docker容器技術課程

容器基礎(2)

Philipz
鄭淳尹

課程大綱

1. Docker Hub 介紹
2. Docker Hub Auto-build
3. Docker Network 指令
4. Docker Volume 指令
5. Docker Compose 基本指令
6. Docker Compose 實際操作
7. 結語

1. Docker Hub介紹

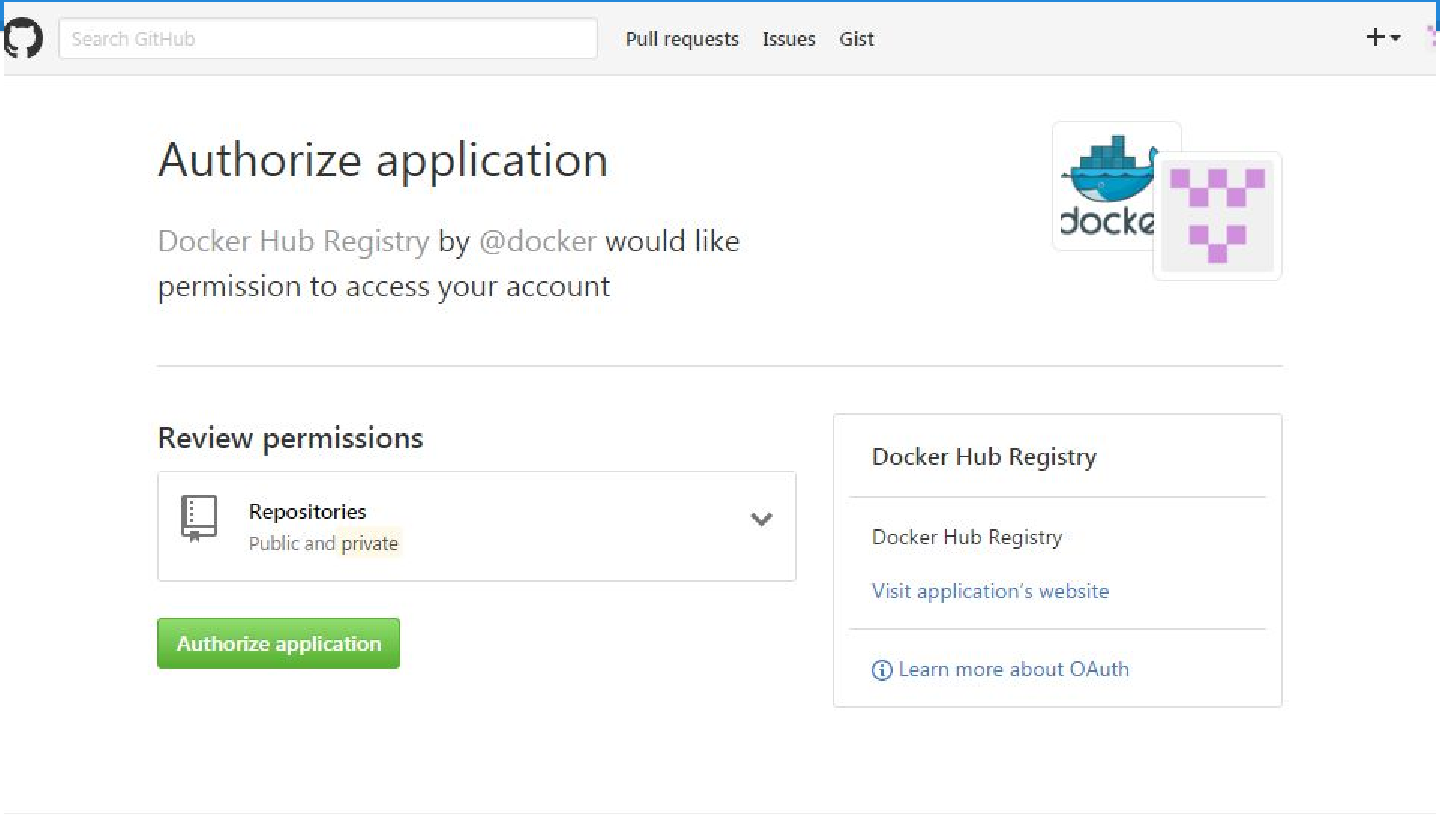
Docker Hub = App Store

- 公開 Docker Registry
- 只允許存放一個私有映像檔
- Auto-build & Webhook
- Security Scanning 是付費功能

Build, Ship, & Run
Any App, Anywhere

Dev-test pipeline automation, 100,000+ free apps, public and private registries

GitHub & Docker Hub



Authorize application

Docker Hub Registry by @docker would like permission to access your account

Review permissions

Repositories
Public and private

Docker Hub Registry

Docker Hub Registry

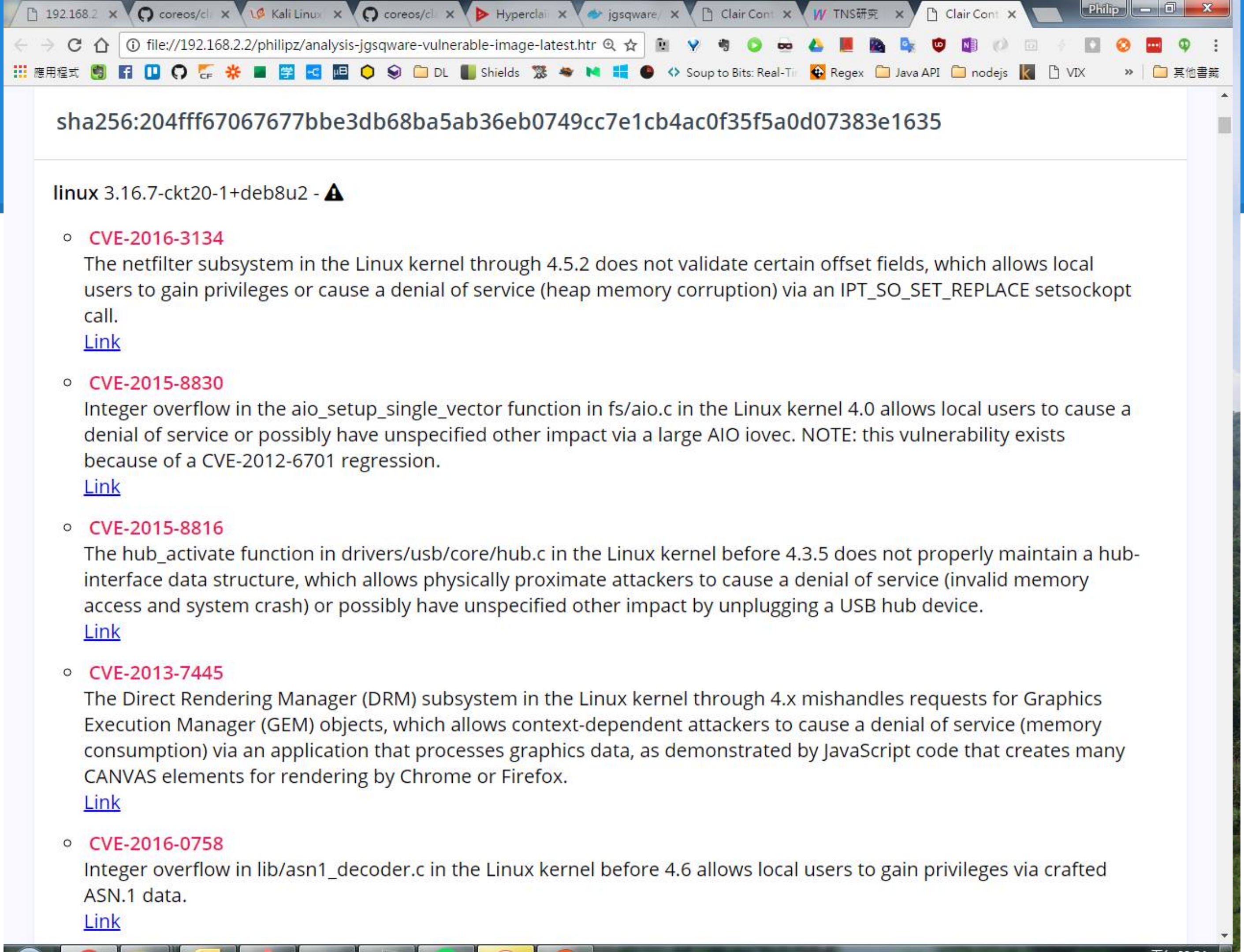
Visit application's website

Learn more about OAuth

Vulnerability Analysis

CoreOS Clair

Anchore



sha256:204fff67067677bbe3db68ba5ab36eb0749cc7e1cb4ac0f35f5a0d07383e1635

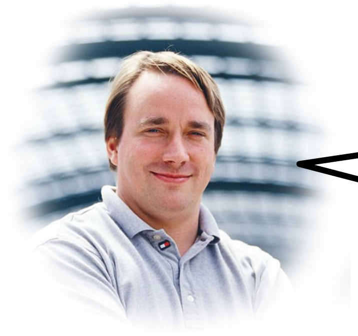
linux 3.16.7-ckt20-1+deb8u2 - ⚠

- CVE-2016-3134
  The netfilter subsystem in the Linux kernel through 4.5.2 does not validate certain offset fields, which allows local users to gain privileges or cause a denial of service (heap memory corruption) via an IPT_SO_SET_REPLACE setsockopt call.
  Link
- CVE-2015-8830
  Integer overflow in the aio_setup_single_vector function in fs/aio.c in the Linux kernel 4.0 allows local users to cause a denial of service or possibly have unspecified other impact via a large AIO iovec. NOTE: this vulnerability exists because of a CVE-2012-6701 regression.
  Link
- CVE-2015-8816
  The hub_activate function in drivers/usb/core/hub.c in the Linux kernel before 4.3.5 does not properly maintain a hub-interface data structure, which allows physically proximate attackers to cause a denial of service (invalid memory access and system crash) or possibly have unspecified other impact by unplugging a USB hub device.
  Link
- CVE-2013-7445
  The Direct Rendering Manager (DRM) subsystem in the Linux kernel through 4.x mishandles requests for Graphics Execution Manager (GEM) objects, which allows context-dependent attackers to cause a denial of service (memory consumption) via an application that processes graphics data, as demonstrated by JavaScript code that creates many CANVAS elements for rendering by Chrome or Firefox.
  Link
- CVE-2016-0758
  Integer overflow in lib/asn1_decoder.c in the Linux kernel before 4.6 allows local users to gain privileges via crafted ASN.1 data.
  Link

2. Docker Hub Auto-build

Git 是必備工具

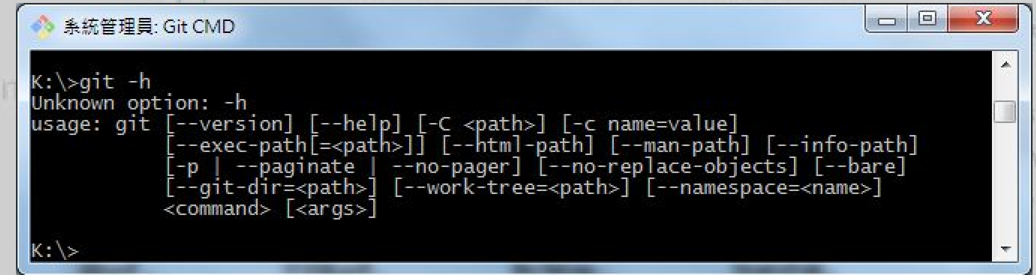
- VCS tool
- Open source community protocol
- GitHub, Bitbucket, GitLab……



