

INTRODUCTION À DOCKER

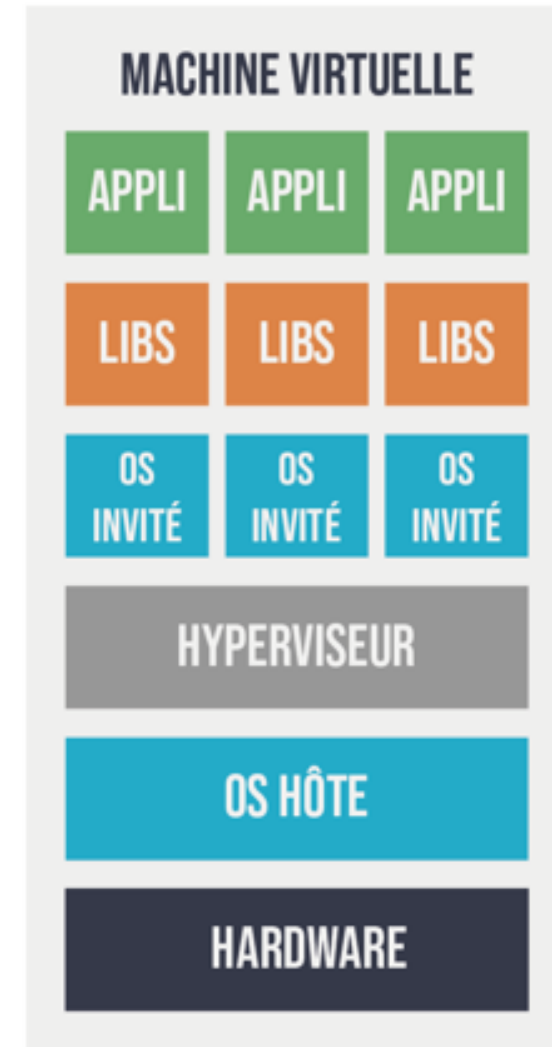
Docker - historique

Docker est une technologie de virtualisation légère qui a révolutionné le développement et le déploiement d'applications. Il a été créé en 2013 par Solomon Hykes et est devenu rapidement un outil incontournable pour les développeurs, les ingénieurs système et les opérations informatiques.

L'histoire de Docker remonte à l'idée de simplifier le processus de déploiement d'applications en utilisant des conteneurs. Avant Docker, la virtualisation était principalement réalisée à l'aide de machines virtuelles, ce qui avait tendance à être lourd et inefficace.

Virtualisation

Avant Docker, la virtualisation était la principale méthode de création d'environnements isolés pour les applications. La virtualisation utilisait des machines virtuelles (VM) pour exécuter des applications. Chaque VM comportait son propre système d'exploitation invité complet, avec son propre noyau. Cela rendait les VM lourdes et nécessitait des ressources importantes.



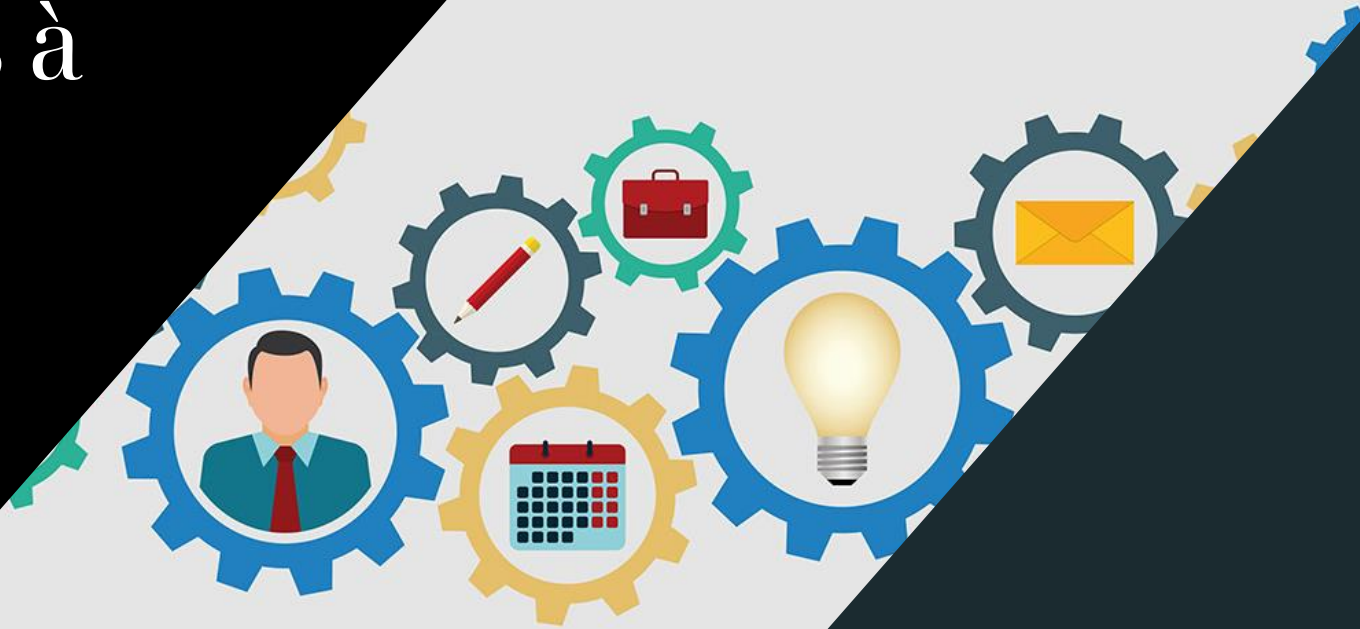
Les problèmes courants à régler

Problème de version d'un langage ou framework sur ma machine :

- Par exemple, j'ai PHP 8.1 sur ma machine, et je veux lancer un projet plus ancien



Les problèmes courants à régler



J'ai plusieurs projets, qui
dépendent de version de
langage différents...
Comment je gère ?

