

# Docker Swarm 起手式 + Portainer

Daniel

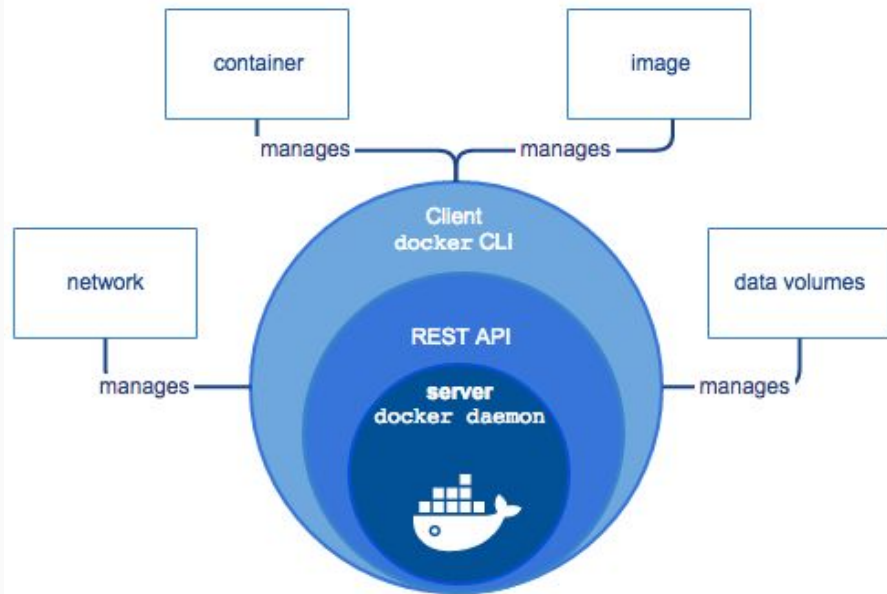
<https://ppt.cc/fgMWJx>

# Agenda

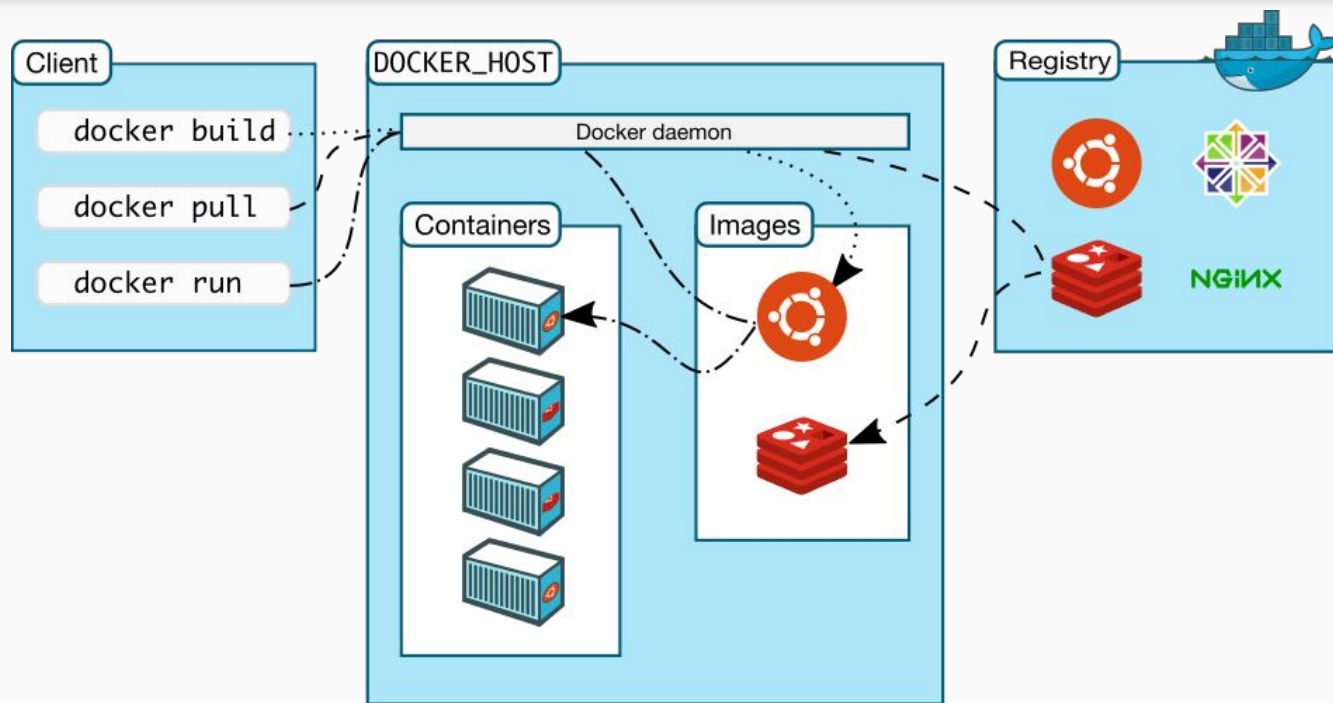
1. 前情提要
2. 什麼是 Docker Swarm
3. 為什麼我選擇 Docker Swarm ?
4. Docker Swarm 簡介與實作
5. 什麼是 Portainer
6. Portainer 簡介與實作

