Introduction aux conteneurs Docker

Docker est un outil de plateforme ouverte qui facilite la création, le déploiement et l'exécution des applications en utilisant des conteneurs. Les conteneurs Docker nous permettent de séparer les applications de l'infrastructure afin de déployer les applications/logiciels plus rapidement. Docker a des composants principaux qui incluent Docker Swarm, Docker Compose, Docker Images, Docker Daemon, Docker Engine. Nous pouvons gérer notre infrastructure de la même manière que nous gérons nos applications.

Qu'est-ce que Docker ?

Docker est un logiciel libre qui permet d'expédier, tester et déployer du code plus rapidement en utilisant le même noyau Linux. Il est différent d'une machine virtuelle car il crée une nouvelle machine virtuelle complète. Pour exécuter des conteneurs à grande échelle, il est nécessaire d'utiliser une plateforme d'orchestration de conteneurs et de planification comme Docker Swarm, Apache Mesos ou AWS ECS.

Quels sont les composants du conteneur Docker ?

- Le cœur de Docker se compose des éléments suivants:
- le moteur Docker
- Conteneurs Docker
- Images Docker
- Client Docker
- Démon Docker

Le moteur Docker crée et exécute les conteneurs Docker, qui sont des instances en cours d'exécution d'une image Docker. Le moteur Docker est constitué d'un serveur, d'une API REST et d'un client d'interface de ligne de commande. Le serveur est un service continu appelé processus démon, qui crée et gère les images Docker, les conteneurs, les réseaux et les volumes. Le client de l'interface de ligne de commande utilise l'API REST de Docker pour interagir avec le démon Docker à l'aide de commandes CLI.

Pourquoi avons-nous besoin du démon Docker ?

Le processus Docker Daemon est utilisé pour contrôler et gérer les conteneurs. Le démon Docker n'écoute que les demandes de l'API Docker et gère les images Docker, les conteneurs, les réseaux et les volumes. Il communique également avec d'autres démons pour gérer les services Docker.

Utilisation du client Docker

Le client Docker est le principal service utilisé par les utilisateurs de Docker pour communiquer avec Docker. Lorsque nous utilisons les commandes "docker run", le client envoie ces commandes à dockerd, qui les exécute. La commande utilisée par Docker dépend de l'API de Docker. Dans Docker, le client peut interagir avec plus d'un processus démon.

Qu'est-ce qu'une image Docker ?

Les images Docker sont des modèles en lecture seule avec des instructions pour créer un conteneur Docker, qui sont la partie la plus importante du cycle de vie de Docker. Les images peuvent être basées sur d'autres images avec des personnalisations supplémentaires. Les images peuvent être créées en utilisant un fichier Docker avec une syntaxe spécifique. Les instructions dans un Dockerfile créent de nouvelles couches dans l'image, qui peuvent être reconstruites si le Dockerfile est modifié. Les images Docker sont légères, petites et rapides par rapport à d'autres technologies de virtualisation.

Les applications web sont conçues pour générer du contenu basé sur des données récupérées qui changent en fonction de l'interaction de l'utilisateur.

Gestion des registres Docker

Un registre Docker conserve les images Docker. Nous pouvons exécuter notre propre registre privé. Lorsque nous exécutons les commandes docker pull et docker run, les images requises sont retirées de notre répertoire de registre configuré. En utilisant la commande Docker push, l'image peut être téléchargée dans notre répertoire de registre configuré.

Conteneurs Docker

Un conteneur est une instance d'une image. Nous pouvons créer, exécuter, arrêter ou supprimer un conteneur à l'aide de la CLI de Docker. Nous pouvons connecter un conteneur à plusieurs réseaux, ou même créer une nouvelle image basée sur son état actuel. Par défaut, un conteneur est bien isolé des autres conteneurs et de sa machine système. Un conteneur est défini par son image ou par les options de configuration que l'on fournit lors de sa création ou de son exécution.

