docker
容器技術入門與實作研習班
Day2

Philipz(鄭淳尹)
桃園市教育局

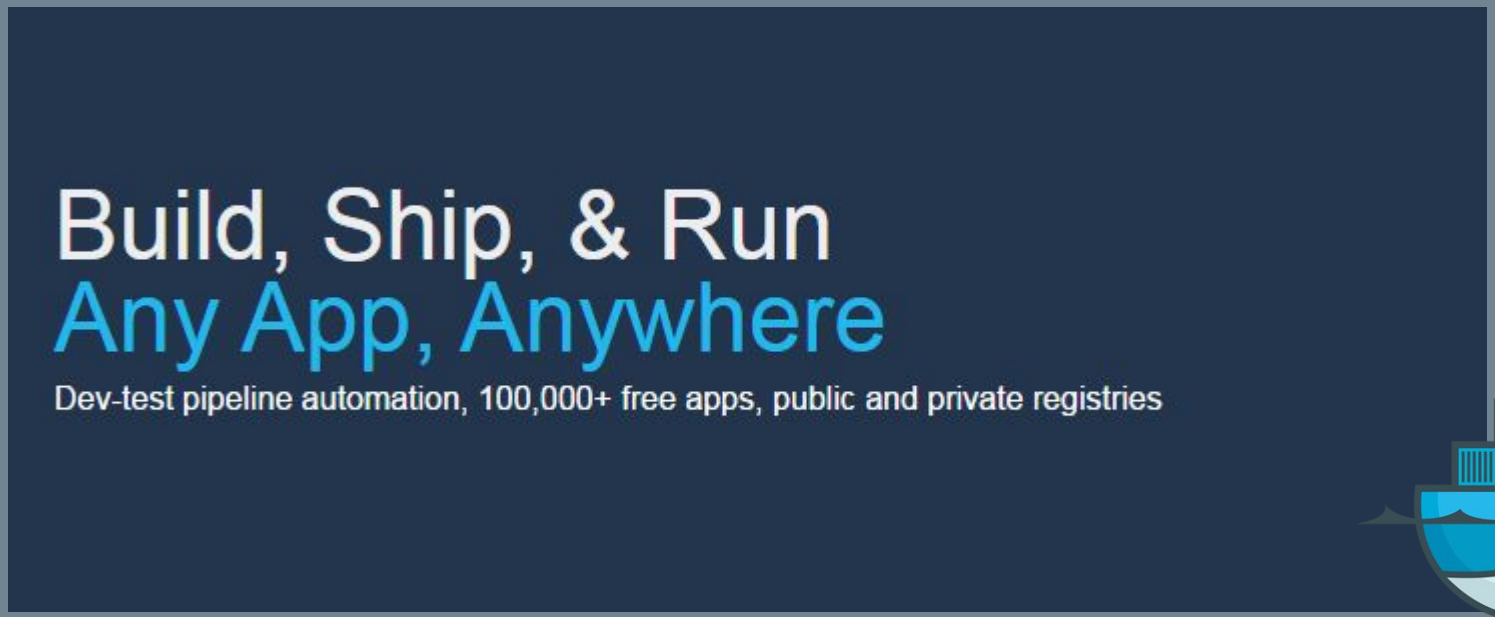
今日課程

1. Docker Hub 介紹
2. Git 基本操作
3. Docker Hub Auto-build image
4. Docker Network、Docker Volume 指令
5. 單一電腦多容器操作
6. Docker Compose 基本指令
7. Docker Compose 實際操作及使用情境
8. Docker + Qemu 模擬 Raspberry Pi Raspbian

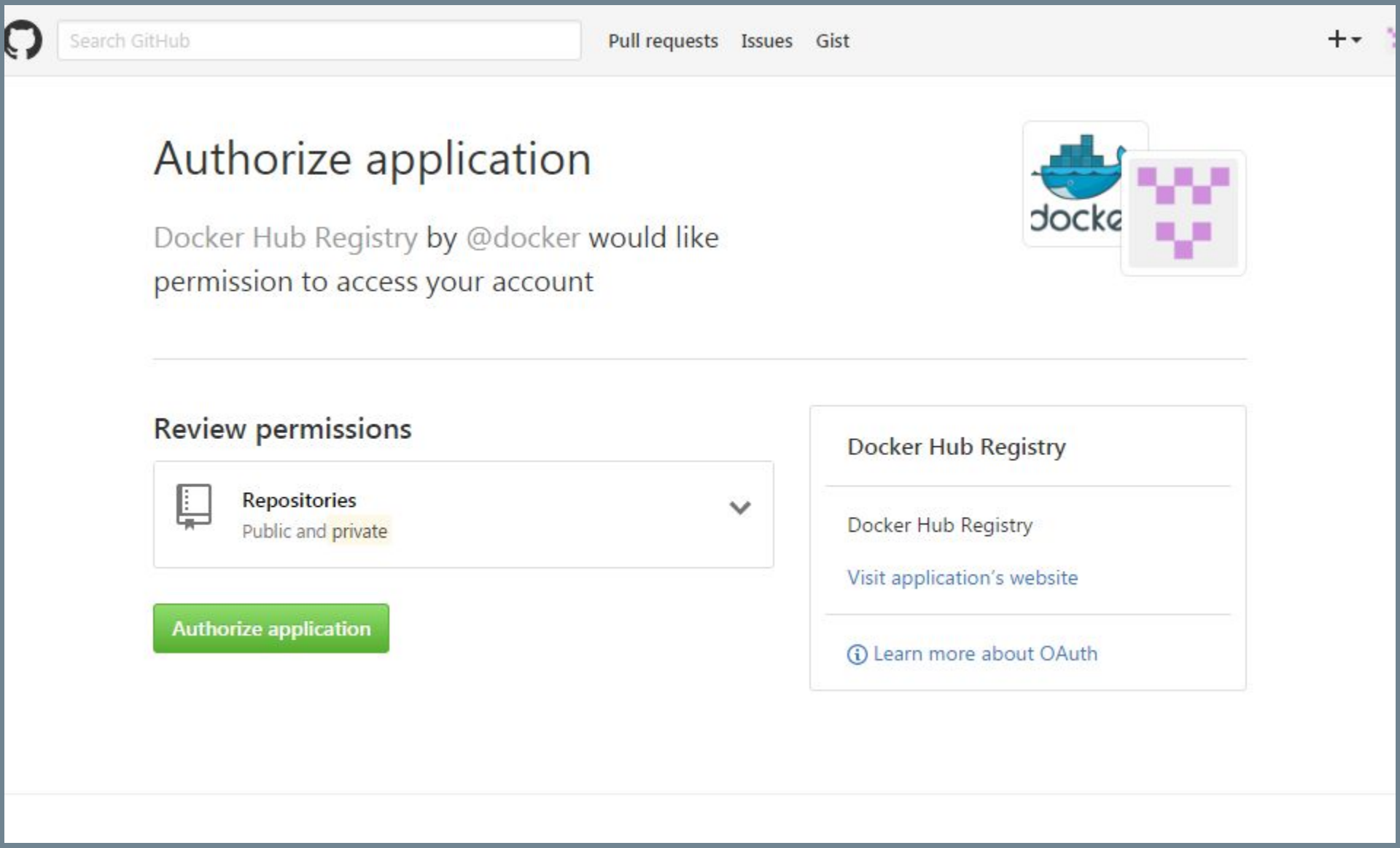
1. Docker Hub 介紹

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



GitHub & Docker Hub



Vulnerability Analysis

CoreOS Clair

Anchore

sha256:204fff67067677bbe3db68ba5ab36eb0749cc7e1cb4ac0f35f5a0d07383e1635

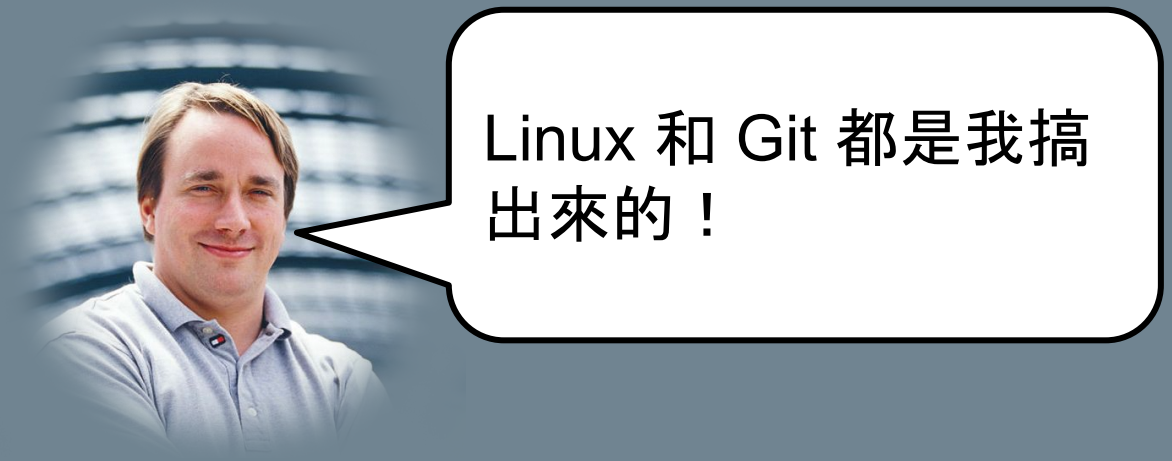
linux 3.16.7-ckt20-1+deb8u2 - ⚠

- 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. Git 基本操作

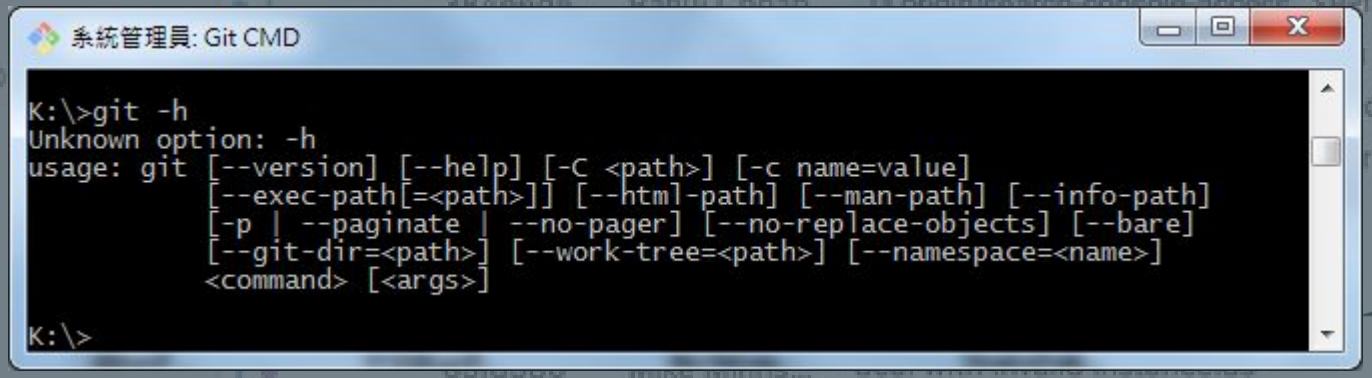
Git by Linus Torvalds

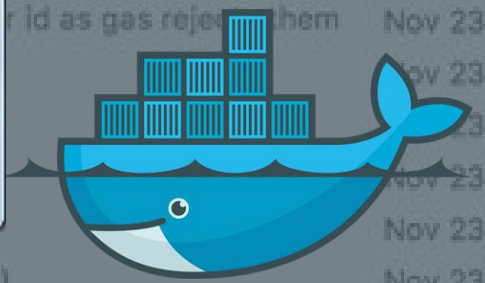
- VCS tool
- Open source community protocol
- GitHub, Bitbucket, GitLab……



