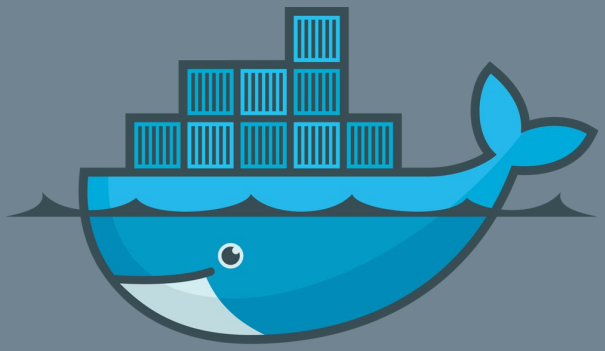


# 逢甲大學 docker 研習班

Docker.Taipei Philipz(鄭淳尹)

2017-01-18

https://github.com/philipz/workshop_fcu

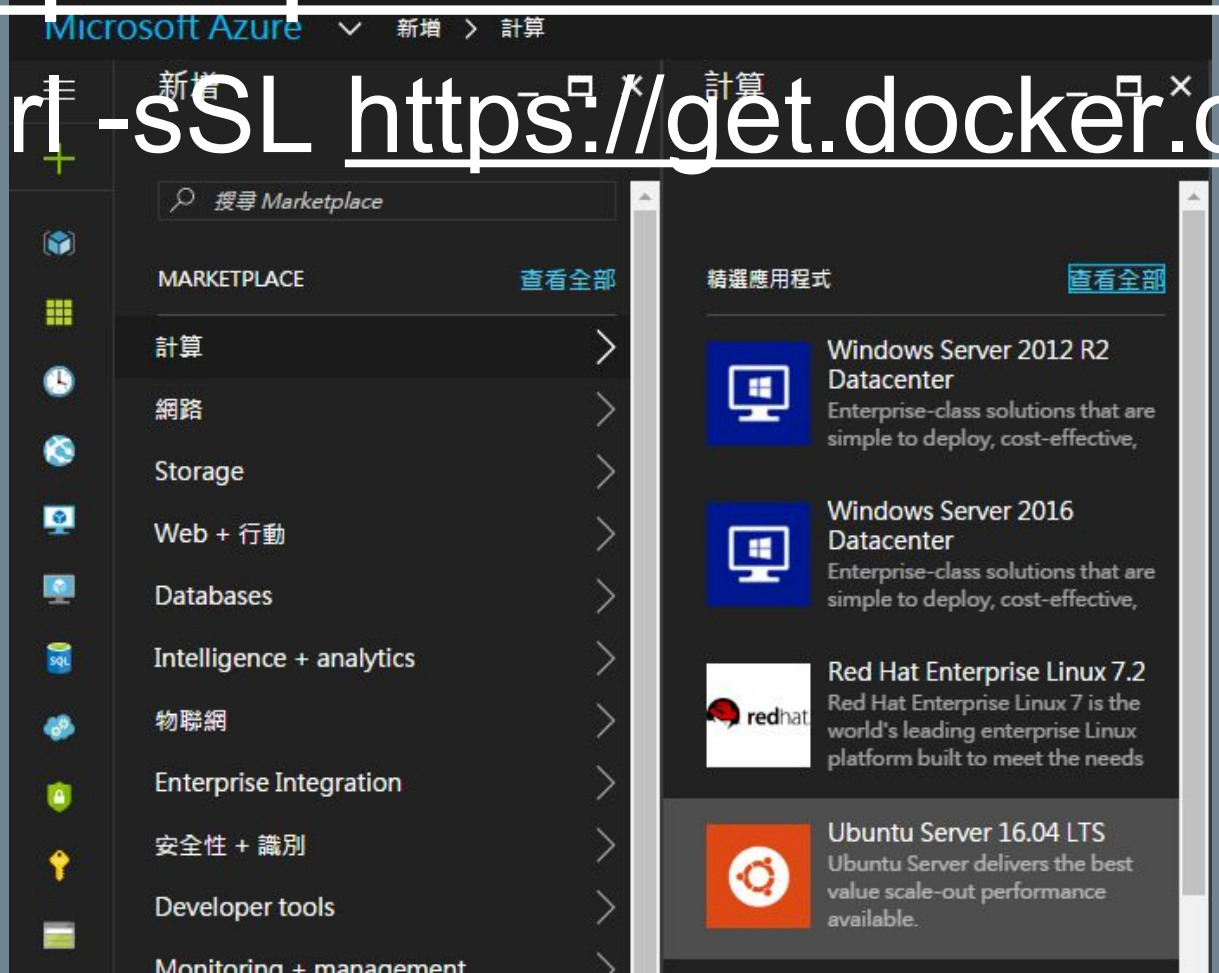
# Today Topics

1. Docker Machine introduction & CLI
2. Docker Machine to create cloud VM
3. Docker Swarm introduction & CLI
4. Machine and Swarm Cluster
5. Docker Swarm networking
6. Docker Swarm playground & Swarm service
7. The future of cloud computing and cloud service scope.

# Microsoft Azure

https://portal.azure.com/

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




# Install Docker Compose

sudo curl -L "https://github.com/docker/compose/releases/download/1.9.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

and

sudo chmod +x /usr/local/bin/docker-compose

docker-compose -v

# Review Docker Compose

GitHub: workshop_ishou/compose_wp_proxy

WordPress example of previous week

Add new service - Nginx Reverse Proxy

docker-compose scale wordpress=2

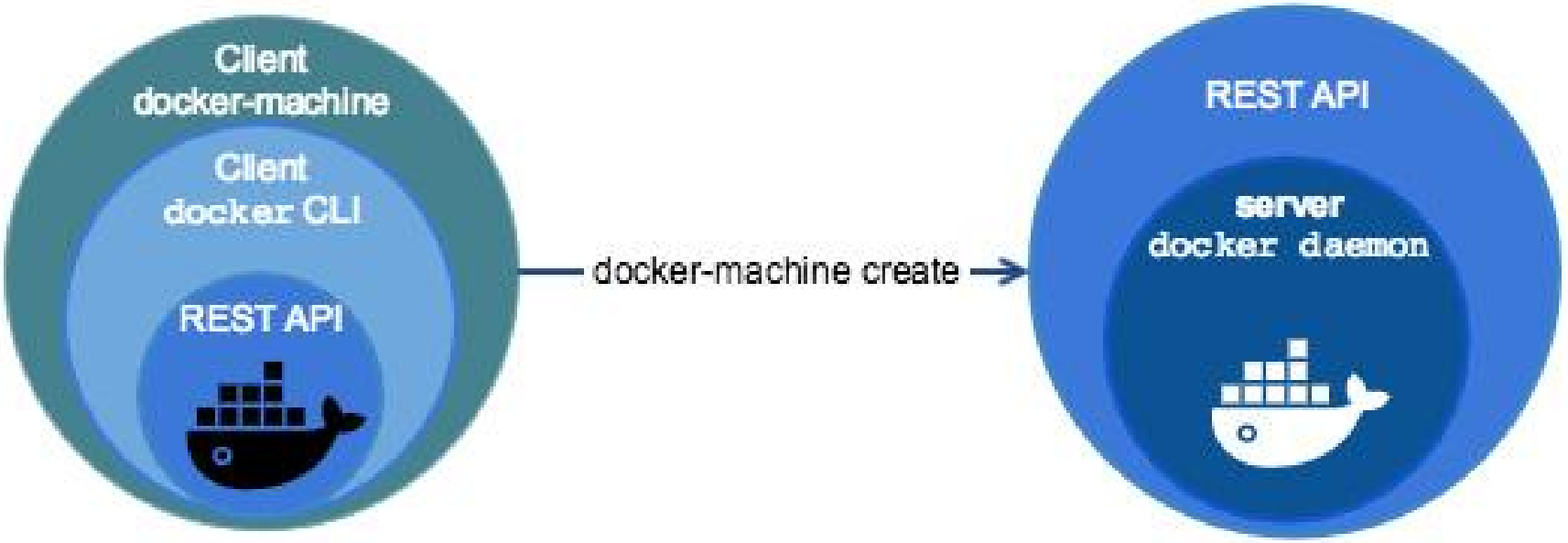
# DNS-based service discovery

# $nslookup wordpress

# 1.1 Docker Machine Introduction

# 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


Client
docker-machine
Client
docker CLI
REST API
docker-machine create
REST API
server
docker daemon


docker run image
vmware
vSphere
Microsoft
Azure
amazon
web services

# 1.2 Docker Machine Command-line

# Install Docker Machine

sudo curl -L "https://github.com/docker/machine/releases/download/v0.8.2/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)

