

# Quand Docker Rencontre les Pipelines

Jenkins Paris Meetup - 19 avril 2017

Pierre Beitz - [PierreBtz](#) - [@PiBtz](#) - [pierre-beitz.eu](#)

# Agenda

- Introduction à Docker

# Agenda

- Introduction à Docker
- Intégration avec les pipelines Jenkins

# Agenda

- Introduction à Docker
- Intégration avec les pipelines Jenkins
- Et l'infra?

# Agenda

- Introduction à Docker
- Intégration avec les pipelines Jenkins
- Et l'infra?
- Demo sur un cas réel

# Pourquoi des Containers ?

# Pourquoi des Containers ?



B 4433

Source: State Library of South Australia



# Pourquoi des Containers ?





# Et dans le monde logiciel...

- On encapsule logiciel et dépendences

# Et dans le monde logiciel...

- On encapsule logiciel et dépendences
- On manipule des containers

# Et dans le monde logiciel...

- On encapsule logiciel et dépendences
- On manipule des containers
- L'environnement d'exécution est connu et reproductible

# Et dans le monde logiciel...

- On encapsule logiciel et dépendances
- On manipule des containers
- L'environnement d'exécution est connu et reproductible
- Le monitoring est facile

# Qu'est-ce qu'un container?



# Qu'est-ce qu'un container?

- Composition de différentes primitives du noyau linux:

# Qu'est-ce qu'un container?

- Composition de différentes primitives du noyau linux:
  - namespaces

# Qu'est-ce qu'un container?

- Composition de différentes primitives du noyau linux:
  - namespaces
  - cgroups

# Qu'est-ce qu'un container?

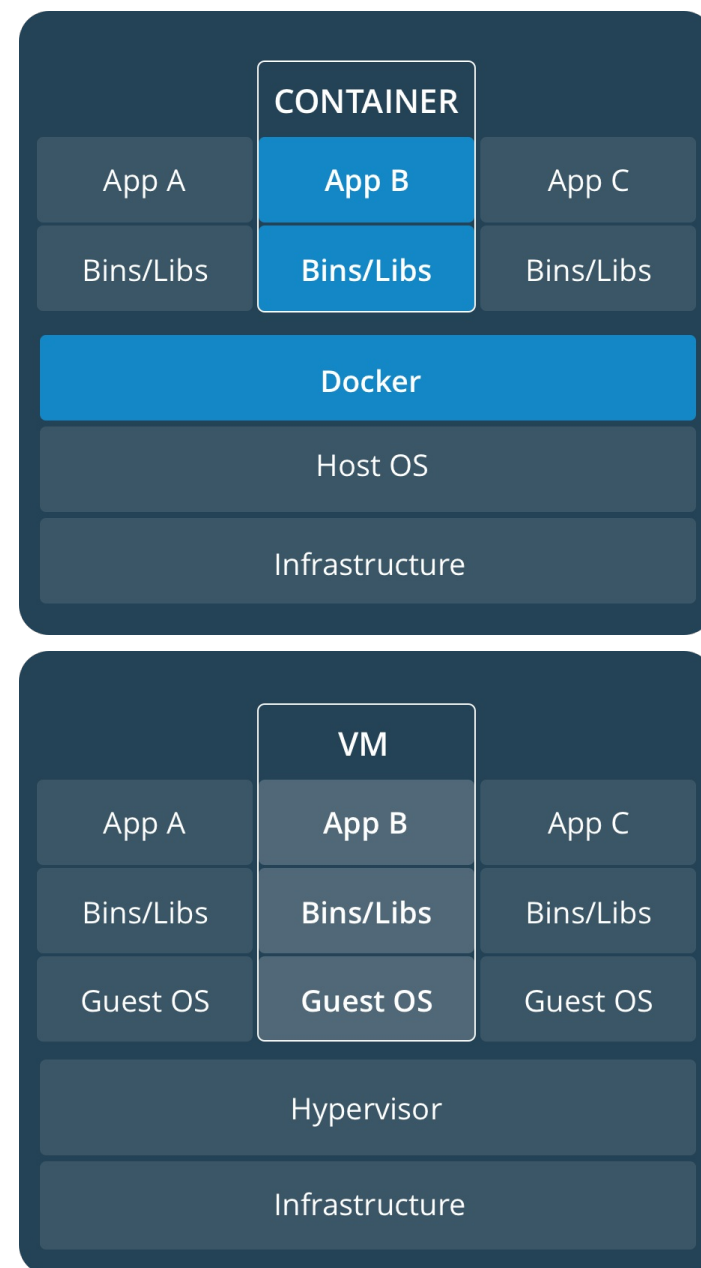
- Composition de différentes primitives du noyau linux:
  - namespaces
  - cgroups
  - et d'autres: apparmor, selinux...

