台北 docker 一日入門篇

Philipz(鄭淳尹)

2017-05-21

東吳大學數學系

Philipz (鄭淳尹)

Docker.Taipei 共同發起人

歐萊禮《Docker 錦囊妙計》譯者

碁峰《Docker入門與實戰》、

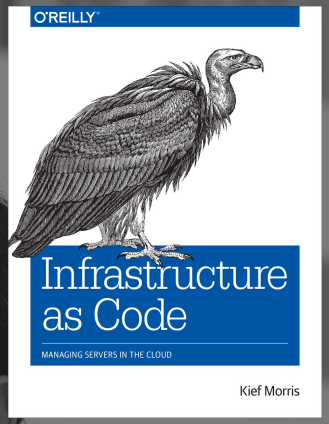
《Kubernetes使用指南》審譯者

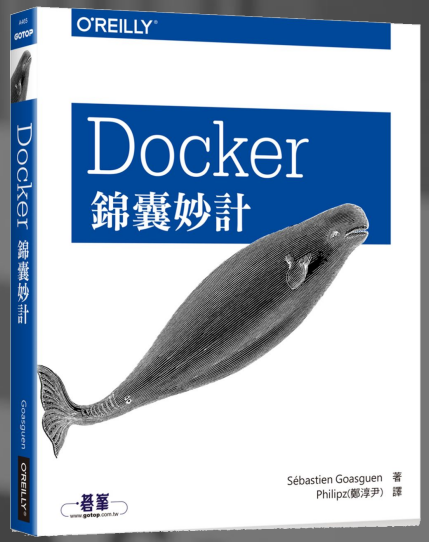
2014 COSCUP/iThome Summit 講者

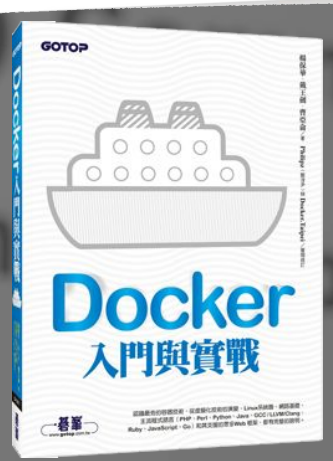
2015 Microsoft Azure 開發者大會 講者

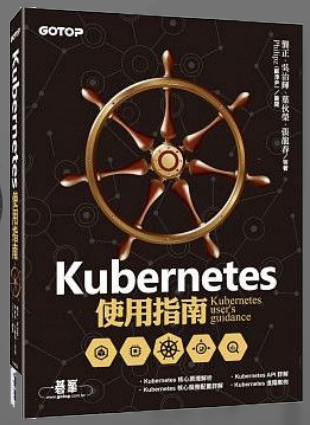
2016 COSCUP Docker 進階工作坊

2016 義守大學資工系 Docker 研習營







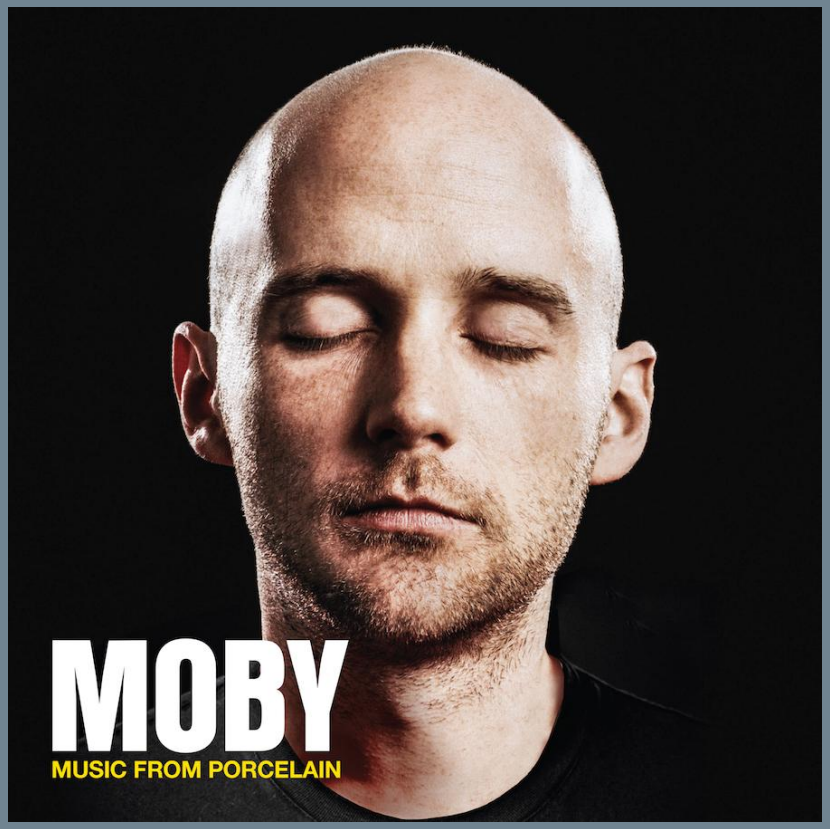


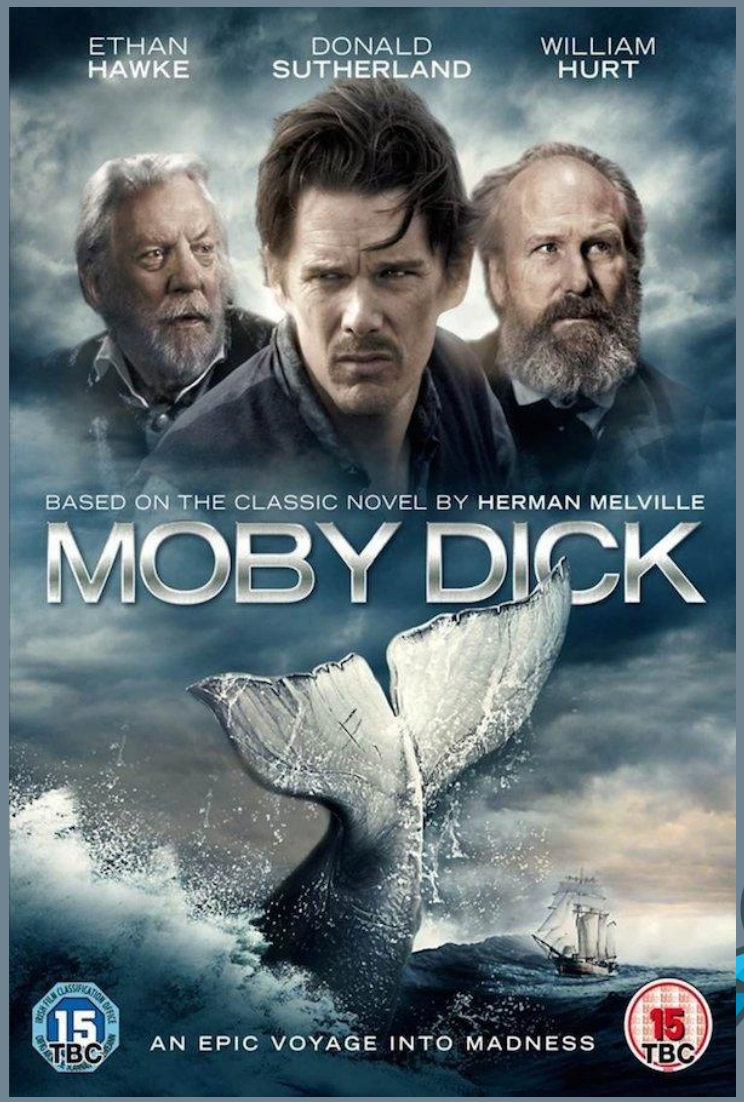
Today Topics

1. The differents between VMs and Container, Container lifecycle.
2. Docker ecosystem tools
3. Linux CLI、Docker CLI
4. Using Docker Engine
5. Docker Hub intoduction
6. Docker image & Docker hub autobuild
7. Deploy Docker image to Azure PaaS
8. Docker Network CLI & Docker Compose
9. Using Docker Compose

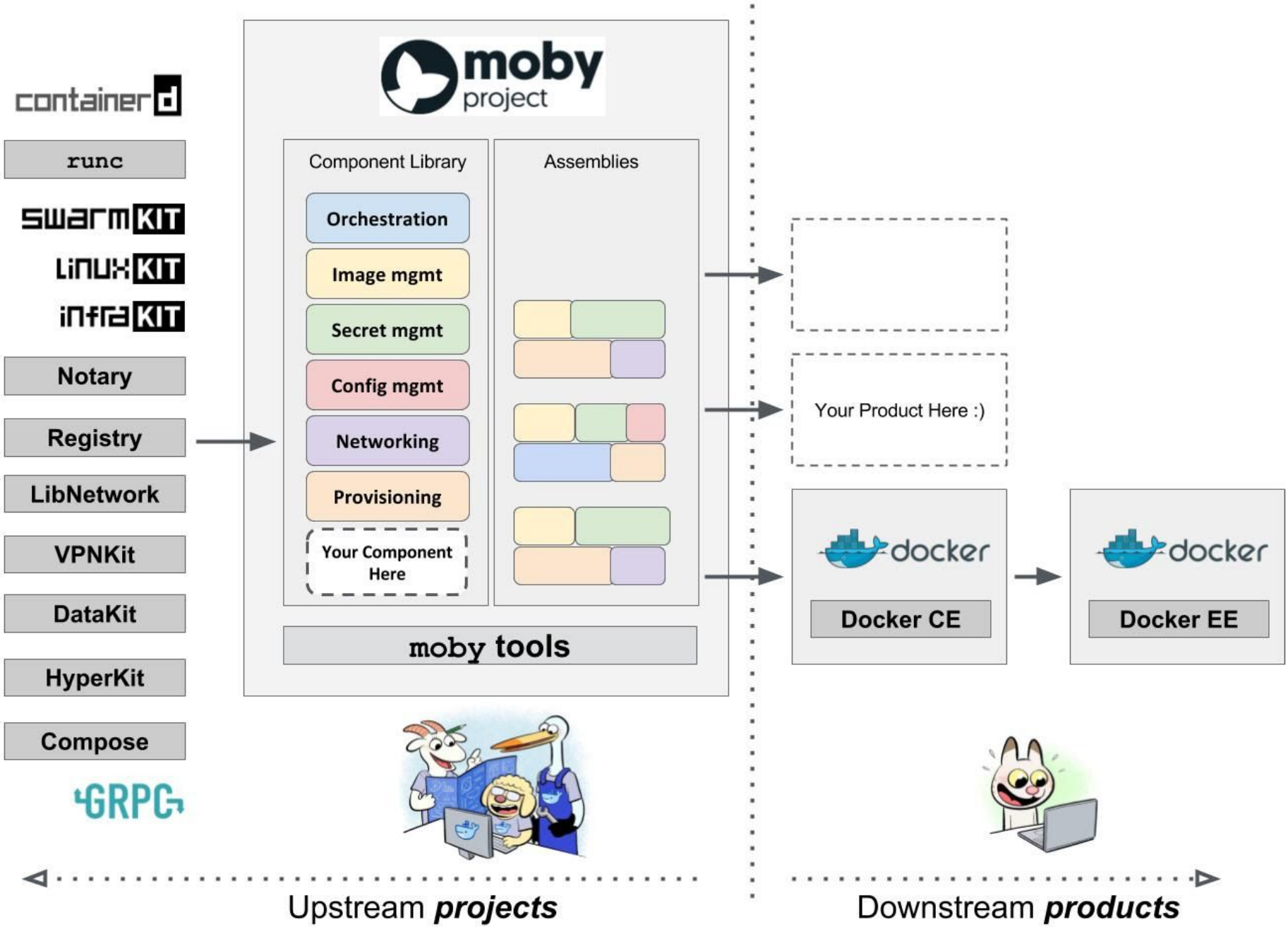
0. Docker now is called Moby

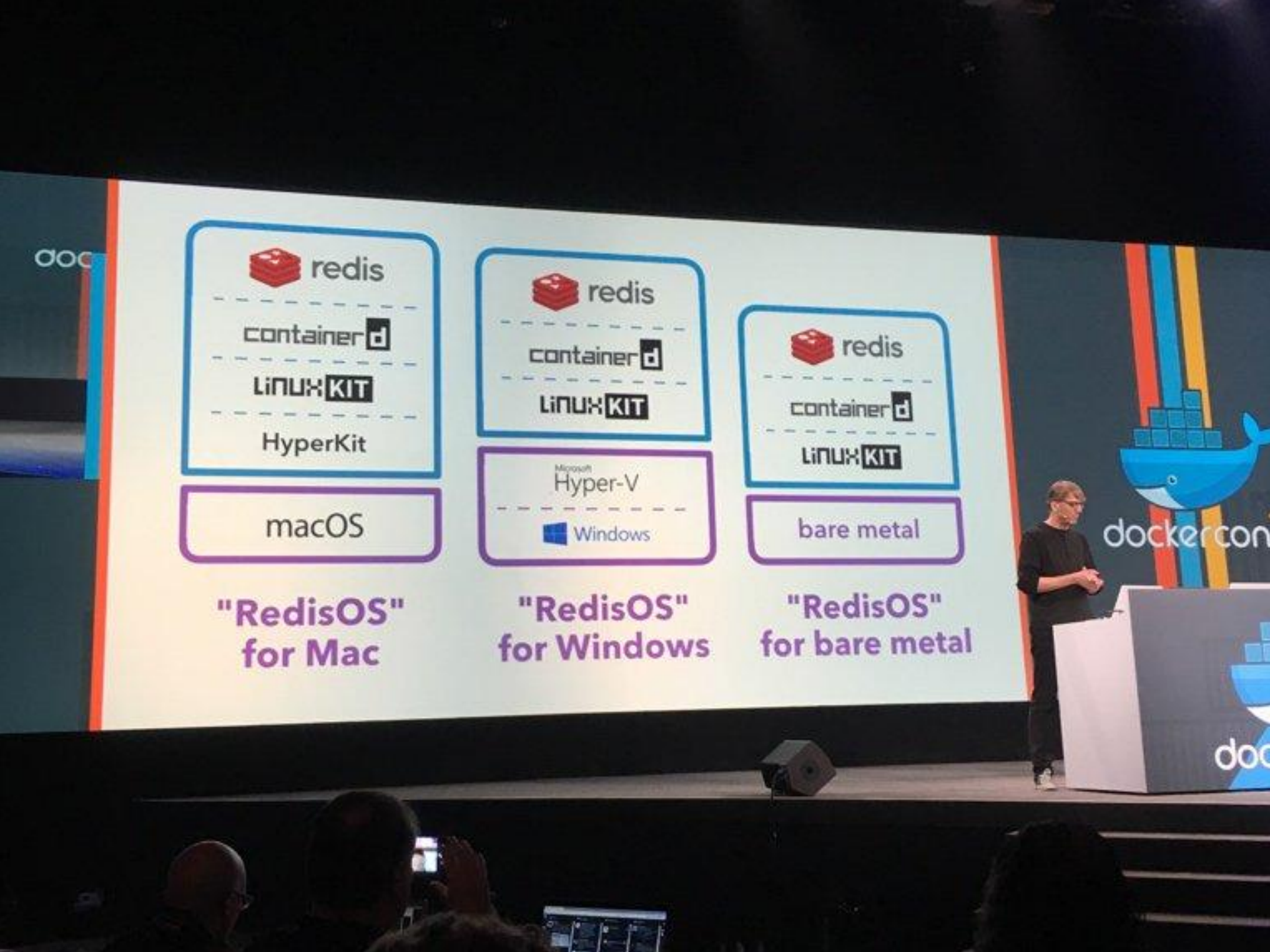
There are a lot of Moby





Blog & IThome description


containerd
runc
swarmKIT
LinuxKIT
infraKIT
Notary
Registry
LibNetwork
VPNKit
DataKit
HyperKit
Compose
GRPC
moby project
Component Library
Assemblies
Orchestration
Image mgmt
Secret mgmt
Config mgmt
Networking
Provisioning
Your Component Here
moby tools
Your Product Here :)
docker
Docker CE
docker
Docker EE
Upstream projects
Downstream products

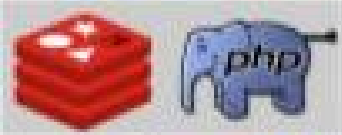
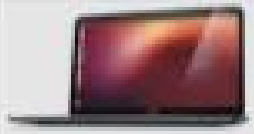
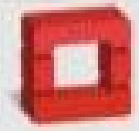

redis
containerd
LinuxKit
HyperKit
macOS
"RedisOS"
for Mac
redis
containerd
LinuxKit
Microsoft
Hyper-V
Windows
"RedisOS"
for Windows
redis
containerd
LinuxKit
bare metal
"RedisOS"
for bare metal
dockercon

1. Compare VM with Container

Virtualization History

- IBM zOS
- Virtual Hardware - VMware, KVM, Xen, VirtualBox
- Hardware-assisted virtualization
- Paravirtualization
- OS-level virtualization
  a. OpenVZ
  b. LXC
  c. Docker
- IaaS, PaaS, SaaS - Snapshot, Migration

The Martix of Hell

	?	?	?	?	?	?
	?	?	?	?	?	?
	?	?	?	?	?	?
	?	?	?	?	?	?
	?	?	?	?	?	?
	?	?	?	?	?	?
						