# 前情提要 - 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**



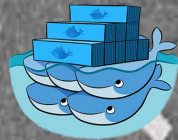
# 前情提要 - Docker Architecture



# 什麼是 Docker Swarm

Docker Swarm 是 Docker 官方自己發行的一個核心項目，主要是用來提供 Docker 容器集群的服務，也可以將多個 Docker Host 主機串成一個容器雲平台的集群。





**FREE**

**PAID**



# 為什麼我選擇 Docker Swarm ?

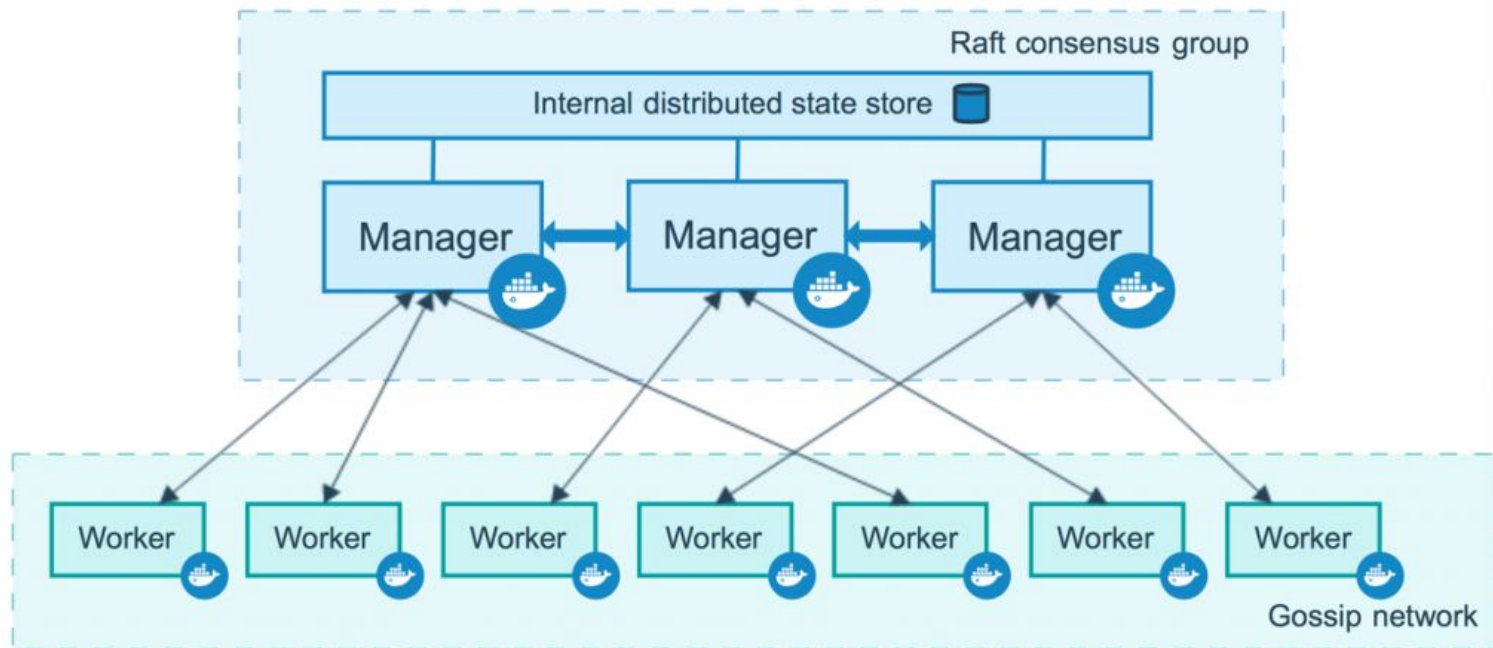
- Docker 官方出品, 原生支持。
- 簡單、上手容易、架構較簡單。
- 擴展容易。
- 用於專案方便。

