

Docker, Kubernetes

技術簡介

November 8



Today's Objectives

- **Container 容器化架構的發展歷史**
- **容器化架構與虛擬機器的比較**
- **Container 市場與產品介紹**
- **Docker 發展及功能架構介紹**
- **Docker CE 安裝、映像檔管理、容器管理**
- **Kubernetes 安裝、使用YAML檔、容器管理、實例演練**

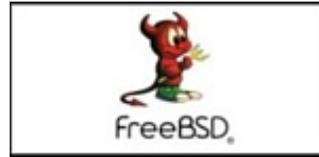


The History of Container Technology



1980s

- Chroot
- Unix V7
- Running Processes



1990s

- Jail
- FreeBSD
- Network Security
- Sandboxing



2000s

- Linux vServer
- Solaris Zone
- cgroups
- LXC
- Isolate instance

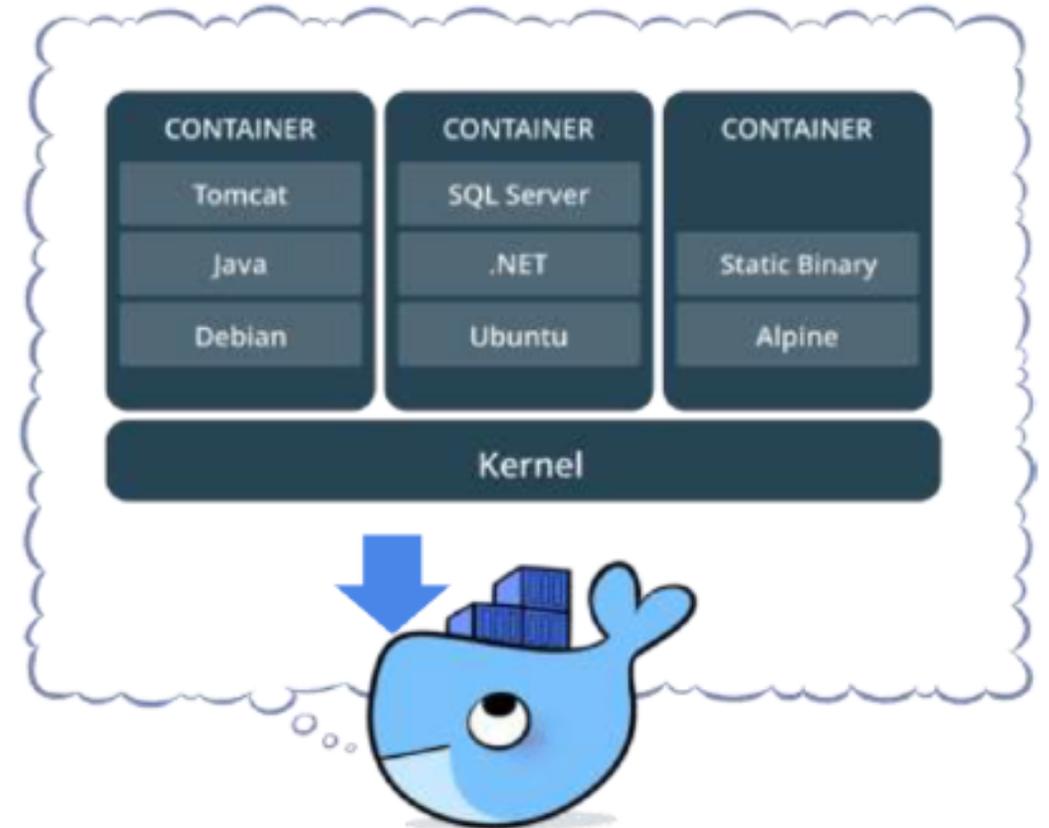


2010s

- Warden
- LMCTFY
- Docker
- Kubernetes
- API, CLI client

What is a container?

- **Standardized packaging for software and dependencies**
- **Isolate apps from each other**
- **Share the same OS kernel**
- **Work with all major Linux and Windows Server**



Days before container?



Matrix from hell

Static Website	?	?	?	?	?	?	?	?
Web Frontend	?	?	?	?	?	?	?	?
Background Workers	?	?	?	?	?	?	?	?
User DB	?	?	?	?	?	?	?	?
Analytics DB	?	?	?	?	?	?	?	?
Queue	?	?	?	?	?	?	?	?
	Desktop 	Test/QA Cluster 	Production Cluster 	Public Cloud 	Data Center 	Mainframe 	Windows Server 	Edge Device 

