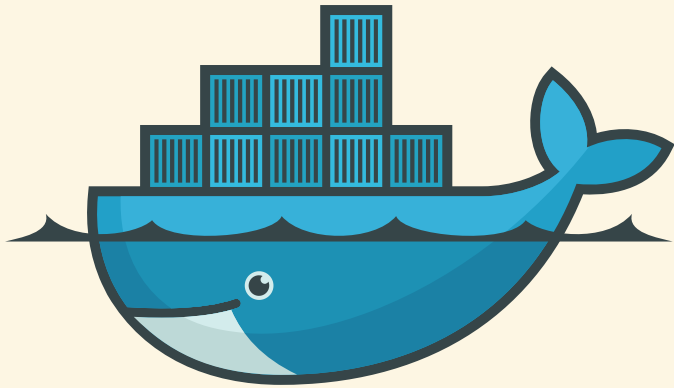


DOCKER



docker



Semifir

HISTORIQUE ET ÉVOLUTION DES TECHNOLOGIES DE CONTENEURISATION

CONTEXTE PRÉ-CONTENEURISATION : VIRTUALISATION TRADITIONNELLE

- Avant l'avènement de la conteneurisation, la virtualisation traditionnelle était dominante.
- Elle impliquait l'utilisation de machines virtuelles (VM) pour exécuter des applications sur des serveurs physiques.
- Chaque VM incluait un système d'exploitation complet, ce qui pouvait entraîner une surcharge significative des ressources système.

ÉMERGENCE DES TECHNOLOGIES DE CONTENEURISATION

1. **Chroot** : Cette fonctionnalité Unix permettait de changer le répertoire racine pour un processus, offrant une forme primitive d'isolation.
2. **LXC (Linux Containers)** : LXC a introduit une virtualisation légère en utilisant les cgroups et les namespaces du kernel Linux pour isoler les processus et les ressources.

3. Cgroups et namespaces du kernel Linux : Ces fonctionnalités du kernel Linux ont permis une isolation efficace des ressources système, ouvrant la voie à des technologies de conteneurisation plus avancées.

4. Docker et l'avènement de la conteneurisation moderne : Docker, lancé en 2013, a popularisé la conteneurisation en simplifiant le processus de création, de distribution et de gestion des conteneurs. Il a introduit un écosystème robuste et des outils conviviaux pour les développeurs et les opérationnels.

LES BASES DE LA CONTENEURISATION

QU'EST-CE QU'UN CONTENEUR ?

- Un conteneur est une unité d'exécution logicielle légère qui encapsule une application et toutes ses dépendances, y compris les bibliothèques et les variables d'environnement requises pour son fonctionnement.

DIFFÉRENCES ENTRE LA VIRTUALISATION ET LA CONTENEURISATION

- La virtualisation crée des machines virtuelles distinctes avec leur propre système d'exploitation, tandis que la conteneurisation partage le noyau du système hôte et isole les processus et les ressources au niveau de l'application.

AVANTAGES DE LA CONTENEURISATION

- Utilisation plus efficace des ressources système
- Démarrage rapide des applications
- Isolation des dépendances logicielles
- Portabilité entre les environnements de développement, de test et de production

LEXIQUE DOCKER

IMAGE

- Une image Docker est un modèle de lecture seule utilisé pour créer des conteneurs.
- Elle contient le système de fichiers de l'application, ainsi que tous les paramètres et dépendances nécessaires à son exécution.

CONTENEUR

- Un conteneur Docker est une instance en cours d'exécution d'une image Docker.
- Il encapsule l'application et ses dépendances, offrant un environnement isolé et portable.

DOCKER ENGINE

- Docker Engine est le moteur qui gère la création, l'exécution et la gestion des conteneurs Docker.
- Il comprend un démon de service et un ensemble d'API pour interagir avec les conteneurs.

DOCKERFILE

- Un Dockerfile est un fichier de configuration utilisé pour décrire les étapes nécessaires à la construction d'une image Docker.
- Il contient des instructions pour installer des dépendances, copier des fichiers, et configurer l'environnement de l'application.

DOCKER COMPOSE

- Docker Compose est un outil permettant de définir et de gérer des applications multi-conteneurs.
- Il utilise un fichier de configuration YAML pour spécifier les services, les réseaux et les volumes nécessaires à l'application.

DOCKER HUB

- Docker Hub est un registre de conteneurs public où les utilisateurs peuvent partager, découvrir et télécharger des images Docker.
- Il offre également des fonctionnalités de gestion des versions et des déploiements automatisés.

COMPOSITION D'UN CONTENEUR

COUCHES D'UNE IMAGE DOCKER

- Une image Docker est composée de couches empilées les unes sur les autres.
- Chaque couche représente une instruction dans le Dockerfile et peut être partagée entre plusieurs images.

SYSTÈME DE FICHIERS EN COUCHES

- Le système de fichiers en couches d'une image Docker permet une gestion efficace des modifications et des mises à jour.
- Les couches sont en lecture seule et les modifications sont stockées dans une couche de conteneur temporaire.

SURCOUCHE DE CONTENEUR

- Lorsqu'un conteneur est lancé à partir d'une image, une surcouche de conteneur en écriture est ajoutée au-dessus des couches de l'image.
- Cela permet au conteneur de stocker les modifications apportées au système de fichiers pendant son exécution.

FONCTIONNEMENT

UTILISATION DES NAMESPACES ET CGROUPS LINUX

- Les conteneurs Docker utilisent les namespaces Linux pour isoler les processus, les utilisateurs, les réseaux et d'autres ressources système.
- Les cgroups sont utilisés pour limiter et contrôler l'accès aux ressources matérielles.

