

# Devbox in Docker

<http://dduportal.github.io/presentations/mixit-2015-devbox-docker/index.html>

<http://dduportal.github.io/presentations/mixit-2015-devbox-docker.pdf>

# Agenda :

1. Introduction
2. Pourquoi ce talk ?
3. "Full stack" Engineer
4. Demo time !

# Introduction

Intro

# Nous venons de Belgique

Qui sommes-nous ?

Intro

# Nous venons de Belgique

Qui sommes-nous ?

Le pays du chocolat...



Intro

# Nous venons de Belgique

Qui sommes-nous ?

**Le pays de la bière...**



Intro

# Nous venons de Belgique

Qui sommes-nous ?

**Le pays où on ne se prend pas (trop) au sérieux...**



Intro

# Nous venons de Belgique

Qui sommes-nous ?

Le pays où on tourne en rond ?



## Intro

# Jean-Marc MEESEN

Qui sommes-nous ?

Vous pourriez me retrouver dans un surgelateur de grande surface



- Senior developer @ Worldline, newcomer coach, development infrastructure
- Developement Middleware sur Fuse / ServiceMix / Camel
- Trop timide pour participer activement à des projets Open Source. Quelques contributions à Asciidoctor et Freeplane.
- [jean-marc@meessen-web.org](mailto:jean-marc@meessen-web.org), @JM\_Meessen

## Intro

# Damien DUPORTAL

Qui sommes-nous ?



- DevOps Engineer @ Worldline
- Grimpeur
- Français émigré en Belgique
- Enseignant la HA et les SIGs (ENSG, EPSI)
- Contact :
  - Gmail : [damien.duportal@gmail.com](mailto:damien.duportal@gmail.com)
  - Twitter : [@DamienDuportal](https://twitter.com/DamienDuportal)
  - Github : [dduportal](https://github.com/dduportal)

# Intro

Qui sommes-nous ?

Remerciements

# Merci !

- Organisateurs et bénévoles Mix-It
- à vous, de venir !
- Aux personnes ayant réalisé et crus en ces outils

## Intro

Qui sommes-nous ?

Remerciements

Et vous ?

# Faisons connaissance...

- Des devs ? Des Ops ? Des chefs ? Autre ?
- Des seniors (des vrais) ? Des moins seniors ? Des juniors ?
- Expérience avec la virtualisation ?
- Vous souvenez-vous de votre premier emploi ?

# Pourquoi ce talk ?

## Pourquoi ce talk ?

# Pourquoi ce talk ?

- J'ai participé à beaucoup d'aventures ces 30 dernières années.
- Toujours aussi enthousiaste pour apprendre et partager
- MAIS, une de mes grandes frustrations : La perte de temps et d'énergie au démarrage de chaque nouvelle aventure.

Pourquoi ce  
talk ?

Quel est le problème ?

# Quel est le problème ?

- Gaspillage d'énergie au mauvais moment
- Mise en place de l'environnement coûteuse :



- coût de maintenance
- "Nivellement par le bas"

## Objectif :

Devbox très rapide à déployer et du premier coup

Pourquoi ce  
talk ?

Quel est le problème ?

Solutions ?

# Solutions "d'avant"

- les terminaux sur le mainframe
- les PC avec images (quid des utilisateurs nomades ?)
- virtualisation avec Vagrant/VB par ex

# Pourquoi ce talk ?

Quel est le problème ?

Solutions ?

Constat

# Constat

- DEVBOX = TOOLBOX = Trousse à outils = Camion atelier



- Même problèmes que pour nos applications, donc mêmes solutions :
  - DaaT : Devbox as a Tool (comme avant)
  - DaaS : Devbox as a Software
  - DaalI : Devbox as an Infrastructure
  - DaaLt :Devbox as a Learning tool

# Pourquoi ce talk ?

Quel est le problème ?

