

docker進階實務班

Philipz(鄭淳尹)

2017-08-19

廣宣學堂

github.com/philipz/advanced_docker_workshop

人工 智慧

AI

最熱門的AI演算技術與實作演練！

Coursera、Udacity 專業認證講師！



授課教師

江義淵

萬能科技大學 助理教授

現職：萬能科技大學 航空暨工程學院 助理教授

經歷：元智大學資訊工程所 博士

專長：人工智慧、訊號分析、腦機介面、數位多媒體



授課教師

徐旺興

萬能科技大學 助理教授

現職：萬能科技大學 航空暨工程學院 助理教授

經歷：國立中央大學電機系 博士

專長：無線通訊網路，嵌入式系統，行動計算，遠距教學，資料庫

授課教師

鄭淳尹(Philipz)

臺北榮民總醫院
資訊工程師



現職：臺北榮民總醫院資訊工程師

經歷：Docker.Taipei 社群發起人、TradingBot 程式交易機器人創作者，系

統維護及開發設計超過15年。現任微軟MVP，翻譯審閱多本容器技術書籍。

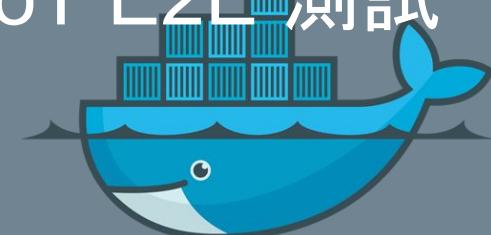
專長：Docker技術，人工智慧

跨域數位人才
加速躍升計畫



Today Topics

1. Docker Machine 介紹與基本指令
2. Docker Machine 建立雲端虛擬機
3. Docker Swarm 介紹與基本指令
4. 以 Machine 建置 Swarm 叢集
5. Docker Swarm networking & Swarm service
6. GitLab 結合 Docker 容器開發測試
7. 容器與 Raspberry Pi IoT 整合應用
8. 以 GitLab 執行 Docker Compose IoT E2E 測試
9. Moby & LinuxKit 介紹與入門



Docker Ecosystem



docker-machine

provision



install

become
ready

Compose (Swarm v1)



Compose + Docker stacks
(Swarm Mode)



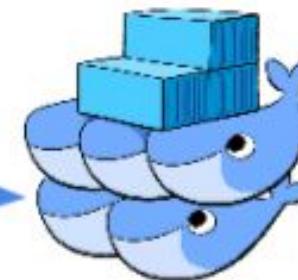
Docker CLI



join cluster



control



Microsoft Azure

<https://portal.azure.com/>

curl -sSL <https://get.docker.com/> | sh

Microsoft Azure 新增 > 計算

新增 計算

搜尋 Marketplace

MARKPLACE 查看全部

計算 網路 Storage Web + 行動 Databases Intelligence + analytics 物聯網 Enterprise Integration 安全性 + 識別 Developer tools Monitoring + management

精選應用程式 查看全部

- Windows Server 2012 R2 Datacenter
Enterprise-class solutions that are simple to deploy, cost-effective,
- Windows Server 2016 Datacenter
Enterprise-class solutions that are simple to deploy, cost-effective,
- Red Hat Enterprise Linux 7.2
Red Hat Enterprise Linux 7 is the world's leading enterprise Linux platform built to meet the needs
- Ubuntu Server 16.04 LTS
Ubuntu Server delivers the best value scale-out performance available.



Review Docker Compose

[GitHub: compose_wp_proxy](#)

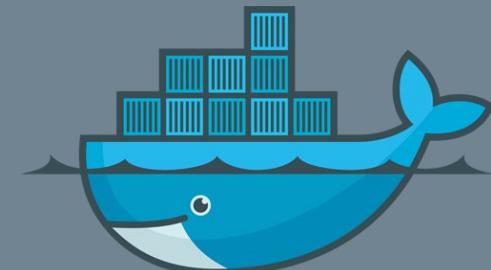
WordPress example of previous week

Add new service - Nginx Reverse Proxy

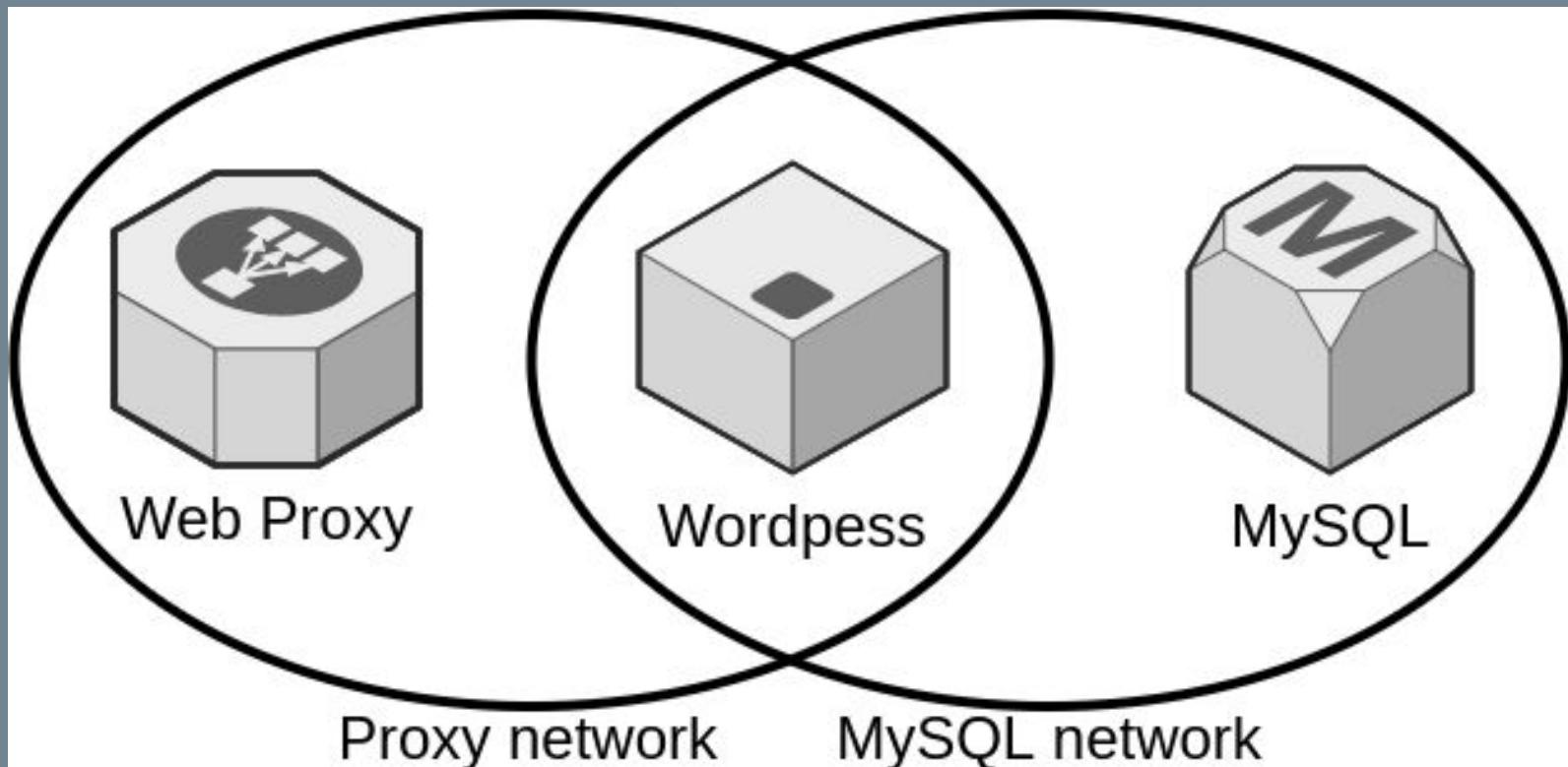
docker-compose scale wordpress=2

DNS-based service discovery

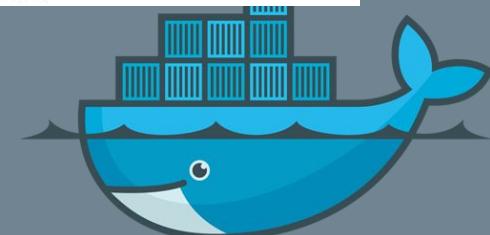
\$nslookup wordpress



Compose & Wordpress

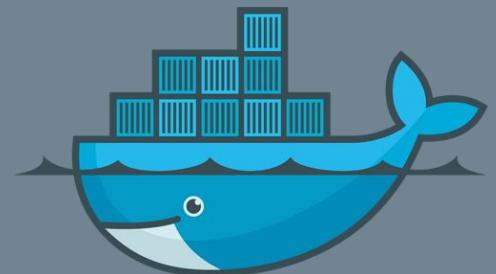


- 水平擴展 wordpress : scale



1.1 Docker Machine

介紹



Docker Machine

- Combind AWS CLI, Azure CLI, VMware CLI.....
- Learn One, Run Everywhere
- VMware vSphere

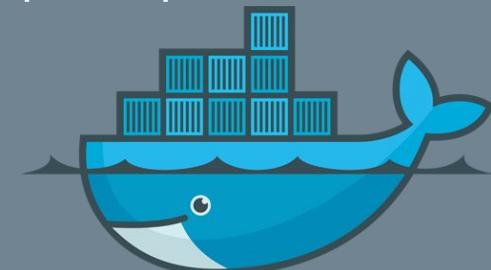
- a. Install govc

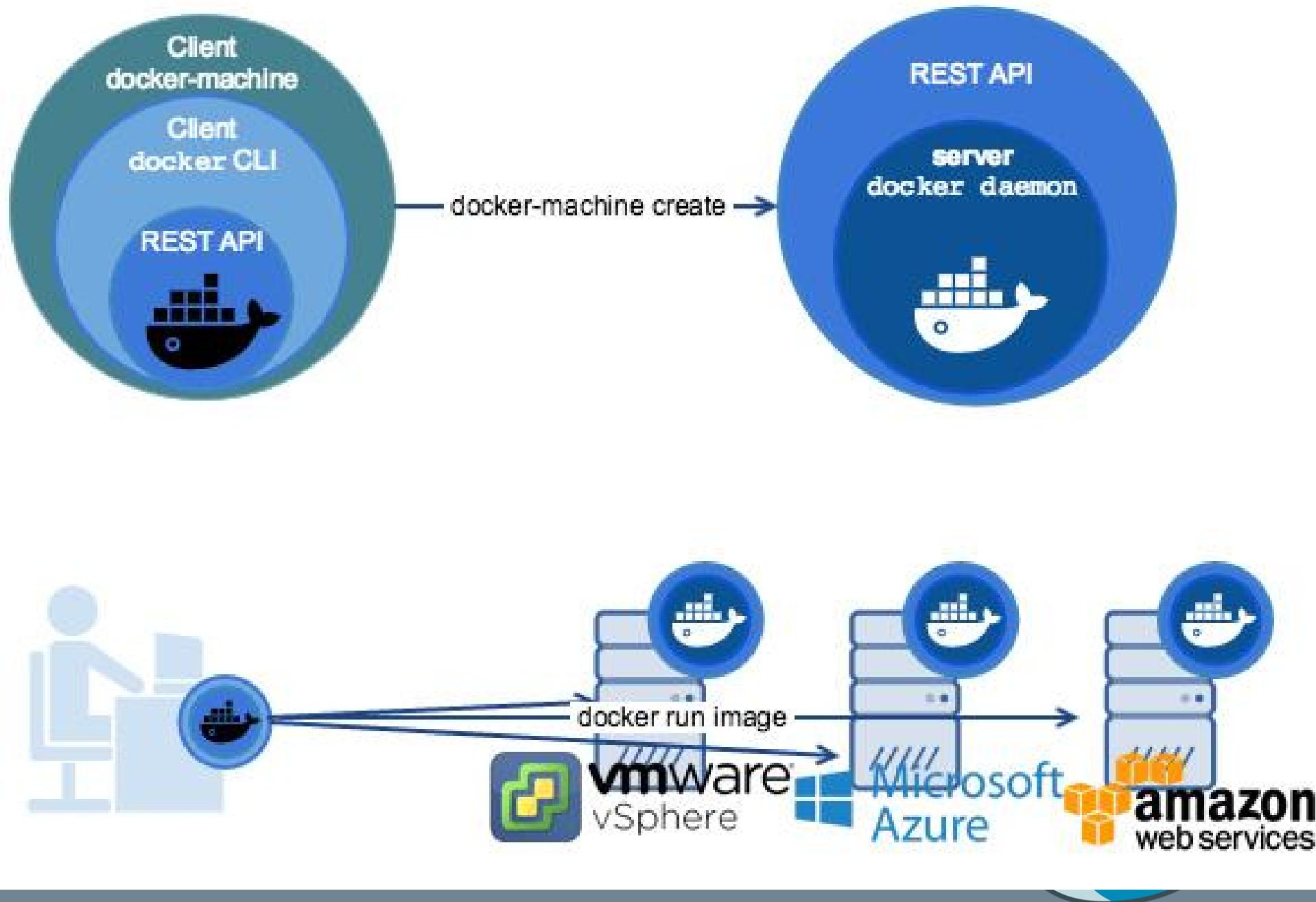
go get github.com/vmware/govmomi/govc

go install github.com/vmware/govmomi/govc

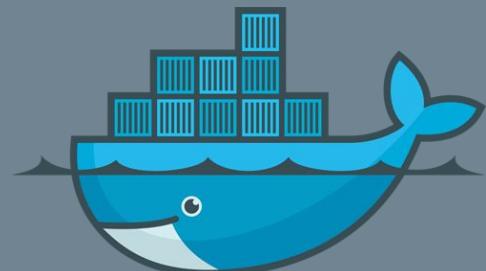
- b. docker-machine create vmdocker --driver vmwarevsphere
--vmwarevsphere-datacenter DCNAME --vmwarevsphere-vcenter
ESX_IP --vmwarevsphere-username root --vmwarevsphere-password
PASSWORD --vmwarevsphere-datastore DSNAME
--vmwarevsphere-network VMNETWORK

- Azure
- AWS
- VirtualBox





1.2 Docker Machine 基本指令



Install Docker Machine

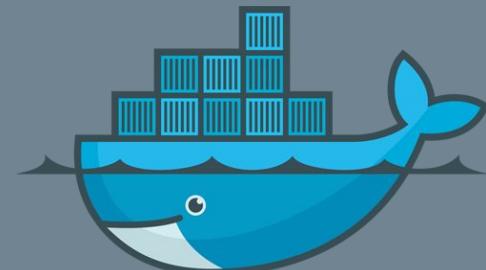
sudo curl -L

```
"https://github.com/docker/machine/releases/download/v0.12.  
2/docker-machine-$(uname -s)-$(uname -m)" -o  
/usr/local/bin/docker-machine
```

and

```
sudo chmod +x /usr/local/bin/docker-machine
```

docker-machine -v



Docker Machine commands (1/2)

