



# UTILISER DOCKER POUR LANCER UN PROJET DRUPAL

**Le pourquoi et le comment 😊**

Créée par Claire Desbois / [@calystod](#)



# QUI SUIS-JE?

Claire Desbois

Développeuse Drupal depuis 6 ans

Travaille chez Symétris

Profil Drupal: <https://www.drupal.org/u/calystod>

# LA PROBLÉMATIQUE D'UN ENVIRONNEMENT LOCAL

Travail en équipe avec différentes machines



DIFFÉRENTS SYSTÈMES

**ARGAH!**

**IL FAUT CONFIGURER CHAQUE  
MACHINE**





**VERSIONS INCOMPATIBLES ENTRE  
LES PROJETS**

## Environnement de développement éloigné de celui de production

- Logiciels différents
- Versions différentes
- Configurations différentes





# COMMENT CRÉER DE LA PERSISTENCE ENTRE CHAQUE MACHINE DE DÉVELOPPEMENT?

En virtualisant son environnement 😊

**COMMENT? ON SE PARTAGE UNE  
IMAGE DE VM PAR PROJET VIA CLÉ  
USB? 🤖**

Heureusement, non.

# QUELS SONT LES SOLUTIONS QUI EXISTENT?

Les solutions les plus populaires

- Vagrant
- Docker

# VAGRANT

Outils de création et de configuration de machine virtuelle

Fonctionne avec beaucoup de logiciels de virtualisation (VirtualBox, VMWare, Hyper-V, etc)

Installation et configuration de la VM via du code (le provisioning)

## AVANTAGES

Extrêmement configurable et puissant  
Beaucoup de recettes proposées par la  
communauté

# INCONVÉNIENTS

C'est compliqué

C'est lourd et lent à lancer

C'est overkill pour faire du développement en local

*Symetris*  
DÉVELOPPEMENT WEB

# PROJET EXISTANT SPÉCIFIQUEMENT POUR DRUPAL

DrupalVM: <https://github.com/geerlingguy/drupal-vm>

Projet obèse (vous installez plus d'outils que vous n'en avez nécessairement besoin)

Ce serait une erreur de s'imaginer que comprendre Vagrant et Ansible est inutile

# DOCKER

Outils de création et de configuration de containers

Installation et configuration des containers via du  
code



## AVANTAGES

Très léger et rapide

Plus simple que Vagrant à apprendre pour faire quelque chose de basique

## INCONVÉNIENTS

Compatibilité moyenne selon votre OS

L'intégralité de l'outil n'est pas Open Source



# PROJETS EXISTANT SPÉCIFIQUEMENT POUR DRUPAL



Kalabox: <http://www.kalabox.io/>



DrupalVM: <https://github.com/geerlingguy/drupal-vm>

# COMMENT DOCKER FONCTIONNE?

Docker n'est pas un système de machines  
virtuelles

Un container est un ensemble d'éléments logiciels  
nécessaire pour votre application