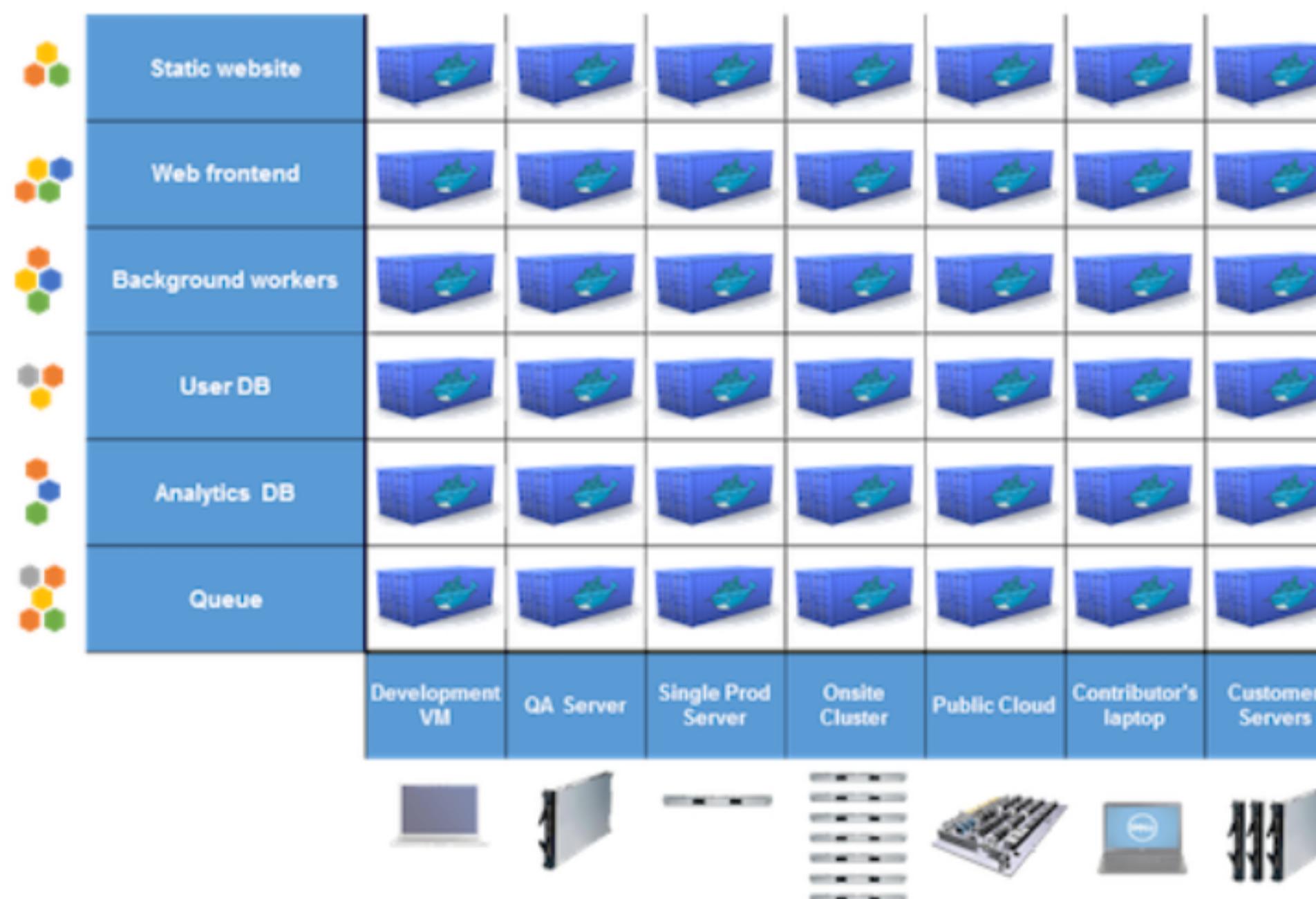
Solutions ?

Constat

Docker !

# Docker !

- Docker résouds le pb de "Matrix of hell"
- Docker contractualise
- Docker est rapide
- Docker est "à la mode"
- Permet de nouveaux produits/solutions
  - Souvenez vous de la "libération" du GPS par B. Clinton



# Utiliser les recettes de l'ingénierie logicielle

Nous sommes TOUS des ingénieurs "Full stack"

"Full stack"