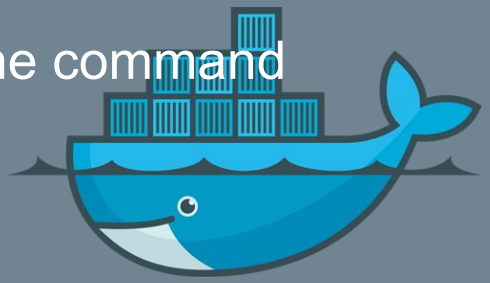
Commands:

| | |
|---|---|
| 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:

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



# 2. Docker Machine to create cloud VM

# 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 microsoft/azure-cli

azure login, then enter the code

azure vm image list-skus

azure vm image list eastasia canonical ubuntuserver 16.04.0-LTS

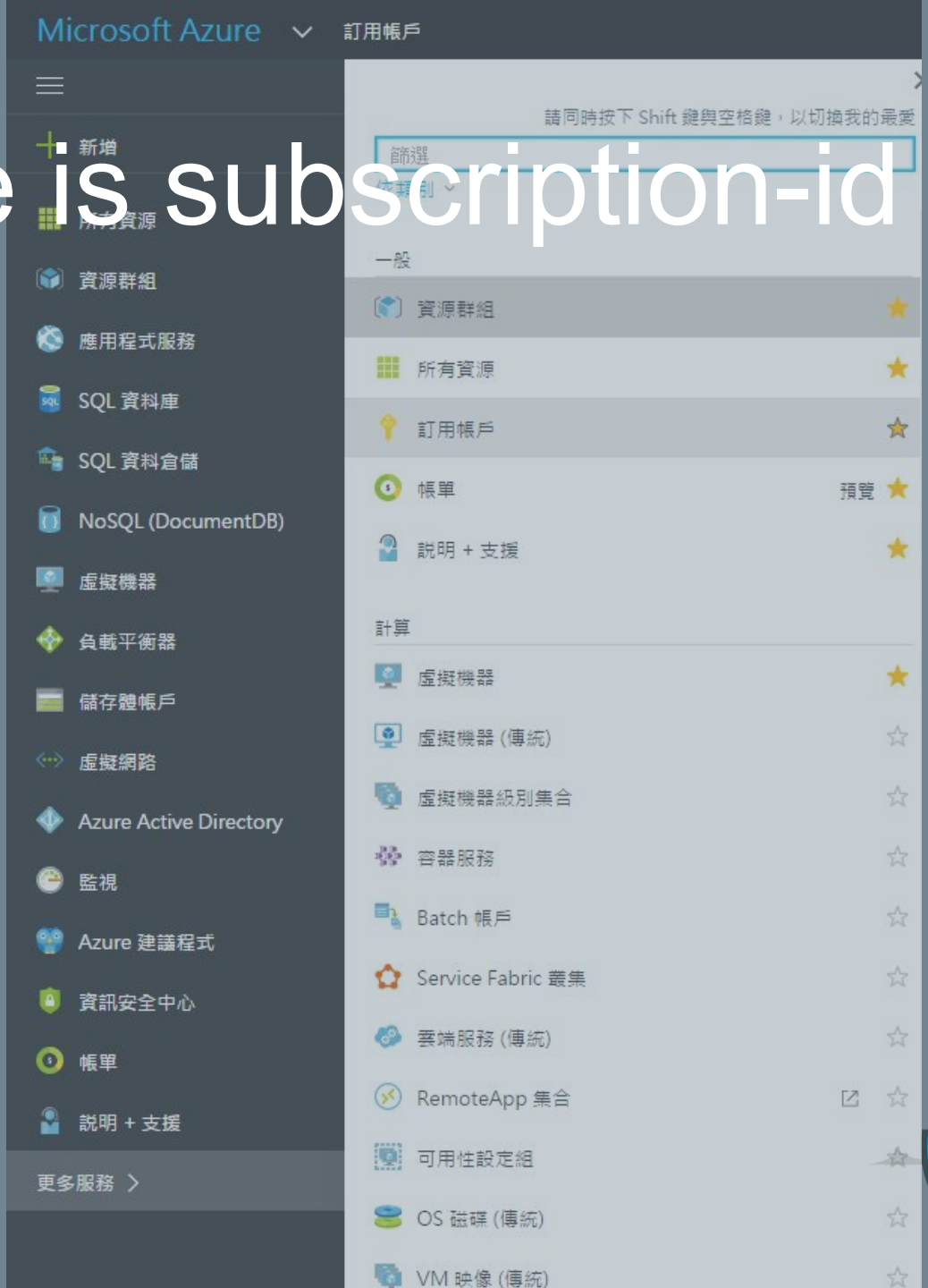
azure vm docker create

azure vm sizes --location "eastasia" | more

# Machine Create

- Azure VM Size
- docker-machine create -d azure --azure-subscription-id="XXXXX" --azure-location="eastasia" --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 ?




