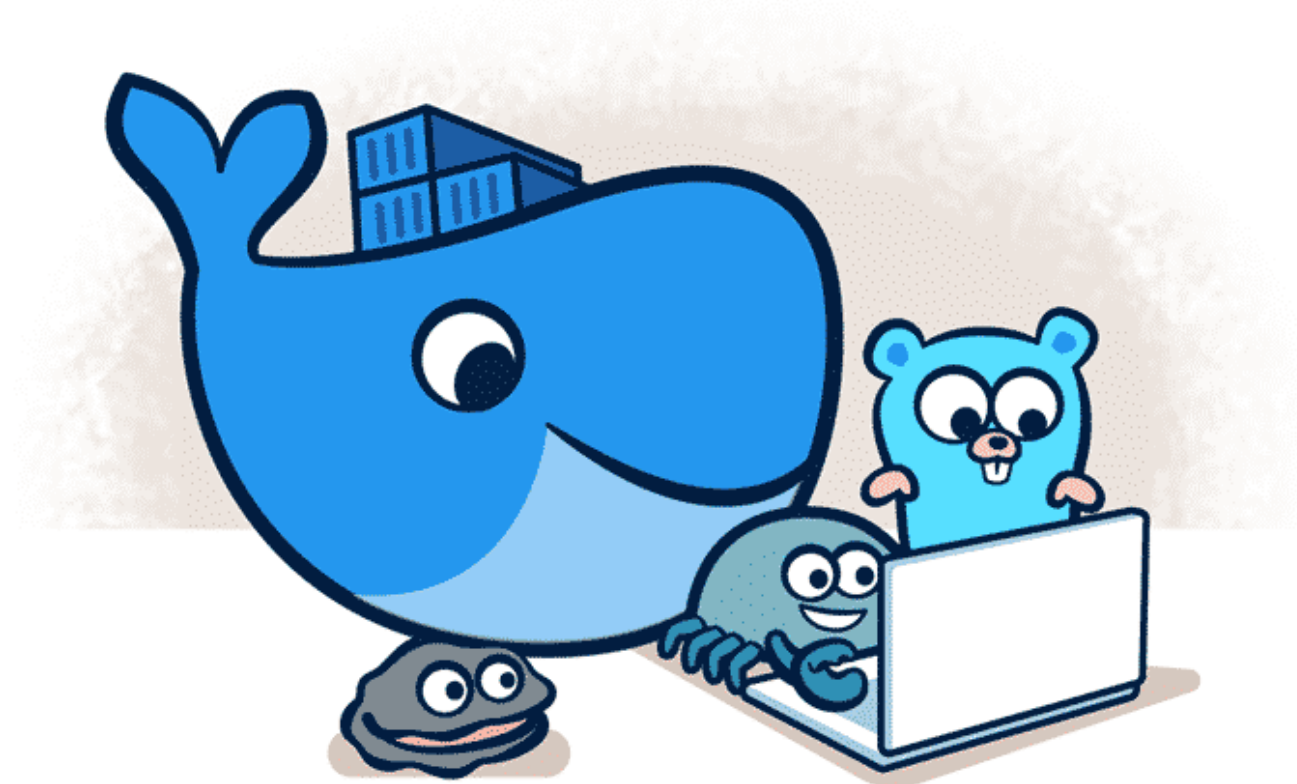
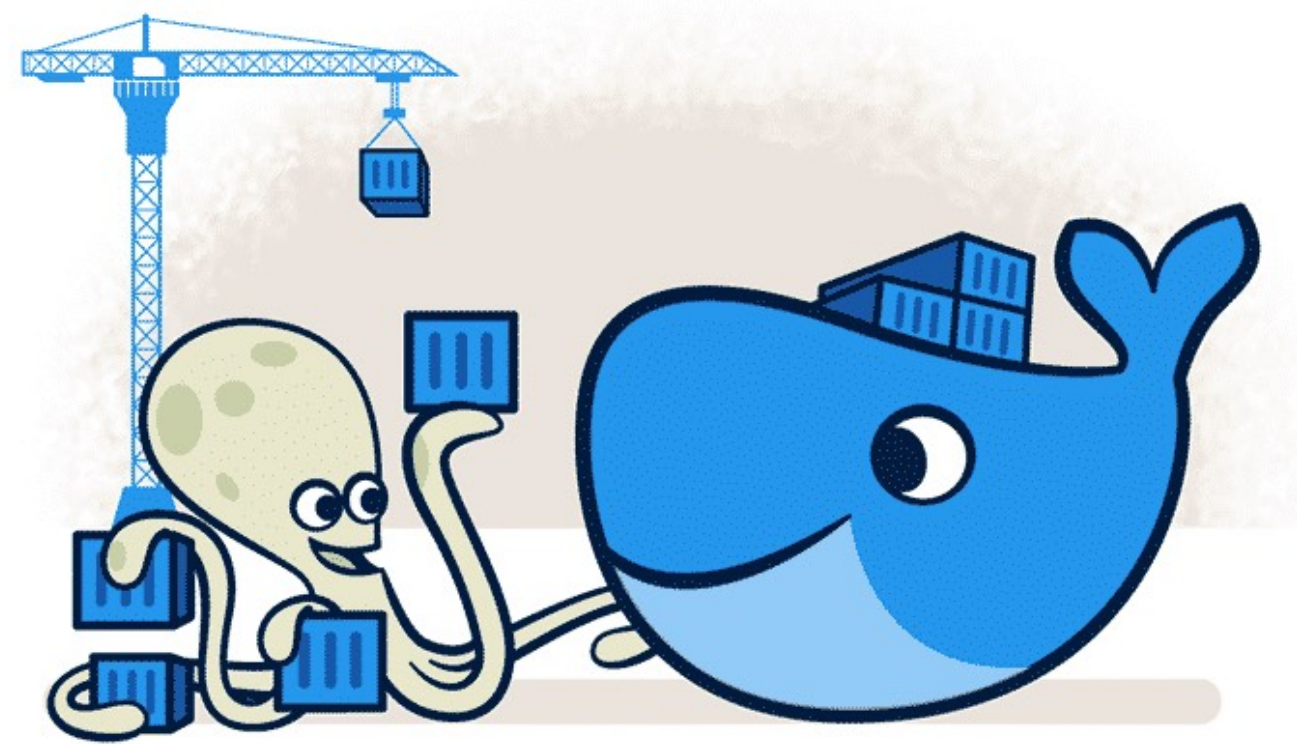
Heui Yuan Kao

# Agenda

- Learning Goal
- Docker Intro
  - VM vs. Containers
- Clarify Docker Words
  - Dockerfile
  - Docker-compose
  - DockerHub
- Hands on Docker
  - Docker info
  - Docker ps
  - Docker pull
  - Docker create(冷)
  - Docker start
  - Docker run
  - Docker history
- Q & A

