

## **PRESENTATION**

La technologie du conteneur est de plus en plus utilisée de nos jours. Cette technologie a été développée sur une base Linux. En effet un conteneur est une enveloppe virtuelle dans laquelle on installe tous les éléments dont une application aura besoin pour fonctionner : fichiers source, runtime, bibliothèques, outils et fichiers. C'est dans ce registre que s'inscrit Docker.

### **Définition de Docker**

La technologie de Docker a été développée par SOLOMON HYKES EN 2013.

Un docker est défini comme le gestionnaire de conteneurs pour les systèmes d'exploitation basés sur UNIX et sur Windows depuis 2016. Il peut être défini comme un projet Open Source proposant une surcouche qui automatise et simplifie le déploiement d'applications dans des conteneurs virtuels.

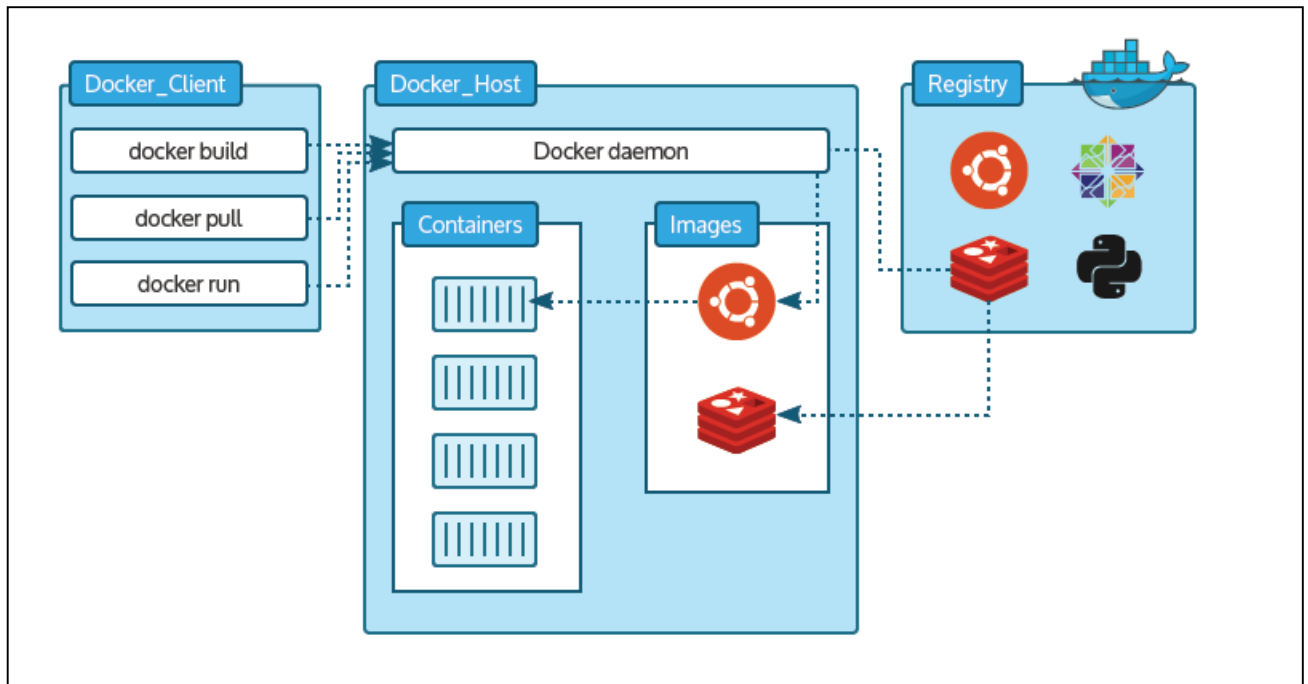
Son principal rôle est de permettre la création de nombreux environnements isolés sur un seul noyau. Il peut être défini comme une plate-forme ouverte pour construire, expédier et exécuter des applications distribuées. Les organisations approchent la technologie docker pour simplifier et accélérer leur processus de développement et de déploiement d'applications.

### **Le Fonctionnement de Docker**

La technologie Docker utilise le noyau Linux et des fonctions de ce noyau, telles que les groupes de contrôle cgroups et les espaces de noms, pour séparer les processus afin qu'ils puissent s'exécuter de façon indépendante.

A l'origine basée sur le format de Conteneurs Linux LXC, la société Docker a élargi le projet en proposant notamment une API qui permet d'exécuter des conteurs standards et portables d'un serveur Linux à l'autre.

