



逢甲大學 docker 研習班

Docker.Taipei Philipz(鄭淳尹)

2017-01-17

https://github.com/philipz/workshop_fcu

Today Topics

1. Docker Hub introduction
2. Git CLI
3. Docker Hub Auto-build from **Github**
4. Docker Network CLI
5. Docker Volume CLI
6. Docker Compose CLI
= Multi-Container on Single Host
7. Using Docker Compose & official voting application example



Live Restore

- Keep containers alive during daemon downtime
- Control and configure Docker with systemd

systemctl show --property=FragmentPath docker

debian/ubuntu - /lib/systemd/system/docker.service

ExecStart=/usr/bin/dockerd -H fd:// **--live-restore**

sudo systemctl daemon-reload

sudo systemctl restart docker

1. Docker Hub introduction



Docker Hub = App Store


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

Build, Ship, & Run
Any App, Anywhere

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





GitHub & Docker Hub




[Pull requests](#) [Issues](#) [Gist](#) + ▾

Authorize application

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



Review permissions



Repositories
Public and private

▾

[Authorize application](#)

Docker Hub Registry

Docker Hub Registry

[Visit application's website](#)

[Learn more about OAuth](#)

Vulnerability Analysis

CoreOS Clair

Anchore



clair

CLAIR CONTROL REPORT

Image: jgsqware/vulnerable-image

Total : 295 vulnerabilities

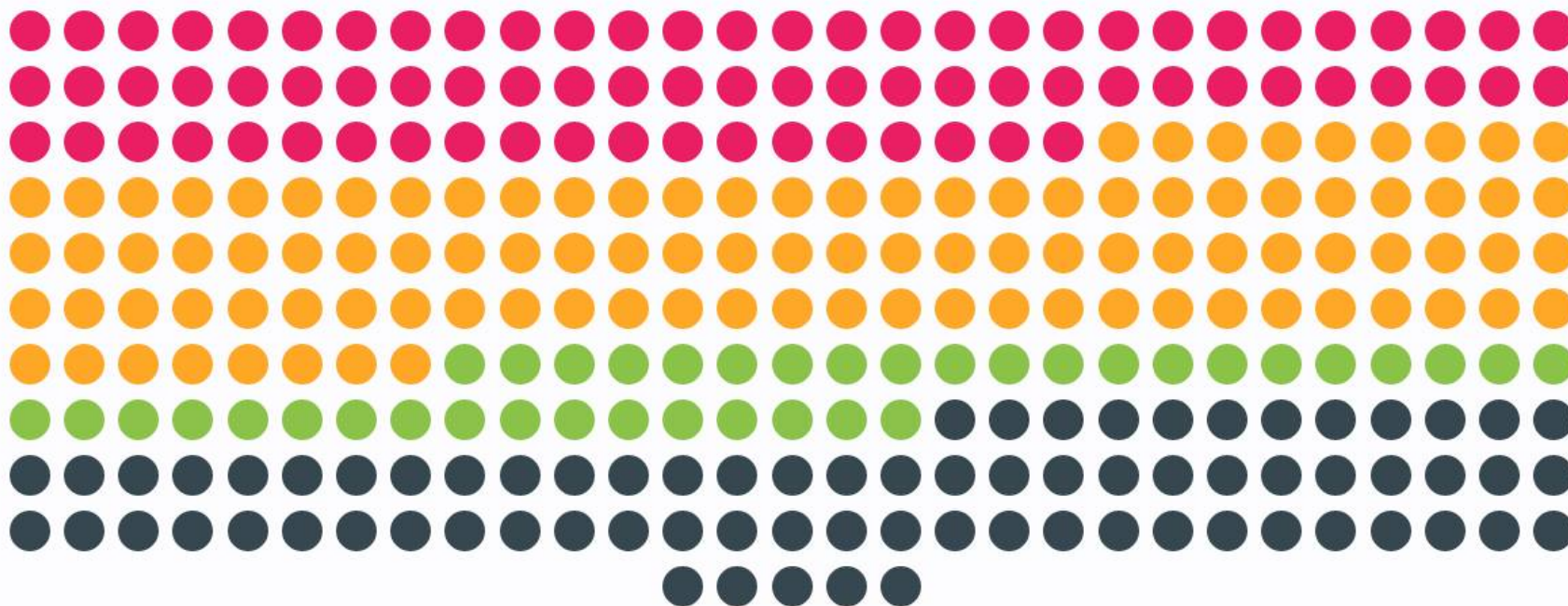
● Unknown : 17

● Negligible : 58


● Low : 38

● Medium : 104

● High : 78



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 command-line



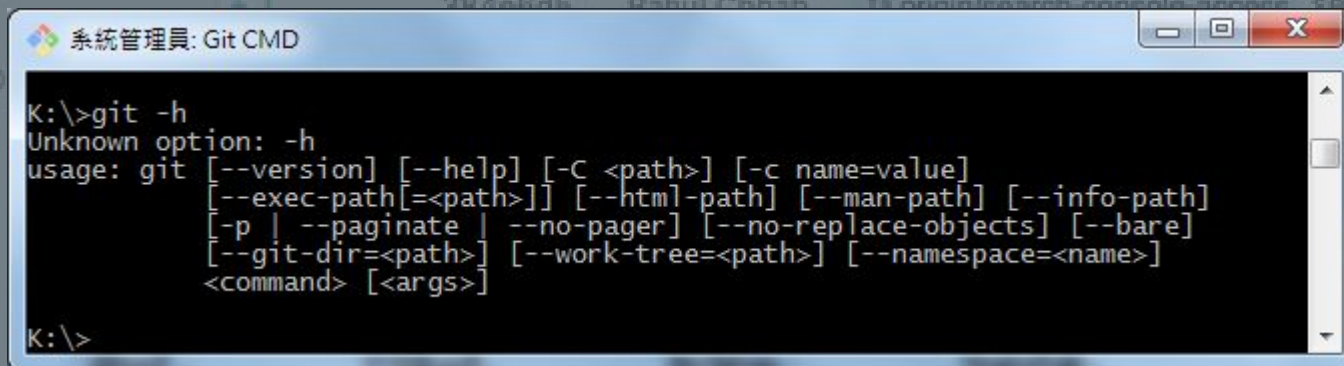
Git by Linus Torvalds

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