Linux 和 Git 都是我搞出來的！

Install Git

- sudo apt-get install git
- Git cmd for windows
- SourceTree is best choice!
- GitHub is a git web-UI and repository.
- Git 教室



```
K:\>git -h
Unknown option: -h
usage: git [--version] [--help] [-C <path>] [-c name=value]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           <command> [<args>]

K:\>
```

Dockerfile

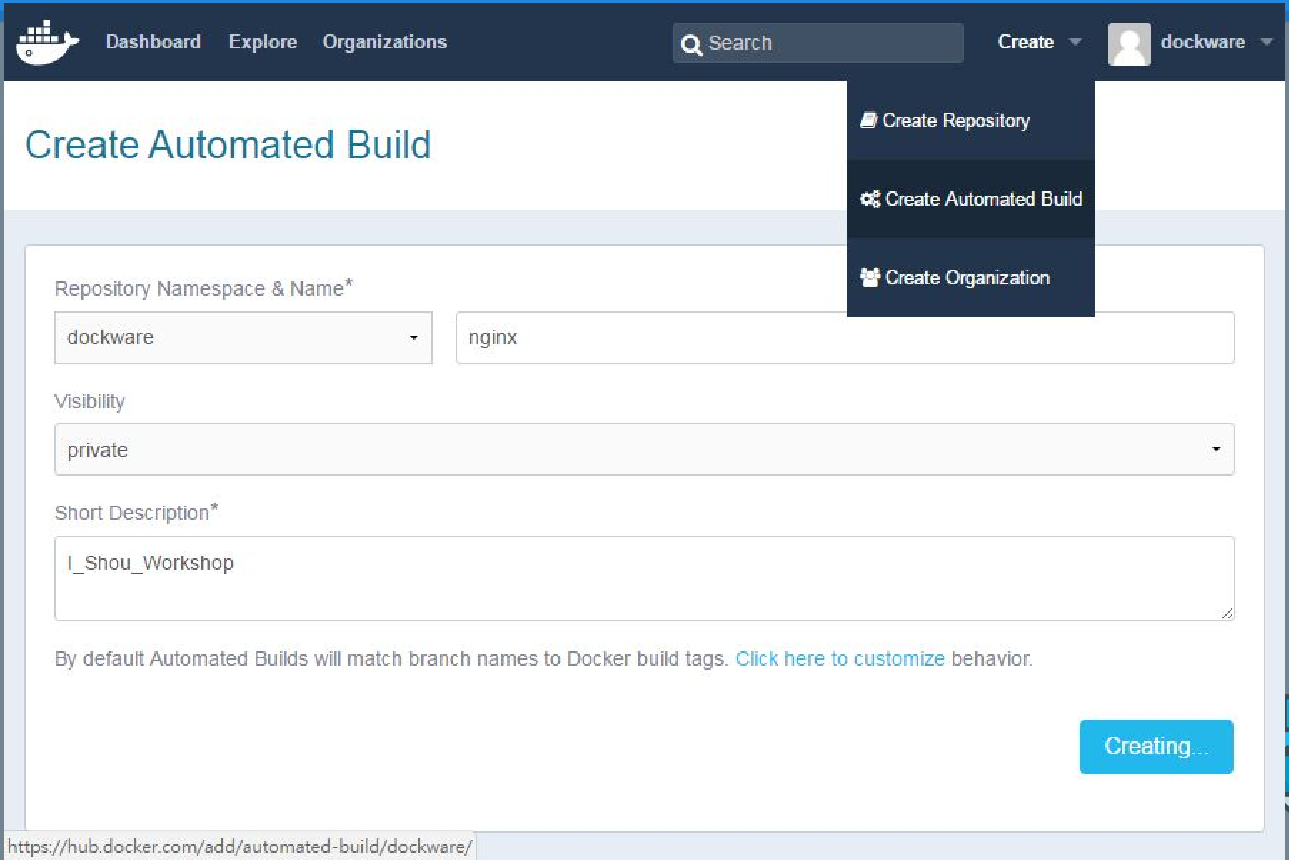
範例 :

```
FROM debian:jessie
MAINTAINER docker "docker@nginx.com"
RUN apt-get update && apt-get install -y nginx
CMD ["nginx", "-g", "daemon off;"]
```

Git 操作流程

1. git init or init on GitHub.
2. git add Dockerfile
3. git commit -m “First init”
4. git remote add origin https://github.com/YOURNAME/docker_build.git
5. git push origin master

建立 Auto-build Repo.



Create Automated Build

Repository Namespace & Name*

dockware

nginx

Visibility

private

Short Description*

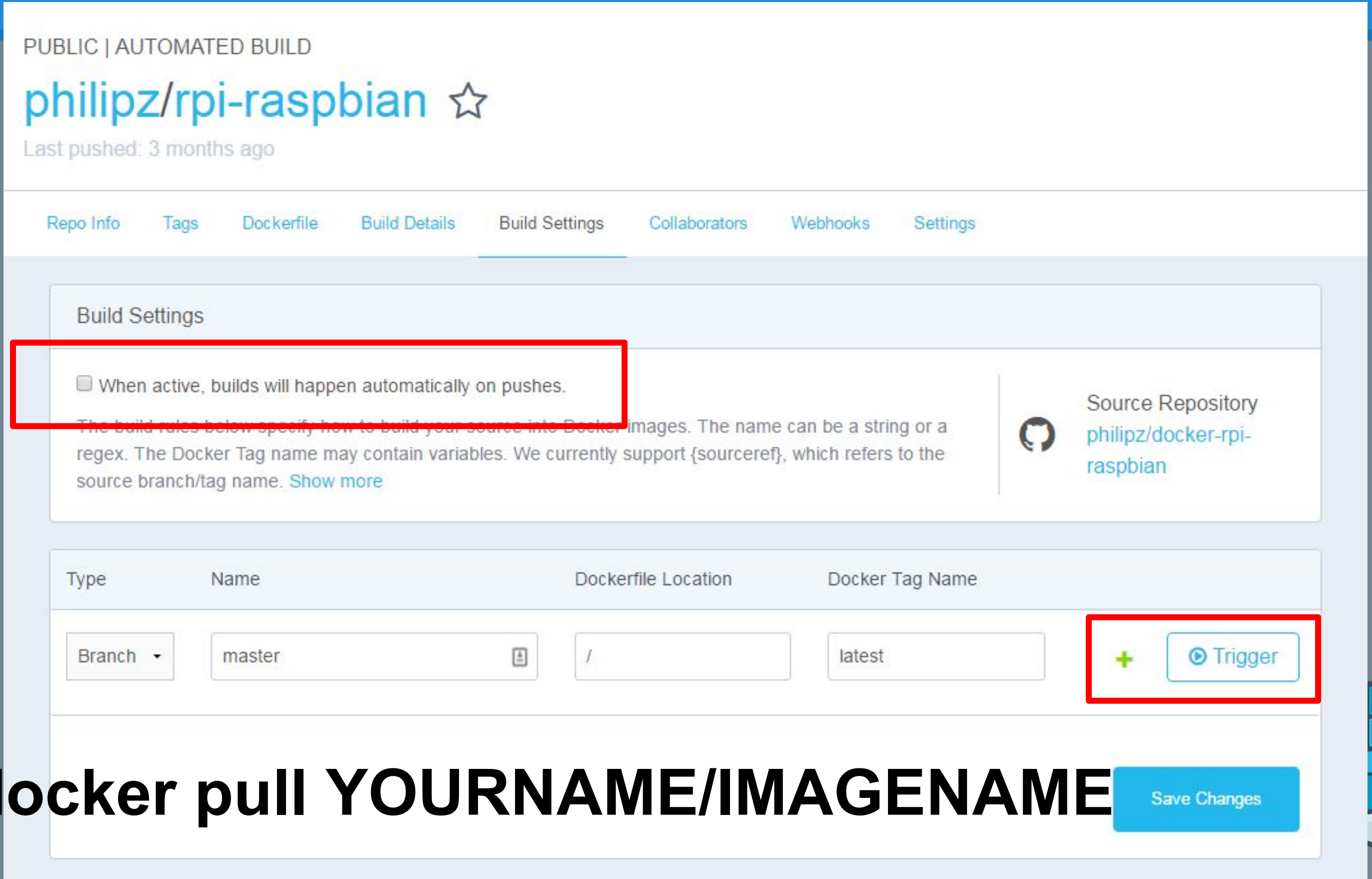
I_Shou_Workshop

By default Automated Builds will match branch names to Docker build tags. Click here to customize behavior.

Creating...

https://hub.docker.com/add/automated-build/dockware/

建置設定



PUBLIC | AUTOMATED BUILD

philipz/rpi-raspbian ☆

Last pushed: 3 months ago

Build Settings

□ When active, builds will happen automatically on pushes.

The build rules below specify how to build your source into Docker images. The name can be a string or a regex. The Docker Tag name may contain variables. We currently support {sourceref}, which refers to the source branch/tag name. Show more

Source Repository
philipz/docker-rpi-raspbian

Type	Name	Dockerfile Location	Docker Tag Name
Branch	master	/	latest

docker pull YOURNAME/IMAGENAME

3. Docker Network 指令

TCP/IP Foundation

www.google.com, www 是 hostname, google.com 是 domain name.