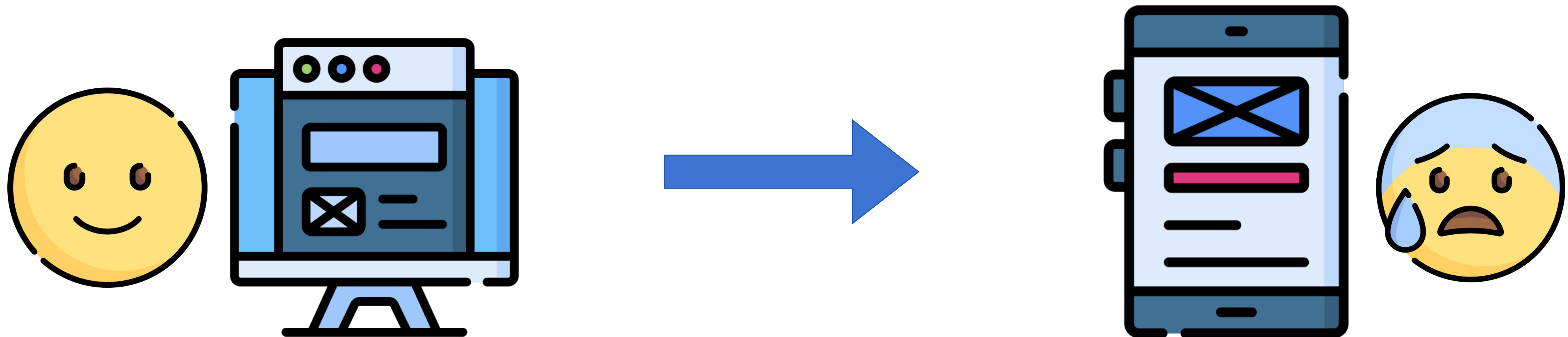
# Learning Goal

- 了解、認識、會使用，能跟別人解釋什麼是Docker
- 能分辨 Image、Container、Repository



# Docker Intro

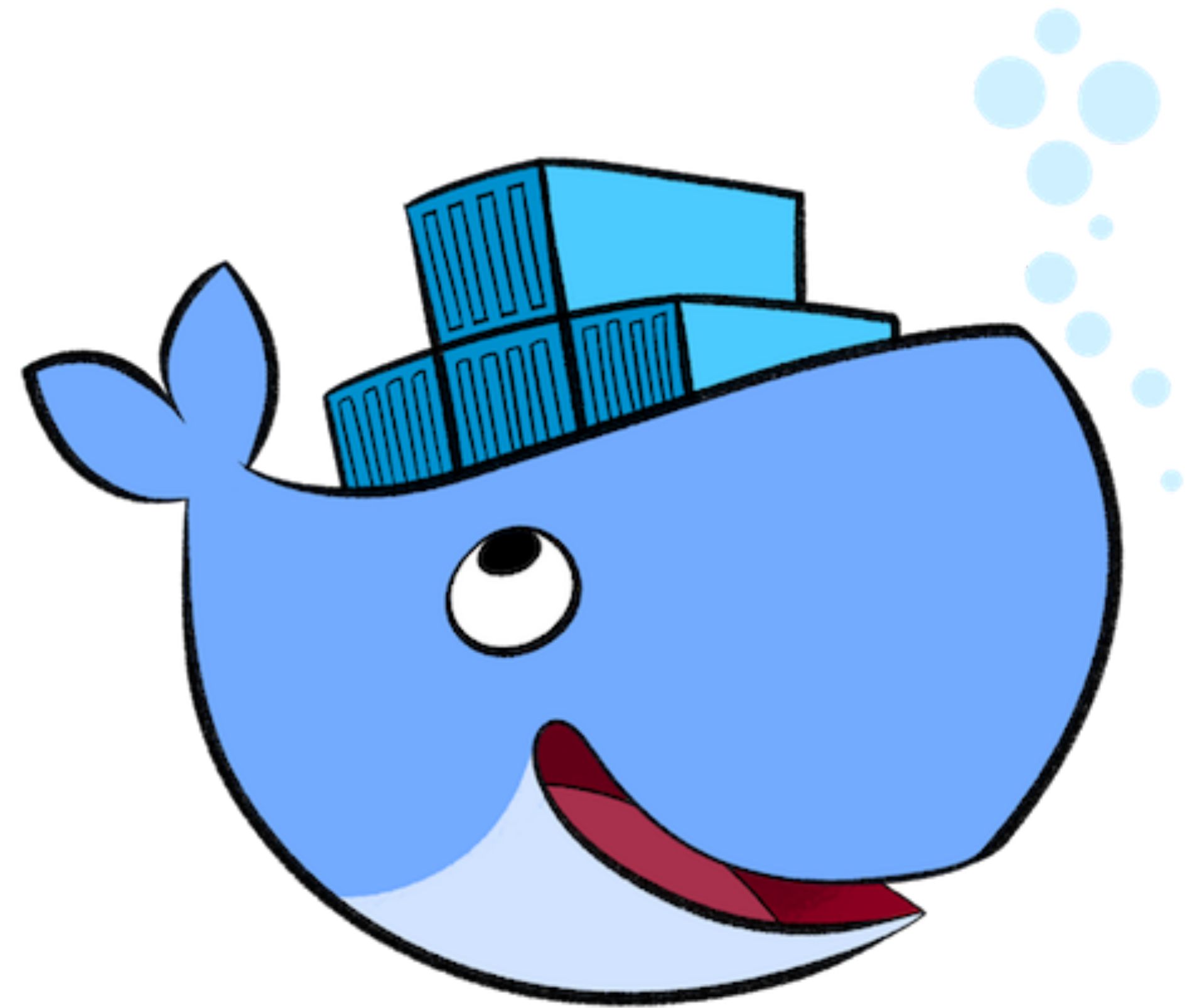
# Why Docker





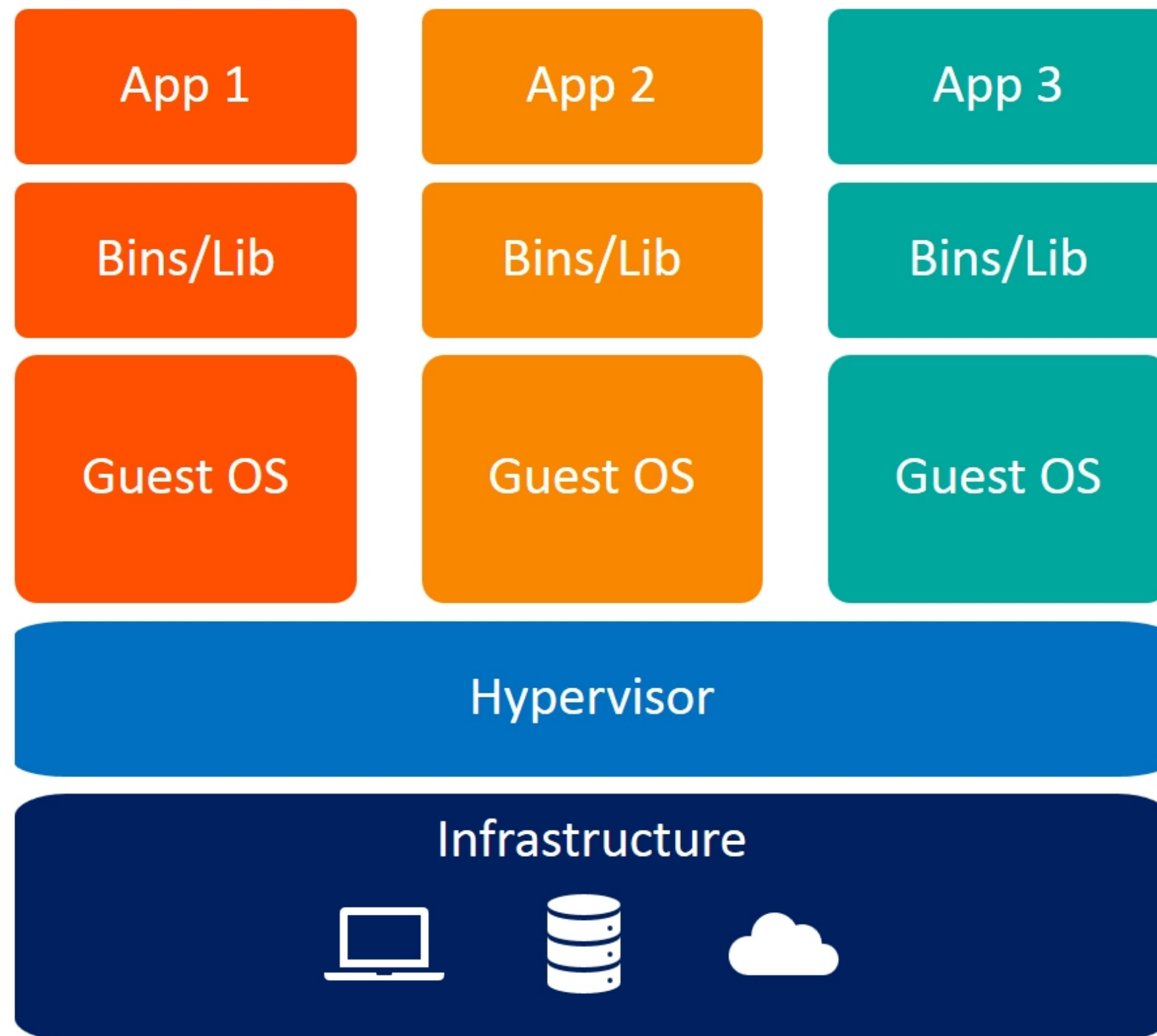
# Docker Intro

- 2013 dotCloud(PaaS) Go開源專案
- 輕量級虛擬化的作業系統
- 基於Linux Container (LXC) 技術
- ~~實現撇除環境因素的黑魔法~~