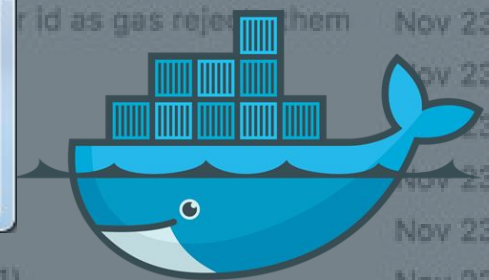


Install Git

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

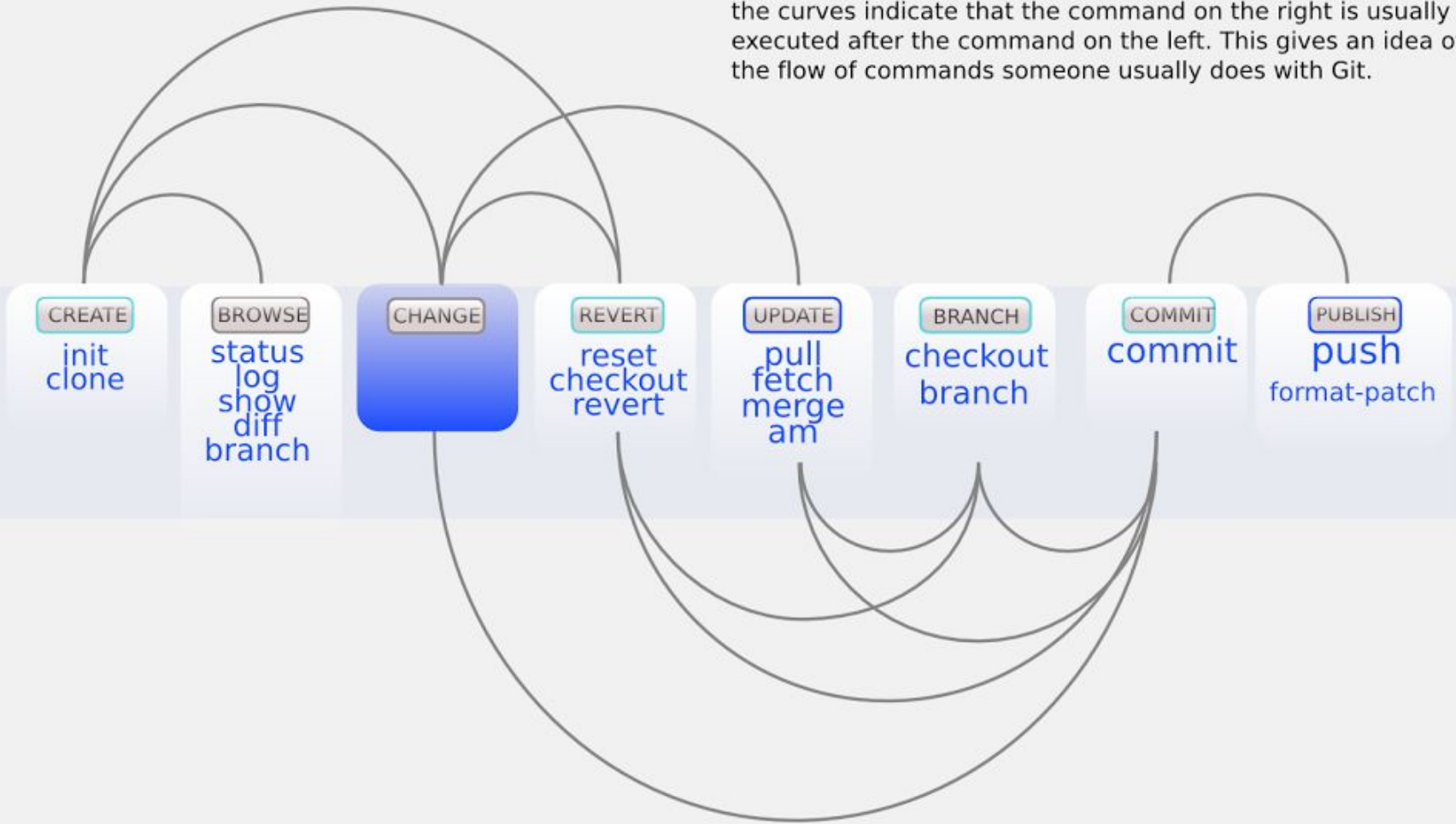


```
系統管理員: Git CMD
K:\>git -h
Unknown option: -h
usage: git [--version] [--help] [-C <path>] [-c name=value]
         [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
         [-p | --paginate | --no-pager] [--no-replace-objects] [--bare]
         [--git-dir=<path>] [--work-tree=<path>]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



Commands Sequence

the curves indicate that the command on the right is usually executed after the command on the left. This gives an idea of the flow of commands someone usually does with Git.



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)

`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 fetch`
(but th

Pull lat

`git pull`
(does

Apply

`git an`

Useful Commands

Find

`git`
`git`
`git`

`git`
`git`
`git`

Che

`git`
`git`

Sea

`git`

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

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`

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()

To view the merge conflicts

`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

Sample:

FROM debian:jessie

MAINTAINER docker "docker@nginx.com"

RUN apt-get update && apt-get install -y nginx

CMD ["nginx", "-g", "daemon off;"]

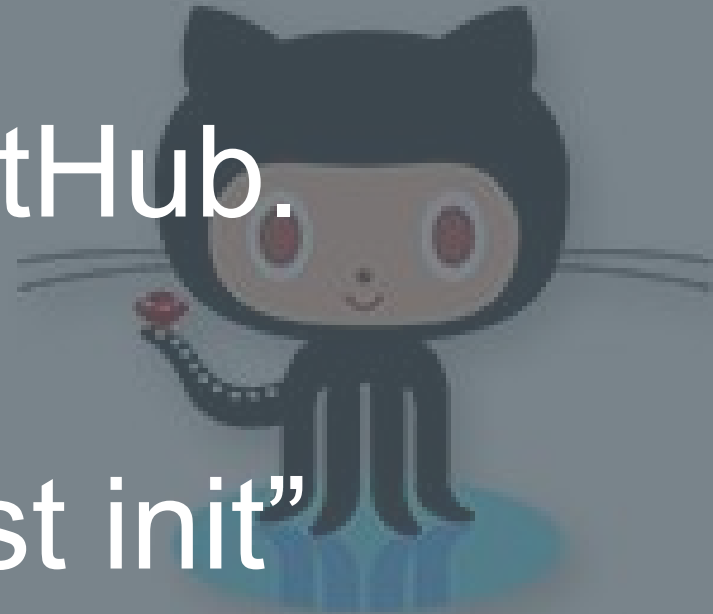


Git Workflow

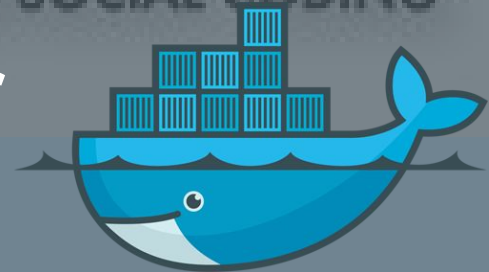
1. `git init` or init on GitHub.
2. `git add Dockerfile`
3. `git commit -m "First init"`
4. `git remote add origin`

https://github.com/YOURNAME/docker_build.git

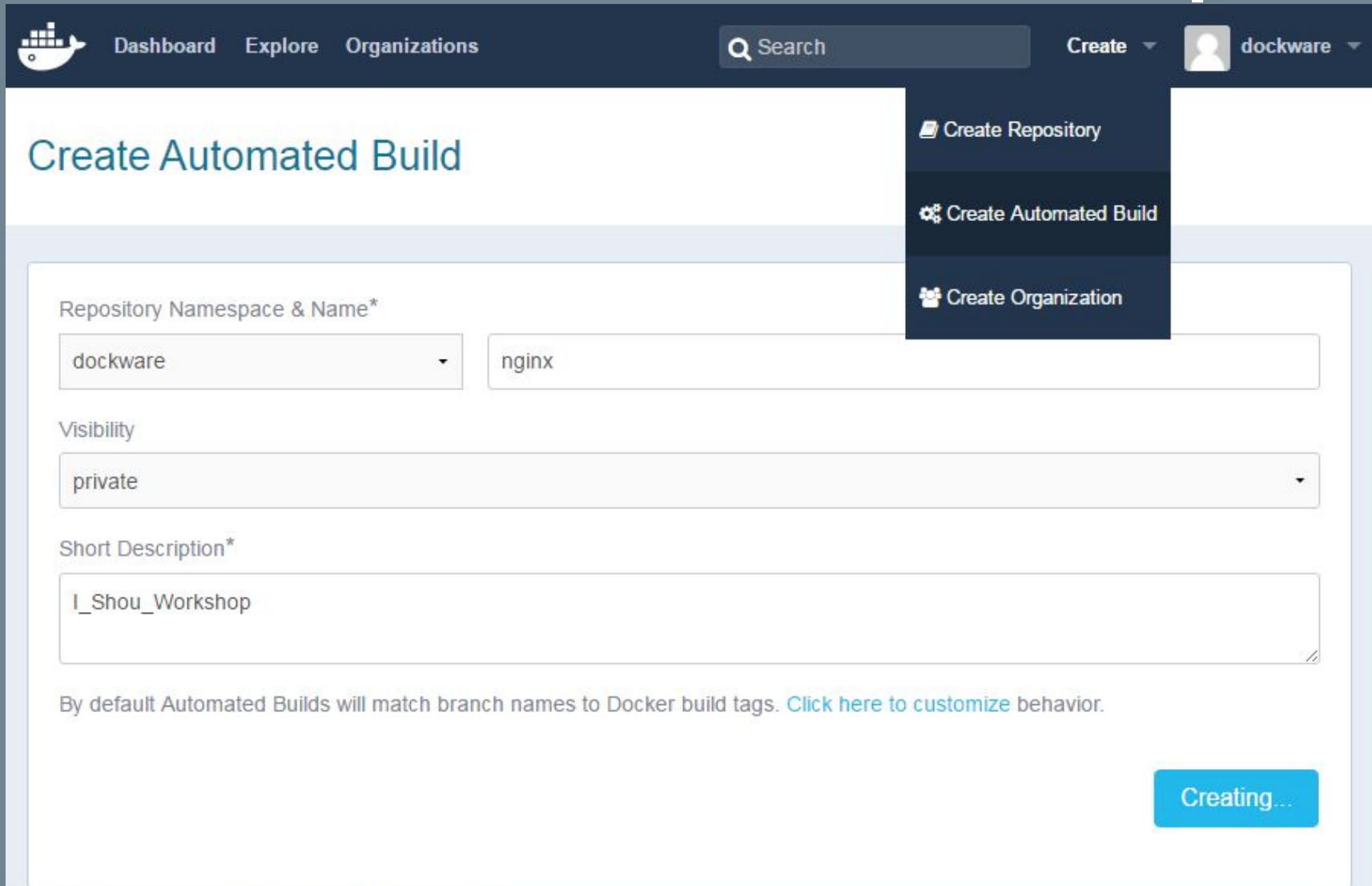
