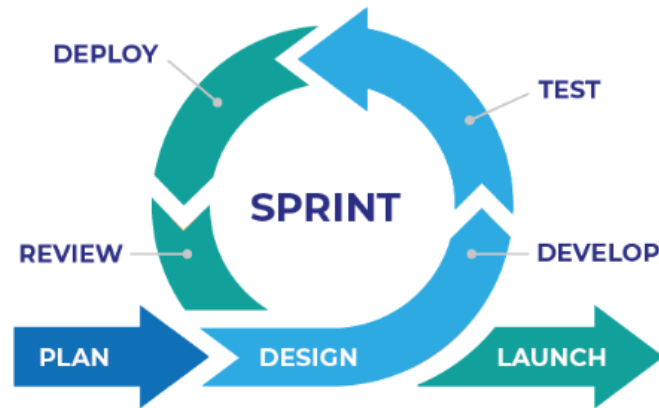


# Une introduction à Docker et aux conteneurs

Thibaud Martinez

[thibaud.martinez@dauphine.psl.eu](mailto:thibaud.martinez@dauphine.psl.eu)

# Évolutions du développement logiciel



**Méthodes agiles**



**Microservices**

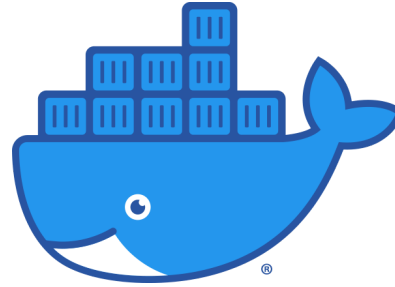


**Cloud computing**

## **Comment répondre aux nouveaux défis ?**

- Déployer plus rapidement des programmes hétérogènes.
- Assurer la portabilité des programmes.
- Gérer l'exécution et la communication de multiples applications.

# Docker



Docker permet d'**empaqueter** et d'**exécuter** une application dans un environnement isolé appelé "**conteneur**".

## Quels avantages ?

- Livraison rapide et cohérente des applications.
- Déploiement et mise à l'échelle réactifs.
- Exécution d'un grand nombre de charges de travail sur un même matériel.

# Fonctionnement de Docker

```
FROM ubuntu:18.04  
COPY . /app  
RUN make /app  
CMD python /app/app.py
```

Dockerfile



build



Image

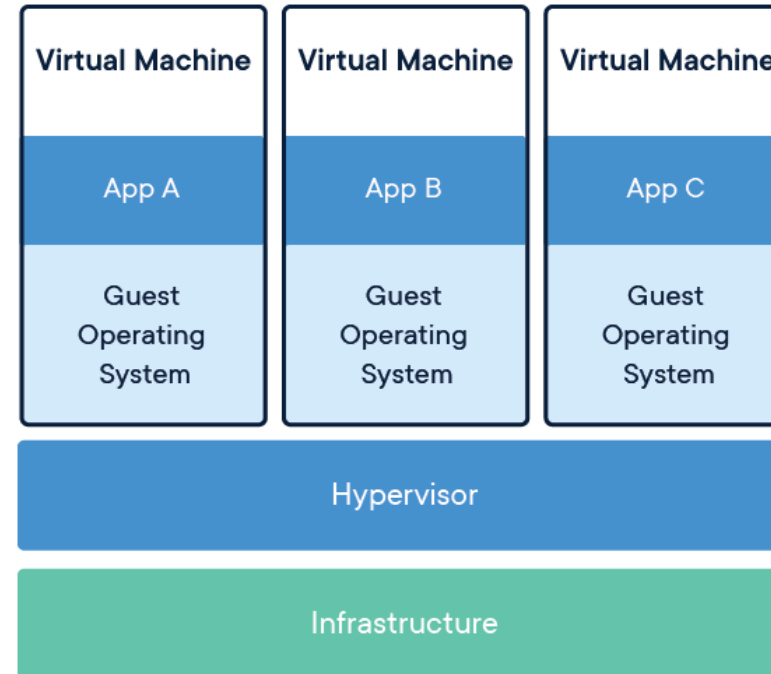
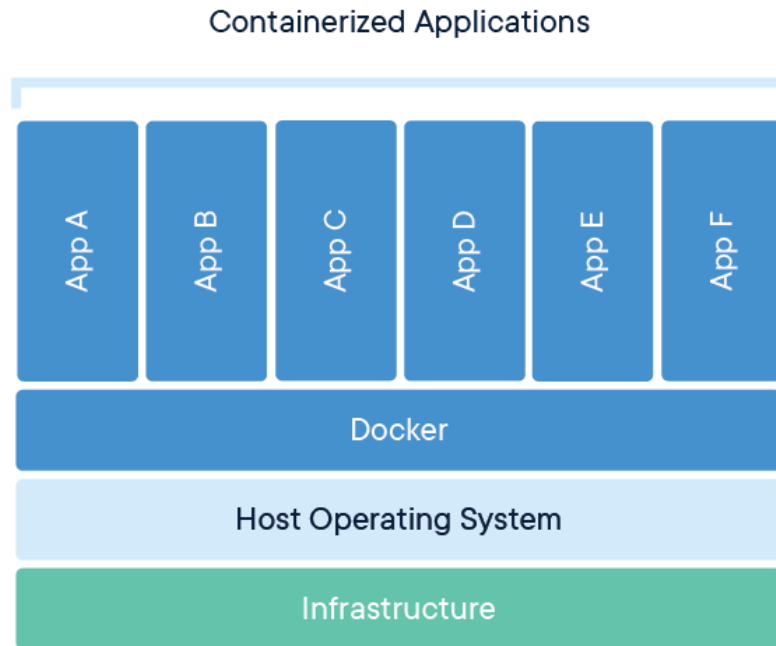


run



Conteneur

# Les conteneurs ne sont pas des machines virtuelles !



→ Plusieurs conteneurs peuvent fonctionner sur la même machine et **partager le noyau du système d'exploitation avec d'autres conteneurs**, chacun fonctionnant comme **un processus isolé dans l'espace utilisateur**.

# Docker s'appuie sur les mécanismes du noyau Linux

Un conteneur est un processus qui est isolé de tous les autres processus sur la machine hôte grâce, notamment, aux :

- ***namespaces*** : partitionne les ressources du noyau de telle sorte qu'un ensemble de processus voit un ensemble de ressources tandis qu'un autre ensemble de processus voit un autre ensemble de ressources.
- **cgroups (control groups)** : pour limiter, compter et isoler l'utilisation des ressources (processeur, mémoire, utilisation disque, etc.).

# Docker 101 tutorial

[Play with Docker](#) est une aire de jeu interactive qui vous permet d'exécuter des commandes Docker sur un terminal Linux, sans aucun téléchargement.

1. Connectez-vous à <https://labs.play-with-docker.com/> pour accéder à un terminal.
2. Tapez la commande suivante dans le terminal: `docker run -dp 80:80`  
`docker/getting-started:pwd`
3. Attendez que le conteneur démarre et cliquez sur le badge du port `80`.
4. À vous de jouer !

 Vous pouvez également utiliser [Docker Desktop](#) pour lancer le tutorial.



## Pour aller plus loin

- [Containers From Scratch](#) • Liz Rice • GOTO 2018
- [Demystifying Containers - Part I: Kernel Space](#)
- [Docker Documentation](#)