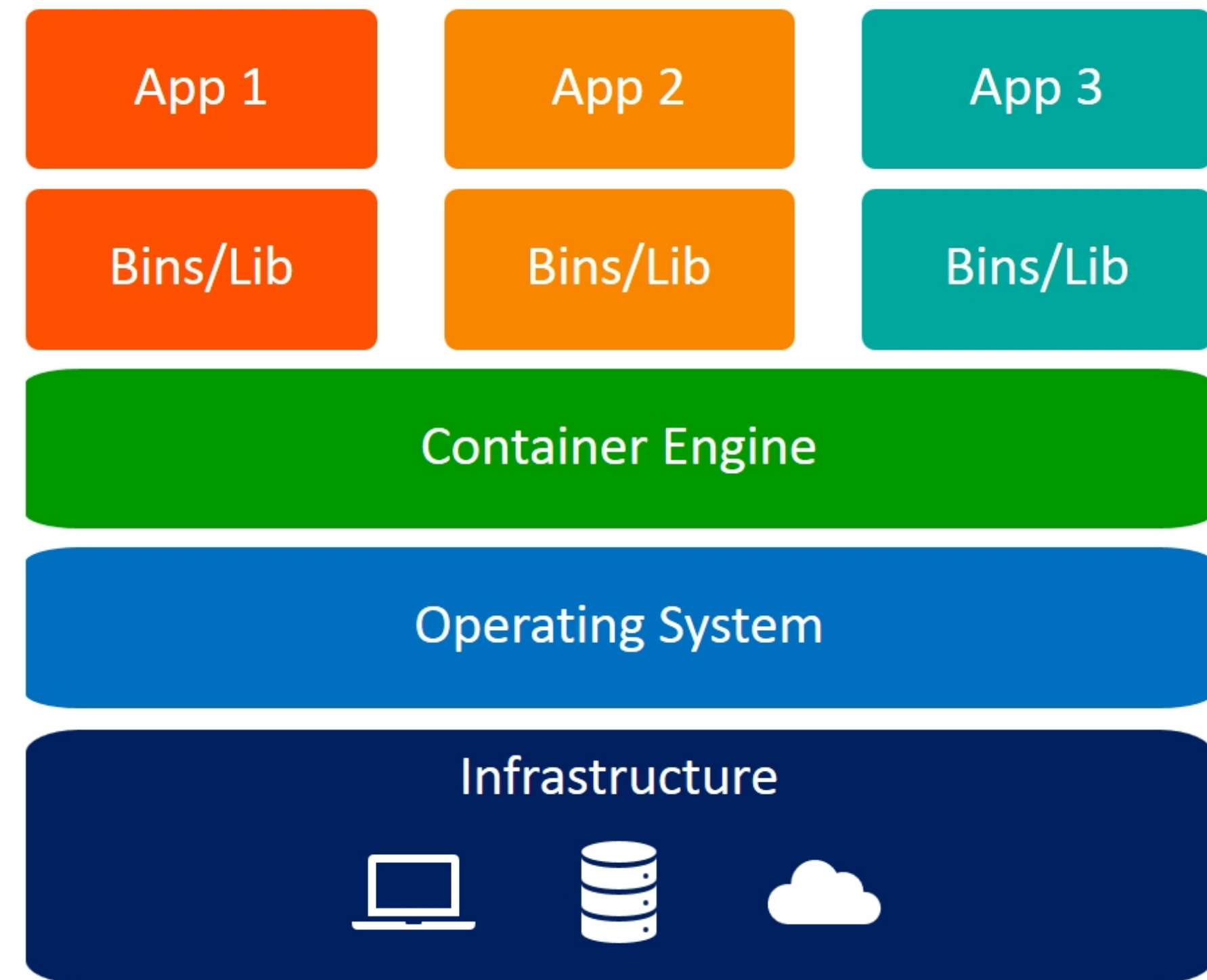


# VM vs. Container

文言文



