

A stylized graphic of a robot head, primarily white with a large blue circular face. The face has a vertical blue bar for a nose and several small blue dots for eyes and a mouth. The robot has black circular ears and a small orange circle at the bottom of its neck. The background is a solid orange color.

# DOCKER

## POUR NOUS LES DEVS

*(... et les astronautes)*

# QUI SUIS-JE ?

**Guillem CANAL**

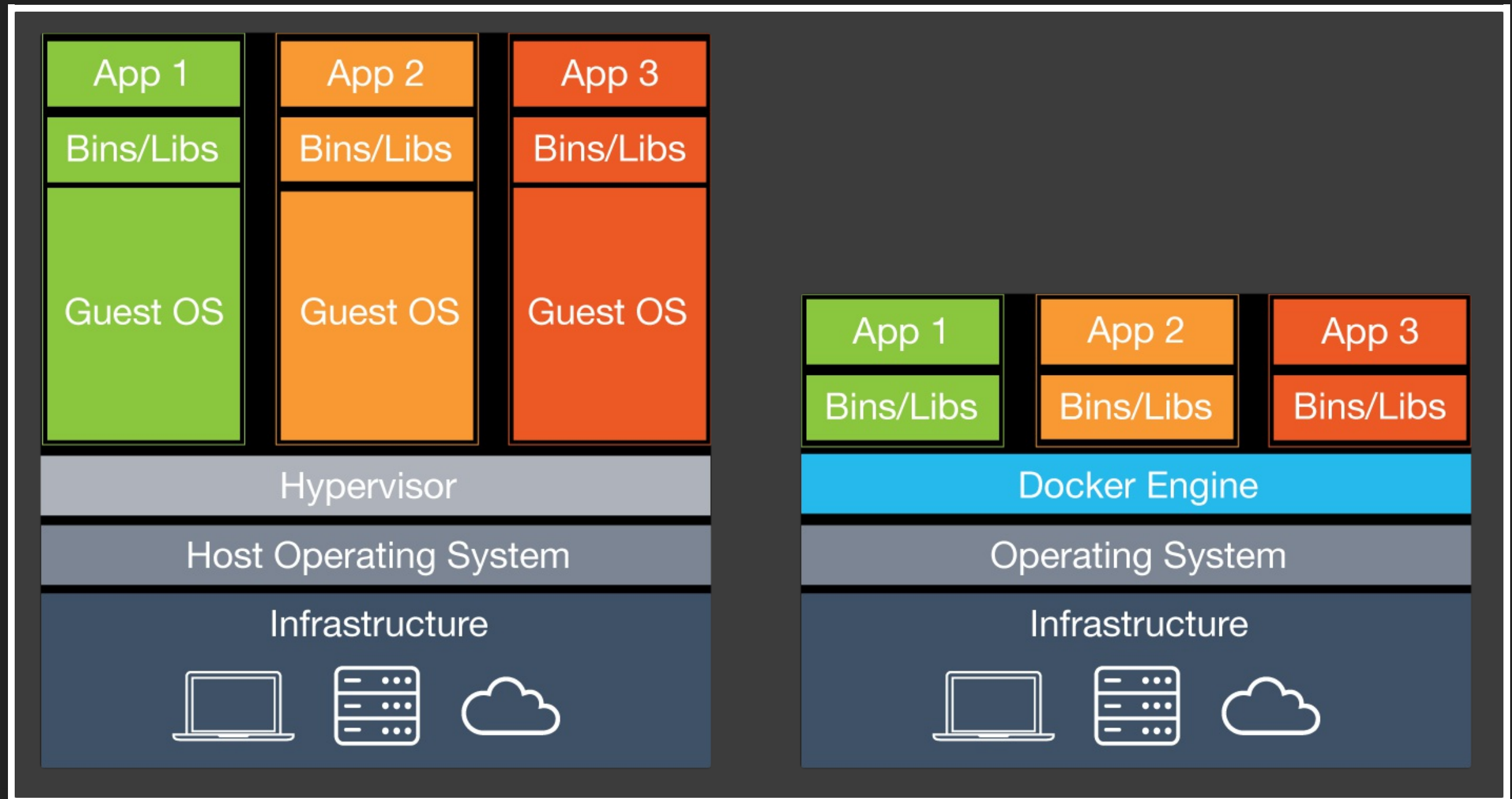
Développeur Back PHP @ [ElevenLabs](#)