A Brief History of Containers

1979: Unix V7

2000: FreeBSD Jails

2005: Open VZ

2008: LXC

2013: LMCTFY

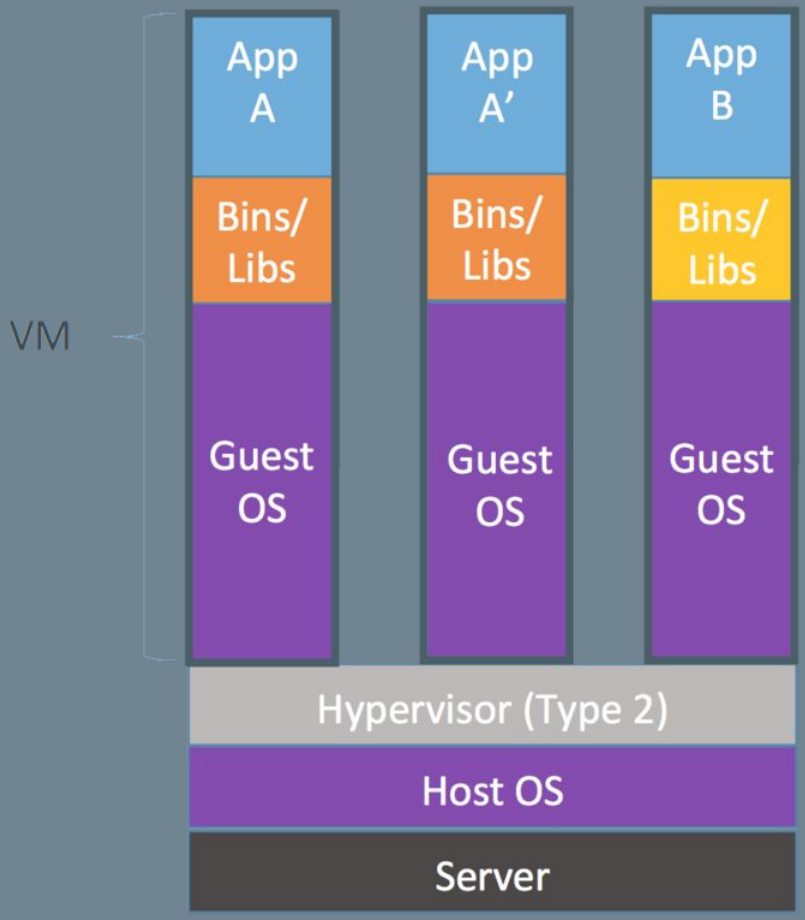
2013: Docker

2016: Windows Container

From: A Brief History of Containers: From 1970s chroot to Docker 2016

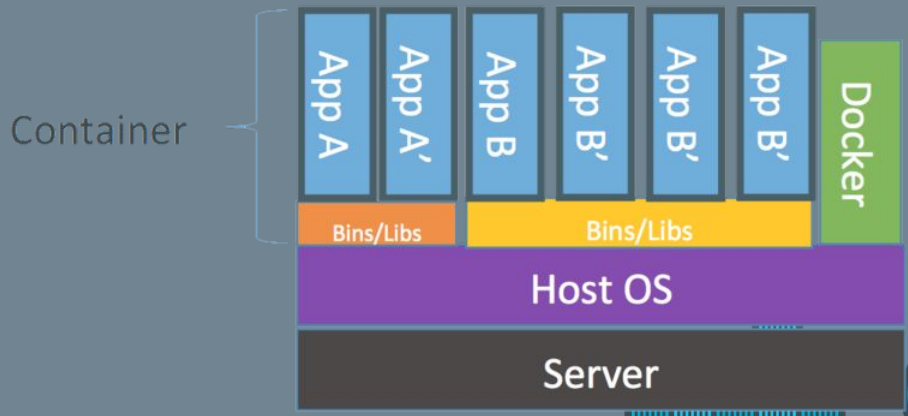
www.docker.com/enterprise

Containers vs. VMs



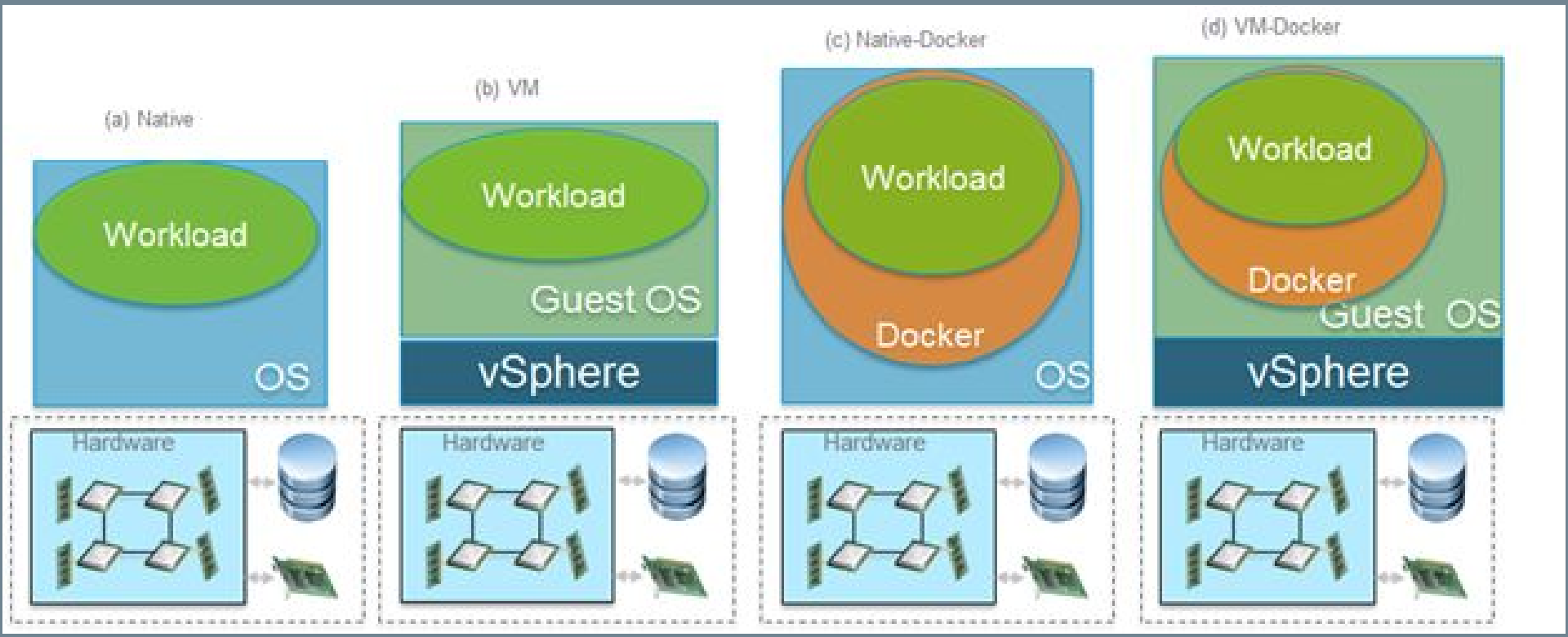
Containers are isolated, but share OS and, where appropriate, bins/libraries

…result is significantly faster deployment, much less overhead, easier migration, faster restart



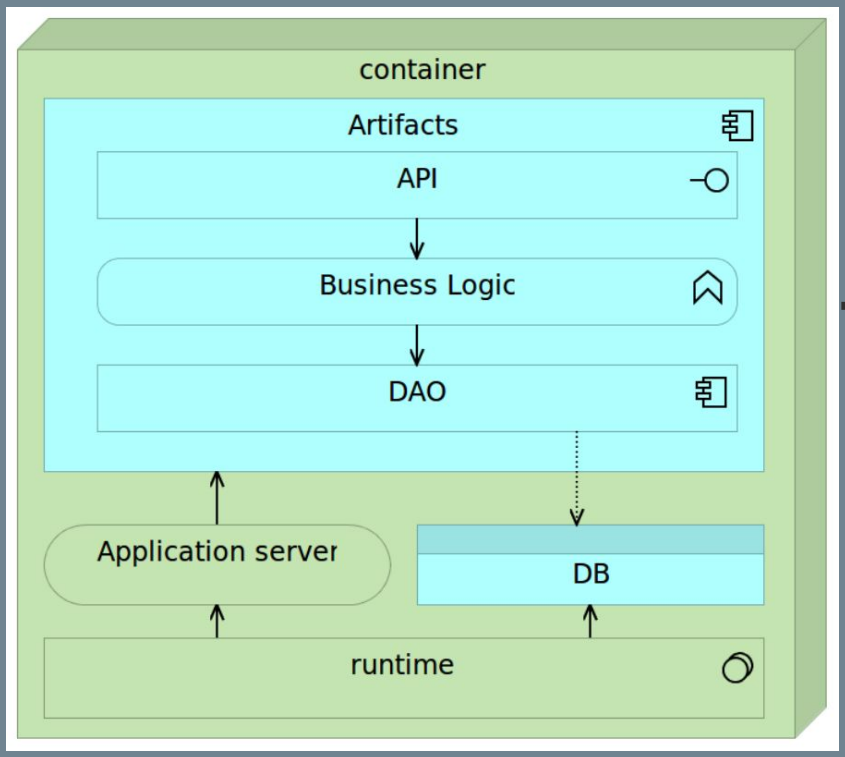
Blog description

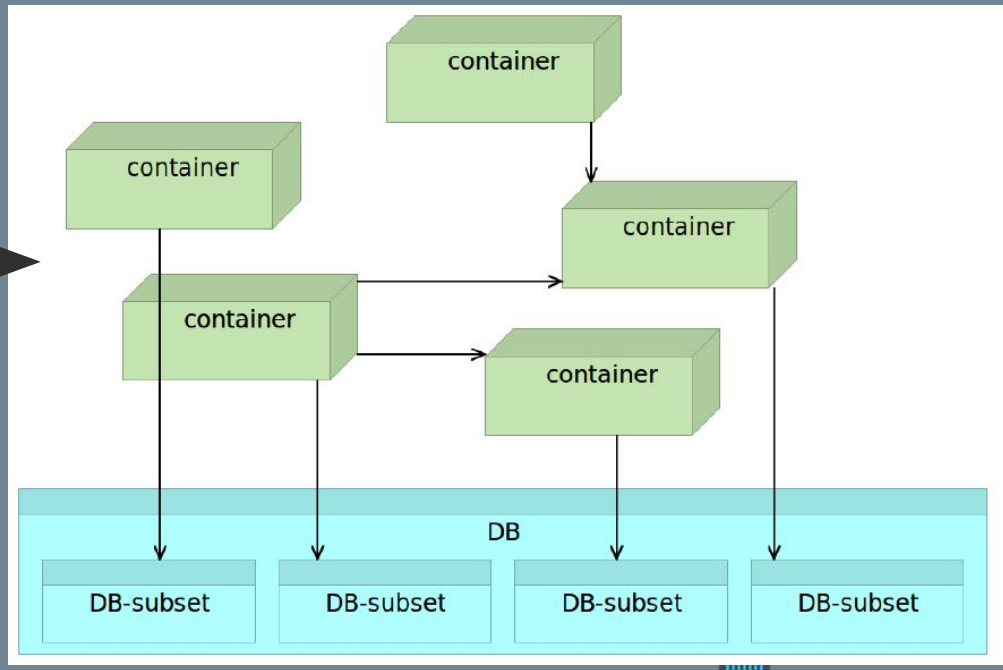
Containers vs. VMs



Blog description

Containers are not VMs





Blog description

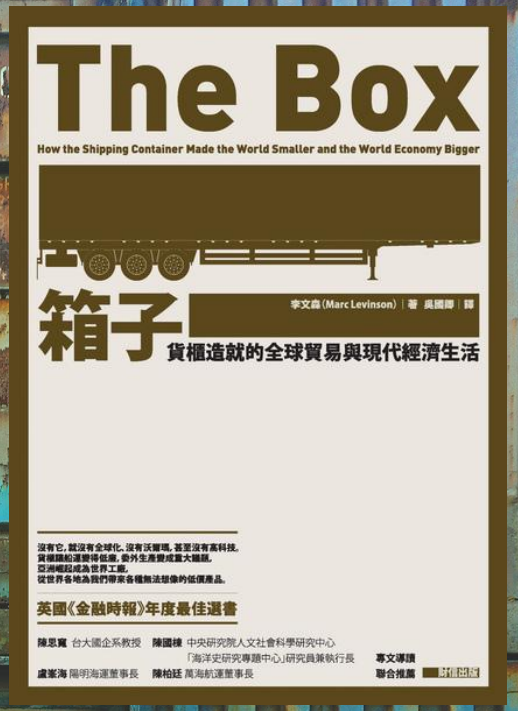
Container Principle

Real Container

- One Container
- One Customer
- One Commodity

Software Container

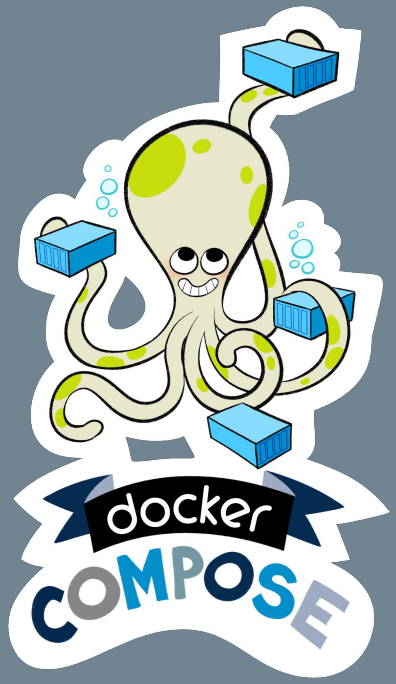
- One Container
- One Process

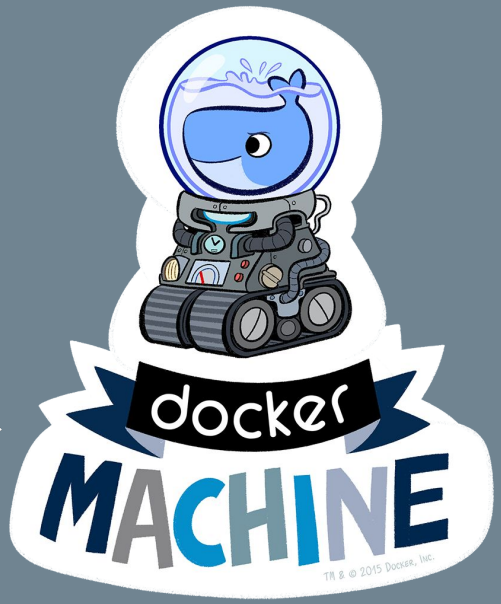


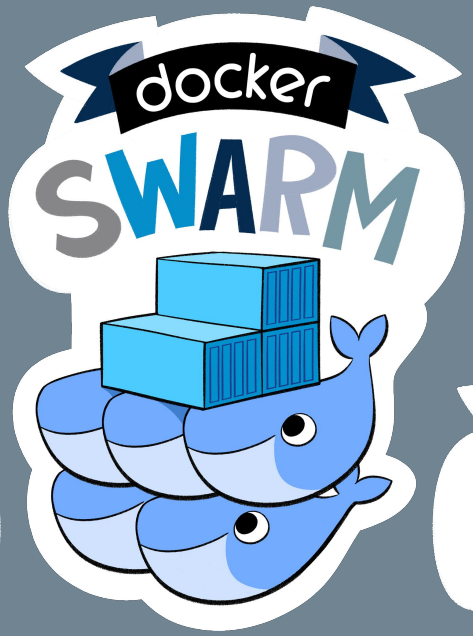
Adolf Hitler & Docker

2. Docker ecosystem tools

Docker Tools







Still No Silver Bullet

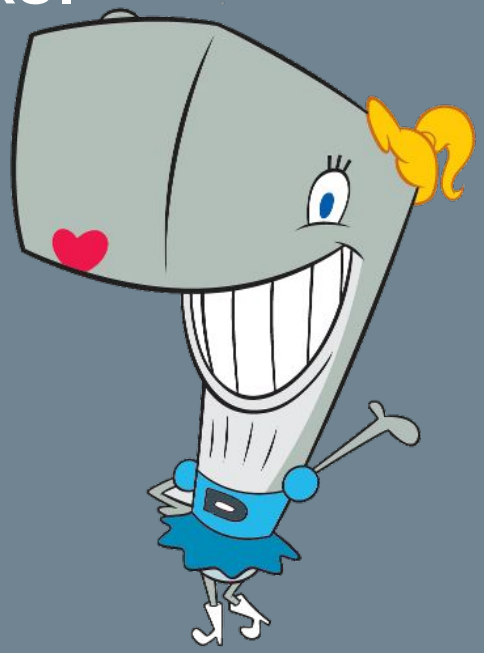
Container is one key element, not all.