Localhost: 127.0.0.1

TCP/UDP Port: 0-65535 = 2^16,

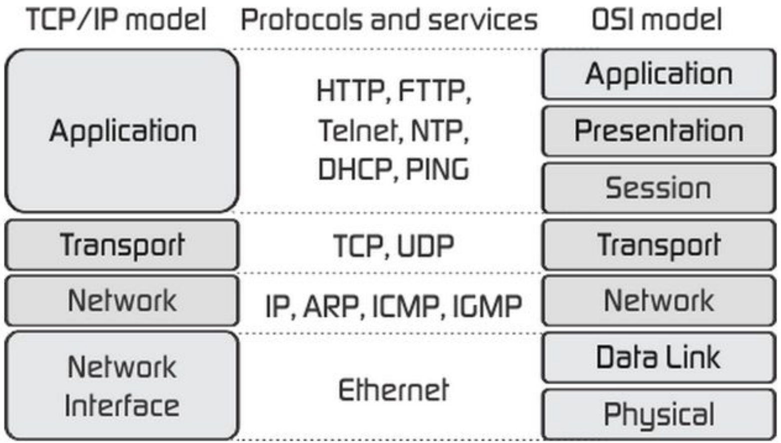
but 0 是保留不可使用的連接埠

Private IP:

10.0.0.0/8

172.16.0.0/12 ~ 172.31.0.0/12

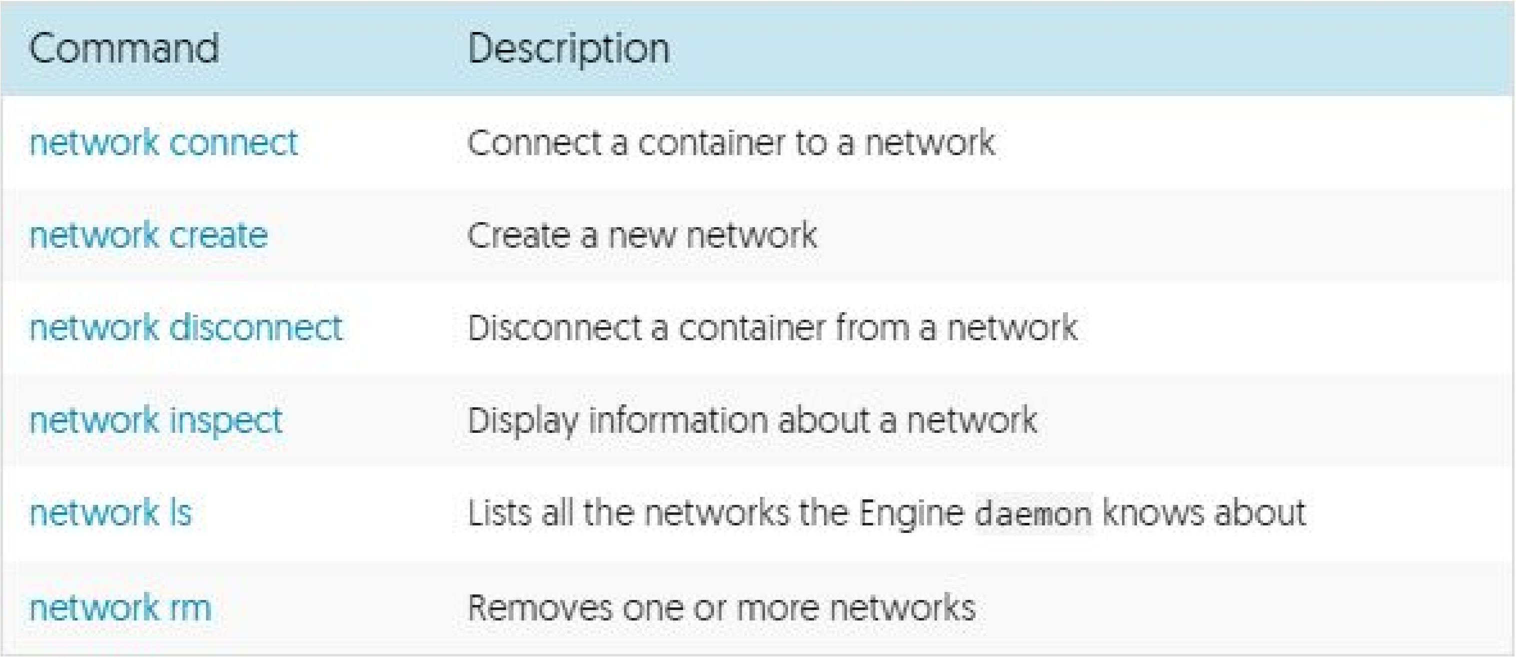
192.168.0.0/16



TCP/IP model	Protocols and services	OSI model
Application	HTTP, FTTP, Telnet, NTP, DHCP, PING	Application Presentation Session
Transport	TCP, UDP	Transport
Network	IP, ARP, ICMP, IGMP	Network
Network Interface	Ethernet	Data Link Physical

Network 相關指令

https://docs.docker.com/engine/userguide/networking/



Command	Description
network connect	Connect a container to a network
network create	Create a new network
network disconnect	Disconnect a container from a network
network inspect	Display information about a network
network ls	Lists all the networks the Engine daemon knows about
network rm	Removes one or more networks

Docker 內建 Network Drivers

- Bridge
- Overlay
- MACVLAN
- Host
- None

Docker Plug-In Network Drivers

- weave
- calico

Docker Plug-In IPAM Drivers

- infoblox

不要再使用 “link”, 改用 network.

Docker Reference Architecture: Designing Scalable, Portable Docker Container Networks

練習一

```
$ docker network ls
$ ifconfig
$ docker run -ti --rm busybox sh
      cat /etc/hosts, ifconfig
$ docker network inspect bridge
$ docker run -itd --name=container1 busybox
$ docker run -itd --name=container2 busybox
$ docker exec -ti container2 sh
      ping -w3 172.17.0.2, ping container1
```

練習二

```
$ docker network create vlan_1
$ docker network inspect vlan_1
$ ifconfig | more
$ docker run --network=vlan_1 -itd --name=container3 busybox
$ docker network inspect vlan_1
$ docker run --network=vlan_1 -itd --name=container4 busybox
$ docker exec -ti container4 sh
    ping -w3 172.17.0.2, ping container1, ping container3
```

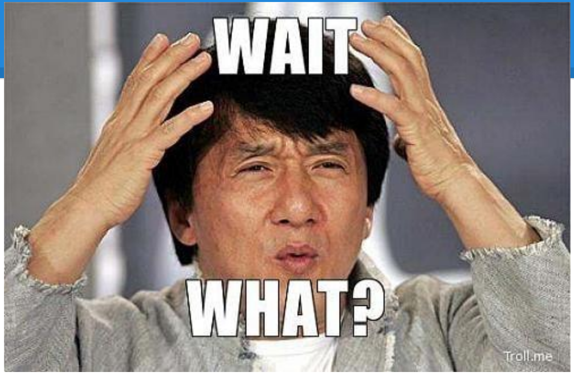
練習三

```
$ docker network create wp_db
$ docker pull mysql:5.7
$ docker pull wordpress
```

```
$ docker run -d --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 run -d --name wp -p 80:80 --network=wp_db
        -e WORDPRESS_DB_HOST=db:3306
        -e WORDPRESS_DB_PASSWORD=wordpress
        wordpress
```

練習四



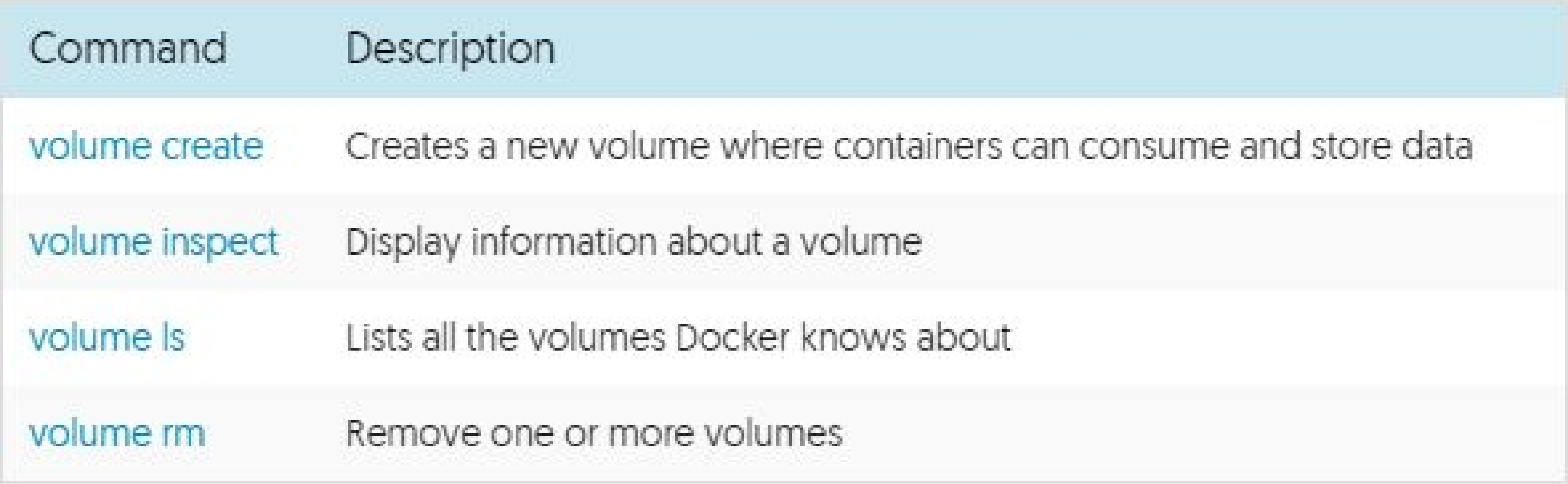
```
$ docker network create -d macvlan
    --subnet=10.0.0.0/24
    --gateway=10.0.0.1
    -o parent=eth0 mvnet
$ docker run -itd --name c1 --net mvnet --ip 10.0.0.5 busybox
$ docker run -it --name c2 --net mvnet --ip 10.0.0.6 busybox sh
     ping -c 4 10.0.0.5
     ip a show eth0, ip route
$ docker run -d --name --net mvnet --ip 10.0.0.7 nginx
```

Get started with Macvlan network driver

4. Docker Volume 指令

Shared data volume commands

Manage data in containers



Command	Description
volume create	Creates a new volume where containers can consume and store data
volume inspect	Display information about a volume
volume ls	Lists all the volumes Docker knows about
volume rm	Remove one or more volumes

Exercise

```
$ docker volume create \
    --name composewp_db_data

$ docker pull mysql:5.7

$ docker pull wordpress

$ docker run -d --name db --network=wp_db
    -e MYSQL_ROOT_PASSWORD=wordpress
    -e MYSQL_DATABASE=wordpress
    -e MYSQL_USER=wordpress
    -e MYSQL_PASSWORD=wordpress
    -v composewp_db_data:/var/lib/mysql
    mysql:5.7

$ docker run -d --name wp -p 80:80 --network=wp_db
    -e WORDPRESS_DB_HOST=db:3306
    -e WORDPRESS_DB_PASSWORD=wordpress
    wordpress
```

vSphere Docker Volume Plugin

https://github.com/vmware/docker-volume-vsphere

```
$ docker volume create --driver=vsphere
--name=ESXVolume -o sze=5gb
```

```
$ docker run -ti --name u1 --rm -v ESXVolume:/data ubuntu
```

AWS EFS、Azure File Service

Use Docker Engine plugins

5. Docker Compose 基本指令

安裝 Docker Compose

```
sudo curl -L "https://github.com/docker/compose/releases/download/1.14.0/docker-compose-$(uname -s)-$(uname -m)" -o \
/usr/local/bin/docker-compose
```

然後

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

```
docker-compose -v
```

Docker Compose 指令 (1/2)

Commands:

build	Build or rebuild services
bundle	Generate a Docker bundle from the Compose file
config	Validate and view the compose file
create	Create services
down	Stop and remove containers, networks, images, and volumes
events	Receive real time events from containers
exec	Execute a command in a running container
help	Get help on a command
kill	Kill containers
logs	View output from containers
pause	Pause services
port	Print the public port for a port binding

Docker Compose 指令 (2/2)

Commands:

ps	List containers
pull	Pull service images
push	Push service images
restart	Restart services
rm	Remove stopped containers
run	Run a one-off command
scale	Set number of containers for a service
start	Start services
stop	Stop services
unpause	Unpause services
up	Create and start containers
version	Show the Docker-Compose version information

Compose 檔案說明

一次執行多個容器，建構完整服務

必須是 docker-compose.yml

相同目錄：docker-compose up -d

Docker 會自動建置包含 Dockerfile 的子目錄

支援 Docker Network, Volume

1.13 版本支援 Swarm mode.