**Twitter:** [@buraitopengin](#)

**Github:** [guillemcanal](#)

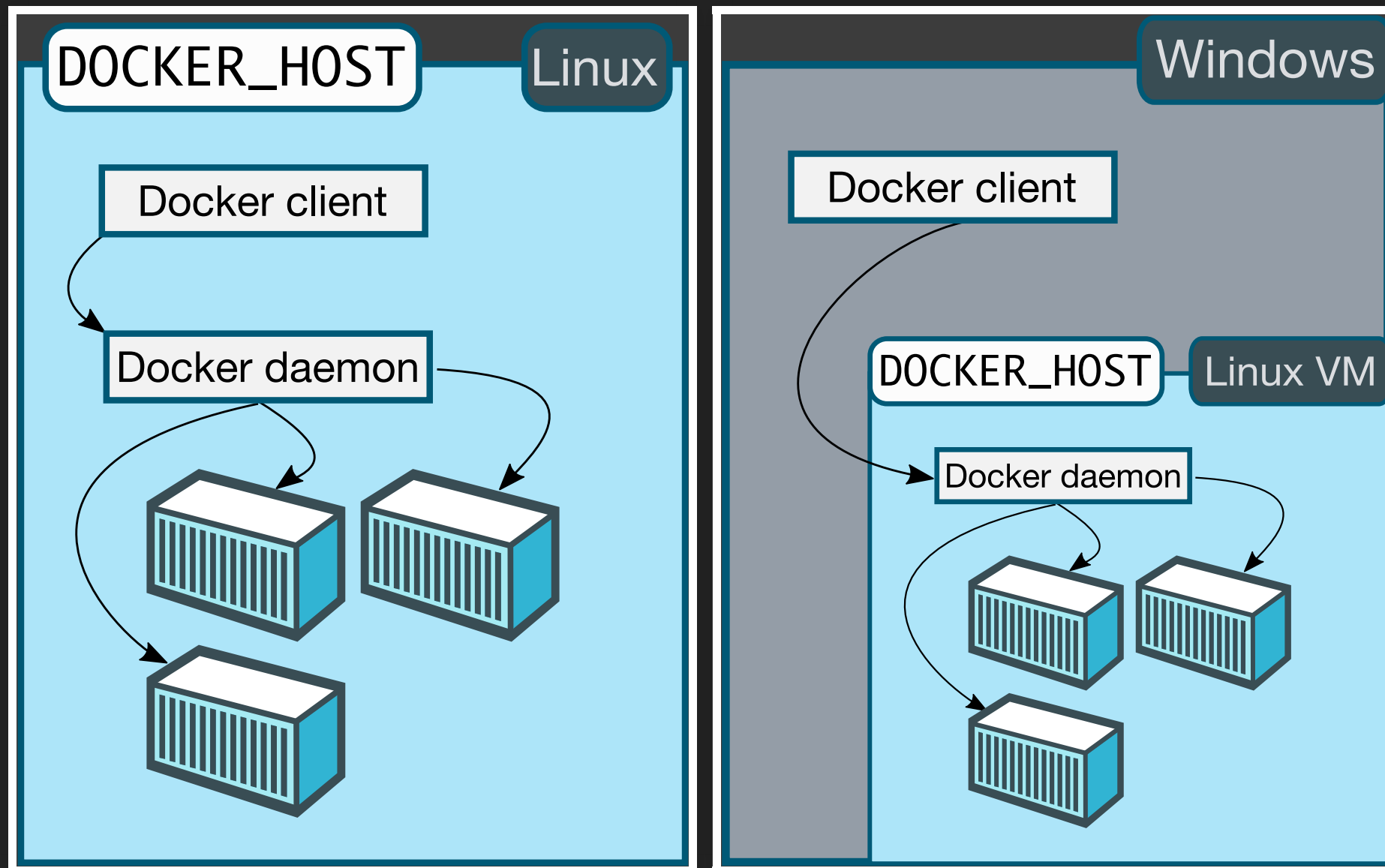
Consulter la présentation sur Github Pages:  
[guillemcanal.github.io/docker-prez](https://guillemcanal.github.io/docker-prez)

# VM VS DOCKER



# SAUF QUE...

Windows et Mac n'ont pas de Kernel sous la main.  
GogoGadgets VM ?!