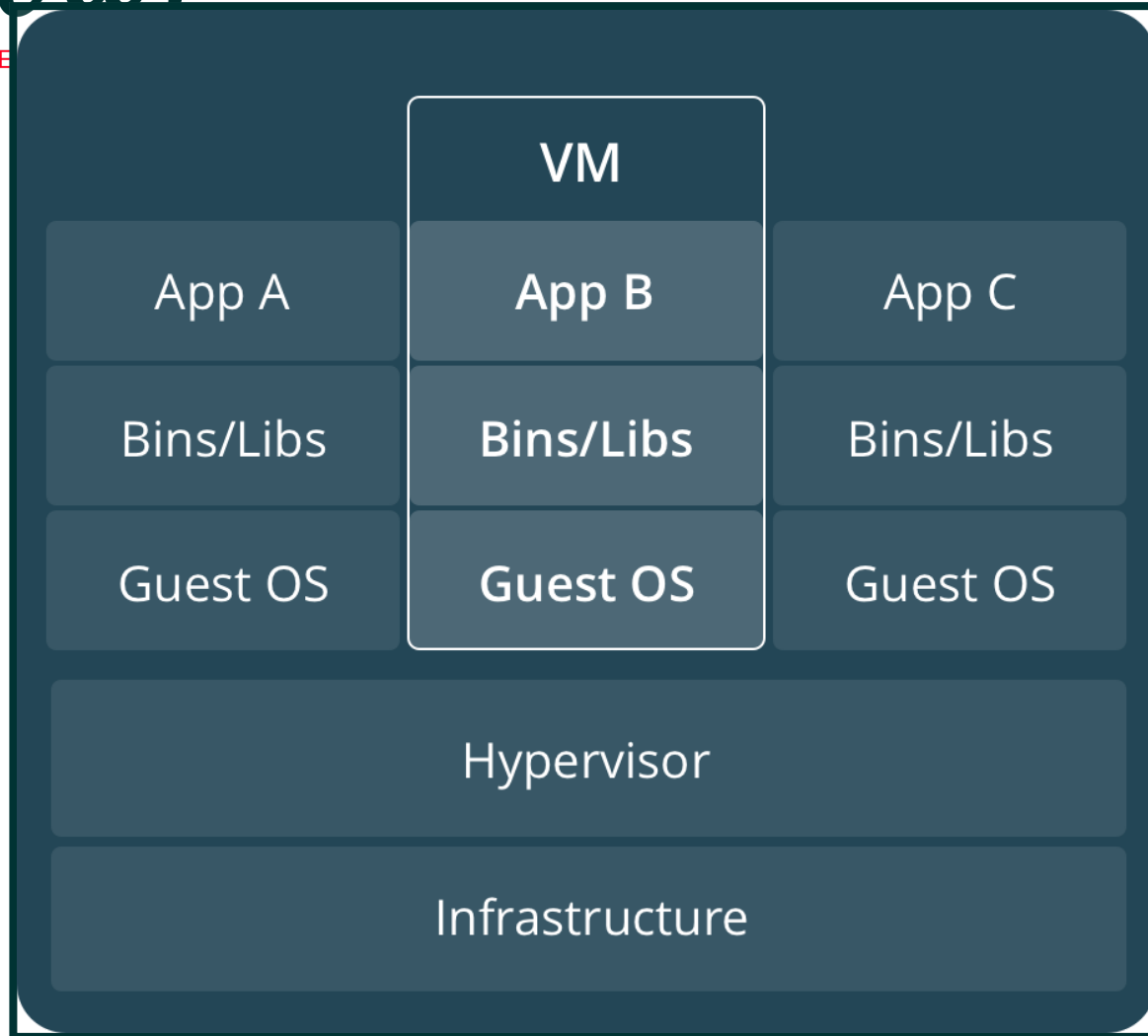
Un container ne fait pas rouler l'intégralité d'un système d'exploitation

Plusieurs containers peuvent partager le même système d'exploitation tout en restant isolés les uns des autres