### **Les composants majeurs d'une installation Docker.**



- ❖ Au centre, Le **Docker Host** représente la machine physique ou la machine virtuelle dans laquelle Docker Daemon et les containers sont déployés
- ❖ Le **Docker Daemon** est à la fois responsable de la création, du démarrage et du monitoring des containers, mais aussi de la construction et du stockage des images. Le système d'exploitation a la charge de démarrer le **Docker Daemon**.
- ❖ Le **Docker Client** se trouve à gauche de l'image. Il communique avec le Docker Daemon via sockets par une API RESTful. L'objectif du **Docker Client** est de contrôler l'hôte, créer des images, publier, exécuter et gérer les containers correspondants à l'instanciation de ces images. La communication via HTTP facilite le contrôle des connexions des Docker Daemons.
- ❖ Le **Docker Registry** à droite de l'image, permet de stocker et distribuer des images. Par défaut, le Registry est le Docker Hub, qui détient des milliers d'images publiques. les containers Docker sont créés en utilisant ces images de base. Une image Docker peut soit, inclure simplement les fondamentaux du système d'exploitation, soit consister à une pile d'applications pré-construites et prêtes à être lancées. Pour créer une image, l'option la plus courante est l'écriture d'un fichier de scénarios incluant des commandes diverses et nommé Docker File et de l'exécuter ensuite. Beaucoup d'entreprises exécutent leur propre Registry qui peut être utilisé pour stocker des images privées. Le Docker Daemon téléchargera ensuite des images du Registry pour répondre aux différentes demandes.

### Les Avantages de Docker

Contrairement à la virtualisation de serveurs et à une machine virtuelle, le conteneur n'intègre pas de système d'exploitation, il s'appuie directement sur le système d'exploitation du serveur sur lequel il est déployé. Cette technologie de conteneur présente plusieurs avantages :

- ❖ **Faible utilisation de ressources** : Les containers partagent les mêmes ressources avec le système d'exploitation sur lequel il est installé, ce qui permet de les rendre plus efficaces. Les containers permettent de diminuer le temps de démarrage, diminuer l'espace de stockage et la consommation du processeur.
- ❖ **Déploiement des applications plus facile et rapide** : Les containers disposent d'un environnement d'exécution minimal pour les applications, réduisant leur taille, on a donc un conteneur léger, ce qui rend le déploiement plus rapide.
- ❖ **Portabilité des applications** : Le conteneur étant léger facilite la portabilité. On évite ainsi de télécharger plusieurs librairies pour le faire fonctionner. Ceci permet d'éliminer un nombre important de bugs provoqués par les changements d'environnement et de systèmes.
- ❖ **Restauration** : La fonction la plus intéressante de la superposition de couches est sans doute la restauration. Chaque image est composée de couches. Aussi, si l'itération actuelle d'une image ne vous convient pas, vous pouvez restaurer la version précédente. Cette fonction favorise le développement agile et vous aide à mettre en œuvre les pratiques d'intégration et de distribution continues (CI/CD) au niveau des outils.

Les conteneurs offrent de nouvelles perspectives aux développeurs ainsi qu'aux opérateurs de Datacenter. Mais cette technologie présente aussi des inconvénients. Les inconvénients de Docker sont :

- ❖ **Environnement** : Docker fonctionne seulement sur Linux, pour l'utiliser sur Windows il faut lancer une micro machine virtuelle appelé Boot2Docker qui aura toute la technologie pour faire fonctionner Docker.
- ❖ **Expertise** : La mise en œuvre de conteneurs nécessite de l'expertise et une base de connaissance solide des ingénieurs. En effet cette technologie est différente de la virtualisation

- ❖ **Sécurité** : est le premier inconvénient des conteneurs. Mais néanmoins il existe beaucoup de manières de sécuriser les conteneurs. Par exemple il est possible de monter un système de fichiers en lecture seule, de forcer le conteneur à écrire uniquement sur des systèmes de fichiers de conteneurs spécifiques, et de créer un espace de noms réseau qui ne se connecte uniquement avec un intranet privé spécifique, et ainsi de suite. Mais, rien de tout cela n'est disponible par défaut. Il faut investir beaucoup de temps pour sécuriser les conteneurs. La règle est de traiter les conteneurs exactement comme n'importe quelle application serveur.
- ❖ **Problème de dépendances** : Les machines virtuelles sont autonomes car chacune exécute un système d'exploitation propre à elle avec ses propres pilotes et des composants d'application. Elles peuvent également migrer vers n'importe quel autre système du moment qu'un hyperviseur approprié est disponible alors que les conteneurs s'exécutent sur le système d'exploitation de l'hôte et partagent la majeure partie du noyau ainsi qu'un grand nombre de fichiers binaires et bibliothèques.

## **CONCLUSION**

La technologie Docker est particulièrement utile si le besoin est de faire tourner un maximum d'applications sur un minimum de serveurs mais il faut garder à l'esprit que la sécurité n'est pour l'instant pas suffisante.

En revanche, si le besoin est d'exécuter plusieurs applications sur des serveurs et/ou d'avoir une grande variété de systèmes d'exploitation, mieux vaut s'orienter vers les machines virtuelles.