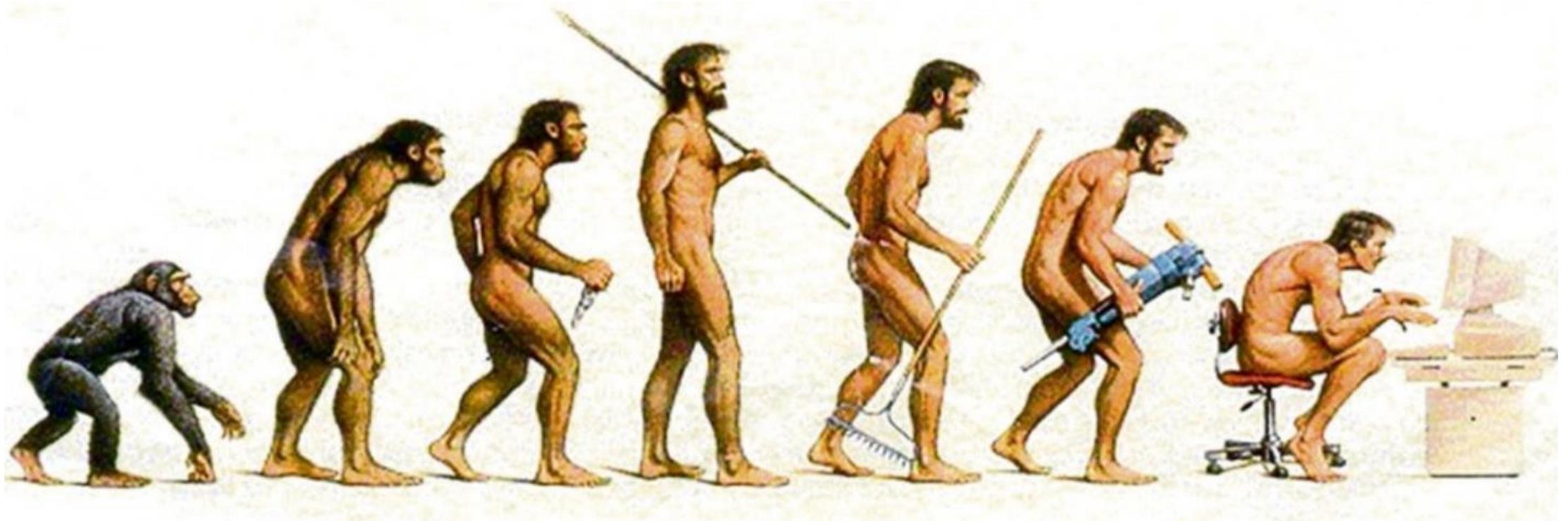
Containers cut complexity



What is Docker ?

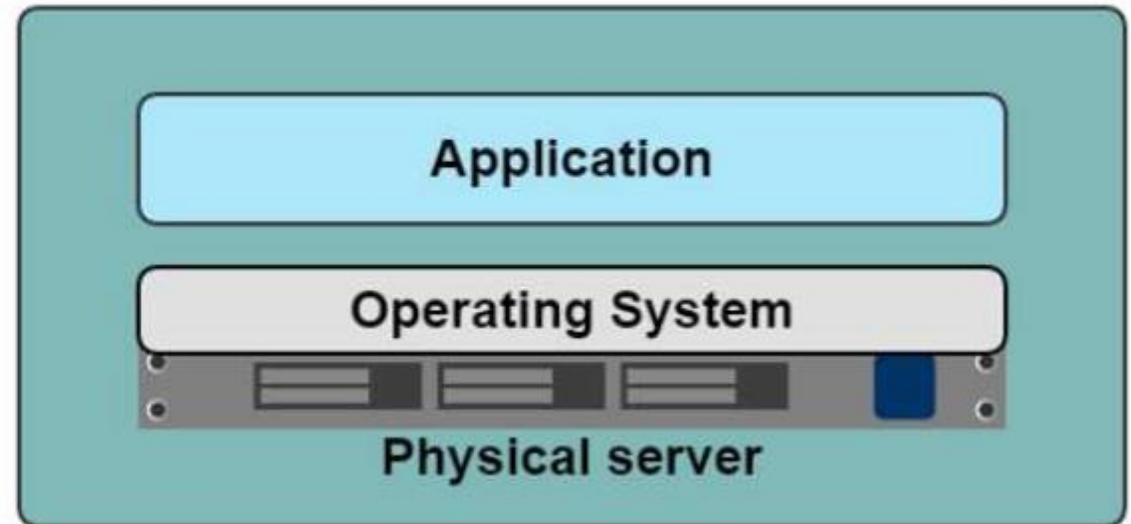
- **Docker** 是一個能讓程式容易建立、佈署並且運行在容器內的工具
- **Docker** 的設計能讓程式開發者及管理者有許多**DevOps (Developers + Operations)**工具來提升效益
- 程式的設計師可以更專心撰寫程式，而不用擔心最後所要運轉的系統是什麼
- 對於維運人員來說，**Docker**佔用空間較小而且成本較低，因此具有靈活性，並可僅量減少系統的數量

The journey so far...



