究竟 Kubernetes 哪裡受歡迎？

我是誰

- 安安我是 Rico
- 翻譯《Google The Site Reliability Workbook》
- 曾在 Bincentive 工作
- DevOps Taiwan 志工
- 冰雪奇緣愛好者

????

我是誰

Gopher!!!!

大綱

如何用正確的姿勢閱讀這個簡報

QDD

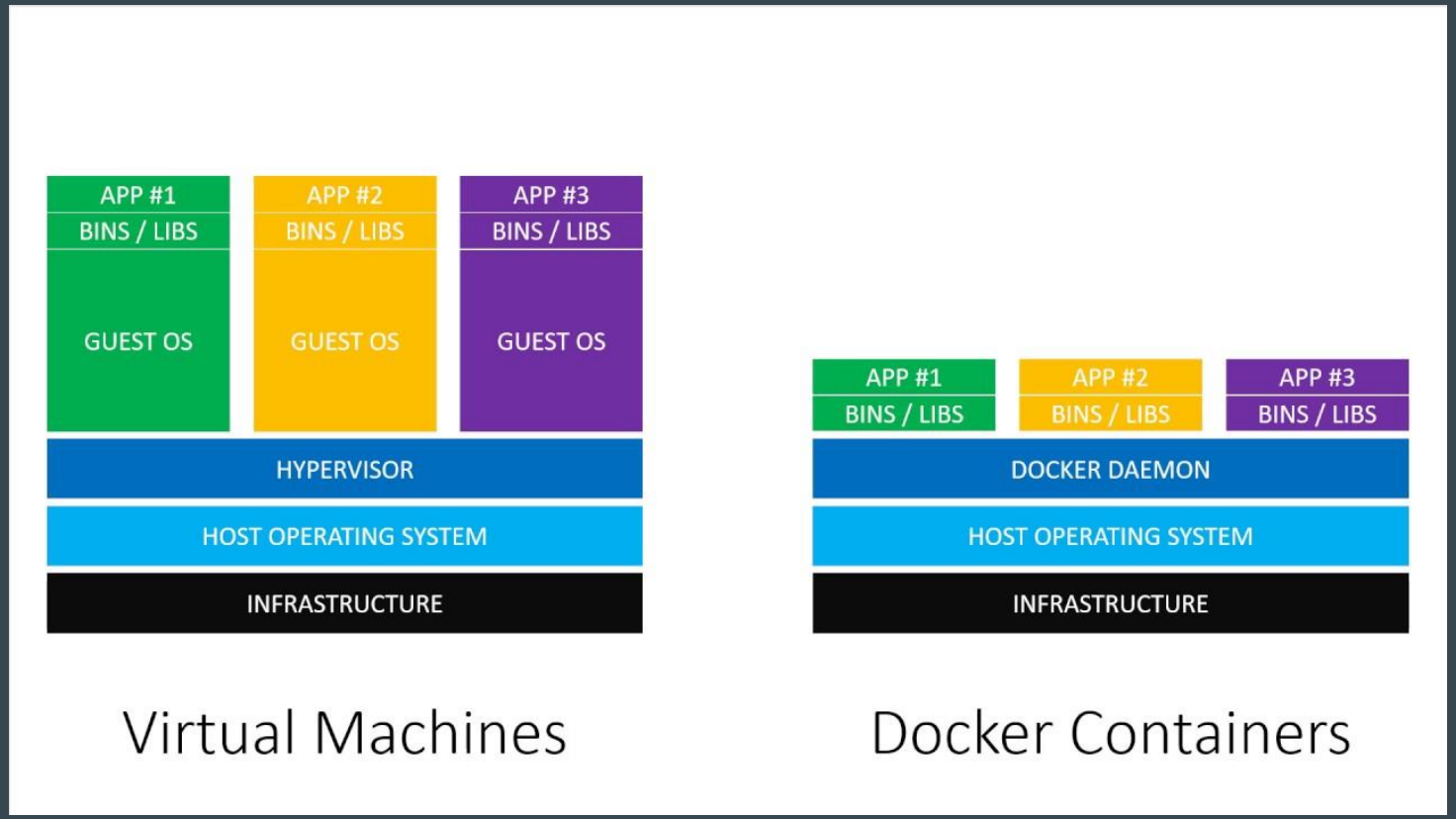
Question-driven design 是在下本人想出的演講方式，第一次在公開場合實驗，希望結束後能夠得到心得回饋。

