



## Référentiel de compétences

Semestre 1



Module



Semestre 2



Artéfact



Session de formation



Business plan

Lecture individuelle



Systèmes d'information &amp; management des organisations

**Docker**

Méthodologie en cascade

La veille informatique

Semestre 2 &gt; Artéfact &gt; Lecture individuelle &gt; Docker

# Docker

## Historique, but & définition

La conteneurisation est une technologie informatique qui permet de créer et de déployer des applications dans des environnements isolés et portables, appelés conteneurs. Docker est l'un des systèmes de conteneurisation les plus populaires.

L'histoire de la conteneurisation remonte à la fin des années 1970, avec le développement des "chroot jails" dans les systèmes Unix. Les chroot jails permettaient d'isoler des processus dans un environnement limité, en leur donnant un accès restreint au système de fichiers et aux ressources de l'ordinateur.

Plus tard, au début des années 2000, la technologie des conteneurs a été développée pour permettre l'exécution de plusieurs applications sur un même système d'exploitation, tout en isolant les processus les uns des autres. Cependant, ces premières implémentations étaient souvent complexes et difficiles à utiliser.



En 2013, Docker a été lancé en tant que projet open source, et a rapidement gagné en popularité pour sa facilité d'utilisation et sa capacité à créer des conteneurs portables et reproductibles. Depuis lors, la conteneurisation et Docker en particulier sont devenus des technologies clés dans le déploiement et la gestion d'applications modernes.

Le but de la conteneurisation est de fournir un environnement isolé et portable pour les applications, ce qui permet de les déployer rapidement et facilement sur différents systèmes. Les conteneurs sont conçus pour être légers et rapides, et peuvent être déployés sur des serveurs locaux, des centres de données cloud ou des clusters de serveurs.

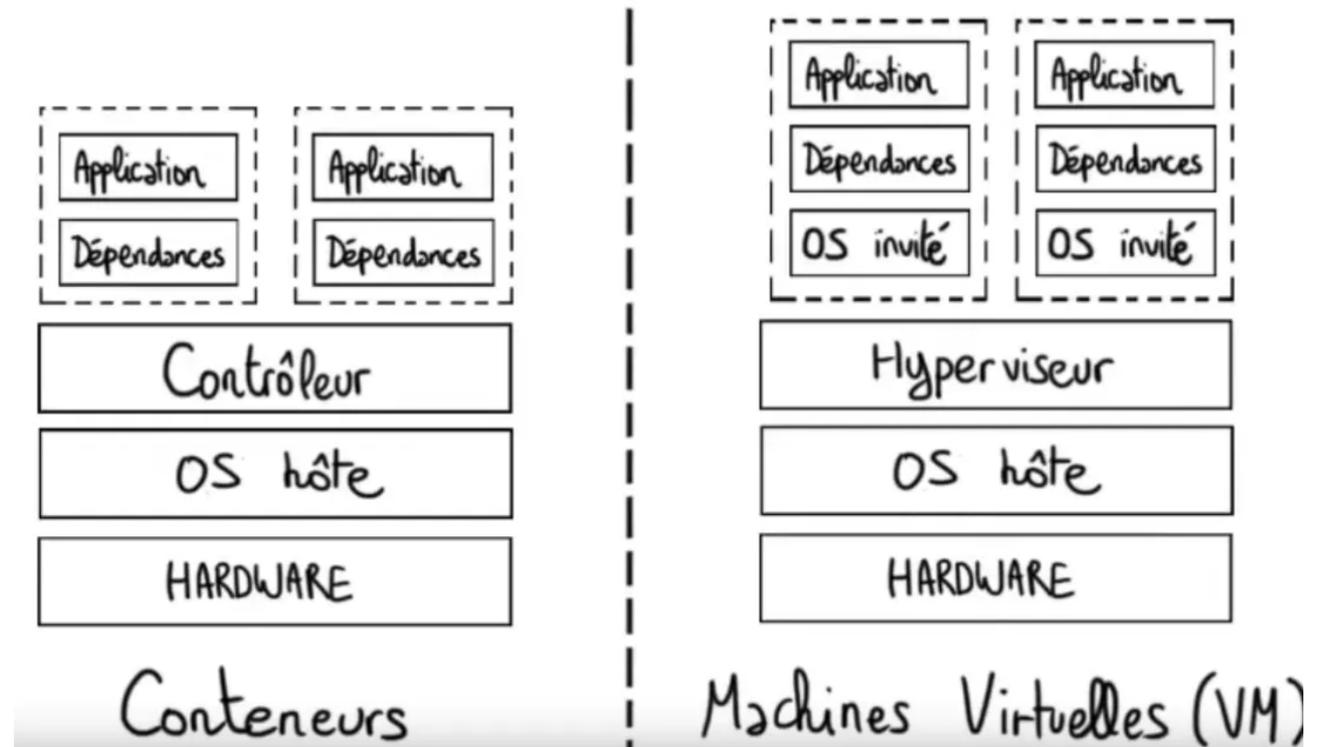
## Comparaison Virtualisation légère vs lourde