5. `git push origin master`



github
SOCIAL CODING



Create Auto-build Repo.



Dashboard Explore Organizations Search Create dockware

Create Automated Build

- Create Repository
- Create Automated Build
- Create Organization

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...

Build Settings

PUBLIC | AUTOMATED BUILD

philipz/rpi-raspbian ☆

Last pushed: 3 months ago

[Repo Info](#) [Tags](#) [Dockerfile](#) [Build Details](#) [Build Settings](#) [Collaborators](#) [Webhooks](#) [Settings](#)

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



⌕ Trigger

docker pull YOURNAME/IMAGENAME

Save Changes

4. 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 ~

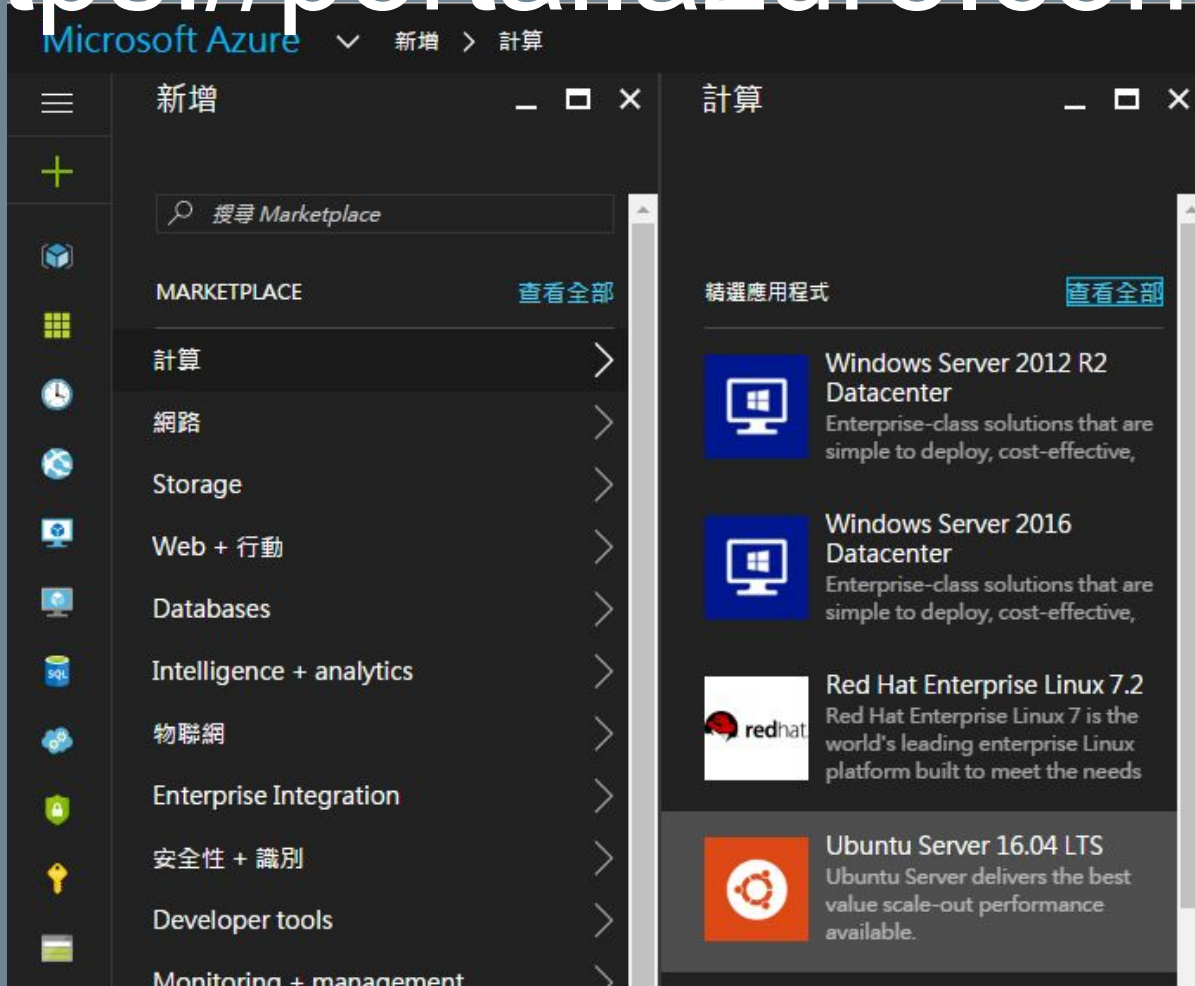
172.31.0.0/12

192.168.0.0/16

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

Microsoft Azure

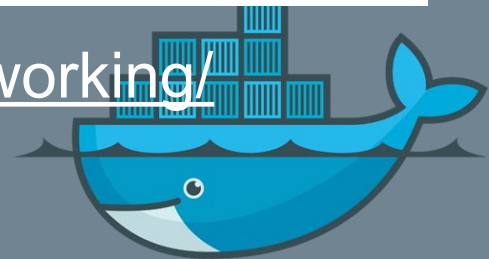
<https://portal.azure.com/>



Network and connectivity commands

Command	Description
<code>network connect</code>	Connect a container to a network
<code>network create</code>	Create a new network
<code>network disconnect</code>	Disconnect a container from a network
<code>network inspect</code>	Display information about a network
<code>network ls</code>	Lists all the networks the Engine daemon knows about
<code>network rm</code>	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

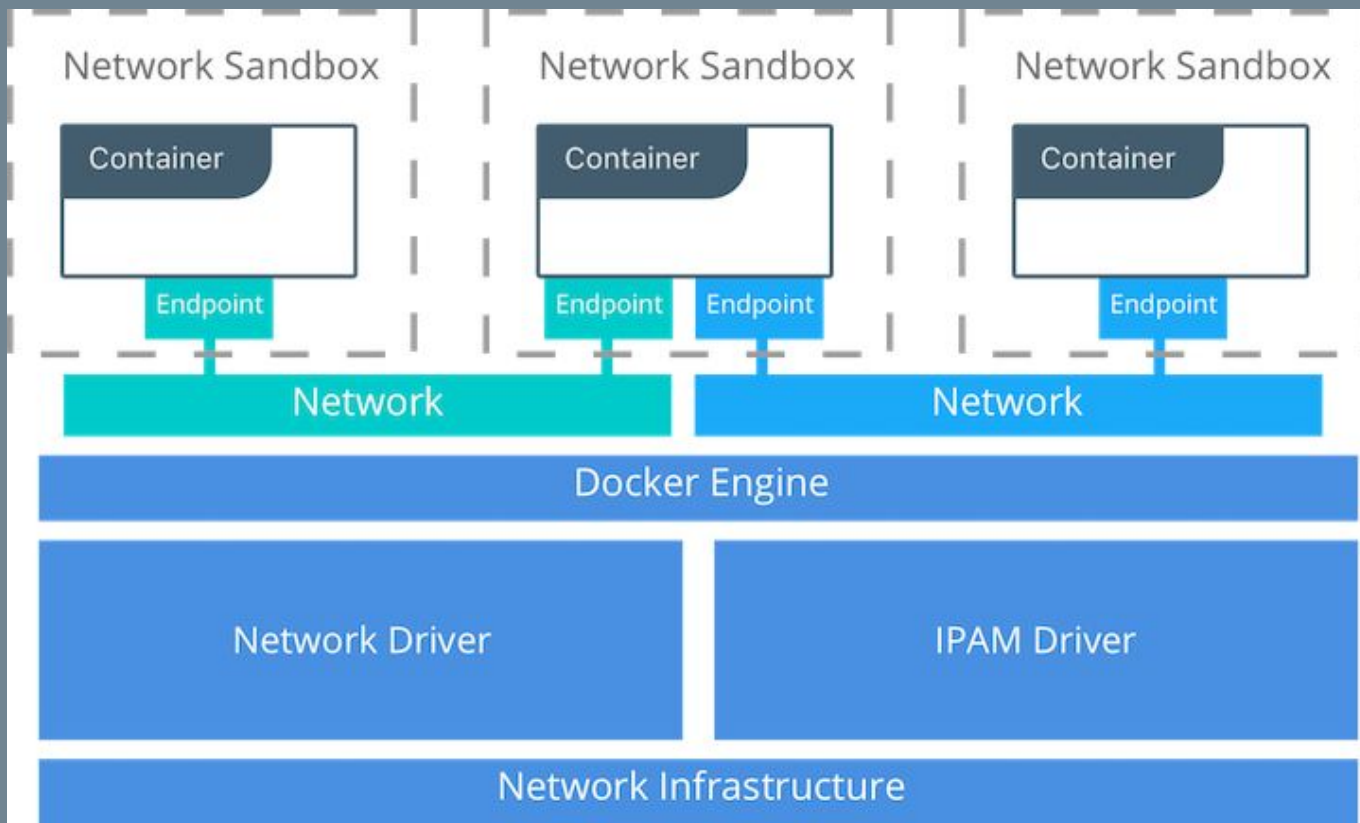
- infoblox

No more “link”, just use network.

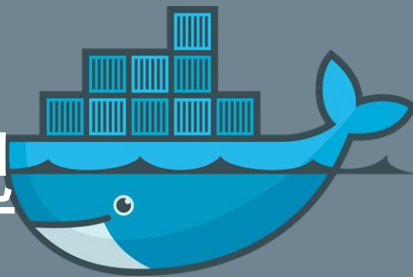
Docker Reference Architecture: Designing Scalable,
Portable Docker Container Networks



CNM vs CNI



容器网络聚焦：CNM和CNI，原文
SDNLAB技术分享(十五)：容器网络大观



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
```