Les problèmes courants à régler



Problème de téléchargements de dépendance sur ma machine et qui manquent pendant le déploiement sur le serveur back (exemple lib python absente, oubliée, pas mises à jour au cours de montées de version pendant la vie de l'application, etc.)

Les problèmes courants à régler

Problème de travail en collaboration avec plusieurs personnes sur plusieurs OS différents (mac, windows, Linux), des versions différentes de langages, pas les mêmes dépendances dispos sur tous les postes, etc.

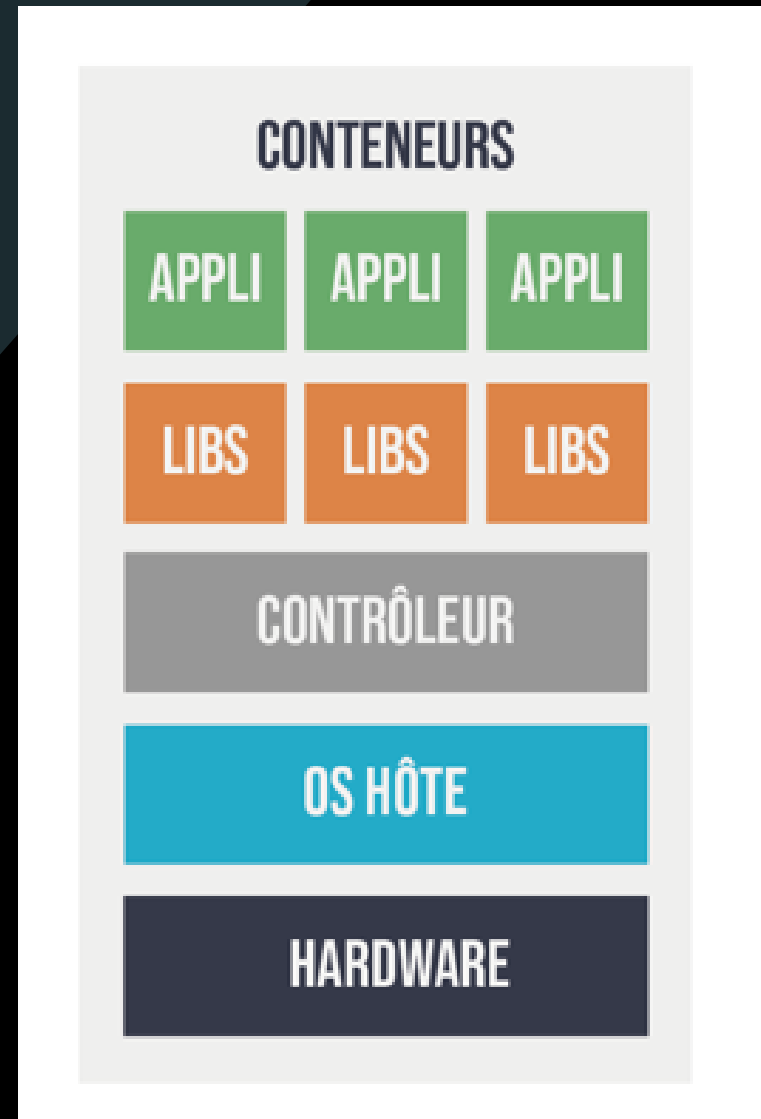


Docker - historique

Docker a introduit une approche plus légère en utilisant des conteneurs, qui sont des environnements d'exécution isolés pour les applications. Ces conteneurs partagent le même noyau du système d'exploitation de l'hôte, mais ont leurs propres systèmes de fichiers et processus. Cette approche a permis de créer des environnements plus portables, reproductibles et faciles à gérer.

Conteneurisation

Docker, en revanche, utilise des conteneurs. Les conteneurs partagent le même noyau de système d'exploitation de l'hôte, ce qui les rend beaucoup plus légers en termes de ressources. Les conteneurs sont isolés les uns des autres, mais ils partagent la même infrastructure de base. Cela signifie que vous pouvez exécuter de nombreuses applications dans des conteneurs sur un même hôte, en utilisant beaucoup moins de ressources que les machines virtuelles.



Les conteneurs

- **Isolation:** Chaque conteneur est isolé des autres, ce qui signifie qu'il ne peut pas accéder aux ressources d'un autre conteneur sans autorisation.
 - **Léger:** Les conteneurs partagent le même noyau de système d'exploitation de l'hôte, ce qui les rend légers en termes de ressources.
 - **Portabilité:** Les conteneurs sont portables, ce qui signifie que vous pouvez les exécuter de manière cohérente sur n'importe quel hôte Docker, que ce soit sur un ordinateur de développement, un serveur de production ou un service cloud.
 - **Reproductibilité:** Les conteneurs sont définis par des fichiers Docker appelés Dockerfiles, ce qui permet de reproduire facilement l'environnement de l'application sur n'importe quelle machine.
-

Virtualisation vs Conteneurisation