Quickstart: Compose and WordPress

6. Docker Compose 實際操作

Compose File Sample (1/2)

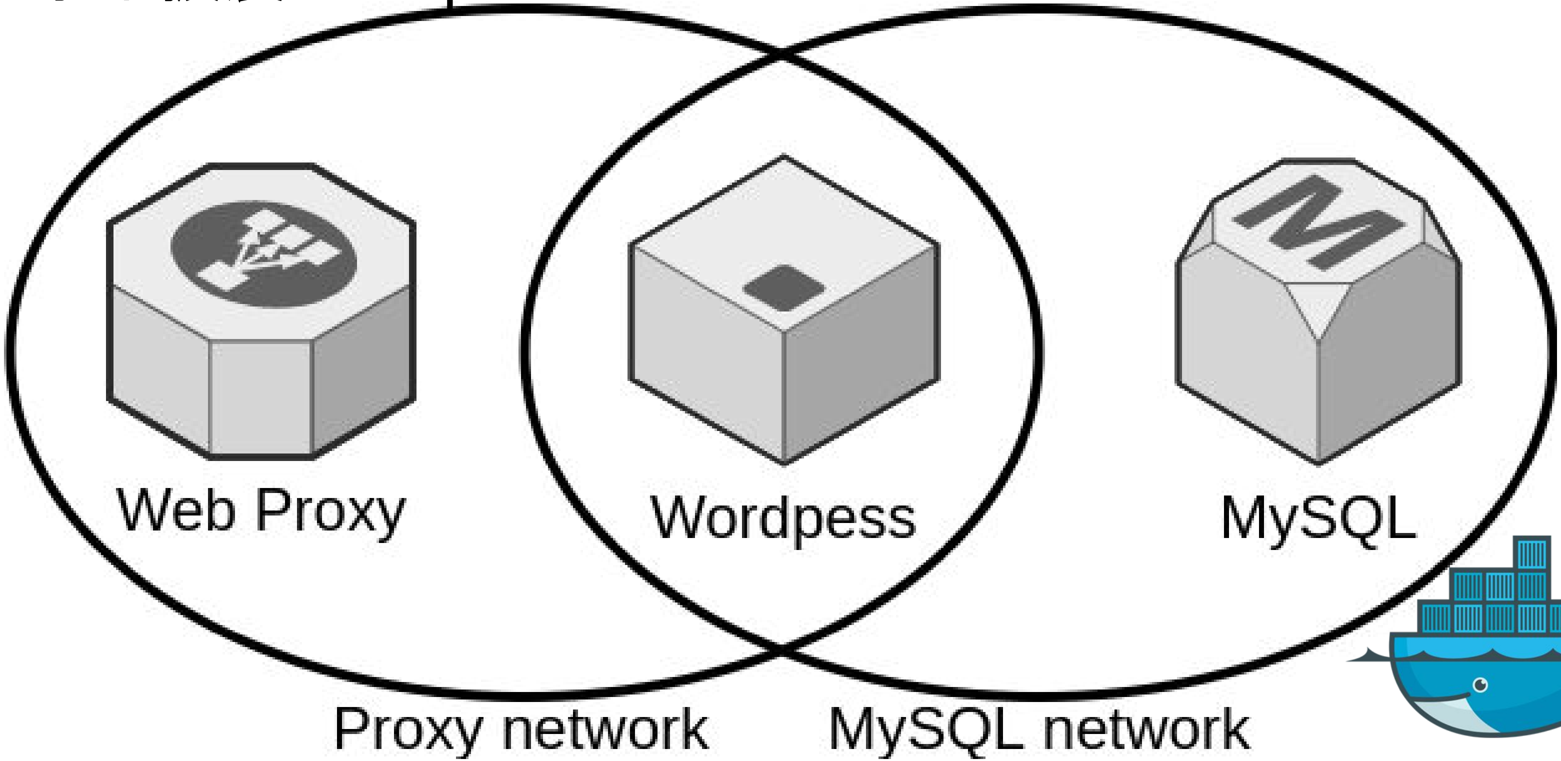
```
version: '2'
services:
  db:
    image: mysql:5.7
    volumes:
      - db_data:/var/lib/mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: wordpress
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: wordpress
```

Compose File Sample (1/2)

```
  wordpress:
    depends_on:
      - db
    image: wordpress:latest
    ports:
      - "80:80"
    restart: always
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_PASSWORD: wordpress
volumes:
  db_data:
```

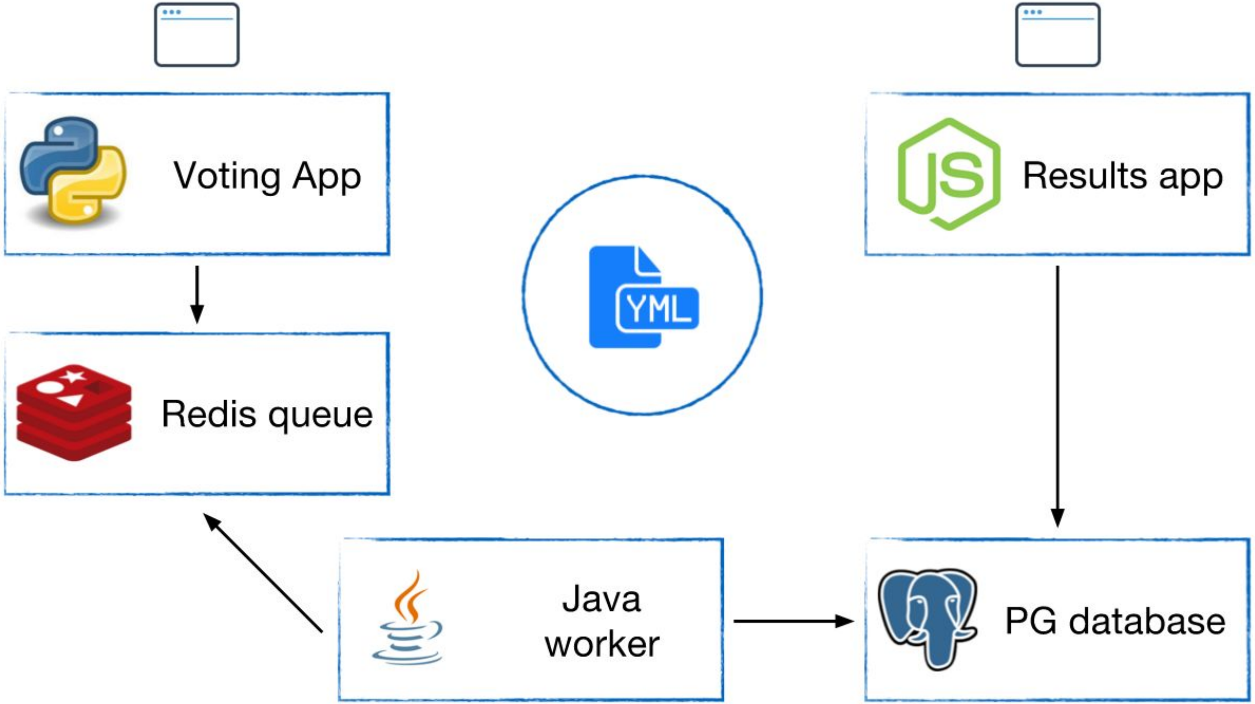
Compose & Wordpress

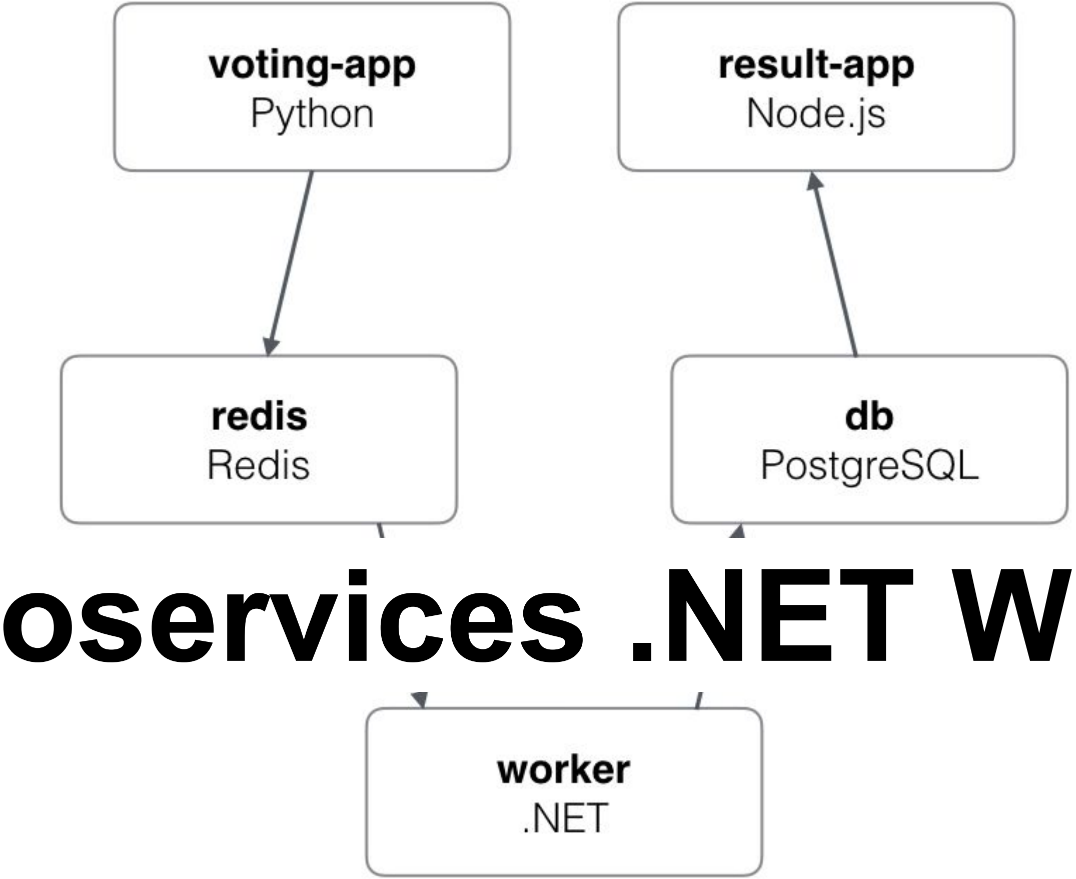
- 水平擴展 wordpress：scale



Microservices Java Worker

[Docker Birthday #3 training]





Microservices .NET Worker

[Docker Birthday #3 training]