DevOps pipeline process

Microservices, or other service stacks.

Infrastructure as Code

Business model



The Docker Stack

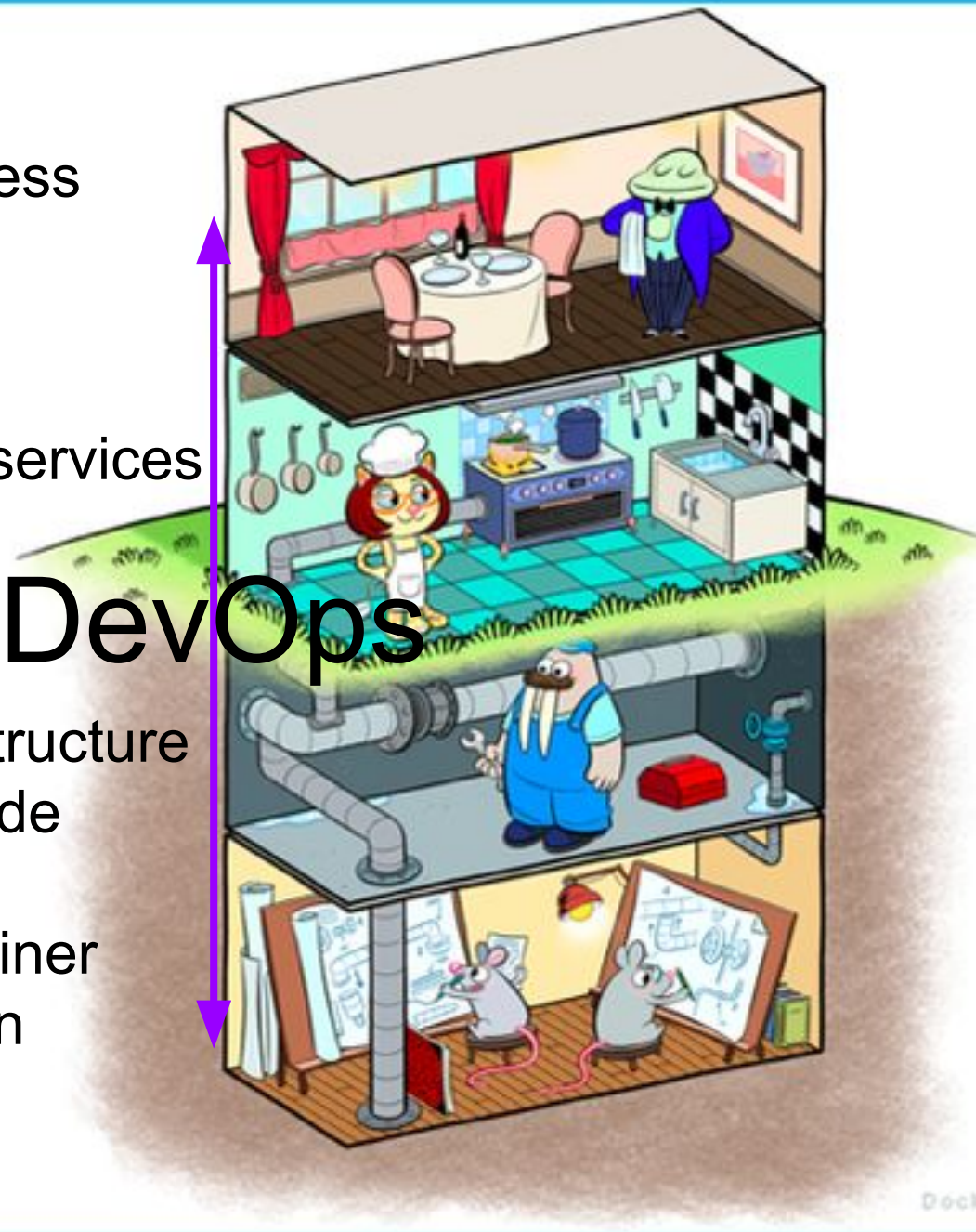
Business model

Microservices

DevOps

Infrastructure as Code

Container Design



The Docker Stack

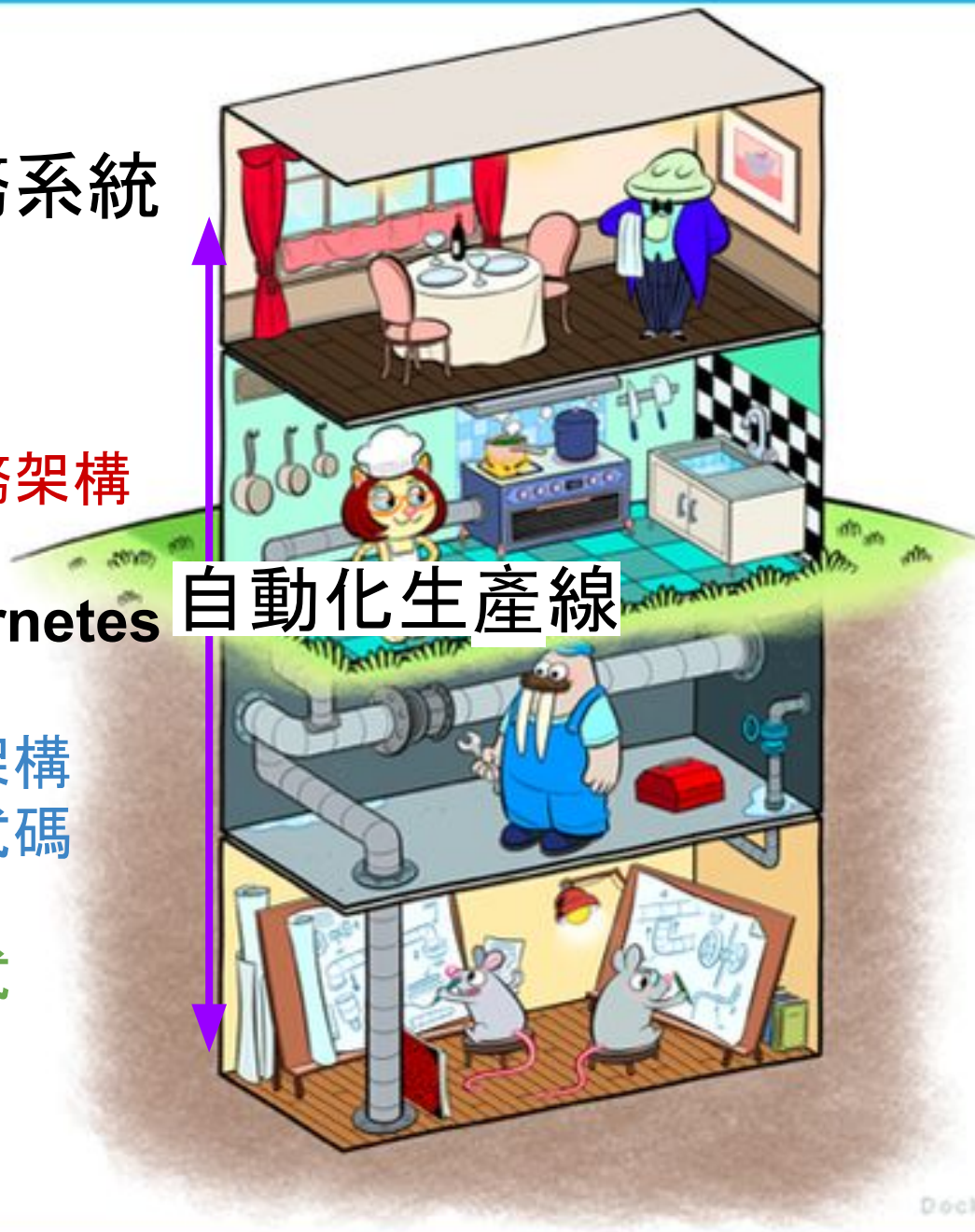
*業務系統

微服務架構

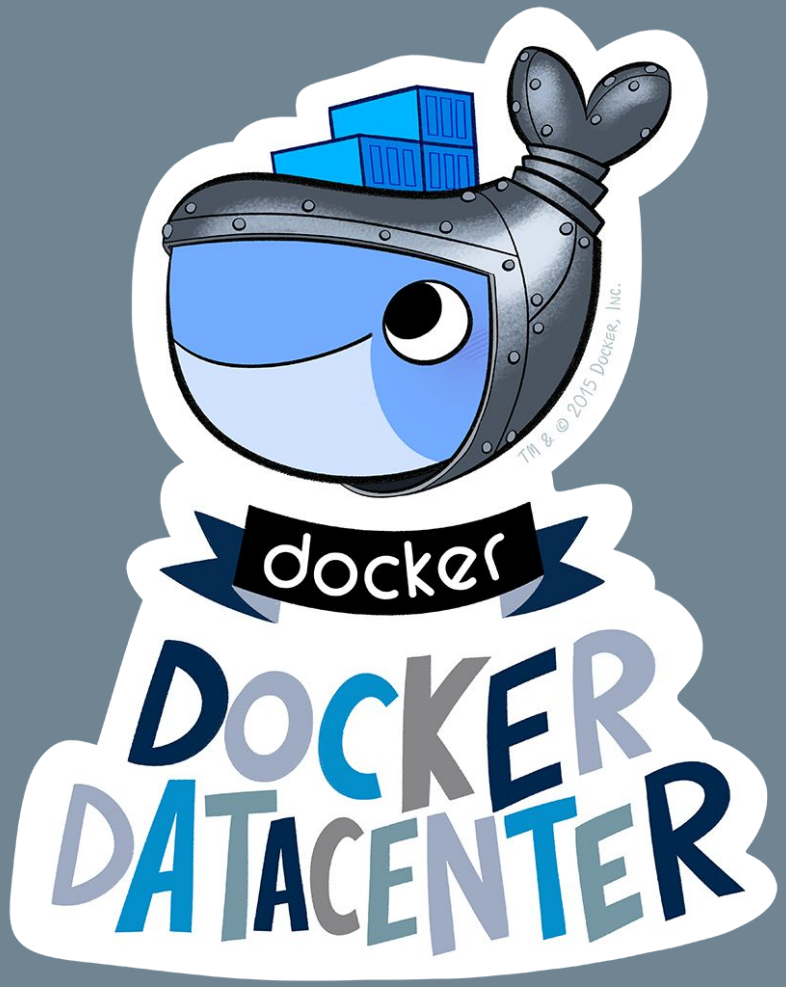
Kubernetes

基礎架構
即程式碼

容器式
設計



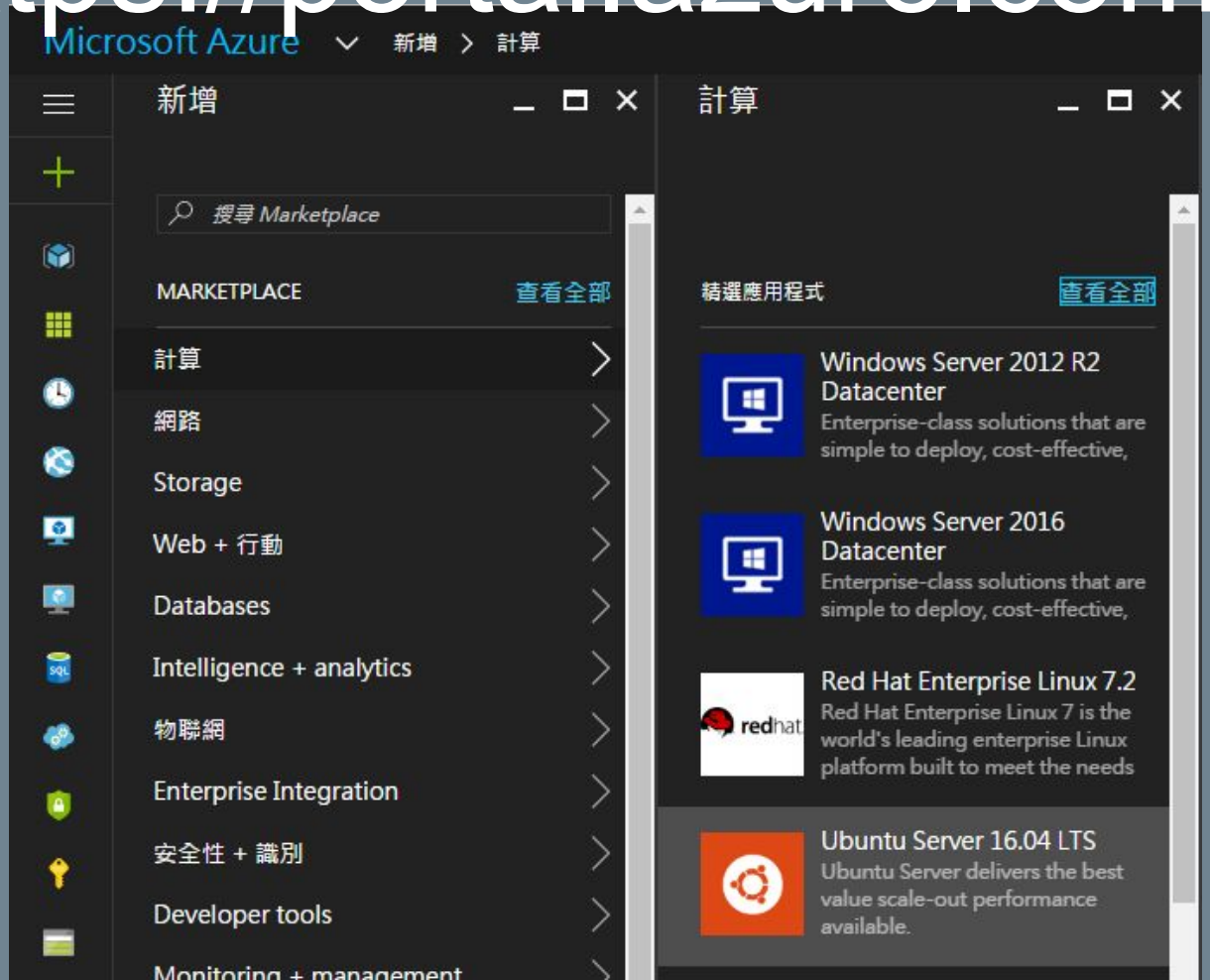
Docker Datacenter



3.1 Linux command-line

Microsoft Azure

https://portal.azure.com/



FILE COMMANDS

- ls - directory listing
- ls -al - formatted listing with hidden files
- cd dir - change directory to dir
- cd - change to home
- pwd - show current directory
- mkdir dir - create a directory dir
- rm file - delete file
- rm -r dir - delete directory dir
- rm -f file - force remove file
- rm -rf dir - force remove directory dir *
- cp file1 file2 - copy file1 to file2
- cp -r dir1 dir2 - copy dir1 to dir2; create dir2 if it doesn't exist
- mv file1 file2 - rename or move file1 to file2
  if file2 is an existing directory, moves file1 into directory file2
- ln -s file link - create symbolic link to file
- touch file - create or update file
- cat > file - places standard input into file
- more file - output the contents of file
- head file - output the first 10 lines of file
- tail file - output the last 10 lines of file
- tail -f file - output the contents of file as it grows, starting with the last 10 lines