傳統程式開發的限制

- 佈署速度緩慢
- 成本高昂
- 資源浪費
- 難以擴張
- 難以移轉
- 不易更換廠牌



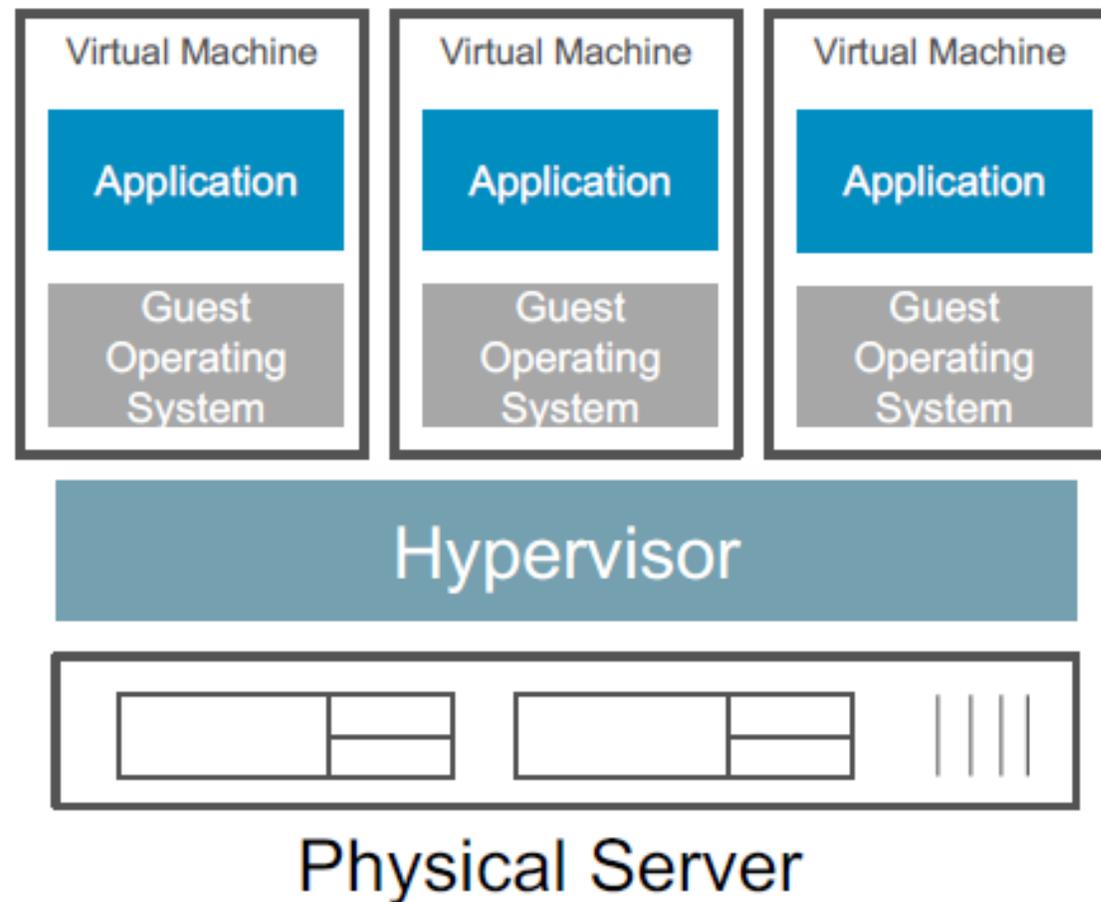
虛擬化架構

■ 優點

- 以資源池方式進行更好的管理
- 容易擴張
- 可在雲端佈署，即付即用

■ 限制

- 虛擬機仍然需要CPU, RAM, Storage, OS
- 每個服務都需要完整作業系統代表著資源的浪費
- 無法保證軟體的移動性



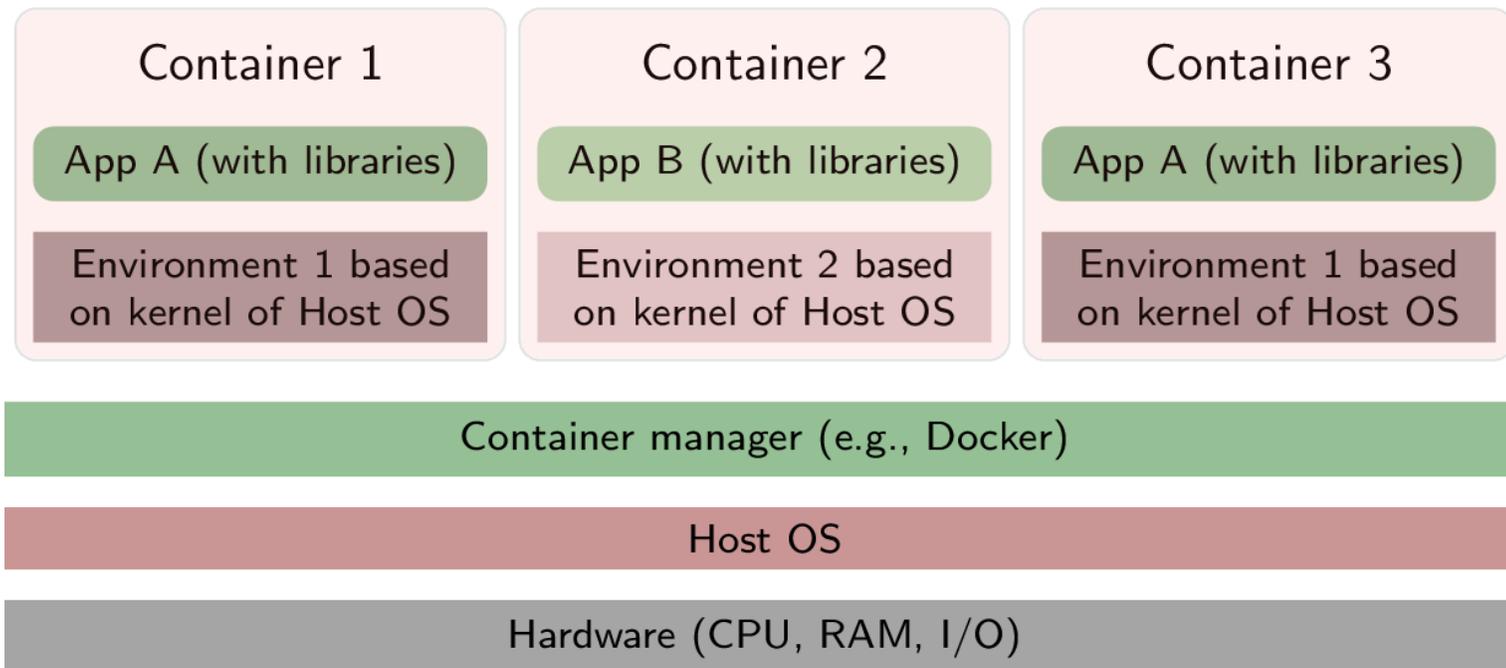