docker-machine ssh docker-0-0-1

sudo usermod -aG docker $USER

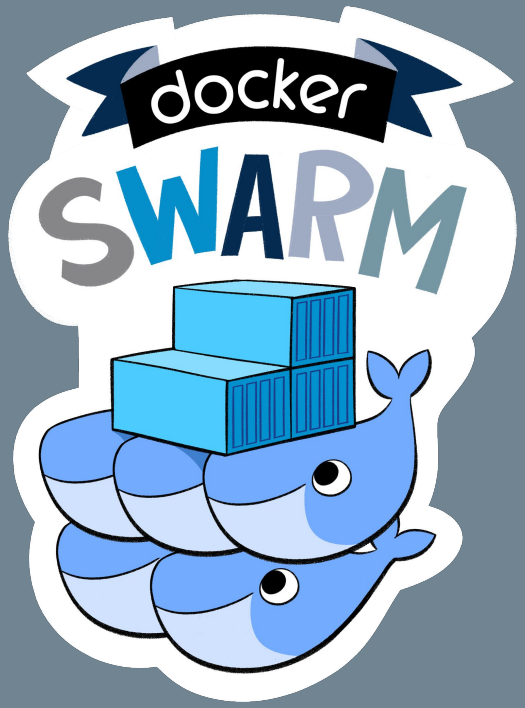
# 3.1 Docker Swarm Introduction

# Docker Swarm

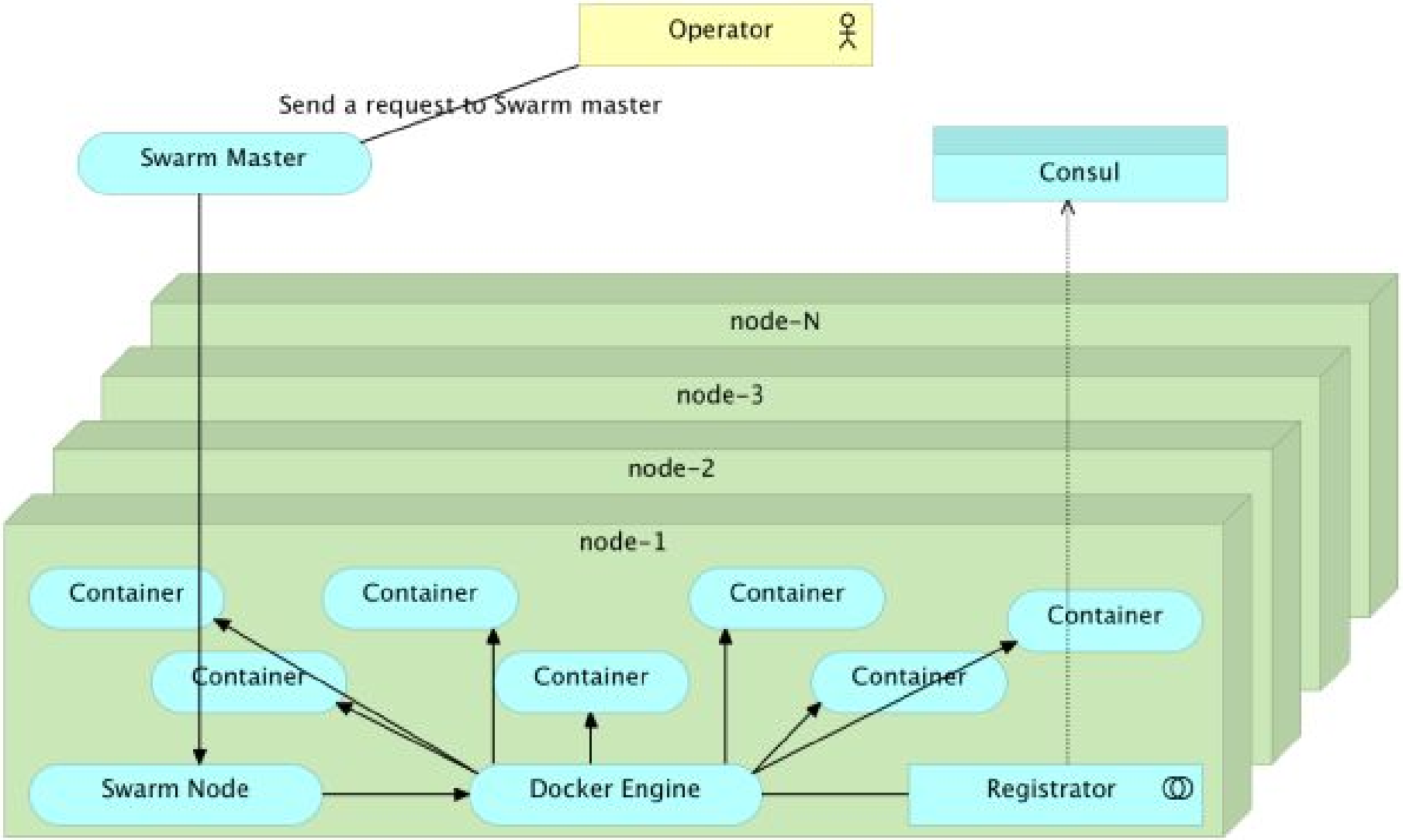
- Docker-native clustering system
- From v1.12 is default feature.
- Docker overlay network

[Play Swarm by Docker in Docker]