Commands:

active	Print which machine is active
config	Print the connection config for machine
create	Create a machine
env	Display the commands to set up the environment for the Docker client
inspect	Inspect information about a machine
ip	Get the IP address of a machine
kill	Kill a machine
ls	List machines
provision	Re-provision existing machines
regenerate-certs	Regenerate TLS Certificates for a machine
restart	Restart a machine



Docker Machine commands (2/2)

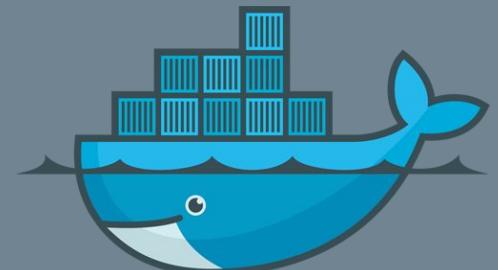
Commands:

<code>rm</code>	Remove a machine
<code>ssh</code>	Log into or run a command on a machine with SSH.
<code>scp</code>	Copy files between machines
<code>start</code>	Start a machine
<code>status</code>	Get the status of a machine
<code>stop</code>	Stop a machine
<code>upgrade</code>	Upgrade a machine to the latest version of Docker
<code>url</code>	Get the URL of a machine
<code>version</code>	Show the Docker Machine version or a machine docker version
<code>help</code>	Shows a list of commands or help for one command



2. Docker Machine

建立雲端虛擬機



Azure VM

- Azure CLI
- 使用 Docker 電腦搭配 Azure 驅動程式
- 使用 Azure CLI 選取 Linux VM 映像

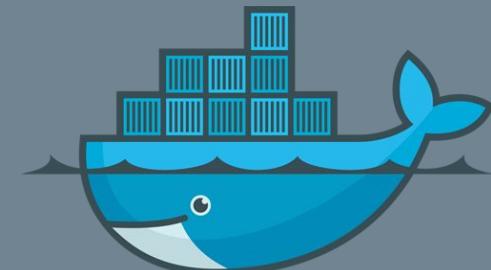
```
$ docker run -it azuresdk/azure-cli-python:0.2.8  
az login, then enter the code
```

```
az vm image list --output table
```

```
az vm image list-skus -l eastasia -p canonical - f ubuntuserver
```

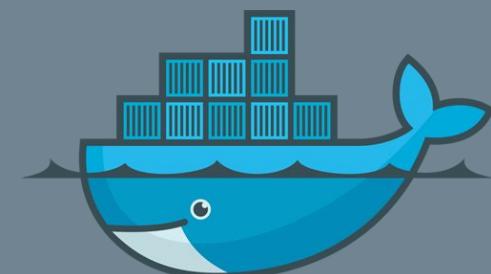
[azure vm docker create](#)

```
az vm list-sizes -l eastasia | more
```



Machine Create

- Azure VM Size
- docker-machine create -d azure
--azure-subscription-id="XXXXXX"
--azure-location="southeastasia" --azure-image
canonical:ubuntuserver:16.04.0-LTS:16.04.201611150
--azure-size Standard_D1_v2 --engine-install-url
<https://get.docker.com> **docker-0-0-1**
- VM size list
- VM size pricing



Where is subscription-id ?



+ 新增



資源群組



應用程式服務



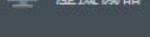
SQL 資料庫



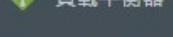
SQL 資料倉儲



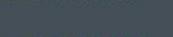
NoSQL (DocumentDB)



虛擬機器



負載平衡器



儲存體帳戶



虛擬網路



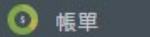
Azure Active Directory



監視



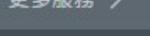
Azure 建議程式



資訊安全中心



帳單



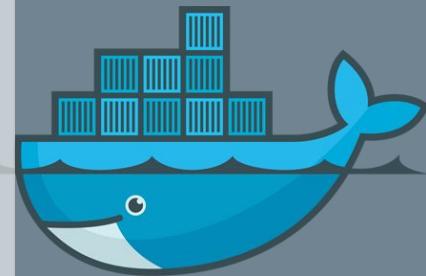
說明 + 支援

更多服務 >

請同時按下 Shift 鍵與空格鍵，以切換我的最愛

篩選
所有資源

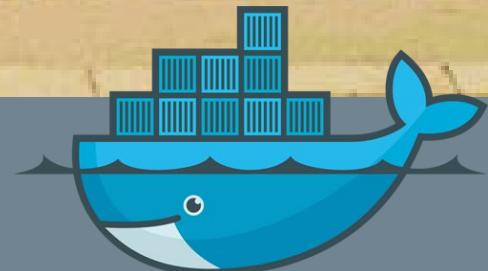
類別	內容	最愛	
一般	資源群組	★	
	所有資源	★	
	訂用帳戶	★	
	帳單	預覽 ★	
	說明 + 支援	★	
	計算	虛擬機器	★
		虛擬機器 (傳統)	★
		虛擬機器級別集合	★
		容器服務	★
		Batch 帳戶	★
Service Fabric 繪集		★	
雲端服務 (傳統)		★	
RemoteApp 集合		★	
可用性設定組		★	
OS 磁碟 (傳統)		★	
VM 映像 (傳統)	★		





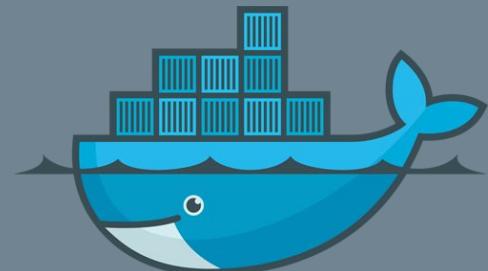
docker-machine ssh docker-0-0-1

`sudo usermod -aG docker $USER`



3.1 Docker Swarm

介紹



Container Orchestration

Docker Swarm

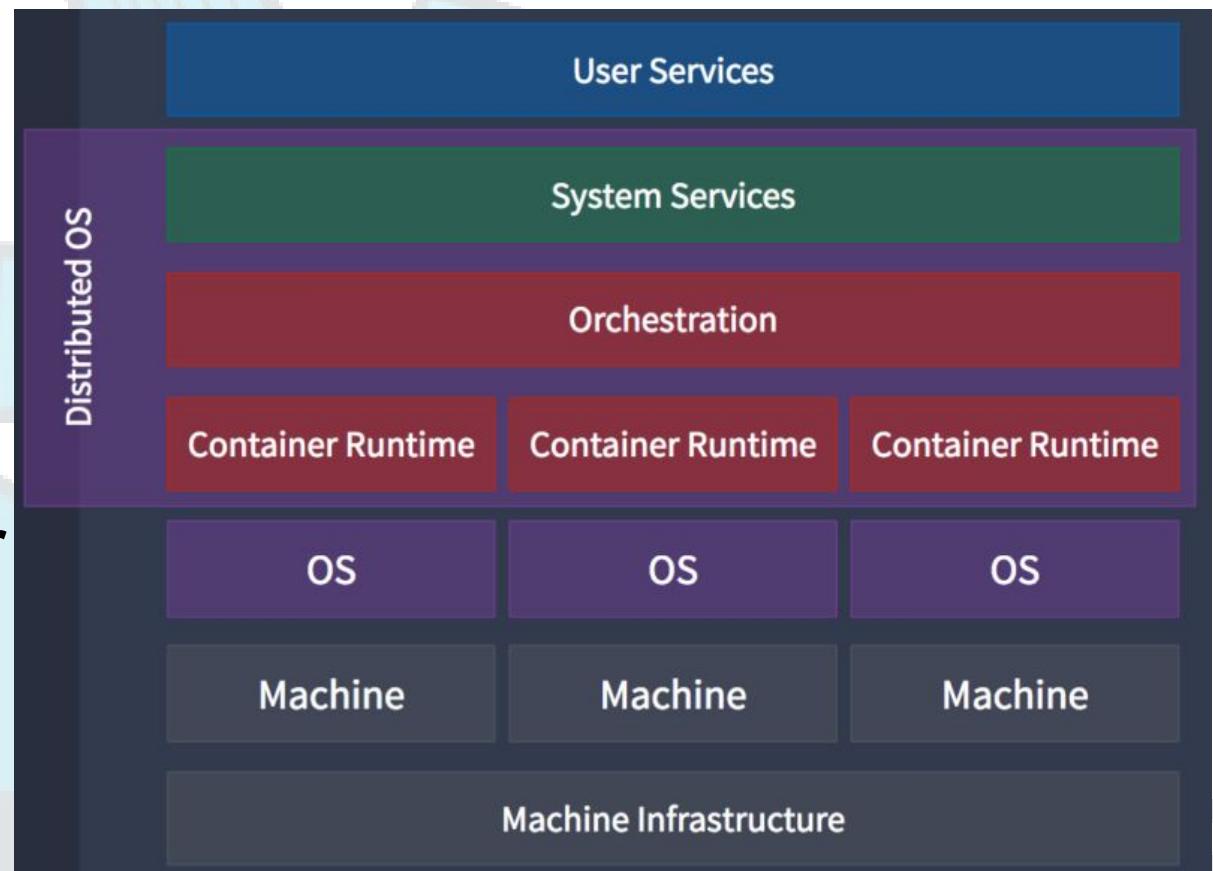
Kubernetes

DC/OS

Rancher

Docker Datacenter

???



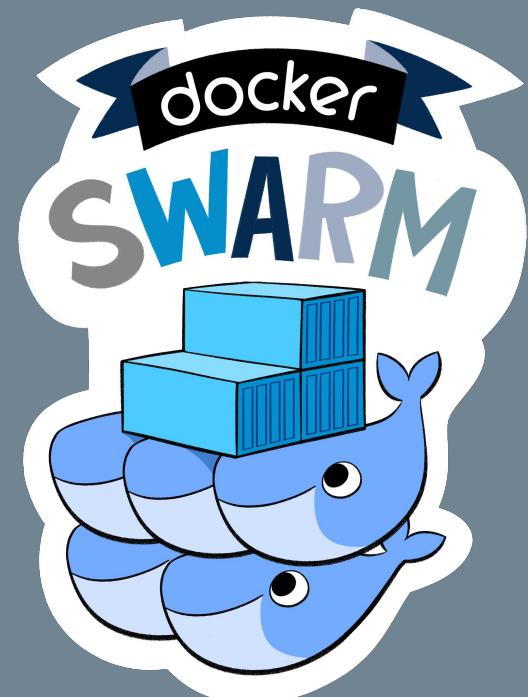
Docker Swarm

- Docker-native clustering system
- From v1.12 is default feature.
- Docker **overlay** network

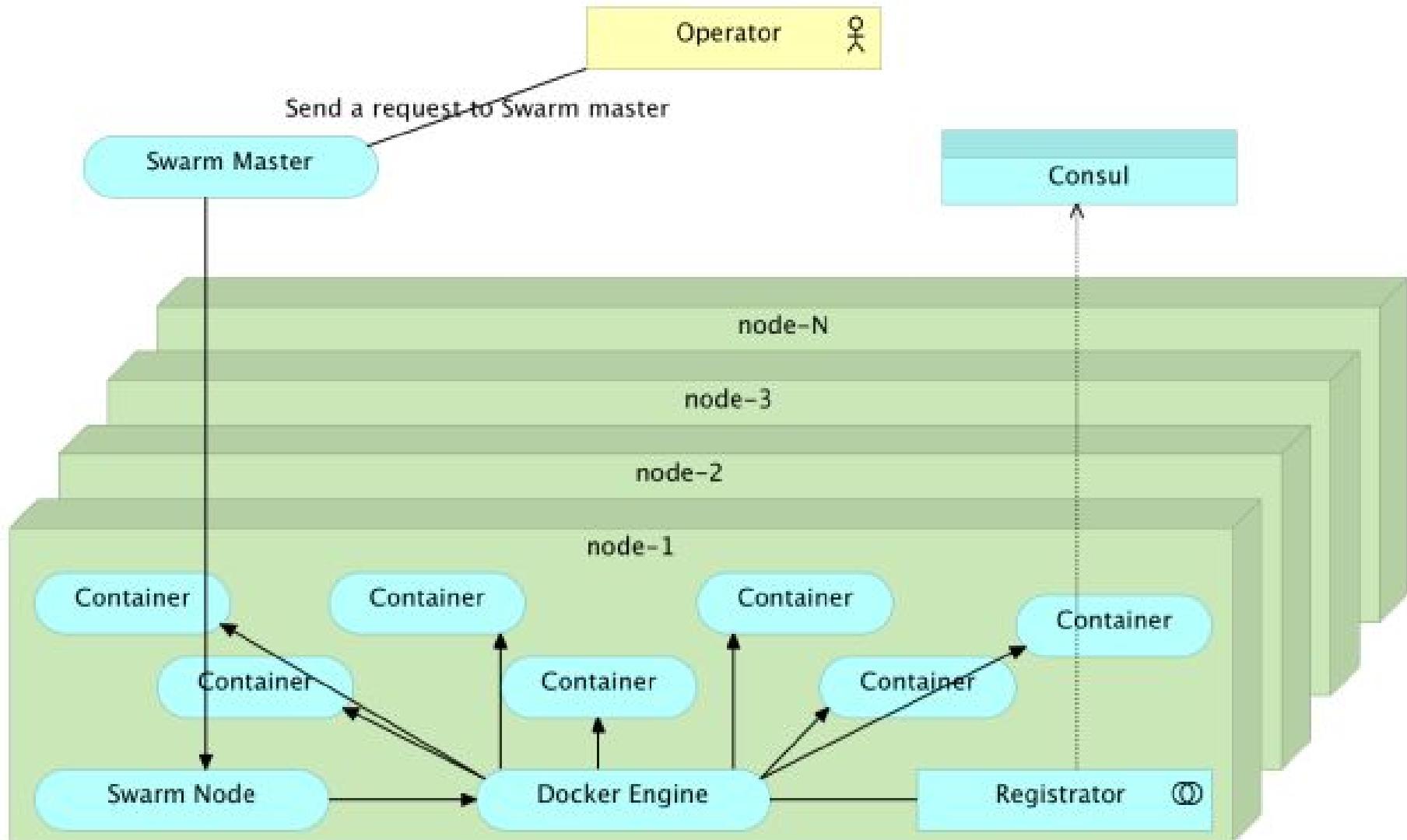
[Play-with-docker Online Lab](#)