SHORTCU

- Ctrl+C
- Ctrl+Z
  fg in t
- Ctrl+D
- Ctrl+W
- Ctrl+U
- Ctrl+R
- !! - re
- exit -

SYSTEM

- date
- cal -
- uptim
- w - d
- whoam
- finge
- uname
- cat /
- cat /
- man c
- df -
- du -
- free

- which

SEARCHING

- grep pattern files - search for pattern in files
- grep -r pattern dir - search recursively for pattern in dir
- command | grep pattern - search for pattern in the output of command

PROCESS MANAGEMENT

- ps - display your currently active processes
- top - display all running processes
- kill pid - kill process id pid
- killall proc - kill all processes named proc (use with extreme caution)
- bg - lists stopped or background jobs; resume a stopped job in the background
- fg - brings the most recent job to foreground
- fg n - brings job n to the foreground

COMP

- tar
  file
- tar
- tar
  Gzip
- tar
- tar
  comp
- tar
- gzip
  file
- gzip
  file

- !! - repeats the last command
- exit - log out of current session

it

SYSTEM INFO

- date - show the current date and time
- cal - show this month's calendar
- uptime - show current uptime
- w - display who is online
- whoami - who you are logged in as
- finger user - display information about user
- uname -a - show kernel information
- cat /proc/cpuinfo - cpu information
- cat /proc/meminfo - memory information
- man command - show the manual for command
- df - show disk usage
- du - show directory space usage
- free - show memory and swap usage
- whereis app - show possible locations of app
- which app - show which app will be run by default

INSTA

- Insta
  ./con
  make
  make

- dpkg
- rpm -

Cli

- which app - show which app will be run by default

cli commands

COMPRESSION

- tar cf file.tar files - create a tar named file.tar containing files
- tar xf file.tar - extract the files from file.tar
- tar czf file.tar.gz files - create a tar with Gzip compression
- tar xzf file.tar.gz - extract a tar using Gzip
- tar cjf file.tar.bz2 - create a tar with Bzip2 compression
- tar xjf file.tar.bz2 - extract a tar using Bzip2
- gzip file - compresses file and renames it to file.gz
- gzip -d file.gz - decompresses file.gz back to file

make
make

- dpkg
- rpm -

FILE

- chmod
to oc

group

Examp

chmod
chmod
For m

3.2 Docker command-line

Install Docker

Install Docker on Ubuntu

or

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

and

docker run hello-world

Docker Management commands

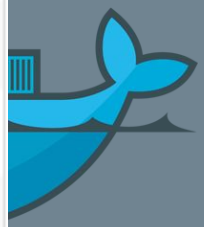
Command	Description
dockerd	Launch the Docker daemon
info	Display system-wide information
inspect	Return low-level information on a container or image
version	Show the Docker version information

Docker image commands

Command	Description
build	Build an image from a Dockerfile
commit	Create a new image from a container's changes
history	Show the history of an image
images	List images
import	Import the contents from a tarball to create a filesystem image
load	Load an image from a tar archive or STDIN
rmi	Remove one or more images
save	Save images to a tar archive
tag	Tag an image into a repository

Docker container commands (1/2)

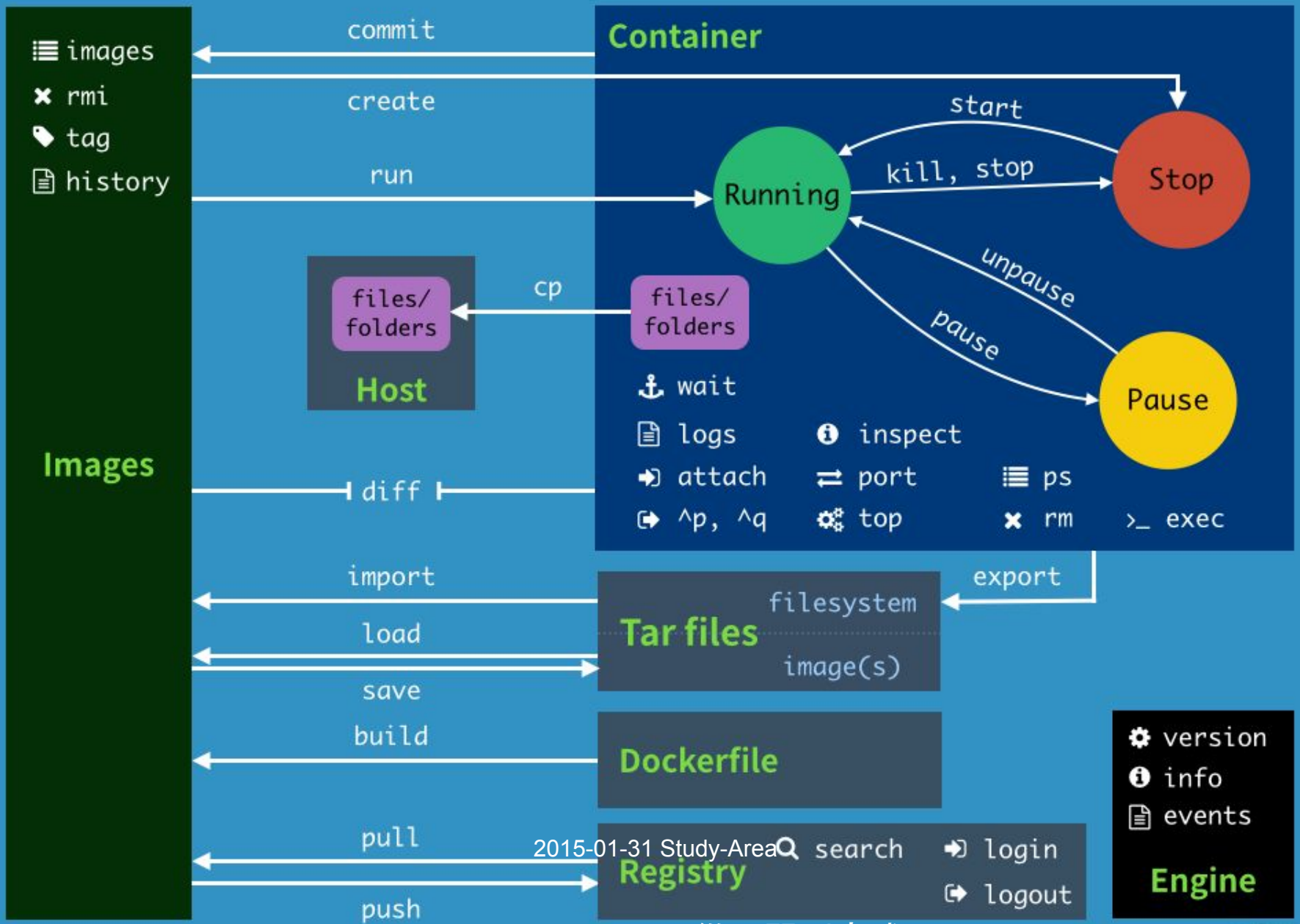
Command	Description
attach	Attach to a running container
cp	Copy files/folders from a container to a HOSTDIR or to STDOUT
create	Create a new container
diff	Inspect changes on a container's filesystem
events	Get real time events from the server
exec	Run a command in a running container
export	Export a container's filesystem as a tar archive
kill	Kill a running container
logs	Fetch the logs of a container
pause	Pause all processes within a container
port	List port mappings or a specific mapping for the container
ps	List containers



Docker container commands (2/2)

Command	Description
rename	Rename a container
restart	Restart a running container
rm	Remove one or more containers
run	Run a command in a new container
start	Start one or more stopped containers
stats	Display a live stream of container(s) resource usage statistics
stop	Stop a running container
top	Display the running processes of a container
unpause	Unpause all processes within a container
update	Update configuration of one or more containers
wait	Block until a container stops, then print its exit code

Docker Commands Diagram



2015-01-31 Study-Area

Gitbook: Docker 從入門到實踐

@fntsrlike

4. Docker Engine Playground

Azure Firewall

docker run -d -p 80:80 nginx

2 連入安全性規則

優先順序	名稱	來源	目的地	服務	動作
1000	default-allow-ssh	Any	Any	SSH (TCP/22)	Allow
1010	web	Any	Any	HTTP (TCP/80)	Allow

docker run -ti --rm -p 80:80 nginx

docker run -ti --rm -p 80:80 nginx bash

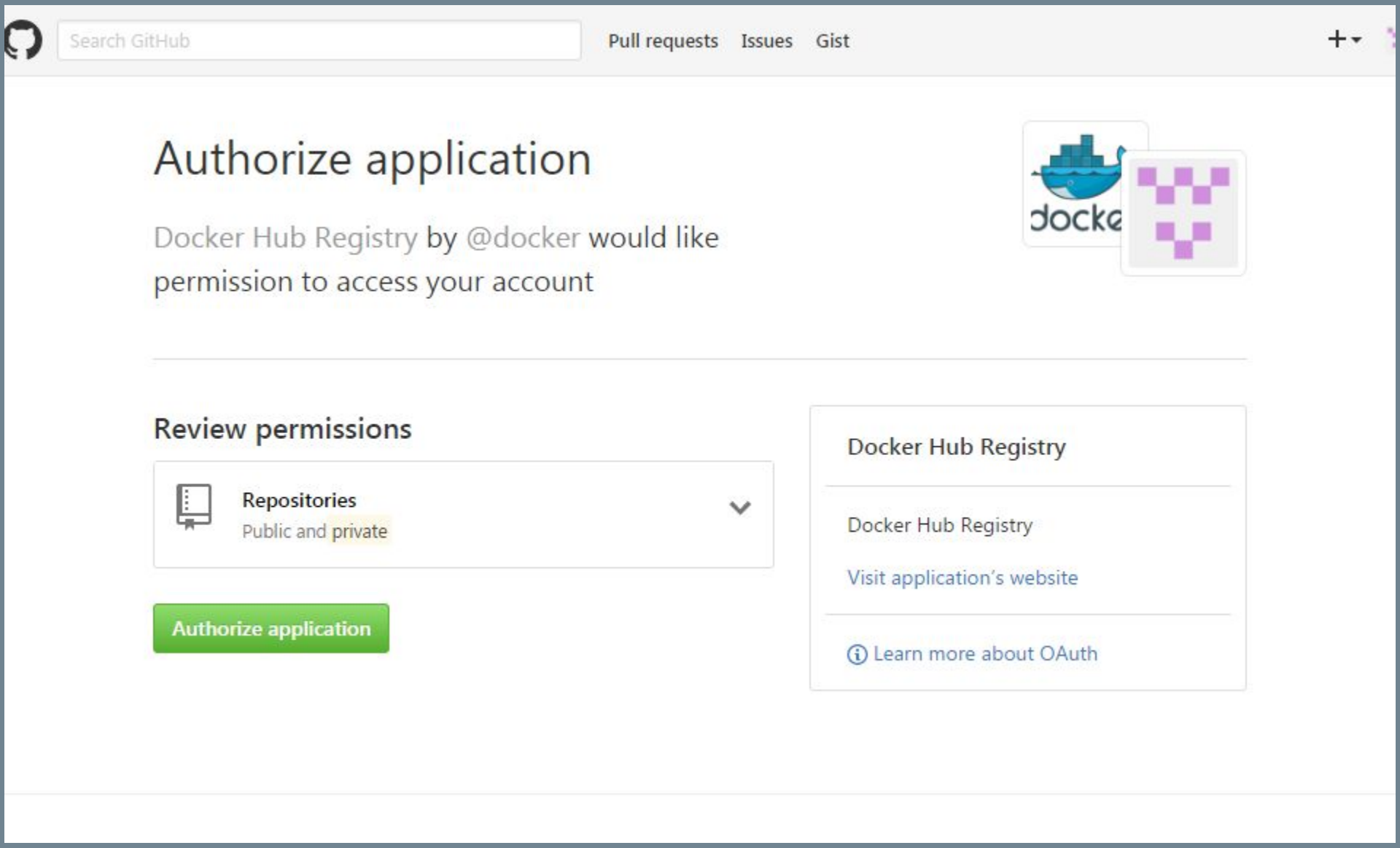
Azure DNS Setting

5. Docker Hub introduction

Docker Hub = App Store

- Public Docker Registry
- One free private repo.
- Auto-build & Webhook
- Security Scanning is not free.

GitHub & Docker Hub



Vulnerability Analysis

CoreOS Clair

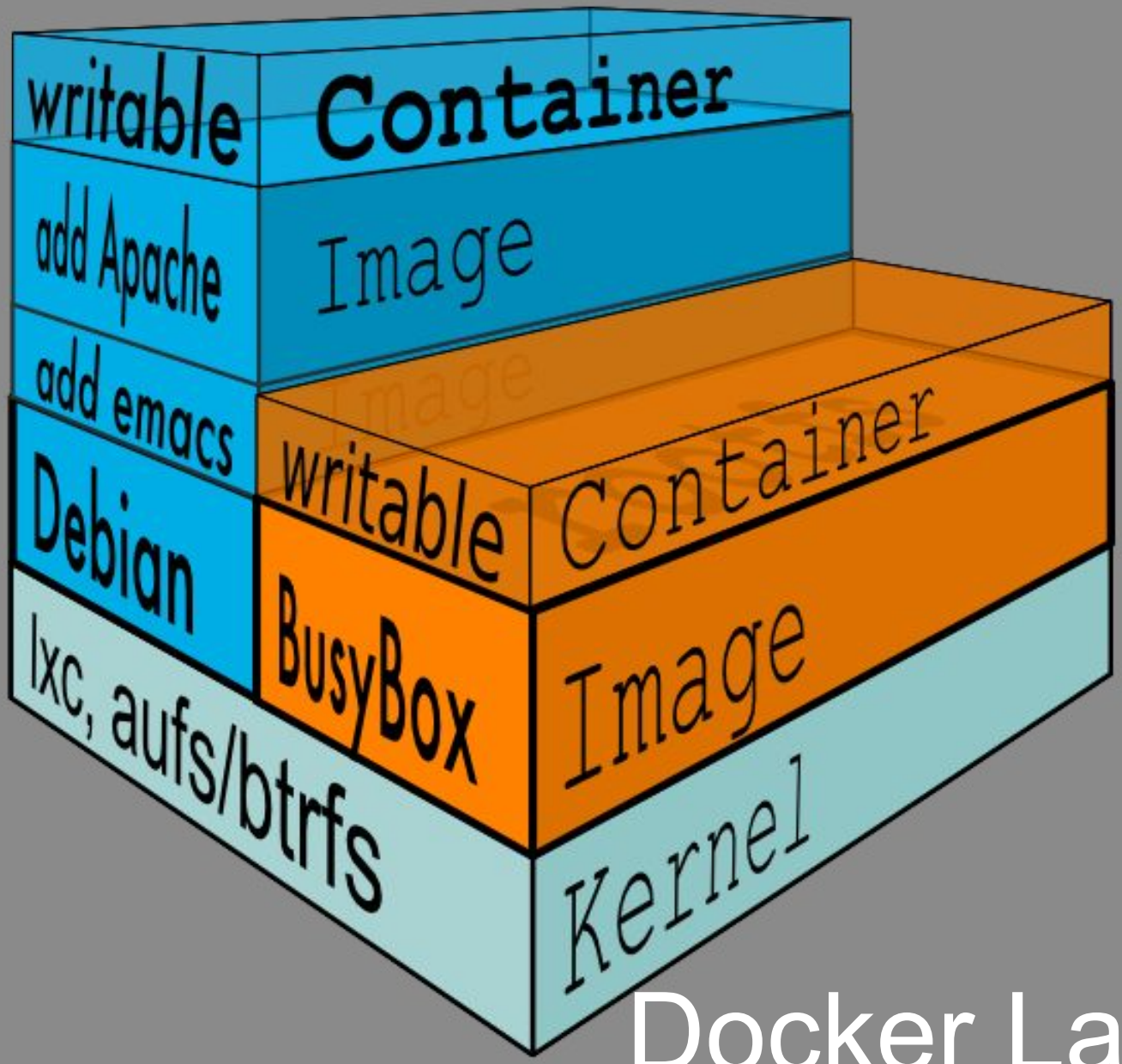
Anchore

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

6.1 Docker image & Dockerfile



Docker Layers

Create Docker image

1. Docker commit
2. Dockerfile - docker build
3. Docker Hub auto-build
4. FROM scratch
5. Based on others, ubuntu, alpine...