# INSTALLATION

## UBUNTU (14.04 LTS)

...ou [#RTFM](#)

```
$ echo "deb https://apt.dockerproject.org/repo ubuntu-trusty main" \  
| sudo tee -a /etc/apt/sources.list  
$ apt-get update  
$ apt-get install \  
    apt-transport-https \  
    ca-certificates \  
    linux-image-extra-$(uname -r) \  
    apparmor \  
    docker-engine  
$ sudo usermod -aG docker $(id -un)
```

Quitter la session puis revenir

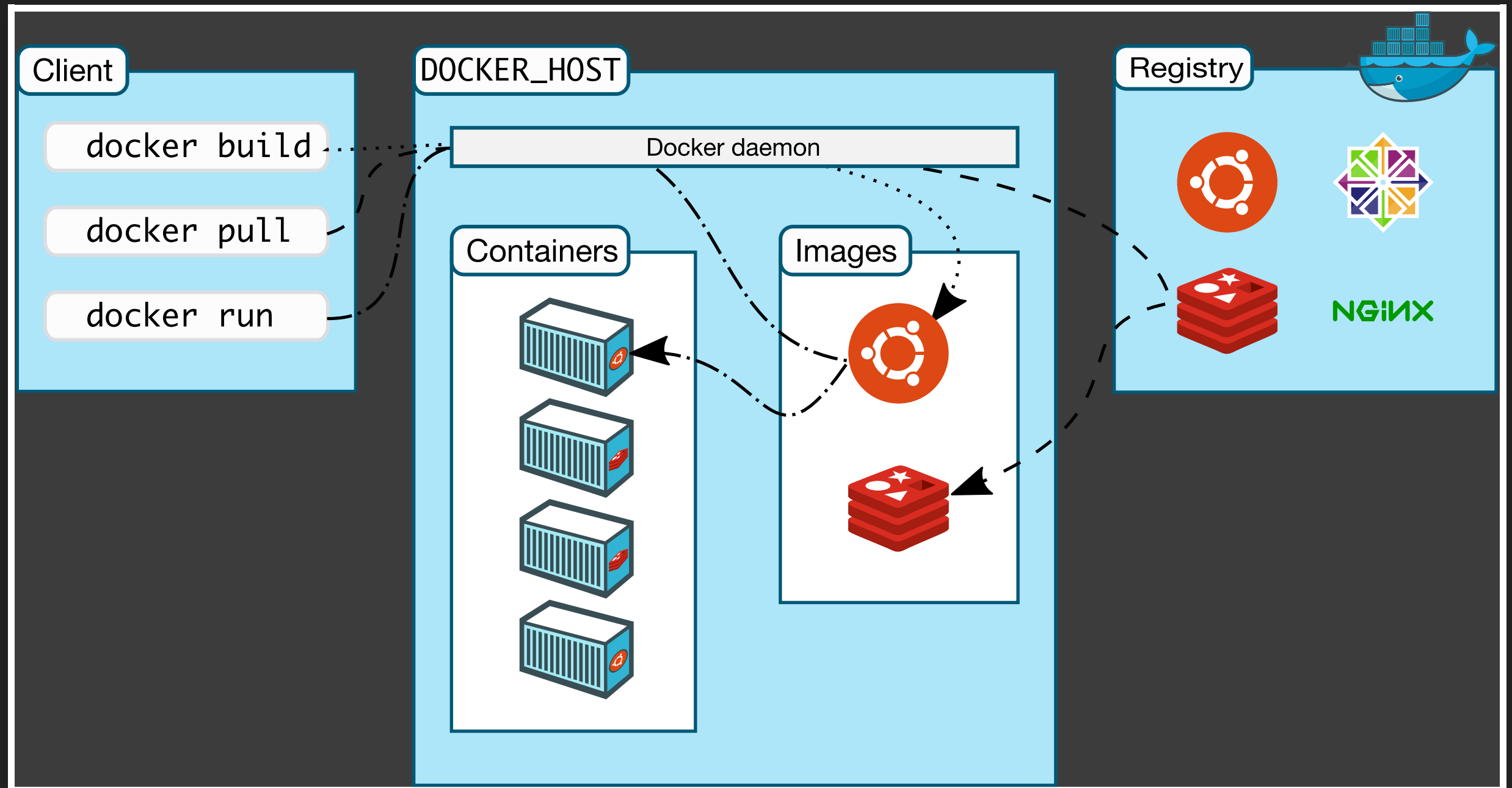
# INSTALLATION

## POUR WINDOWS ET MAC

Installer la [Docker Toolbox](#)

```
# choisir le nom de sa VM
$ NOM_MACHINE=""
# créer la machine
$ docker-machine create -d virtualbox \
--virtualbox-cpu-count "4" \
--virtualbox-disk-size "50000" \
--virtualbox-hostonly-nictype "Am79C973" \
--virtualbox-memory "4096" \
--virtualbox-no-share $NOM_MACHINE
# créer un partage NFS avec l'hôte
# @see: https://github.com/adlogix/docker-machine-nfs
$ docker-machine-nfs meetic \
--mount-opts="noacl,async,actimeo=1" --force
```