7. 結語

Still No Silver Bullet

容器只是其中一個關鍵，並非全部

DevOps pipeline 軟體開發流程

Microservices微服務，或其他架構

Infrastructure as Code

Business model

The Docker Stack

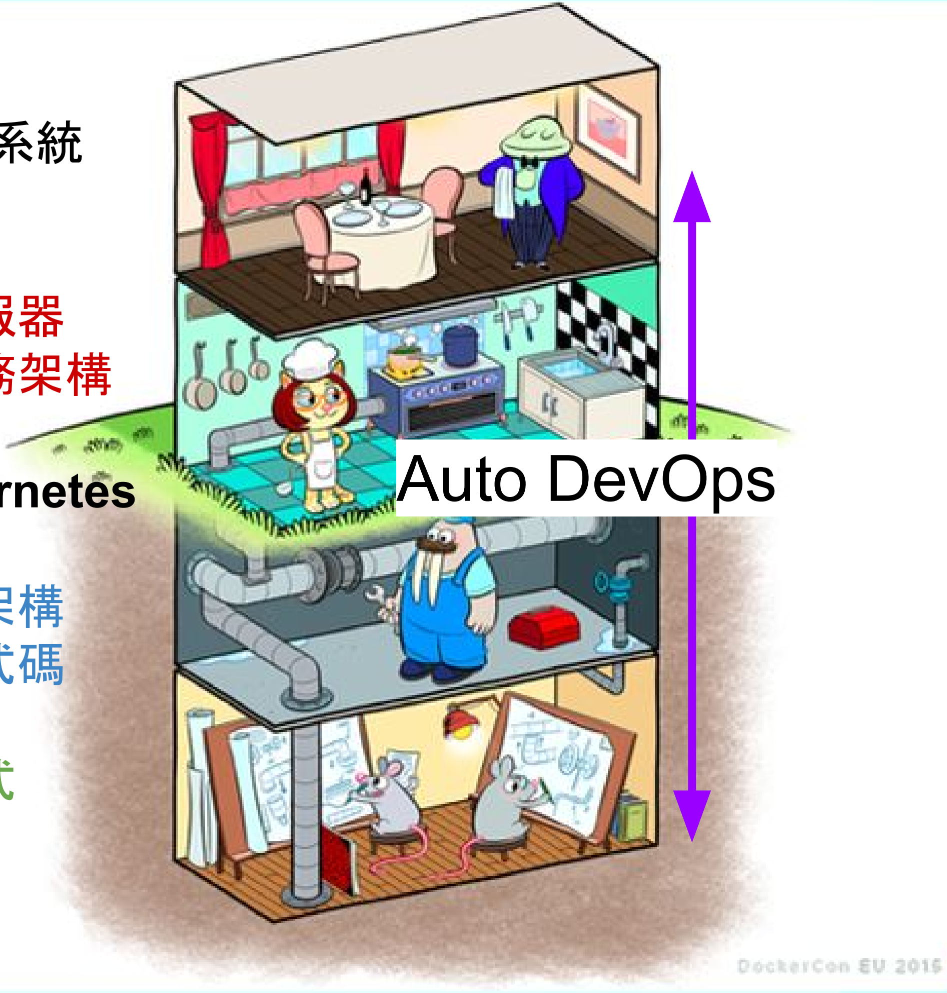
*業務系統

無伺服器
微服務架構

Kubernetes

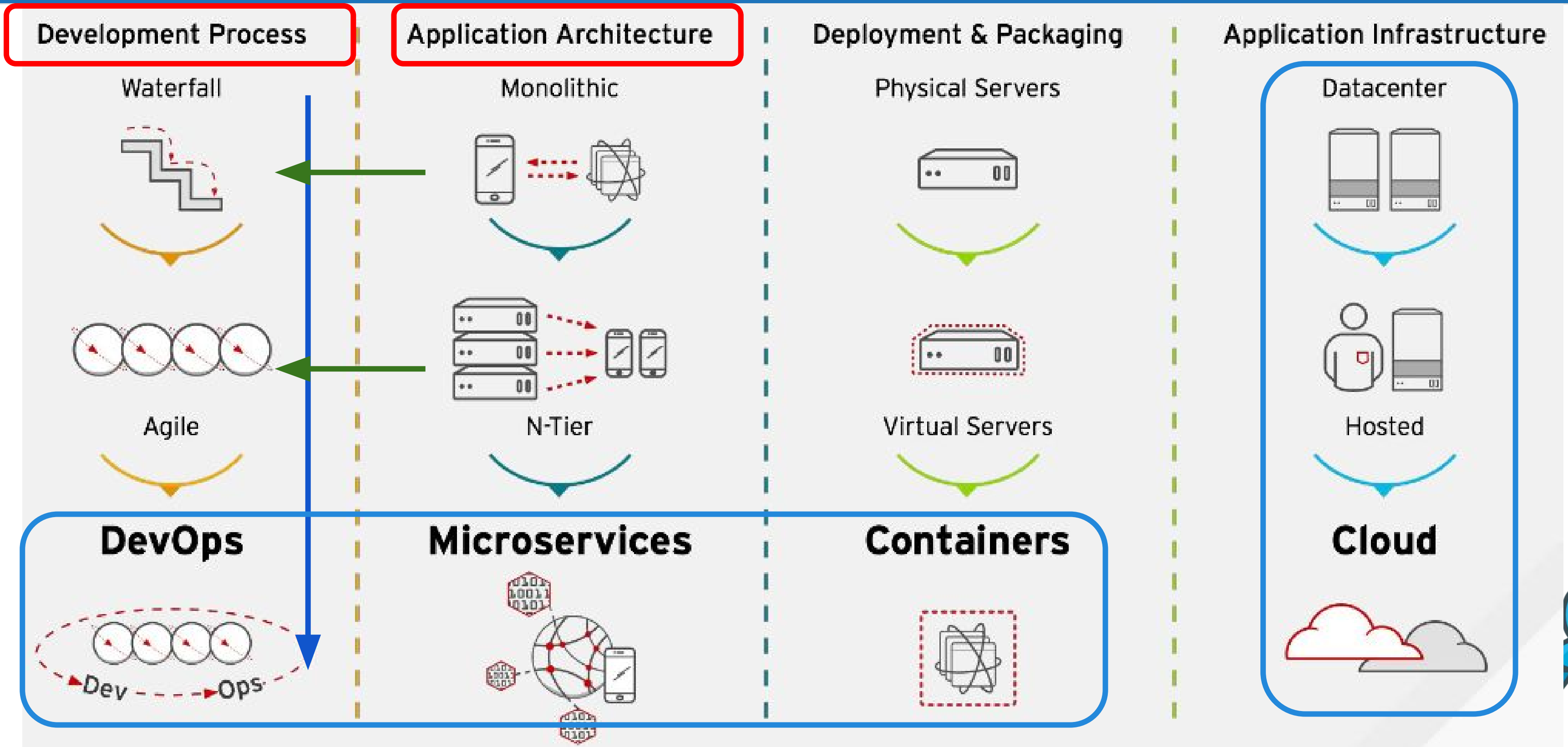
基礎架構
即程式碼

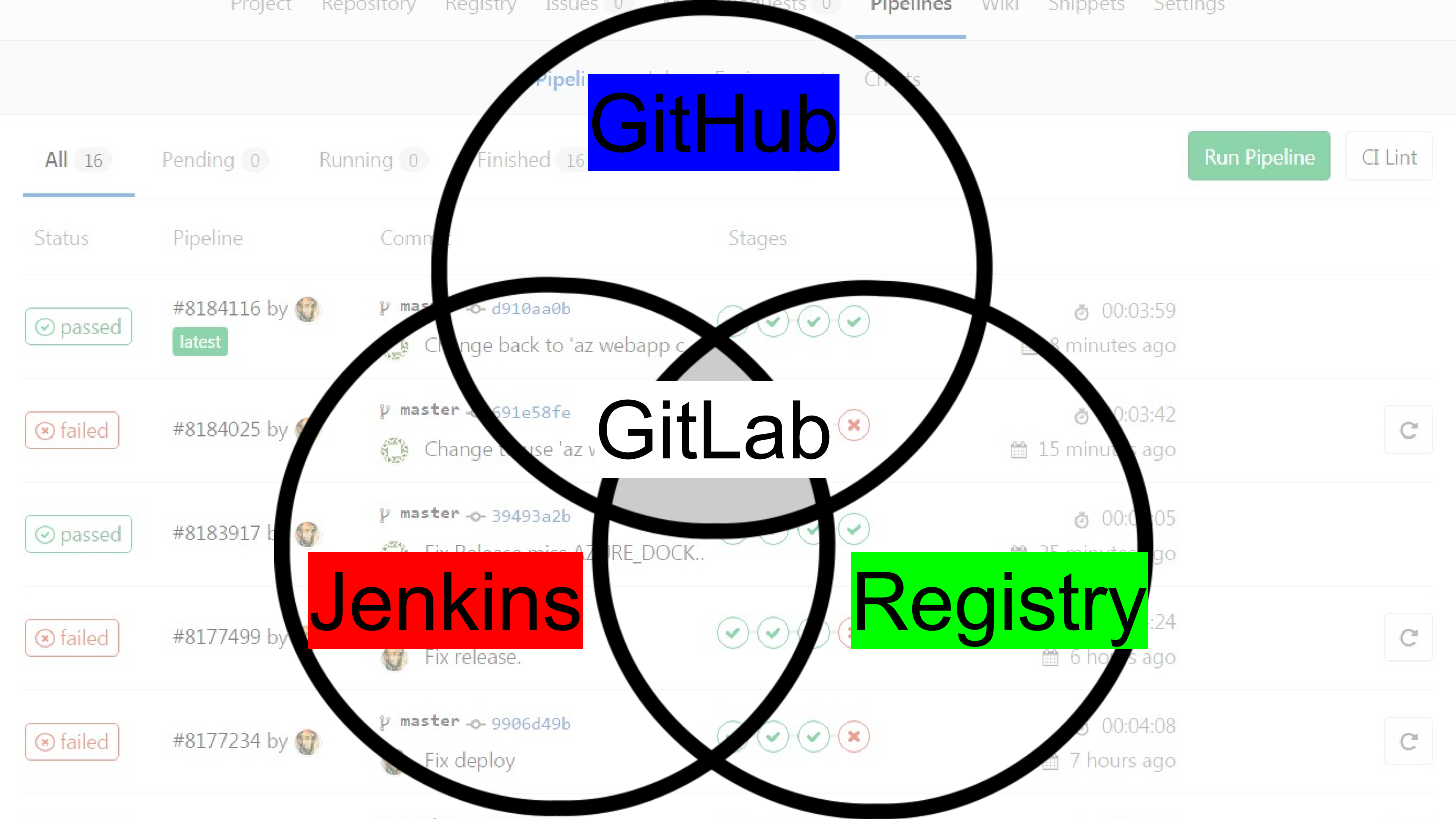
容器式
設計



Auto DevOps

容器式系統架構


Development Process
Waterfall
Agile
DevOps
Dev
Ops
Application Architecture
Monolithic
N-Tier
Microservices
Deployment & Packaging
Physical Servers
Virtual Servers
Containers
Application Infrastructure
Datacenter
Hosted
Cloud