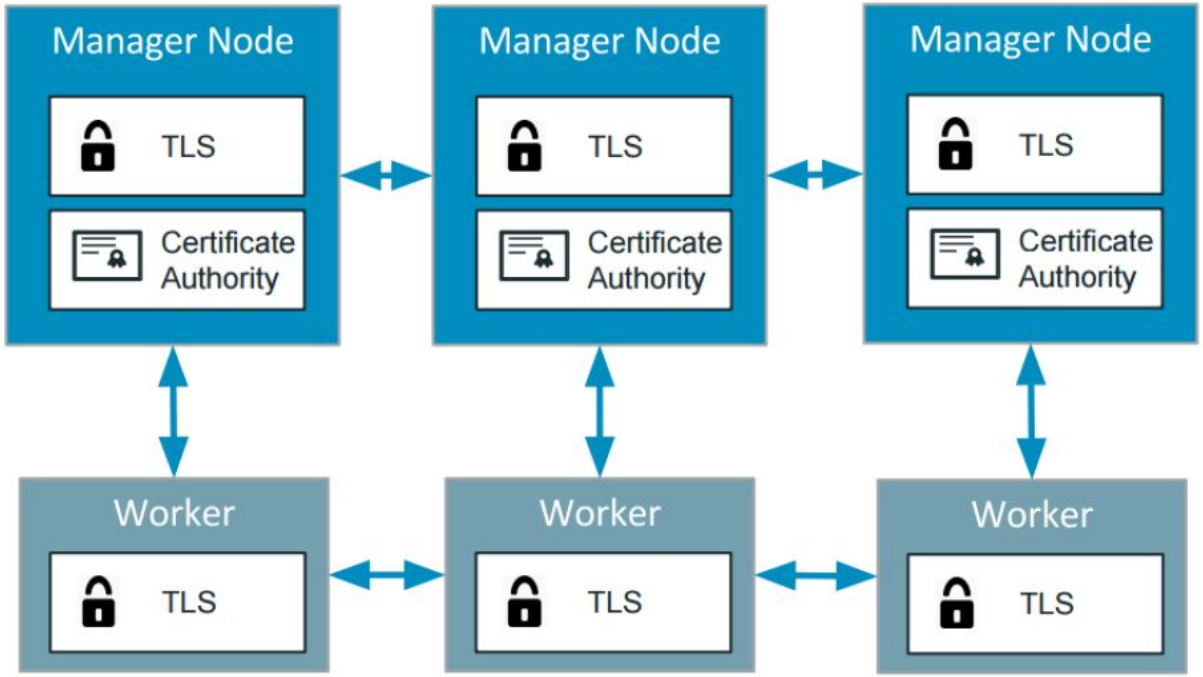
kubernetes



# Old Swarm Architecture



# New Swarm Mode (1/2)

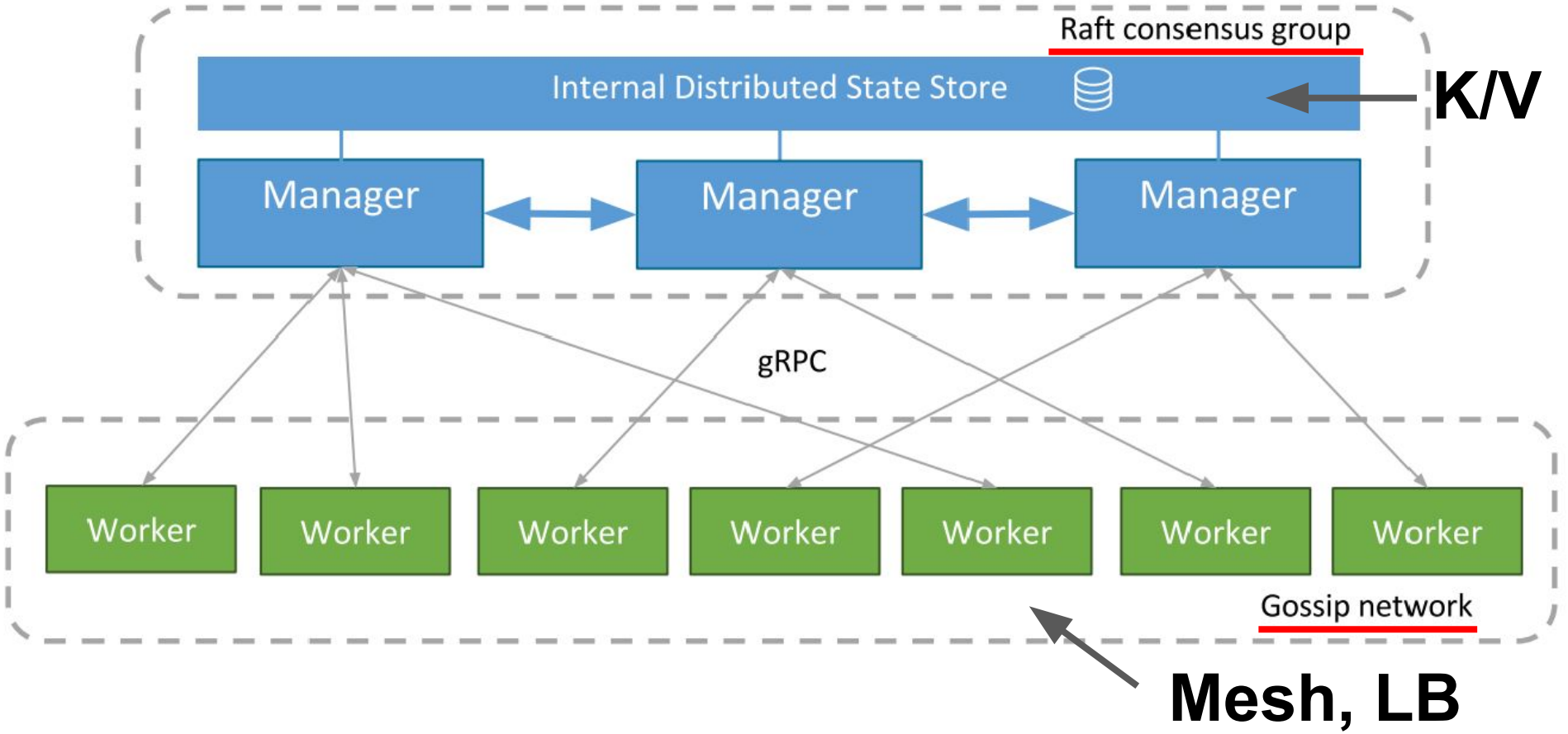



- Cryptographic node identity
- Automatic encryption and mutual auth (TLS)
- Automatic cert rotation
- External CA integration

[Byzantine Generals Problem]

# New Swarm Mode (2/2)

Consul, HashiCorp



# 3.2 Docker Swarm Command-line

# 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. Docker Machine and Swarm Cluster

# Machine Create Again

- docker-machine create -d azure --azure-subscription-id="XXXXX" --azure-location="eastasia" --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

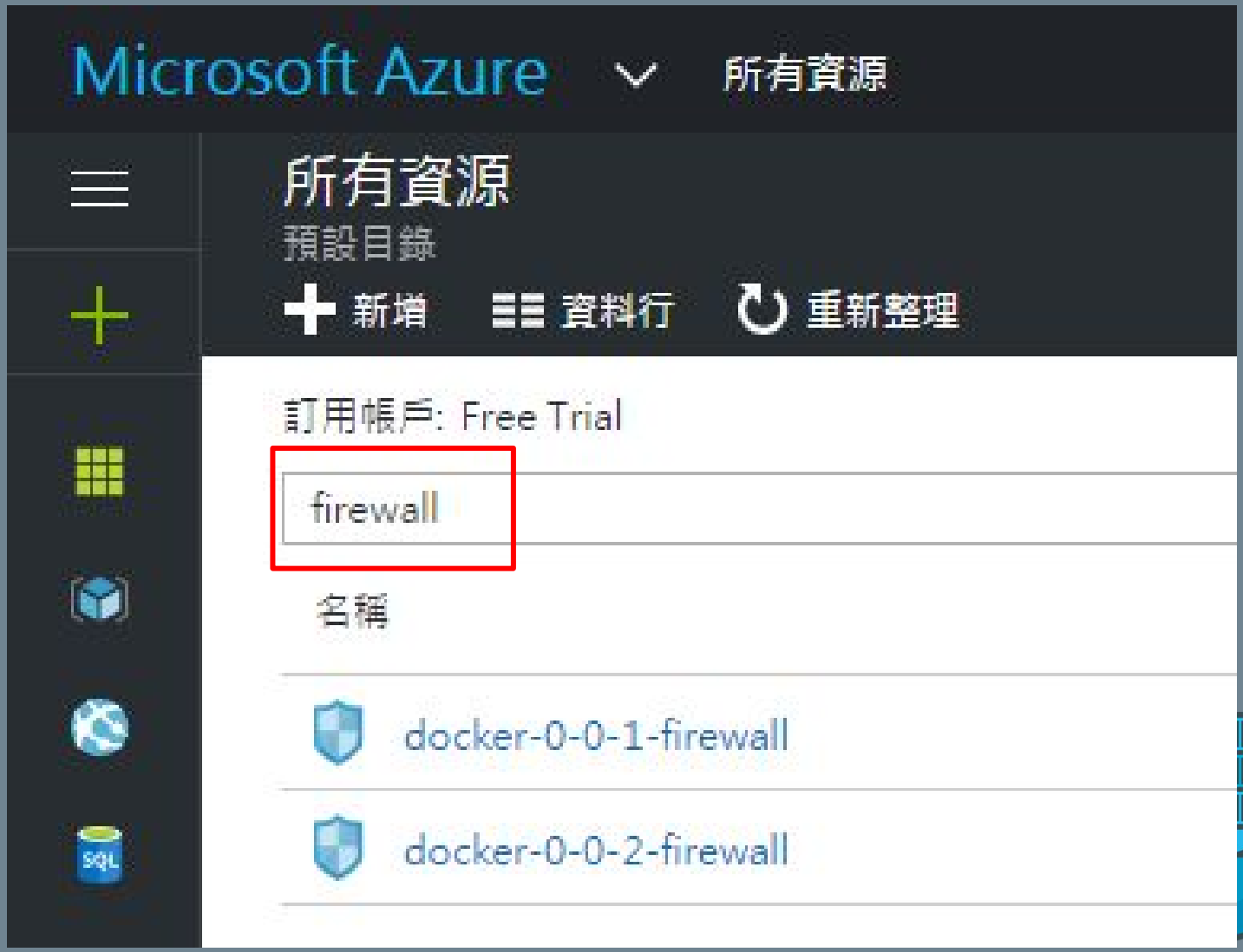
$ docker-machine ssh docker-0-0-3

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

# Communication Ports

Docker Remote API: 2376  Swarm Listen Port: 2377

Container network discovery: 7946 TCP/UDP

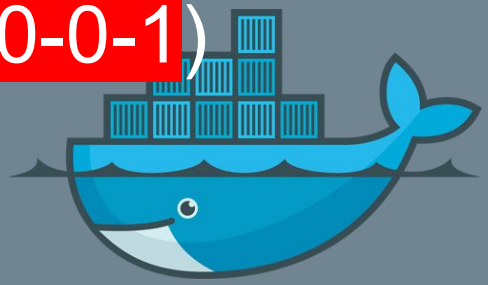
Container overlay network: 4789 UDP

| 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 promtoe docker-0-0-2
$ docker node ls
$ docker demtoe docker-0-0-2
$ docker node ls
```

# 5. 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

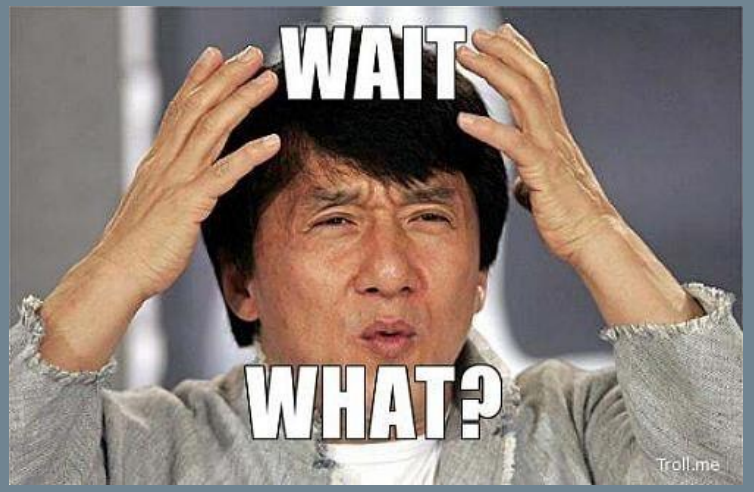
- infoblox

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

# 6. Docker Swarm playground & 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 my-network
$ 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
```