容器化架構

■ 優點

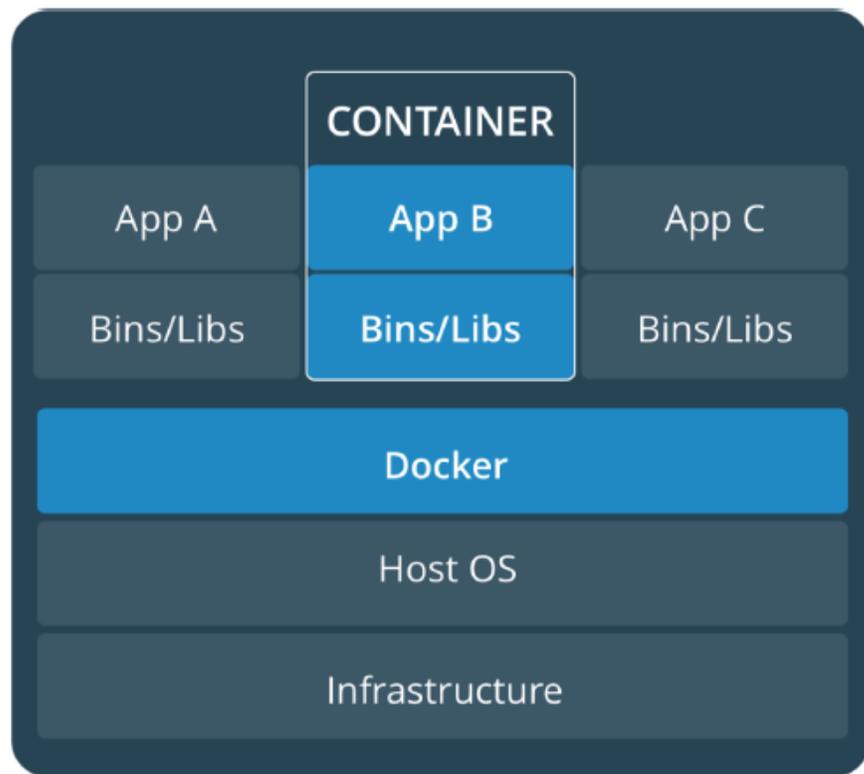
- 大幅降低硬碟空間需求
- 消耗更少的運算資源
- 啟動迅速

■ 限制

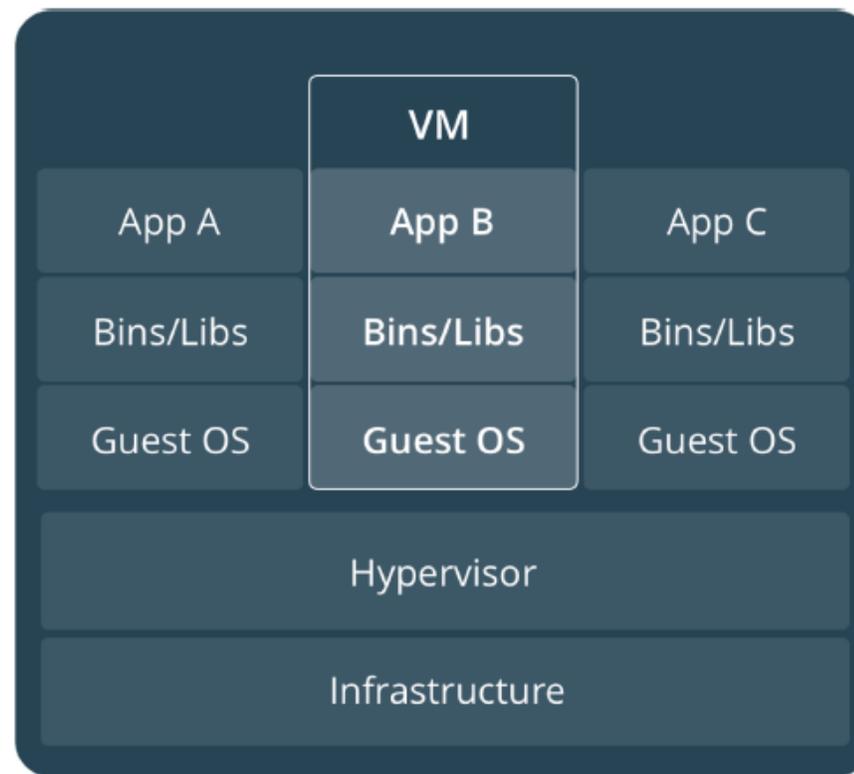
- 備份備援設計門檻較高
- 無法動態調整容器的資源分配
- 無法保證軟體的移動性



容器化 V.S. 虛擬化

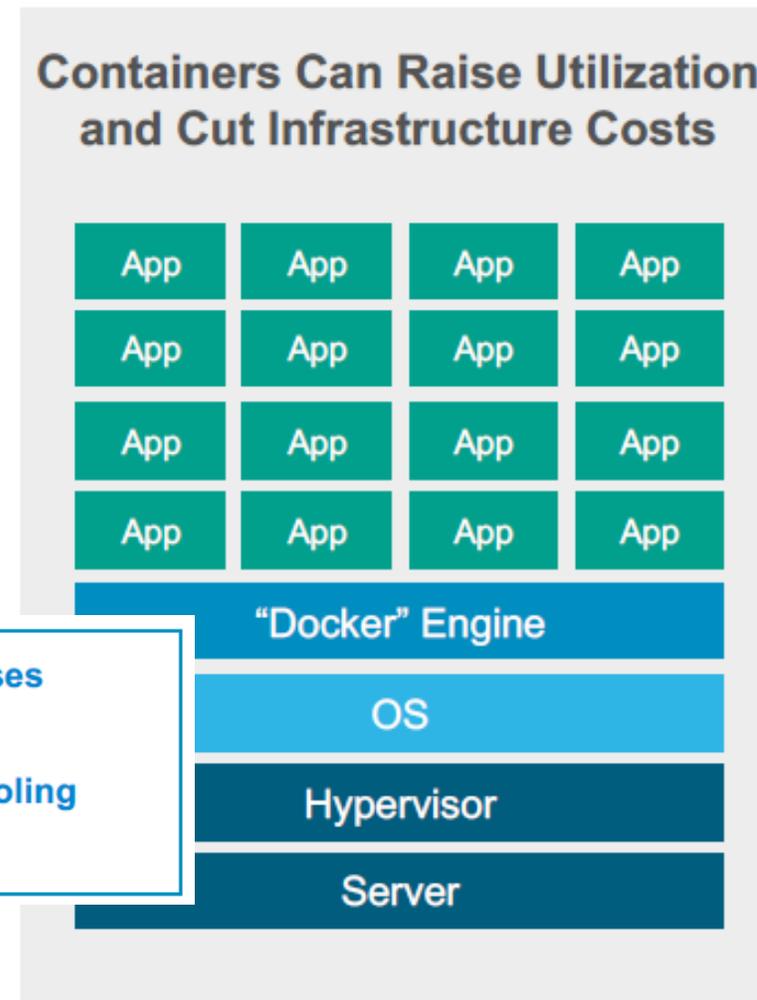
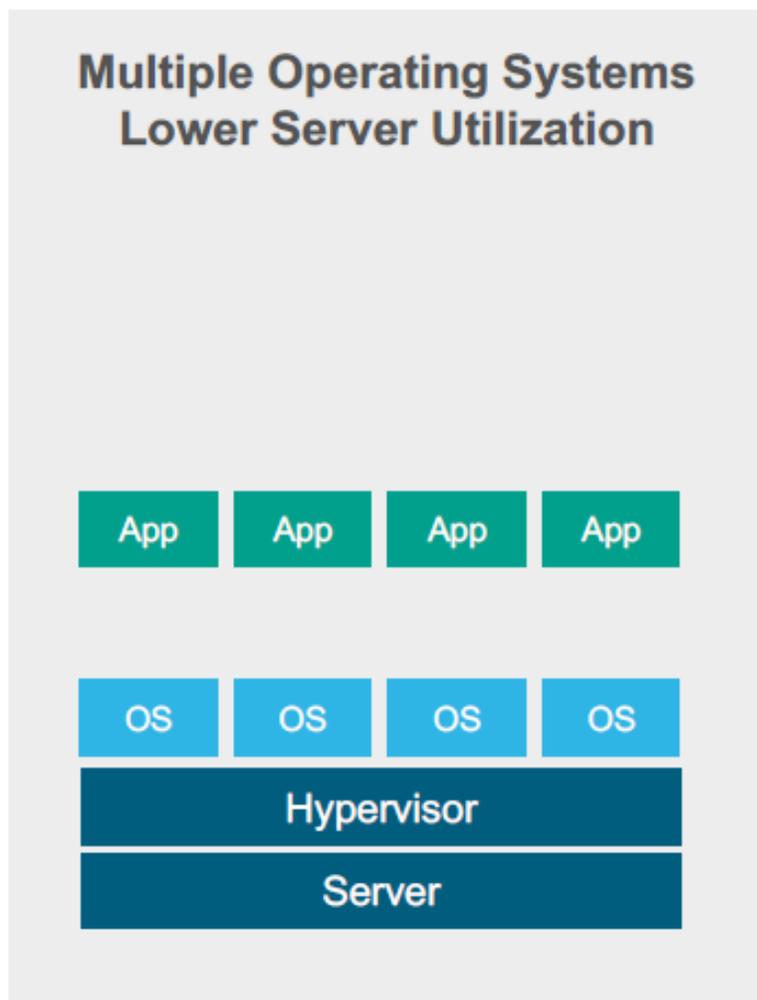