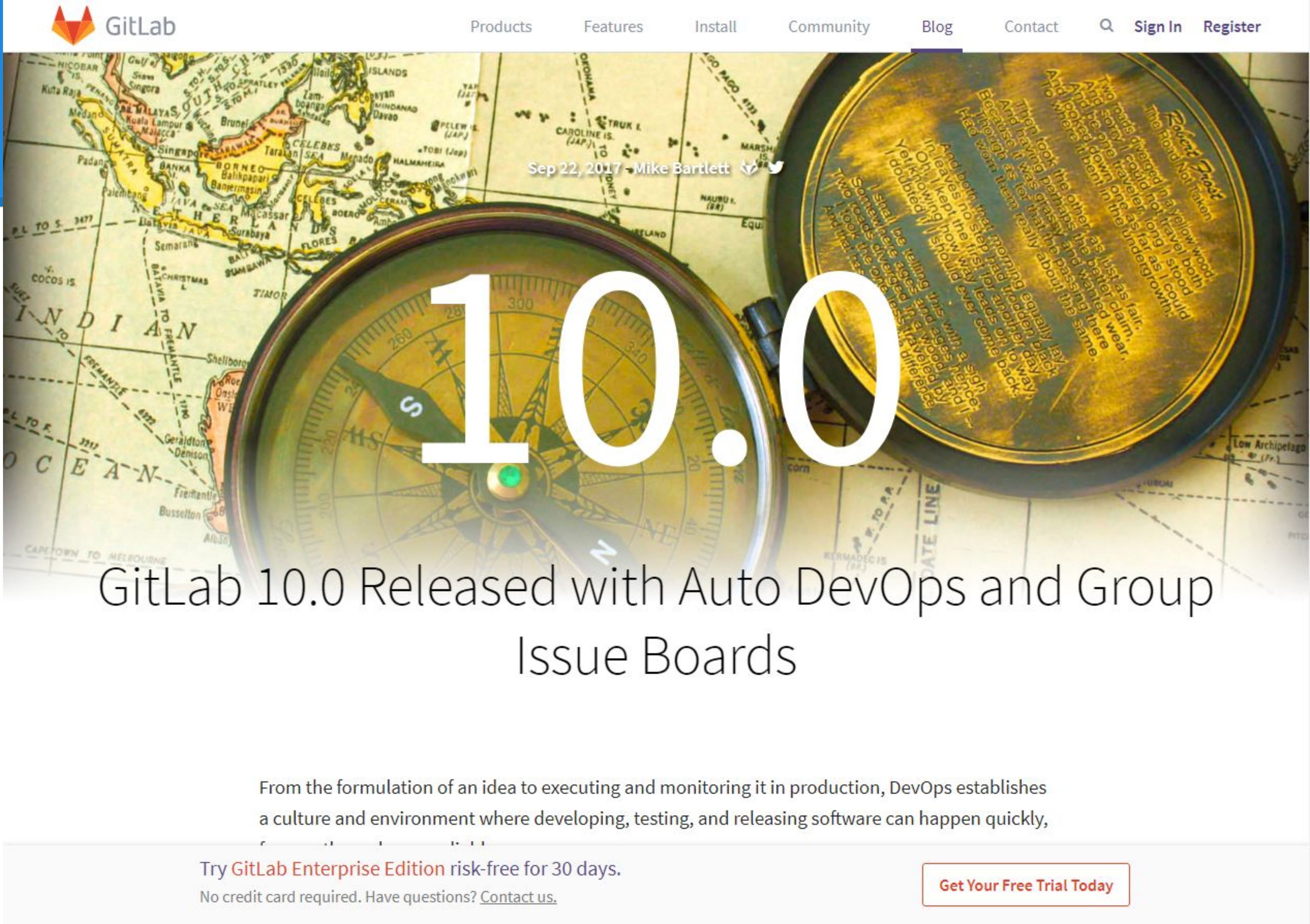
GitHub

GitLab

Jenkins

Registry

Status	Pipeline	Commit	Stages	
passed	#8184116 by latest	mas d910aa0b Change back to 'az webapp c	✓ ✓ ✓ ✓	00:03:59 8 minutes ago
failed	#8184025 by	master 691e58fe Change to use 'az w	✗	0:03:42 15 minutes ago
passed	#8183917	master 39493a2b Fix Release miss AZURE_DOCK..	✓ ✓ ✓ ✓	00:0 05 25 minutes ago
failed	#8177499 by	Fix release.	✓ ✓ ✓	:24 6 hours ago
failed	#8177234 by	master 9906d49b Fix deploy	✓ ✓ ✓ ✗	00:04:08 7 hours ago



Sep 22, 2017 · Mike Bartlett

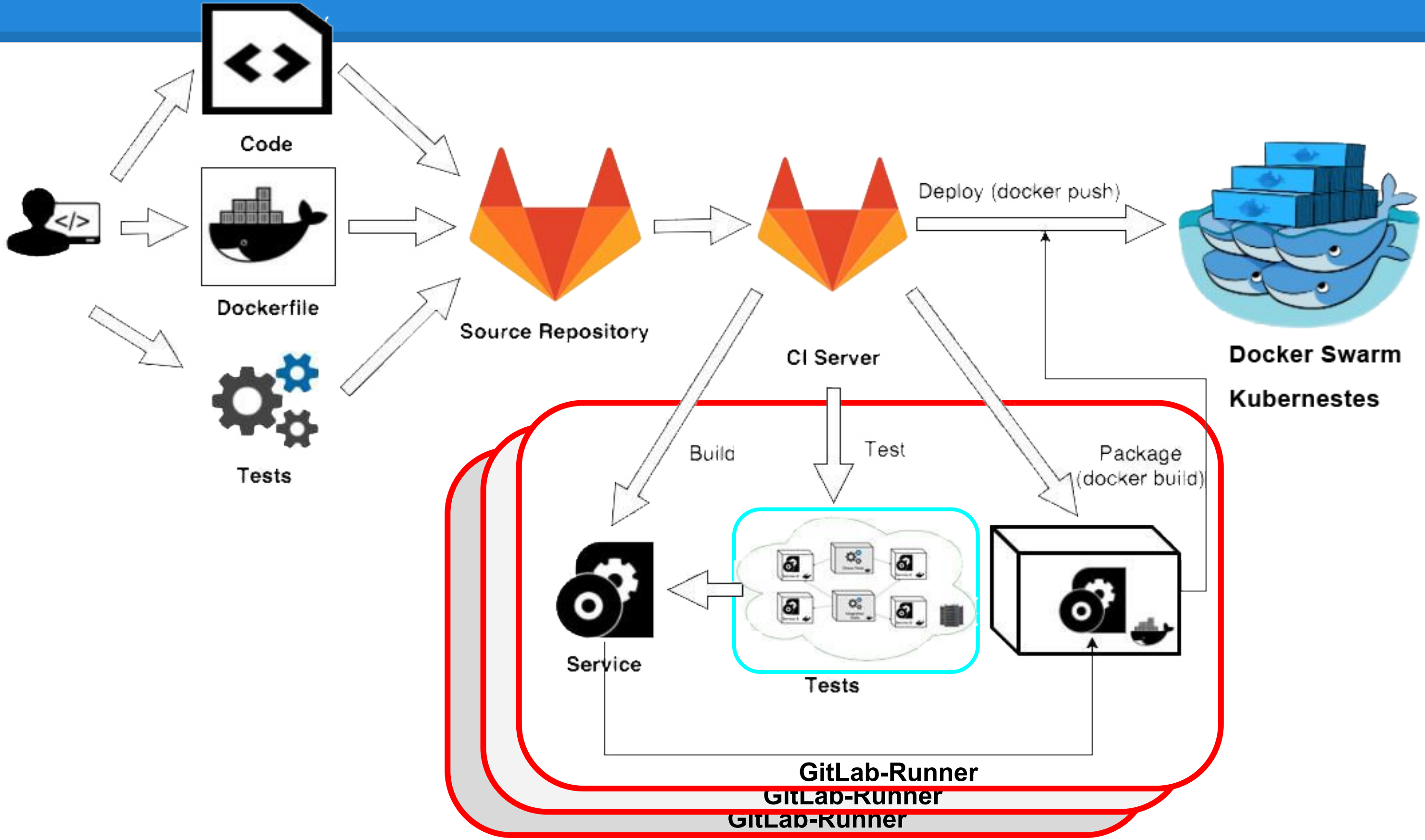
10.0

GitLab 10.0 Released with Auto DevOps and Group Issue Boards

From the formulation of an idea to executing and monitoring it in production, DevOps establishes a culture and environment where developing, testing, and releasing software can happen quickly,

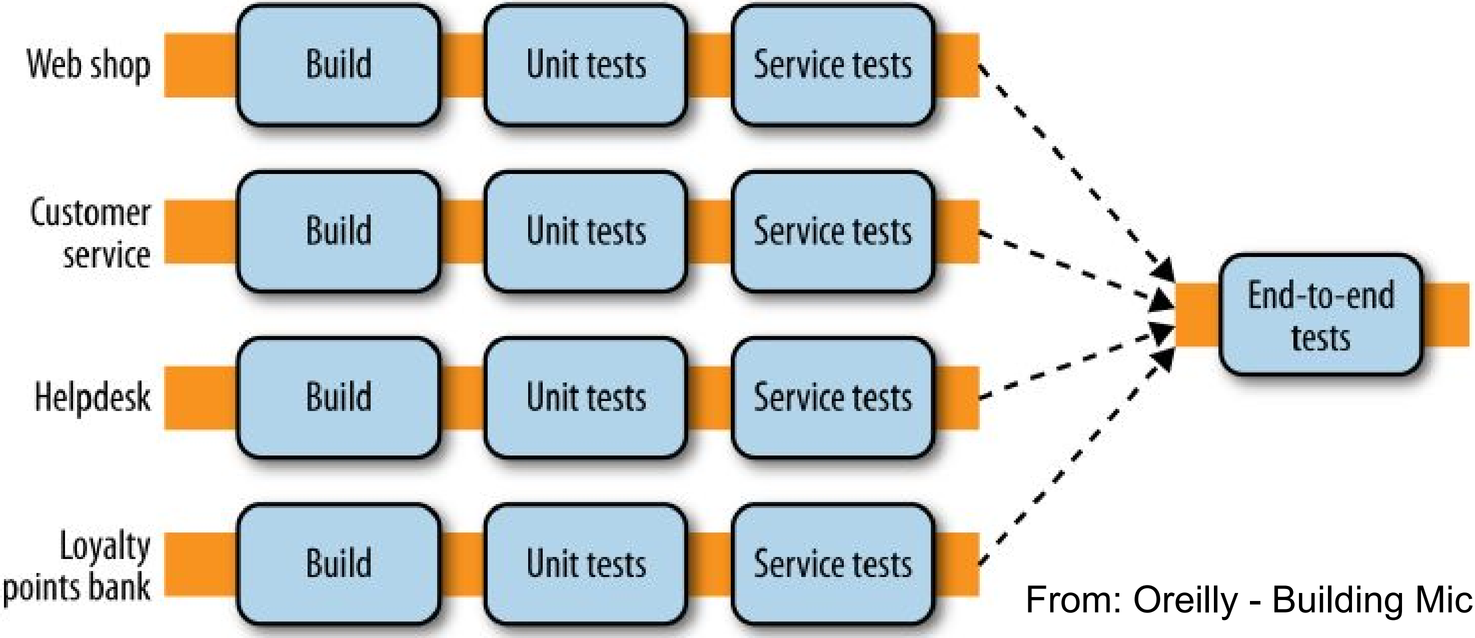
Try GitLab Enterprise Edition risk-free for 30 days.
No credit card required. Have questions? Contact us.