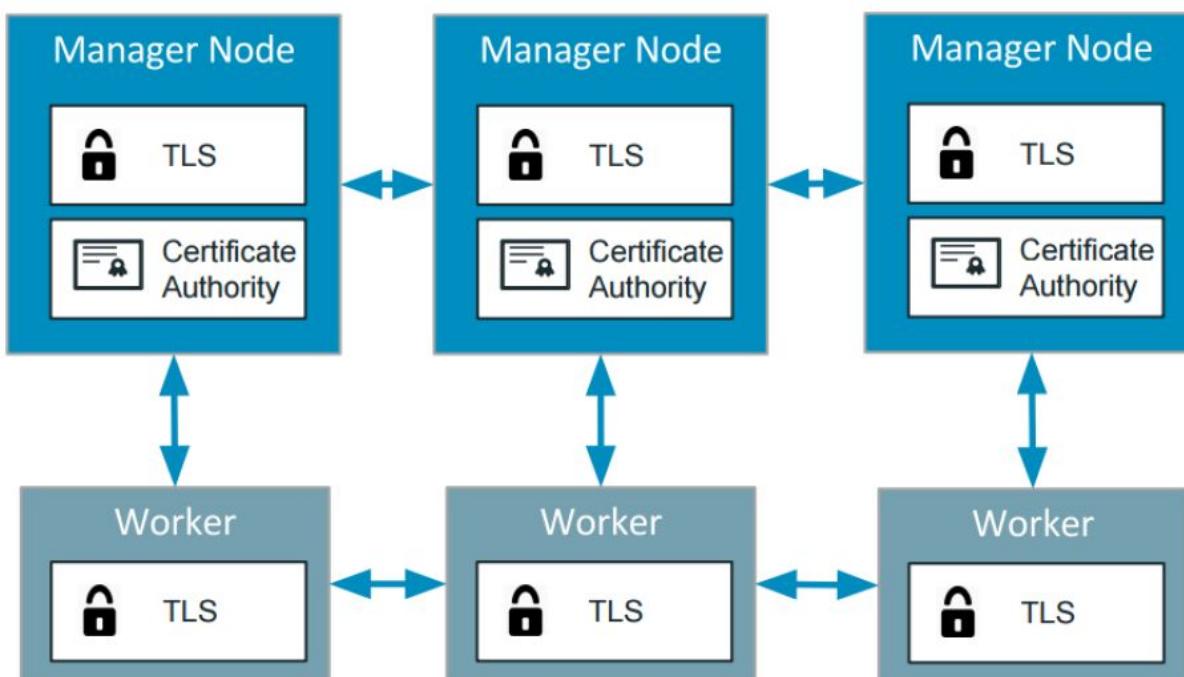
kubernetes



Old Swarm Architecture



New Swarm Mode (1/2)

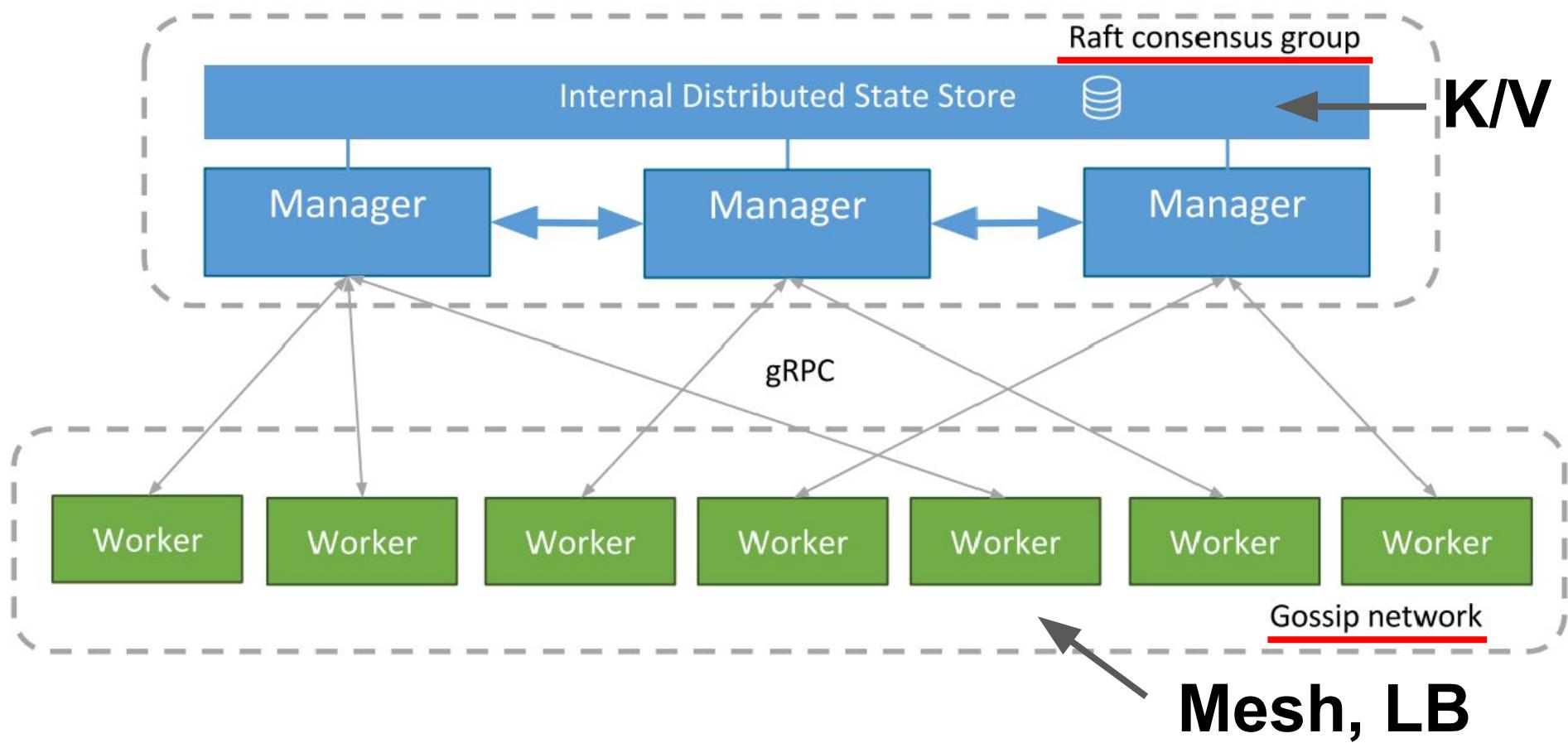


- Cryptographic node identity
- Automatic encryption and mutual auth (TLS)
- Automatic cert rotation
- External CA integration

Byzantine
Generals
Problem

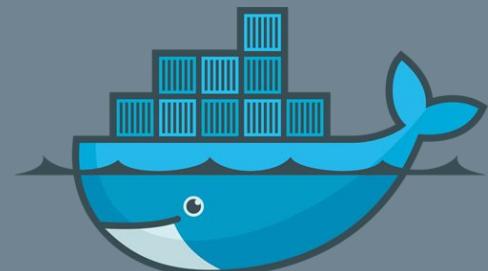
New Swarm Mode (2/2)

Consul, HashiCorp



3.2 Docker Swarm

基本指令



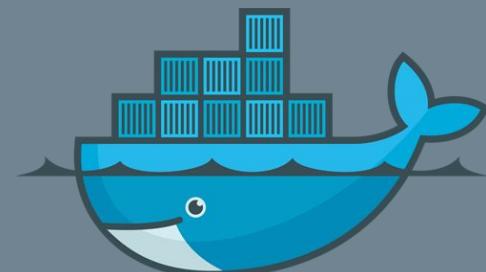
Docker Swarm commands

Commands:

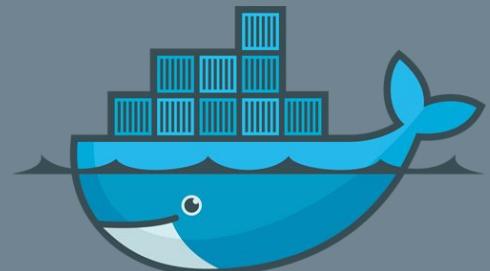
- init Initialize a swarm
- join Join a swarm as a node and/or manager
- join-token Manage join tokens
- update Update the swarm
- leave Leave the swarm (workers only)

Manager also can leave

\$ docker swarm -h

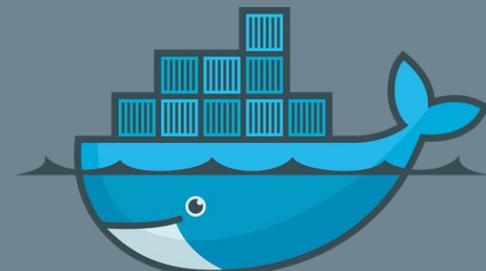


4. 以 Machine 建置 Swarm 叢集



Machine Create Again

- docker-machine create -d azure
--azure-subscription-id="XXXXXX"
--azure-location="southeastasia" --azure-image
canonical:ubuntuserver:16.04.0-LTS:16.04.201611150
--azure-size Standard_D1_v2 --engine-install-url
<https://get.docker.com> docker-0-0-2
- docker-machine create again



Create Swarm Cluster

Check version: \$ docker -v

\$ docker info

\$ docker swarm init

docker swarm join \

--token SWMTKN-1-44ze8j7xkq5t \

192.168.0.4:2377

\$ docker-machine ssh docker-0-0-2

COPY & PASTE

\$ docker-machine ssh docker-0-0-3

[docker swarm join docs](#)



It's so EASY!!!

docker-machine-vnet 虛擬網路

刪除

搜尋 (Ctrl+ /)

概觀 活動記錄 存取控制 (IAM) 標記

設定 位址空間 已連線的裝置 子網路 DNS 伺服器

程式集 ^

資源群組 docker-machine
位置 東亞
訂用帳戶名稱 Free Trial
訂用帳戶 ID 38ac9a31-2f7f-4ada-9483-6d284a16e7...

位址空間 192.168.0.0/16，及另外 2 個
DNS 伺服器 Azure 提供的 DNS 服務

2 連線的裝置

裝置	類型	IP 位址	子網路
docker-0-0-1-nic	網路介面	192.168.0.4	docker-machine
docker-0-0-2-nic	網路介面	192.168.0.5	docker-machine

Finding Firewall



Microsoft Azure 所有資源

所有資源
預設目錄

+ 新增 資料行 重新整理

訂用帳戶: Free Trial

firewall

名稱

docker-0-0-1-firewall

docker-0-0-2-firewall

A Microsoft Azure resource list interface. The search bar at the top contains the text "firewall", which is highlighted with a red rectangle. Below the search bar, there is a section titled "Name" containing two entries: "docker-0-0-1-firewall" and "docker-0-0-2-firewall". The interface includes standard Azure navigation elements like a sidebar with icons for SQL, Docker, and other services, and a header with account information and resource filtering options.

Communication Ports

Docker Remote API: 2376 Swarm Listen Port: 2377

Container network discovery: 7946 TCP/UDP

Container overlay network: 4789 UDP

優先順序	名稱	來源	目的地	服務	動作	
100	SSHAllowAny	任何	任何	SSH (TCP/22)	Allow	...
300	DockerAllowAny	任何	任何	自訂 (TCP/2376-2377)	Allow	...
310	DockerNode	任何	任何	自訂 (任何/7946)	Allow	...
320	overlay	任何	任何	自訂 (任何/4789)	Allow	...
330	web	任何	任何	自訂 (任何/80)	Allow	...



Check Swarm Cluster

```
$ docker info
```

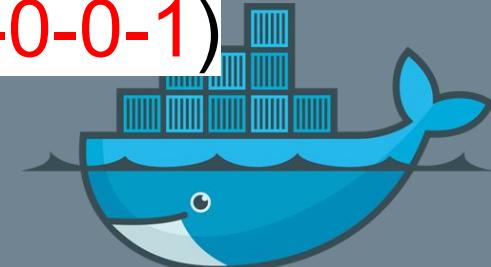
Managers: 1

Nodes: 2

```
$ docker node ls
```

ID	HOSTNAME	STATUS	AVAILABILITY	MANAGER
STATUS				
29zkgygdq6el0ylwtov5xksy2	docker-0-0-2	Ready	Active	
bbf3b27xkybups1foh750qf15	*	docker-0-0-1	Ready	Active Leader

```
$ eval $(docker-machine env docker-0-0-1)
```



Docker node commands

Commands:

demote	Demote one or more nodes from manager in the swarm
inspect	Display detailed information on one or more nodes
ls	List nodes in the swarm
promote	Promote one or more nodes to manager in the swarm
rm	Remove one or more nodes from the swarm
ps	List tasks running on a node
update	Update a node

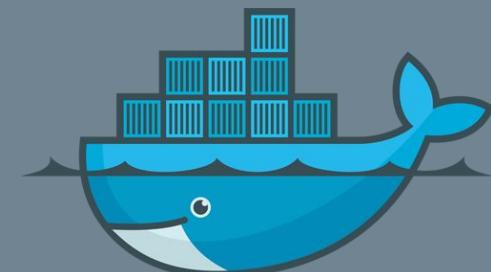
```
$ docker node ls
```

```
$ docker promote docker-0-0-2
```

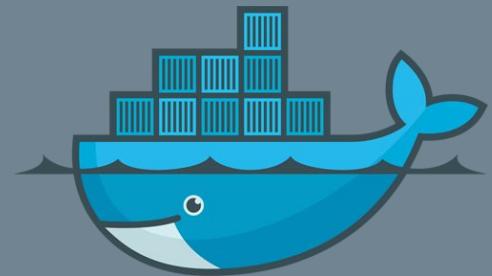
```
$ docker node ls
```

```
$ docker demote docker-0-0-2
```

```
$ docker node ls
```



5.1 Docker Swarm Networking

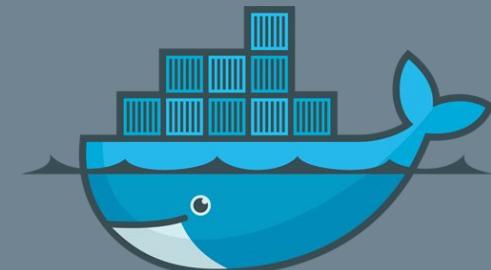


Docker Built-In Network Drivers

- Bridge
 - Overlay
 - MACVLAN
 - Host
 - None
- Docker Plug-In Network Drivers
- weave
 - calico
- Docker Plug-In IPAM Drivers
- infoblox

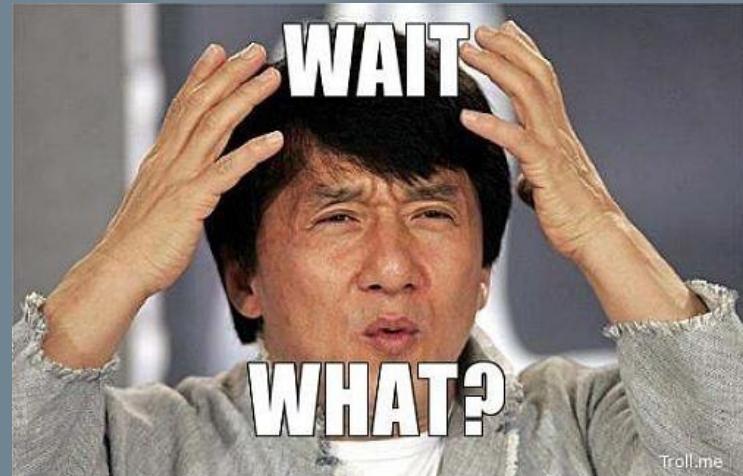
No more “link”, just use network.