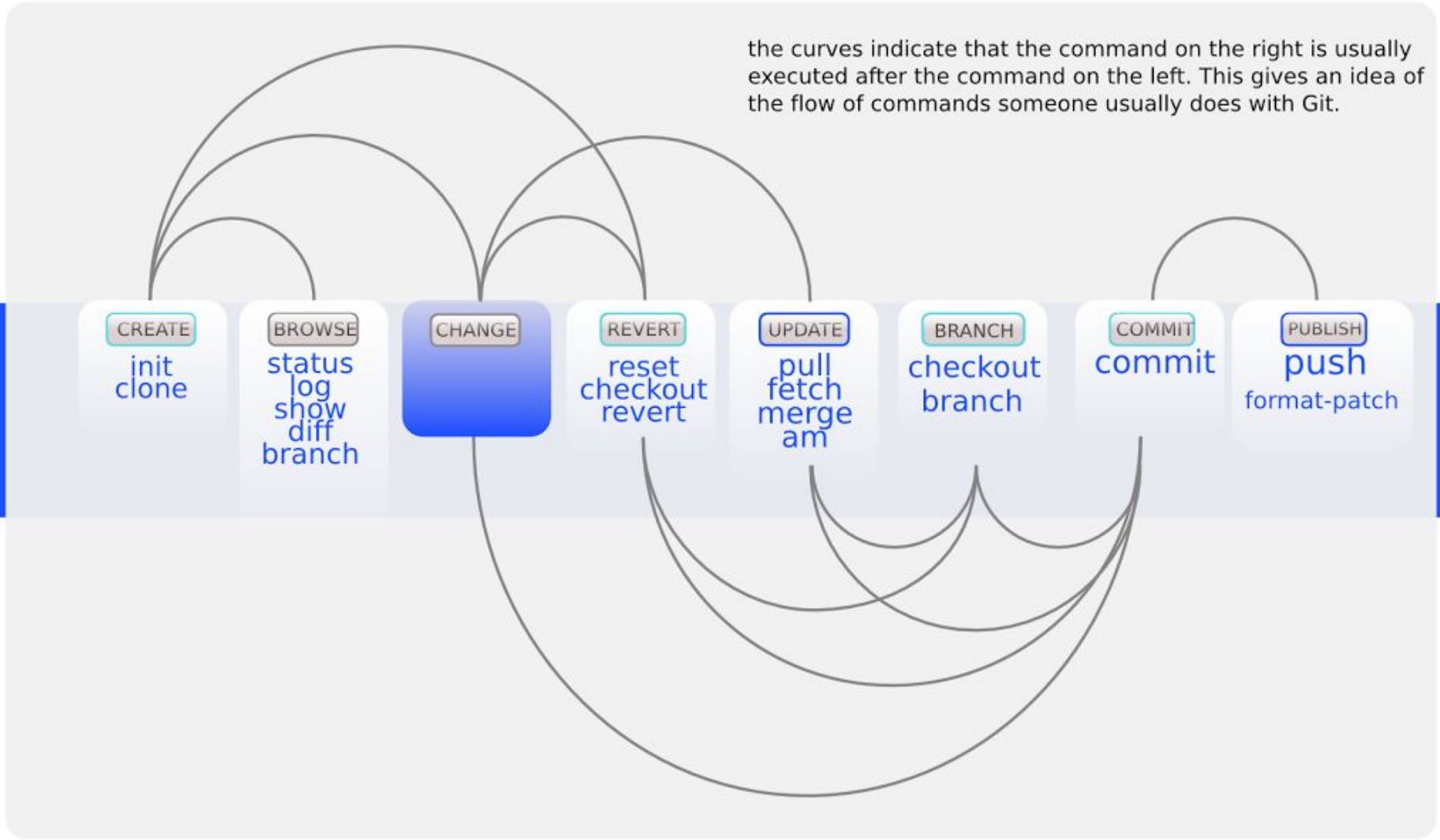
Install Git

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





Commands Sequence



Update

Publish

Git Cheat Sheet

http://git.or.cz/

Remember: git command --help

Global Git configuration is stored in $HOME/.gitconfig (git config --help)

Create

From existing data

```
cd ~/projects/myproject
git init
git add .
```

From existing repo

```
git clone ~/existing/repo ~/new/repo
git clone git://host.org/project.git
git clone ssh://you@host.org/proj.git
```

Show

Files changed in working directory

Concepts

Git Basics

```
master    : default development branch
origin    : default upstream repository
HEAD      : current branch
HEAD^     : parent of HEAD
HEAD~4    : the great-great grandparent of HEAD
```

Revert

Return to the last committed state

```
git reset --hard
```

you cannot undo a hard reset

Files changed in working directory
git status

Changes to tracked files
git diff

What changed between $ID1 and $ID2
git diff $id1 $id2

History of changes
git log

History of changes for file with diffs
git log -p $file $dir/ec/tory/

Who changed what and when in a file
git blame $file

A commit identified by $ID
git show $id

A specific file from a specific $ID
git show $id:$file

All local branches
git branch
(star '*' marks the current branch)

Cheat Sheet Notation

$id : notation used in this sheet to represent either a commit id, branch or a tag name
$file : arbitrary file name
$branch : arbitrary branch name

git reset --hard
⚠ you cannot undo a hard reset

Revert the last commit
git revert HEAD Creates a new commit

Revert specific commit
git revert $id Creates a new commit

Fix the last commit
git commit -a --amend
(after editing the broken files)

Checkout the $id version of a file
git checkout $id $file

Branch

Switch to the $id branch
git checkout $id

Merge branch1 into branch2
git checkout $branch2
git merge branch1

Create branch named $branch based on the HEAD
git branch $branch

Create branch $new_branch based on branch $other and switch to it
git checkout -b $new_branch $other

Delete branch $branch
git branch -d $branch

Upd

Fetch
git fe
(but th

Pull lat
git pu
(does

Apply
git an

Useful Commands

Find
gi
gi
gi
gi
gi
gi

Che
gi
gi

Sea
gi

Update

Fetch latest changes from origin
git fetch
(but this does not merge them).

Pull latest changes from origin
git pull
(does a fetch followed by a merge)

Apply a patch that some sent you
git am -3 patch.mbox
(in case of a conflict, resolve and use
git am --resolved)

Publish

Commit all your local changes
git commit -a

Prepare a patch for other developers
git format-patch origin

Push changes to origin
git push

Mark a version / milestone
git tag v1.0

useful commands

Finding regressions

git bisect start (to start)
git bisect good $id ($id is the last working version)
git bisect bad $id ($id is a broken version)

git bisect bad/good (to mark it as bad or good)
git bisect visualize (to launch gitk and mark it)
git bisect reset (once you're done)

Check for errors and cleanup repository

git fsck
git gc --prune

Search working directory for foo()

olve Merge Conflicts

To view the merge conclicts

git diff (complete conflict diff)
git diff --base $file (against base file)
git diff --ours $file (against your changes)
git diff --theirs $file (against other changes)

To discard conflicting patch

git reset --hard
git rebase --skip

After resolving conflicts, merge with

git add $conflicting_file (do for all resolved files)

3. Docker Hub Auto-build

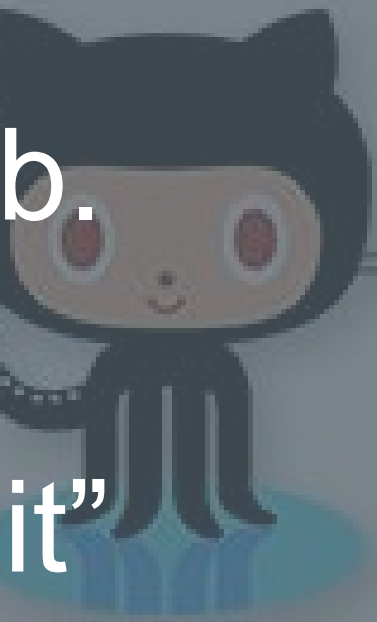
Dockerfile

範例 :