# Qu'est-ce qu'un container?

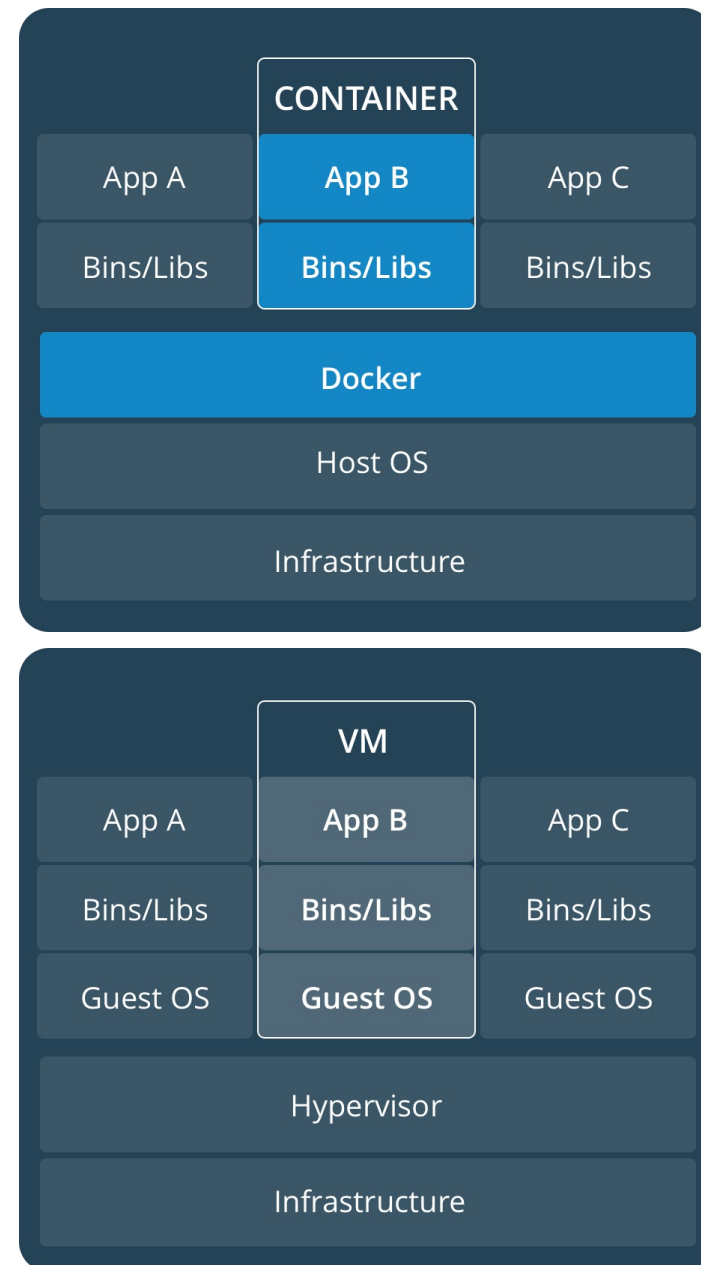
- Composition de différentes primitives du noyau linux:
  - namespaces
  - cgroups
  - et d'autres: apparmor, selinux...
- A partir de ça, pleins de projets différents: docker, rkt, LXC/LXD