QDD 是以問題做驅動的設計，在演講時以問題來做龐大知識的切入點，問對問題就可以刀刀見骨看清知識的本質。

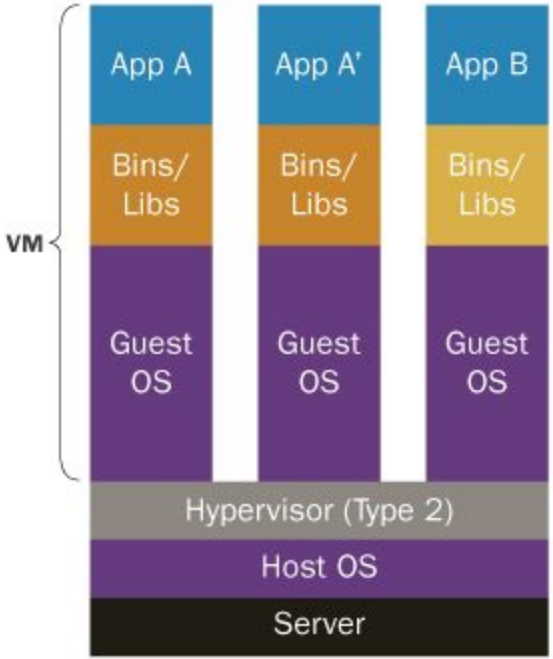
理想的情況就是能夠激起聽眾多問問題的意願，越是簡單或看似很笨的發問有時候就是那一刀見骨的問題。

Container 基礎概念

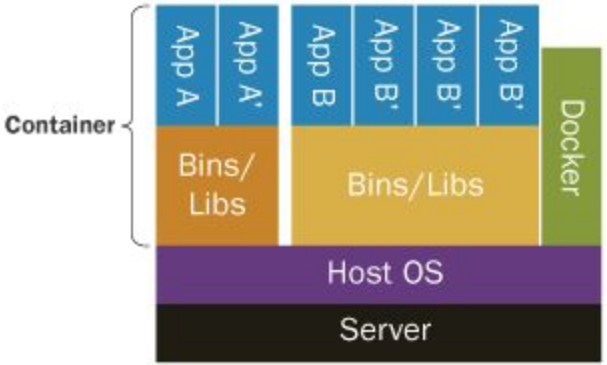
Container 跟 VM 差在哪裡？



Containers vs. VMs



Containers are isolated, but share OS and, where appropriate, bins/libraries



所以 container 到底是怎麼實作的？

1. Container 的本質就是 process（行程），跟 VM 那種扎扎實實虛擬化技術的 Hypervisor 是不同的。
2. Container 工具有很多種，像是 lxc、rkt、podman、containerd 還有最著名的 docker，每個 container 底層實作都不同，這邊都以 docker 做範例。
3. Docker 主要是靠 Cgroup 和 Linux Namespace 做實踐（還是有很多細節沒有提到，但我們要趕火車）

Docker 與 Cgroup 的關係？

1. 全名為 control group
2. Cgroup 是用來限制 process 資源的工具，像是 CPU、記憶體、儲存、網路頻寬等等
3. 例如觀看 docker 現在的記憶體狀態可以到本機 /sys/fs/cgroup/memory/docker/<longid>/ 裡面看到

Docker 與 Linux Namespace 的關係？

1. Namespace 是限制「看」到的內容
2. Namespace 的機制可以讓 process 看不到其他的 process，有各式各樣的 Namespace 像是 PID、Mount、Network 或 User 各種 Namespace
3. 但是時間無法隔離的，只有 VM 才做得到
4. docker exec 這個指令的實踐原理就是進入 Linux Namespace

Docker 的網路跟儲存又是怎麼實踐的呢？

1. Docker 的網路跟儲存都深受 Cgroup & Namespace 的影響
2. 網路的實作是靠 Linux kernal 的 iptables 工具
3. 儲存的實作是靠 chroot 工具

Orchestration

什麼是 orchestration？

生活的例子裡，orchestration（編程、協調）就像是音樂 orchestra（管弦樂團）一樣，指揮家在台上指揮著各個不同的樂器，為了讓音樂的節拍、曲風和情感保持一致性。

Container orchestration 常見的工具有 docker-compose、Docker Swarm 和 Kubernetes。

管弦樂團的指揮家的目標跟系統的 orchestration 目標有什麼不同呢？

- 音樂指揮家的目標在於「為音樂服務」
- 系統 orchestration 的目標在於「為使用者服務」，當使用者想要依據服務的特性跑，orchestration 就要能夠做到使用者期望的事情，例如無狀態的網頁服務跟需要做 HA 的資料庫服務要用不同的方式跑起來。