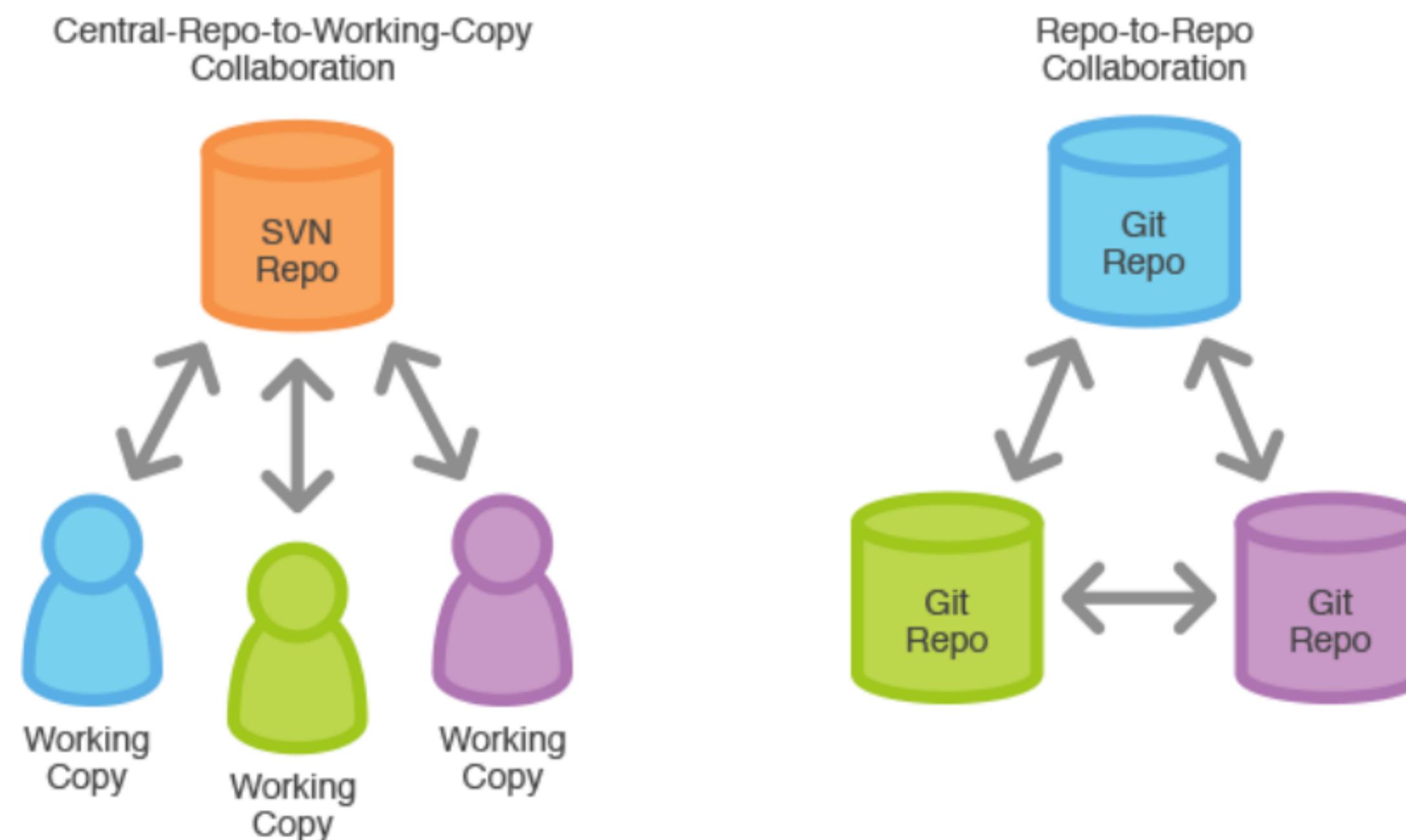
Engineer

Code == Valeur

# "La vérité est dans le code"

(<http://programming-motherfucker.com>)

Utilisez des SCMs, si possible des DVCS => nouvelles façons de travailler :



"Full stack"

Engineer

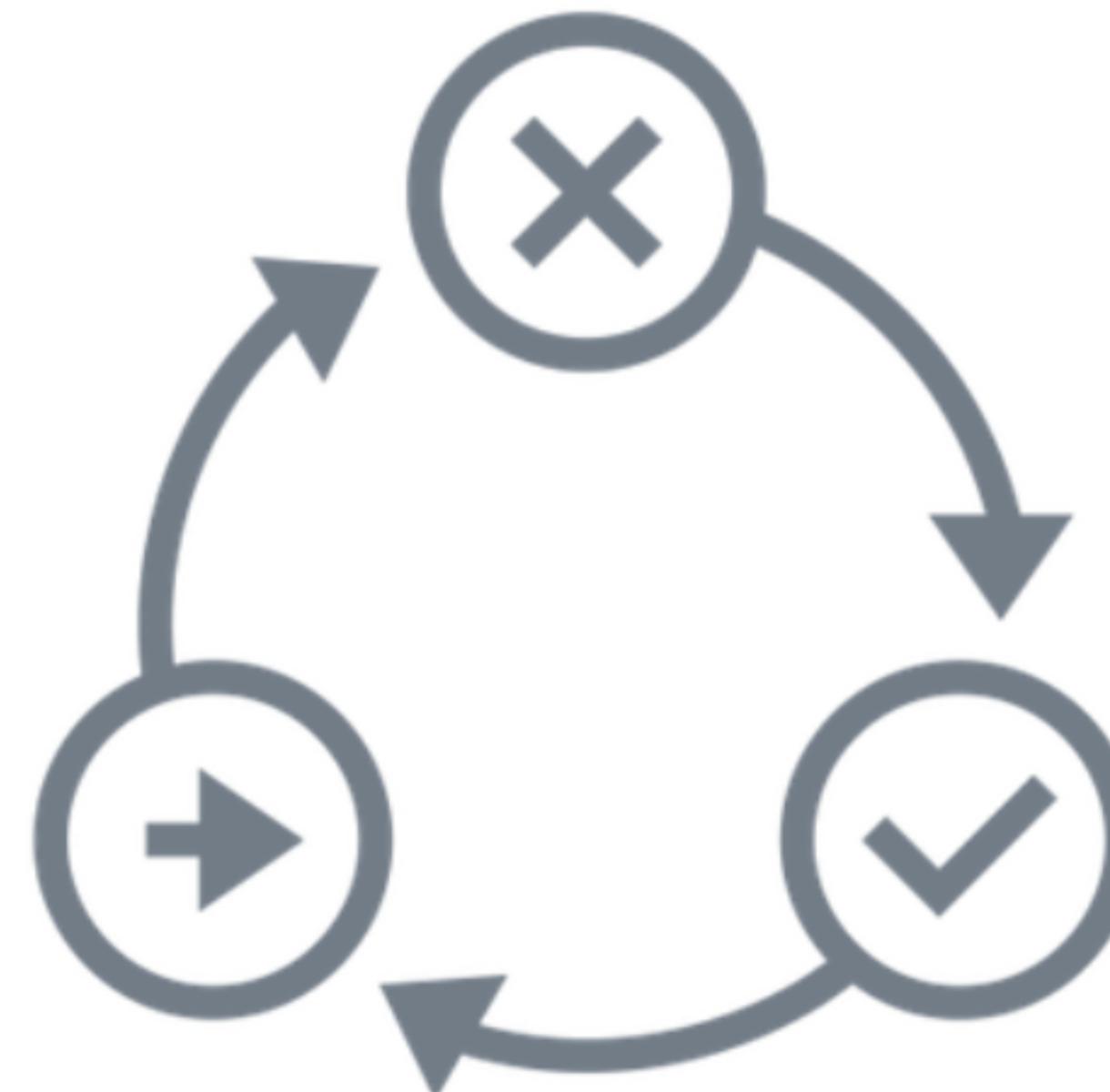
Code == Valeur

Nouvelles pratiques

# "Nouvelles" pratiques

N'ayons pas peur d'*essayer, plusieurs fois*, toutes ces nouvelles façons de faire, sans être dogmatiques

- Agile
- TDD/BDD/Doc. as code
- DevOps



"Full stack"