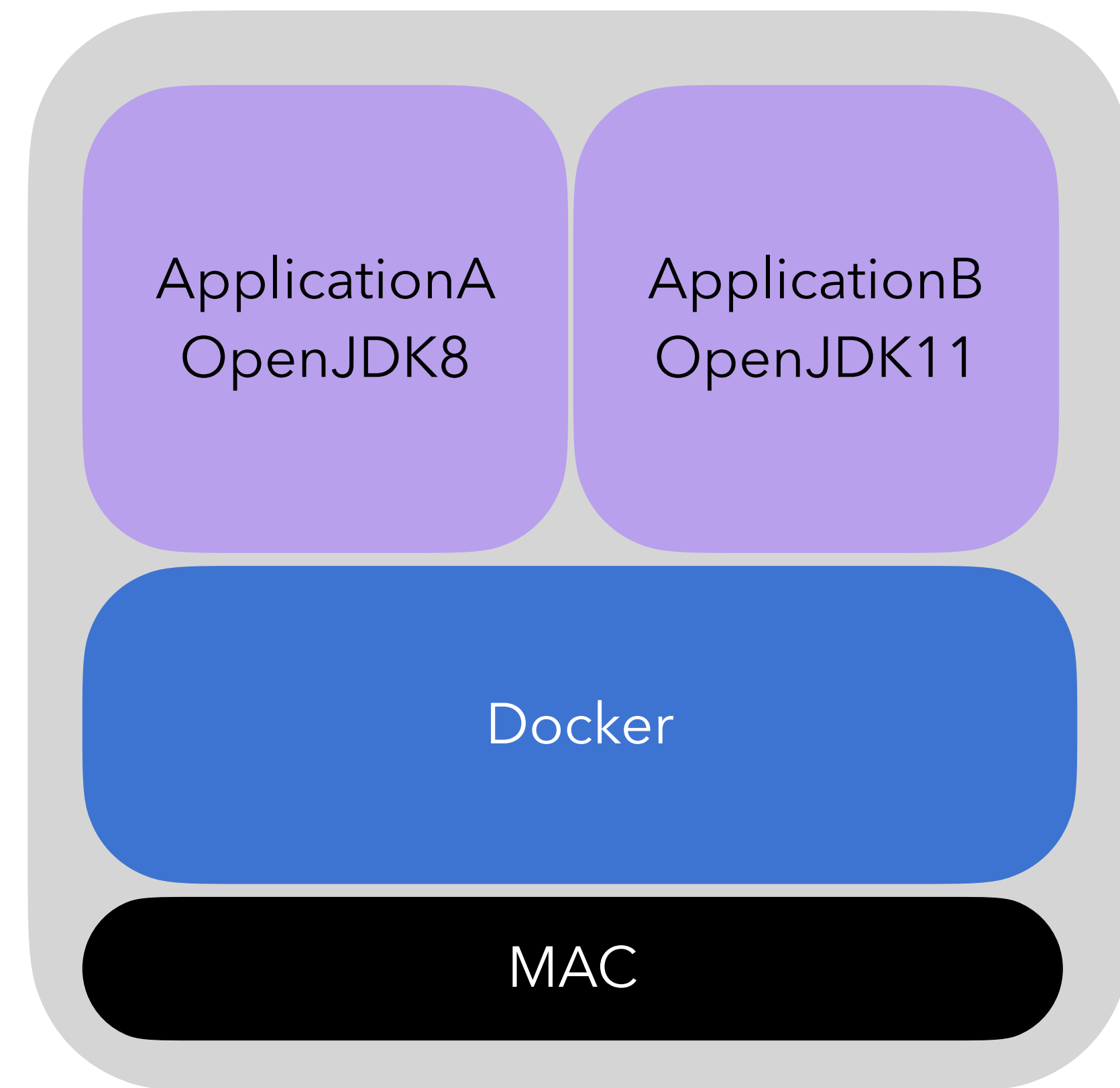
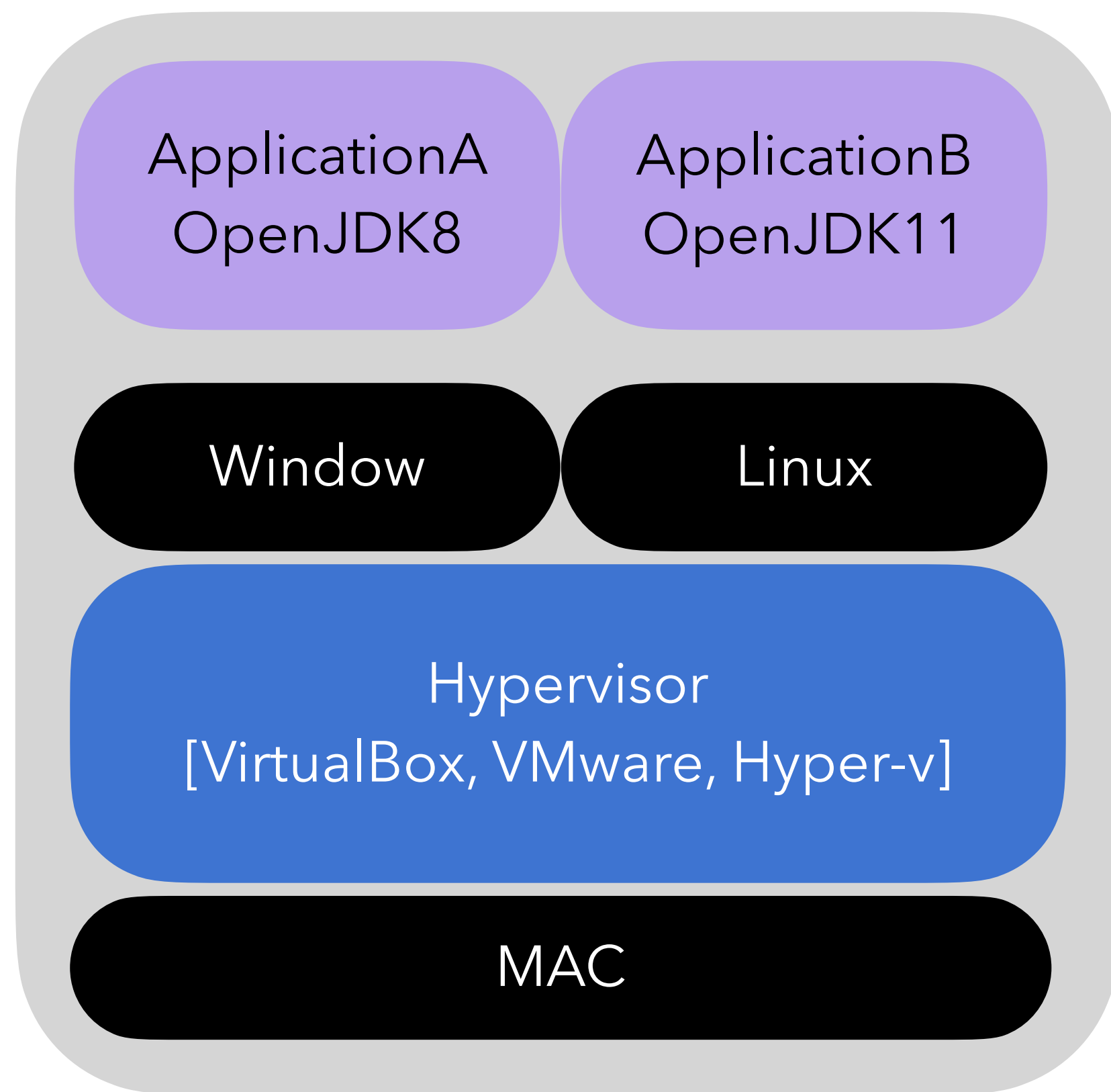
Virtual Machines