Docker Reference Architecture: Designing Scalable,
Portable Docker Container Networks

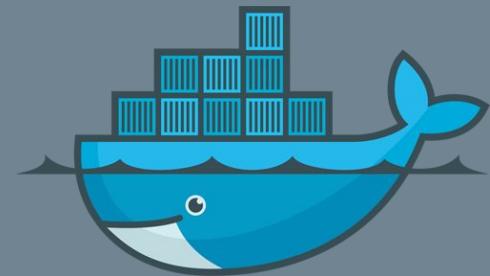


Exercise

```
$ docker network ls  
$ docker network create --driver overlay my-network  
$ docker network inspect my-network  
$ docker service create \  
  --replicas 3 \  
  --name my-web \  
  --network my-network \  
  nginx:alpine  
$ docker service ps my-web  
$ docker network inspect my-network  
$ docker ps  
$ docker exec -ti XXXXX sh  
$ nslookup my-web          $nslookup tasks.my-web
```



5.2 Swarm service



Docker service commands

Commands:

create Create a new service

inspect Display detailed information on one or more services

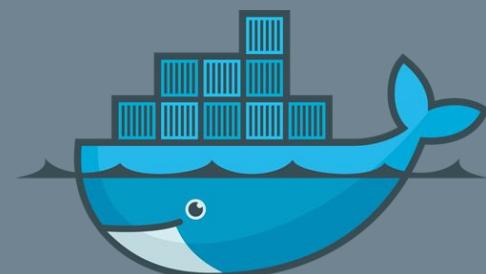
ps List the tasks of a service

ls List services

rm Remove one or more services

scale Scale one or multiple services

update Update a service



Service Create Exercise

```
$ docker network create --driver overlay wp_db
$ docker network inspect wp_db
$ docker service create \
--name db --network=wp_db \
-e MYSQL_ROOT_PASSWORD=wordpress \
-e MYSQL_DATABASE=wordpress \
-e MYSQL_USER=wordpress \
-e MYSQL_PASSWORD=wordpress \
mysql:5.7

$ docker service create \
--name wp -p 80:80 --network=wp_db \
-e WORDPRESS_DB_HOST=db:3306 \
-e WORDPRESS_DB_PASSWORD=wordpress \
wordpress:4.5
```



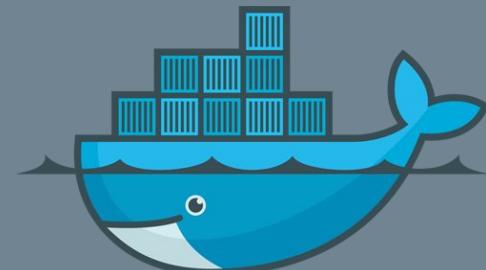
Service Rolling updates

```
$ docker service scale wp=3
```

```
$ docker service update \  
--image wordpress:4.6 \  
--update-delay 10s \  
--update-parallelism 1 \  
wp
```

```
$ docker service ps wp
```

[docker service update docs](#)



Docker stack

```
$ docker network create --driver overlay mysql
```

```
$ docker network create --driver overlay proxy
```

```
$ docker stack deploy
```

```
--compose-file=wordpress.yml wordpress
```

```
$ docker stack ps wordpress
```

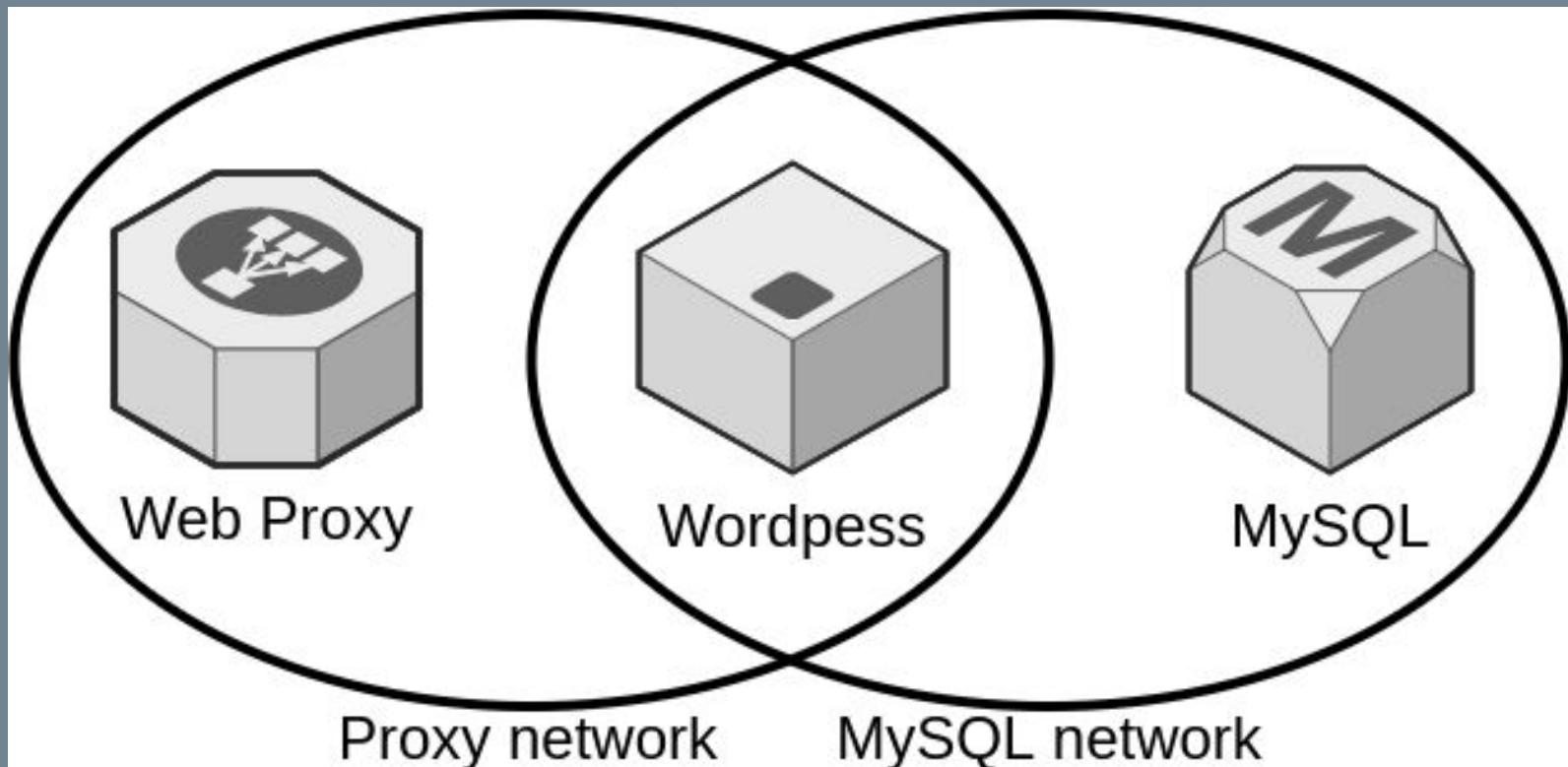
```
$ docker service ls
```

Remove external: true & philipz/reproxy

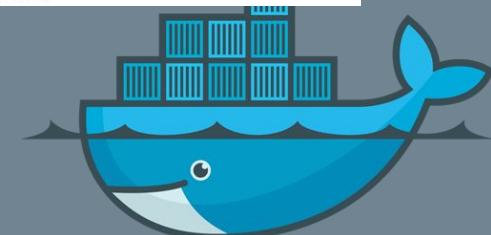
Then try again & Scale down!



Compose & Wordpress



- 水平擴展 wordpress : scale

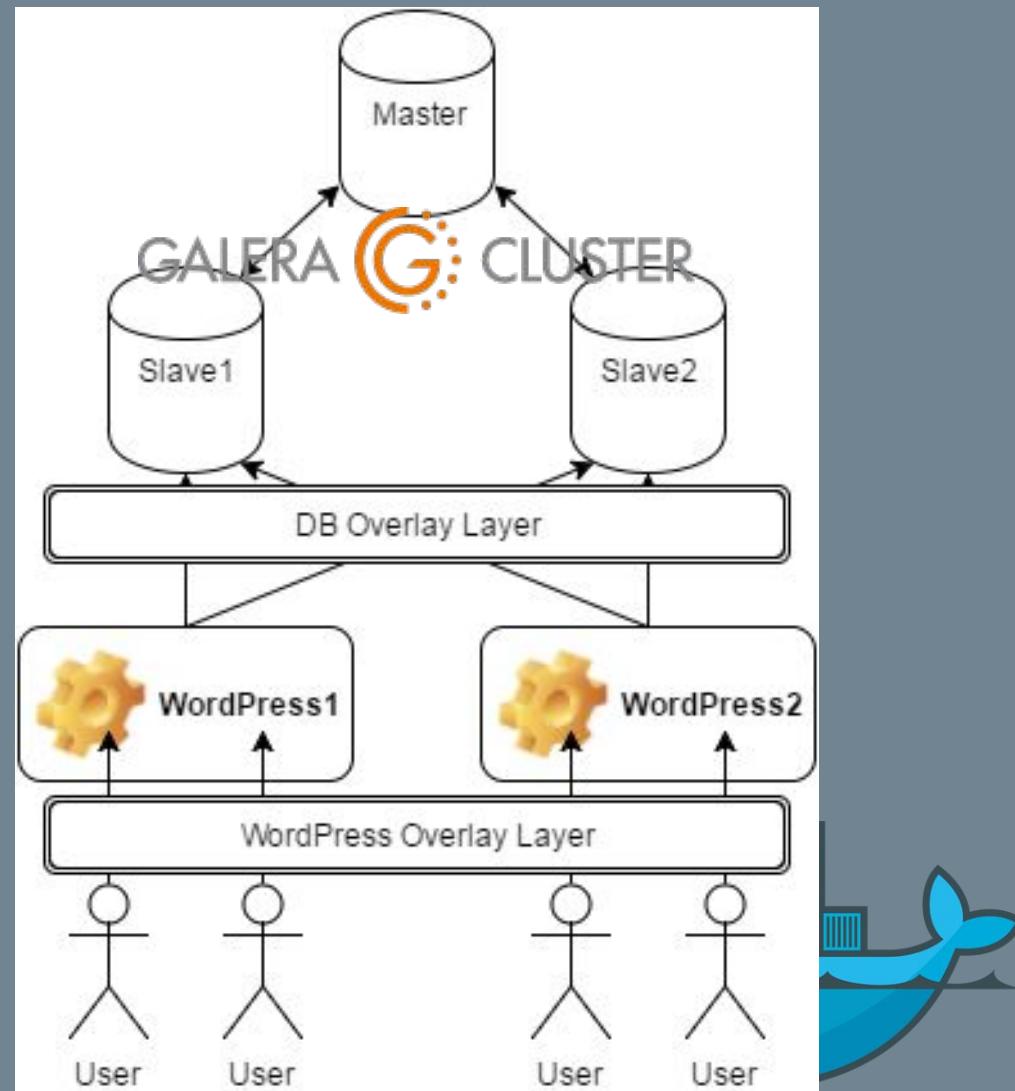


Swarm & MySQL Cluster

Docker Swarm for
MySQL Cluster &
WordPress

[Katacoda online lab.](#)

1. [Getting Started Galera with Docker, part 1](#)
2. [Getting Started Galera with Docker, part 2](#)



More
Advanced
Docker Workshop



Play Bigger!!!
ALL Docker Machines
Join Together!!!



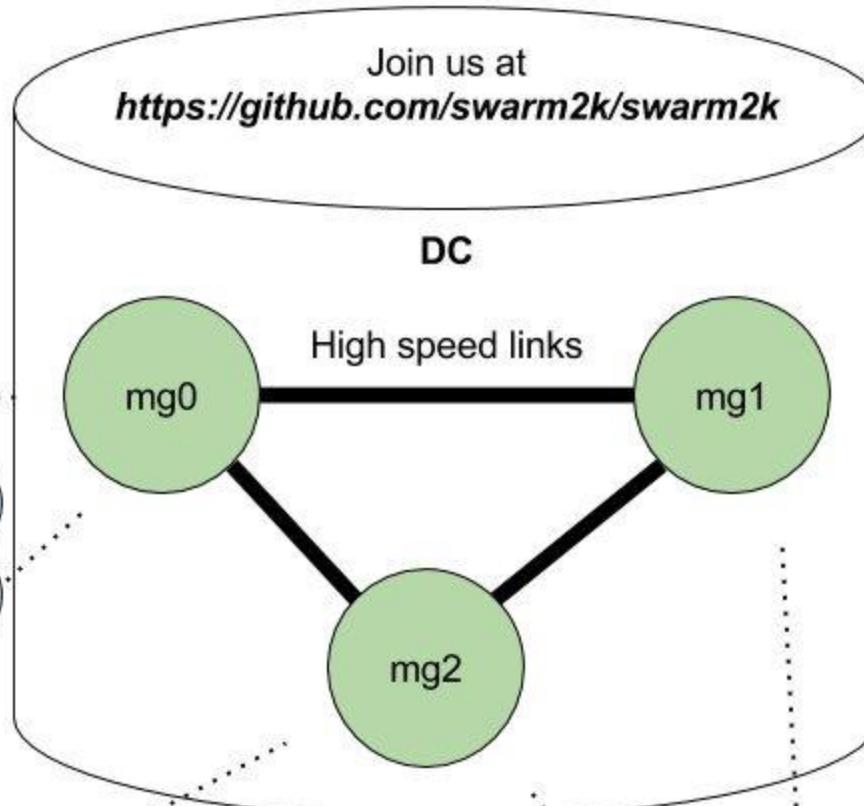
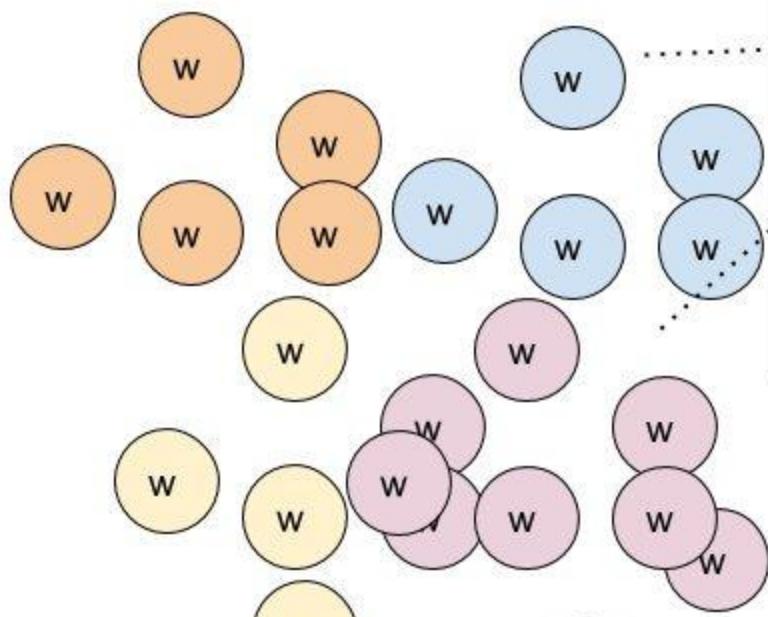
A Global-Scale Collaborative Experimental Project on the Docker Swarm mode