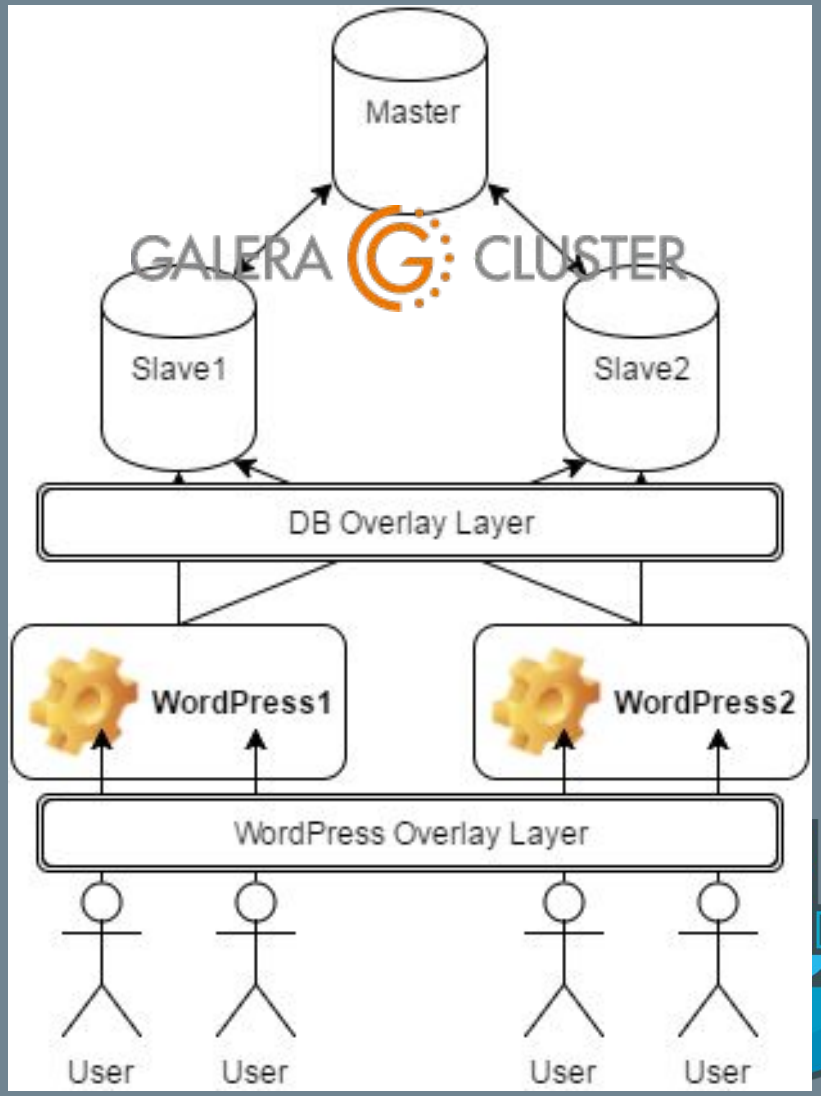
docker service update docs

# 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



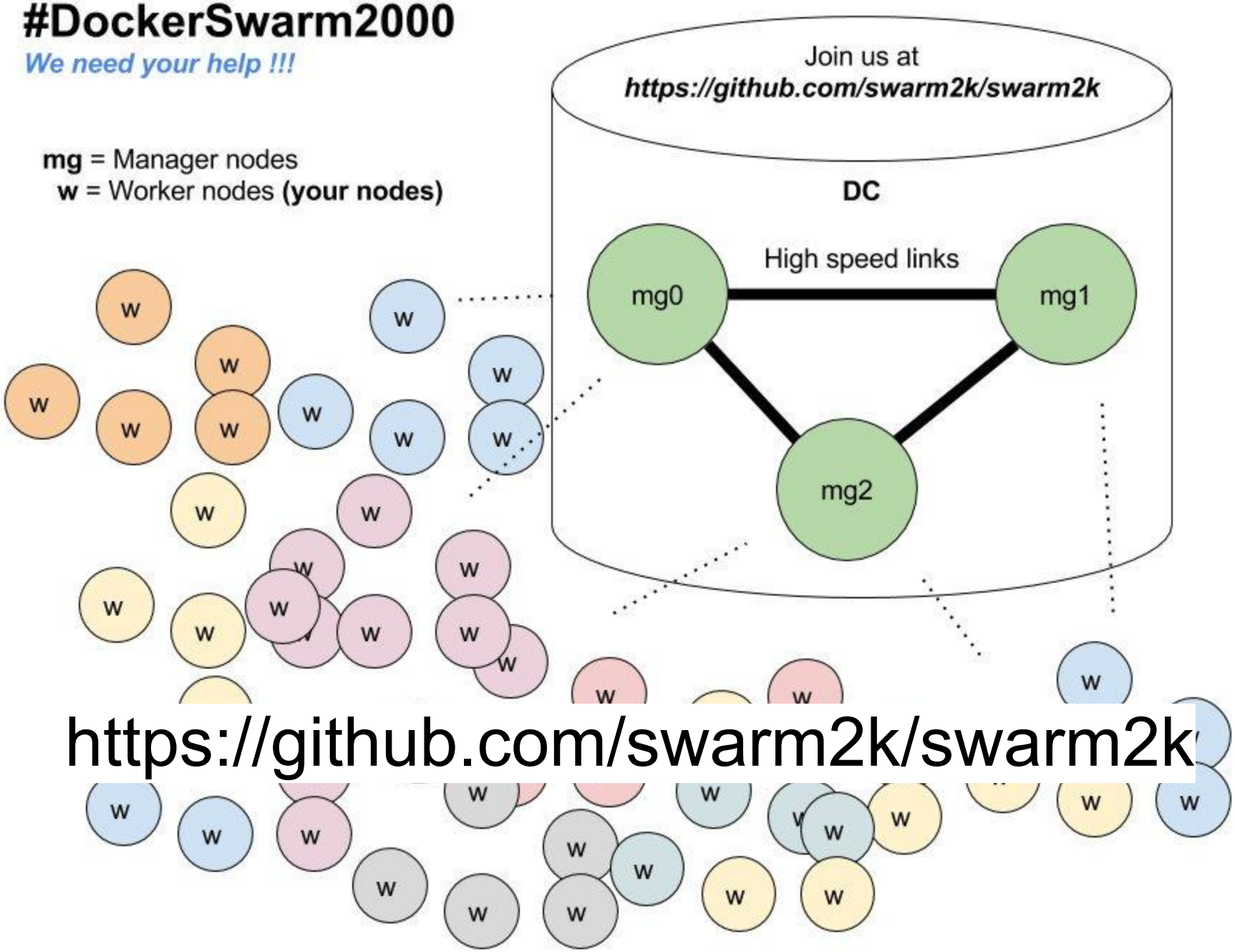
# More Advanced Docker Workshop

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

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

# #DockerSwarm2000

*We need your help !!!*




https://github.com/swarm2k/swarm2k

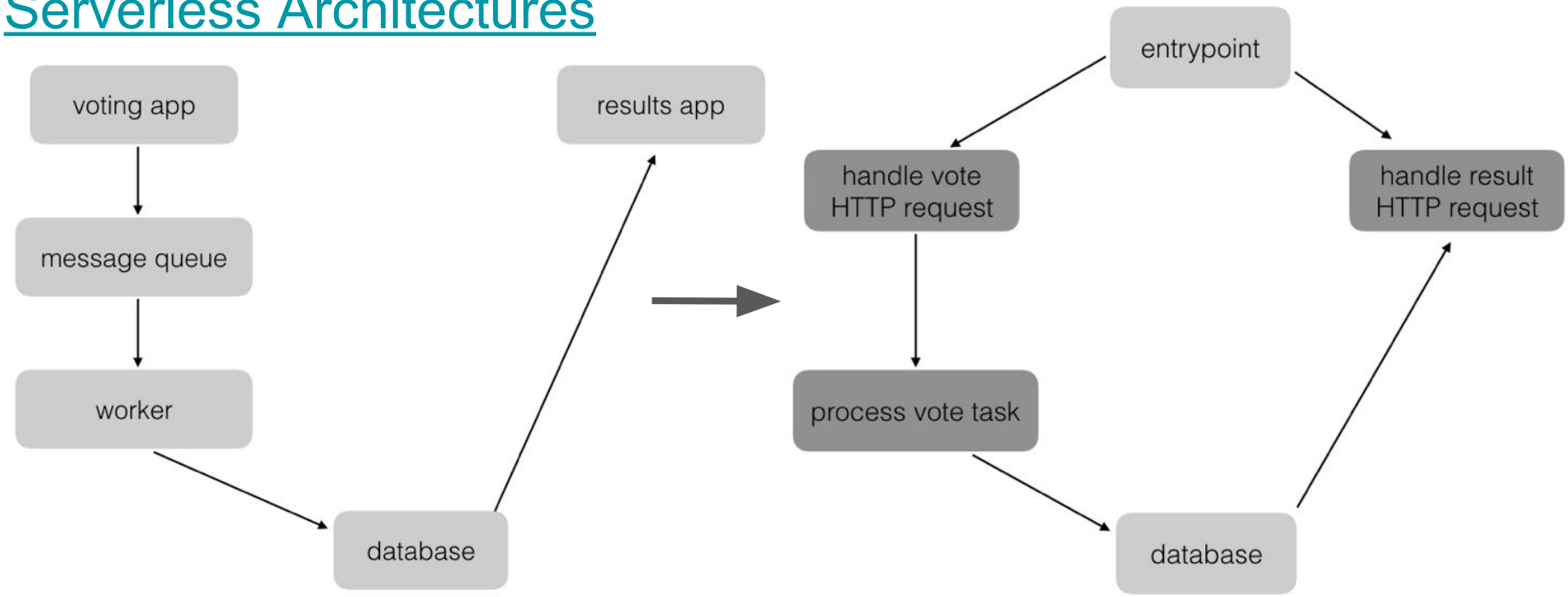
# 7. The Future of Cloud Computing

# Serverless Arch. & Docker

[Building Serverless Apps With Docker]

[Serverless Docker Example Voting App]

[Serverless Architectures]




Container is the SAME.
LIVE. DIE. REPEAT.