Example:

https://github.com/docker/labs/tree/master/beginner/static-site

```
docker save busybox > busybox.tar
docker load < busybox.tar
```

Dockerfile Reference

Same folder, docker build .

docker build -f /other/folder/file .

Add tag, docker build -t TAG_NAME .

Sample:

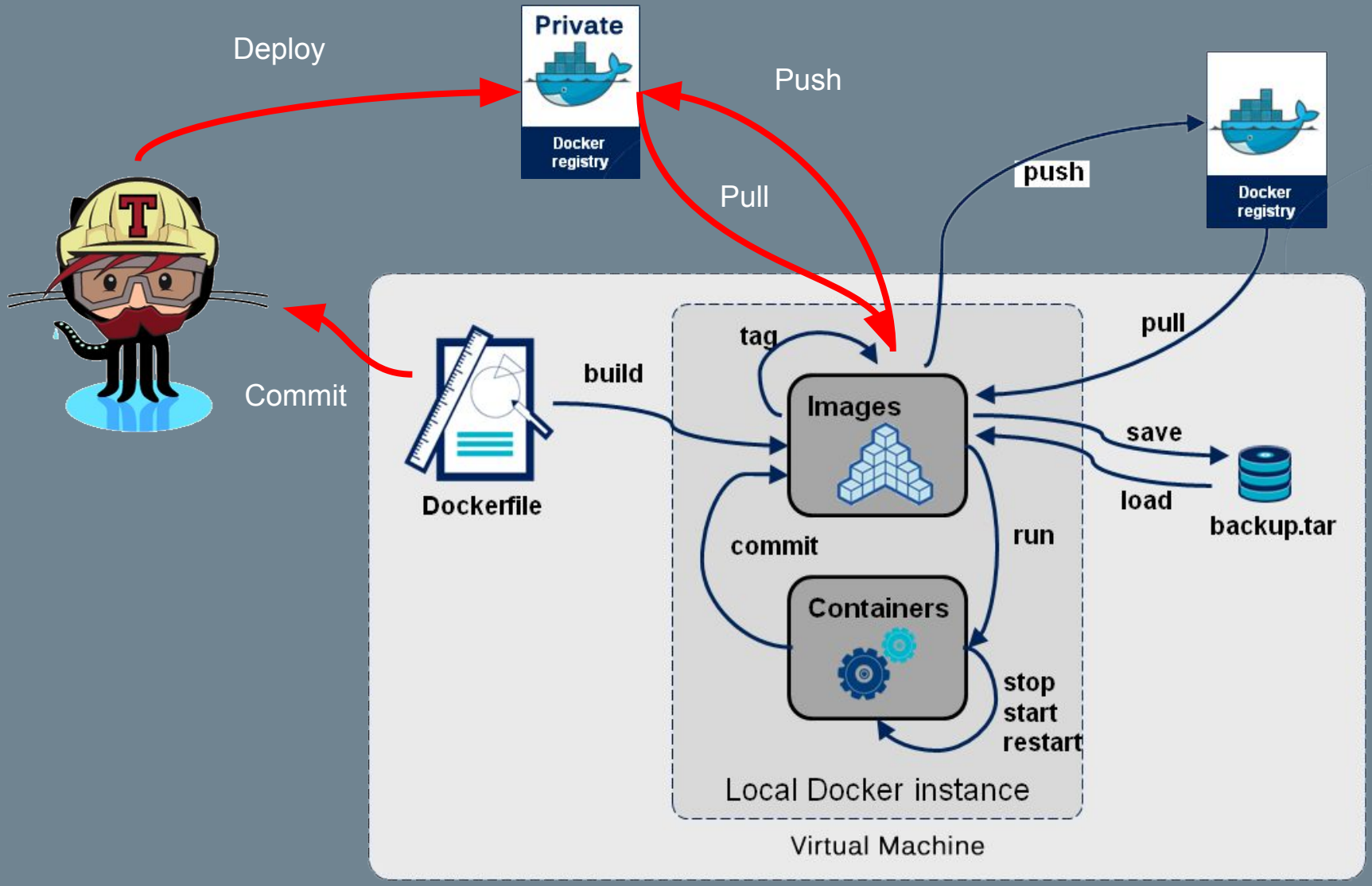
```
FROM debian:jessie
MAINTAINER docker "docker@nginx.com"
RUN apt-get update && apt-get install -y nginx
CMD ["nginx", "-g", "daemon off;"]
```

Healthcheck from 1.12

Dockerfile Practice

1. Must be “Dockerfile”.
2. Use a .dockerignore file, like .gitignore.
3. Minimize the number of layers
4. Sort multi-line arguments
5. ADD or COPY, in detail
6. CMD or ENTRYPOINT
7. ONBUILD
8. EXPOSE and USER
9. WORKDIR and ENV

Use Scenario



6.2 Docker Hub Auto-build

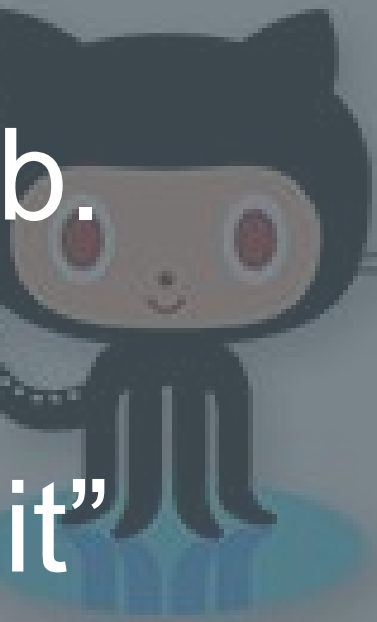
Dockerfile

Sample:

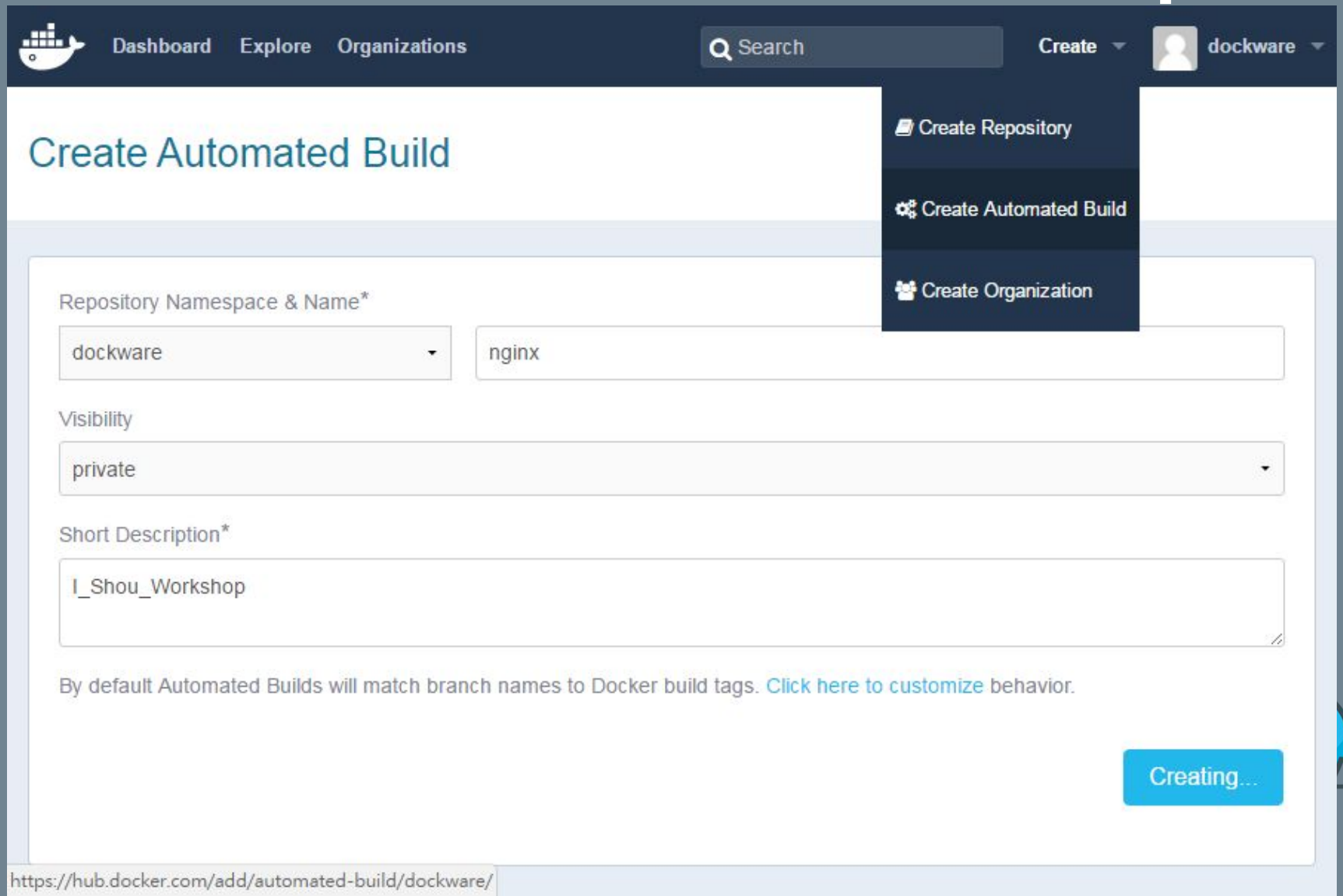
```
FROM debian:jessie
MAINTAINER docker "docker@nginx.com"
RUN apt-get update && apt-get install -y nginx
CMD ["nginx", "-g", "daemon off;"]
```

Git Workflow

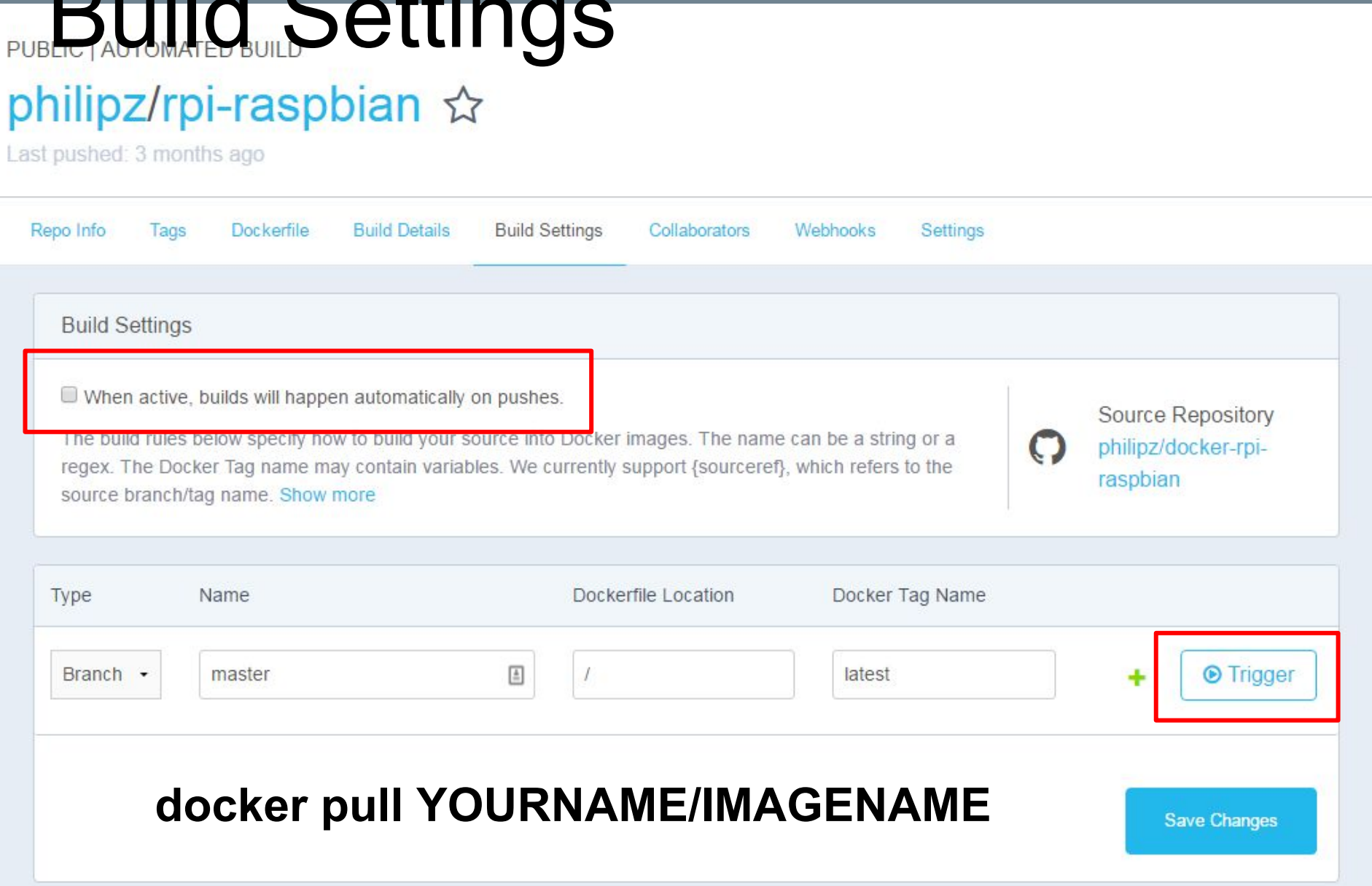
1. git init or init on GitHub.
2. git add Dockerfile
3. git commit -m “First init”
4. git remote add origin
   https://github.com/YOURNAME/docker_build.git
5. git push origin master



Create Auto-build Repo.

