

La conteneurisation

La conteneurisation

- *Qu'est-ce que c'est ?*

La conteneurisation

- *Qu'est-ce que c'est ?*

Approche par la virtualisation

- Hyperviseur
- VM
- Système hôte / Système Invité
- Optimisation de ressources physiques
- Migration de VM sur un autre environnement

La conteneurisation

- *Qu'est-ce que c'est ?*

Approche par la virtualisation

- Hyperviseur
- VM
- Système hôte / Système Invité
- Optimisation de ressources physiques
- Migration de VM sur un autre environnement

Au niveau sécurité = séparation des systèmes dans un environnement

Au niveau programmation = isolation code dans un environnement

La conteneurisation

- *Qu'est-ce que c'est ?*

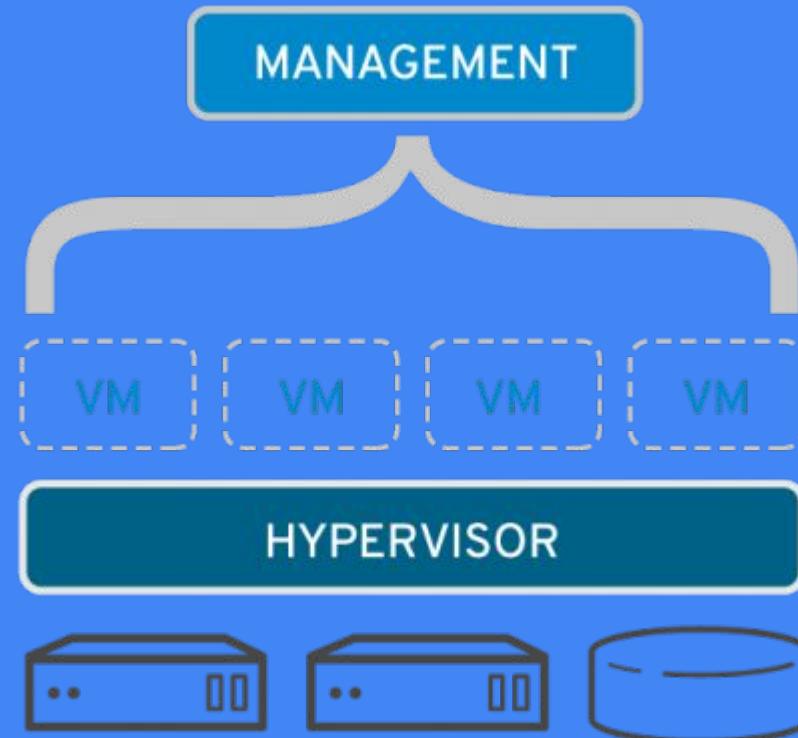
Approche par la virtualisation

LIMITES

- Dépendance de l'OS pour les applications
- Risque pour la machine hôte au niveau de l'execution d'applications test
- Impossibilité de déployer plusieurs applications répliquées
- Manque de cohérence des environnements