# Oui, mais moi, j'ai déjà les VMs!

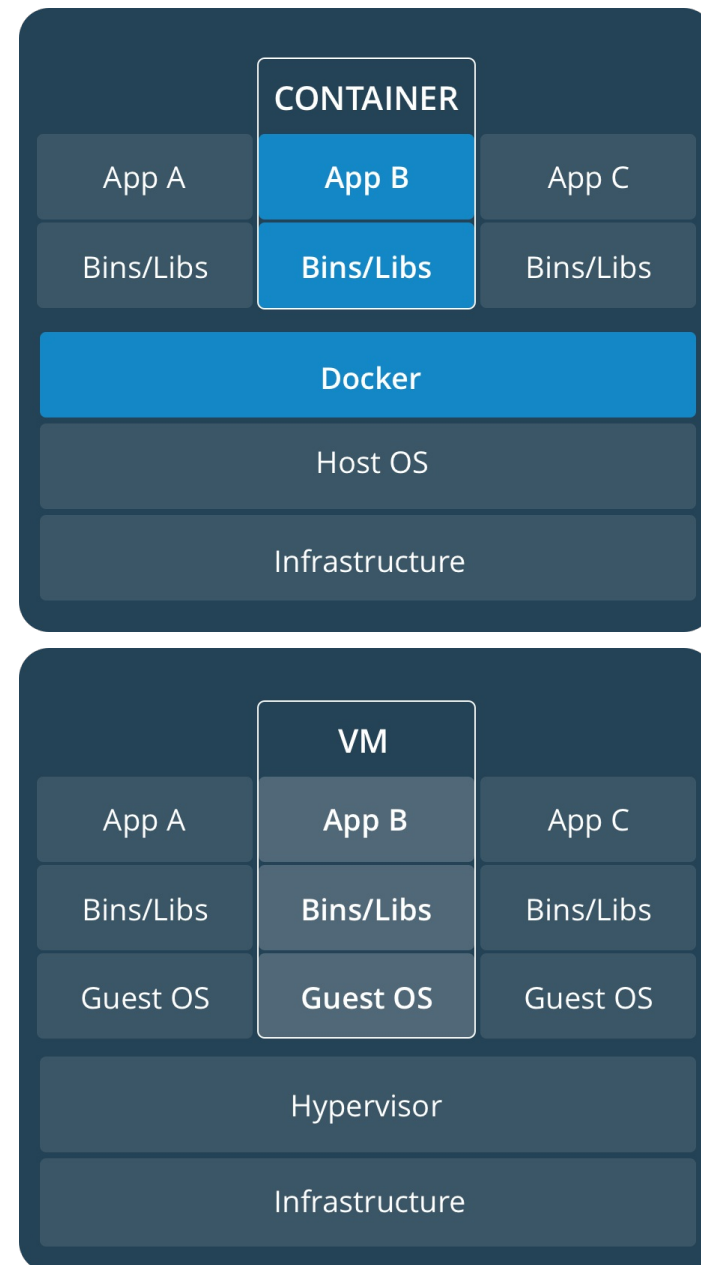


# Oui, mais moi, j'ai déjà les VMs!



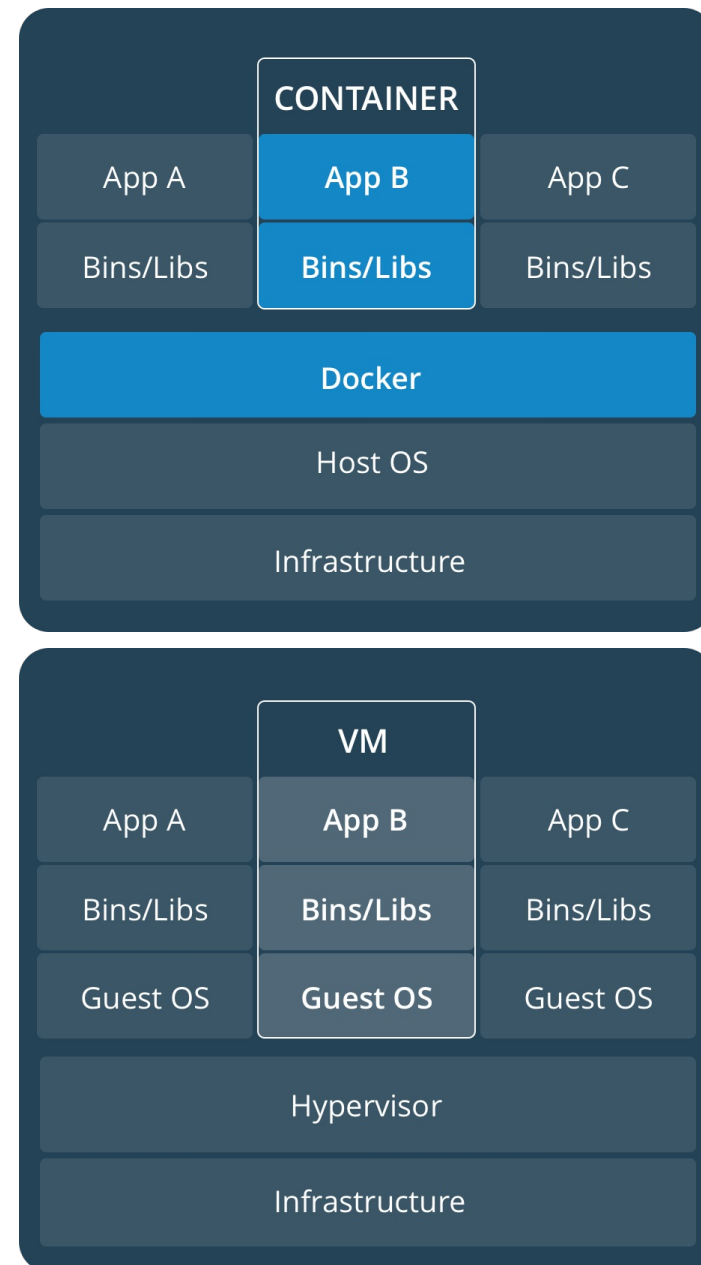
- Pas la même implémentation (émulation hardware, OS...)