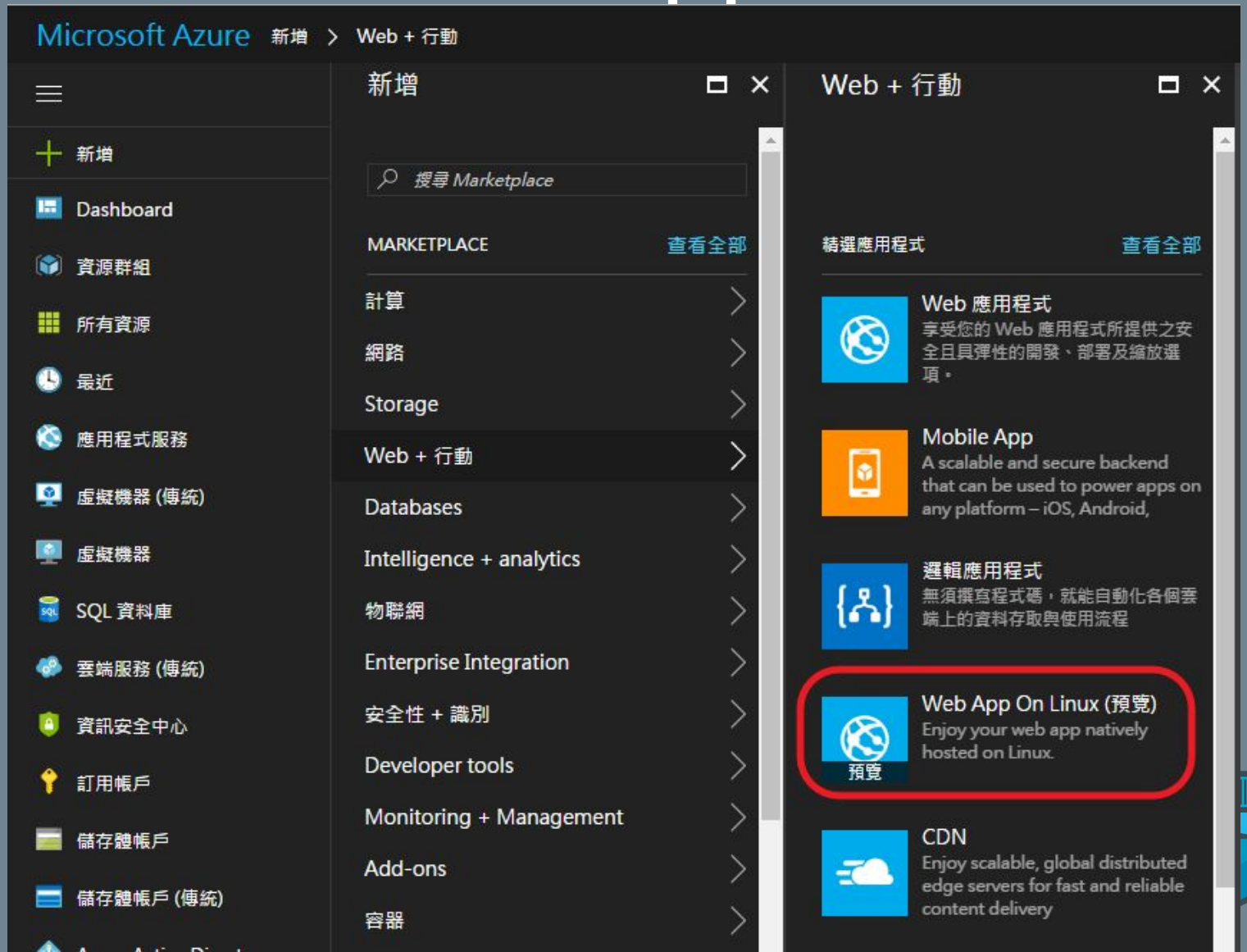


Build Settings



7. Deploy Docker image to Azure PaaS

Azure Web App on Linux



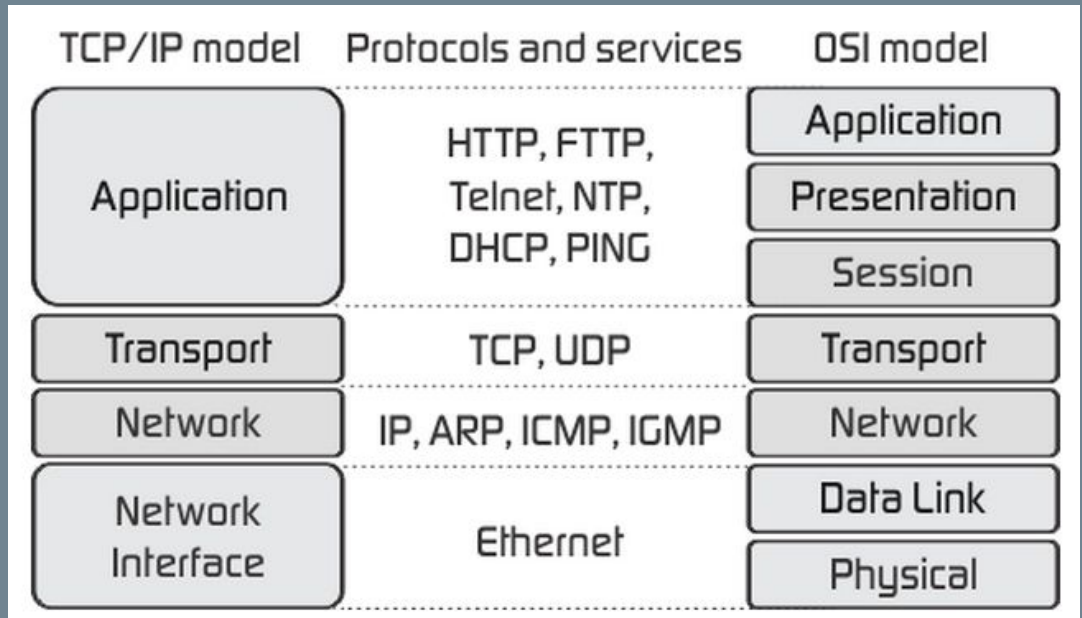
Use Docker image for Web AP

Azure PaaS Price Models

8.1 Docker Network command-line

TCP/IP Foundation

www.google.com, www is hostname, google.com is domain name.
Localhost: 127.0.0.1
TCP/UDP Port: 0-65535 = 2^16,
but 0 is a reserved port.
Private IP:
10.0.0.0/8
172.16.0.0/12 ~ 172.31.0.0/12
192.168.0.0/16



Network and connectivity commands

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 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 1

$ 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

Exercise 2

$ 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

Exercise 3

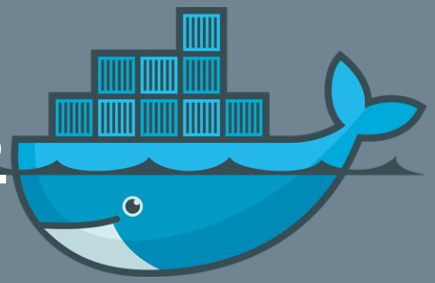
```
$ docker network create wp_db

$ 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
```

8.2 Docker Volume command-line

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
```

SDS

Software Define Storage

EMC: REX-Ray

Azure: File storage

AWS: Elastic File System

9.1 Docker Compose command-line

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
```

Docker Compose commands (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 commands (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 File Reference

Run Multi-container at the same time.

Must be docker-compose.yml

Same folder, docker-compose up -d

Docker will build the Dockerfile of subfolders.

Docker Network, Volume supports

1.13 has supported Swarm mode.

Quickstart: Compose and WordPress

Kompose = Kubernetes + Compose

9.2 Using Docker Compose

Compose File Sample (1/2)

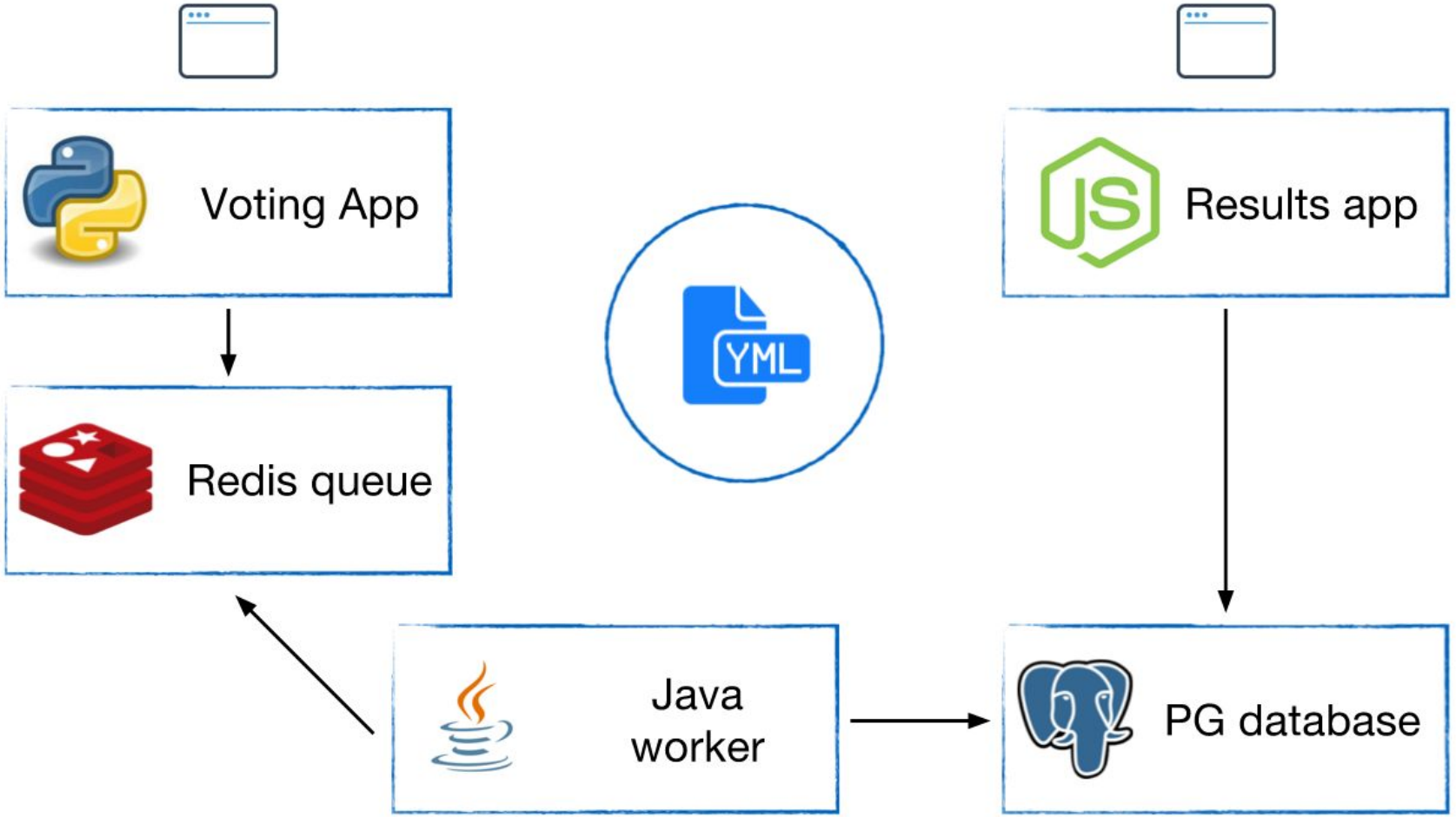
```
version: '2'
services:
  db:
    image: mysql:5.7
    volumes:
      - db_data:/var/lib/mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: wordpress
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: wordpress
```

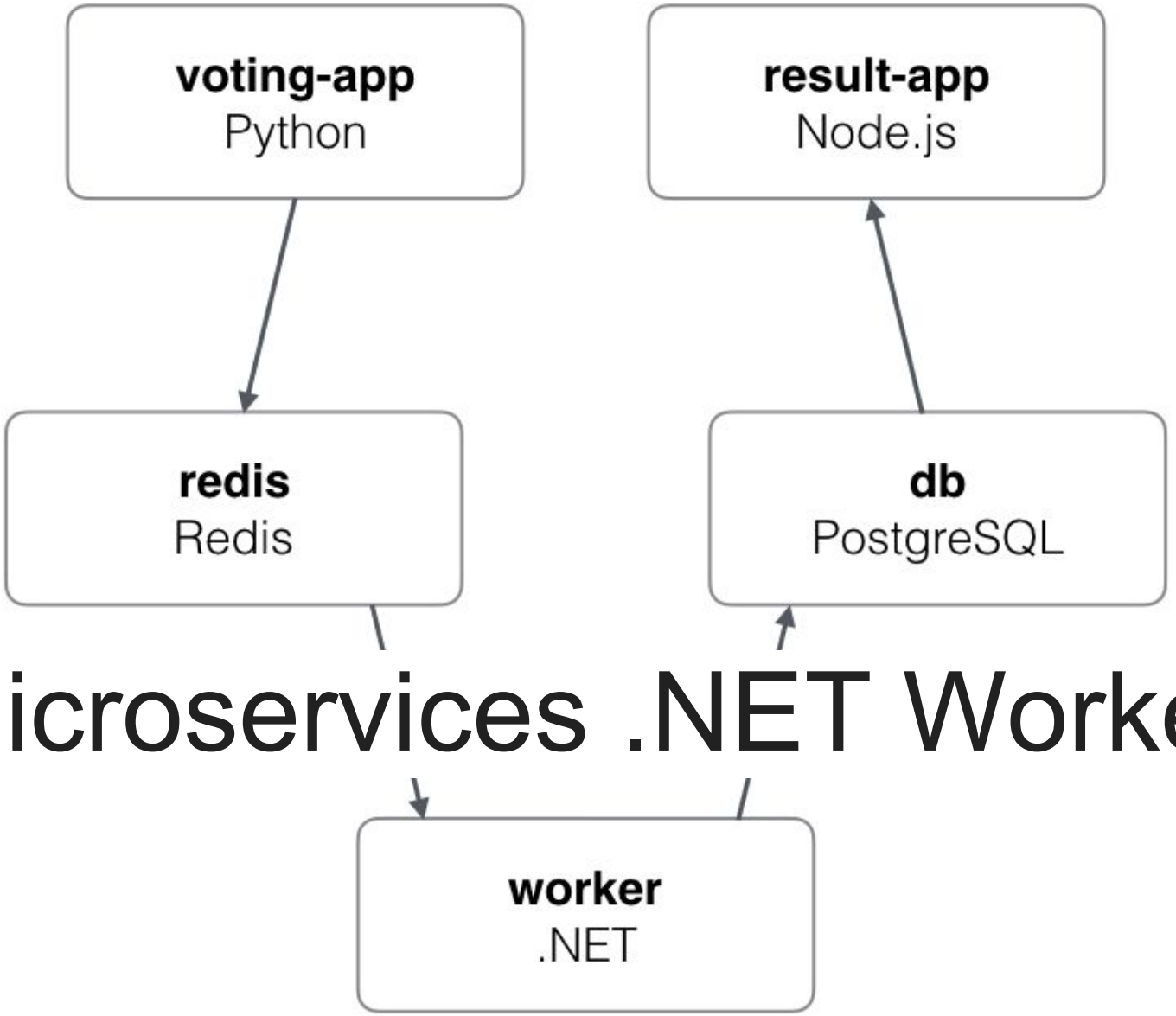
Compose File Sample (1/2)

```
  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 ***
```

Microservices Java Worker

[Docker Birthday #3 training]





