

# Cours : Présentation de Docker

Avec la virtualisation complète, la création d'une VM nécessite l'installation complète d'un système d'exploitation qui intègre un grand nombre de couche logicielles afin de permettre l'exécution d'une diversité de services ou d'applications. Cependant l'application ou le service qui sera configuré ne nécessite qu'une partie des fonctionnalités de l'OS. Par exemple la mise en place d'un serveur DNS ne nécessite pas la présence de composants multimédia ou de programmation comme python. Les pratiques courantes consistent également à créer de nouvelles VM par duplication d'une image contenant un OS personnalisé et à assurer une isolation forte entre les VMs. Dans ce cas de figure on virtualise des systèmes d'exploitation identiques sur un même hôte physique. Mais cela représente une consommation importante de ressources disque et de calcul pour implémenter l'application ou le service.

La technologie de conteneur mise en œuvre par Docker est une réponse à ces problématiques :

- Permettre la **création** de service s'exécutant sur un **système d'exploitation identique**,
- Partager une grande partie des **fonctionnalités** et des bibliothèques de base du système d'exploitation,
- Assurer une **isolation** forte entre les VMs.
- **Limiter** la consommation de ressources : espace disque, mémoire, temps de mise en œuvre très réduit , administration et maintenance lors des mises à jour notamment.

Docker permet des usages qu'il était difficilement envisageable auparavant. A titre d'exemple, un développeur web pourra très rapidement mettre en œuvre un environnement vierge pour réaliser ses tests puis supprimer cet environnement une fois les tests terminés afin de libérer des ressources. De plus il pourra tester son application dans des environnements différents. S'il doit faire cela plusieurs fois par jours, la solution Docker est particulièrement adaptée. C'est le principe des **micro-services** en facilitant la création de conteneurs Docker pour chaque service ou applications à provisionner selon les besoins.

Docker n'est ni un émulateur ni une solution de virtualisation. C'est un **logiciel client-serveur** basé sur des fonctionnalités de bas niveau du noyau Linux et ou de Windows. Si la virtualisation classique consiste à simuler le fonctionnement d'une machine physique en isolant les applications dans des systèmes d'exploitation situés dans des VMs différentes. L'approche de Docker consiste à assurer cette isolation entre application avec les fonctionnalités du système d'exploitation lui-même. La machine contenant le système d'exploitation est centrale dans cette architecture et l'isolation est moins stricte. Docker est ainsi associé à de la virtualisation légère et de ce fait est perçu comme fournissant moins de garantie d'isolation que la virtualisation classique. Docker est probablement à considérer comme **complémentaire** à la virtualisation classique.

Docker se compose :

- d'un client,
- d'une API,
- et d'un serveur sous forme de démon.

Docker dispose d'une fonctionnalité de dépôt central qui **Docker Hub**. Cette fonctionnalité est essentielle et consiste en un dépôt public, accessible avec une simple connexion Internet, qui permet de favoriser la réutilisation des images déjà préparées par des éditeurs de logiciels, des entreprises ou des particuliers. Ces contributeurs constituent la communauté qui s'est développée autour de la technologie Docker et c'est une des raisons principales de sa rapide adoption.

Site de Docker hub : <https://hub.docker.com/>

Sur Docker Hub :

- les images communautaires sont identifiables par leur nom préfixé par l'identifiant du propriétaire de l'image et séparé par le symbole /.
- les images officielles ne sont pas préfixées.

Vous pouvez également faire partie de cette communauté et partager vos propres images Docker en créant au préalable **votre compte** sur Docker Hub.

## INFORMATION

Pour prendre connaissance des images officielles maintenues sur le dépôt GitHub nommé docker-library/official-images, accédez au site <https://github.com/docker-library/official-images> au répertoire library.

Docker c'est aussi :

- **Docker Compose** pour automatiser la mise en œuvre de plusieurs conteneurs en lien les uns avec les autres.
- **Docker Machine** pour provisionner des machines de déploiement de manière unifiée, et ce quel que soit le support utilisé, local ou dans un cloud. Il permet de réserver rapidement des machines pour déployer des conteneurs sur Amazon, Azure ou d'autres prestataires de Cloud.
- **Docker Swarm** pour administrer un cluster de machines Docker, le faisant apparaître comme une seule machine prête à recevoir de multiples conteneurs.

From:

/ - **Les cours du BTS SIO**

Permanent link:

</doku.php/docker/presentation?rev=1567455470>

Last update: **2019/09/02 22:17**