# Container Orchestration

**Docker Swarm**

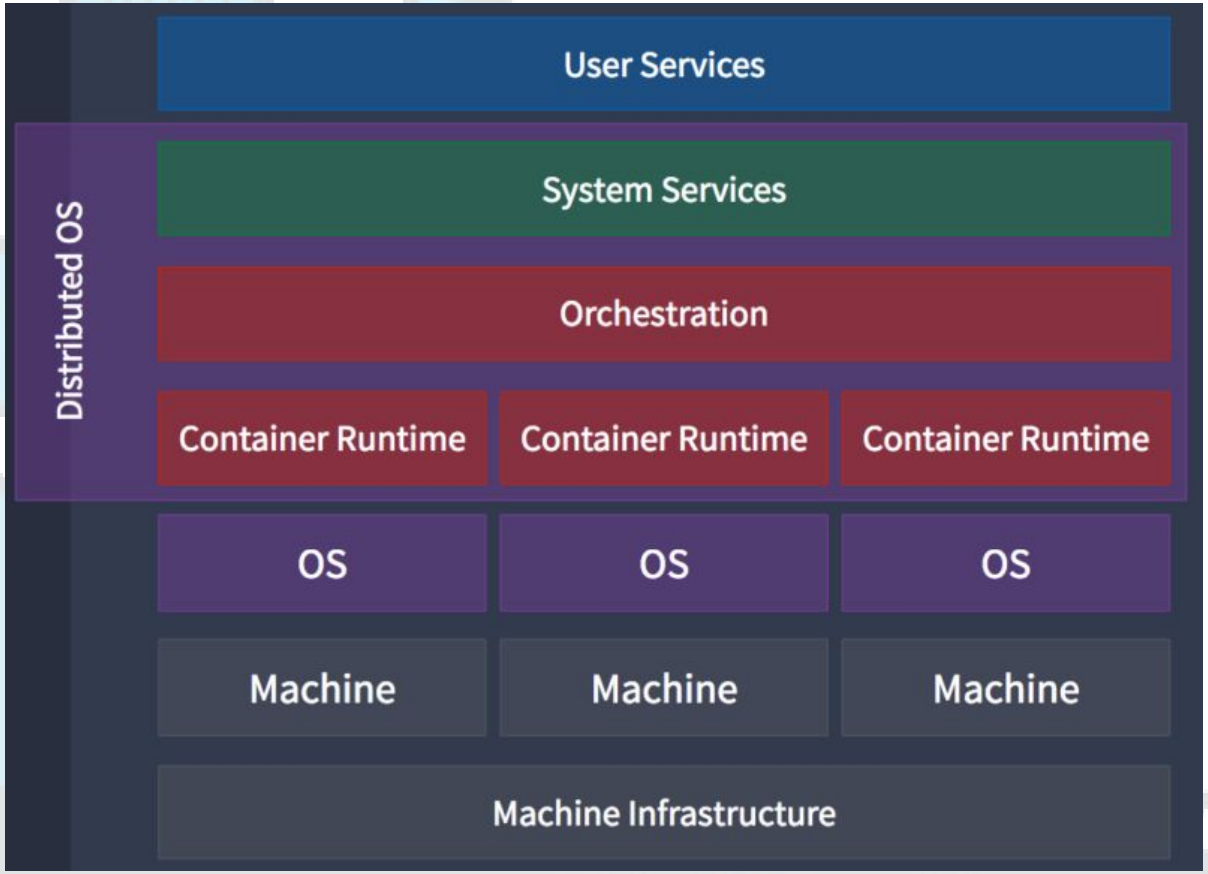
**<u>Kubernetes</u>**

**DC/OS**

**Rancher**

**Docker Datacenter**

**???**



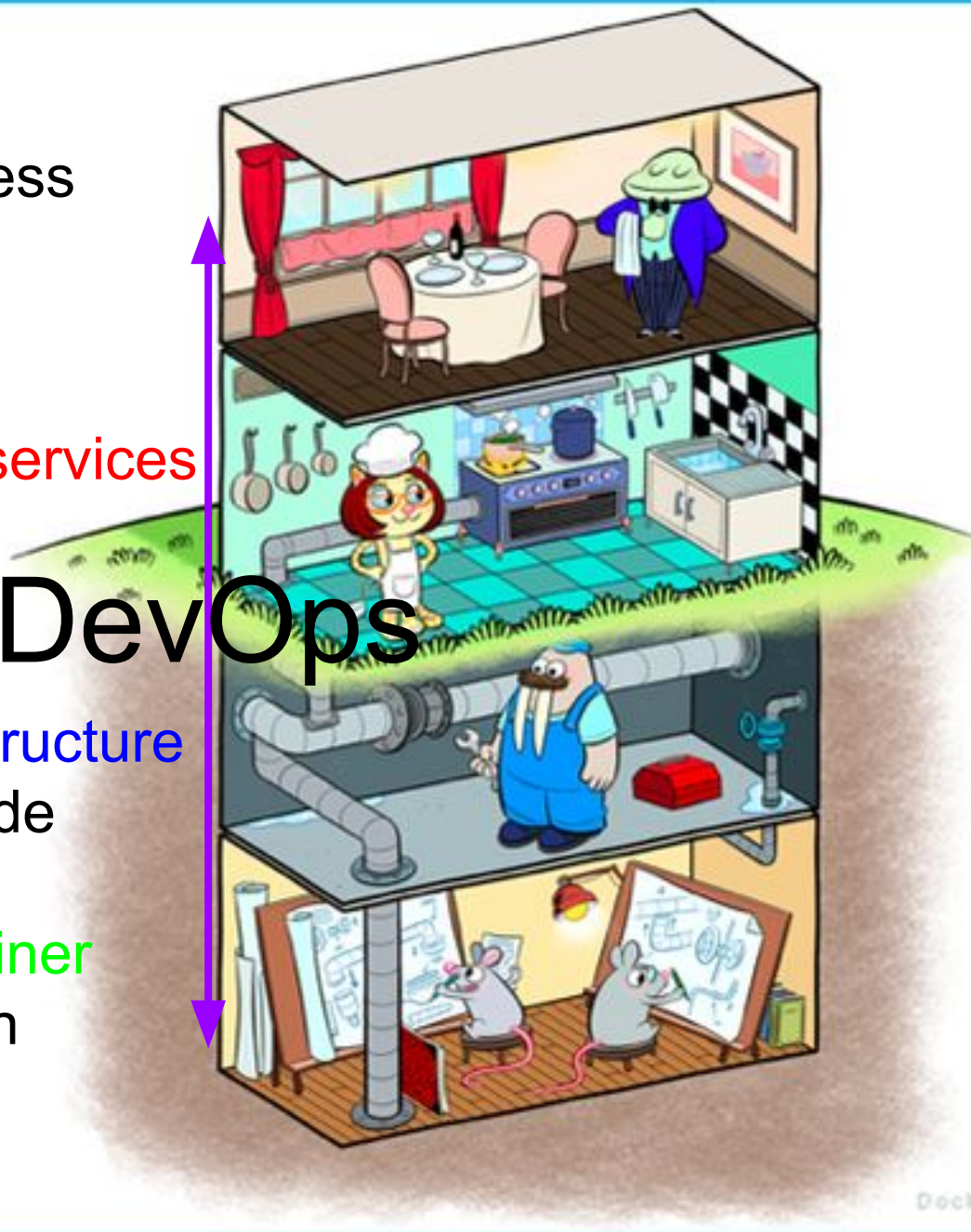
# The Docker Stack

Business model

Microservices

Infrastructure as Code

Container Design



# The Docker Stack

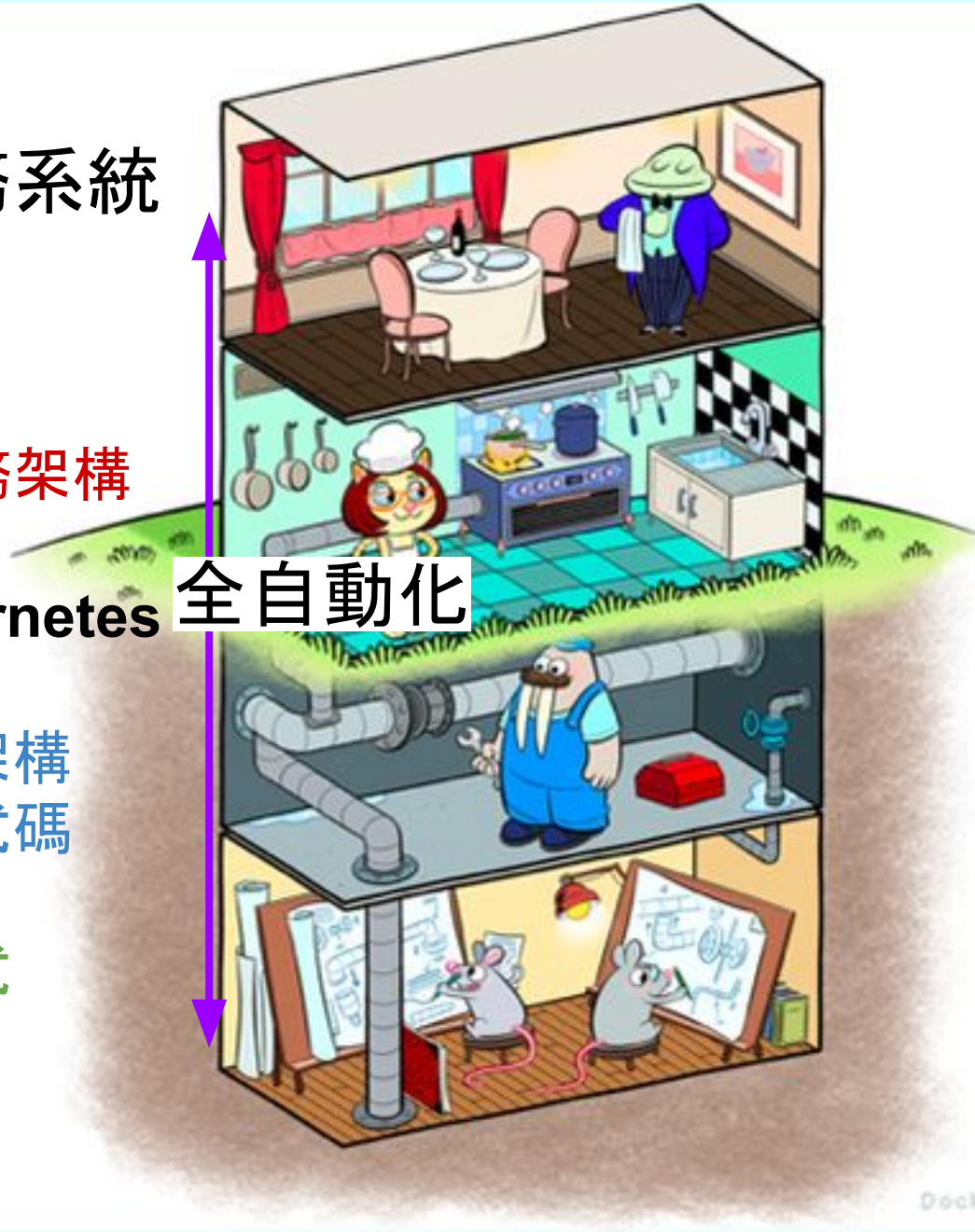
*業務系統

微服務架構

Kubernetes 全自動化

基礎架構
即程式碼

容器式
設計



# System architecture
# = Organizational architecture

大量自動化、系統人員減少