---

La virtualisation est une technologie qui permet de créer des environnements informatiques virtuels, appelés machines virtuelles, en utilisant des ressources matérielles d'un ordinateur hôte. Il existe deux types de virtualisation : la virtualisation légère et la virtualisation lourde.

La virtualisation légère, également appelée virtualisation basée sur le système d'exploitation (OS-level virtualization), utilise un seul système d'exploitation pour créer plusieurs environnements isolés, appelés conteneurs. Les conteneurs partagent le même noyau du système d'exploitation de l'hôte, mais ont leur propre espace de noms, système de fichiers, processus et ressources. Les conteneurs sont légers, car ils ne nécessitent pas de système d'exploitation complet pour chaque environnement, et sont donc plus efficaces en termes d'utilisation des ressources.

La virtualisation lourde, également appelée virtualisation basée sur la machine (machine-level virtualization), utilise un hyperviseur pour créer plusieurs machines virtuelles, chacune ayant son propre système d'exploitation complet. Les machines virtuelles sont plus lourdes que les conteneurs, car chaque machine virtuelle nécessite son propre système d'exploitation, pilotes de périphériques, et une quantité significative de mémoire et de puissance de traitement.



Voici une comparaison entre la virtualisation légère et la virtualisation lourde :

1. Isolation: Les deux technologies offrent une isolation, mais la virtualisation lourde fournit une isolation plus complète car chaque machine virtuelle a son propre système d'exploitation et est donc complètement isolée des autres machines virtuelles et de l'hôte. La virtualisation légère utilise

un noyau partagé, ce qui offre une isolation plus limitée, mais cela permet une utilisation plus efficace des ressources.

2. Utilisation des ressources : La virtualisation légère est plus efficace en termes d'utilisation des ressources car elle utilise un noyau partagé et n'a pas besoin de répliquer complètement le système d'exploitation pour chaque environnement. La virtualisation lourde utilise des machines virtuelles complètes, qui nécessitent plus de ressources et sont donc moins efficaces.
3. Portabilité : Les conteneurs sont plus portables que les machines virtuelles, car ils sont conçus pour être déployés sur différentes plates-formes avec peu ou pas de modifications. Les machines virtuelles peuvent être plus difficiles à déployer sur différentes plates-formes, car elles nécessitent souvent des ajustements pour s'adapter aux différentes configurations de matériel et de logiciel.

Avantages de la virtualisation légère :

1. Légers : Les conteneurs sont beaucoup plus légers que les machines virtuelles, car ils ne nécessitent pas d'hyperviseur ni de système d'exploitation invité. Ils partagent plutôt le noyau et les bibliothèques de l'hôte, ce qui les rend plus rapides à démarrer et à exécuter.
2. Efficaces : Étant donné que les conteneurs partagent les ressources du système hôte, ils sont plus efficaces en termes d'utilisation des ressources que les machines virtuelles. Plusieurs conteneurs peuvent être exécutés sur un seul hôte, ce qui permet d'optimiser l'utilisation des ressources.
3. Portables : Les conteneurs sont portables, ce qui signifie qu'ils peuvent être facilement déplacés entre différents environnements, tels que des ordinateurs portables, des serveurs locaux et des environnements cloud. Cela facilite le développement, les tests et le déploiement d'applications.
4. Isolés : Les conteneurs offrent un haut niveau d'isolation entre les applications, ce qui les rend plus sûrs et plus fiables. Chaque conteneur a son propre système de fichiers et son propre espace de