Containers are an app level construct



VMs are an infrastructure level construct to turn one machine into many servers

容器化在基礎架構的優勢



Reducing

- VM and OS Licenses
- Server Count
- Power, Space, Cooling
- Admin

Docker 的基本零件



Image

The basis of a Docker container. The content at rest.



Container

The image when it is 'running.' The standard unit for app service



Engine

The software that executes commands for containers. Networking and volumes are part of Engine. Can be clustered together.



Registry

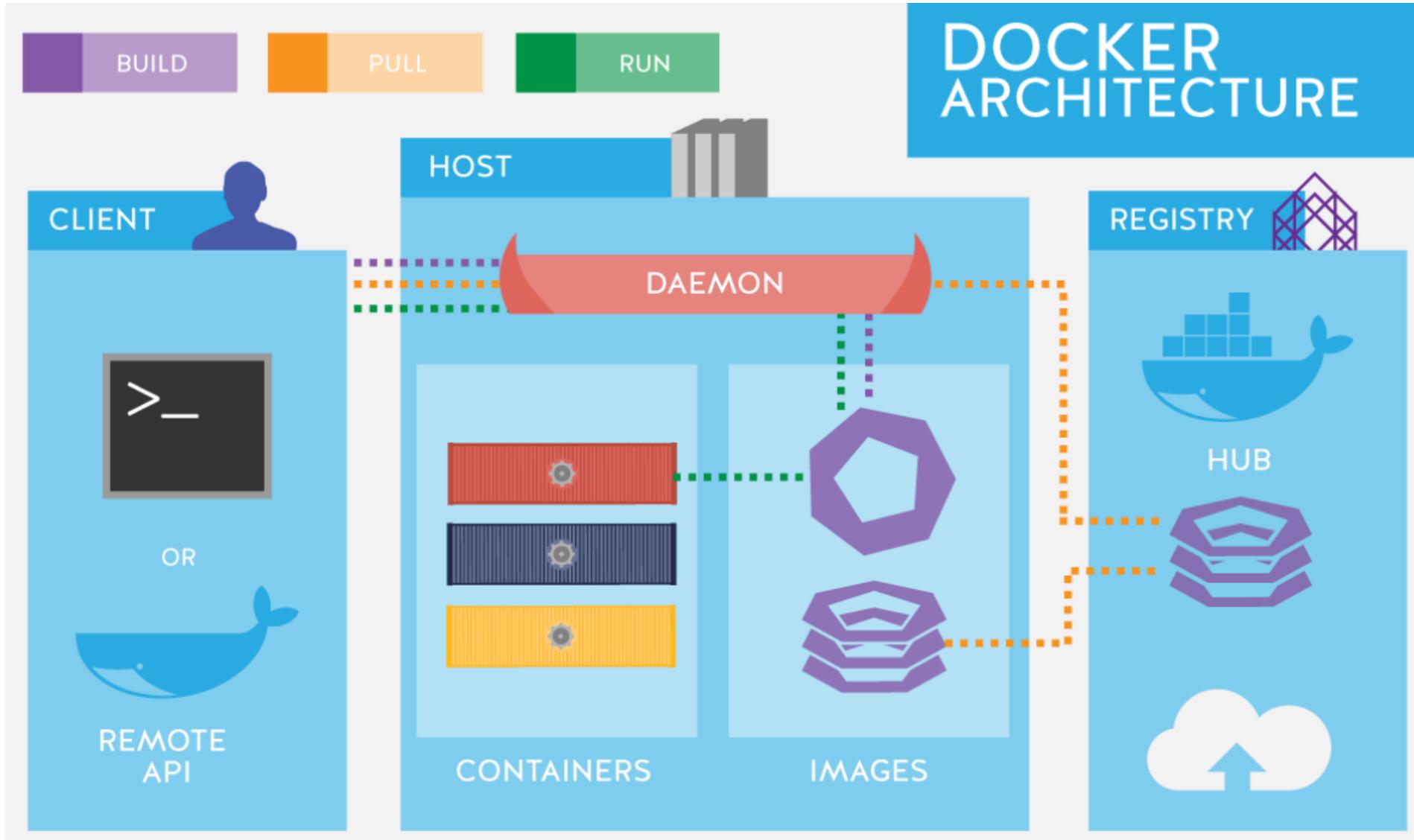
Stores, distributes and manages Docker images