Conway's Law

DevOps in the Enterprise

Microservices AntiPatterns

Microservices in action

奇谈怪论:从容器想到去IOE、去库存和独角兽

容器化技术构建一个“反脆弱”的交易系统

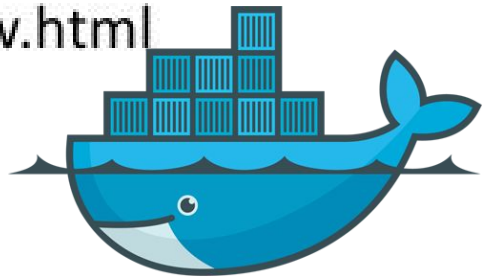
# Conway's Law

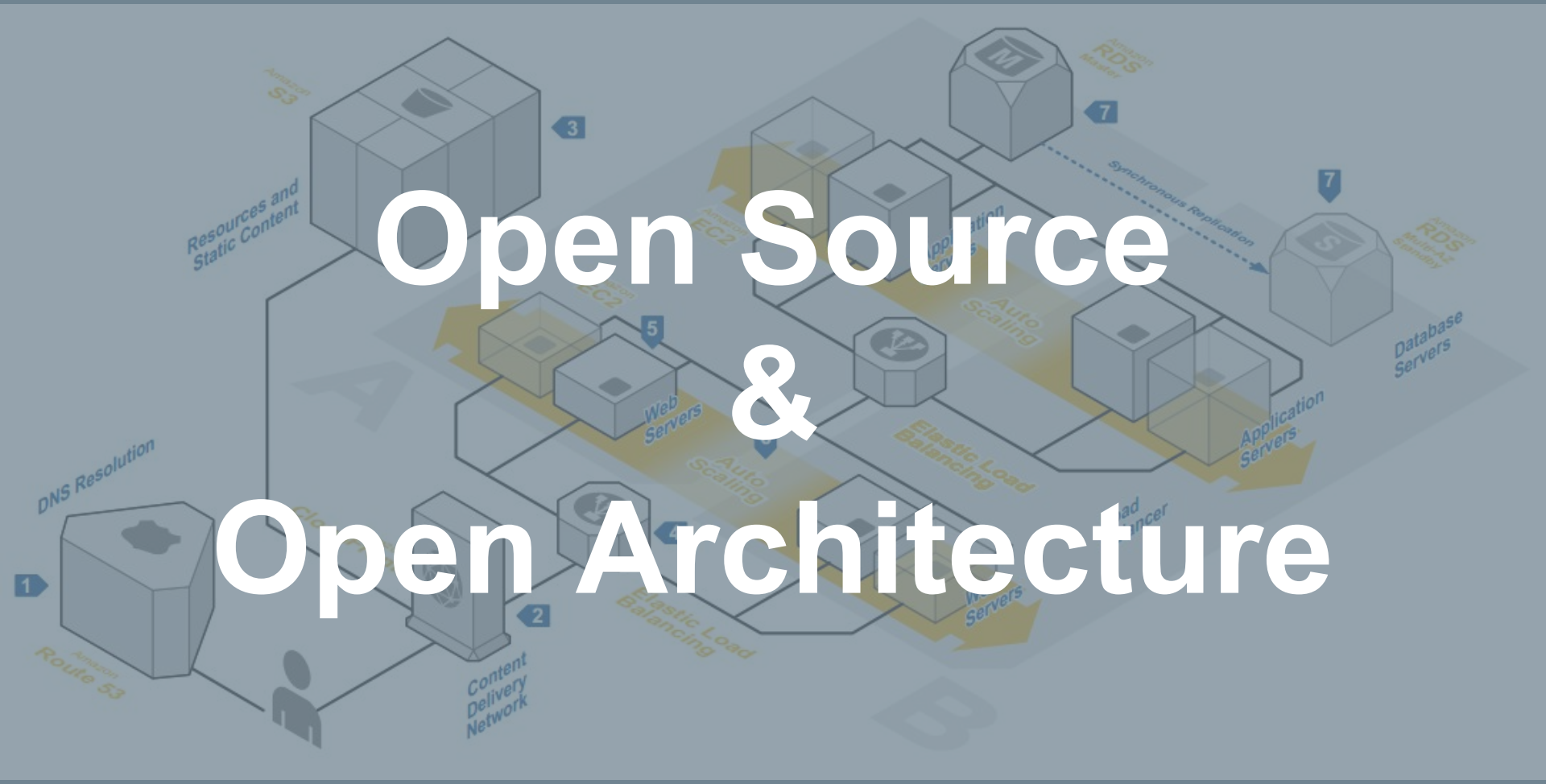
**organizations which design systems** (in the broad sense used here) are constrained to **produce designs which are copies of the communication structures**

of these organizations

Melvin Conway, Datamation, **1968**