# Oui, mais moi, j'ai déjà les VMs!



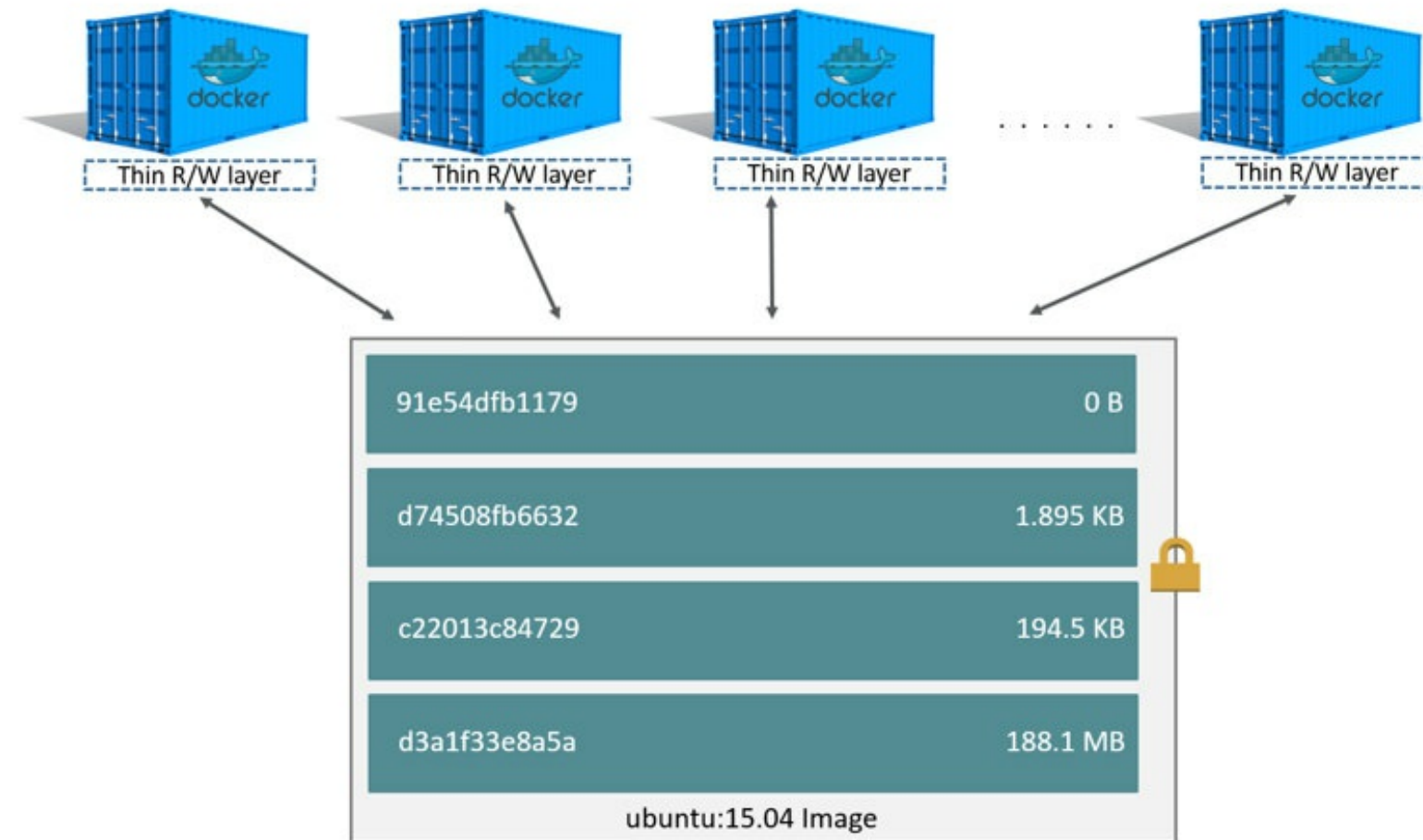
- Pas la même implémentation (émulation hardware, OS...)
- Un container démarre beaucoup plus vite qu'une VM