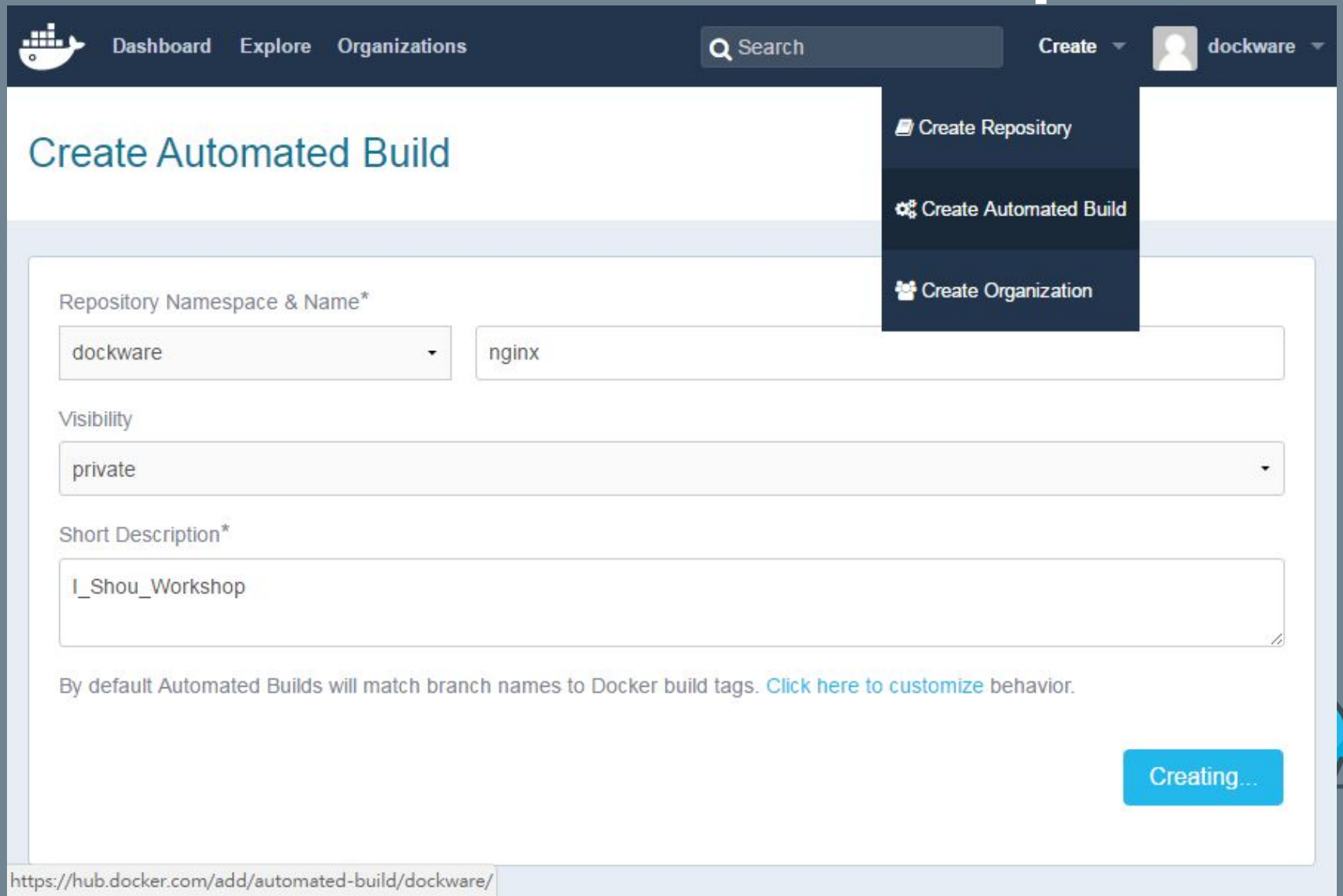
```
FROM debian:jessie
MAINTAINER docker "docker@nginx.com"
RUN apt-get update && apt-get install -y nginx
CMD ["nginx", "-g", "daemon off;"]
```

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.



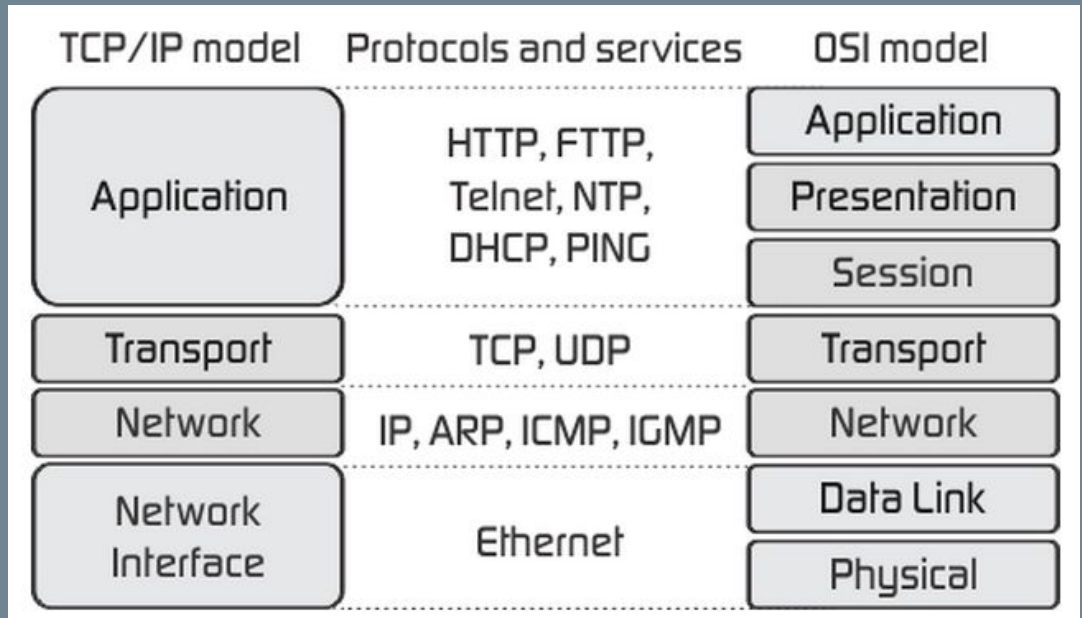
建置設定

docker pull YOURNAME/IMAGENAME

4.1 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



Network 相關指令

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

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

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

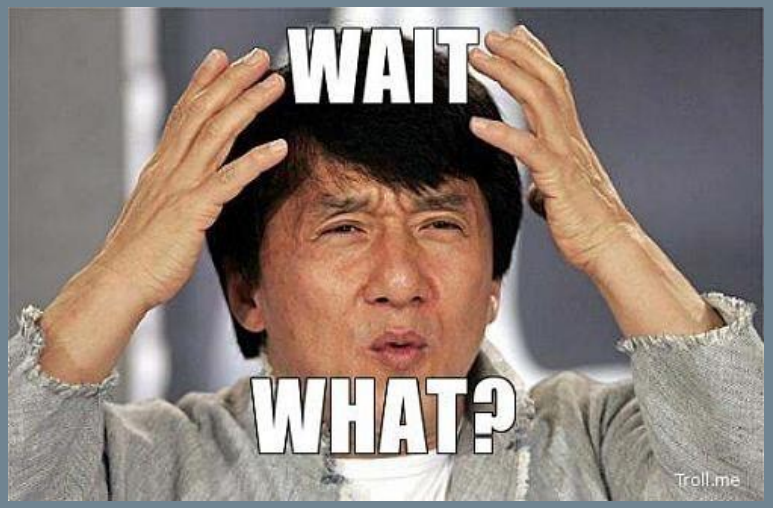
ping -w3 172.17.0.2, ping container1, ping container3

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

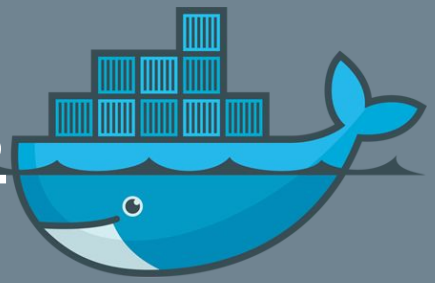
Get started with Macvlan network driver

4.2 Docker Volume 指令

Shared data volume commands

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

Manage data in containers



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
```

5. 單一電腦多容器操作

6. Docker Compose
基本指令

安裝 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

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

Kompose = Kubernetes + Compose

7.1 Docker Compose
實際操作

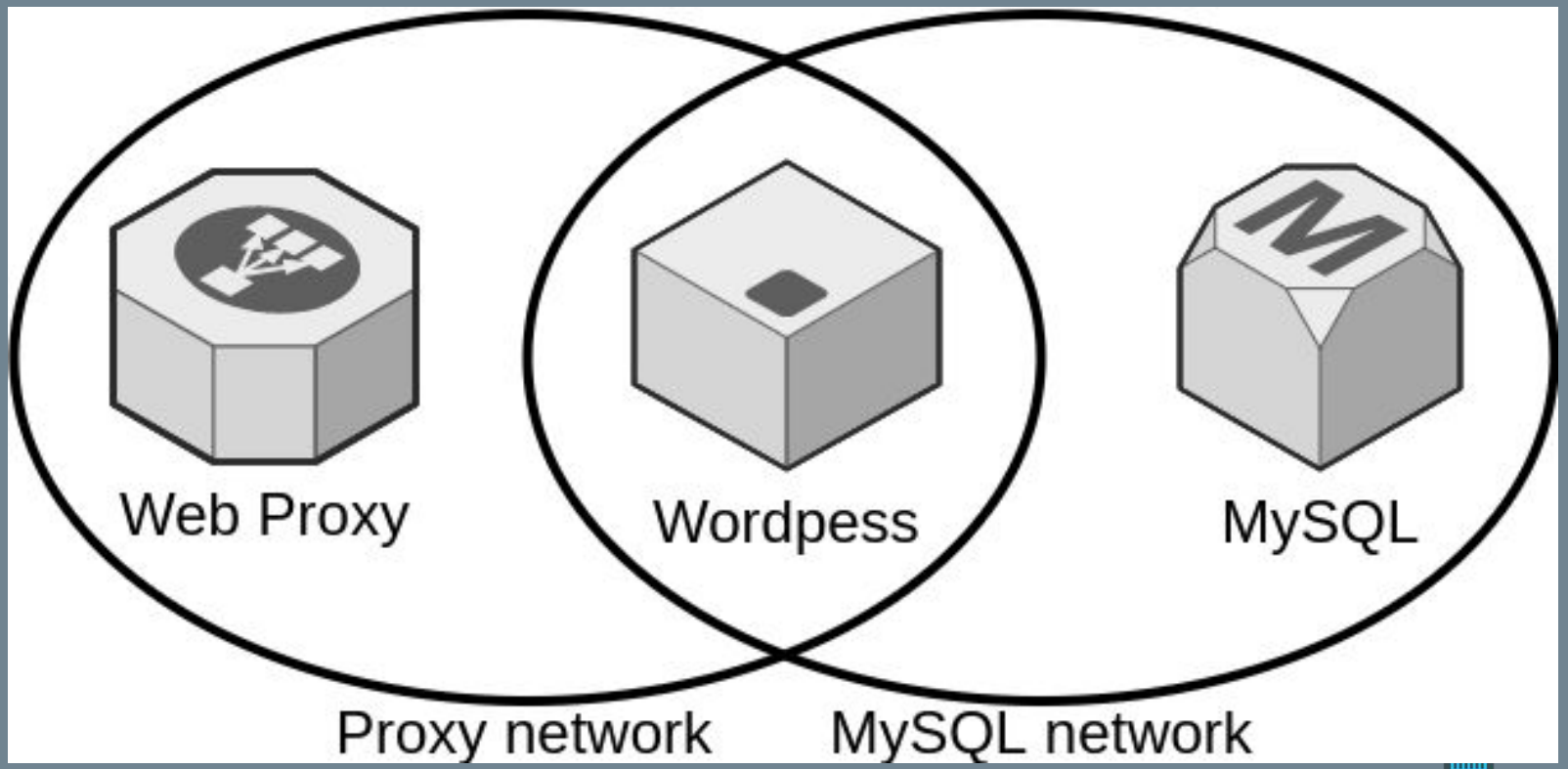
Compose File Sample (1/2)