Engineer

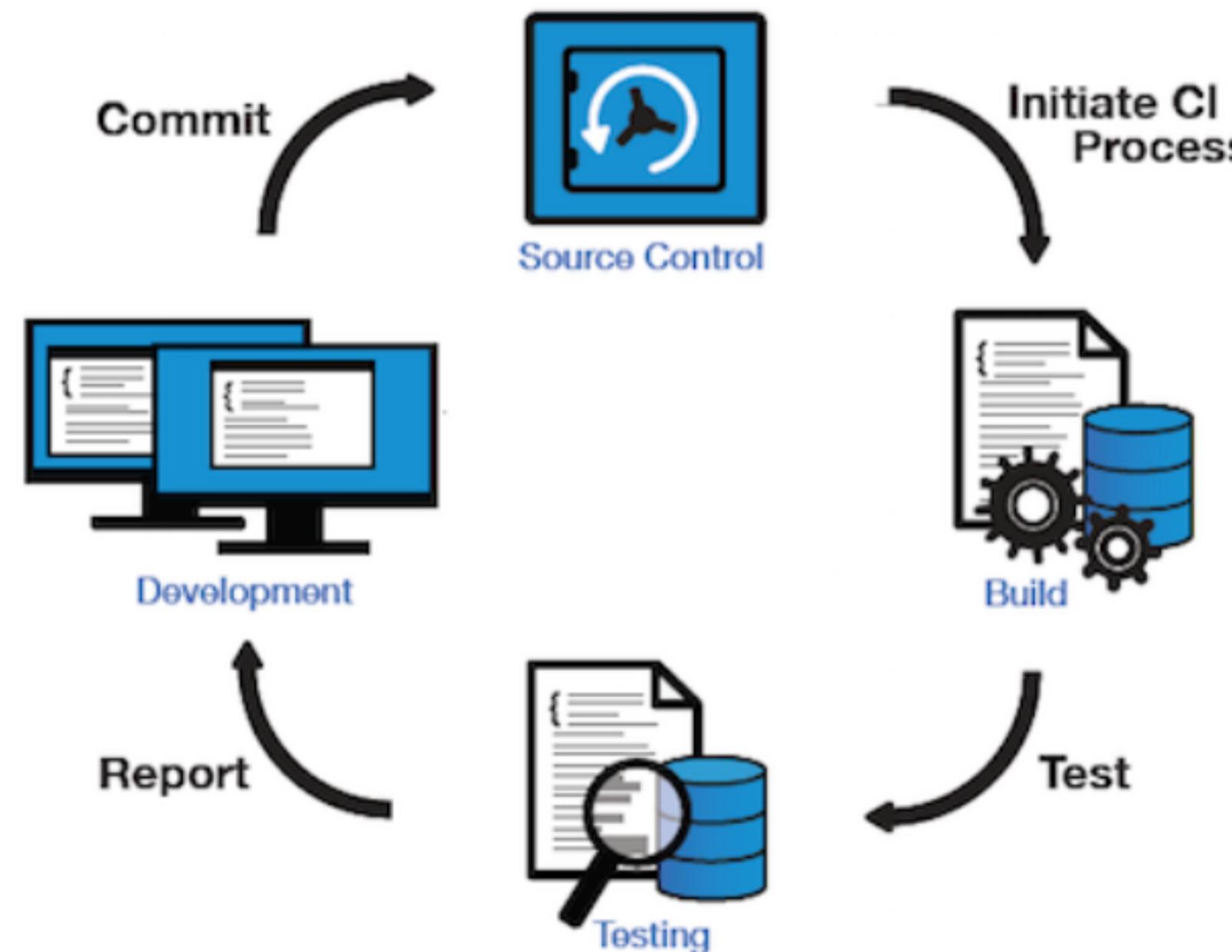
Code == Valeur

Nouvelles pratiques

Continuous\*

# Continuous\*

- Continuous Integration
- Continuous Testing (TDD right ?)
- Continuous documentation
- Continuous Benchmarking
- Continuous delivery



"Full stack"