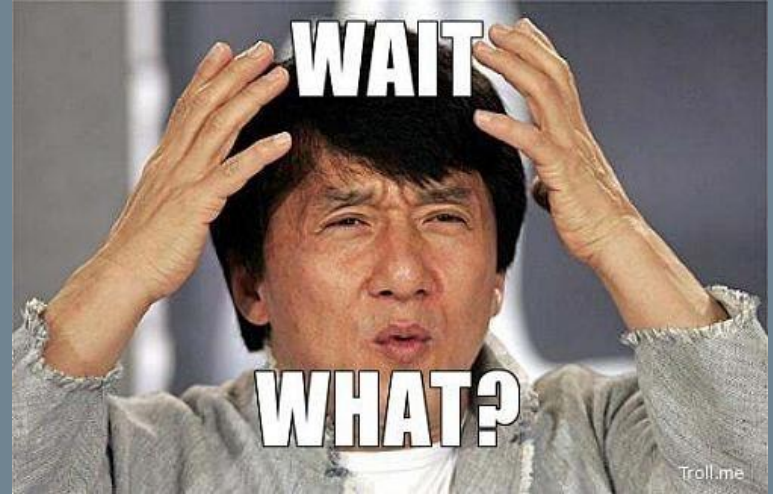
```
$ docker run -d --name db --network=wp_db  
-e MYSQL_ROOT_PASSWORD=wordpress  
-e MYSQL_DATABASE=wordpress  
-e MYSQL_USER=wordpress  
-e MYSQL_PASSWORD=wordpress  
mysql:5.7
```