```
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:
      - "8000:80"
    restart: always
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_PASSWORD: wordpress
volumes:
    db_data:
*** nslookup wordpress ***
```

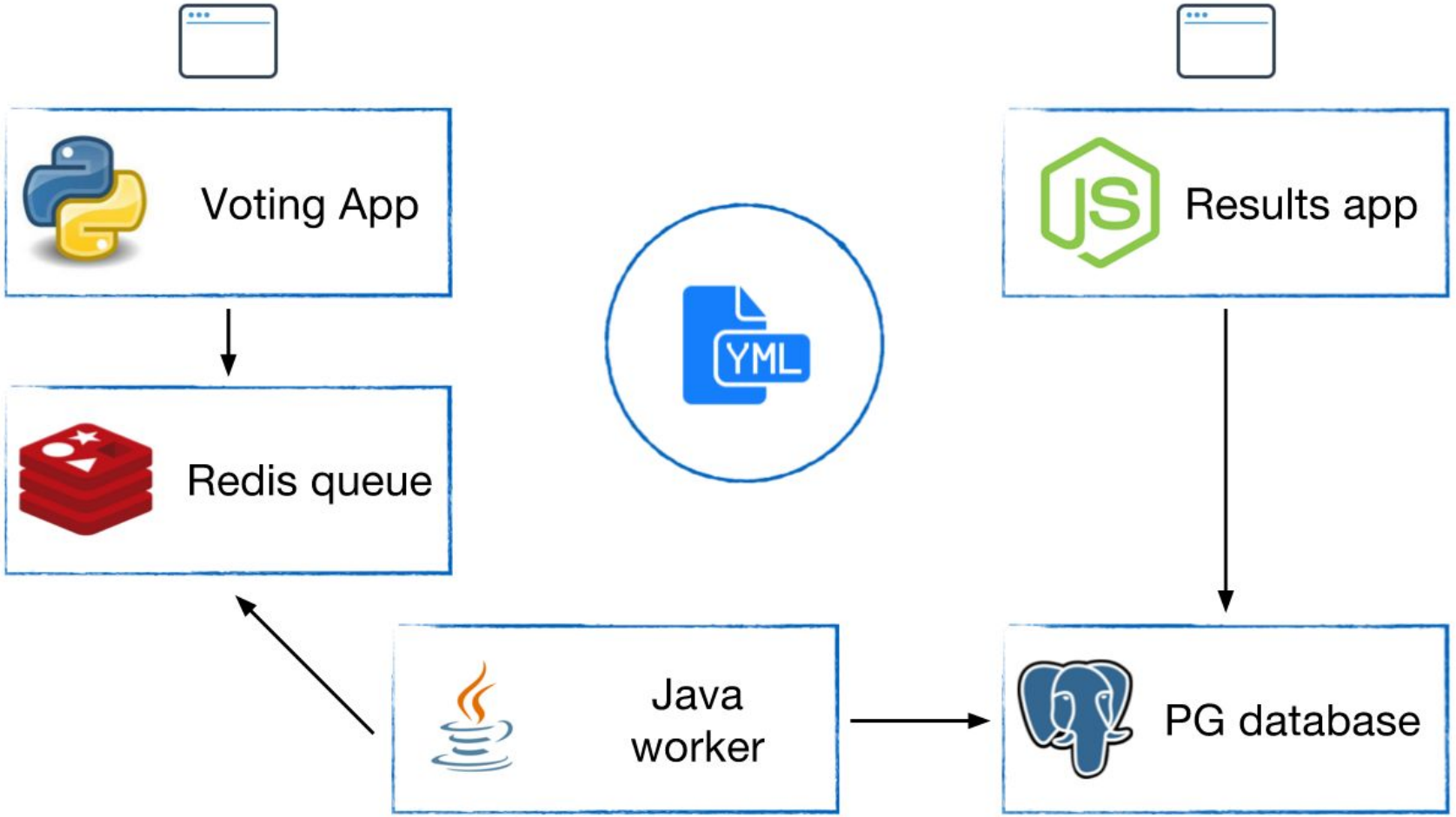
Compose & Wordpress

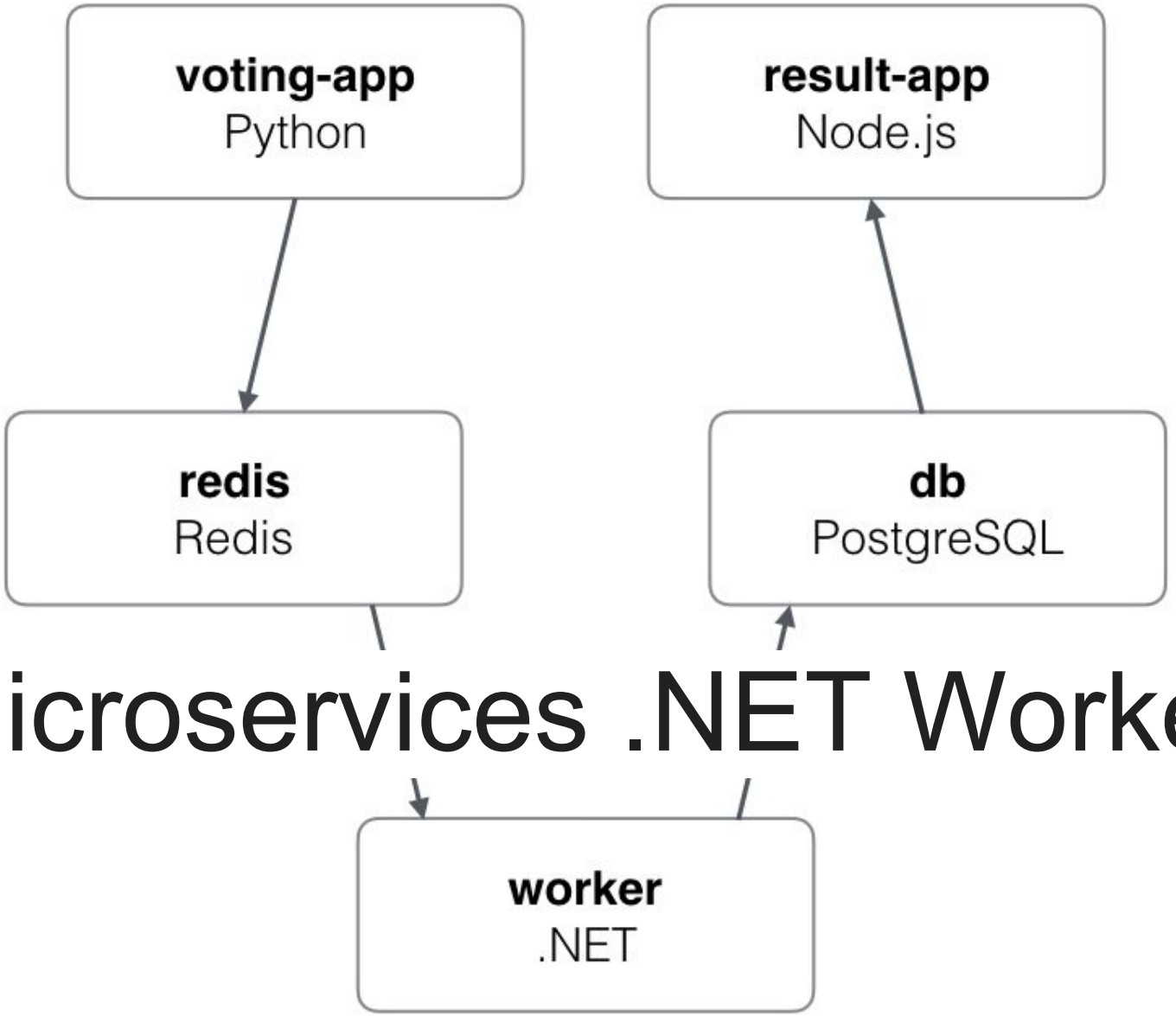


- 水平擴展 wordpress:scale

Microservices Java Worker

[Docker Birthday #3 training]





Microservices .NET Worker

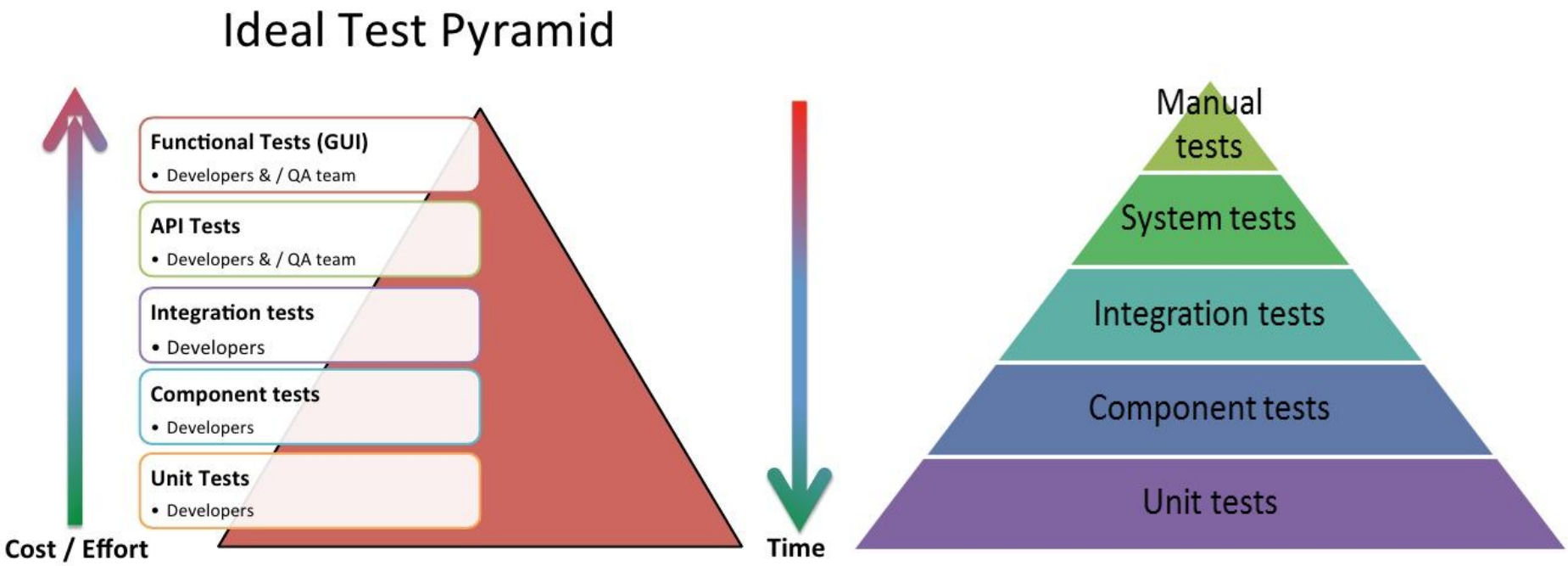
Docker Birthday #3 training

7.2 Docker Compose 使用情境

Docker Compose & CI/CD

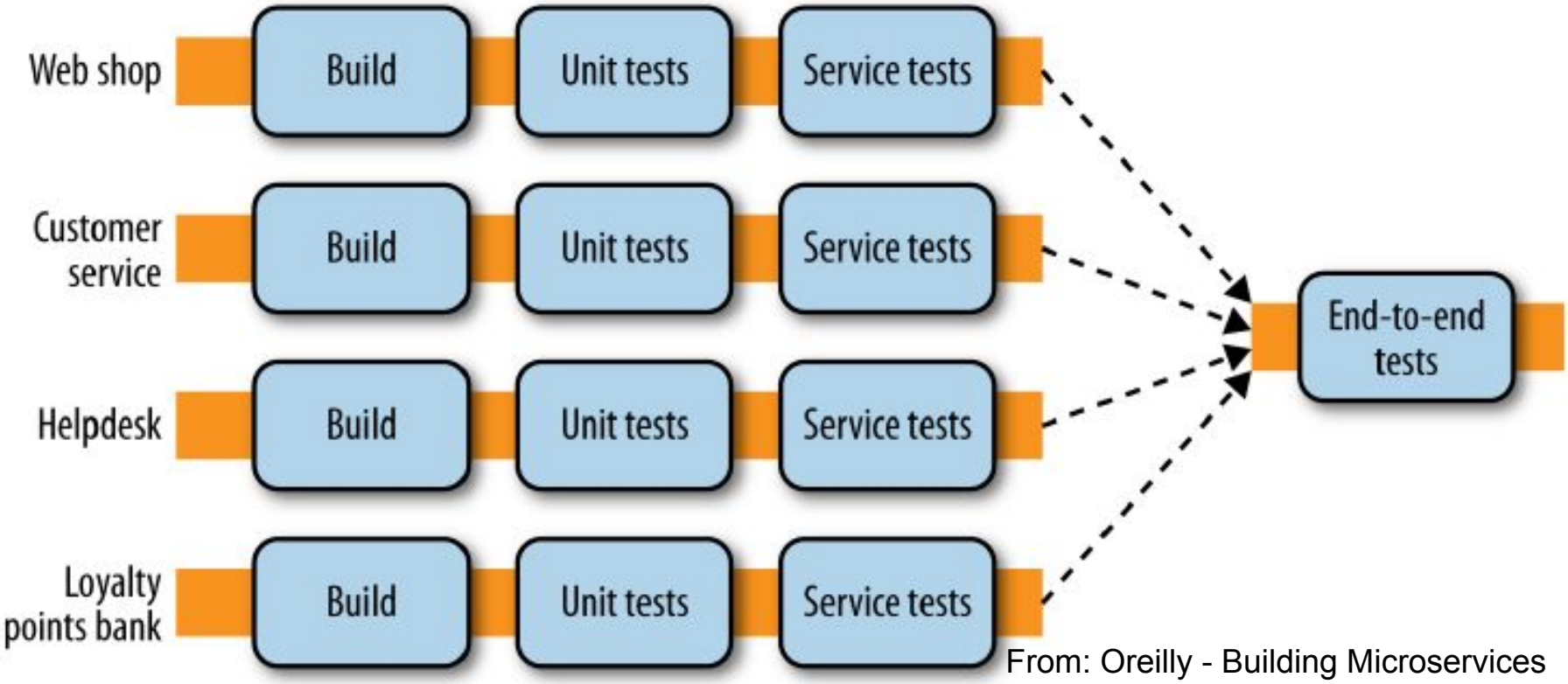
~~GitHub~~, ~~CircleCI~~, ~~Docker Hub~~ = GitLab

Testing level? Coding effort? Env. build-up



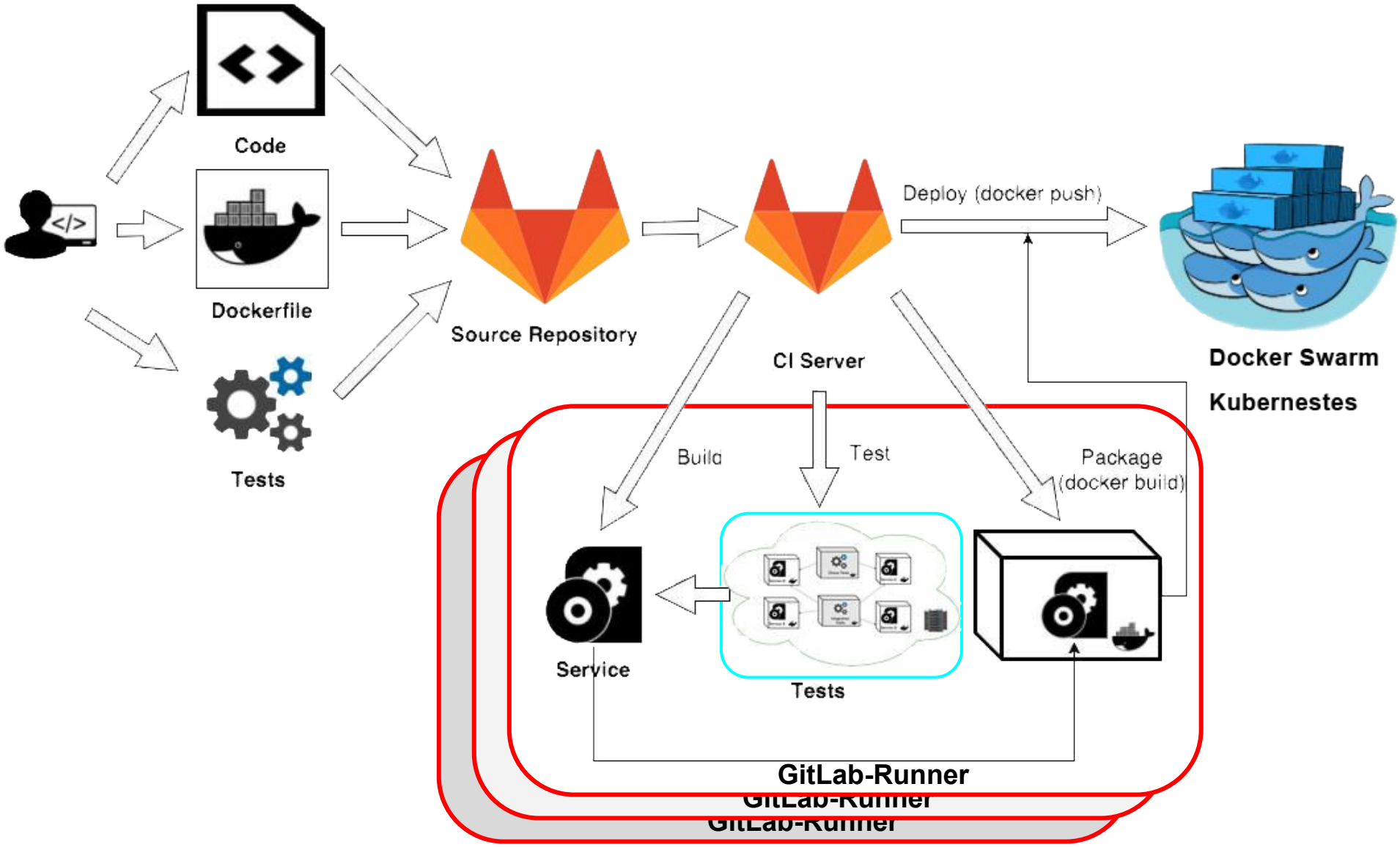
End to End Tests

CI with Docker Compose is easy to implement.

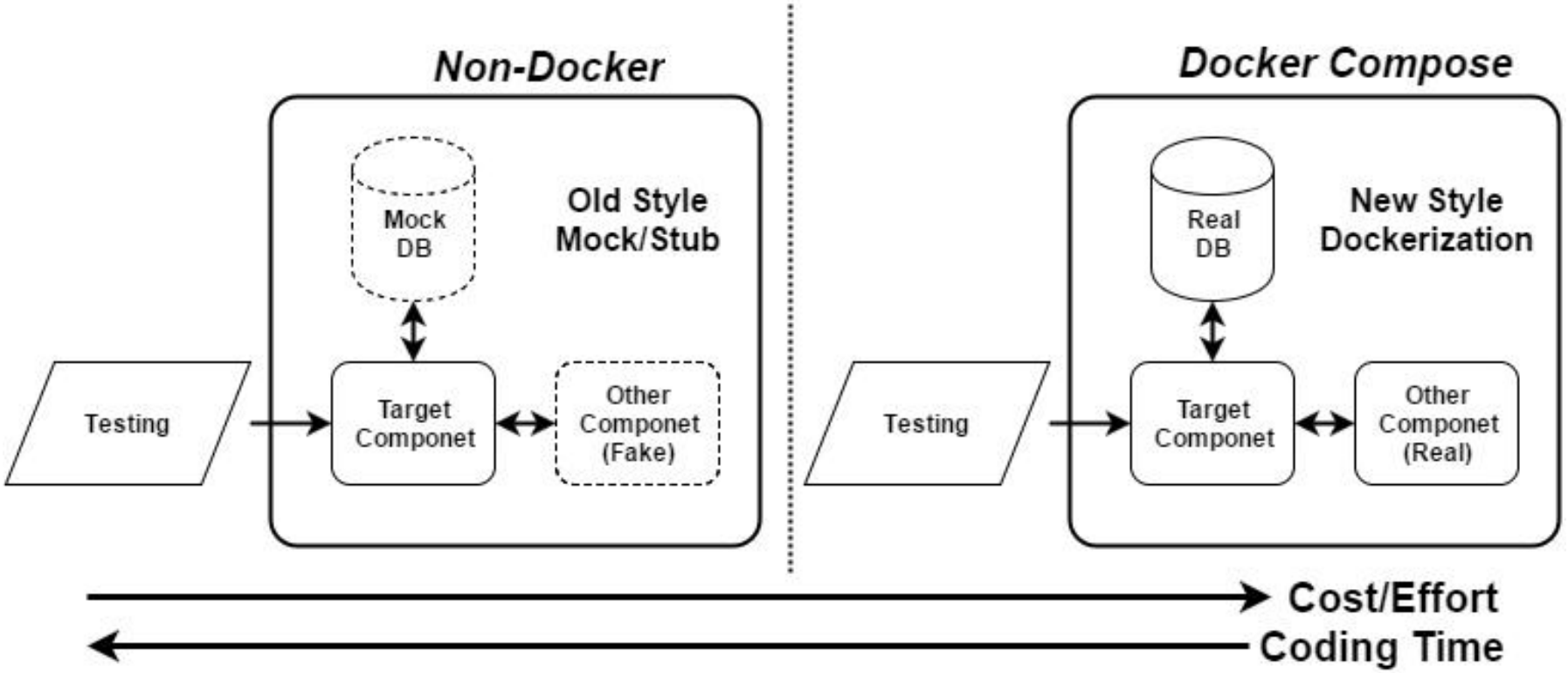


From: Oreilly - Building Microservices

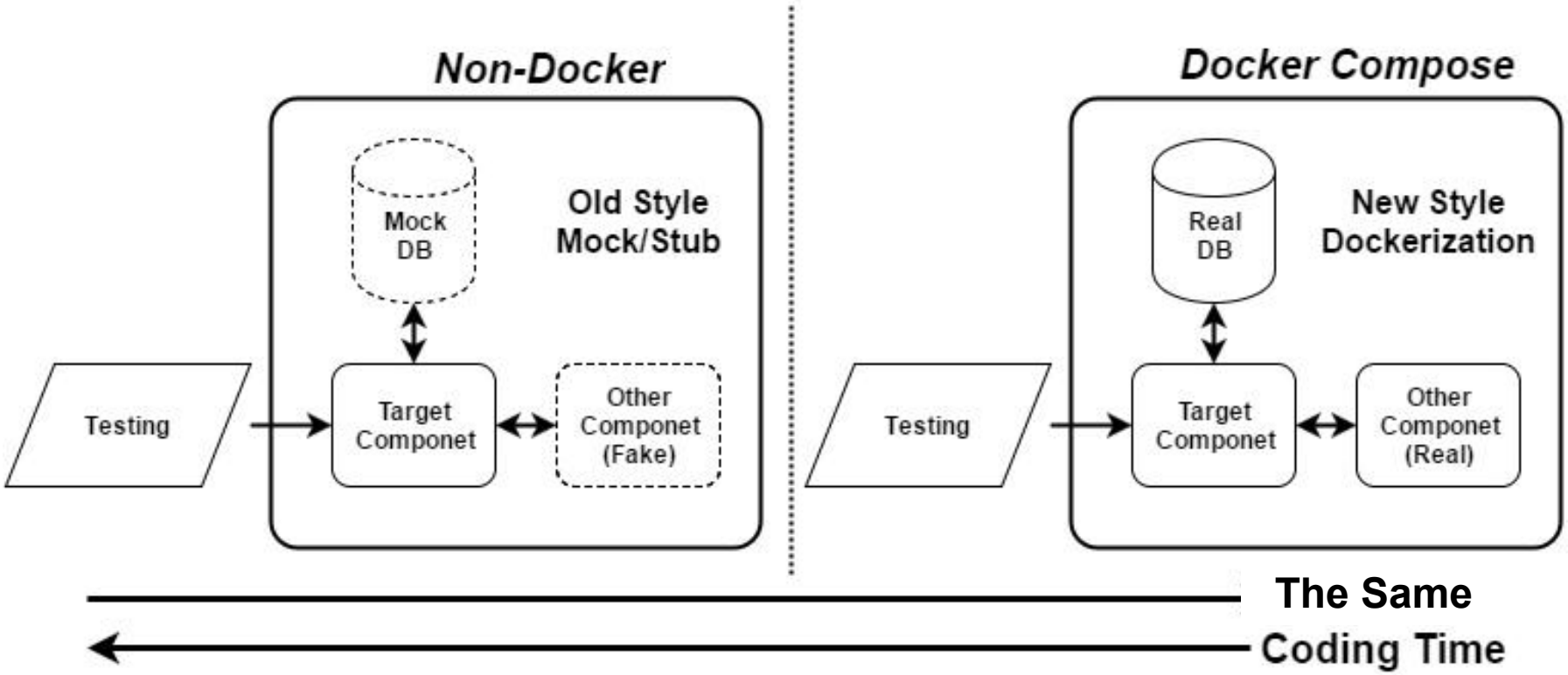
容器開發流程

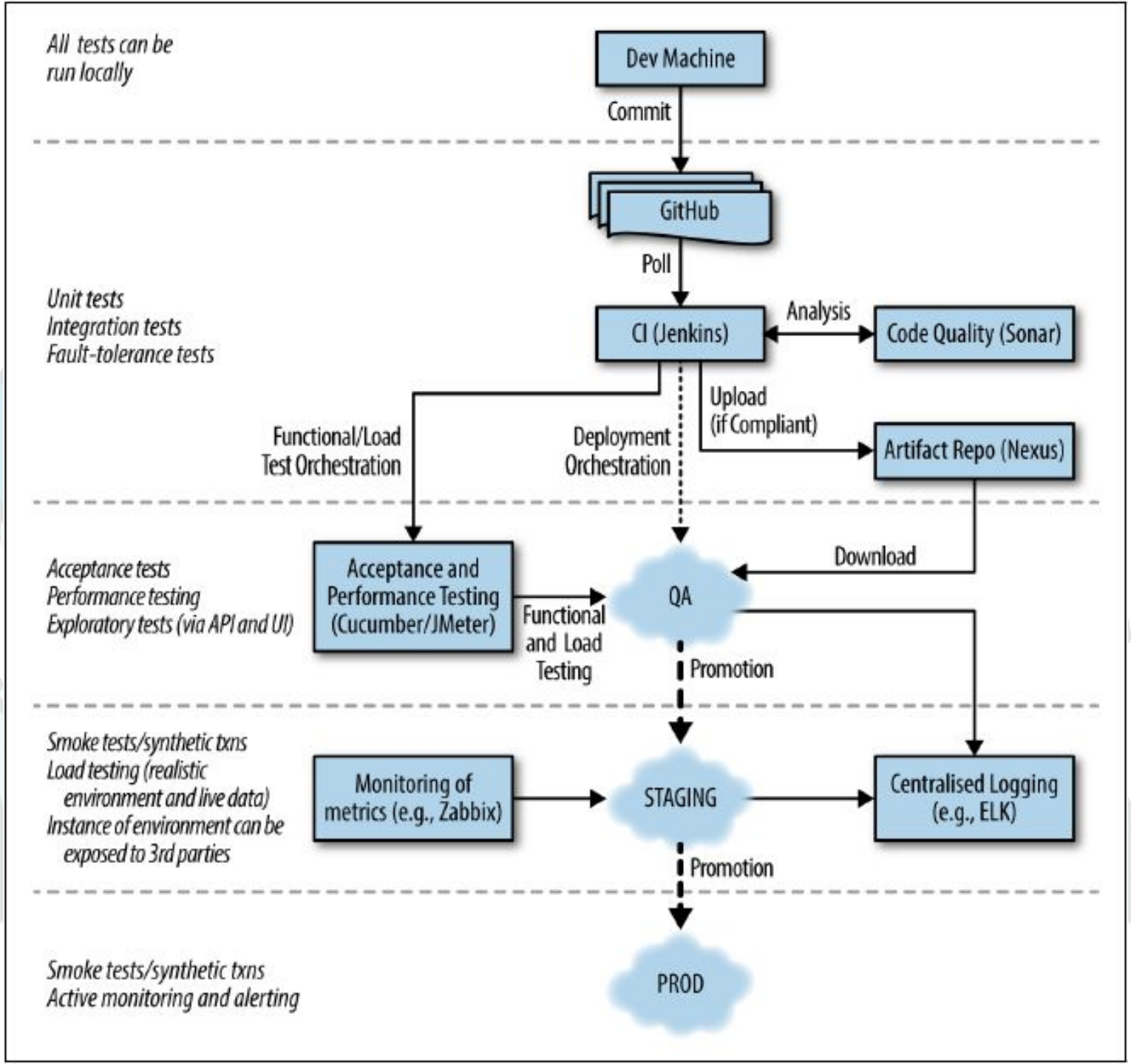

Code
Dockerfile
Tests
Source Repository
CI Server
Deploy (docker push)
Docker Swarm
Kubernestes
Build
Test
Package
(docker build)
Service
Tests
GitLab-Runner

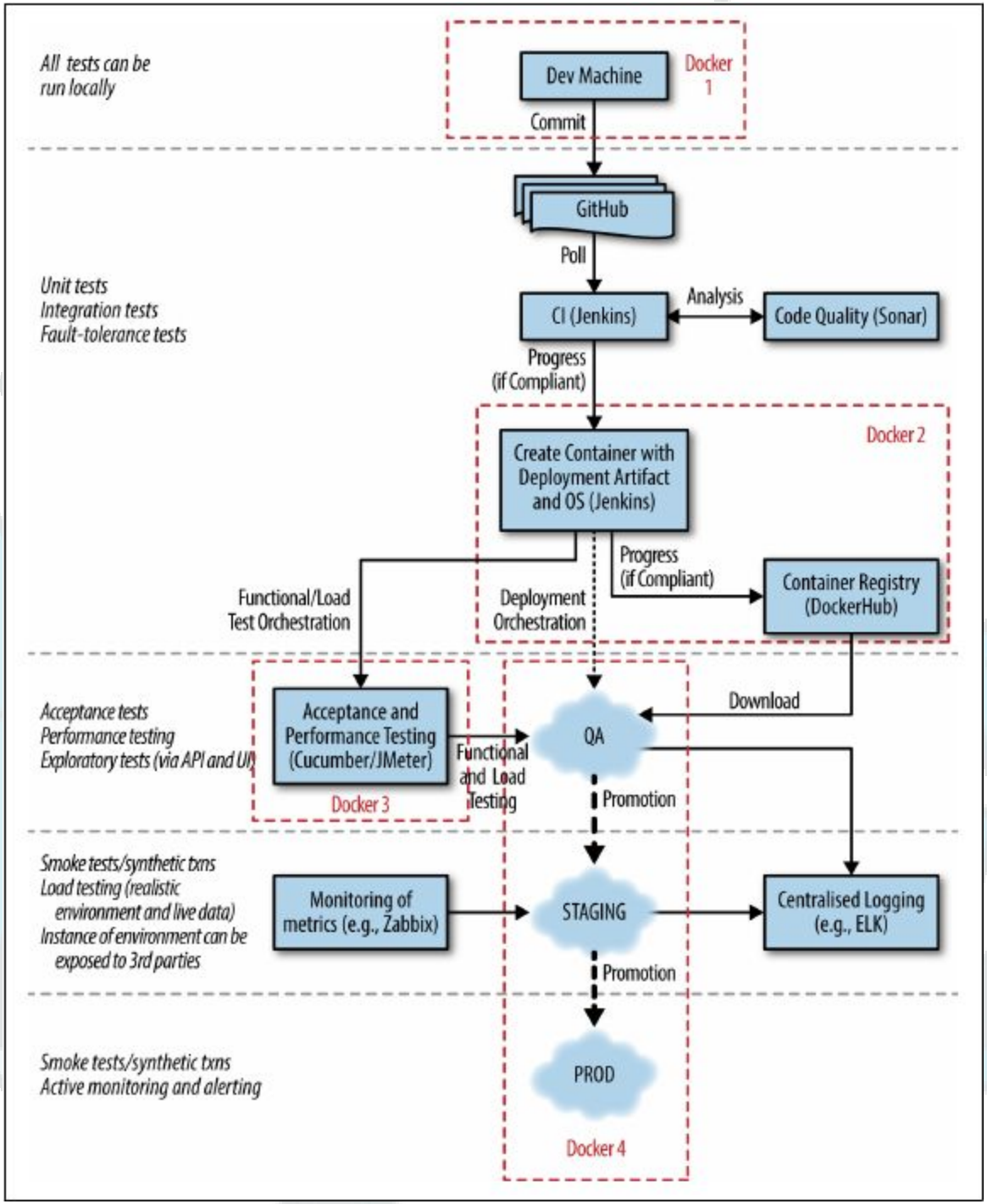