# Oui, mais moi, j'ai déjà les VMs!



- Pas la même implémentation (émulation hardware, OS...)
- Un container démarre beaucoup plus vite qu'une VM
- L'image d'un container peut tenir dans une centaine voire une dizaine de Mo

# Un peu de Vocabulaire Docker





# Un peu de Vocabulaire Docker

## Container:

- Créé à partir d'une image, démarré/redémarré/tué en quelques secondes

# Un peu de Vocabulaire Docker

## Container:

- Créé à partir d'une image, démarré/redémarré/tué en quelques secondes
- Un point d'entrée

# Un peu de Vocabulaire Docker

## Container:

- Créé à partir d'une image, démarré/redémarré/tué en quelques secondes
- Un point d'entrée
- Expose des ports

# Un peu de Vocabulaire Docker

## Container:

- Créé à partir d'une image, démarré/redémarré/tué en quelques secondes
- Un point d'entrée
- Expose des ports
- Stocke ses données sur des volumes

# Construire une image

- Dockerfile: la recette d'une image



# Construire une image

- Dockerfile: la recette d'une image

```
FROM groovy:2.4.10-jre8
MAINTAINER Pierre Beitz <pibeitz@gmail.com>
RUN echo "println \"Hello world!\"" > hello.groovy
CMD groovy hello.groovy
```

# Stocker une image

- Un repository central: [Docker Hub](#)

# Stocker une image

- Un repository central: [Docker Hub](#)
- Possibilité d'utiliser des repository privés

# Et Jenkins dans tout ça ?

- Un build a aussi des dépendences (ex outils de build)

# Et Jenkins dans tout ça ?

- Un build a aussi des dépendences (ex outils de build)
- Des solutions existent pour les gérer:

# Et Jenkins dans tout ça ?

- Un build a aussi des dépendences (ex outils de build)
- Des solutions existent pour les gérer:
  - On peut préconfigurer tous les agents de build avec les outils

# Et Jenkins dans tout ça ?

- Un build a aussi des dépendences (ex outils de build)
- Des solutions existent pour les gérer:
  - On peut préconfigurer tous les agents de build avec les outils
  - On peut se reposer sur l'installateur d'outils de Jenkins

# Et Jenkins dans tout ça ?

- Un build a aussi des dépendences (ex outils de build)
- Des solutions existent pour les gérer:
  - On peut préconfigurer tous les agents de build avec les outils
  - On peut se reposer sur l'installateur d'outils de Jenkins
- Ces solutions ne 'scalent' pas forcément



# Le Docker Pipeline Plugin

- Pour vos agents linux...

# Le Docker Pipeline Plugin

- Pour vos agents linux...
- Donne accès aux principales commandes nécessaires pour utiliser Docker avec Jenkins

# Le Docker Pipeline Plugin

- Pour vos agents linux...
- Donne accès aux principales commandes nécessaires pour utiliser Docker avec Jenkins
- Intégré à Jenkins
  - Docker fingerprints

# Le Docker Pipeline Plugin

- Pour vos agents linux...
- Donne accès aux principales commandes nécessaires pour utiliser Docker avec Jenkins
- Intégré à Jenkins
  - Docker fingerprints
  - Montage du workspace, passage des variables d'environnement...