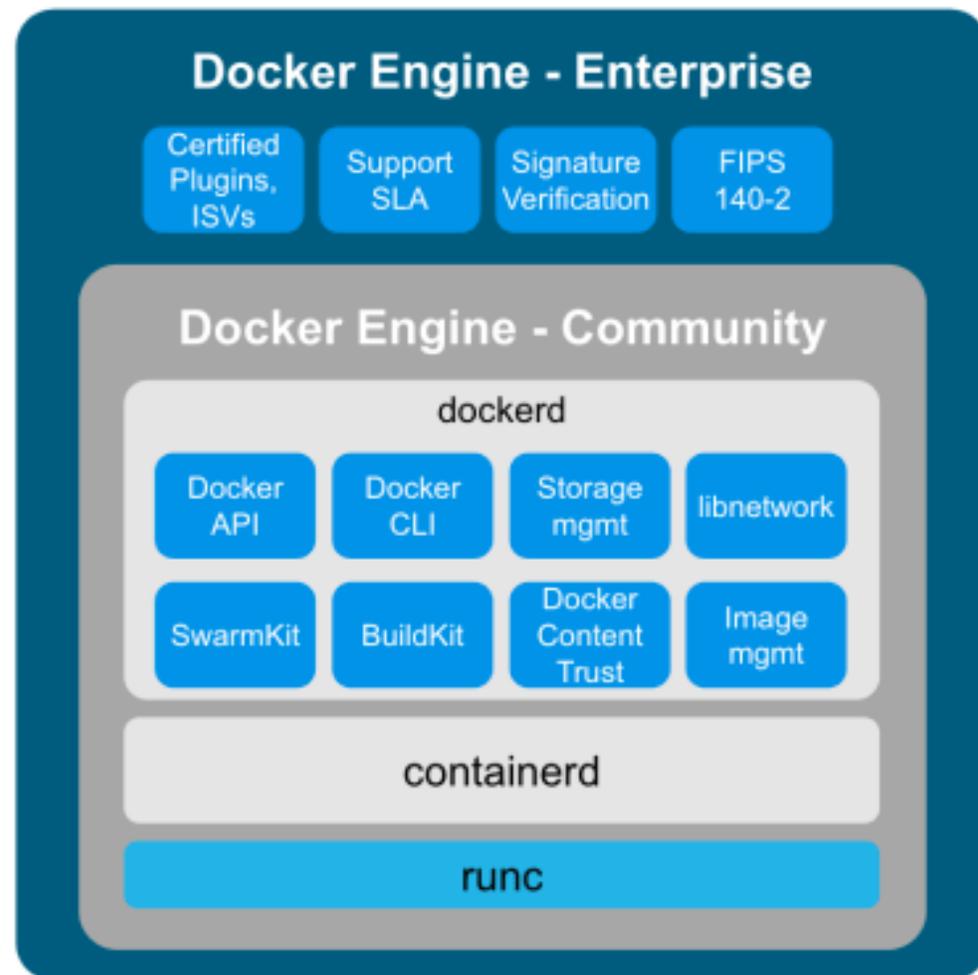
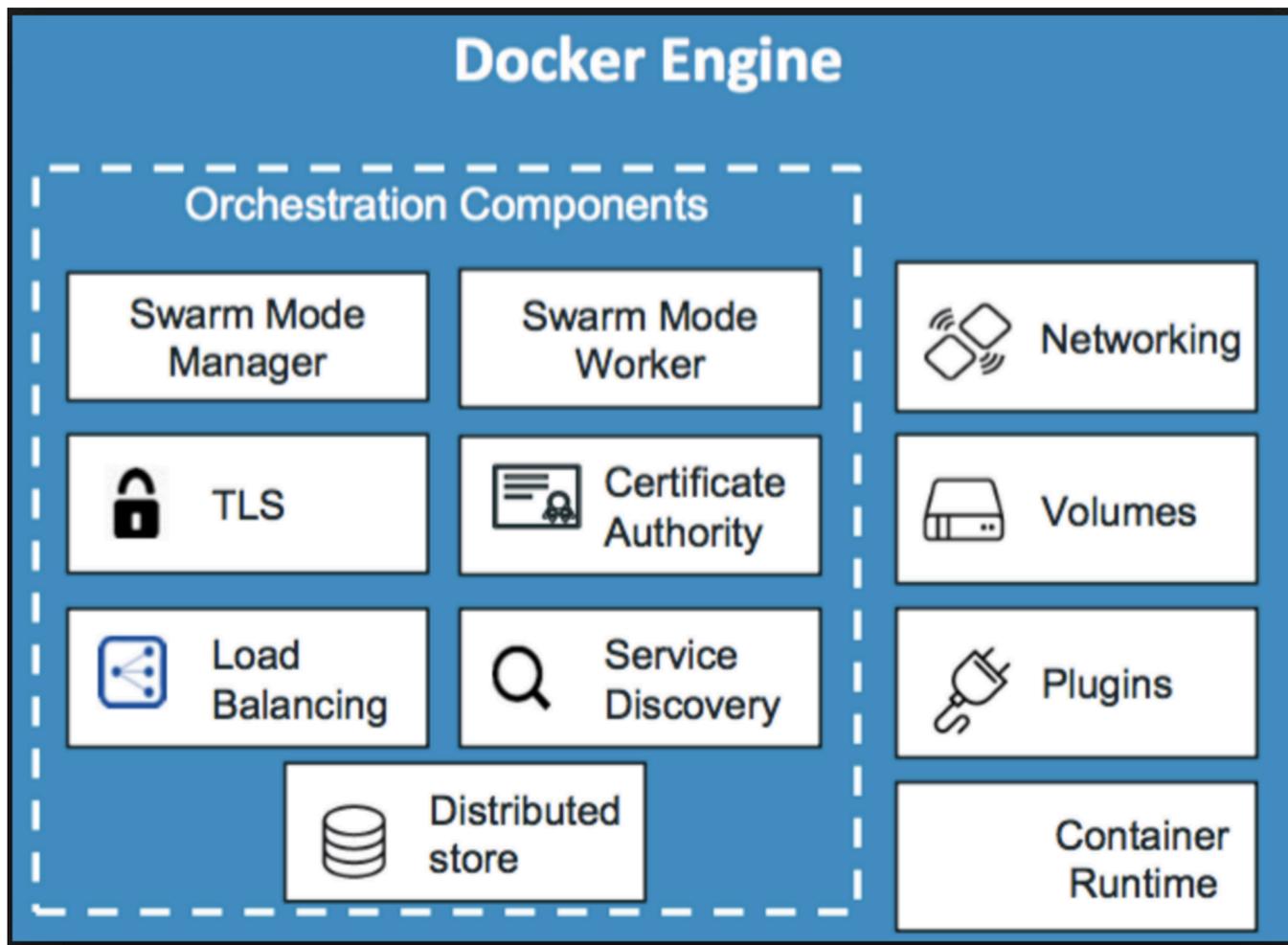
Control Plane