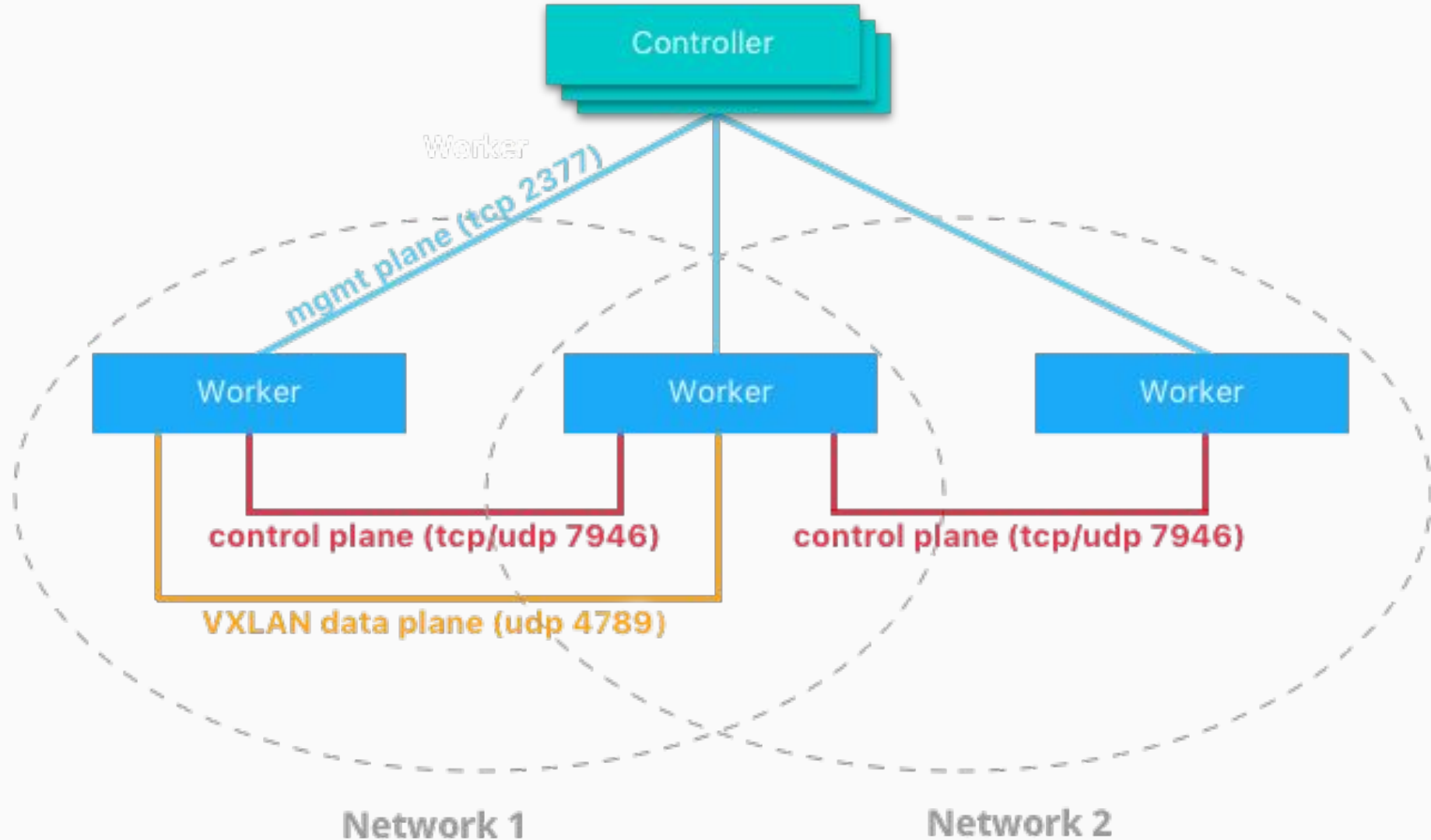
# Docker Swarm 簡介 - 特性

- 內建 Docker Engine ( Docker 1.12+ )
- 分散式設計
- 聲明性服務模型
- 協調期望狀態與實際狀態之間的差異
- 多主機網路
- 服務發現、負載平衡、安全
- 滾動升級