Containers

# VM vs. Container

白話文





# Clarify Docker Words

# Docker Words

- Docker, Inc
  - dotCloud 公司的名稱
- Docker Hub
  - 官方提供的 Docker Registry
- Docker Desktop
  - 官方提供快速安裝Docker環境的應用程式
- Dockerfile
  - 製作 image 的製作文件
- Docker Compose
  - 同時管理多個 container 的工具
- Docker
  - 應用程式打包成 container 並管理的工具

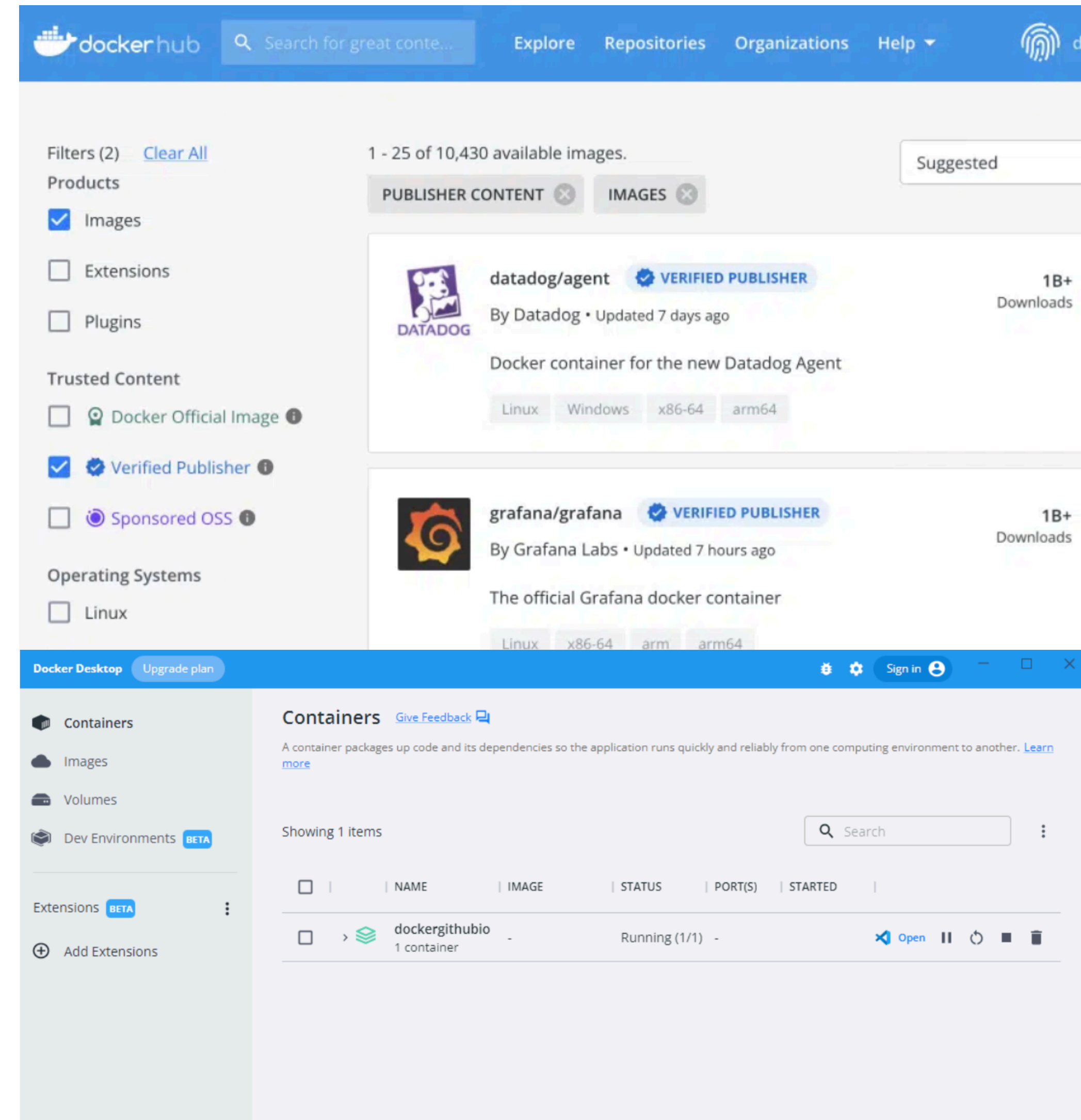
# Docker Words

- Docker, Inc  
dotCloud 公司的名稱



- Docker Hub  
官方提供的 Docker Registry

- Docker Desktop  
官方提供快速安裝Docker環境的應用程式



# Docker Words

```
Dockerfile x
Dockerfile > ...
1 FROM golang
2 ARG name
3 RUN mkdir -p WORK_REPO/logs
4 WORKDIR /WORK_REPO
5 EXPOSE 8080
6 ADD $name.go .
7 RUN go build -o exec_file $name.go
8 ENTRYPOINT [ "./exec_file" ]

docker-compose.yml x
docker-compose.yml
1 version: '2'
2 services:
3   server:
4     build:
5       context: .
6       args:
7         name: server
8     image: server
9     container_name: server_container
10    expose:
11      - "8080"
12    volumes:
13      - "$PWD/logs:/WORK_REPO/logs"
14    networks:
15      vlan:
16        ipv4_address: 172.18.0.2
```

- Dockerfile  
製作 image 的製作文件
- Docker Compose  
同時管理多個 container 的工具
- Docker  
應用程式打包成 container 並管理的工具