容器開發流程


Code
Dockerfile
Tests
Source Repository
CI Server
Deploy (docker push)
Docker Swarm
Kubernestes
Build
Test
Package
(docker build)
Service
Tests
GitLab-Runner
GitLab-Runner
GitLab-Runner

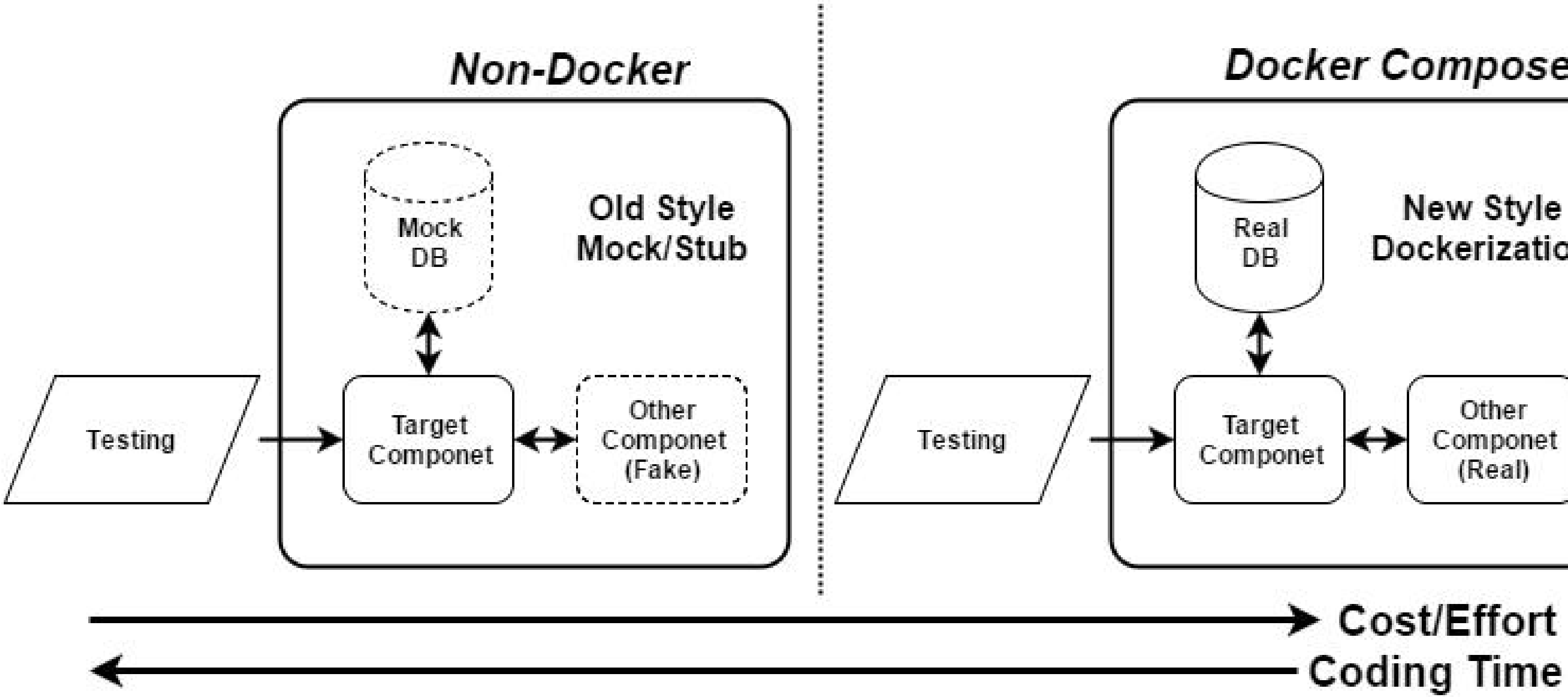
End to End Tests

CI with Docker Compose is easy to implement.