#DockerSwarm2000

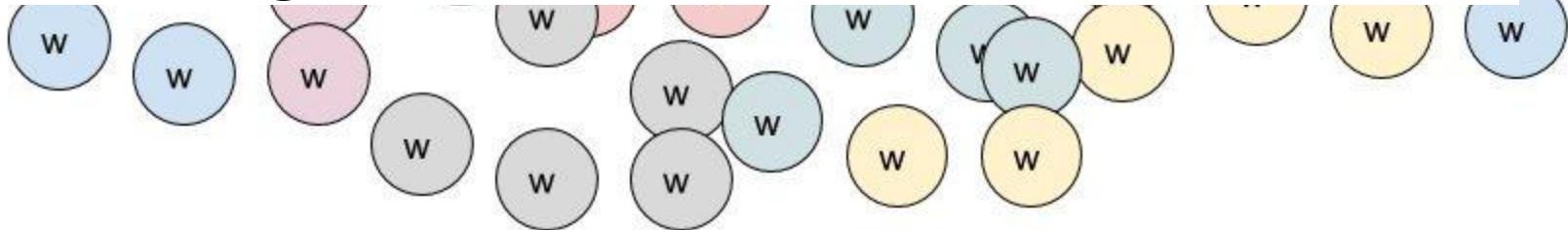
We need your help !!!

mg = Manager nodes

w = Worker nodes (**your nodes**)



<https://github.com/swarm2k/swarm2k>



Azure Container Instances



The screenshot shows the Microsoft Azure Container Instances landing page. The title is '容器執行個體' (Container Instances) with a '預覽' (Preview) badge. Below it, the sub-headline reads '使用單一命令輕鬆在 Azure 上執行容器' (Run containers in Azure with a single command). A bulleted list highlights the benefits:

- ✓ 幾秒鐘內開始使用容器，並利用依秒計費來控制您的基礎結構成本
- ✓ 不需要進行虛擬機器 (VM) 管理
- ✓ 不需要學習新的容器協調流程工具，只要您的容器應用程式在雲端中執行
- ✓ 依使用量付費 - 容器執行個體會依秒、vCPU、GB 記憶體計費
- ✓ 使用命令列參數指定核心計數、記憶體數量及其他容器屬性
- ✓ 輕鬆從 Docker Hub 或 Azure Container Registry 部署映像

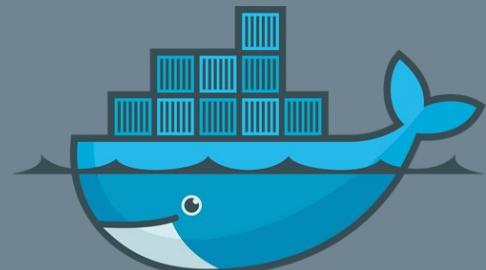
A large green button labeled '立即試用 >' (Try Now) is prominently displayed. At the bottom, there are links for '探索容器執行個體' (Explore Container Instances), '定價詳細資料' (Pricing details), and '文件' (Documentation).

立即開始使用容器

輕鬆啟動並執行，不需要學習新的協調流程概念或程式設計模型。輕鬆開始部署容器，以支援您的目標使用案例或其他應用程式開發和測試情境。使用 Docker 技術將您的應用程式容器化，按一下就可以立即執行。



6. GitLab 結合 Docker 容器開發測試



Azure Web App on Linux

Microsoft Azure > 新增 > Web + 行動

The screenshot shows the Azure portal interface. On the left, a sidebar lists various service categories like Dashboard, Resource Groups, and Storage. The main area is titled '新增' (Create) and shows the 'Marketplace' search bar. Under 'MARKETPLACE', there's a list of service categories: 計算, 網路, Storage, Web + 行動, Databases, Intelligence + analytics, 物聯網, Enterprise Integration, 安全性 + 識別, Developer tools, Monitoring + Management, Add-ons, and 容器. To the right, under 'Web + 行動', there are several options: 'Web 應用程式' (Web Application), 'Mobile App' (A scalable and secure backend that can be used to power apps on any platform - iOS, Android, ...), '邏輯應用程式' (Logic App) (無須撰寫程式碼，就能自動化各個雲端上的資料存取與使用流程), and 'Web App On Linux (預覽)' (Web App On Linux (Preview)) (Enjoy your web app natively hosted on Linux). The 'Web App On Linux (Preview)' option is circled in red.

新增

搜尋 Marketplace

MARKETPLACE

計算 >

網路 >

Storage >

Web + 行動 >

Databases >

Intelligence + analytics >

物聯網 >

Enterprise Integration >

安全性 + 識別 >

Developer tools >

Monitoring + Management >

Add-ons >

容器 >

Web + 行動

精選應用程式

查看全部

Web 應用程式

享受您的 Web 應用程式所提供之安全且具彈性的開發、部署及縮放選項。

Mobile App

A scalable and secure backend that can be used to power apps on any platform - iOS, Android, ...

邏輯應用程式

無須撰寫程式碼，就能自動化各個雲端上的資料存取與使用流程

Web App On Linux (預覽)

Enjoy your web app natively hosted on Linux.

CDN

Enjoy scalable, global distributed edge servers for fast and reliable content delivery

Use Docker image for Web AP

Web App On Linux (預覽) □ X

建立

* 應用程式名稱
philipz ✓ .azurewebsites.net

* 訂用帳戶
Developer Program Benefit

* 資源群組 i
● 新建 ● 使用現有項目

* App Service 方案/位置
ServicePlane1acc0a9-a710(Wes... >

* 設定容器
philipz/nginx >

釘選到儀表板

建立 **自動化選項**

Docker 容器

 Docker 容器

Linux 上的 Web Apps 可善用 Docker 容器的威力，讓您從 Azure Container Registry、Docker Hub、私人容器登錄使用自訂容器，或是使用 App Service 所提供的其中一項預設容器。

影像來源
內建 **Docker Hub** 私人登錄

存放庫存取
公用 私用

* 影像及選擇性標籤 (例如 'image:tag')
philipz/nginx

啟動檔案
[empty input field]

確定

Azure PaaS 價格選型

新增 App Service 方案

為 Web 應用程式建立方案

* App Service 方案

請輸入您 App Service 方案的名稱

* 位置

Southeast Asia

* 定價層

S1 標準

確定

選擇定價層

瀏覽所提供的方案及其功能

* Linux App Service 方案的預覽版將有 50% 的折扣。進一步了解 [\[\]](#)

S1 標準

1 核心

1.75 GB RAM

50 GB
儲存體

自訂網域 / SSL
包括 SNI 與 IP SSL 支援

最多 10 個執行個體
自動調整規模

每天
備份

5 個位置
Web 應用程式預備環境

預覽定價
下方為 50% 折扣的價格

2,308.45
TWD/月 (估計)

S2 標準

2 核心

3.5 GB RAM

50 GB
儲存體

自訂網域 / SSL
包括 SNI 與 IP SSL 支援

最多 10 個執行個體
自動調整規模

每天
備份

5 個位置
Web 應用程式預備環境

預覽定價
下方為 50% 折扣的價格

4,616.89
TWD/月 (估計)

S3 標準

4 核心

7 GB RAM

50 GB
儲存體

自訂網域 / SSL
包括 SNI 與 IP SSL 支援

最多 10 個執行個體
自動調整規模

每天
備份

5 個位置
Web 應用程式預備環境

預覽定價
下方為 50% 折扣的價格

9,233.78
TWD/月 (估計)

B1 基本

1 核心

B2 基本

2 核心

B3 基本

4 核心

選取

ChatBot

引言 競賽時程 參賽辦法 獎項 初選 決賽 注意事項 DevDays 決賽隊伍名單 立即報名

另外我們也設計了 AI + Office 365 組，希望可以激發開發者們的想像力，創造出更多商業面的應用，像是 ChatBot on Microsoft Teams, Slack, Skype for Business 等，只要您使用 Microsoft Bot Framework 建置 Chatbot 的同時開啟用 Microsoft Teams 選項即符合參賽資格，或是結合 Office Add-ins 加值服務，非常歡迎各方開發好手來小試身手，將現金大獎抱回家！

決賽隊伍名單 (依隊伍名稱筆畫排序，無關名次)

- AI + ChatBot 組 (含北市府市政 ChatBot 特別獎)：
Akachan、DTC、TradingBot 交易顧問、仙女與他的快樂夥伴隊、台鐵時靈、房價貓、洽吧智能、嘉里大榮、罰單申訴機器人、輔力社 / 扶老 2.0
- AI + Office 365 組：
AiBot、SOLO、WiAdvance OAOA、仙女與他的快樂夥伴隊、脫魯奶爸 (E