Engineer

Code == Valeur

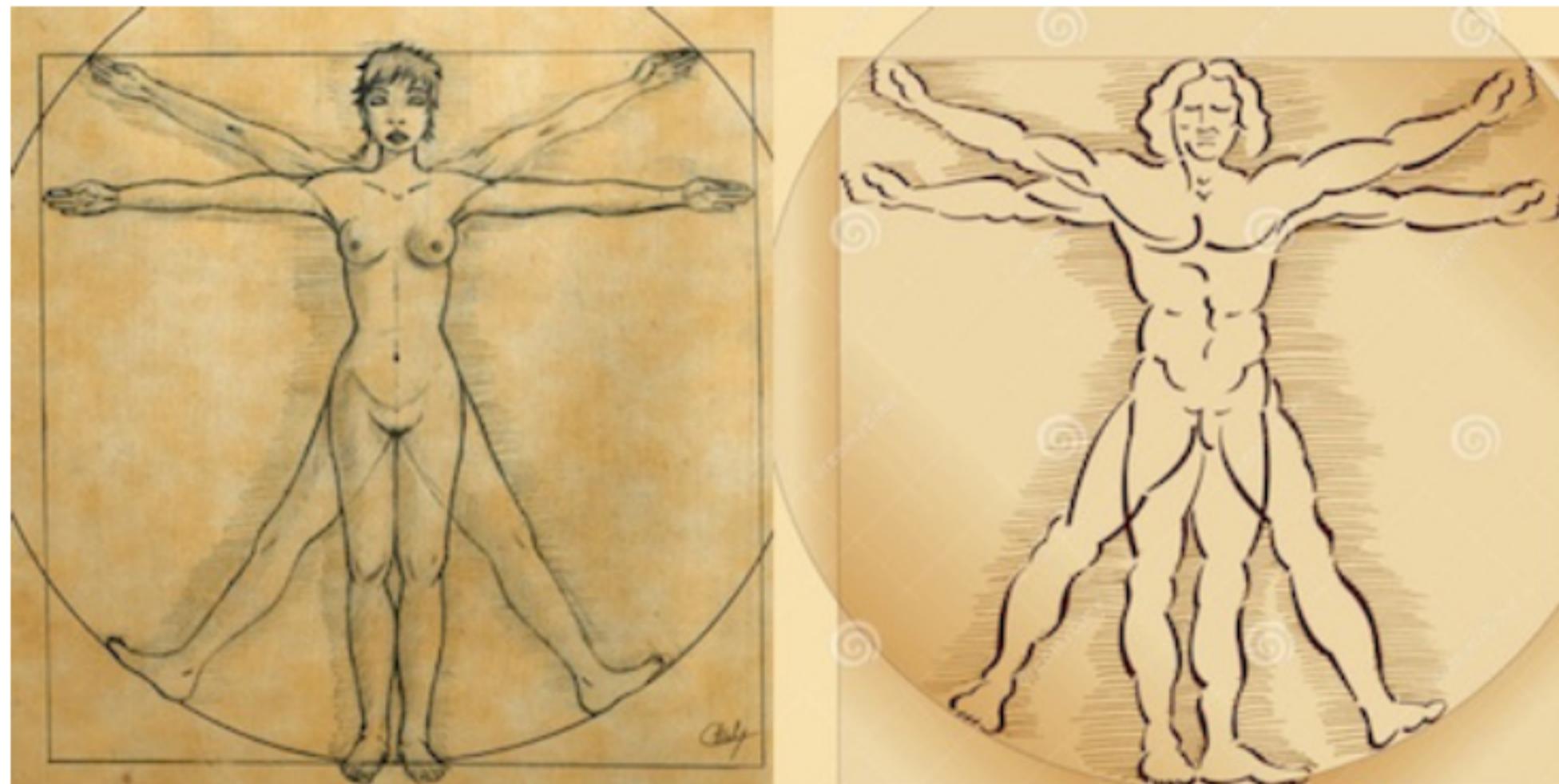
Nouvelles pratiques

Continuous\*

"Human Stack"

# "Human stack"

Culture



Confiance

- "Compagnonnage"
- Eat your own dog food

# Démo Time !

# Démo Time !

# Démo Time !

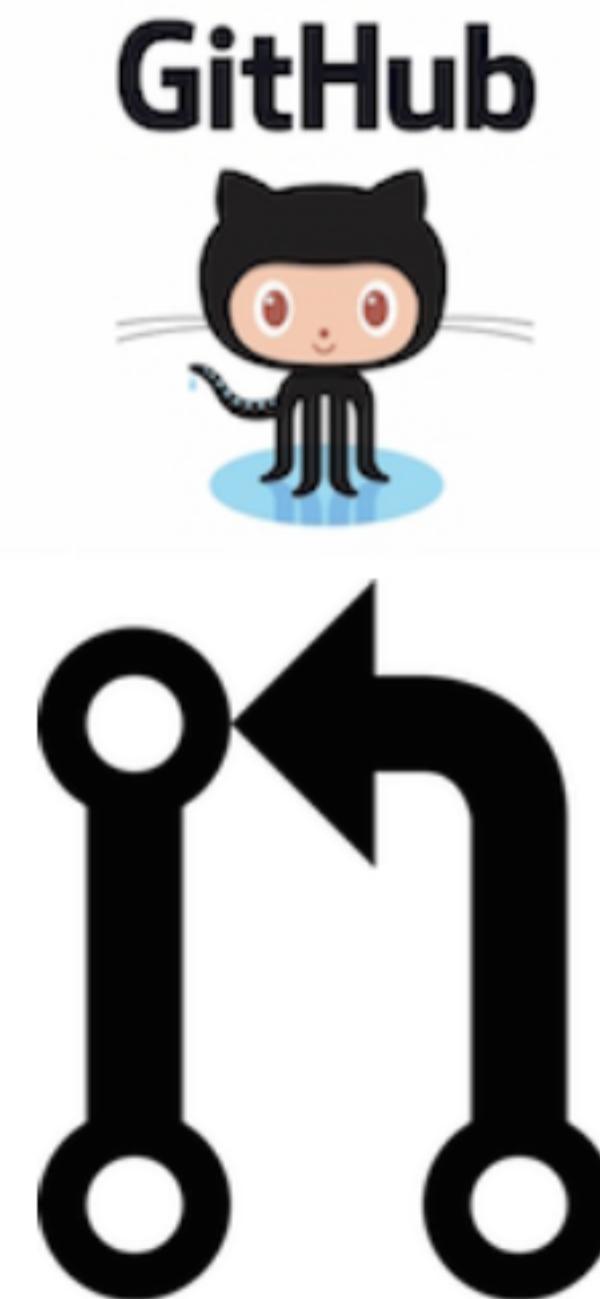
- Objet : Spring boot "hello world" application
- Java DK, Maven, IntelliJ Idea, Docker, Compose
- Open and reusable code :  
<https://github.com/jmMeessen/devbox>
- Use cases :
  - Onboarding
  - Dévelopement simple basé sur Spring Boot
  - Maj IDEA sans casser le dev workflow
  - Portabilité
  - Legacy (Long living applications)

Démo Time !