http://www.melconway.com/Home/Conways_Law.html



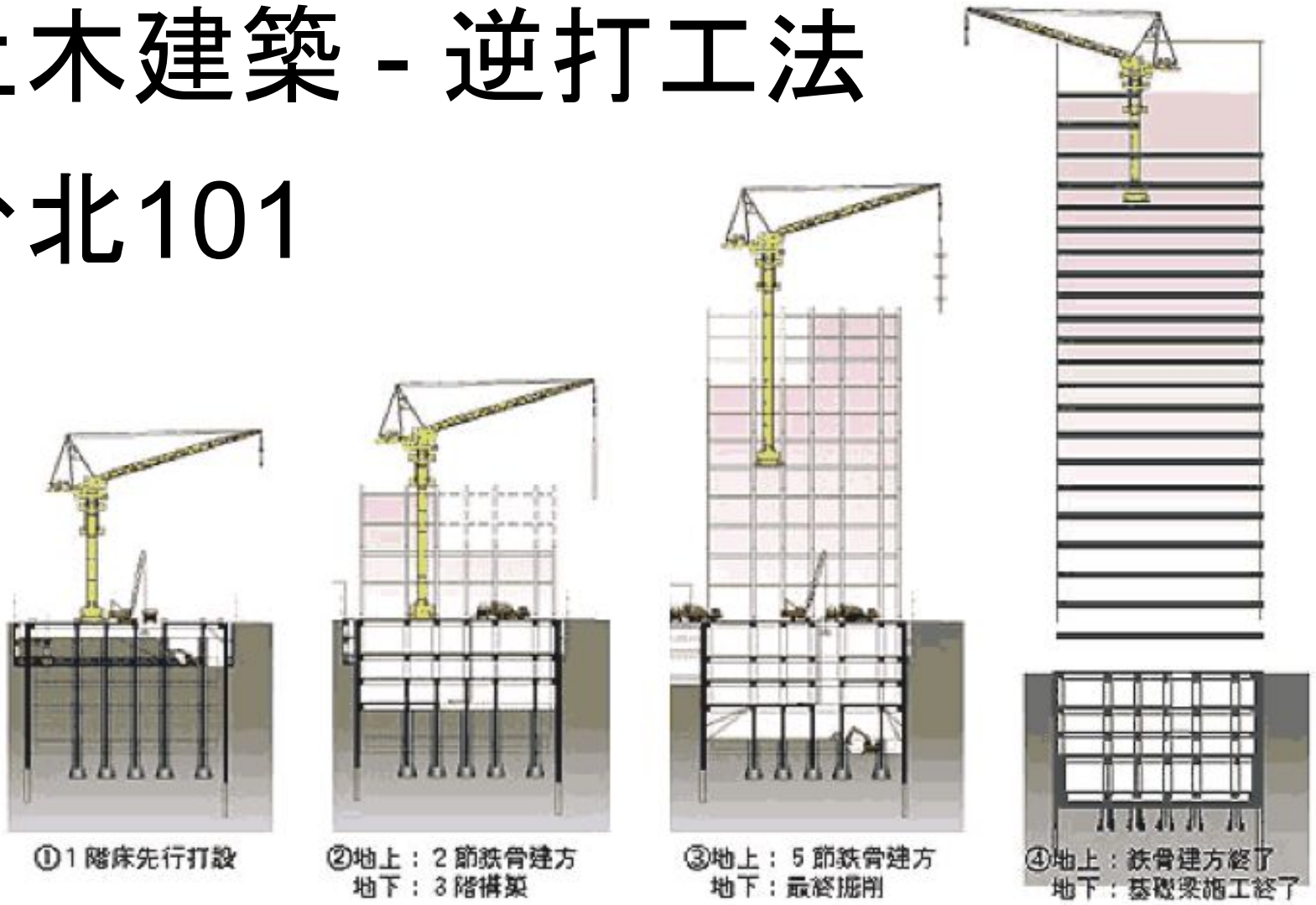



# Open Source & Open Architecture

# Top-down Approach

## 土木建築 - 逆打工法

## 台北101



| | 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 |

# docker

## Online Self-paced exam.

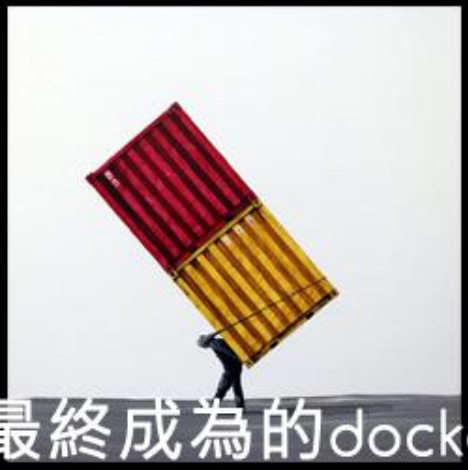
老闆眼中的docker


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

Thanks Microsoft Taiwan provide Azure with Lab environment.

m(_ _)m

# Hope You Love Docker
# So long!