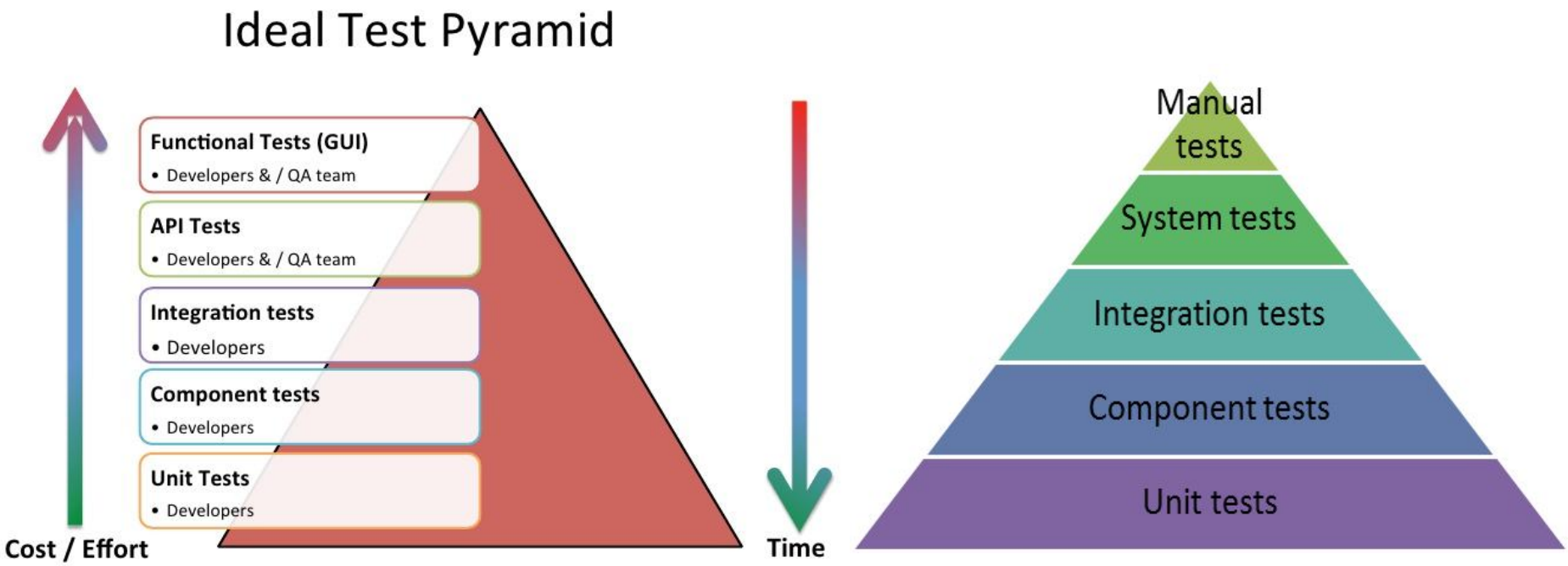
Microservices .NET Worker

Docker Birthday #3 training

Docker Compose & CI/CD

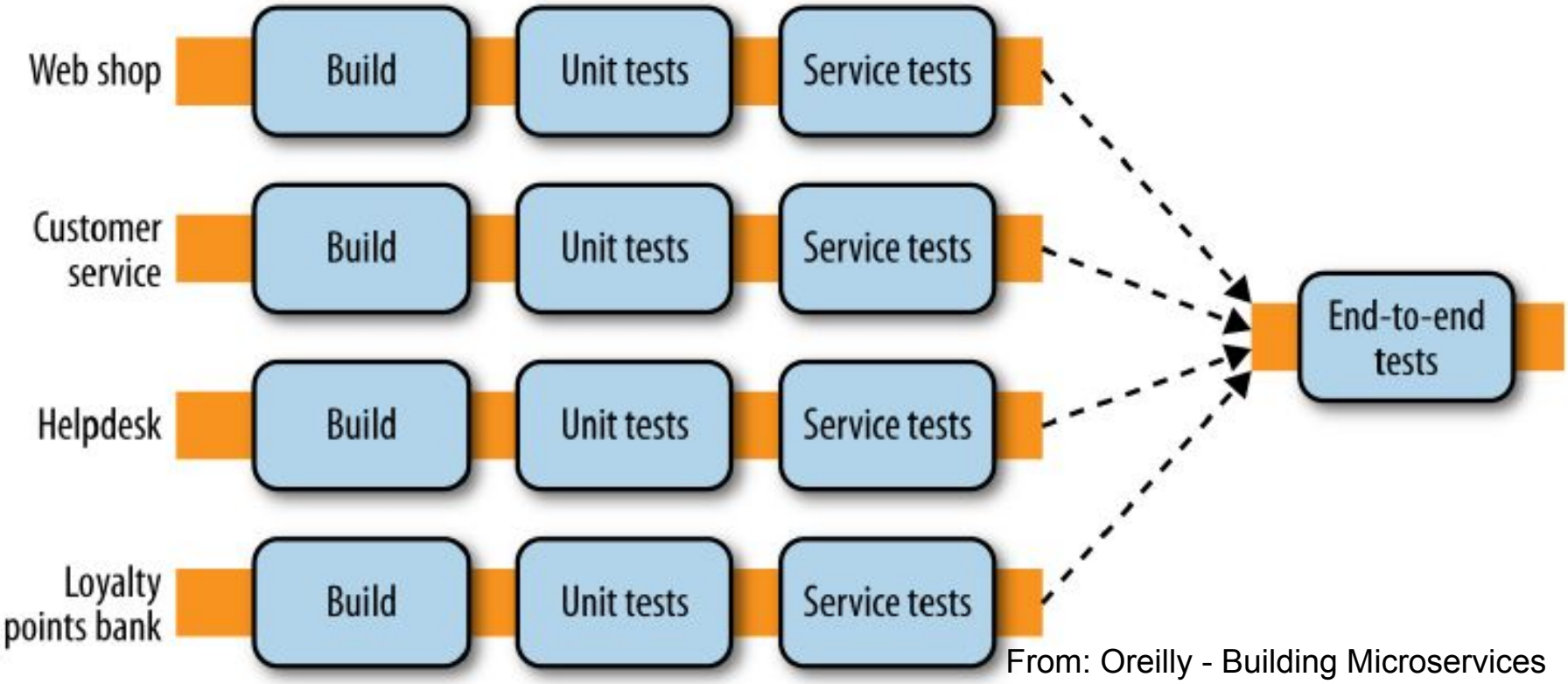
~~GitHub~~, ~~CircleCI~~, ~~Docker Hub~~ = GitLab

Testing level? Coding effort? Env. build-up



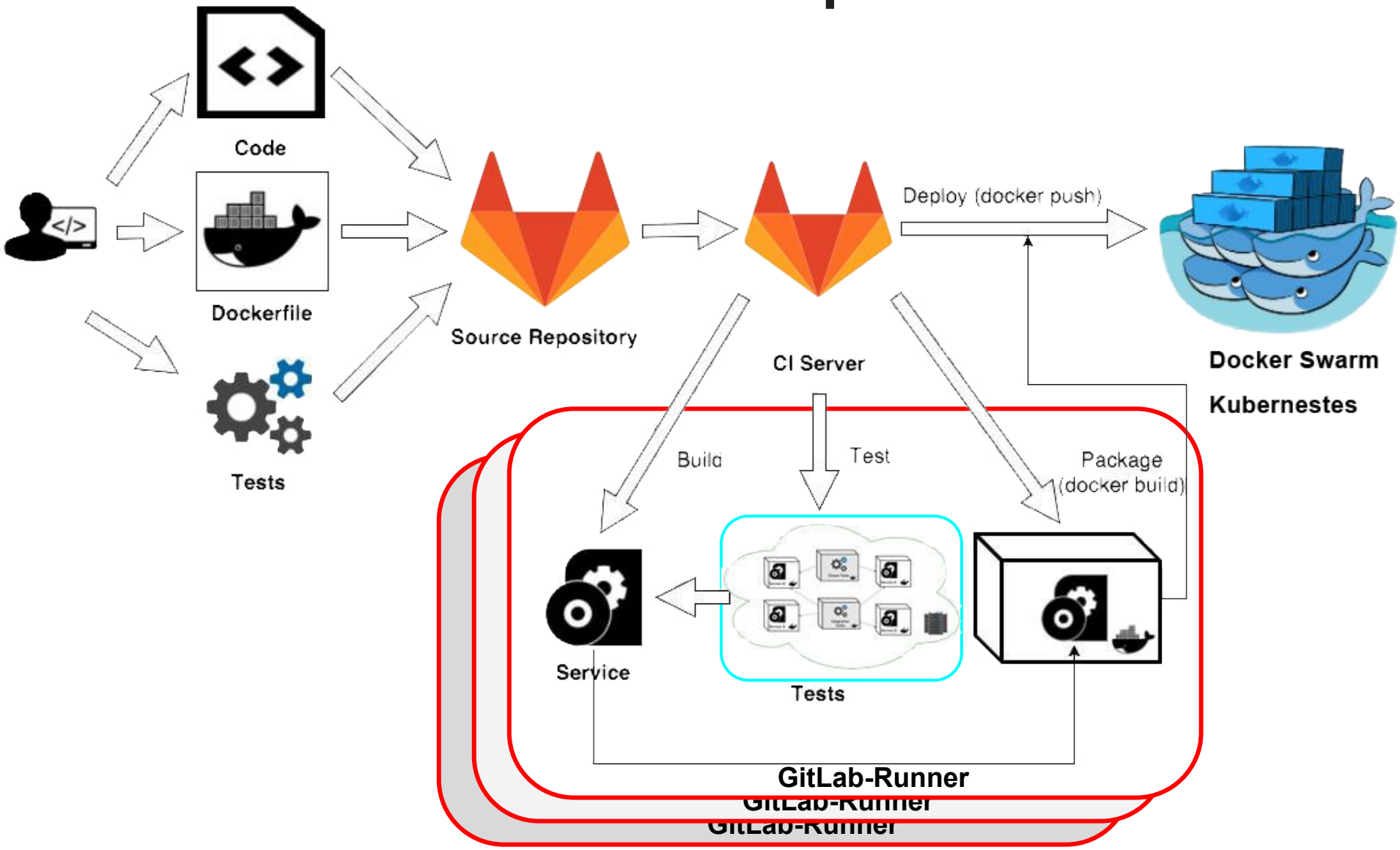
End to End Tests

CI with Docker Compose is easy to implement.