Management plane for container and cluster orchestration

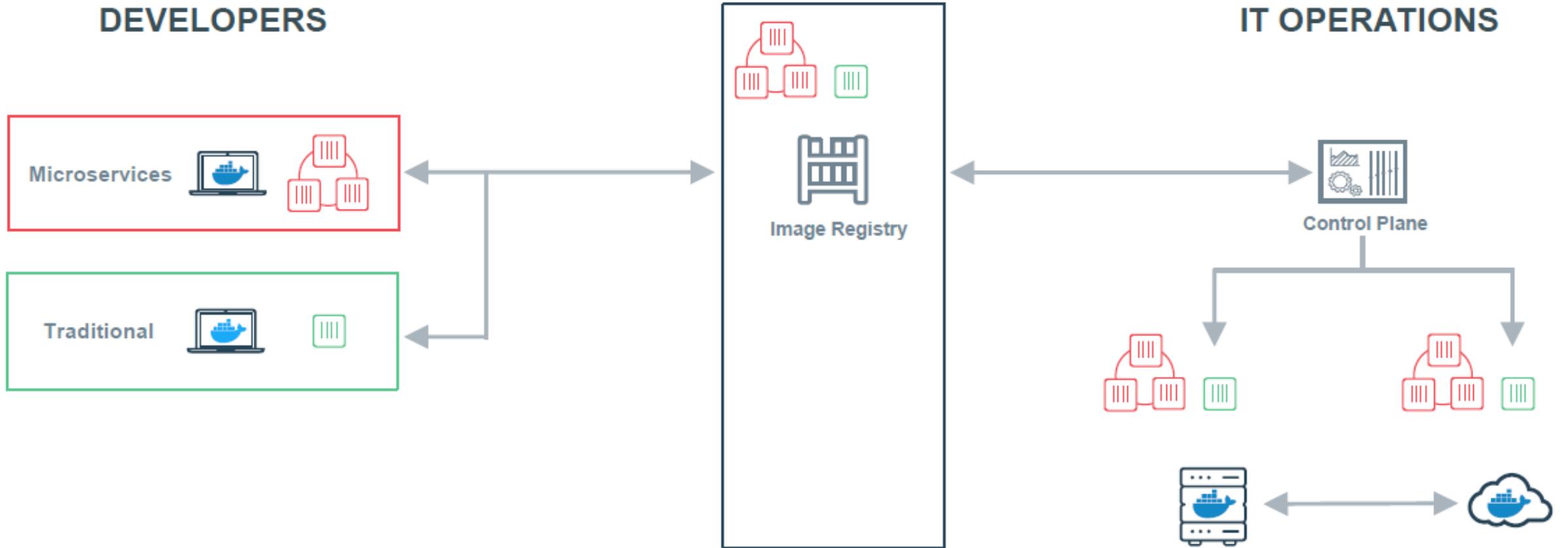
Docker 的運作架構



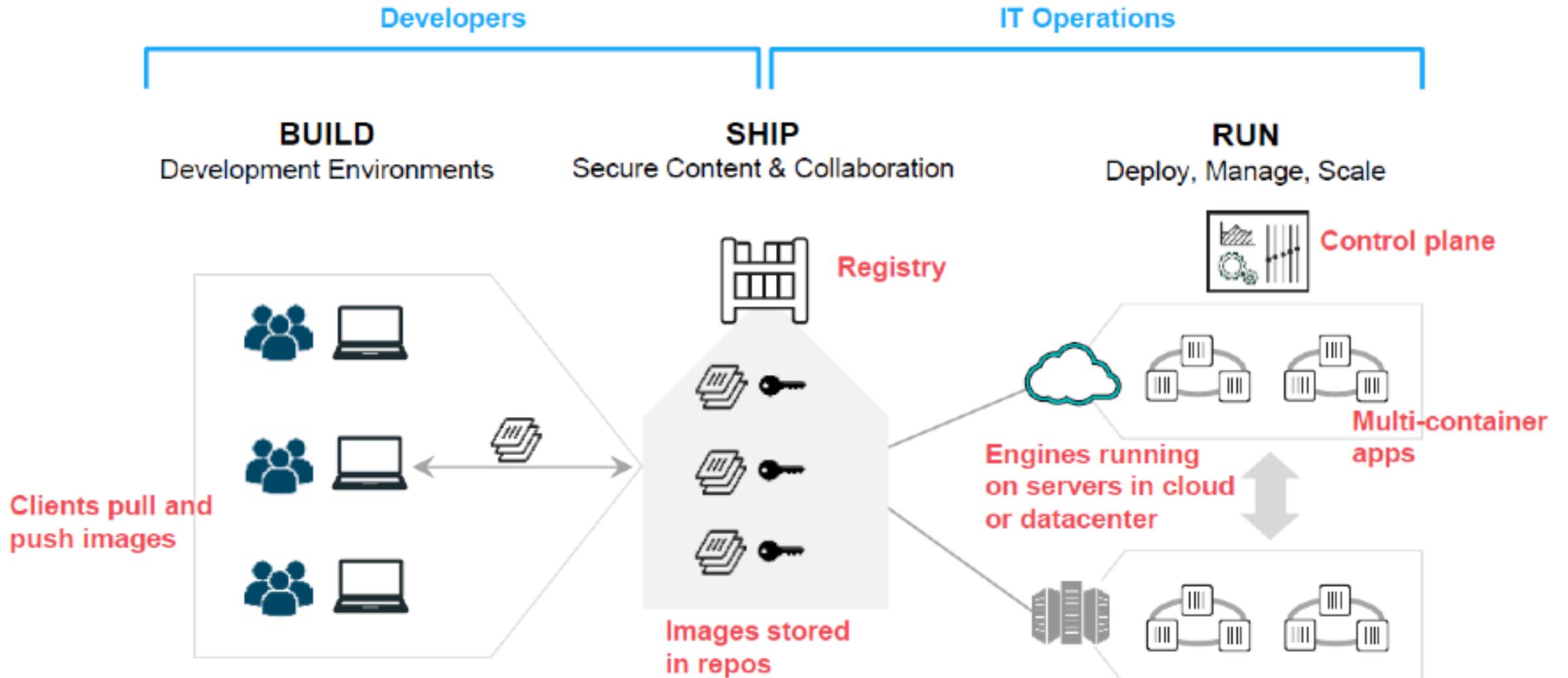
Docker Engine



建立新的軟體供應



CaaS 容器化服務



THE LAB

Lab

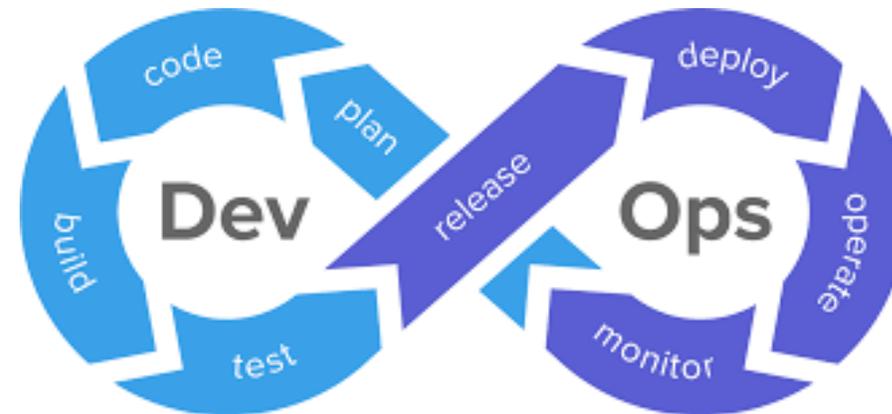
..And lab some more

What is Kubernetes?

- 同時部署多個 **containers** 到一台機器上，甚至多台機器。
- 管理各個 **container** 的狀態。如果提供某個服務的 **container** 不小心 **crash** 了，**Kubernetes** 會偵測到並重啟這個 **container**，確保持續提供服務
- 將一台機器上所有的 **containers** 轉移到另外一台機器上。
- 提供機器高度擴張性。**Kubernetes cluster** 可以從一台機器，延展到多台機器共同運行。