Utilisation d'un fichier Docker

Un fichier Docker est un fichier texte qui contient toutes les commandes que l'utilisateur peut appeler sur la ligne de commande pour construire une image. L'utilisation de l'image Docker de base permet d'ajouter et de copier des fichiers, d'exécuter des commandes et d'exposer les ports. Le fichier Docker peut être considéré comme le code source et les images à compiler pour notre conteneur qui exécute le code. Les fichiers Docker sont des fichiers portables qui peuvent être partagés, stockés et mis à jour selon les besoins. Certaines des instructions des fichiers Docker sont les suivantes

- **FROM - Ceci est utilisé pour définir l'image de base pour les instructions. Il est très important de le mentionner dans la première ligne du fichier docker.**
- **MAINTAINER - Cette instruction est utilisée pour indiquer l'auteur du fichier docker et son caractère non exécutable.**
- **RUN - Cette instruction nous permet d'exécuter la commande au-dessus de la couche existante et de créer une nouvelle couche avec le résultat de l'exécution de la commande.**
- **CMD - Cette instruction n'exécute rien pendant la construction de l'image docker. Elle spécifie simplement les commandes qui sont utilisées dans l'image.**
- **LABEL - Cette instruction est utilisée pour assigner les métadonnées sous la forme de paires clé-valeur. Il est toujours préférable d'utiliser peu d'instructions LABEL.**
- **EXPOSE - Cette instruction est utilisée pour écouter sur des serveurs d'application spécifiques.**
- **ENV - Cette instruction est utilisée pour définir les variables d'environnement dans le fichier Docker pour le conteneur.**
- **COPY - Cette instruction est utilisée pour copier les fichiers et les répertoires d'un dossier spécifique vers un dossier de destination.**
- **WORKDIR - Cette instruction est utilisée pour définir le répertoire de travail actuel pour les autres instructions, c'est-à-dire RUN, CMD, COPY, etc.**

Docker est le constructeur d'images de conteneurs de la prochaine génération, qui nous aide à rendre les images Docker plus efficaces, plus sûres et plus rapides.

Qu'est-ce que la plateforme et l'architecture de conteneurs Docker ?

Docker est un système de conteneurs qui permet d'exécuter des applications dans un environnement de sandbox. Les conteneurs Docker utilisent la virtualisation du système d'exploitation pour combiner les composants d'un système d'application. Ils sont légers et exécutent directement dans le noyau de la machine, ce qui permet d'en exécuter plusieurs en parallèle. Docker utilise un modèle d'architecture client-serveur et communique via l'API REST et l'interface réseau.

Quelles sont les principales caractéristiques de Docker ?

Docker permet d'assembler rapidement des applications à partir de composants, évitant les erreurs d'expédition du code. Il aide également à tester le code avant de le déployer en production et est simple à utiliser. Les applications peuvent être dockerisées en quelques heures et les conteneurs peuvent être lancés en une minute. Les conteneurs Docker fonctionnent partout, des ordinateurs de bureau aux nuages publics et privés.

Qu'est-ce que la sécurité dans Docker ?

Avant de déployer et d'exécuter Docker, il faut prendre en compte sa sécurité. Les principaux points à vérifier sont le niveau de sécurité du kernel et sa prise en charge des espaces de noms et des groupes, la surface du démon Docker, le fichier de configuration du conteneur qui peut présenter des vulnérabilités par défaut ou avoir été personnalisé, ainsi que la politique de sécurité d'application du kernel et la manière dont il interagit avec les conteneurs.

Qu'est-ce que Docker Compose ?

Docker Compose est un outil pour exécuter plusieurs conteneurs dans les applications Docker. Il utilise un fichier compose pour configurer les services et les démarrer en une seule commande. C'est bénéfique pour les environnements de développement, de test et de mise en scène. Le processus en trois étapes consiste à définir l'environnement de l'application, définir les services dans docker-compose.yml et exécuter l'application avec docker-compose up.

Quelles sont les fonctionnalités de Docker Compose ?

Les caractéristiques de docker compose qui le rendent unique sont les suivantes

- Plusieurs environnements isolés peuvent être exécutés sur un seul hôte
- Stockage des données de volume lors de la création des conteneurs
- Recréer uniquement les conteneurs dont les configurations ont été modifiées.

Les technologies de conteneurisation et d'orchestration de conteneurs telles que Docker et Kubernetes ont permis d'adapter une approche de microservice pour le développement d'applications.