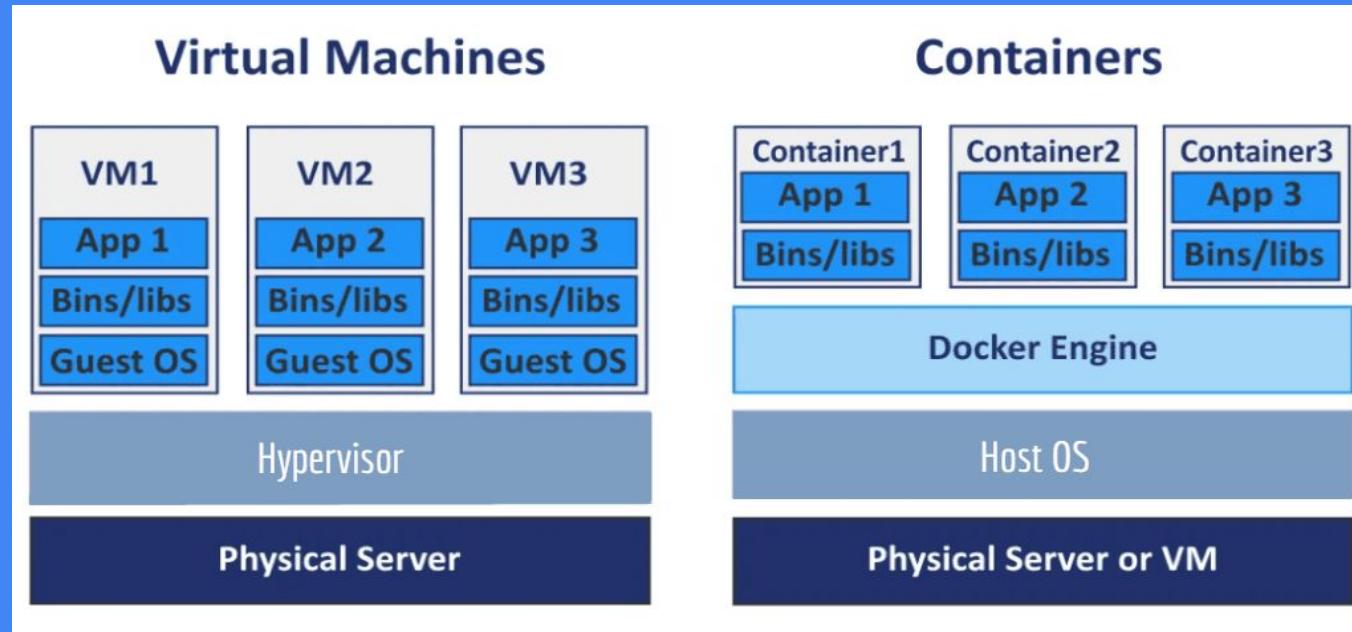
La conteneurisation

Approche par la virtualisation



La conteneurisation

Approche par la virtualisation



La conteneurisation

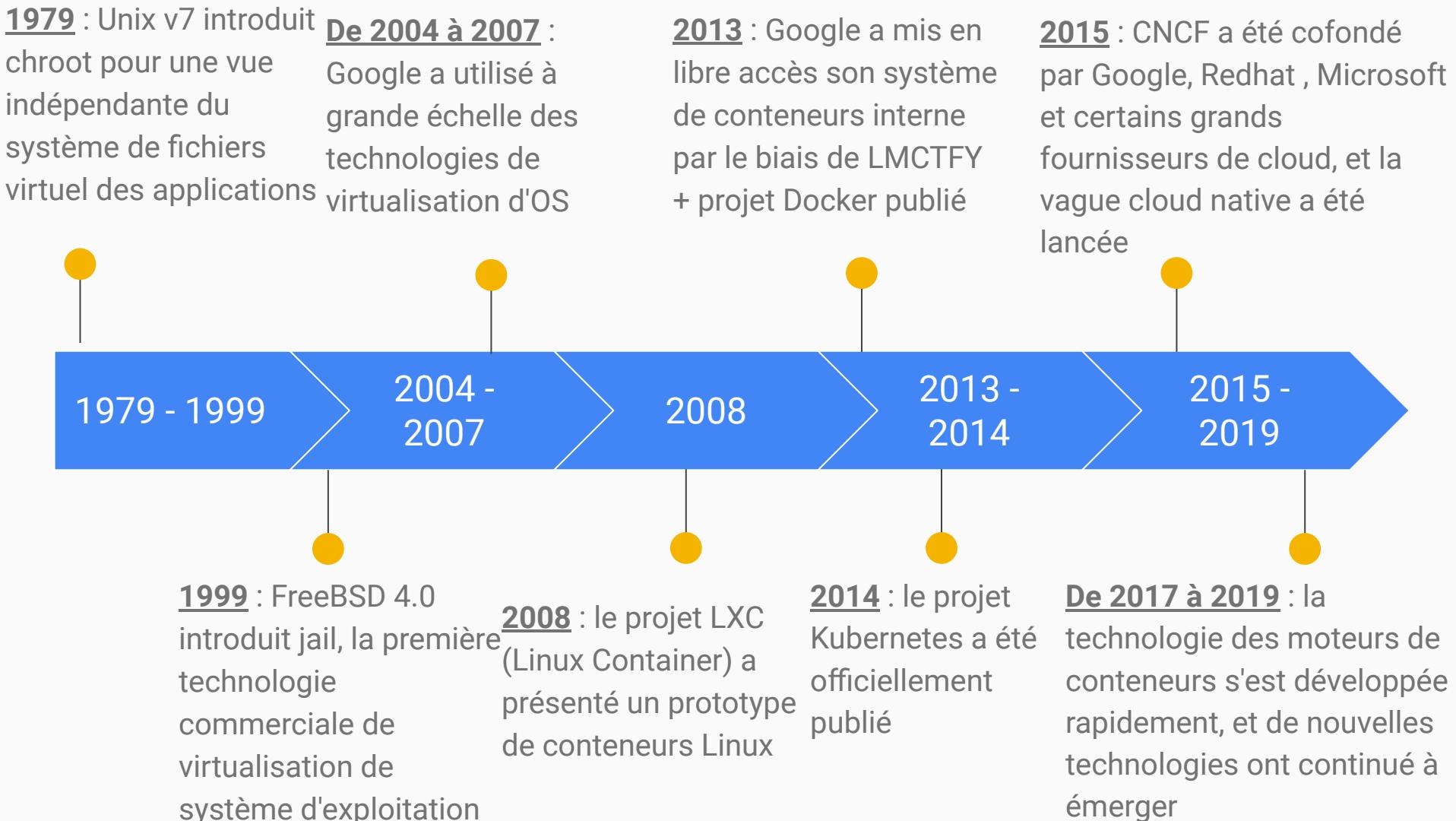
- *Qu'est-ce que c'est ?*

La **conteneurisation informatique** permet de packager tous les services, scripts, API, bibliothèques dont une application a besoin. L'objectif : en permettre l'exécution sur n'importe quel noyau compatible.

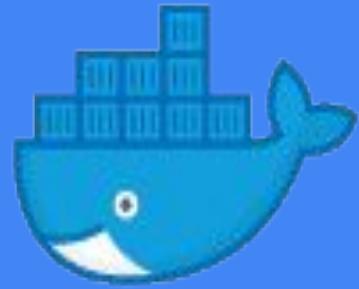
 Elle évite de se soucier d'interactions ou d'incompatibilités avec les conteneurs déjà présents ou à venir sur cette machine.

 Elle permet de ne pas occuper autant de ressources que réclamerait une **machine virtuelle** (ou *virtual machine*, VM), qui emporte son propre système d'exploitation et bloque des ressources à son lancement.

Historique :