系統 orchestration 實際在做什麼？

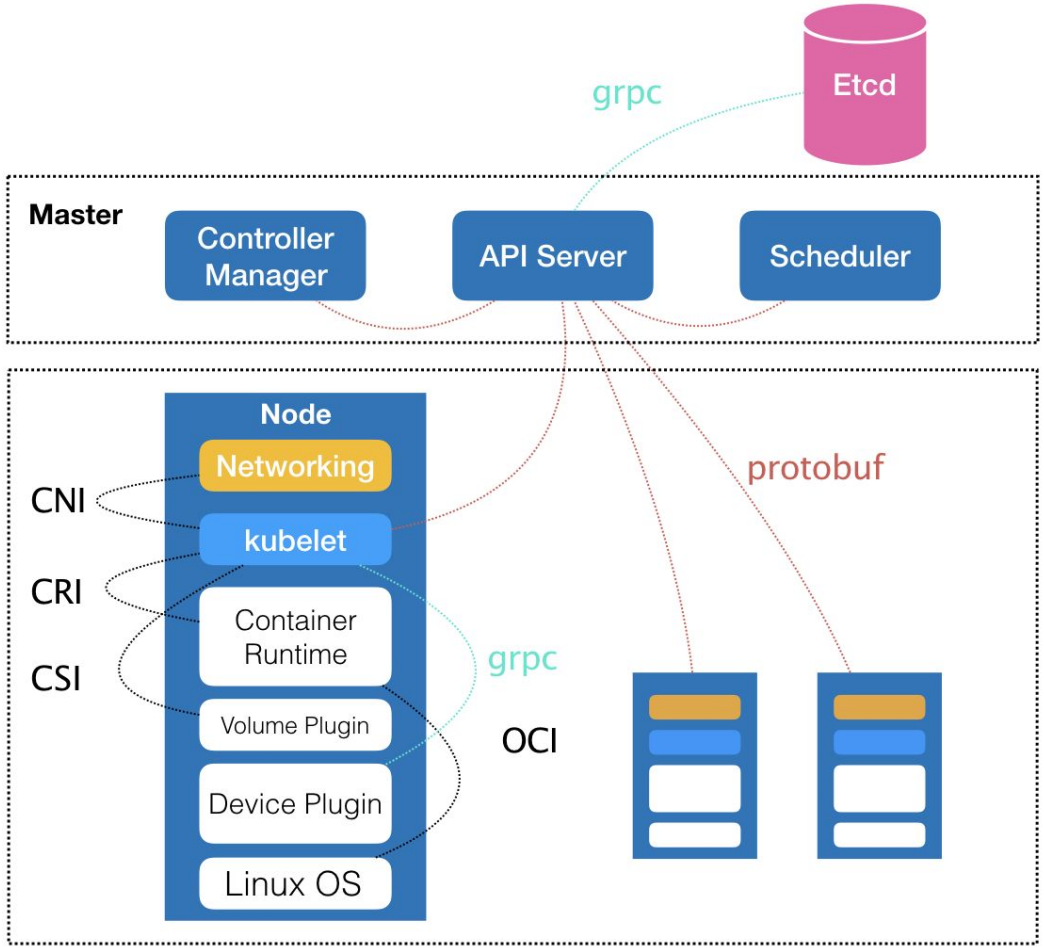
- 持續保持服務健康
- 處理服務跟服務之間的依賴性，例如資料庫要先啟動後網頁服務才能夠啟動
- 服務的資源使用量、網路、儲存管理等等
- 以上的功能都是因為使用者的需求而產生的