# SCM :Git (on Github)

Devbox as a code

- Public
- Partagé
- Pull-Request systématique (Github workflow)



# Démo Time !

# TDD/BDD

Devbox as a code

Utilisation de **Bats** (testing en bash) au maximum

```
@test "Check JDK presence" {
    docker run --user "dockergx" which java
}
```

- Test Driven Developemt (BDD) oriented
- Basé sur Bash : compromis "démarrage rapide" vs. maintenabilité
- Objectif pour une fonctionnalité :
  - Rouge : Rédaction des tests qui vont être "fail"
  - Green : Implémentation, c'est OK
- Pragmatisme : ne pas tout tester non plus, par petits pas !

Lancement facile :

```
$ make test
```

Démo Time !

# Continuous\*

Devbox as a code

Utilisation de **CircleCI** :

- Basé sur les hooks Github (git push == build)
- Support de Docker
- Configuration as a code :

```
circle.yml:  
  
machine:  
  services:  
    - docker  
dependencies:  
  override:  
    - make build  
test:  
  override:  
    - make test
```

Démo Time !

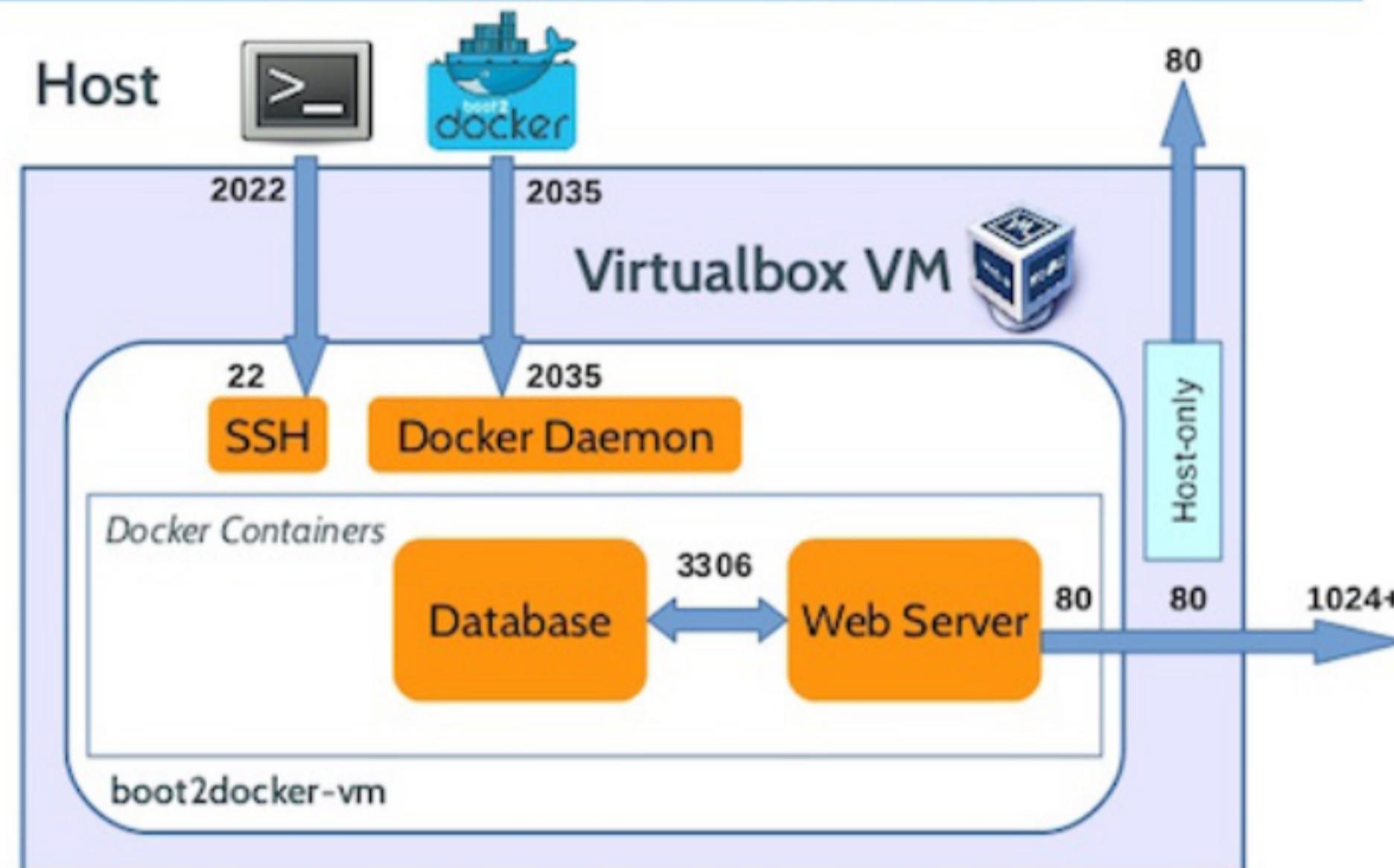
# Docker stack

Devbox as a code

Utilisation de **boot2docker** :

- Multi-plateforme (virtualbox)
- Support officiel de Docker

## Boot2docker architecture



Jirayut Nimsaeng

Docker for DevOps and Continuous Delivery Workshop  
October 11, 2014 @ OSS Festival 2014

# Démo Time !

# Cinématique

Devbox as a code

Utilisation de **GNU Make** :

- Indépendance cinématique / outils (On peut s'affranchir de Docker)
- Multi-platforme

Makefile:

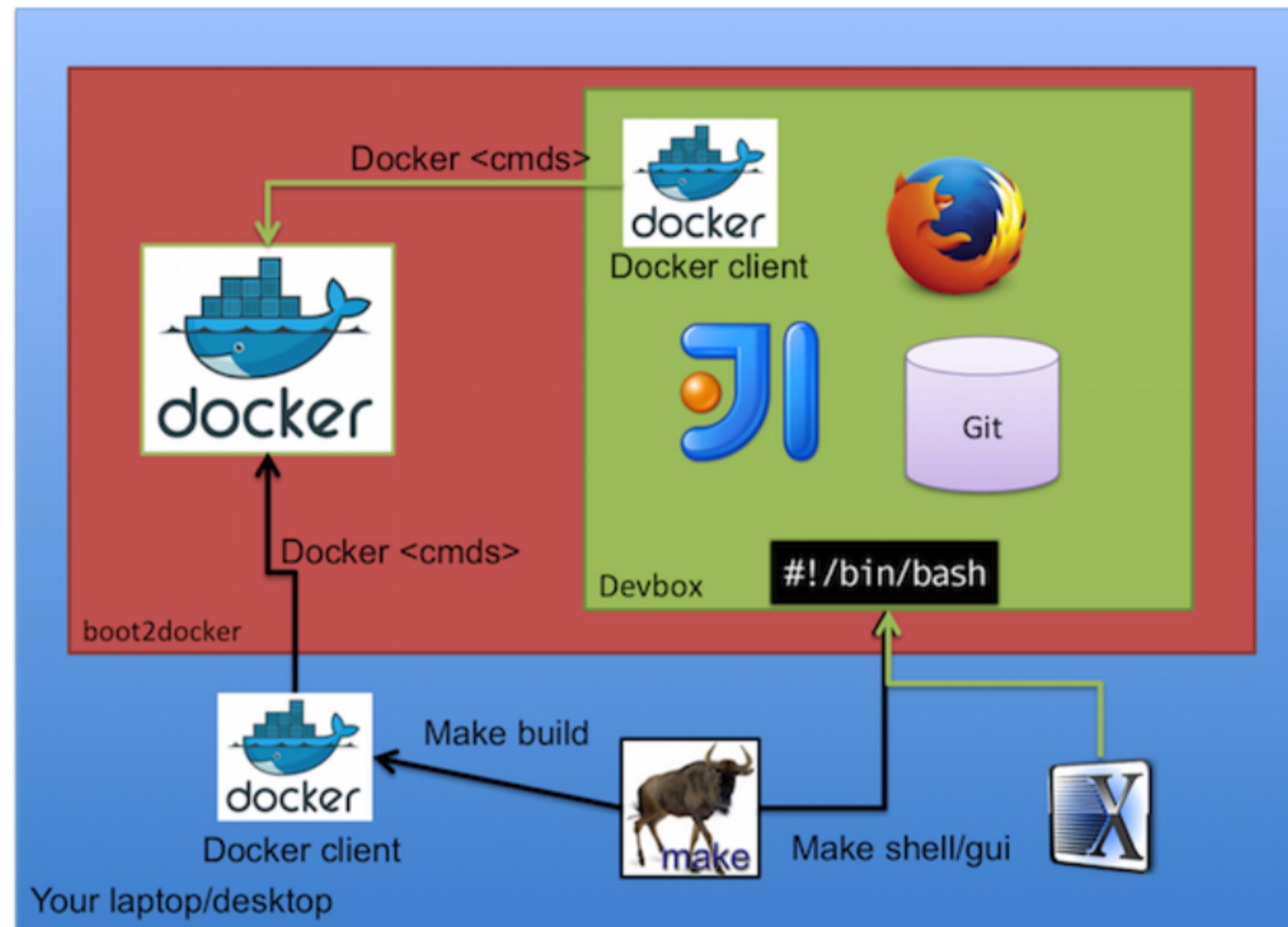
```
all: build test

build:
...
start:
...
test:
...
backup:
...
clean:
...
```

# Démo Time !

# Demo 1 :

Devbox as a code



# Demo 1

See code on Github :

<https://github.com/jmMeessen/devbox/tree/Demo1>

# Démo Time !

Devbox as a code

Devbox as an  
infrastructure

## Données

- Comment conserver ses données si on détruit/met à jour la VM ?
- Comment sauvegarder ses données ?

## Réseau

- Travailleurs nomades, pauvres employé de SSI...
- Ressources locales : cache proxy, registre docker, cache Maven ?

Démo Time !

Devbox as a code

Devbox as an infrastructure

# Devbox as an infrastructure

Solution : Docker-compose (Was known as Fig) !