Test Double 測試方法

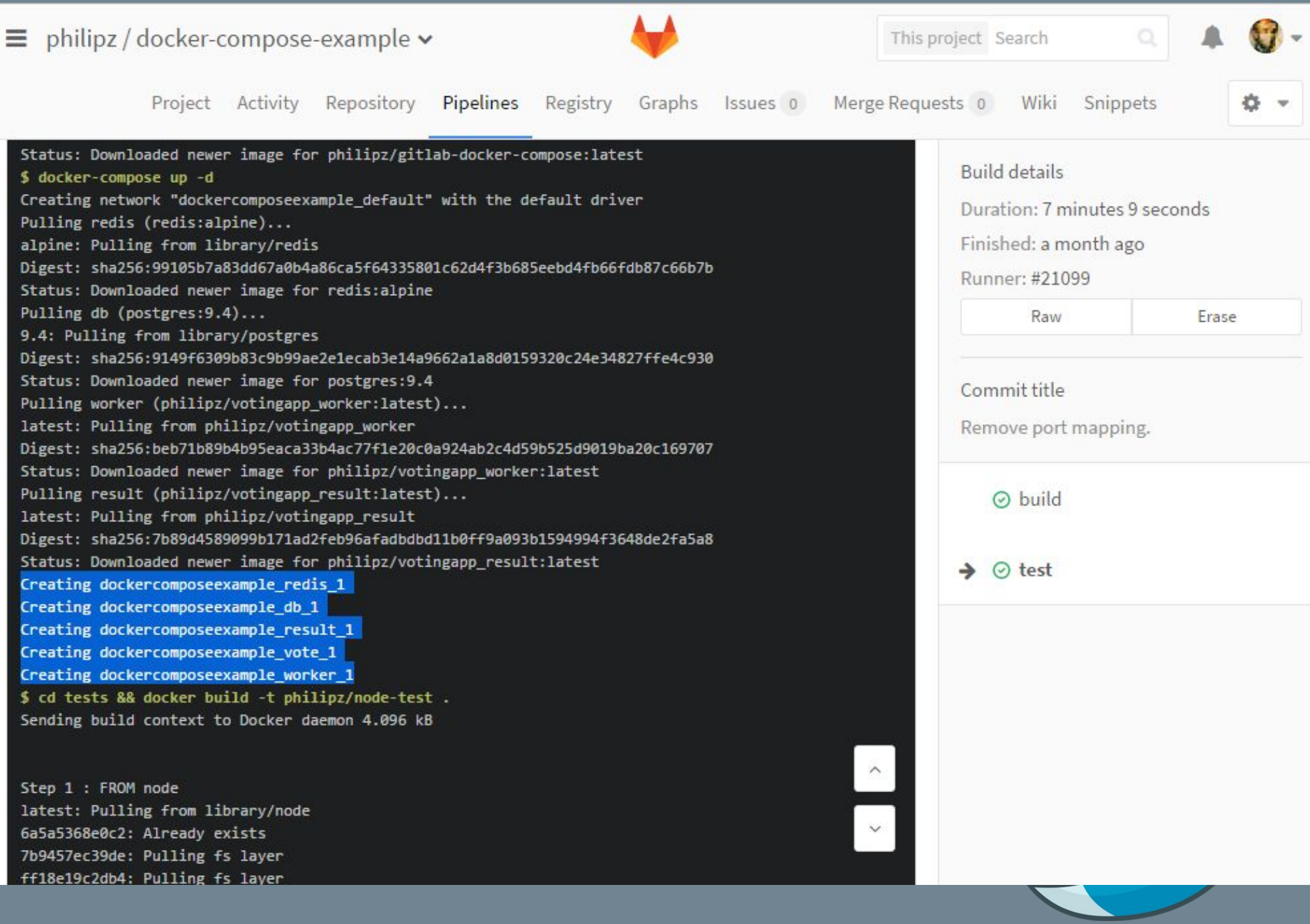


新 Compose 測試方法




All tests can be run locally
Dev Machine
Commit
GitHub
Poll
Unit tests
Integration tests
Fault-tolerance tests
CI (Jenkins)
Analysis
Code Quality (Sonar)
Upload (if Compliant)
Artifact Repo (Nexus)
Functional/Load Test Orchestration
Deployment Orchestration
Download
Acceptance tests
Performance testing
Exploratory tests (via API and UI)
Acceptance and Performance Testing (Cucumber/JMeter)
Functional and Load Testing
QA
Promotion
Smoke tests/synthetic txns
Load testing (realistic environment and live data)
Instance of environment can be exposed to 3rd parties
Monitoring of metrics (e.g., Zabbix)
STAGING
Centralised Logging (e.g., ELK)
Promotion
Smoke tests/synthetic txns
Active monitoring and alerting
PROD


All tests can be run locally
Dev Machine
Docker 1
Commit
GitHub
Poll
Unit tests
Integration tests
Fault-tolerance tests
CI (Jenkins)
Analysis
Code Quality (Sonar)
Progress (if Compliant)
Docker 2
Create Container with Deployment Artifact and OS (Jenkins)
Progress (if Compliant)
Container Registry (DockerHub)
Functional/Load Test Orchestration
Deployment Orchestration
Download
Acceptance tests
Performance testing
Exploratory tests (via API and UI)
Acceptance and Performance Testing (Cucumber/JMeter)
Docker 3
Functional and Load Testing
QA
Promotion
Smoke tests/synthetic txns
Load testing (realistic environment and live data)
Instance of environment can be exposed to 3rd parties
Monitoring of metrics (e.g., Zabbix)
STAGING
Centralised Logging (e.g., ELK)
Promotion
Smoke tests/synthetic txns
Active monitoring and alerting
PROD
Docker 4


philipz / docker-compose-example
This project Search
Project Activity Repository Pipelines Registry Graphs Issues 0 Merge Requests 0 Wiki Snippets
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
ff18e19c2db4: Pulling fs layer
Build details
Duration: 7 minutes 9 seconds
Finished: a month ago
Runner: #21099
Raw
Erase
Commit title
Remove port mapping.
build
test

8. Docker & Qemu & Raspberry Pi Raspbian

RPi & Docker

How to build a base image

Cross-compiler

1. Building ARM containers on x86 machine
2. Qemu-static-Docker IoT CI/CD
3. Using GPIO with Docker

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

Builds » philipz » rpi_circleci » master » build 19

0
(03:23)

Add Containers +