Docker-compose vs Docker Swarm vs Kubernetes 三者關係？

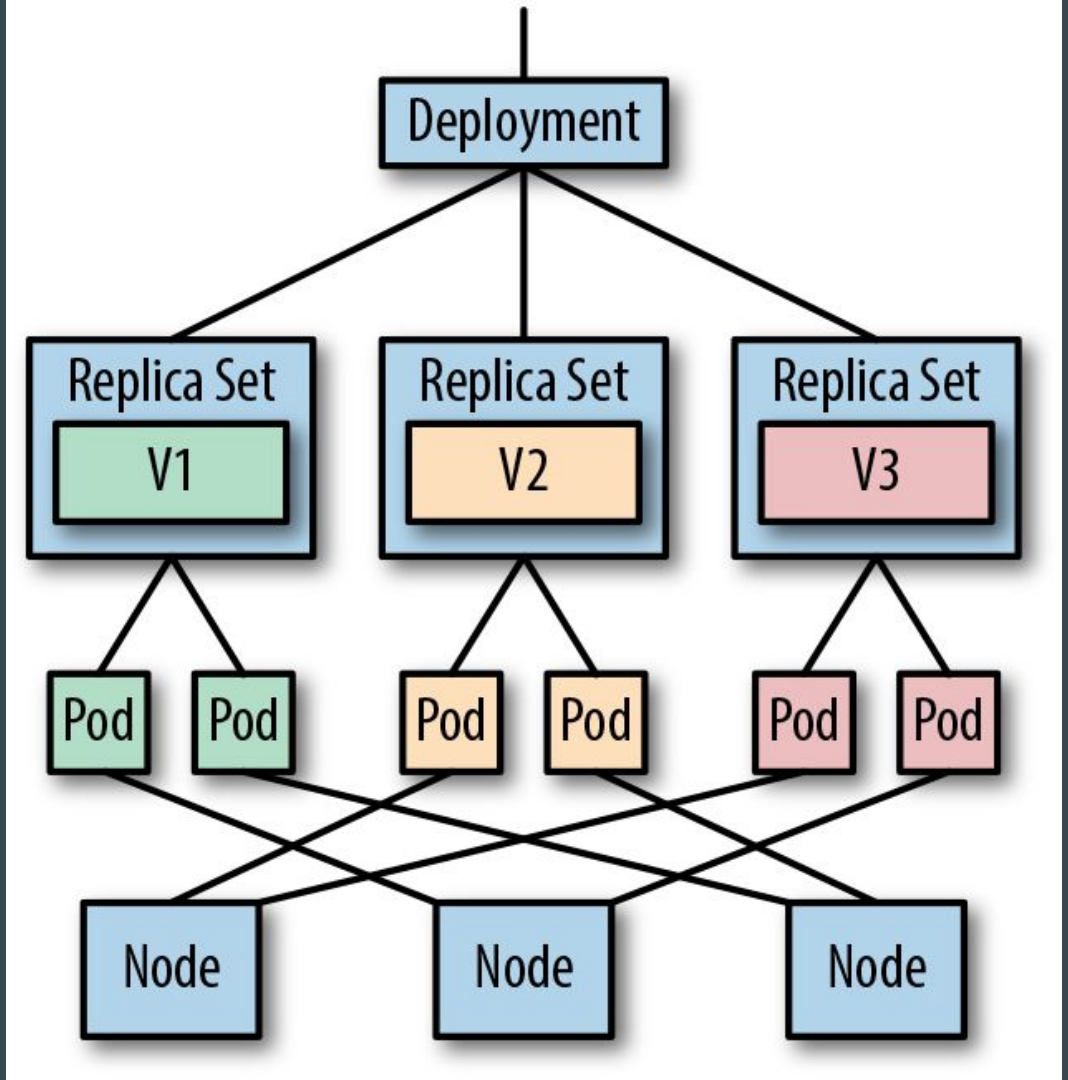
- Docker-compose 是 Docker 公司開發的商業產品，只限於在一台機器裡把多個 container 跑起來，Docker Swarm & Kubernetes 可以在多台機器裡。
- Docker Swarm 跟 Kubernetes 的終極目標是一致的，最大的差異在於對於社群的擁抱程度，Kubernetes 比 Docker Swarm 更能夠滿足客製化的需求，吸引很多廠商開發 Kubernetes 相關的服務。

Docker Swarm vs Kubernetes 最大的差異是什麼？

- ~~Docker Swarm 還有人用嗎，先替你默哀 3 秒鐘。~~
- Docker Swarm 在建立任何資源時都是 imperative（指令式），而 Kubernetes 則是 declarative（聲明式），Docker Swarm 下指令失敗就失敗了，而且會覆蓋先前的設定；Kubernetes 會不停地嘗試達成使用者的需求，兩份設定不會直接覆蓋，而是會合併。
- Kubernetes 有 interface 的設計，可以依照自己的喜好選擇 container、網路或儲存要什麼工具。


Etcd
grpc
Master
Controller Manager
API Server
Scheduler
Node
Networking
kubelet
Container Runtime
Volume Plugin
Device Plugin
Linux OS
CNI
CRI
CSI
grpc
OCI
protobuf

- Kubernetes 有把常見 orchestration 抽象化成 object（物件），例如適合網頁的 Deployment、適合資料庫的 StatefulSets 或適合短暫任務的 CronJob 。
- 如果常見的 orchestration object 還不能滿足特殊需求，使用者可以靠 Operator 框架寫自己想要的 CustomResourceDefinition object，自己定義 orchestration 方式，例如：在 AI 領域裡在算圖時怎麼有效率的分配顯示卡。