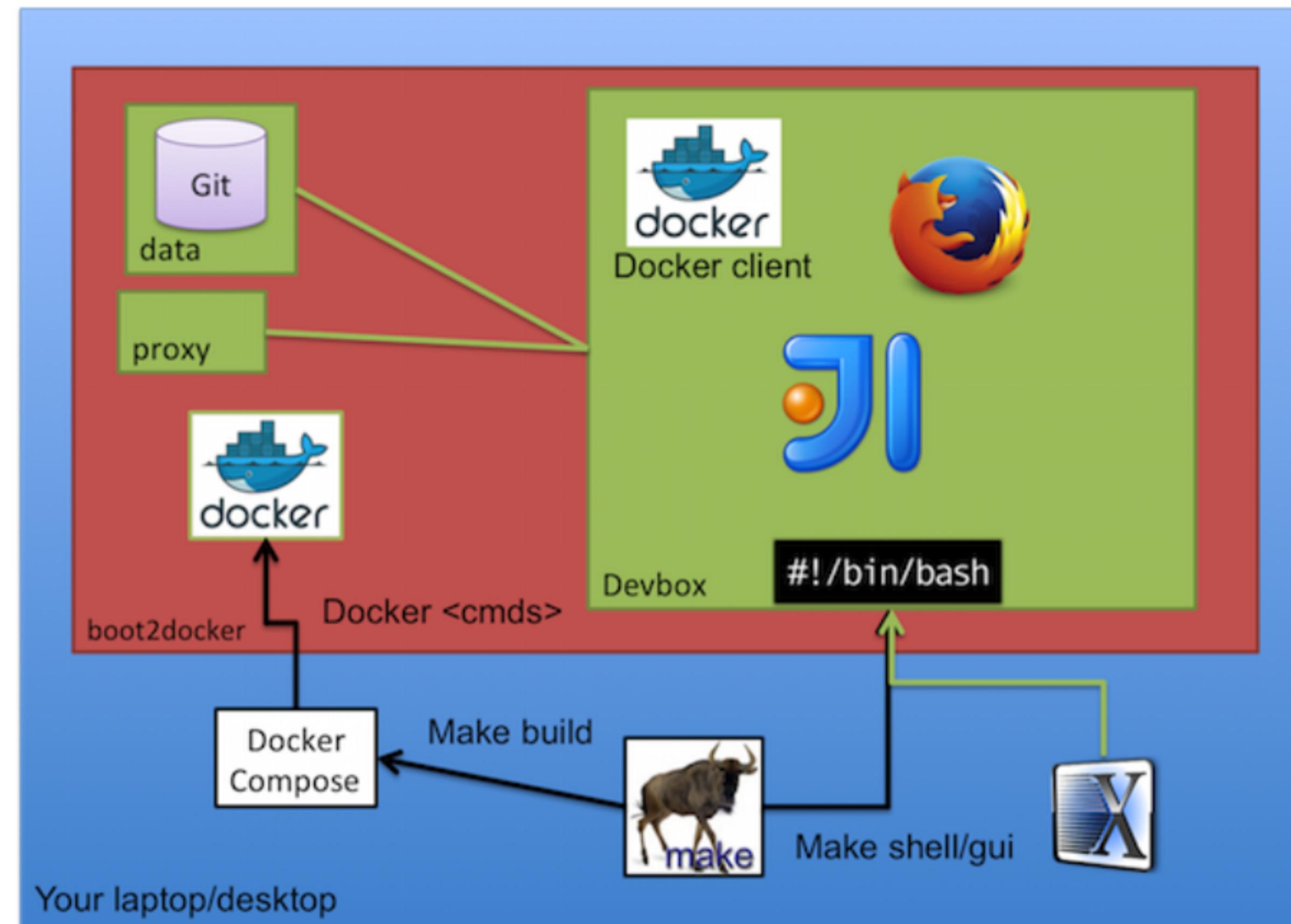
```
web:  
  image: my-java-app:1.0.0  
  links:  
    - db  
  ports:  
    - "8000:8000"  
db:  
  image: postgres
```

# Démo Time !

## Demo 2 :

Devbox as a code

Devbox as an infrastructure



# Demo 2

See code on Github :

<https://github.com/jmMeessen/devbox/tree/Demo2>

# Conclusion

## Qu'a t'on appris ?

Qu'a t'on appris ?

- Jean-Marc :
  - Nouvelles méthodes & technos
  - Confirmation que de la conception à la production, le flux est continu (mort aux silos !)
  - Apprentissage du travail communautaire/ OSS (des valeurs qui me sont très proches) aussi à distance.
- Damien :
  - Peer-working : programmer, revue, présentation !
  - "Faire faire" : 3ème étape du formateur
  - Confiance en NOUS

# Conclusion

Qu'à t'on appris ?

Le bon et le moins bon

# Pas cool

- GUI : des difficultés et bugs bizarre
- Adhésion et motivation nécessaires

# Cool

- Performances
- CLI : parfait
- Excellent indicateur de la culture de vos devs.
- Notre intuition est correcte:
  - Encore rien rencontré de bloquant et très, très prometteur.
  - Il y a encore du travail
  - j'attends avec impatience l'épreuve du feu (situation réelle)

# Améliorations

- Processus de release rigoureux avec le DockerHub
- README à améliorer (contributions ? )

The end  
Thanks ! Questions ?

# Crédits images

- chokotoff2.jpg - <http://www.bonbonsgourmands.fr/Bonbons-envelopp%C3%A9s-34-1.html>
- belgianBeers.jpg - <http://www.lessentiel.lu/fr/lifestyle/tendances/story/23830677>
- defile20106.jpg - <http://www.noblesseetroyautes.com/2010/07/la-famille-royale-belge-au-defile-militaire-de-la-fete-nationale/>
- bruxelles\_brussels.jpg - <https://chaudoir.wordpress.com/2009/11/25/tout-est-dit-dans-un-seul-panneau/>
- sortieDePiste.png - <http://www.pixauto.net/2011/tour-auto/circuit-du-vigeant.php/> and <http://motorsport.nextgen-auto.com/12-tours-pour-Grosjean-et-une-sortie-de-piste-Maldonado-bloque-au-stand,73785.html>
- toolbox.png - <http://www.autostock04.fr/> and <http://buildingabrandonline.com/BrianElrod/promote-products-with-an-affiliate-toolbox/>
- eliminates-matrix-from-hell.png - <http://dduportal.github.io/presentations/docker-meetup-lyon-20140528/>
- continuousintegrationcycle.png - <http://blog.anthonybaker.me/2012/05/about-continuous-integration.html>
- vitr.jpg - ?
- gitflow.jpg - <https://www.atlassian.com/pt/git/workflows#!workflow-gitflow>