



```
logo_treeptik
```

L'essentiel pour maîtriser Docker by Treeptik



L'essentiel pour maîtriser

Sommaire

- Prérequis
- Evolution de la virtualisation et des applications
- Présentation de Docker
- Mise en oeuvre de conteneurs
- Gestion des images
- Gestion des volumes et drivers
- Le modèle de réseau Docker
- Configuration du Docker Daemon avec "systemd"
- Déployer une application multi conteneurs
- Provisionnement sur le Cloud
- Intégration / déploiement continu avec Docker
- Cluster et orchestration d'une application multi conteneurs
- Gestion de logs avancée
- Comparatif des orchestrateurs
- Aperçu des outils de supervision dans un environnement de conteneurs

Installation de Docker

Installation de Docker

Les différentes versions

Docker est disponible en en deux versions :

- Docker CE (édition communauté)
- Docker EE (édition entreprise)

L'ensemble du programme de formation est basé sur la version Docker CE. Docker CE est disponible sur les plateformes suivantes :

- Mac
- Windows
- Debian, Fedora, Ubuntu, CentOS
- Amazon Web Services
- Microsoft Azure

Installation de Docker

Les différentes méthodes d'installation

Vous pouvez installer Docker CE de différentes façons :

- Depuis les dépôts officiels de Docker
- Depuis leur gestionnaire de paquets
- Des scripts fournis par Docker

Installation de Docker

Installation (1/4)

- Mise à jour des paquets

```
$ sudo yum update
```

- Préparation des pré-requis Docker

```
$ sudo yum install -y yum-utils device-mapper-persistent-data lvm2
```

- Ajout de la clé officielle

```
$ sudo rpm --import https://download.docker.com/linux/centos/gpg
```

Installation de Docker

Installation (2/4)

- Vérifiez que vous avez maintenant la clé avec l'empreinte digitale 060A 61C5 1B55 8A7F 742B 77AA C52F EB6B 621E 9F35, en recherchant les 8 derniers caractères de l'empreinte digitale

```
$ sudo rpm -qi gpg-pubkey-621e9f35
```

```
Name       : gpg-pubkey
Version    : 621e9f35
Release    : 58adea78
Architecture: (none)
Install Date: mar. 21 nov. 2017 15:17:30 CET
Group      : Public Keys
Size       : 0
License    : pubkey
Signature  : (none)
Source RPM : (none)
Build Date : mer. 22 févr. 2017 20:46:00 CET
Build Host : localhost
Relocations : (not relocatable)
Packager   : Docker Release (CE rpm) <docker@docker.com>
Summary    : gpg(Docker Release (CE rpm) <docker@docker.com>)
Description :
```


Installation de Docker

Installation (3/4)

- Utilisez la commande suivante pour configurer le dépôt stable

```
$ sudo yum-config-manager \
--add-repo \
https://download.docker.com/linux/centos/docker-ce.repo
```

- Mettre à jour les paquets

```
$ sudo yum update
```

- Installer docker-ce, l'activer au démarrage et le lancer

```
$ sudo yum install docker-ce
```

```
$ sudo systemctl enable docker && sudo systemctl start docker
```

Installation de Docker

Installation (4/4)

- Vérifier que Docker CE est installé en exécutant la commande suivante

```
$ sudo docker run hello-world
```

- Cette commande exécute un conteneur de test et affiche un message d'information

```
$ sudo groupadd docker
```

- Permettre d'utiliser le client Docker sans sudo à l'utilisateur courant

```
$ sudo usermod -aG docker treeptik
```

- Redémarrer la session pour prendre en compte les changements

Evolution de la virtualisation et des applications

Objectifs :

- Introduire le concept de virtualisation à base de conteneur
- Connaître les bienfaits des conteneurs
- Comprendre les différences entre les VMs et les conteneurs

Evolution de la virtualisation et des applications

Un peu d'histoire...