Deployment
Replica Set
V1
Replica Set
V2
Replica Set
V3
Pod
Pod
Pod
Pod
Pod
Pod
Node
Node
Node

Kubernetes 的生態

Kubernetes 社群的輪廓長怎樣？

- Kubernetes 為 Cloud Native Computing Foundation（CNCF）裡其中的一個專案，該基金會是由 Google & RedHat 共同創立，一開始是為了恆抗 Docker 公司在 container 主宰的地位，CNCF 所有專案。
- Kubernetes 的社群非常龐大，甚至有專門的專案處理社群事宜，有很多完整自動化和文件來管理來自世界各地的程式碼貢獻，像是 GitHub 機器人專案 Prow
- Kubernetes 不同的組件都有專門的專案，例如 API、指令、生命週期等等

CNCF Cloud Native Landscape
2020-01-21T04:19:06Z 284fcca

Overwhelmed? Please see the CNCF Trail Map. That and the interactive landscape are at l.cncf.io

Greyed logos are not open source

App Definition and Development

Database

CNCF Graduated: Vitess
CNCF Incubating: KV

Streaming & Messaging

CNCF Incubating: cloudevents, NATS

Application Definition & Image Build

CNCF Incubating: HELM

Continuous Integration & Delivery

Orchestration & Management

Scheduling & Orchestration

CNCF Graduated: kubernetes

Coordination & Service Discovery

CNCF Graduated: CoreDNS
CNCF Incubating: etcd

Remote Procedure Call

CNCF Incubating: gRPC

Service Proxy

CNCF Graduated: envoy

API Gateway

Service Mesh

CNCF Incubating: LINKERD

Runtime

Cloud Native Storage

CNCF Incubating: ROOK

Container Runtime

CNCF Graduated: containerd
CNCF Incubating: cri-o

Cloud Native Network

CNCF Incubating: CNI

Provisioning

Automation & Configuration

Container Registry

CNCF Incubating: HARBOR

Security & Compliance

CNCF Graduated: Falco
CNCF Incubating: notary, TUF, Open Policy Agent

Key Management

Platform

Certified Kubernetes - Distribution

Certified Kubernetes - Hosted

Certified Kubernetes - Installer

PaaS/Container Service

Observability and Analysis

Monitoring

CNCF Graduated: Prometheus

Logging

CNCF Graduated: fluentd

Tracing

CNCF Graduated: Jaeger
CNCF Incubating: OPENTRACING

Chaos Engineering

Serverless

Kubernetes Certified Service Provider

Kubernetes Training Partner

Special

CLOUD NATIVE Landscape
CLOUD NATIVE COMPUTING FOUNDATION
Redpoint Amplify

This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path.

l.cncf.io

Members

該怎麼看 Kubernetes 社群生態？

- 使用者
  - 在 Kubernetes 有龐大的社群資源和教學，不會設定都不會孤單
  - 開心地選擇想要的解決方案，實體機和在公有雲上的解決方案百百種
- 進階玩家
  - Kubernetes 生態背後還是有廠商在支持，商業與開源是相輔相成
  - 可以開始考慮貢獻一下社群

- 開發者
  - CNCF 裡所有的成員不會干預 CNCF 裡專案的發展，不論什麼公司或甚至 CNCF 裡的技術監督委員會都不會去決定 Kubernetes 專案的走向，都是交給 Kubernetes 自身的社群決定。
  - 在 Kubernetes 社群裡總是會有比較會社交或者技術很強的人，兩者的角色在社群裡都很重要，像是我在社群裡就是技術不強只會社交的。

總結

- Container 本質是 process 跟 VM 差異很大
- Kubernetes 奉行使用者至上原則
- 更宏觀且前瞻的 orchestration 設計理念
- 活躍的社群

更多的問題

~~其實就是怕講不完被我刪掉的內容啦，啊哈哈哈哈 XDDDD~~

- Dockerfile 是怎麼實踐的呢？
- 託管的 Kubernetes 是什麼？
- 託管的 Kubernetes 這麼多，常見的有 GKE、EKS 和 AKS 等等，我該怎麼選？

参考資料

- gopherize
- Docker Container: My first learning
- Containerization and how it differs from Virtual Machines
- 6 Alternatives to Docker
- Docker runtime metric
- Docker-Internals
- iptables

- 關於指揮的10個大哉問:有請指揮家瓦格
- Working with Kubernetes Objects
- Kubernetes Explained in 100 Seconds
- cncf-annual-report-2019