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



Exercise 4

```
$ 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
```

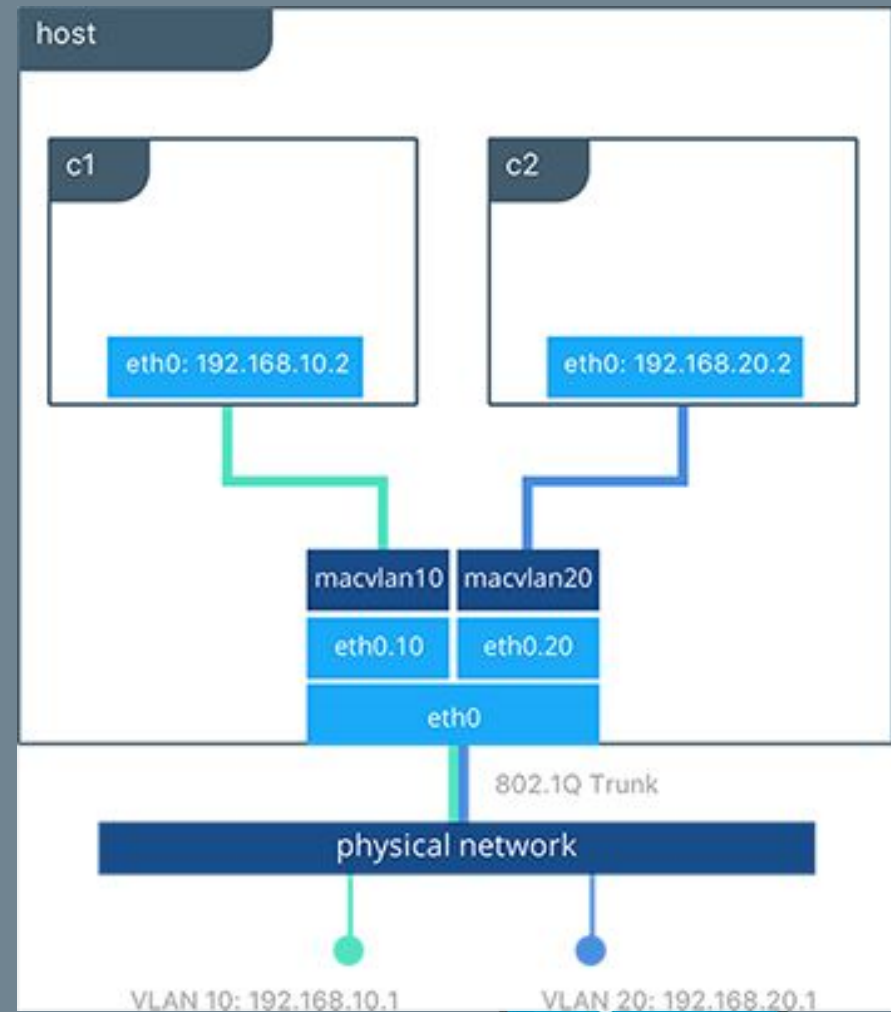
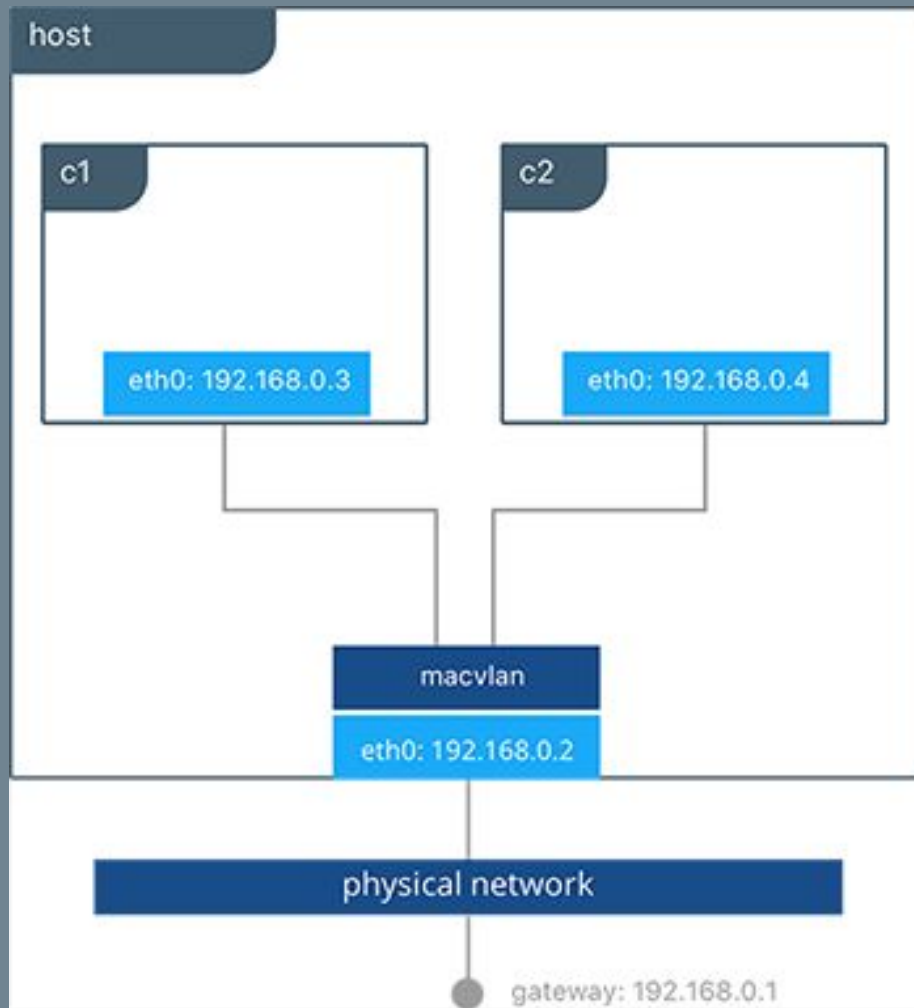
```
$ docker run -it --name c2 --net mvnet --ip 10.0.0.6 busybox sh
```

```
ping -c 4 10.0.0.5
```

```
ip a show eth0, ip route
```

Get started with Macvlan network driver

Macvlan Networking



SDN

Software Define Network
Dynamic Network Topology
Telecom, Cloud service...
Resource allocation
Cost down



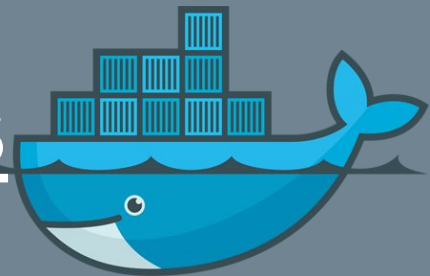
5. Docker Volume command-line



Shared data volume commands

Command	Description
<code>volume create</code>	Creates a new volume where containers can consume and store data
<code>volume inspect</code>	Display information about a volume
<code>volume ls</code>	Lists all the volumes Docker knows about
<code>volume rm</code>	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
```



SDS

Software Define Storage

EMC: REX-Ray

Azure: File storage

AWS: Elastic File System



6. 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

Swarm mode is not support yet.

Quickstart: Compose and WordPress



Compose File Sample (1/2)

version: '2'

services:

db:

image: mysql:5.7

volumes:

- db_data:/var/lib/mysql

restart: always

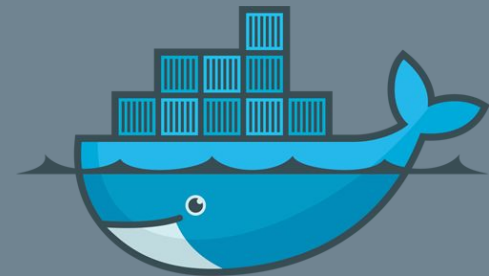
environment:

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:
```