# ARCHITECTURE



# POUR NOUS LES BOURRINS

Supprimer les conteneurs qui ne tourne pas

```
# docker remove containers (docker list containers id where status is exited)  
docker rm $(docker ps -q -f status=exited)
```

Oblitérer tous les conteneurs qu'ils tournent ou non

```
# docker remove containers force (docker list all containers ID)  
docker rm -f $(docker ps -aq)
```

Supprimer les images qui ne sont pas tagguées

```
# docker remove image (docker list untagged images ID)  
docker rmi $(docker images -f "dangling=true" -q)
```



# PARTAGE DES IMAGES AVEC LES COPAINS

## Build

```
$ # je build à partir d'un fichier Dockerfile  
$ docker build -t monimage:1.0 .
```

## Tag

```
$ # je tag mon image avec un numéro de version  
$ docker tag monimage:1.0 guillem.ninja/monimage:1.0
```

## Push

```
$ # j'envoie le bouzin sur ma registry privée  
$ docker push guillem.ninja/monimage:1.0
```

# CI

## GITLAB RUNNER

Utilitaire en cli permettant de faire tourner les tests de son projets en local.

```
image: jolicode/php56
```

```
before_script:
```

```
- composer install
```

```
# démarrer un conteneur mysql
```

```
# pour les tests fonctionnels
```

```
services:
```

```
- mysql
```

```
variables:
```

```
MYSQL_DATABASE: hello_world_test
```

```
MYSQL_ROOT_PASSWORD: mysql
```

```
test:mysql:
```

```
script:
```

```
- vendor/bin/phpunit --configuration phpunit_mysql.xml --coverage-text
```

# Exemple d'usage sur un projet PHP

You are running composer with xdebug enabled. This has a major impact on runtime performance. See <https://getcomposer.org/xdebug>

Loading composer repositories with package information

Installing dependencies (including require-dev) from lock file

- Installing phpdocumentor/reflection-docblock (2.0.4)  
Downloading: 100%

- Installing phpunit/php-token-stream (1.4.8)  
Downloading: 100%

- Installing symfony/yaml (v2.7.7)  
Downloading: 100%

- Installing sebastian/version (1.0.6)  
Downloading: 100%

- Installing sebastian/resource-operations (1.0.0)  
Downloading: 100%

- Installing sebastian/global-state (1.1.1)  
Downloading: 100%

- Installing sebastian/recursion-context (1.0.1)  
Downloading: 100%

- Installing sebastian/exporter (1.2.1)  
Downloading: 100%

- Installing sebastian/environment (1.3.2)  
Downloading: Connecting...



00:00



## Outil maison en shell chez Meetic

```
#!/bin/bash

source ./docker-functions

PROJECT_NAME="php-api-inbox"
BUILD_TARGET=$1

echo_info "starting docker services"
docker_add_service "memcached" "memcached"
docker_add_service "meetic/mysql-base" "mysql"
docker_add_service "meetic/bdd-oracle-xe-11g:v1" "oracle"
docker_add_service "meetic/broker-rh-5" "broker" "oracle:oracle-host"

build_env "SYMFONY_ENV=test"

build_script "echo 'tout va bien se passer :)'"
build_script "phing $BUILD_TARGET"

build_execute "meetic/php53-oracle:latest"

exit $EXIT_CODE
```



```
[exec] docker rm -f 7ed542733a01 76340c97c6ed f45e936b27a3 b2b7cafba6a5
```

```
7ed542733a01
```

```
76340c97c6ed
```

```
f45e936b27a3
```

```
b2b7cafba6a5
```

```
guillem@mbp ~/Projects/meetic/back-ws-private/php-api-inbox <chore/use-sql-import-vendor-gcanal>
```

```
$ # on lance toute la suite de test
```

```
guillem@mbp ~/Projects/meetic/back-ws-private/php-api-inbox <chore/use-sql-import-vendor-gcanal>
```

```
$ ./task-tests.sh a
```

```
Every 1,0s: docker ps --filter=name=php-api-inbox* --format="{{.Name}} {{.Command}}"
```

```
Thu Mar 31 16:56:28 2016
```

```
[0] 0:zsh*
```

```
"guillem@mbp: ~/Projec" 16:56 31-mar-16
```

```
▶ 00:00
```

The background is a dark blue space scene. It features several stylized celestial bodies: a large planet with horizontal stripes in the bottom left, a smaller planet with a ring in the top left, a planet with a ring in the top right, and a planet with vertical stripes in the middle right. A rocket with a bright orange flame is positioned on the left, pointing towards the center. A small, elongated object is in the bottom right. Numerous small white dots representing stars are scattered across the background.

**MERCI**  
**QUESTIONS ?**