# Docker Swarm 簡介 - Architecture





# Docker Swarm 簡介 - 容錯

Swarm Size	Majority	Fault Tolerance
1	1	0
2	2	0
<b>3</b>	2	<b>1</b>
4	3	1
<b>5</b>	3	<b>2</b>
6	4	2
<b>7</b>	4	<b>3</b>
8	5	3
<b>9</b>	5	<b>4</b>

# Docker Swarm Hands-On Lab

# Docker Swarm 初始化

- 初始化 Docker Swarm

```
$ docker swarm init --advertise-addr ${IP}
```

- 加入 Docker Swarm ( 管理群 | 工作群 )

```
$ docker swarm join-token ( manager | worker )
```

- 只顯示 Docker Swarm ( 管理群 | 工作群 ) Token

```
$ docker swarm join-token -q ( manager | worker )
```

- 離開 Docker Swarm 集群

```
$ docker swarm leave -f
```

# 了解 Docker Swarm Node 的狀態

- 列出 node

```
$ docker swarm node ls
```

ID	HOSTNAME	STATUS	AVAILABILITY	MANAGER STATUS	ENGINE VERSION
qe7nzq03hd3vg5vlh6z68qqnr	node1	Ready	Active	Leader	18.06.1-ce
o1iksz1362cgvaklifg1ral13 *	node2	Ready	Active	Reachable	18.06.1-ce
ura30kdpij81e4hkhn4awh5mg	node3	Ready	Active	Reachable	18.06.1-ce
kd4monej7ai25jeuloidaoivx	node4	Ready	Active		18.06.1-ce
vu7k2t8td50kvs1xvk76qzyr7	node5	Ready	Active		18.06.1-ce

## Availability:

- Active: 可以分派新工作到任何節點。
- Pause: 暫停分派新工作給節點，但不影響現有服務。
- Drain: 不會分派任何新工作給節點，會把現有的服務移至到可使用的節點上。

## Manager Status:

- 沒有值表示不是 Docker Swarm manage node。
- Leader: 主要的管理節點，並且為整個 Docker Swarm 集群做決策。
- Reachable: 當主要的管理節點掛點，有機會成為主要的管理節點。
- Unavailable: 表示不可以用了。

## 操作 docker node 相關指令

- 更改管理節點可用性

```
$ docker node update --availability ( active | pause | drain ) ${node_name}
```

- 工作節點變更為管理節點

```
$ docker node promote ${node_name}
```

- 管理節點變更為工作節點

```
$ docker node demote ${node_name}
```

- 刪除工作節點

```
$ docker node rm -f ${node_name}
```

- 給節點新增或刪除標籤

```
$ docker node update --label-add <key> or <key>=<value>
```

```
$ docker node update --label-rm <key> or <key>=<value>
```

Q: Docker Swarm 管理節點壞了一個，該如何修復，並重新加回來到管理集群中？



# scenario



## 操作 Docker Service 相關指令

- 建立容器服務

```
$ docker service create --name web nginx
```

- 查看容器服務

```
$ docker service ls
```

- 查看容器服務中的任務

```
$ docker service ps ${SERVICE_NAME}
```

- 更新容器服務的埠號

```
$ docker service update --publish-add local_port:container_port ${SERVICE_NAME}
```

- 更新容器服務的副本數

```
$ docker service update --replicas number ${SERVICE_NAME}
```

- 刪除容器服務

```
$ docker service rm ${SERVICE_NAME}
```

# 什麼是 Portainer ?

## MANAGE YOUR DOCKER ENVIRONMENTS WITH EASE

PORTAINER IS AN OPEN-SOURCE LIGHTWEIGHT MANAGEMENT UI WHICH ALLOWS YOU TO  
EASILY MANAGE YOUR DOCKER HOSTS OR SWARM CLUSTERS