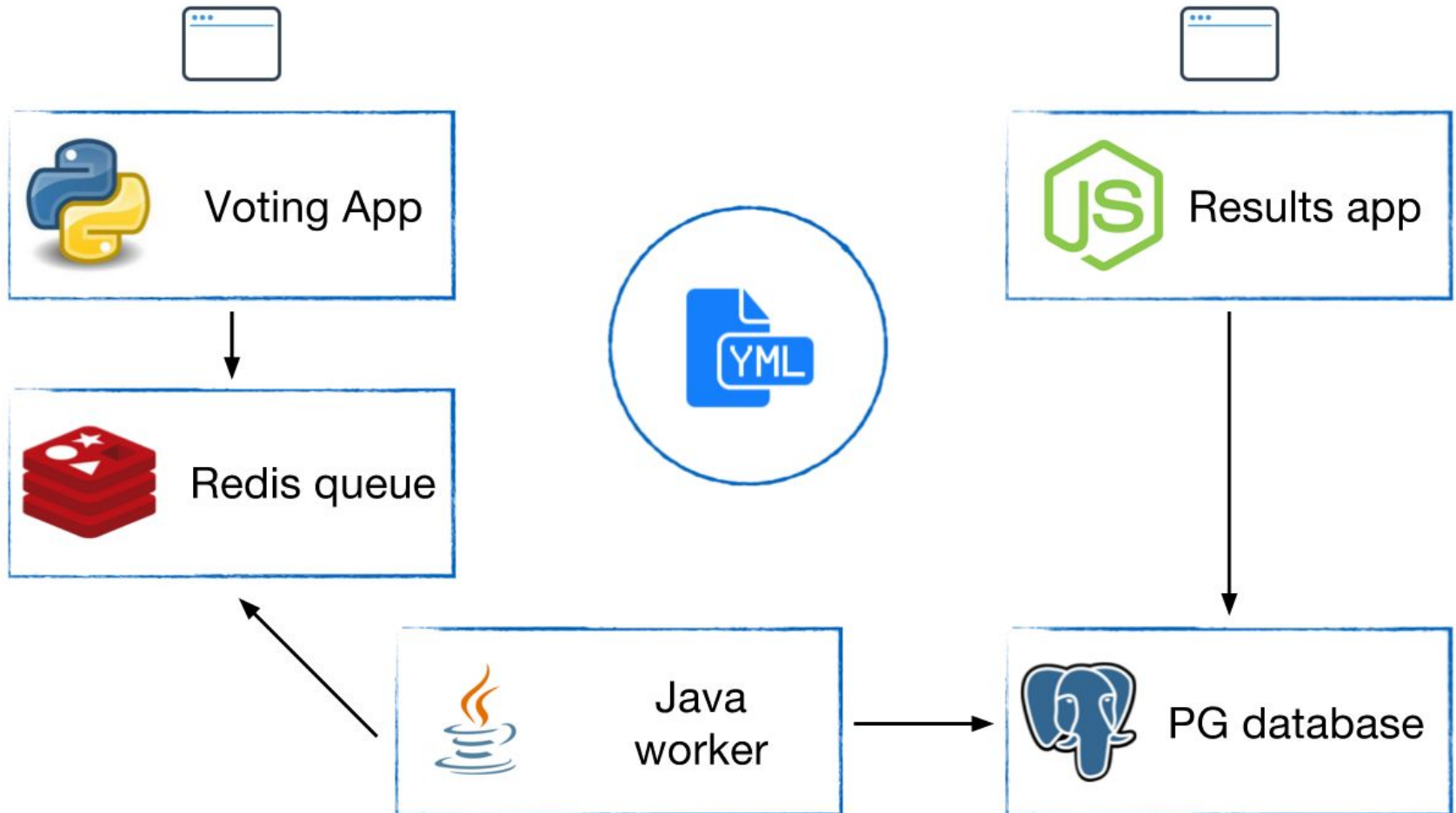


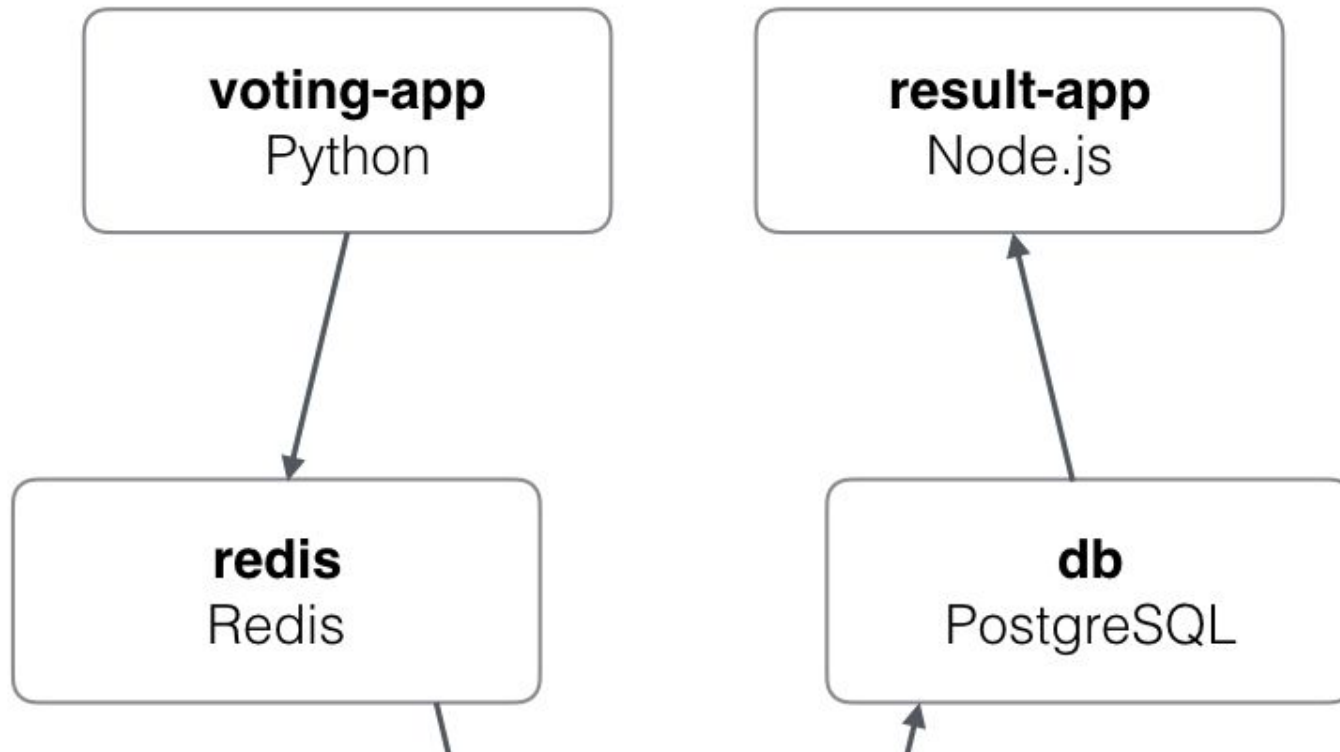
7. Using Docker Compose



Microservices Java Worker

Docker Birthday #3 training





Microservices .NET Worker