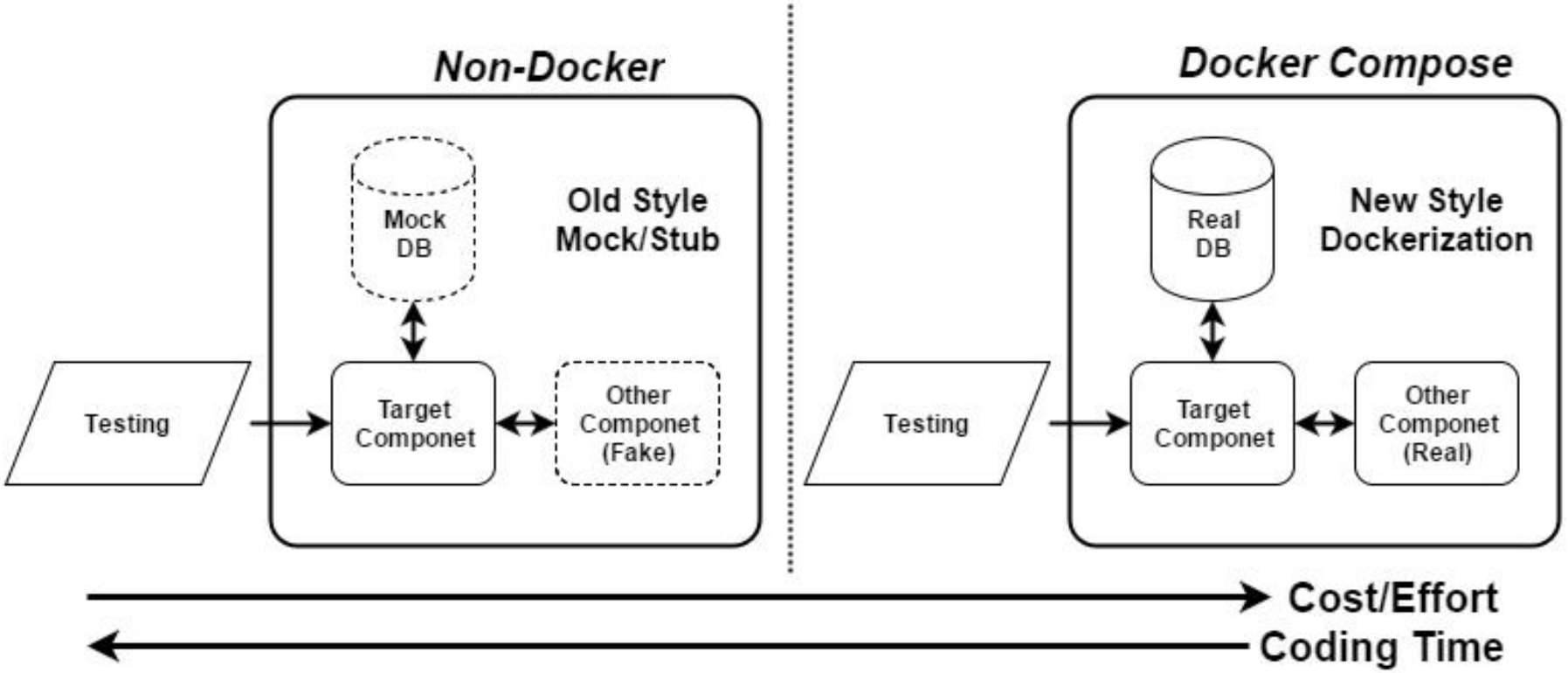


From: Oreilly - Building Microservices

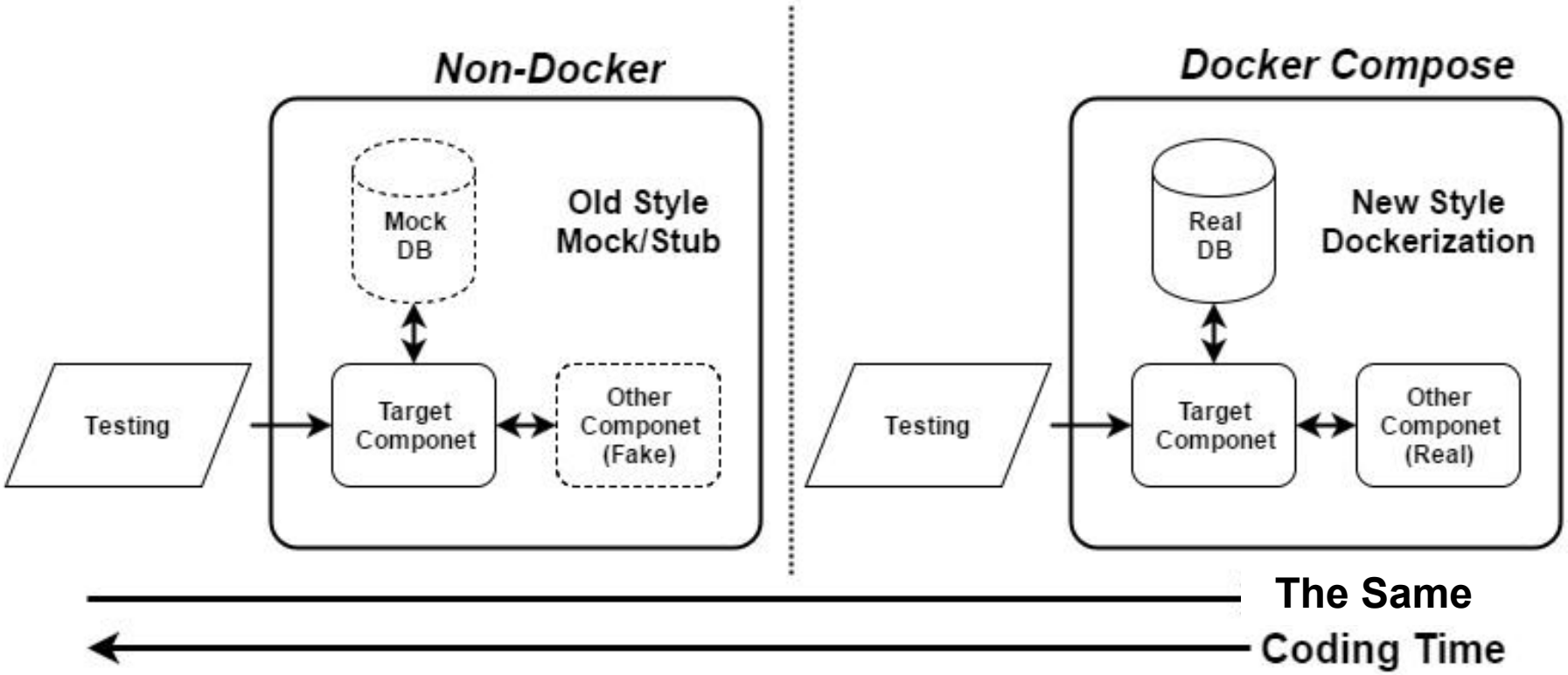
Container Development Flow

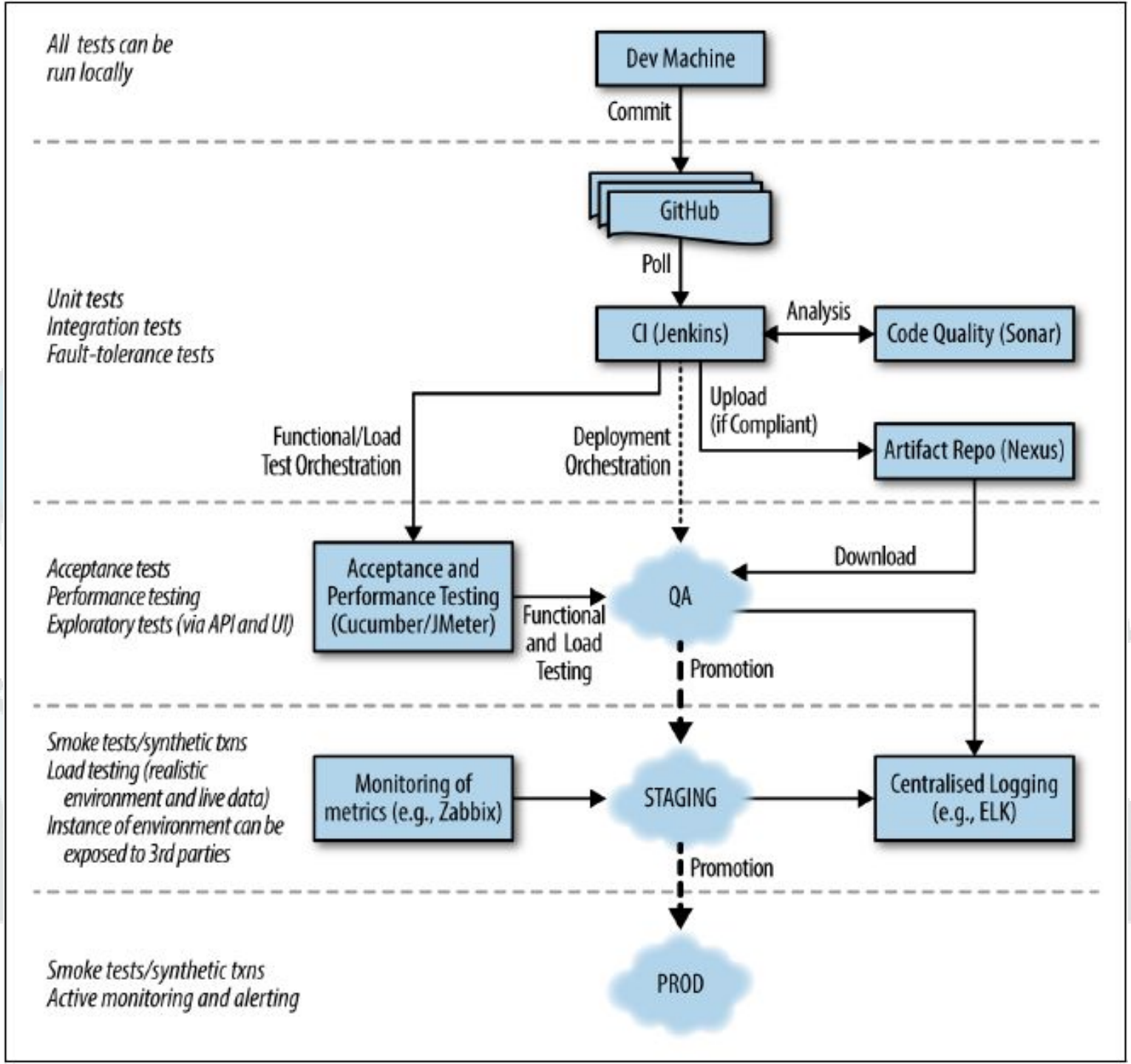


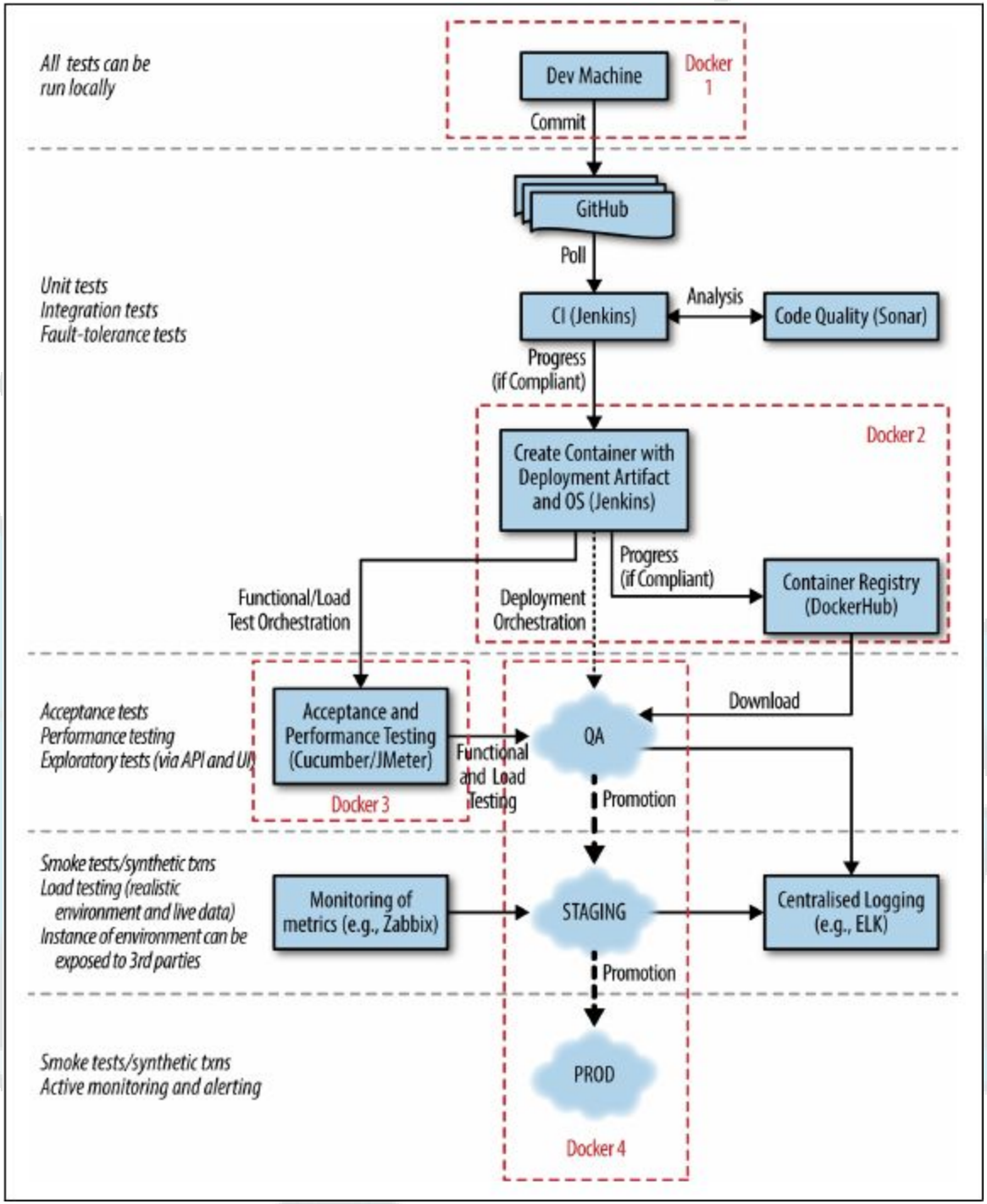
Test Double Approach

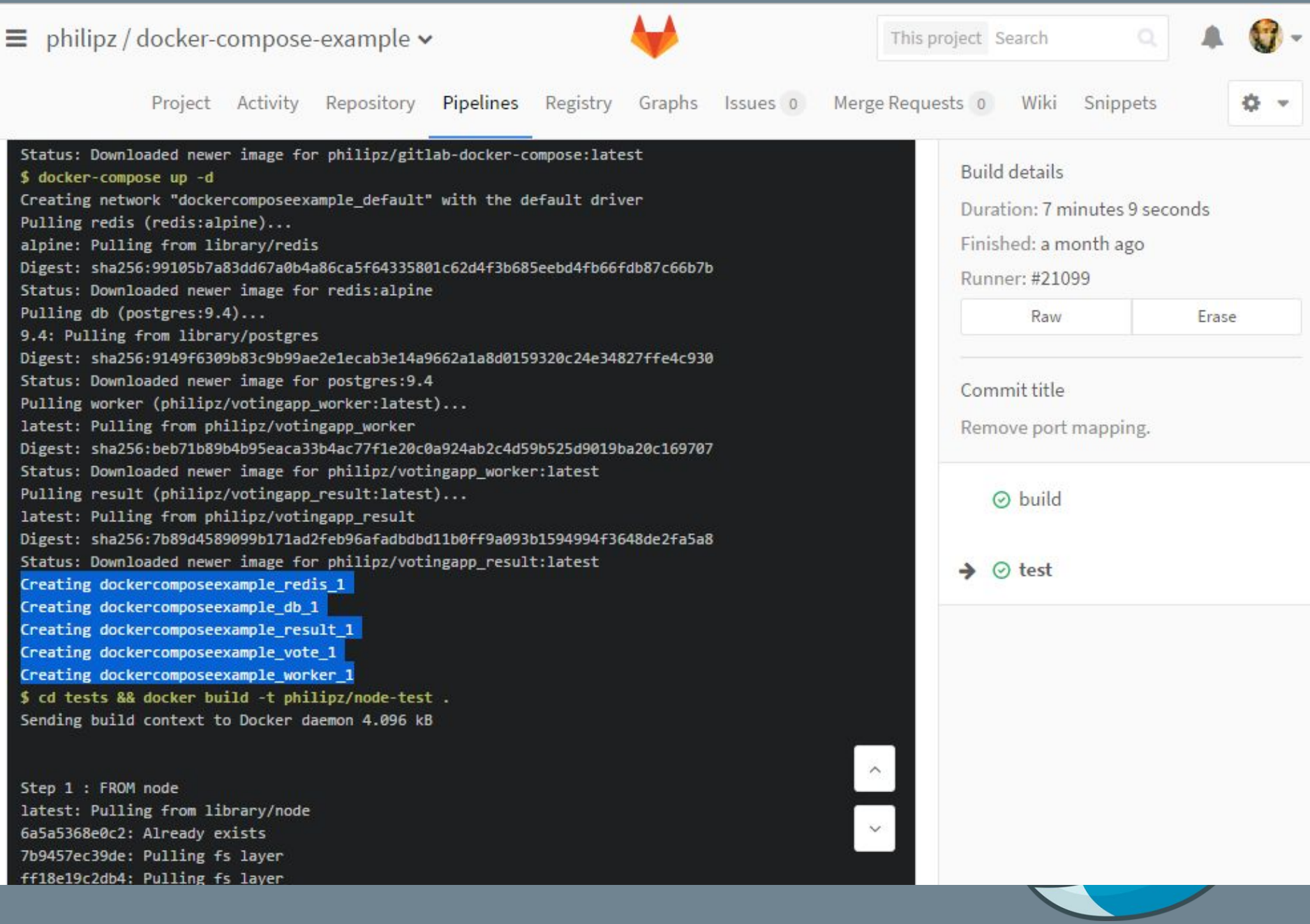


New Compose Test Approach




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

10. 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


JUST DO IT

Builds » philipz » rpi_circleci » master » build 19

0
(03:23)

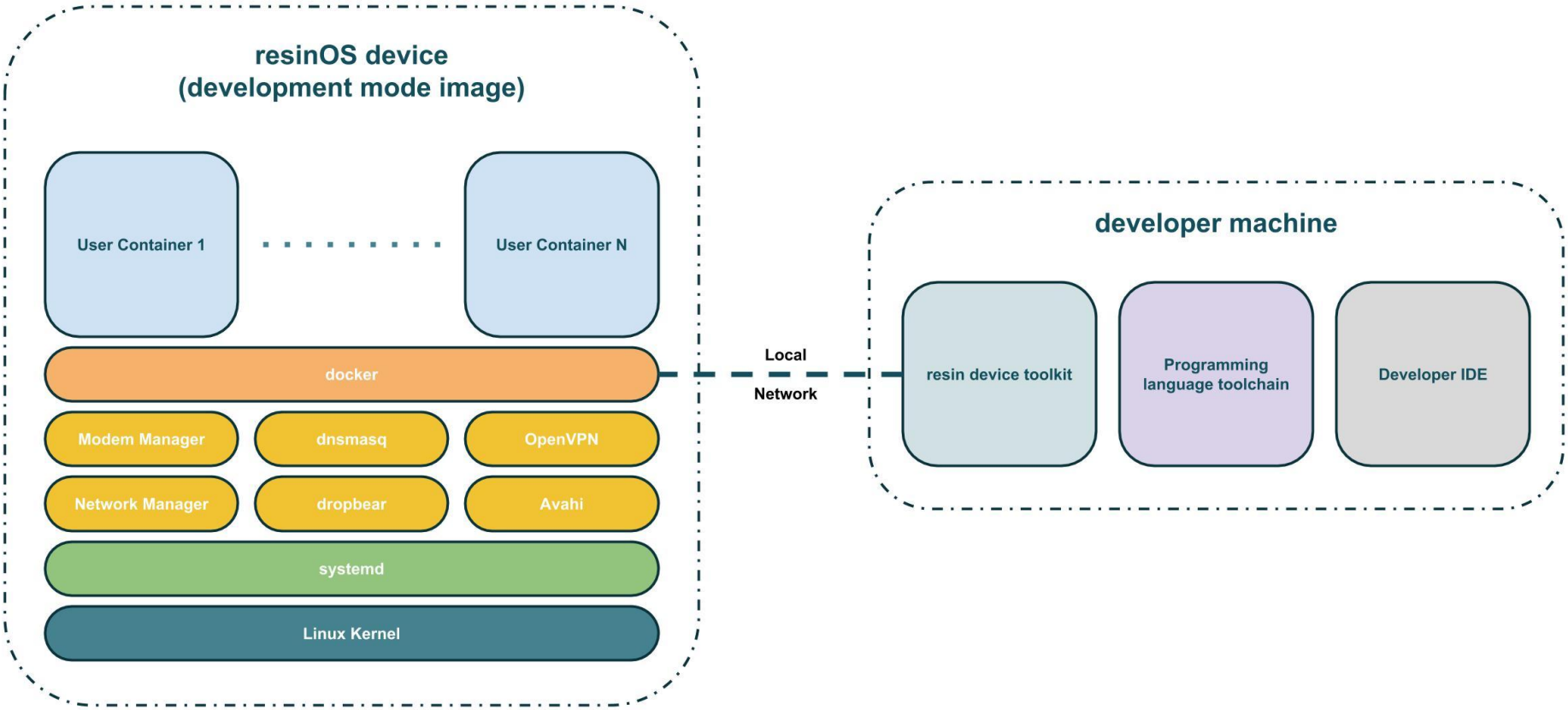
Add Containers +

```
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.)
```

?

Why resinOS

https://resinos.io/



11. Demo TensorFlow with Docker

Docker + TensorFlow + GPU

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

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

Hope You Love Docker
So long!