AVAILABLE ON  LINUX,  WINDOWS &  OSX

Q: 請使用 docker service create  
來建立 portainer



# scenario

```
$ docker network create --driver overlay --attachable  
portainer_agent_network
```

- --driver ( bridge | overlay ) 決定要用哪一個 driver 來建立網路, 預設是 bridge
- --attachable 啟用手動容器附件

## 使用 docker service create 建立 Portainer - Step 2

```
$ docker service create \  
    --name portainer_agent \  
    --network portainer_agent_network \  
    -e AGENT_CLUSTER_ADDR=tasks.portainer_agent \  
    --mode global \  
    --constraint 'node.platform.os == linux' \  
    --mount type=bind,src=/var/run/docker.sock,dst=/var/run/docker.sock \  
    --mount type=bind,src=/var/lib/docker/volumes,dst=/var/lib/docker/volumes \  
    portainer/agent
```

--name 給容器服務一個名字

--network 使用網路附件

-e 設定環境變數

--mode ( global | replicated ) 容器服務的模式, 預設是使用 replicated

--constraint 限制要在執行的節點

--mount 掛載檔案系統到容器服務裡

## 使用 docker service create 建立 Portainer - Step 3

```
$ docker service create \
  --name portainer \
  --publish 9000:9000 \
  --replicas=1 \
  --constraint 'node.role == manager' \
  --mount type=bind,src=/root/portainer_data/,dst=/data \
  portainer/portainer
```

- constraint

node attribute	matches	example
node.id	Node ID	node.id==2ivku8v2gvtg4
node.hostname	Node hostname	node.hostname!=node-2
node.role	Node role	node.role==manager
node.labels	user defined node labels	node.labels.security==high
engine.labels	Docker Engine's labels	engine.labels.operatingsystem==ubuntu 14.04

## 使用 docker service create 建立 Portainer - Step 4

```
$ docker network ls
```

```
$ docker service ls
```

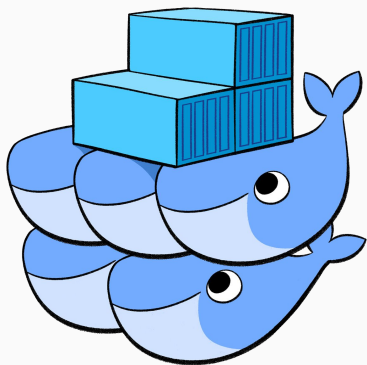
```
$ docker service update --network-add portainer_agent_network portainer
```

### DIFFERENCES BETWEEN "--MOUNT" AND "--VOLUME"

The `--mount` flag supports most options that are supported by the `-v` or `--volume` flag for `docker run`, with some important exceptions:

- The `--mount` flag allows you to specify a volume driver and volume driver options *per volume*, without creating the volumes in advance. In contrast, `docker run` allows you to specify a single volume driver which is shared by all volumes, using the `--volume-driver` flag.
- The `--mount` flag allows you to specify custom metadata ("labels") for a volume, before the volume is created.
- When you use `--mount` with `type=bind`, the host-path must refer to an *existing* path on the host. The path will not be created for you and the service will fail with an error if the path does not exist.
- The `--mount` flag does not allow you to relabel a volume with `z` or `Z` flags, which are used for `selinux` labeling.

Q: 請使用 docker stack 來部署  
portainer



# scenario



## 使用 docker stack 部署 portainer

```
$ curl -L https://portainer.io/download/portainer-agent-stack.yml -o  
portainer-agent-stack.yml
```

```
$ docker stack deploy --compose-file=portainer-agent-stack.yml portainer
```

# 參考資料鏈結

- [docker overview](#)
- [Manage nodes in a swarm](#)
- [docker service create](#)
- [change node availability](#)
- [add or remove label metadata](#)
- [change a node role](#)
- [Docker Swarm strategies](#)
- [docker network understanding](#)

# Image credits

- [DCOS LOGO](#)
- [K8S LOGO](#)
- [Docker Swarm LOGO](#)
- [Docker Swarm Architecture](#)
- [docker logo](#)
- [rocket](#)
- [portainer logo](#)

<https://www.surveycake.com/s/ooPrr>

Q & A