# Un build simple

```
node {  
  git 'https://github.com/jglick/simple-maven-project-with-tests.git'  
  def mvnHome = tool 'M3'  
  sh "'${mvnHome}/bin/mvn' clean install"  
}
```

# Un build simple

```
node {  
  git 'https://github.com/jglick/simple-maven-project-with-tests.git'  
  def mvnHome = tool 'M3'  
  sh "'${mvnHome}/bin/mvn' clean install"  
}
```

```
node {  
  git 'https://github.com/jglick/simple-maven-project-with-tests.git'  
  docker.image('maven:3.3.9-jdk-8').inside {  
    sh 'mvn clean install'  
  }  
}
```

# Autre stratégie de build

```
FROM node:6.10.0
MAINTAINER Pierre Beitz <pibeitz@gmail.com>
COPY package.json
RUN npm install
COPY . .
RUN npm run build
```

# Autre stratégie de build

```
FROM node:6.10.0

MAINTAINER Pierre Beitz <pibeitz@gmail.com>

COPY package.json

RUN npm install

COPY . .

RUN npm run build
```

```
node {
  sh 'docker build .'
```



# Un changement de paradigme

- La charge du déploiement d'une nouvelle version / configuration dépend du dev

# Un changement de paradigme

- La charge du déploiement d'une nouvelle version / configuration dépend du dev
- L'ops livre une infrastructure totalement agnostique

# Un changement de paradigme

- La charge du déploiement d'une nouvelle version / configuration dépend du dev
- L'ops livre une infrastructure totalement agnostique
- Cahier des charges de l'agent de build: docker... et c'est tout!

# Quelques features

- Isolation

```
node {  
  sh 'curl --version'  
  // curl 7.47.0 (x86_64-pc-linux-gnu) libcurl/7.47.0  
  // GnuTLS/3.4.10 zlib/1.2.8 libidn/1.32 librtmp/2.3  
  docker.inside('maven:3.3.9-jdk-8') {  
    sh 'curl --version'  
    // curl 7.38.0 (x86_64-pc-linux-gnu) libcurl/7.38.0 OpenSSL/1.0.1  
    // zlib/1.2.8 libidn/1.29 libssh2/1.4.3 librtmp/2.3  
  }  
}
```

# Quelques features

- Build, test et déploiement d'une image:

```
stage('build') {  
    def myImage = docker.build 'my-image'  
}
```

# Quelques features

- Build, test et déploiement d'une image:

```
stage('build') {  
    def myImage = docker.build 'my-image'  
}
```

```
stage('test') {  
    myImage.withRun { c ->  
        // Run a test against the container c started from myImage  
    }  
}
```

# Quelques features

- Build, test et déploiement d'une image:

```
stage('build') {  
    def myImage = docker.build 'my-image'  
}
```

```
stage('test') {  
    myImage.withRun { c ->  
        // Run a test against the container c started from myImage  
    }  
}
```

```
stage('publish') {  
    docker.withRegistry('my-registry', 'my-credentials') {  
        myImage.push()  
    }  
}
```

# Au delà des pipelines, docker pour l'infra

## Jenkins

- Des images officielles: jenkinsci/jenkins (pour le master)



# Au delà des pipelines, docker pour l'infra Jenkins

- Des images officielles: jenkinsci/jenkins (pour le master)
- Docker Plugin pour provisionner dynamiquement des agents avec un docker host

# Au delà des pipelines, docker pour l'infra Jenkins

- Des images officielles: jenkinsci/jenkins (pour le master)
- Docker Plugin pour provisionner dynamiquement des agents avec un docker host
- Remarque: ne pas faire du Docker in Docker!

# Remarques

- Attention à la mémoire

# Remarques

- Attention à la mémoire
- Faire des scripts pour nettoyer les images/ containers

# Remarques

- Attention à la mémoire
- Faire des scripts pour nettoyer les images/ containers
- Utiliser une version récente de docker et de linux!

# Remarques

- Attention à la mémoire
- Faire des scripts pour nettoyer les images/ containers
- Utiliser une version récente de docker et de linux!
- Connaissance minimale de docker pour tous les dev qui utilisent un docker pipeline

**Demo!**

# Questions?