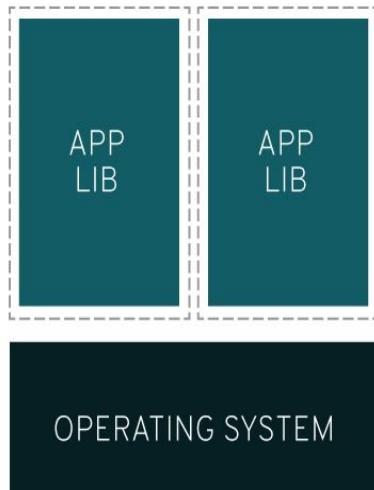
Docker



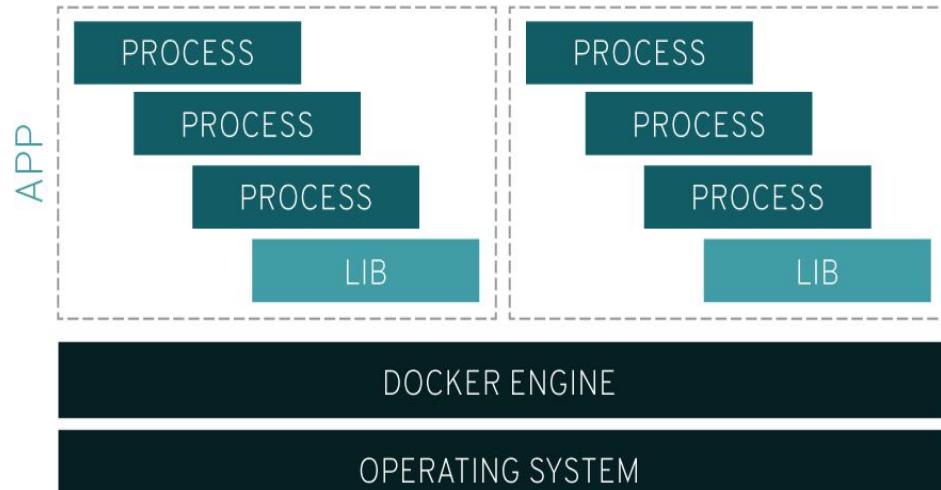
docker

Docker

Traditional Linux containers vs. Docker



LXC



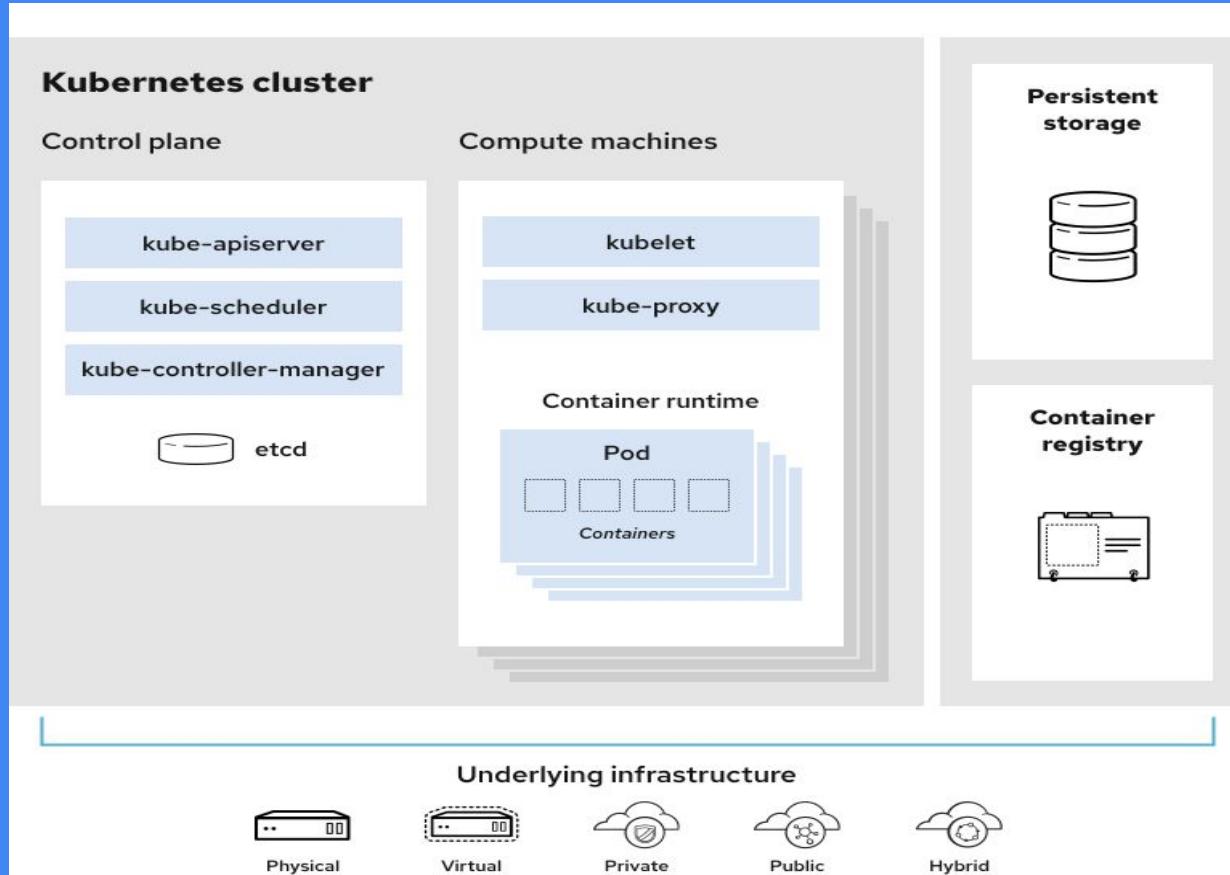
DOCKER

Kubernetes



kubernetes

Kubernetes



Avantages :

Inconvénients :

Avantages :

- L'accélération des développements
- Facilité d'utilisation pour les développeurs
- La portabilité et donc l'accélération des déploiements
- L'impact moindre sur les performances du serveur
- Fin de dépendance de programme pour les applications
- Gestion de la scalabilité avec **Kubernetes**
- Création facile d'image via Docker
- Mise à jour + ajout de fonctions sans risque
- Isolation d'applications pour la sécurité
- Facilité de gestion

Inconvénients :

- Lourd travail à mettre en place
- Changement d'OS difficile
- Une sécurité pas infaillible
- Orchestration des conteneurs limitée
- Risque de perte de données
- Supervision accrue nécessaire

Alternatives =

D'autres solutions facilitant la conteneurisation existent bien évidemment, sous Linux comme sous Windows, FreeBSD ou Solaris :

- Virtualisation
- LXC (la base historique de la conteneurisation sous Linux),
- Rocket (rkt) de CoreOs,
- Windows Hyper-V Containers (qui s'apparente à des VM légères),
- Oracle Solaris Zones
- FreeBSD

La virtualisation

Il est à noter que la conteneurisation n'exclut pas la **virtualisation** : les deux méthodes peuvent être mixées en fonction des besoins.