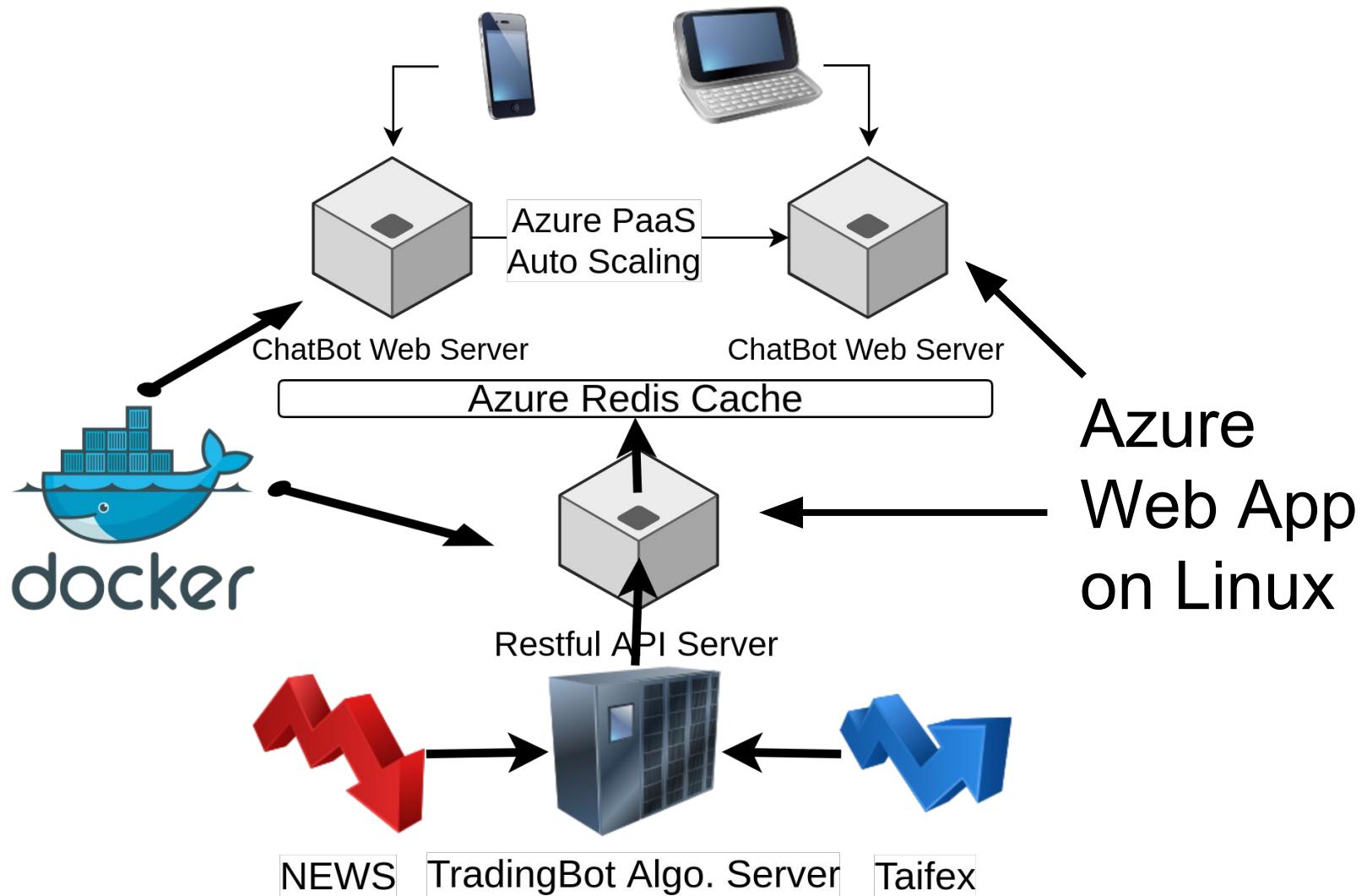
競賽時程

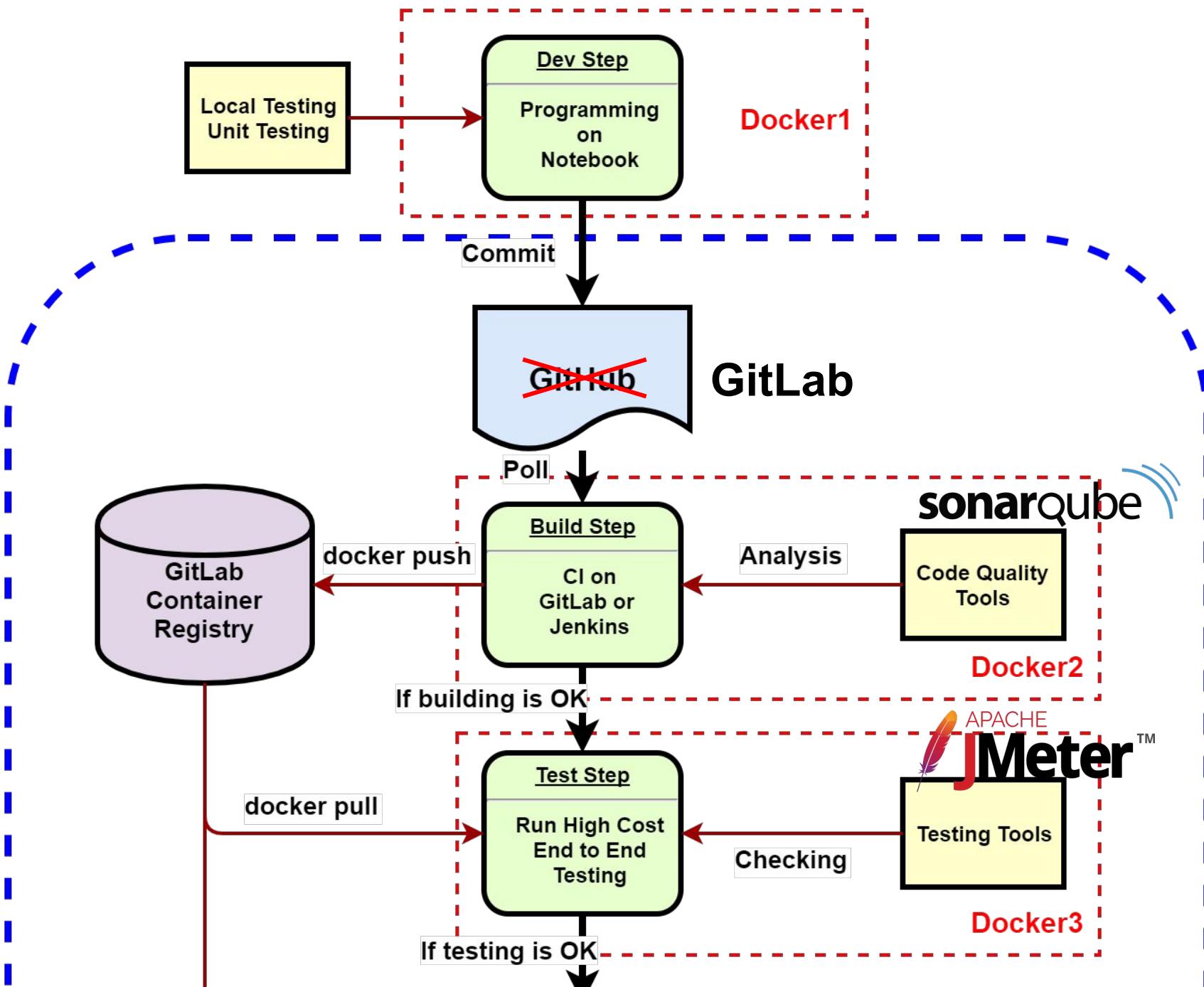
由於比賽文件非常踴躍，公布決賽入選團隊日期調整為 5/12 (五)

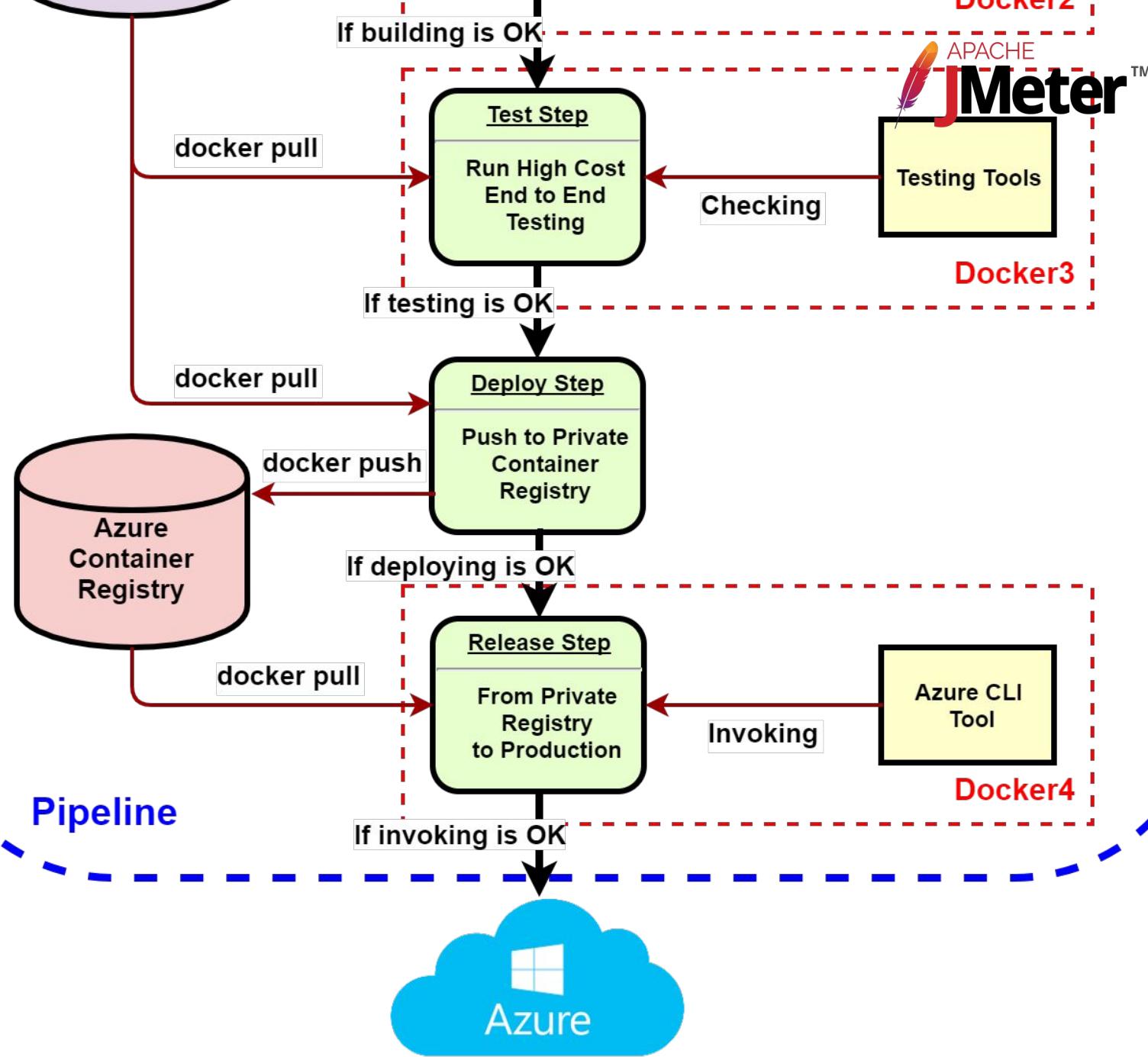
- 即日起開放組隊報名與提交參賽作品



系統架構







Project Repository Registry Issues Merge Requests Pipelines Wiki Snippets Settings

All 16 Pending 0 Running 0 Finished 16 Branches Tags

Run Pipeline CI Lint

Status Pipeline Commit Stages






Pipeline ID	Author	Commit Hash	Commit Message	Stages Status	Duration	Created At
#8184116	latest	d910aa0b	Change back to 'az webapp c...	Passed	00:03:59	8 minutes ago
#8184025		691e58fe	Change to use 'az ...	Failed	00:03:42	15 minutes ago
#8183917		39493a2b	Fix Release miss AZURE_DOCK...	Passed	00:05:05	25 minutes ago
#8177499		9906d49b	Fix deploy	Failed	00:04:08	7 hours ago
#8177234		2b721f9a	add bash & fix.	Failed	00:02:53	7 hours ago

Azure CLI 2.0 login

使用 [Azure CLI 2.0](#) 來建立 Azure 服務主體

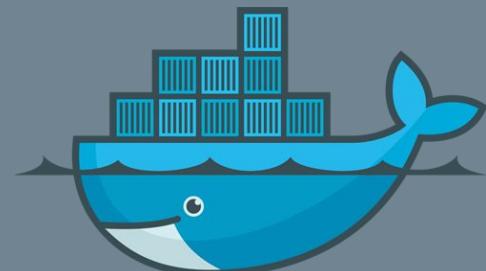
[Azure CLI 2.0 Reference](#)

CI/CD Pipeline

Create GitLab account

Let's play!!!

7. 容器與 Raspberry Pi IoT 整合應用



MQTT

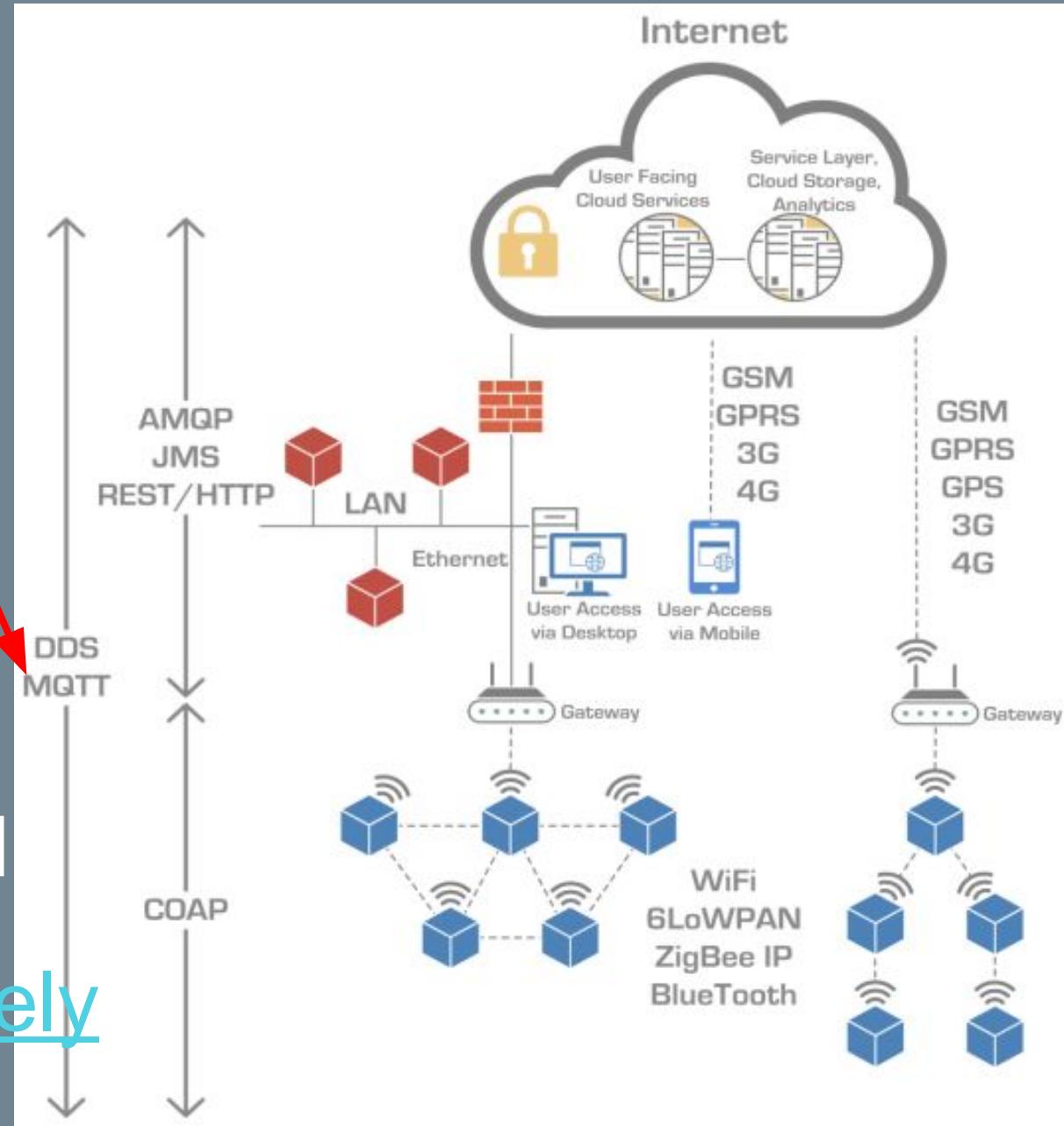
From IBM

IoT Protocol

Pub/Sub, QoS

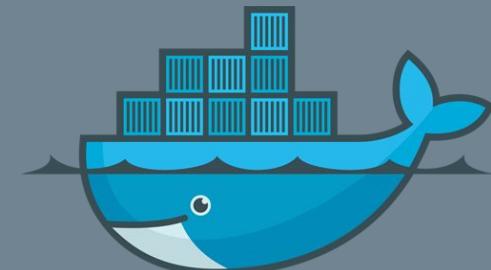
Arduino, mbed

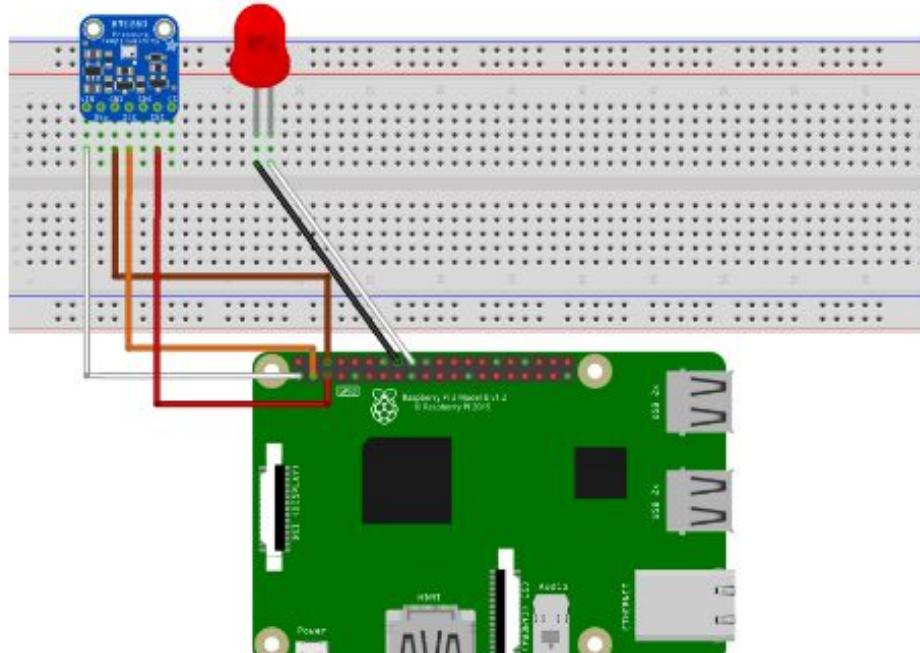
Mosquitto, Xively



QEMU 模擬 ARM 裝置

1. sudo apt-get install
qemu-user-static
2. docker pull philipz/rpi-raspbian
3. docker run -ti -v
/usr/bin/qemu-arm-static:/usr/bin/
qemu-arm-static
philipz/rpi-raspbian bash
4. apt-get update
5. uname -a





```
1  /*
2   * IoT Hub Raspberry Pi NodeJS - Microsoft Sample Code - C
3   */
4  const wpi = require('wiring-pi');
5  const Client = require('azure-iot-device').Client;
6  const Message = require('azure-iot-device').Message;
7  const Protocol = require('azure-iot-device-mqtt').Mqtt;
8  const BME280 = require('bme280-sensor');
9
10 const BME280_OPTION = {
11     i2cBusNo: 1, // defaults to 1
12     i2cAddress: BME280.BME280_DEFAULT_I2C_ADDRESS() // defa
13 };
14
15 const connectionString = '[Your IoT hub device connection
16 const LEDPin = 4;
17
18 var sendingMessage = false;
19 var messageId = 0;
20
21
```

Run Reset

Click 'Run' button to run the sample code(When sample is running,
Click 'Stop' button to stop the sample code running.