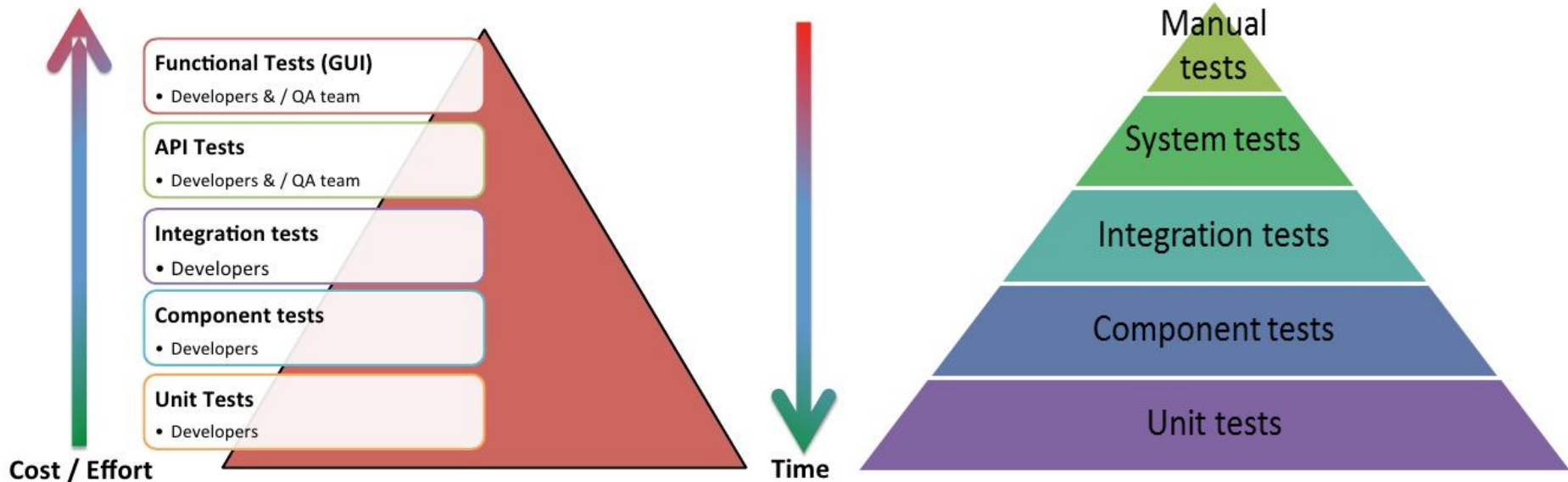
[Docker Birthday #3 training](#)

Docker Compose & CI/CD

[Github](#), [CircleCI](#), [Docker Hub](#) = GitLab

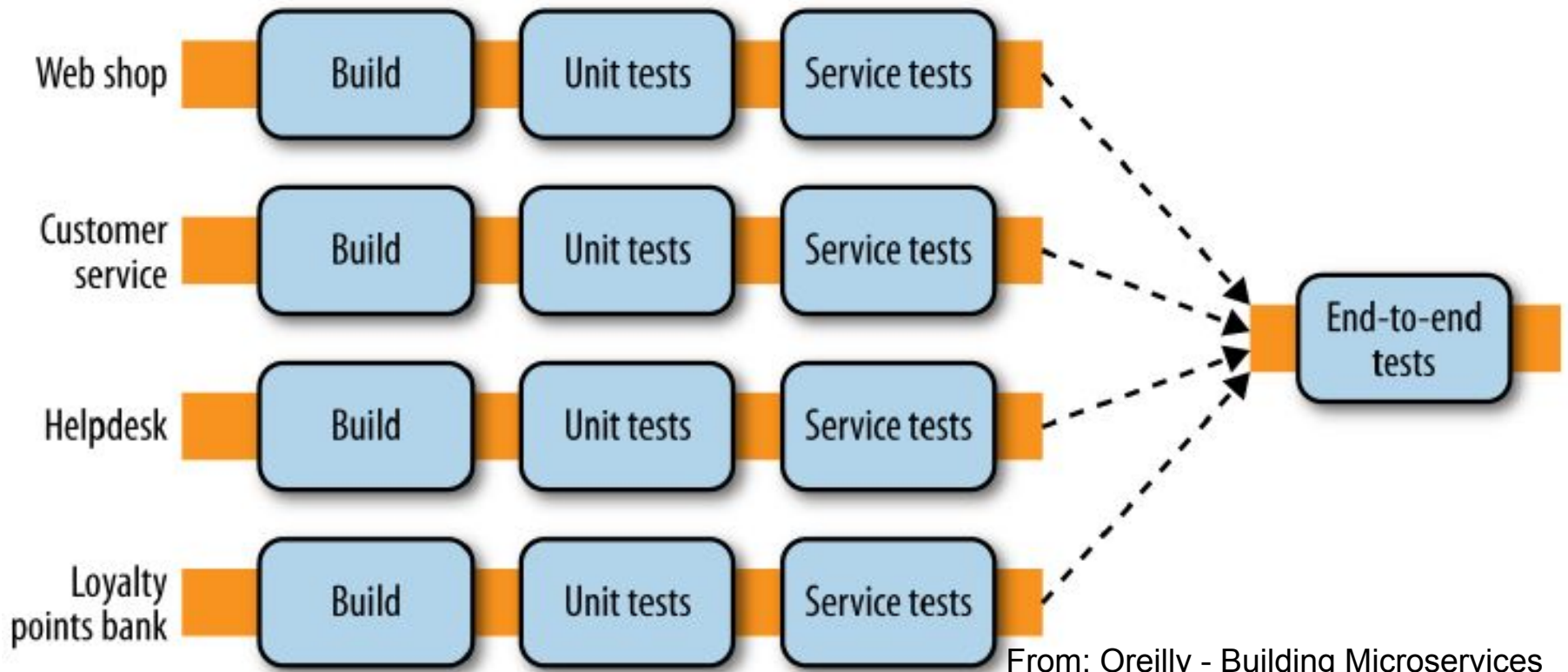
Testing level? Coding effort? Env. build-up