```
Step 4 : COPY qemu/cross-build-end qemu/cross-build-start qemu/qemu-arm-static qemu/sh-shim /usr/bin/
 ---> 6b9181f32891
Error removing intermediate container c2702bd608f7: nosuchcontainer: No such container: c2702bd608f796
2e2939b88af88f15241ee45d5d003c81105890da670df6e203
Step 5 : RUN cross-build-start
 ---> Running in 1d0c6ff52fd3
 ---> a92560a622a5
Error removing intermediate container c2702bd608f7: nosuchcontainer: No such container: c2702bd608f796
2e2939b88af88f15241ee45d5d003c81105890da670df6e203
Step 6 : RUN apt-get update && apt-get install -y mosquitto-clients
 ---> Running in b94e9a36c402
Get:1 http://archive.raspbian.org jessie InRelease [14.9 kB]
Get:2 http://archive.raspbian.org jessie/main armhf Packages [12.5 MB]
Fetched 12.5 MB in 12s (1019 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 34 not upgraded.
Need to get 999 kB of archives.
After this operation, 2542 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+deb8u2 [852 kB]
Get:2 http://archive.raspbian.org/raspbian/ jessie/main libc-ares2 armhf 1.10.0-2 [71.3 kB]
Get:3 http://archive.raspbian.org/raspbian/ jessie/main libmosquitto1 armhf 1.3.4-2 [36.3 kB]
Get:4 http://archive.raspbian.org/raspbian/ jessie/main mosquitto-clients armhf 1.3.4-2 [39.3 kB]
debconf: delaying package configuration, since apt-utils is not installed
Fetched 999 kB in 1s (621 kB/s)
Selecting previously unselected package libssl1.0.0:armhf.