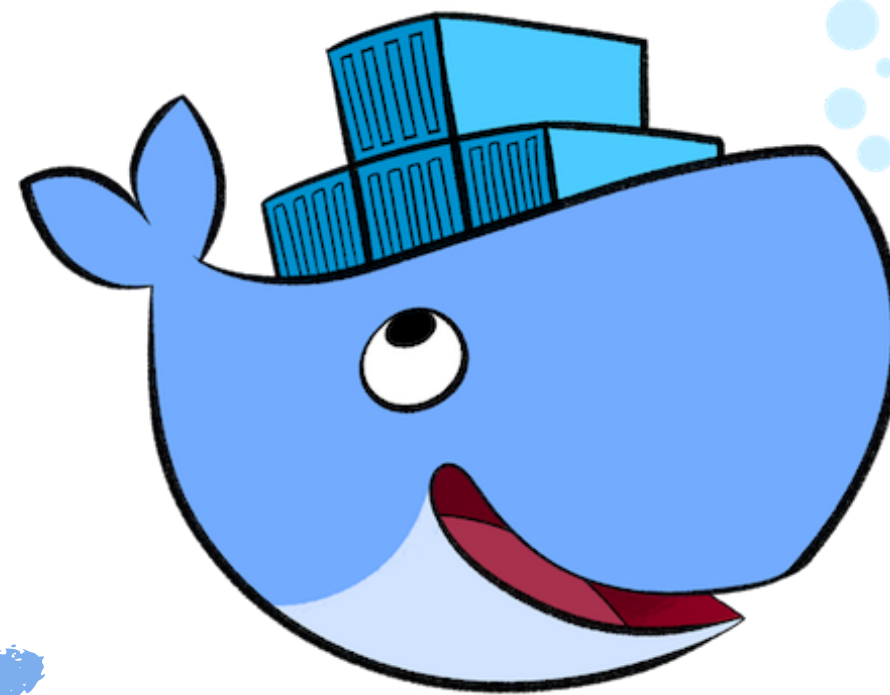


# Container生產線

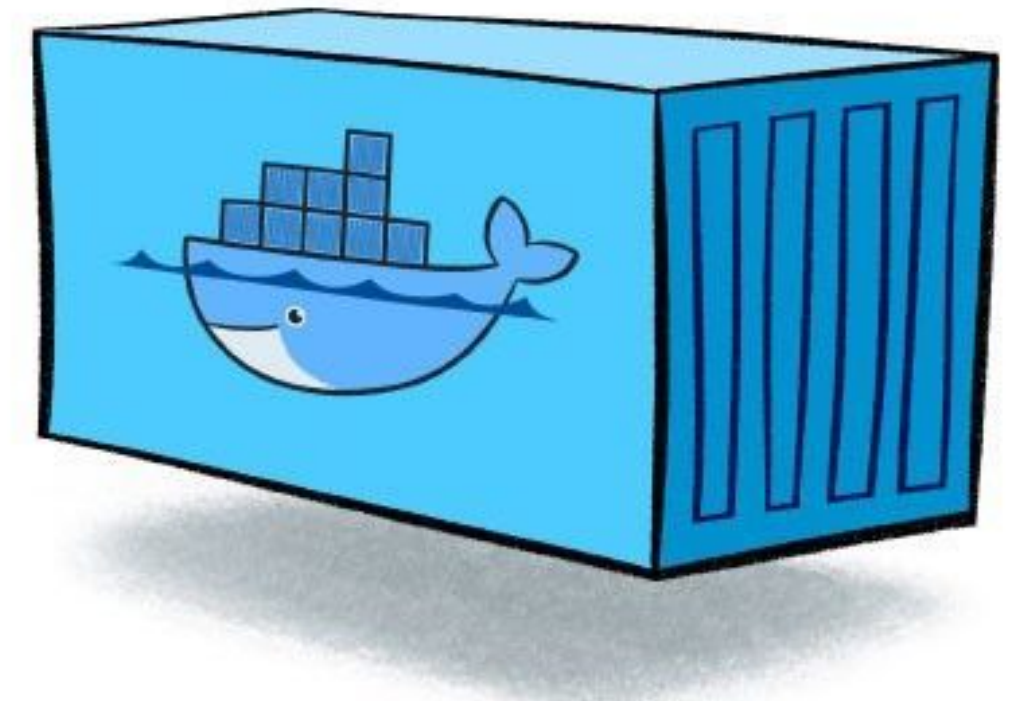
自製 Image

```
Dockerfile ×  
Dockerfile > ...  
1 FROM golang  
2 ARG name  
3 RUN mkdir -p WORK_REPO/logs  
4 WORKDIR /WORK_REPO  
5 EXPOSE 8080  
6 ADD $name.go .  
7 RUN go build -o exec_file $name.go  
8 ENTRYPOINT [ "./exec_file" ]
```