Click 'Reset' to reset the code.We keep your changes to the editor

> □

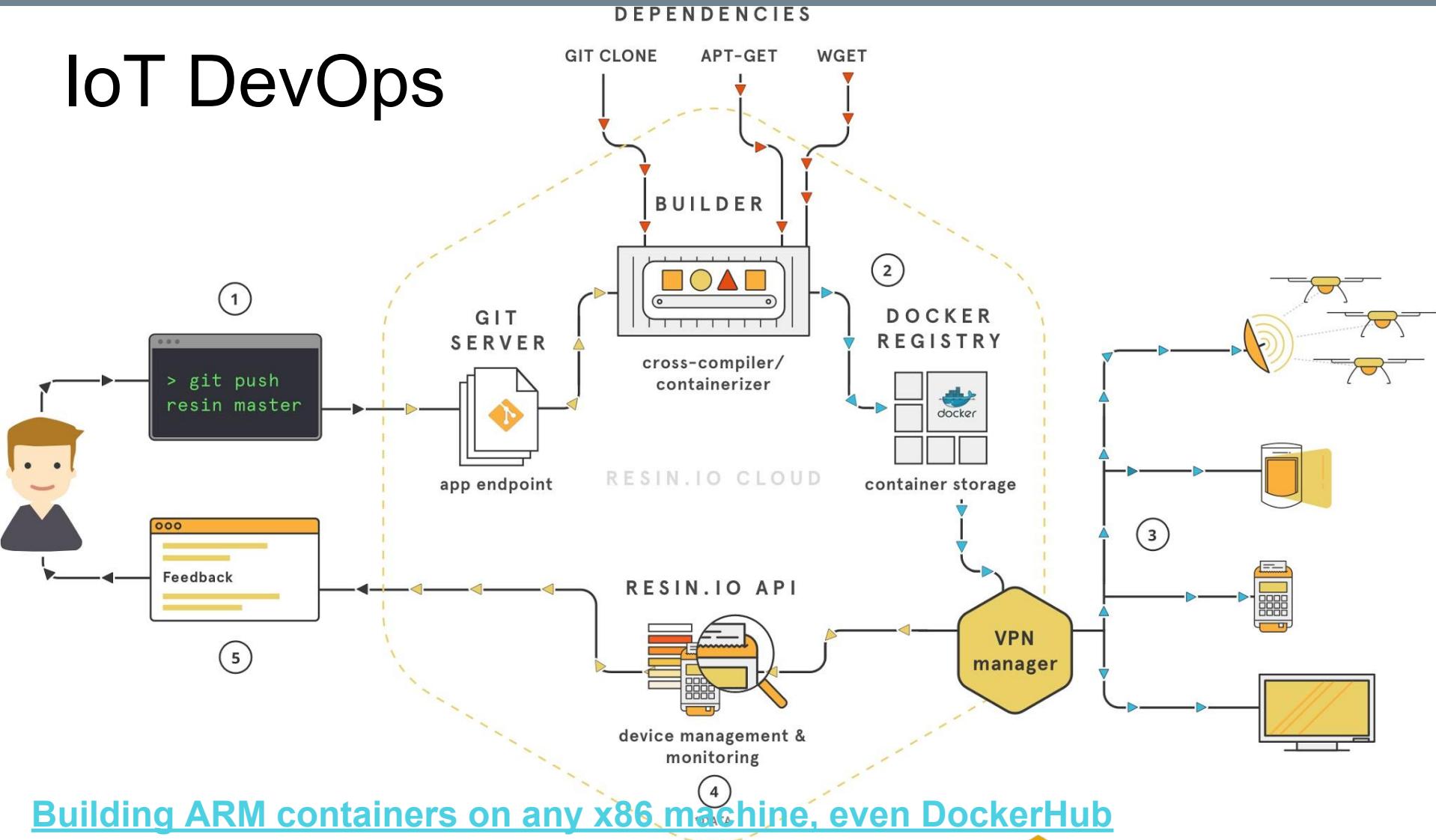
<https://azure-samples.github.io/raspberry-pi-web-simulator/>

將 Raspberry Pi 線上模擬器連線至 Azure IoT Hub (Node.js)

8. 以 GitLab 執行 Docker Compose IoT E2E 測試



IoT DevOps



Building ARM containers on any x86 machine, even DockerHub

GitHub source code



```
Removing intermediate container a3c1624aa8a7
Step 3/7 : ENV QEMU_EXECVE 1
--> Running in 3a295c9fedf2
--> 7c11a03043e3
Removing intermediate container 3a295c9fedf2
Step 4/7 : COPY qemu/cross-build-end qemu/cross-build-start qemu/qemu-arm-static qemu/sh-shim /usr/bin/
--> 080103c97ad8
Removing intermediate container 5c427bee6db9
Step 5/7 : RUN cross-build-start
--> Running in 60962855ba32
--> 117d4786ae11
Removing intermediate container 60962855ba32
Step 6/7 : RUN apt-get update && apt-get install -y mosquitto-clients
--> Running in 17033e80d624
Get:1 http://archive.raspbian.org jessie InRelease [14.9 kB]
Get:2 http://archive.raspbian.org jessie/main armhf Packages [13.3 MB]
Fetched 13.3 MB in 41s (322 kB/s)
Reading package lists...
Reading package lists...
Building dependency tree...
The following extra packages will be installed:
  libc-ares2 libmosquitto1 libssl1.0.0
The following NEW packages will be installed:
  libc-ares2 libmosquitto1 libssl1.0.0 mosquitto-clients
0 upgraded, 4 newly installed, 0 to remove and 44 not upgraded.
Need to get 996 kB of archives.
After this operation, 2552 kB of additional disk space will be used.
Get:1 http://archive.raspbian.org/raspbian/ jessie/main libssl1.0.0 armhf 1.0.1t-1+deb8u6 [853 kB]
Get:2 http://archive.raspbian.org/raspbian/ jessie/main libc-ares2 armhf 1.10.0-2+deb8u2 [66.7 kB]
Get:3 http://archive.raspbian.org/raspbian/ jessie/main libmosquitto1 armhf 1.3.4-2+deb8u1 [37.1 kB]
Get:4 http://archive.raspbian.org/raspbian/ jessie/main mosquitto-clients armhf 1.3.4-2+deb8u1 [39.5 kB]
debconf: delaying package configuration, since apt-utils is not installed
Fetched 996 kB in 1s (878 kB/s)
Selecting previously unselected package libssl1.0.0:armhf.
(Reading database ...
(Reading database ... 5%
(Reading database ... 10%
(Reading database ... 15%
(Reading database ... 20%
(Reading database ... 25%
(Reading database ... 30%
(Reading database ... 35%
(Reading database ... 40%
(Reading database ... 45%
(Reading database ... 50%
(Reading database ... 55%
(Reading database ... 60%
(Reading database ... 65%
```



build

Duration: 4 minutes 54 seconds

Runner: #40788

Commit a93fe409

First init

Pipeline #10935611 from mas

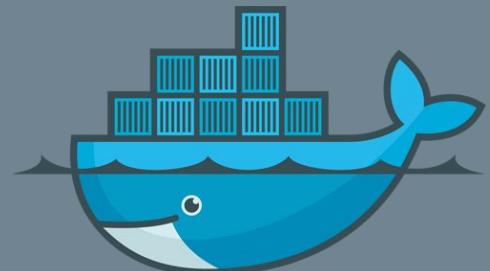
build

build

build

9. Moby & LinuxKit

介紹與入門



containerd

runc

swarmKIT

LINUXKIT

infrakit

Notary

Registry

LibNetwork

VPNKit

DataKit

HyperKit

Compose

GRPC



Component Library

Orchestration

Image mgmt

Secret mgmt

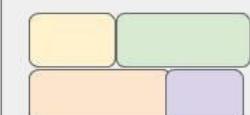
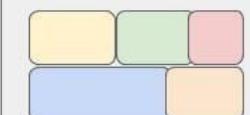
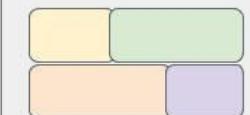
Config mgmt

Networking

Provisioning

Your Component
Here

Assemblies



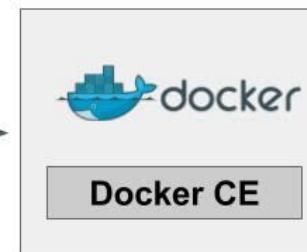
moby tools



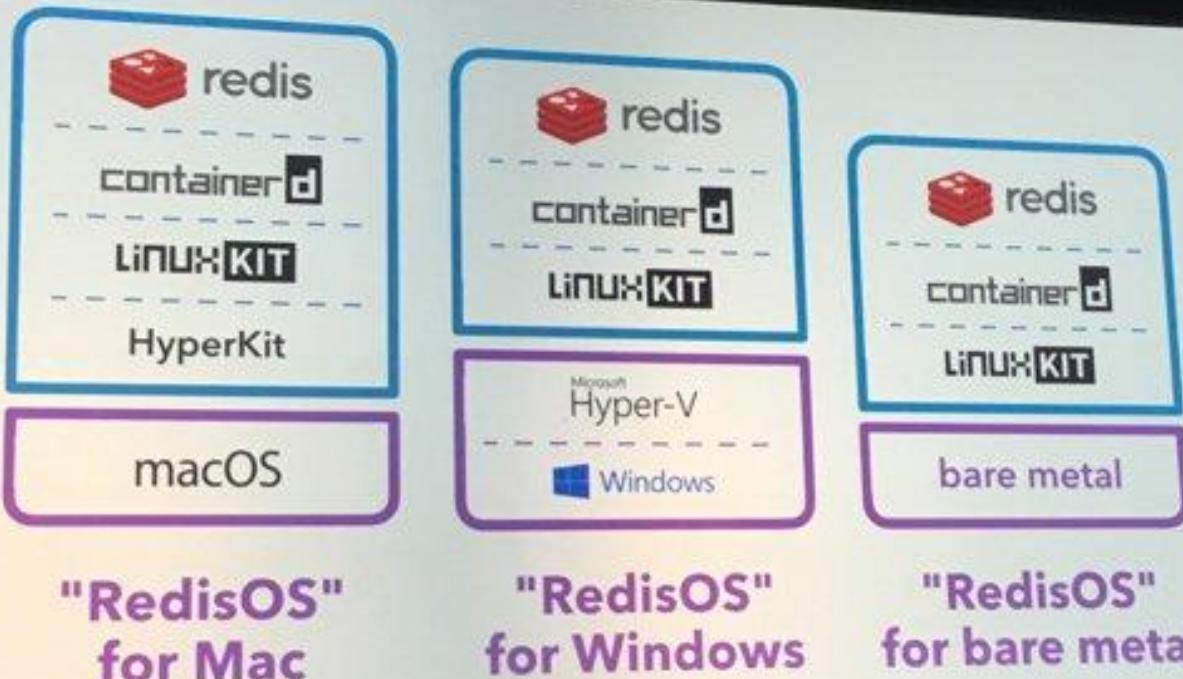
Upstream *projects*



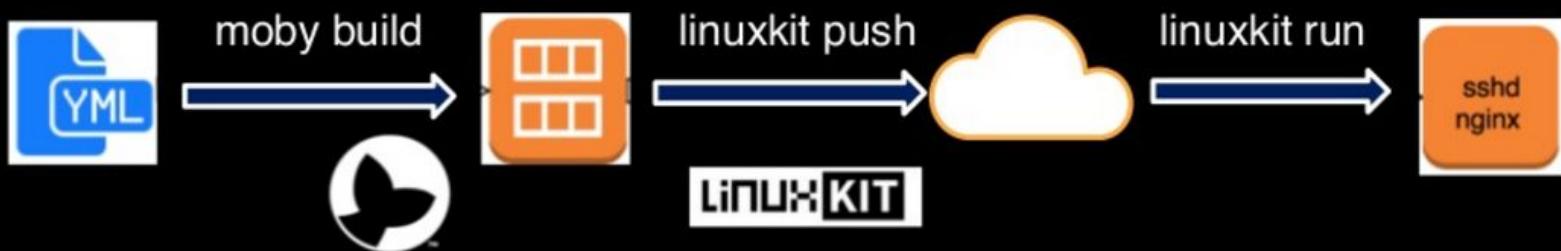
Your Product Here :)



Downstream *products*



LinuxKit Tooling



Moby Tool

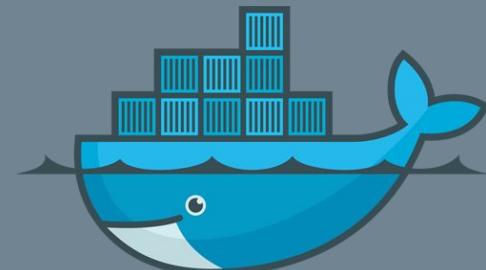
- A Tool for assembling a customised , modular & containerised application
- YAML files that defines the whole system through the set of containers
- Can generate range of output formats