K8S架構的影響



- 將龐大的專案拆成幾個不同面向的小專案，當代碼夠小、容易理解、開發效率能被提高
- 各個服務之間也可獨立部署，不因一個服務癱瘓而癱瘓整個系統
- 各團隊可以依照自己的需求使用適合自己的語言、資料庫開發
- 每個服務也可以依照自己的需求，選擇在不同機器上部署

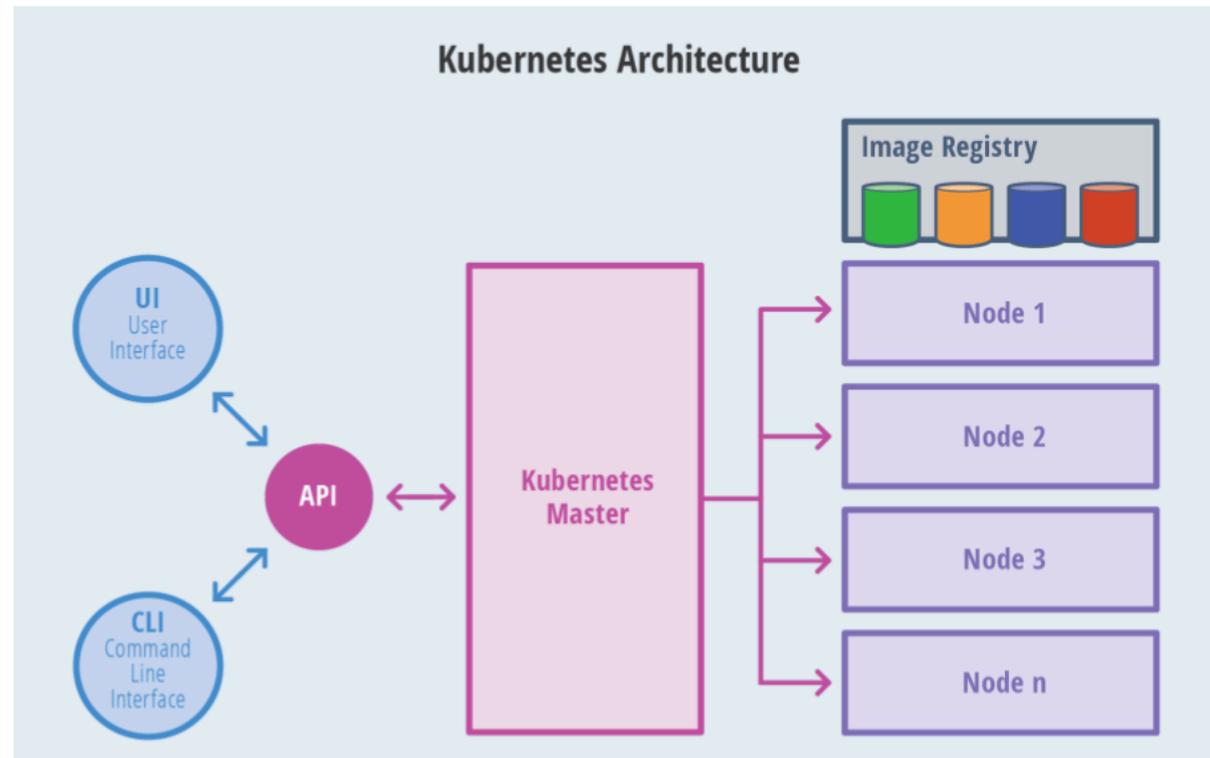
K8S 的優點

- 可以跑在任何地方 **Can run anywhere**
- 高度模組化 **High modular**
- 活躍的社群 **Open source & active community**
- **Google的背書 Backed by Google**



K8S 的元件有什麼

- Pod
- Replication Controller
- Service
- Deployment
- Secrets



THE LAB

Lab

..And lab some more



We are architects of change
Together we own the possible