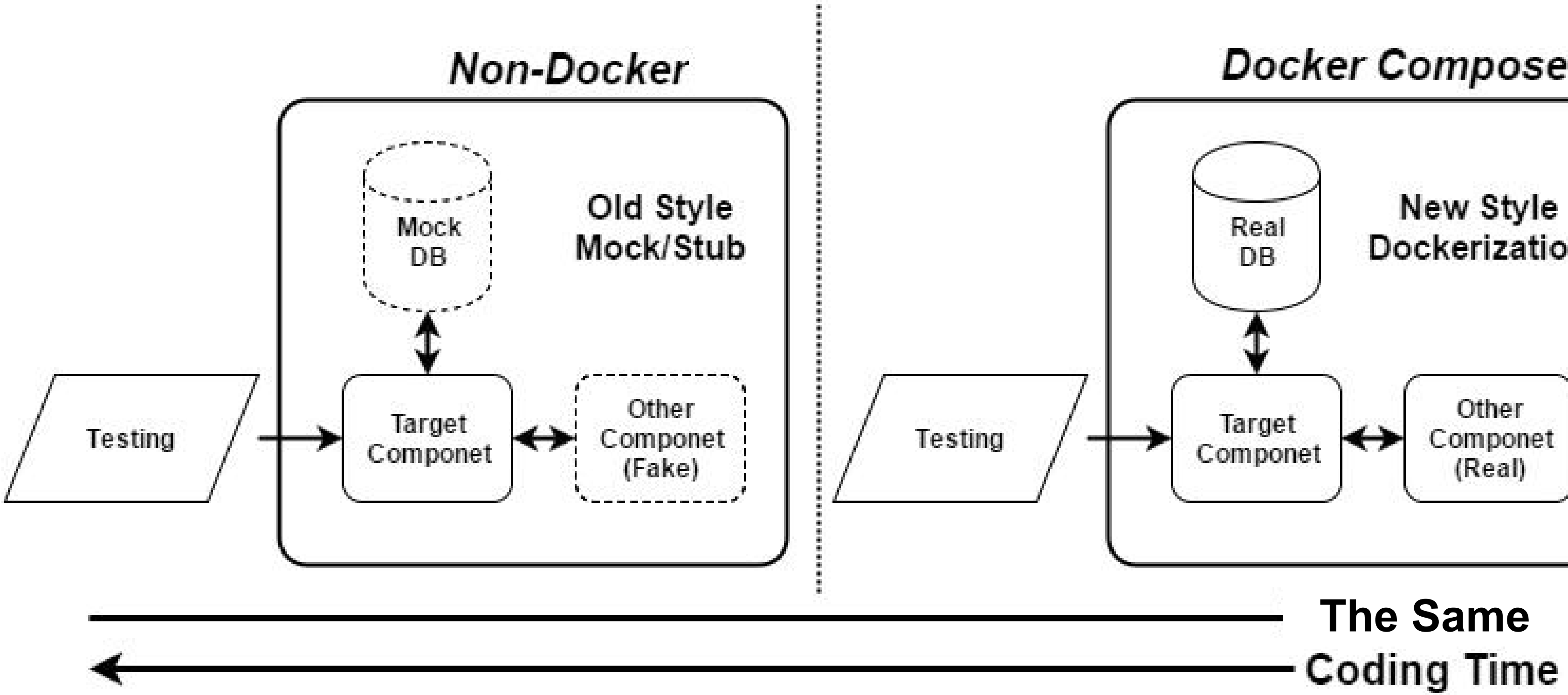


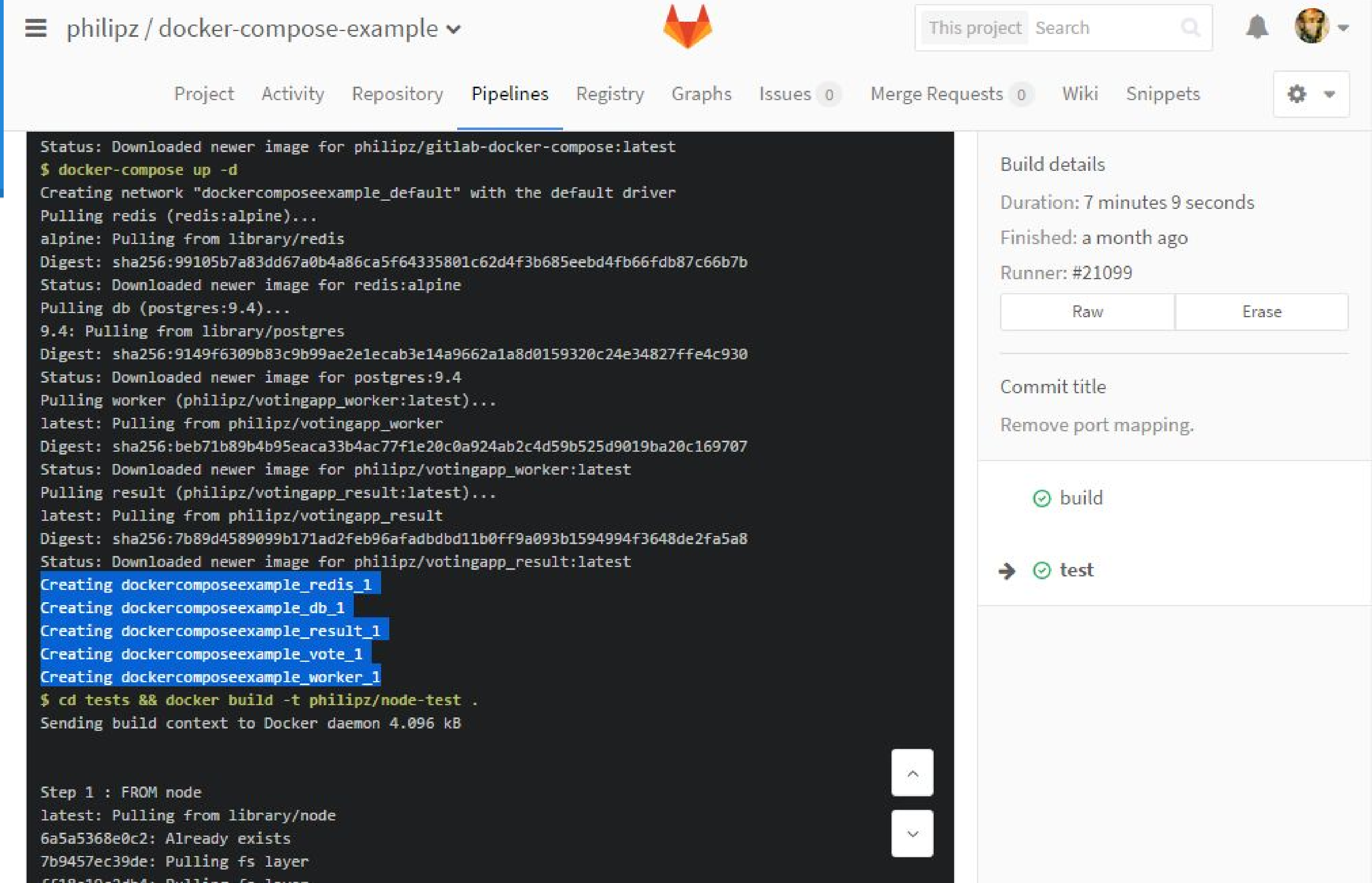
From: Oreilly - Building Microservices

Test Double 測試方法



新 Compose 測試方法





```
Status: Downloaded newer image for philipz/gitlab-docker-compose:latest
$ docker-compose up -d
Creating network "dockercomposeexample_default" with the default driver
Pulling redis (redis:alpine)...
alpine: Pulling from library/redis
Digest: sha256:99105b7a83dd67a0b4a86ca5f64335801c62d4f3b685eebd4fb66fdb87c66b7b
Status: Downloaded newer image for redis:alpine
Pulling db (postgres:9.4)...
9.4: Pulling from library/postgres
Digest: sha256:9149f6309b83c9b99ae2e1ecab3e14a9662a1a8d0159320c24e34827ffe4c930
Status: Downloaded newer image for postgres:9.4
Pulling worker (philipz/votingapp_worker:latest)...
latest: Pulling from philipz/votingapp_worker
Digest: sha256:beb71b89b4b95eaca33b4ac77f1e20c0a924ab2c4d59b525d9019ba20c169707
Status: Downloaded newer image for philipz/votingapp_worker:latest
Pulling result (philipz/votingapp_result:latest)...
latest: Pulling from philipz/votingapp_result
Digest: sha256:7b89d4589099b171ad2feb96afadbdbd11b0ff9a093b1594994f3648de2fa5a8
Status: Downloaded newer image for philipz/votingapp_result:latest
Creating dockercomposeexample_redis_1
Creating dockercomposeexample_db_1
Creating dockercomposeexample_result_1
Creating dockercomposeexample_vote_1
Creating dockercomposeexample_worker_1
$ cd tests && docker build -t philipz/node-test .
Sending build context to Docker daemon 4.096 kB


Step 1 : FROM node
latest: Pulling from library/node
6a5a5368e0c2: Already exists
7b9457ec39de: Pulling fs layer
```

Build details

Duration: 7 minutes 9 seconds

Finished: a month ago

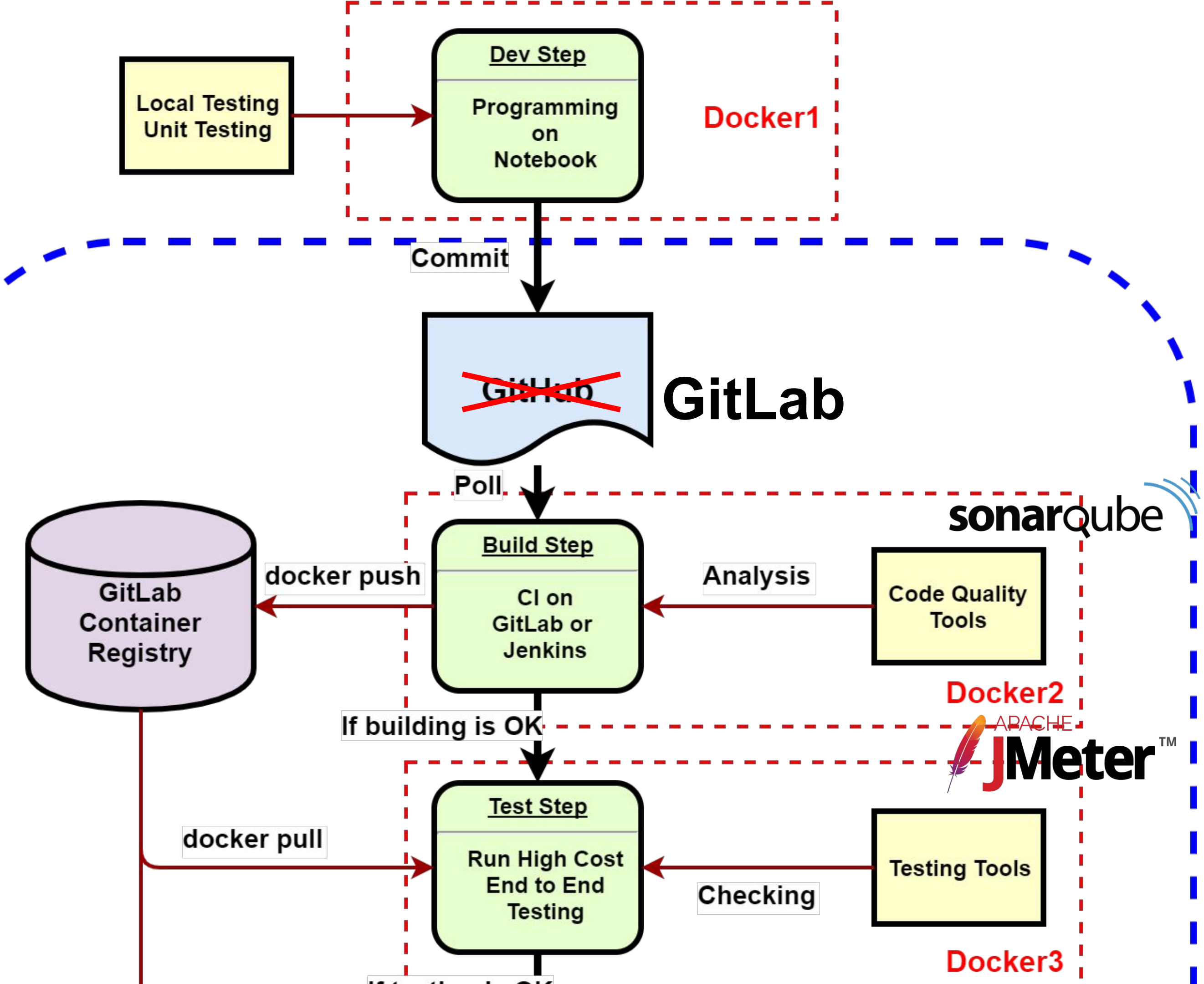
Runner: #21099

Commit title

Remove port mapping.

✓ build

→ ✓ test



Local Testing
Unit Testing

Dev Step

Programming on Notebook

Docker1

Commit

~~GitHub~~

GitLab

Poll

sonarqube

Build Step

CI on GitLab or Jenkins

Analysis

Code Quality Tools

docker push

GitLab Container Registry

Docker2

If building is OK

APACHE JMeter™

Test Step

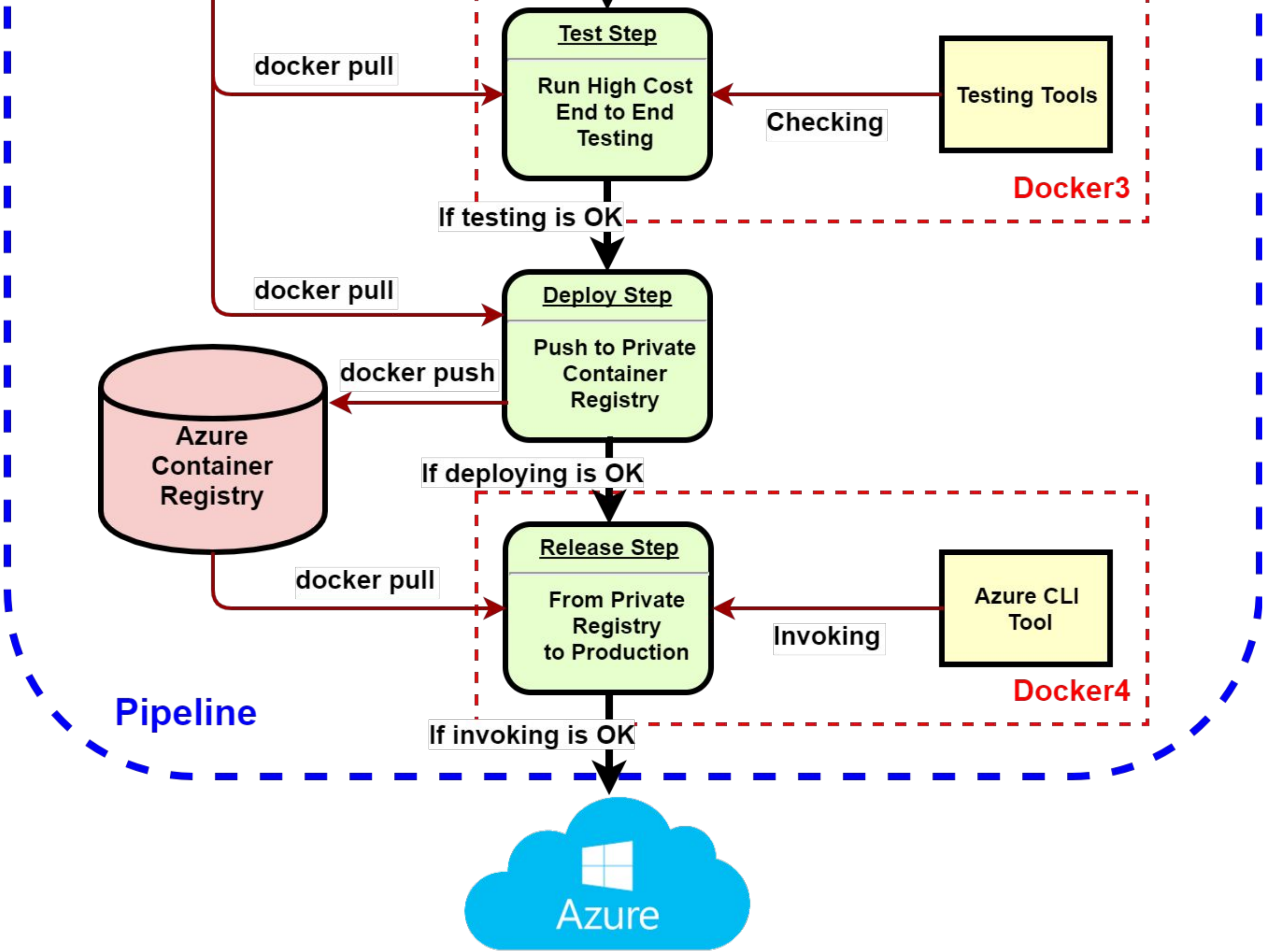
Run High Cost End to End Testing

docker pull

Checking

Testing Tools

Docker3


Test Step
Run High Cost End to End Testing
docker pull
Checking
Testing Tools
Docker3
If testing is OK
docker pull
Deploy Step
Push to Private Container Registry
docker push
Azure Container Registry
If deploying is OK
docker pull
Release Step
From Private Registry to Production
Invoking
Azure CLI Tool
Docker4
Pipeline
If invoking is OK
Azure

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

Thank you

Docker可省下比金錢更寶貴的時間！