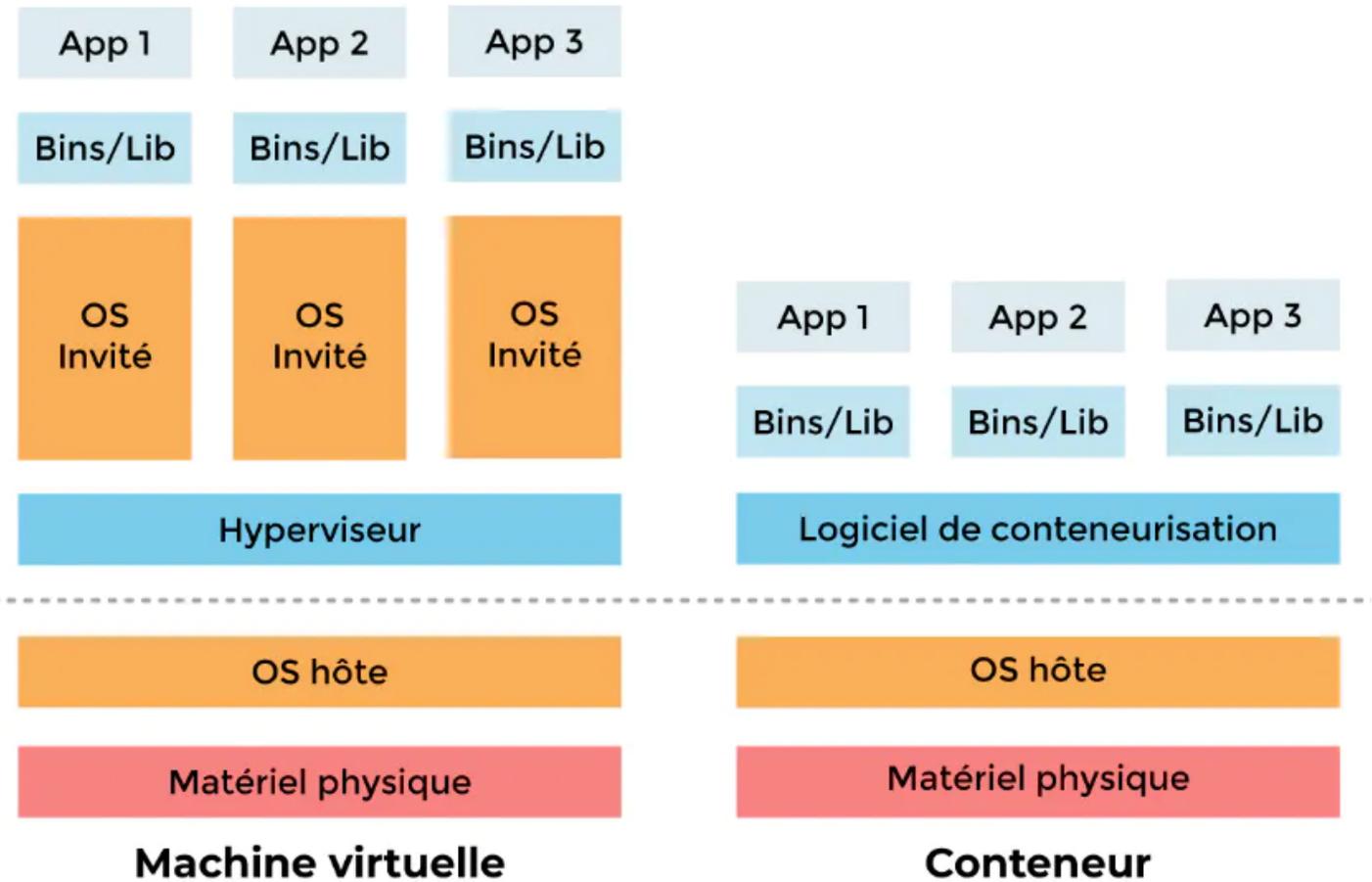
noms réseau, ce qui évite les conflits et les collisions entre les applications.

5. Flexibles : Les conteneurs sont flexibles et évolutifs, ce qui signifie qu'ils peuvent être facilement configurés pour répondre aux besoins de différentes applications. Les développeurs peuvent facilement ajouter des packages logiciels, des dépendances et des bibliothèques à un conteneur pour répondre aux besoins de leur application.
6. Rapides : Les conteneurs sont plus rapides à déployer que les machines virtuelles, car ils n'ont pas besoin d'un système d'exploitation complet et d'un hyperviseur pour être exécutés. Cela signifie que les applications peuvent être déployées rapidement et facilement, ce qui permet aux développeurs de se concentrer sur le développement de l'application plutôt que sur l'infrastructure.

## Docker les *bases*

---

1. Containers Un conteneur est un processus qui regroupe une application et ses dépendances, isolée du reste du système d'exploitation et des autres conteneurs fonctionnant sur le même système. Un conteneur est une instance d'une image de conteneur qui peut être déployée et exécutée sur n'importe quel système prenant en charge les conteneurs.



Les conteneurs sont une forme de virtualisation légère qui utilise une technologie appelée "virtualisation basée sur le système d'exploitation". Contrairement aux machines virtuelles traditionnelles, les conteneurs partagent le même noyau du système d'exploitation de l'hôte, ce qui permet une utilisation plus efficace des ressources système.