(Reading database ... 7086 files and directories currently installed.)
```

?

9. 展示以 Docker 執行 TensorFlow

Docker + TensorFlow + GPU

- Machine Learning, Deep Learning
- TensorFlow Docker images
- nvidia-docker, All-in-one DL image

10. 課程最後結語

Still No Silver Bullet

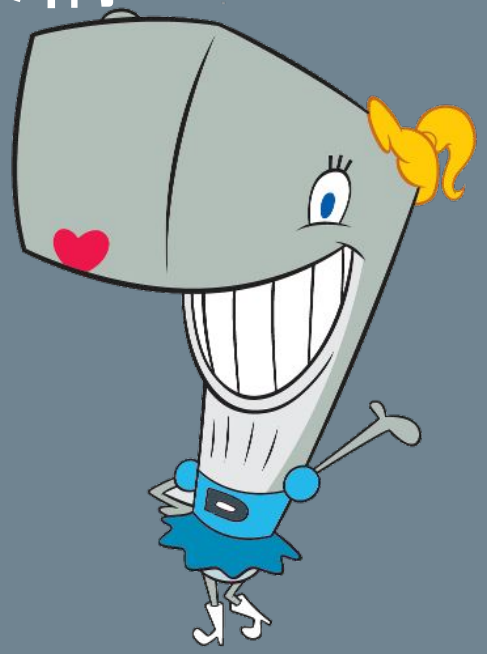
容器只是其中一個關鍵，並非全部.

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

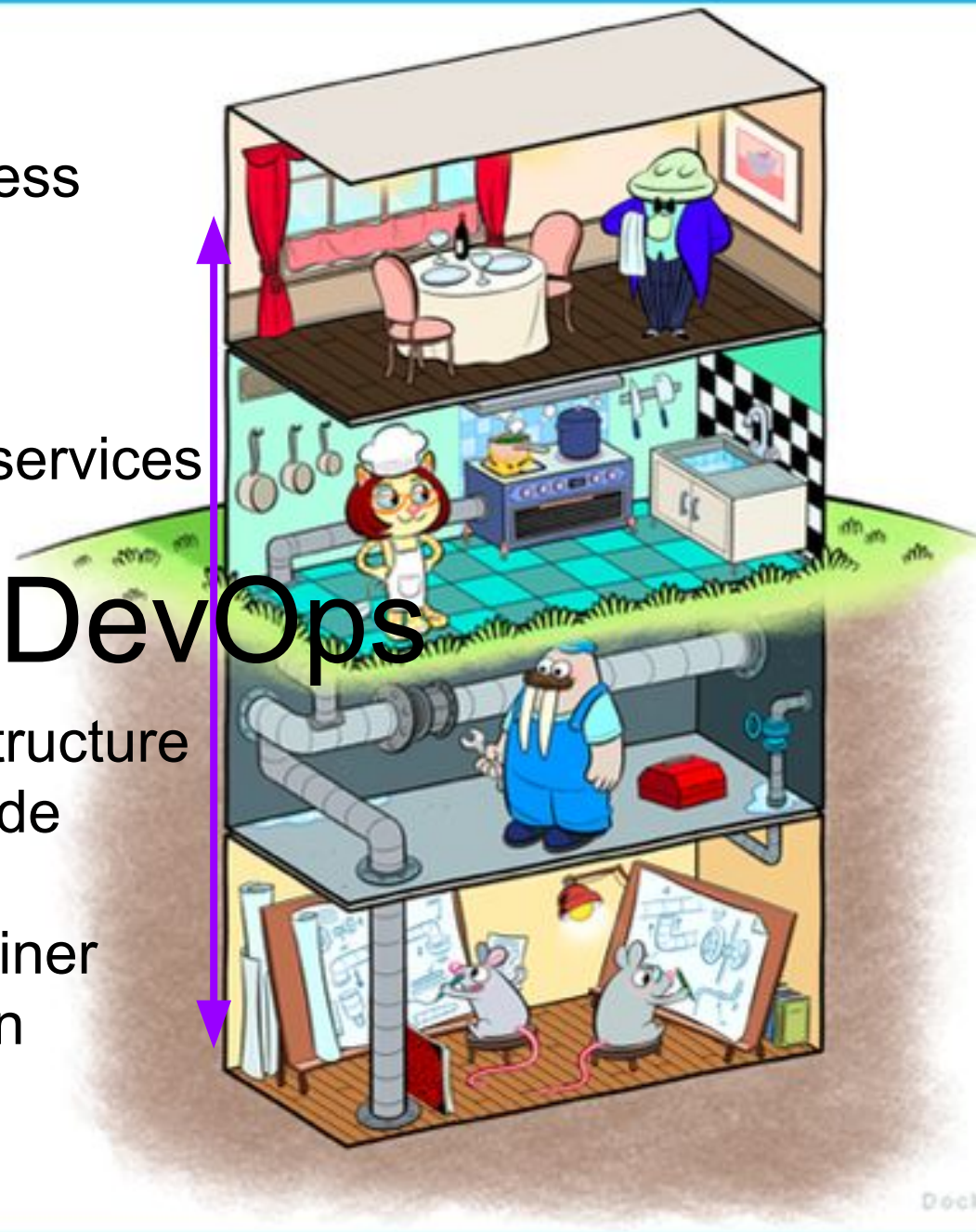
Business model

Microservices

DevOps

Infrastructure as Code

Container Design



The Docker Stack

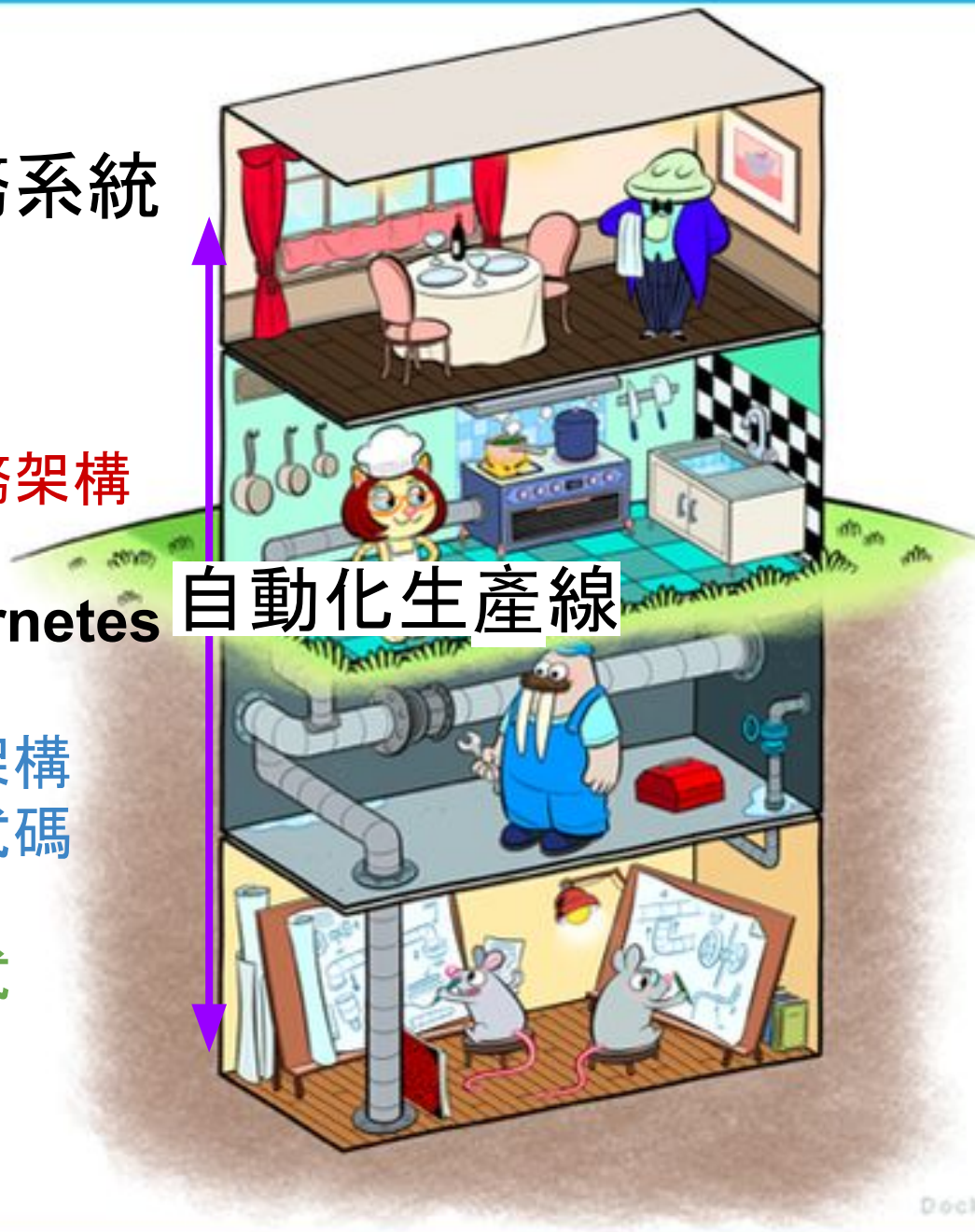
*業務系統

微服務架構

Kubernetes

基礎架構
即程式碼

容器式
設計



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

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

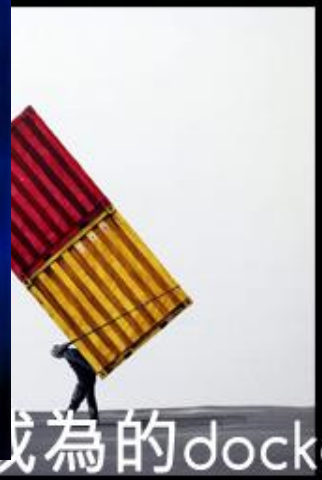
最終成為的docker

docker

老闆眼中的d

眼中的docker

原本以為的docker



最終成為的docker

Hope You Love Docker
So long!