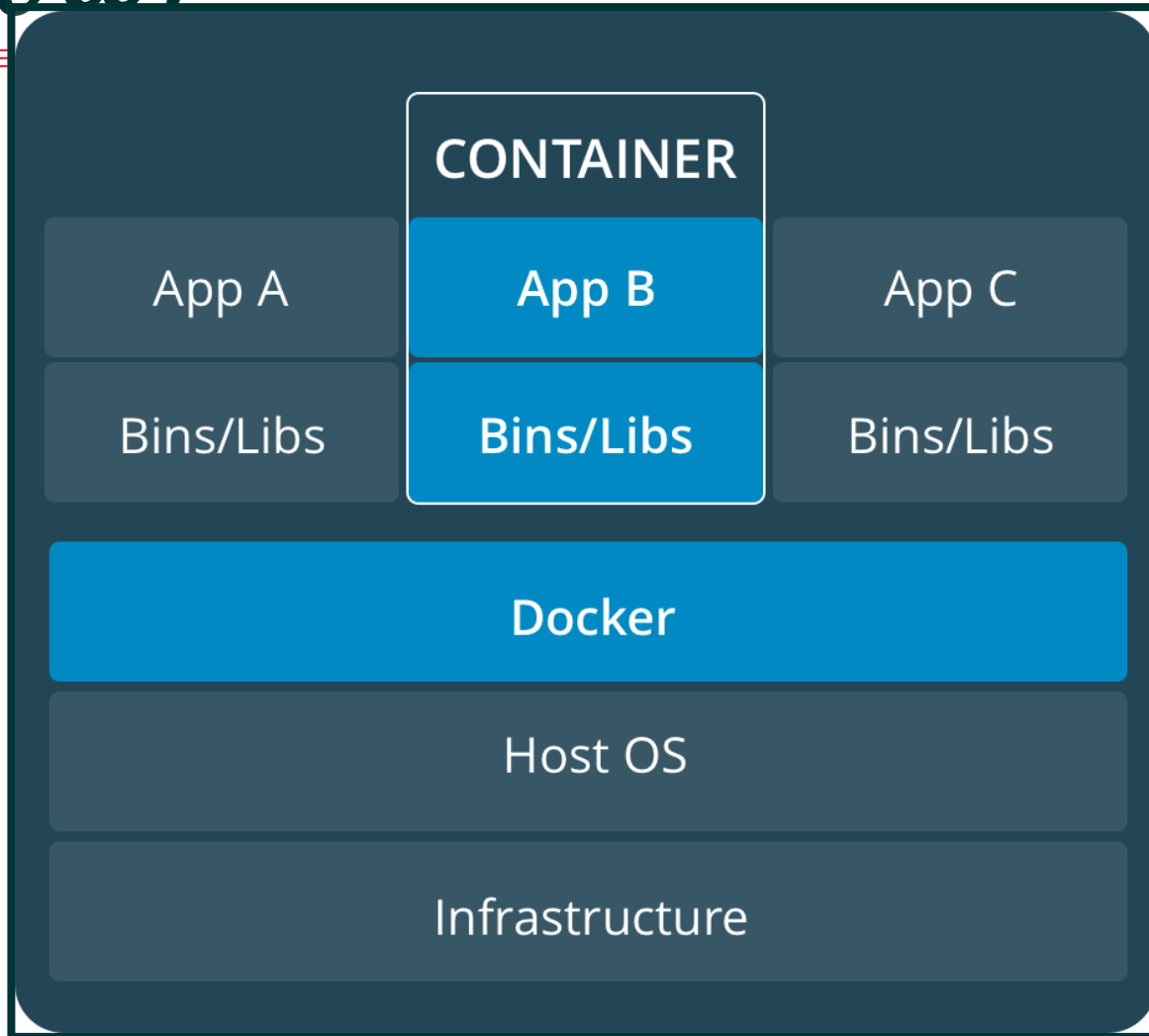


# Schéma d'architecture d'un système de VM

*Symétris*  
DÉVELOPPE



# Schéma d'architecture de Docker





Pas de mémoire persistante sur un container

On utilise des volumes

Il est facile de faire communiquer des containers  
entre eux

On peut s'en servir pour gérer plein de  
microservices 😊

**METTONS EN PLACE UN 1ER  
PROJET**

**FAISONS ROULER DRUPAL AVEC DOCKER**

*Symetris*

DÉVELOPPEMENT WEB

# DE QUOI AVONS-NOUS BESOIN?

D'une base de donnée

D'un serveur web

De PHP

Et d'outils divers pour développer (Drush,  
Composer, Grunt, etc)

# COMMENT ON CRÉE UN CONTAINER?

On a besoin d'une image

2 choix:

La créer

Utiliser une image existante

Docker Hub: <https://hub.docker.com/>

## Utilisons Docker-Compose

Outil pour définir et lancer plusieurs containers





## Container pour MySQL

```
volumes:  
  dbvolume:/var/lib/mysql  
  
db:  
  image: mariadb  
  environment:  
    - MYSQL_ROOT_PASSWORD=root_password  
    - MYSQL_DATABASE=my_db  
    - MYSQL_USER=drupal  
    - MYSQL_PASSWORD=drupal  
  volumes_from:  
    - dbvolume
```

## On créer le container PHP-FPM

```
php-fpm:  
  image: drupaldocker/php-dev:7.0-fpm  
  volumes:  
    - .:/var/www/html:rw  
  environment:  
    - MYSQL_DATABASE=my_db  
    - MYSQL_USER=drupal  
    - MYSQL_PASSWORD=drupal  
  links:  
    - db
```

## On créer le container NGinx

```
web:
  image: drupaldocker/nginx
  environment:
    DOCROOT: /var/www/html/web
  ports:
    - 80:80
  volumes:
    - ./var/www/html:rw
  links:
    - php-fpm:php
```



# créer le container pour gérer les utilitaires en ligne de commande

```
php-cli:
  build:
    context: .
    dockerfile: Dockerfile-php-cli
    args:
      USER_UID: 1000
  volumes:
    # Data of application
    - ./var/www/html:rw
    # Share the local user's composer cache and settings
    - ~/.composer:/home/php/.composer:rw
    # Share the local user's SSH keys and configuration
    - ~/.ssh:/home/php/.ssh:ro
    - ~/.ssh/known_hosts:/home/php/.ssh/known_hosts:rw
```

```
FROM drupaldocker/php:7-cli

ARG USER_UID=1000

# Install needed CLI tools (for Drush, MySQL a
RUN mkdir -p /etc/drush \
&& echo "<!--?php \$options['root'] = '/var/www"

RUN useradd --uid ${USER_UID} --create-home ph

CMD ["bash"]

USER ${USER_UID}
```



**DEMO**

Site de Docker: <https://www.docker.com/>

Documentation de Docker:

<https://docs.docker.com/>

Documentation de Docker Compose:

<https://docs.docker.com/compose/>

Projet de dépôt pour cette présentation:

<https://github.com/Calystod/docker-drupal-example>



**QUESTIONS?**

Merci à tous 😊