Serveur physique

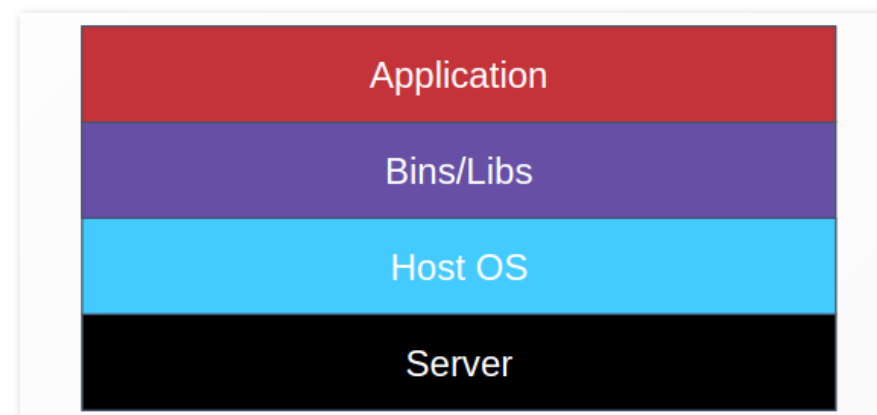
De manière historique, le serveur physique était la plateforme de référence pour le déploiement d'une application. Certaines limites sur ce modèle sont apparues et ont été identifiées:

- Les temps de déploiements peuvent être longs
- Les coûts peuvent être élevés
- Beaucoup de ressources perdues
- Difficultés à mettre à l'échelle pour de la haute disponibilité
- Complexité de migration

Evolution de la virtualisation et des applications

Un peu d'histoire...

Représentation d'une application sur serveur physique

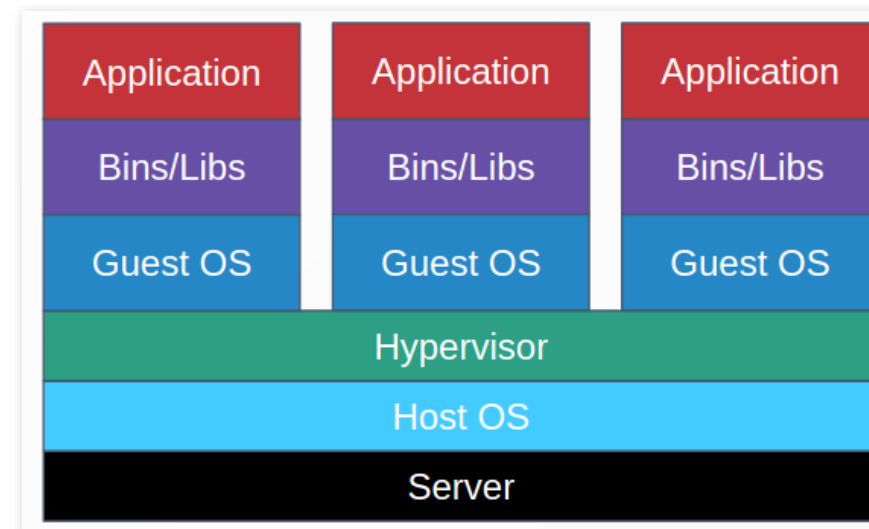


Evolution de la virtualisation et des applications

Un peu d'histoire...

Virtualisation basée sur hyperviseur

- Un serveur physique peut héberger plusieurs applications distinctes
- Chaque application tourne dans une machine virtuelle



Evolution de la virtualisation et des applications

Un peu d'histoire...

Avantages et limitations du modèle de machine virtuelle

Les bénéfices:

- Un serveur physique est divisé en plusieurs machines virtuelles
- Plus facile à mettre à l'échelle qu'un serveur physique
- Modèle de Cloud et paiement à la demande (AWS, Azure, Rackspace..)

Les limitations:

- Chaque VM nécessite une allocation de CPU, de stockage dédié, de RAM et un OS complet
- Modèle linéaire: l'augmentation du nombre de VM nécessite des ressources supplémentaires
- L'utilisation d'un système hôte complet entraîne une surcharge
- La portabilité des applications n'est toujours pas garantie

Evolution de la virtualisation et des applications

Un peu d'histoire...

Historique du modèle d'isolation de processus

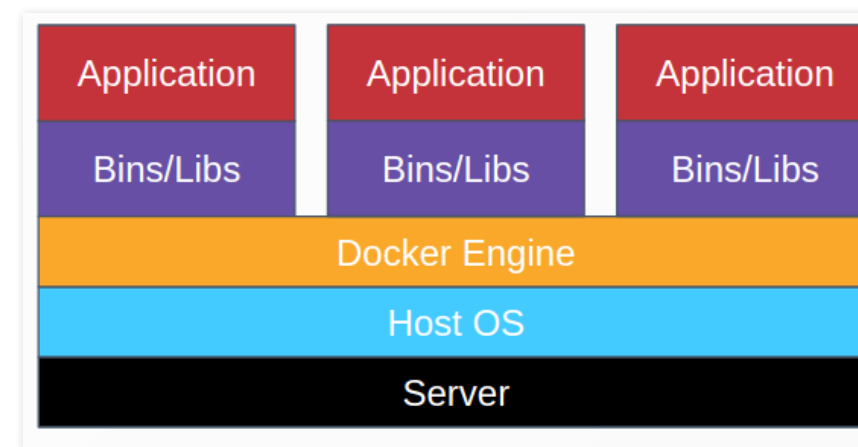
- UNIX chroot (1979-1982)
- BSD Jail (1998)
- Parallels Virtuozzo (2001)
- Solaris Containers (2005)
- Linux LXC (2008)
- Docker (2013)

Docker est une évolution de LXC qui a permis de rendre le container utilisable par un plus grand nombre d'utilisateurs grâce à son API et son client convivial.