Une **machine virtuelle** peut accueillir des conteneurs si cela facilite la gestion et la sécurité de votre organisation par environnement ou domaine.

Outils

Docker est une surcouche qui rend le développement et le déploiement des conteneurs beaucoup plus simple tout en les standardisant. La 1ère version docker 1.0 date de 2014 mais son adoption par les plus grands compte de l'informatique en fait déjà un outil de référence

Kubernetes est un système open source permettant d'automatiser le déploiement, la mise à l'échelle et la gestion des applications containerisées

OpenShift est une plateforme de conteneurs d'applications open source, principalement basée sur Docker et orchestrée à l'aide de la gestion des clusters de conteneur Kubernetes qui est chargé du démarrage et du scaling (rajouter des charges sans pour autant nuire aux performances) des conteneurs

Il propose des outils de configuration plus poussés comme construire des images, intégrer un registre privé paramétrable pour séparer les images par projet, proposer une interface utilisable par un non administrateur, proposer des routeurs vers les applications, gérer les droits/restrictions/authentification etc..

Developpement / Evolution

- Années 60 : Problème systèmes dépendants
- 04/1982 : Chroot par Bill Joy
- 1990 : Jail par Bill Cheswick
- 04/2000 : Free BSD jail dans l'OS
- 2001 : Projet VServer par Jacques Gelinas
- 2004 : Zones de Solaris
- 2006 : Cpusets par Paul Menage
- 2007 : Cgroups
- 2008 : LXC manque de sécurité
- 2013 : Docker
- 2014 : Selinux + Kubernetes

Quelques exemples..