ISOLATION DES RESSOURCES

- Chaque conteneur Docker dispose de ses propres ressources isolées, telles que l'espace de noms réseau, les variables d'environnement et les systèmes de fichiers, garantissant une séparation complète avec les autres conteneurs et le système hôte.

COMMUNICATION AVEC LE SYSTÈME HÔTE

- Les conteneurs Docker peuvent communiquer avec le système hôte via des interfaces réseau et des points de montage partagés.
- Cette communication est contrôlée et isolée pour empêcher les conteneurs d'interférer avec le fonctionnement du système hôte.

COMMUNICATION ENTRE CONTENEURS

DOCKER NETWORK : CONCEPTS ET TYPES DE RÉSEAUX

- Docker prend en charge différents types de réseaux, tels que le pont, l'hôte et le sur-réseau, pour permettre la communication entre les conteneurs.
- Chaque réseau fournit un niveau d'isolation et de performance spécifique.

LIAISONS ENTRE CONTENEURS

- Les conteneurs Docker peuvent être liés entre eux pour établir des communications directes via des adresses IP ou des noms de conteneurs.
- Cette fonctionnalité facilite le déploiement d'applications multi-conteneurs.

EXPOSITION DE PORTS ET ACCÈS AUX SERVICES

- Les ports des conteneurs peuvent être exposés et publiés pour permettre l'accès aux services à partir du système hôte ou d'autres conteneurs.
- Cela facilite l'interaction entre les différents composants d'une application distribuée.

FONCTIONNEMENT DE DOCKER

ARCHITECTURE DE DOCKER ENGINE

- Docker Engine est composé d'un démon de service et d'une API RESTful qui interagit avec les clients Docker.
- Le daemon gère la création, l'exécution et la gestion des conteneurs, tandis que l'API permet le contrôle et la surveillance à distance.

CYCLE DE VIE D'UN CONTENEUR DOCKER

- Le cycle de vie d'un conteneur Docker comprend les étapes de création, de démarrage, d'arrêt, de redémarrage et de suppression.
- Chaque étape peut être contrôlée à l'aide des commandes Docker appropriées.

GESTION DES IMAGES ET DES CONTENEURS

- Docker offre des commandes pour gérer les images et les conteneurs, y compris la construction, le téléchargement, la mise à jour, le démarrage, l'arrêt et la suppression.
- Ces commandes permettent aux utilisateurs de contrôler facilement l'environnement de leurs applications.

DOCKER HUB

PLATEFORME DE PARTAGE D'IMAGES DOCKER

- Docker Hub est un registre de conteneurs public où les développeurs peuvent partager, découvrir et télécharger des images Docker.
- Il offre également des fonctionnalités de gestion des versions, de sécurité et de collaboration.

RECHERCHE ET TÉLÉCHARGEMENT D'IMAGES

- Les utilisateurs peuvent rechercher des images Docker sur Docker Hub en utilisant des mots-clés et des balises spécifiques.
- Ils peuvent ensuite télécharger les images et les utiliser pour leurs propres projets.

PUBLICATION ET PARTAGE D'IMAGES PERSONNALISÉES

- Les développeurs peuvent publier et partager leurs propres images Docker sur Docker Hub pour permettre à d'autres utilisateurs de les utiliser.
- Cela facilite la distribution et le partage des logiciels et des infrastructures.

FIRECRACKER ET LES MICRO VM

<https://firecracker-microvm.github.io/>

<https://katacontainers.io/>

FIRECRACKER : TECHNOLOGIE DE VIRTUALISATION DE CONTENEURS LÉGÈRE

- Firecracker est une technologie de virtualisation de conteneurs légère conçue pour les charges de travail cloud-native.
- Elle utilise la virtualisation basée sur les noyaux KVM pour créer des micro-VM isolées et sécurisées.

AVANTAGES PAR RAPPORT À LA VIRTUALISATION TRADITIONNELLE

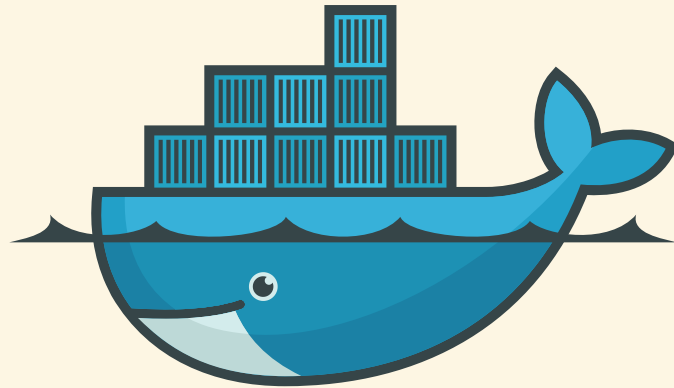
- Firecracker offre un démarrage rapide, une empreinte mémoire réduite et une isolation efficace des ressources par rapport à la virtualisation traditionnelle.
- Cela en fait un choix idéal pour les environnements de conteneurisation hautement évolutifs.

UTILISATION CONJOINTE AVEC DOCKER POUR DES SCÉNARIOS SPÉCIFIQUES

- Docker peut être utilisé en conjonction avec Firecracker pour des scénarios spécifiques nécessitant une isolation supplémentaire ou une sécurité renforcée.
- Firecracker offre une alternative légère à la virtualisation traditionnelle pour les charges de travail cloud-native.

LA SUITE

Dockerfile



docker