Evolution de la virtualisation et des applications

Un peu d'histoire...

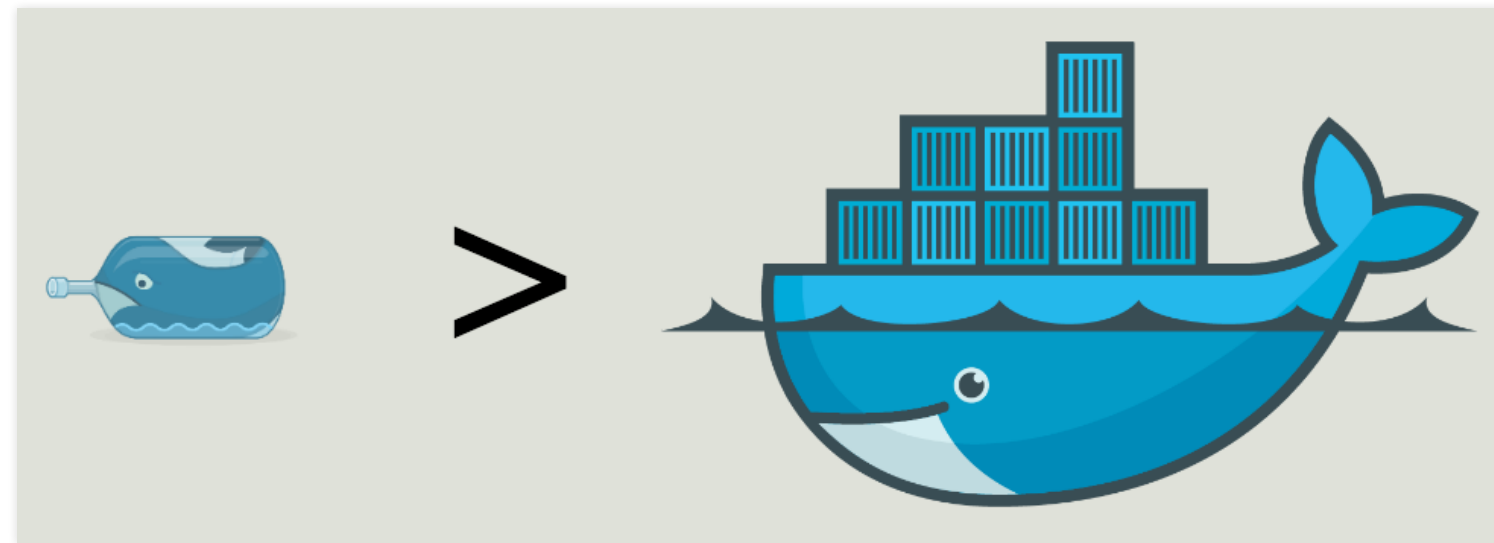
- La containerisation utilise le kernel du système hôte pour démarrer de multiples systèmes de fichiers racine
- Chaque système de fichier racine est appelé container
- Chaque container possède ses propres processus, mémoires, carte réseau