LinuxKit

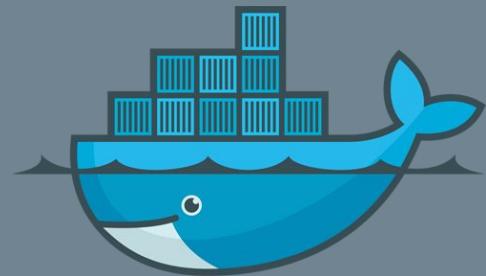
- A tool to push and Run LinuxKit OS / VM Images
- Support various backends & options to push it directly to Cloud Platform

Exercise

1. git clone <https://github.com/linuxkit/linuxkit>
2. sudo make
3. sudo cp bin/moby bin/linuxkit /usr/local/bin
4. cd example && moby build redis-os.yml
5. linuxkit run -publish 6379:6379/tcp redis-os
6. docker run -ti --rm redis bash
7. redis-cli -h 172.17.0.2



10. 結語



The Docker Stack

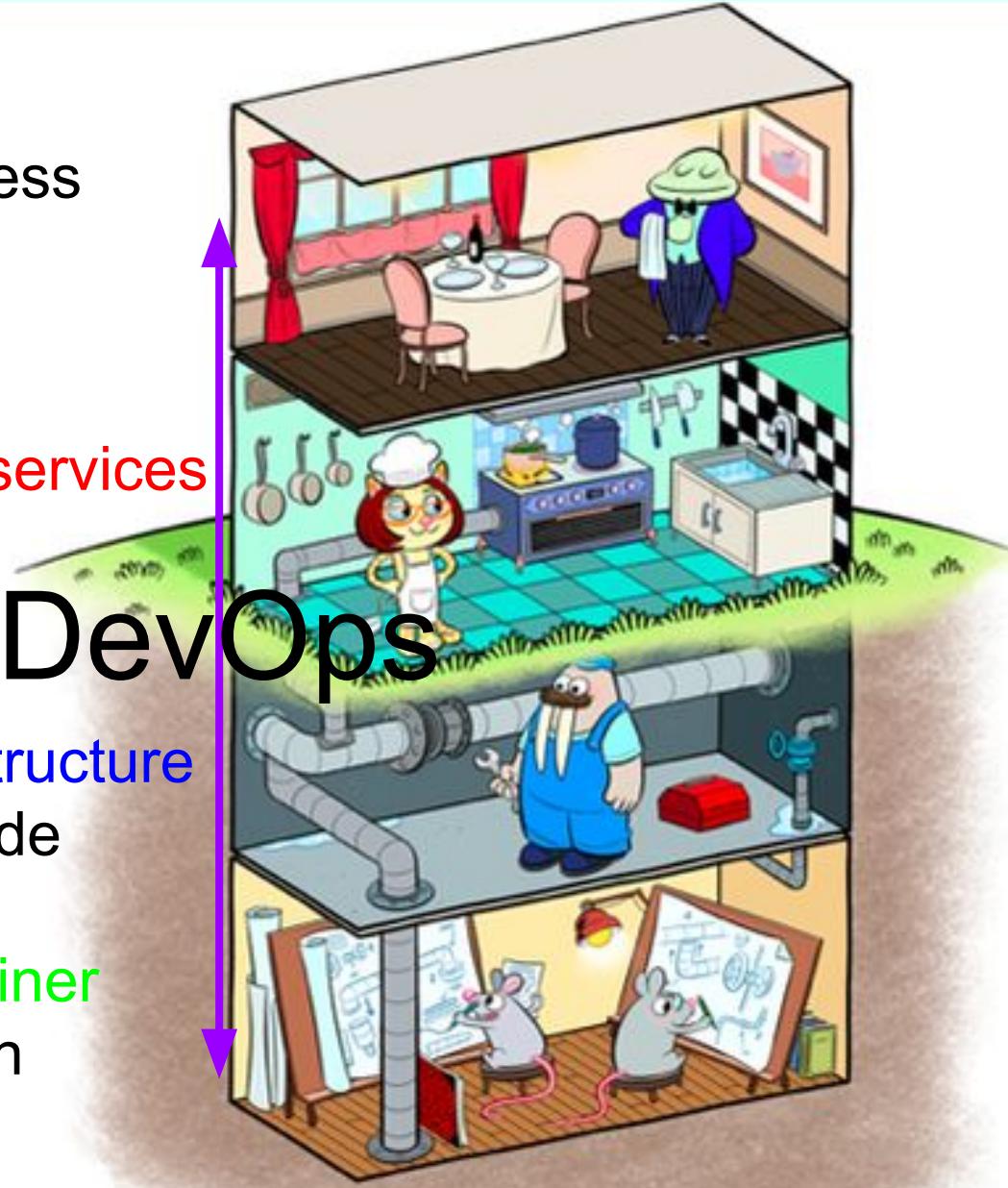
Business
model

Microservices

DevOps

Infrastructure
as Code

Container
Design



The Docker Stack

基礎架構
即程式碼

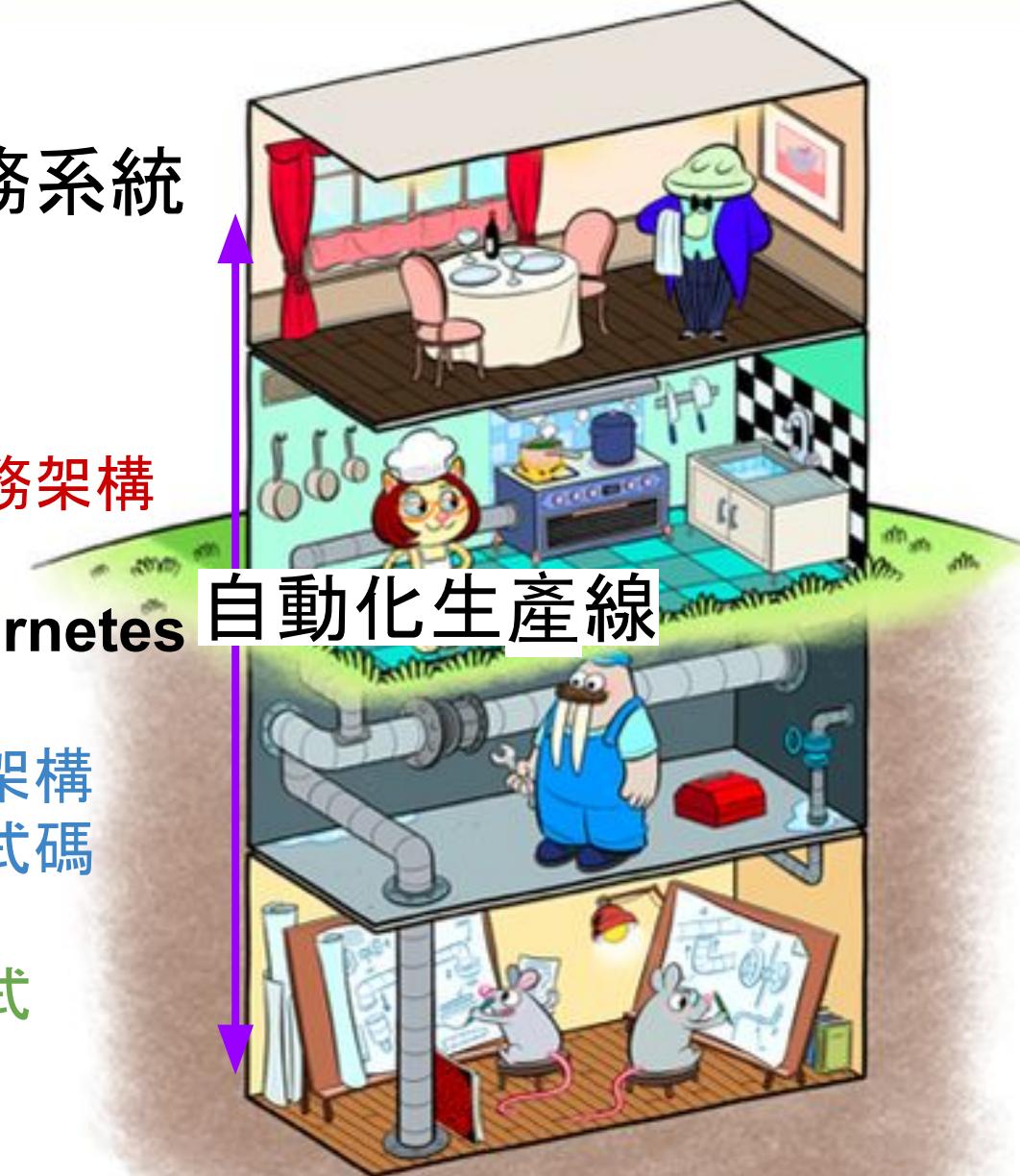
容器式
設計

微服務架構

Kubernetes

*業務系統

自動化生產線



	引爆点	开发技术载体	软件架构风格	客户端发布	服务器端交付
Mainframe	IBM System/360 (第一代多用途计算主机)，首次分离“架构”与“实现”的概念	打孔机、汇编、Fortran 77	一体	“笨终端” – Dumb Terminal	见过并且活着的人已经不多。。。
C/S	X86/PC、RISC、摩尔定律、HP/Sun/传说中的 SGI 和 NexT 工作站	Unix、4GL、Sybase、高级语言、X/Motif、DCE RPC、DCOM、CORBA-IIOP	2 层架构、关系型数据库主导	软件 CD 安装、升级	软件 CD 安装升级、数据库迁移
Multi-tiered	互联网/Web	Struts+Spring+Hibernate、Tomcat、WebLogic、Websphere、Oracle、PHP、Ruby...	3 层至多层 – 展示层、整合层、业务逻辑层、持久层、存储层。。。	浏览器刷新一下页面	手工部署脚本、JAR、WAR、存储过程等等，数据库迁移。开始有 CI
SOA + RIA	互联网技术进入企业	SOAP、REST、Flash/AIR、AJAX、RMI/其他 Remoting、WSDL、UDDI。。。	对于用户像 C/S，对于开发者是 Multi-tiered	浏览器刷新一下、升级（例如通过 AIR）等	同上
Distributed	Web 2.0、NoSQL (BigTable)、云	函数类、动态、脚本语言，非关系型数据库，一致性算法 (Raft、Paxos、Zookeeper)，响应式服务器 (nginx、Node.js。。。)	Reactive 响应式架构、Heroku 12-factors	多元化 – 手机 App、内嵌浏览器（例如 Webkit、Chromium）的富终端、网站	自动化部署 – Chef、Puppet、Ansible、CI/CD、DevOps 开始
Containerized	LXC、Docker	同上，但更规范（通过 PaaS 如 K8S – 遵循其最佳实践）	同上，但更规范	同上	同上，加不可变基础设施 (immutable infrastructure) 运维，加基于容器编排技术的 CI/CD
Serverless?	Amazon Lambda	脚本类语言更容易	透明	同上	仅需交付源代码

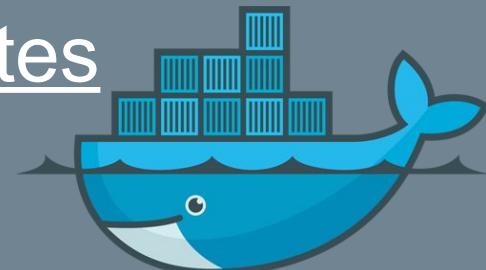
Exercise & Self-learning

1. Docker Basic - Katacoda by Philipz
2. Docker Trainning
3. Docker Free self-paced courses
4. Docker Tutorials and Labs

Online Self-learning, <http://testdriven.io>

Offical Online Lab

Scalable Microservices with Kubernetes
- Udacity



docker



老闆眼中的docker

外界認為的docker

dxxr Inc.眼中的docker



原本以為的docker

實際上的docker



最終成為的docker

dockers
DOCKER



老闆眼中的docker



原本以為的docker

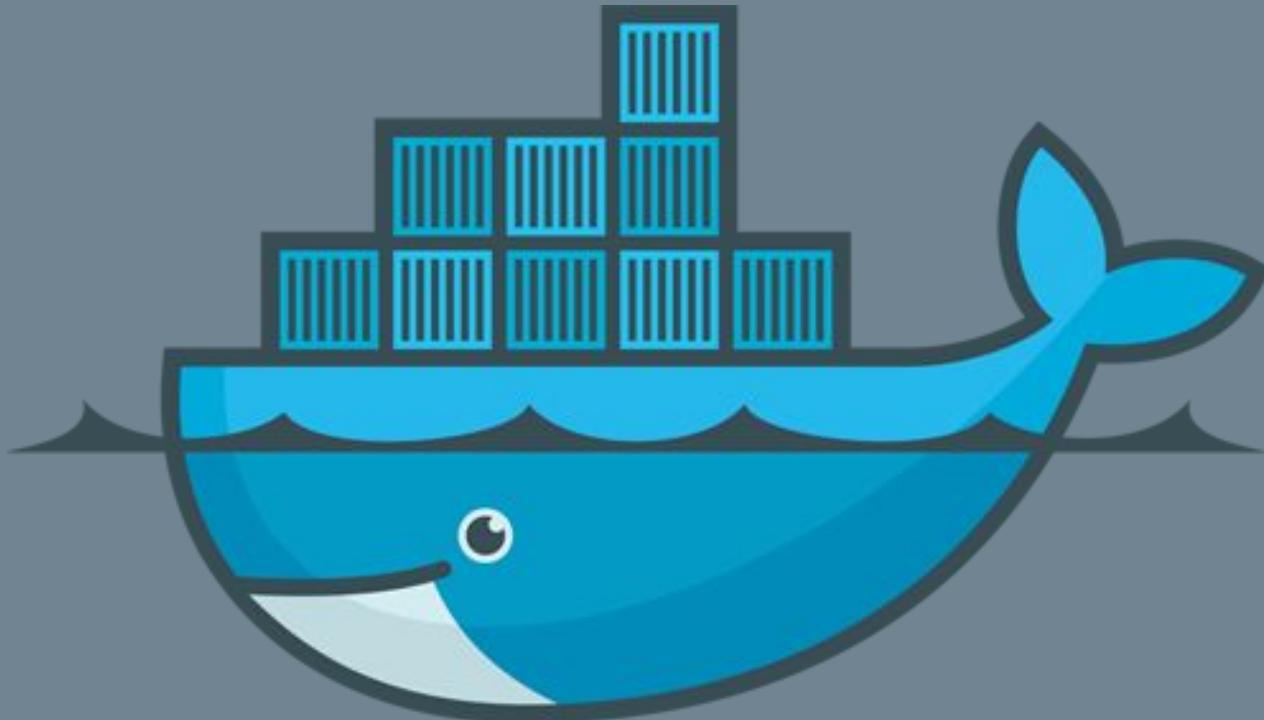
可以解救您的婚姻



.眼中的docker



變成的docker



Hope You Love Docker
So long!