Chaque conteneur est isolé et dispose de sa propre configuration de réseau, de son propre espace de noms, de son propre système de fichiers et de son propre ensemble de processus. Les

conteneurs permettent aux développeurs de créer des environnements de développement et de déploiement cohérents pour leurs applications, ce qui facilite le déploiement sur différents systèmes et permet d'éviter les conflits entre les différentes applications.

Les conteneurs sont souvent utilisés pour créer des environnements de développement, de test et de production pour les applications. Ils sont également couramment utilisés dans les déploiements de microservices, où chaque microservice est déployé dans son propre conteneur. Les technologies de conteneurisation populaires incluent Docker, Kubernetes, et Podman.

## 2. Images

Une image de conteneur est un fichier binaire qui contient tout ce qui est nécessaire pour exécuter une application dans un conteneur. Une image est un modèle préconfiguré de l'environnement d'exécution de l'application, qui comprend les dépendances logicielles, les bibliothèques et les fichiers de configuration nécessaires pour exécuter l'application.

Une image est créée en utilisant un fichier de configuration appelé Dockerfile, qui spécifie les instructions nécessaires pour créer l'image. Le Dockerfile peut inclure des instructions pour installer des packages, des dépendances, des fichiers de configuration, ainsi que pour copier les fichiers de l'application dans l'image.

Les images sont stockées dans un registre d'images, tel que le Docker Hub, qui est un service de registre d'images public où les utilisateurs peuvent partager leurs images. Les images peuvent également être stockées localement sur l'ordinateur de l'utilisateur, dans un registre privé ou dans un service de cloud computing.

Les images de conteneur sont utilisées pour créer des instances de conteneur, qui sont des copies exécutables de l'image. Lorsqu'un conteneur est créé à partir d'une image, il utilise les fichiers de l'image comme base pour créer un système de fichiers en lecture seule. Le conteneur ajoute ensuite une couche de système de fichiers en écriture qui peut être utilisée pour stocker les données générées par l'application.

Les images de conteneur sont un moyen efficace de créer des environnements cohérents pour les applications, en garantissant que toutes les dépendances et les bibliothèques nécessaires sont incluses dans l'image. Les images peuvent être facilement partagées et déployées sur différents systèmes, ce qui simplifie le processus de développement et de déploiement des applications.

### 3. Docker Hub

Le Docker Hub est un service de registre d'images public pour Docker, un système de conteneurisation populaire. Il s'agit d'un service cloud qui permet aux utilisateurs de stocker, partager et télécharger des images de conteneurs pré-construites pour une utilisation ultérieure.

Le Docker Hub permet aux développeurs de télécharger des images de conteneurs pour des applications populaires telles que MySQL, Redis, Node.js, etc. Les images peuvent être téléchargées gratuitement ou moyennant des frais pour les images privées. Les utilisateurs peuvent également publier leurs propres images sur le Docker Hub pour permettre aux autres développeurs de les télécharger et de les utiliser.

Le Docker Hub est utilisé par les développeurs pour stocker et partager des images de conteneurs pour des applications, des bibliothèques, des systèmes d'exploitation et des frameworks. Les images stockées sur le Docker Hub peuvent être facilement téléchargées et utilisées pour créer des

conteneurs. Les images peuvent également être utilisées pour créer des déploiements de conteneurs en utilisant des outils de gestion de conteneurs tels que Kubernetes.

Le Docker Hub offre également des fonctionnalités de sécurité telles que la vérification de sécurité automatisée des images téléchargées pour détecter les vulnérabilités et les erreurs de configuration. Il offre également des fonctionnalités de gestion des utilisateurs pour contrôler l'accès aux images privées.

## Pour aller plus loin

---

1. Docker dans la pratique (image, container, hub => CLI Commands)
2. Création d'un Dockerfile
3. Optimisation des layers (Multi stage)
4. Création de plusieurs containers avec Docker Compose (DB + WEB Server + Websocket Server etc...)
5. Création d'un réseau de containers
6. Quelles sont les bonnes pratiques en équipe de développeurs avec Docker (Environnement dev, test, prod | Makefile)
7. Mettre en place un orchestrateur de conteneur Kubernetes (K8's)
8. Créer une pipeline CI/CD avec Github action > Docker Hub > AWS

## Références & sources

---

- [Docker](#)
- [Docker API Reference](#)
- [OpenClassrooms](#)
- [Educative](#)

Last updated on June 18, 2023

---

< **Systemes d'information &  
management des organisations**

**Méthodologie en cascade** >

Portefolio Dasek Joiakim