Ideal Test Pyramid



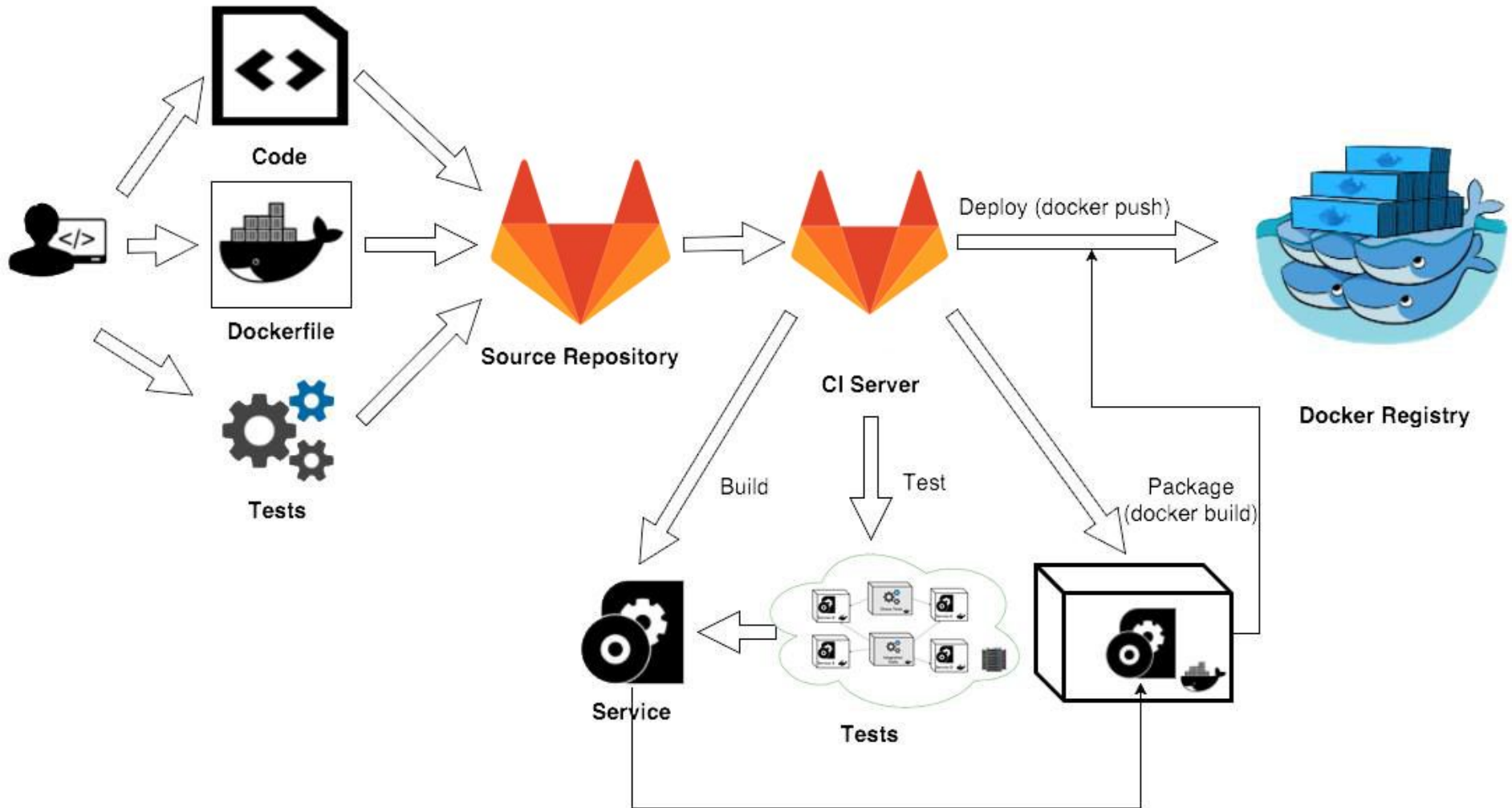
End to End Tests

CI with Docker Compose is easy to implement.



From: Oreilly - Building Microservices

Container Development Flow



From: [Testing Strategies for Docker Containers](#)



```
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



gitlab_cal

Add a note



TRIGGER

1. Catch Hook



ACTION

2. Create Detailed Event

31 Google Calendar

⚡ Create Detailed Event

👤 Google Calendar Account #1

☰ Edit Template

🔧 Test this Step

✎ Rename Step

✖ Delete



Set up Google Calendar Detailed Event

31 Calendar (required)

Note: You must be able to edit the selected calendar from the Google Calendar account you connected.

GitLab

31 Summary (optional)

Summary of the new event

Step 1 Project Name Step 1 Object Kind

31 Description (optional)

Description of the new event

Step 1 Commit Author Name Step 1 Commit Author Email

Project: Step 1 Project Path With Namespace

Branch: Step 1 Project Default Branch

Step 1 Commit Message

Step 1 Object Attributes Url



搜尋日曆



日曆

今天



2016年10月3日 - 9日

天

週

月

4天

待辦事項

更多



建立

▼ 2016年10月

一	二	三	四	五	六	日
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

GMT+08

10/3 (週一)

10/4 (週二)

10/5 (週三)

10/6 (週四)

10/7 (週五)

10/8 (週六)

10/9 (週日)

重陽節
重陽節

▼ 我的日曆

- ☐ Philip Zheng
- ☐ 生日
- ☐ 提醒
- ☒ GitLab

▼ 其他日曆

新增朋友的日曆

- ☒ 天氣
- ☒ 台灣的節慶假日
- ☒ 台灣的節慶假日
- ☒ 程式交易機器人交易...
- ☒ OptionsBot SMS
- ☒ TradingBot SMS
- ☐ 台北市法拍屋行事曆
- ☐ 新北市法拍屋行事曆

條款 - 隱私設定

▼ maven-java-example issue

時間

10月 5日 (週三), 上午10:54

日曆 GitLab

建立者 philipzheng@gmail.com

複製到我的日曆

刪除

編輯活動

ThoughtWorks Tech. Radar

"Docker as process, PaaS as machine,
microservices architecture as programming model"

平台

采用

- 23. Docker
- 24. HSTS
- 25. Linux security modules

试验

- 26. Apache Mesos
- 27. Auth0 new
- 28. AWS Lambda
- 29. Kubernetes
- 30. Pivotal Cloud Foundry
- 31. Rancher
- 32. Realm
- 33. Unity beyond gaming new

评估

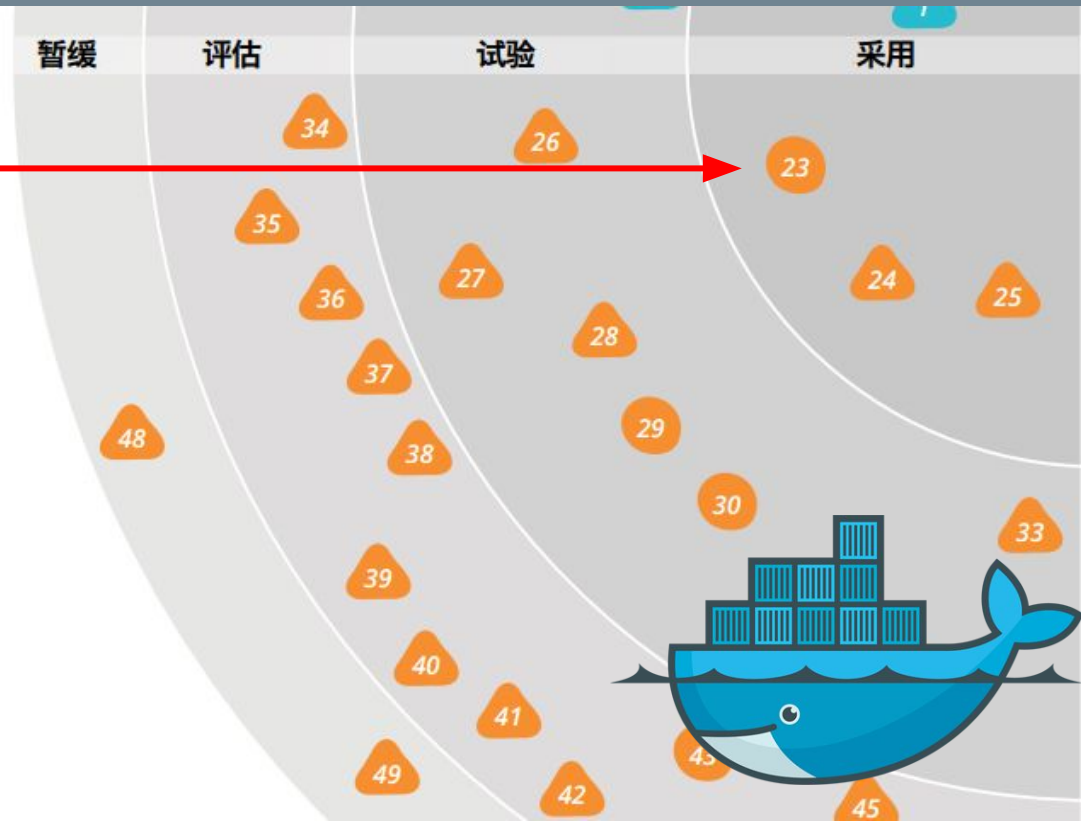
- 34. .NET Core
- 35. Amazon API new
- 36. Apache Flink
- 37. AWS Application Load Balancer new
- 38. Cassandra carefully new
- 39. Electron new
- 40. Ethereum new
- 41. HoloLens new
- 42. IndieStack new

暂缓

评估

试验

采用



Tomorrow 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.





See You Tomorrow