Dockerfile

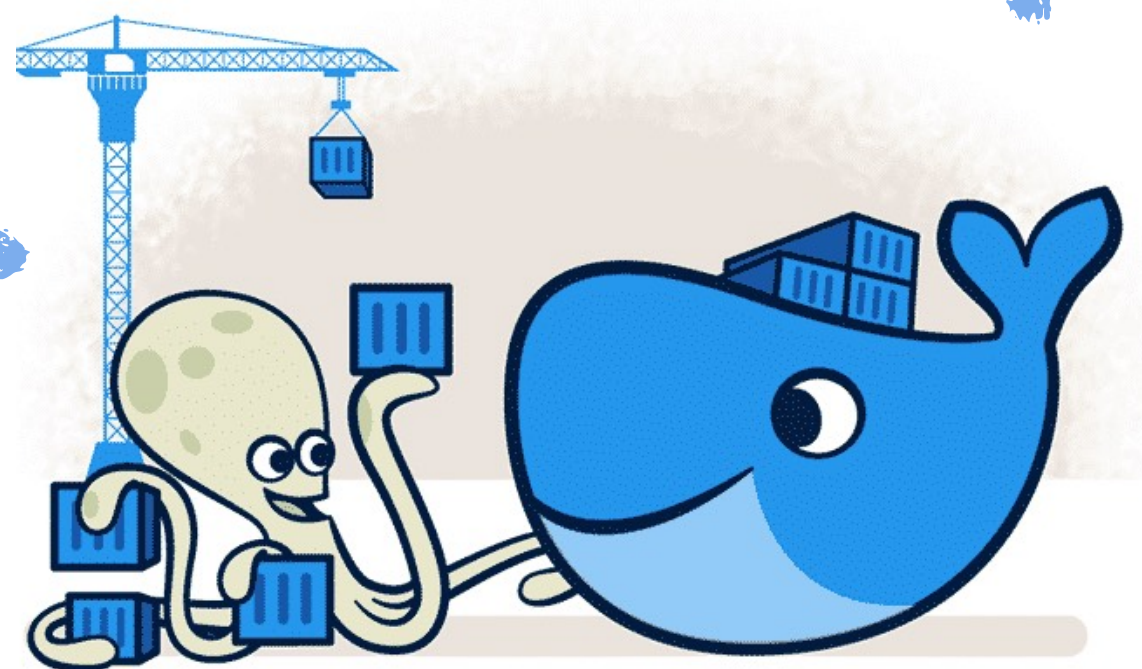


Image

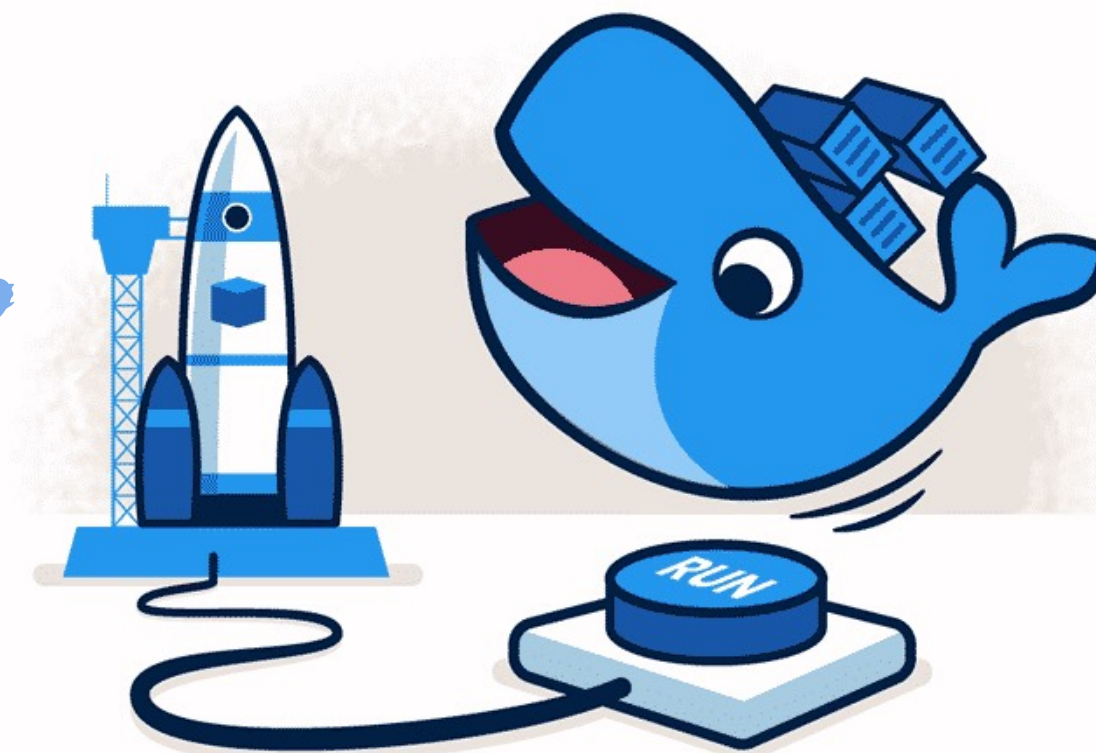


Container

Docker build



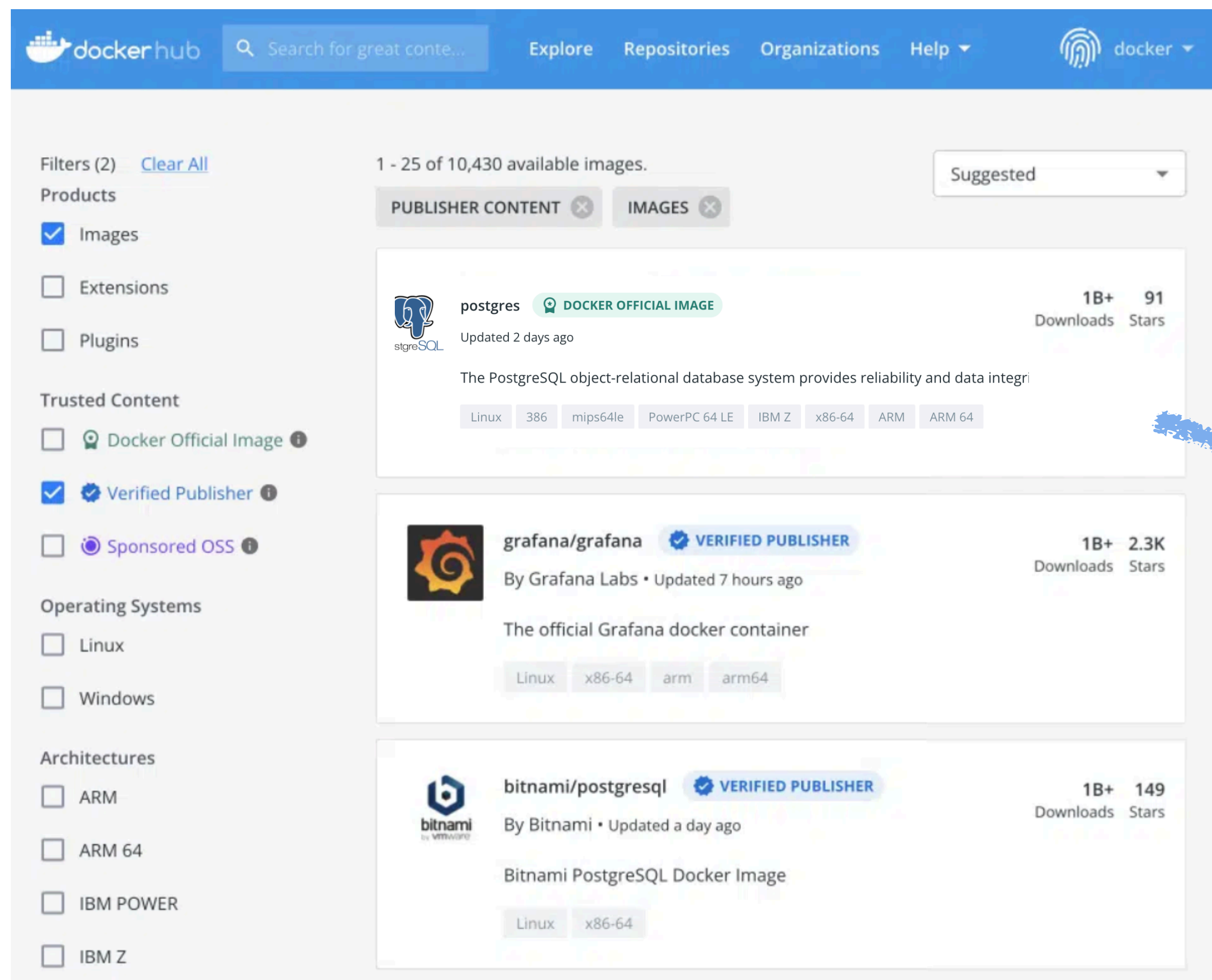
Docker run



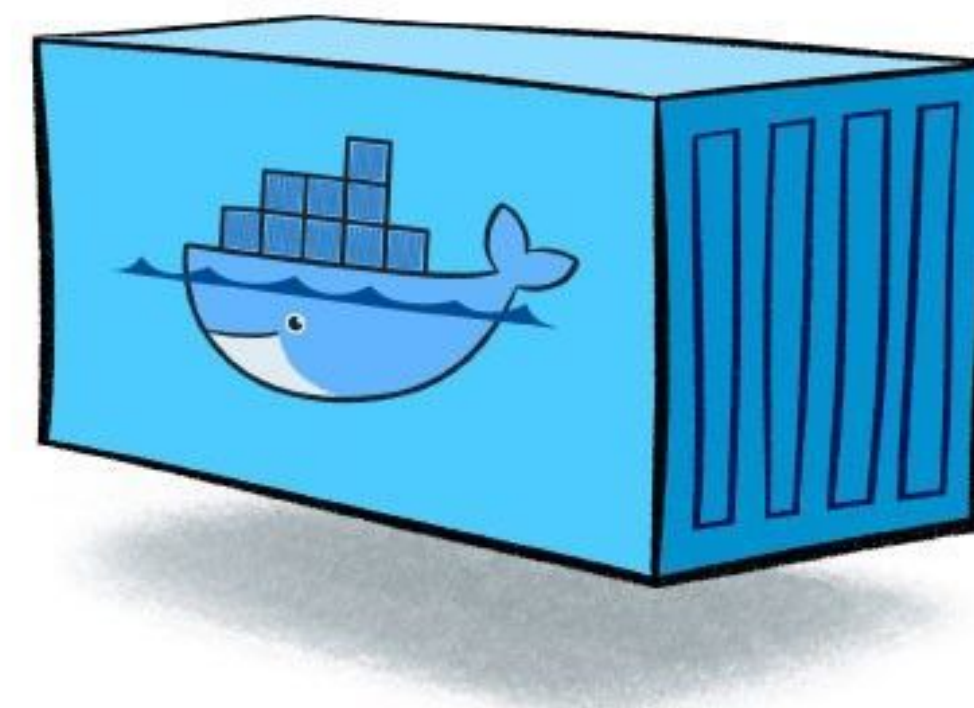


# Container 生產線

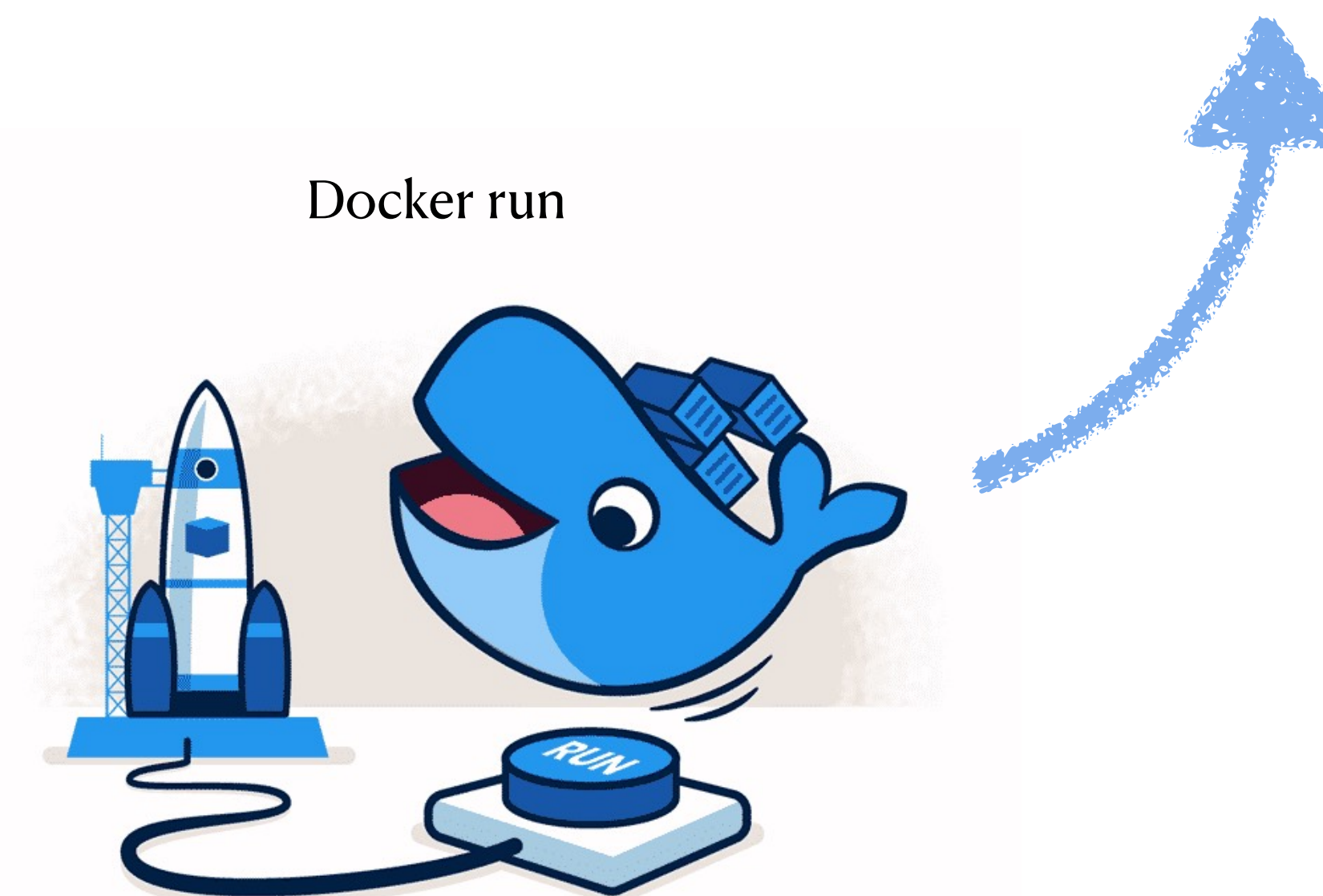
使用現成Image



Docker Hub



Container



Docker run

# Hands on Docker

# Play Time

- 本機安裝：<https://www.docker.com/get-started/> (回家自己裝)
- 快速環境：<https://labs.play-with-docker.com/> (今日遊戲場)
- Step 1:
  - 登入 <https://labs.play-with-docker.com/>
  - 開啟 <https://github.com/cathaylife-devops/docker-tutorial>

# Docker Command

## 初次見面

- Docker info
- Docker ps
- Docker pull
- Docker create(冷)
- Docker start
- Docker run
- Docker rm
- Docker rmi
- Docker history
- Docker logs

# Docker Command

溫故知新

- Docker info
- Docker ps
- Docker pull [image]
- Docker create(冷) [image]
- Docker start [containers]
- Docker run [image]
- Docker rm [containers]
- Docker rmi [image]
- Docker history [image]
- Docker logs [containers]



# Play More

## Docker run

- Docker run
  - d —detach      Run container in background and print container ID
  - p —public      Publish a container's port(s) to the host
  - t —tty      Allocate a pseudo-TTY
  - v —volume      Bind mount a volume
  - name.      Assign a name to the container

# Review

- Docker 輕量級虛擬化技術，方便管理容器的工具，讓開發環境更統一不受干擾。
- Container是由 Image 製成，可以用同一個image 製成很多個，但 name 不能重複。
- Image 是由Dokcerfile製成，也可以從 Repository 拉取。
- 公開的 Repository 有現成的 Image 可以使用。
- -- help 是你的好夥伴

Q&A

# Refrence

- <https://www.docker.com/>
- <https://www.netadmin.com.tw/netadmin/zh-tw/feature/1DB50AE57AC1401F98C641366F8D0720>
- [https://philipzheng.gitbook.io/docker\\_practice/introduction/what](https://philipzheng.gitbook.io/docker_practice/introduction/what)
- <https://cwhu.medium.com/docker-tutorial-101-c3808b899ac6>
- <https://www.weave.works/blog/a-practical-guide-to-choosing-between-docker-containers-and-vms>
-