Evolution de la virtualisation et des applications

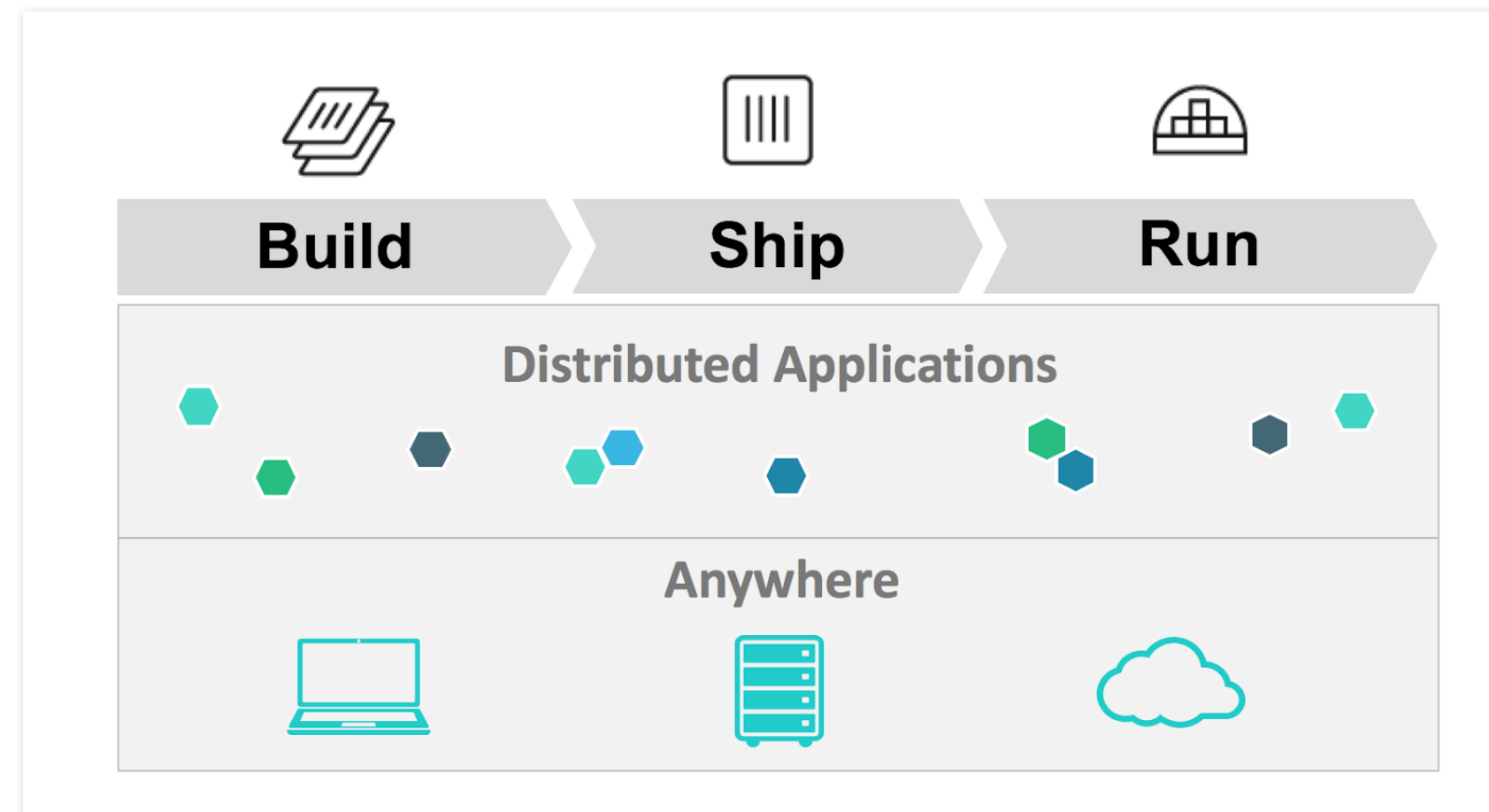
Pourquoi utiliser les containers ?

- Les containers sont plus légers et rapides que les VMs
- Pas besoin d'installer un système d'exploitation complet
- En conséquence, les besoins en CPU, RAM et stockage sont moins contraignants
- On peut faire tourner bien plus de containers sur un serveur que de VMs
- Le concept assure une meilleure portabilité
- Les containers représentent une meilleure solution pour développer et déployer des applications microservices



Evolution de la virtualisation et des applications

La mission de docker



Evolution de la virtualisation et des applications

Docker et le noyau Linux

Le Docker Engine est l'application qui active les containers destinés à être exécutés

Il utilise plusieurs fonctionnalités "natives" du noyau Linux.

Les NAMESPACES permettent:

- l'isolation des processus et du système de fichier
- l'isolation réseau et de disposer de ses propre interfaces

Les CGROUPS (Control Groups) permettent:

- de mesurer et limiter les ressources (RAM, CPU, block I/O, network)
- de donner l'accès au différents périphériques (/dev/)

Les règles IPTABLES permettent:

- d'assurer la communication entre containers sur le même hôte
- d'assurer d'éventuelles communication entre les containers et l'extérieur

Evolution de la virtualisation et des applications

Le cycle de vie d'un container diffère de celui d'une VM

A l'opposé d'une VM, le conteneur n'est pas destiné à une existence perpétuelle. L'orchestrateur se chargera de redémarrer le container sur un autre hôte en cas de défaillance.

Cycle de vie basique d'un container:

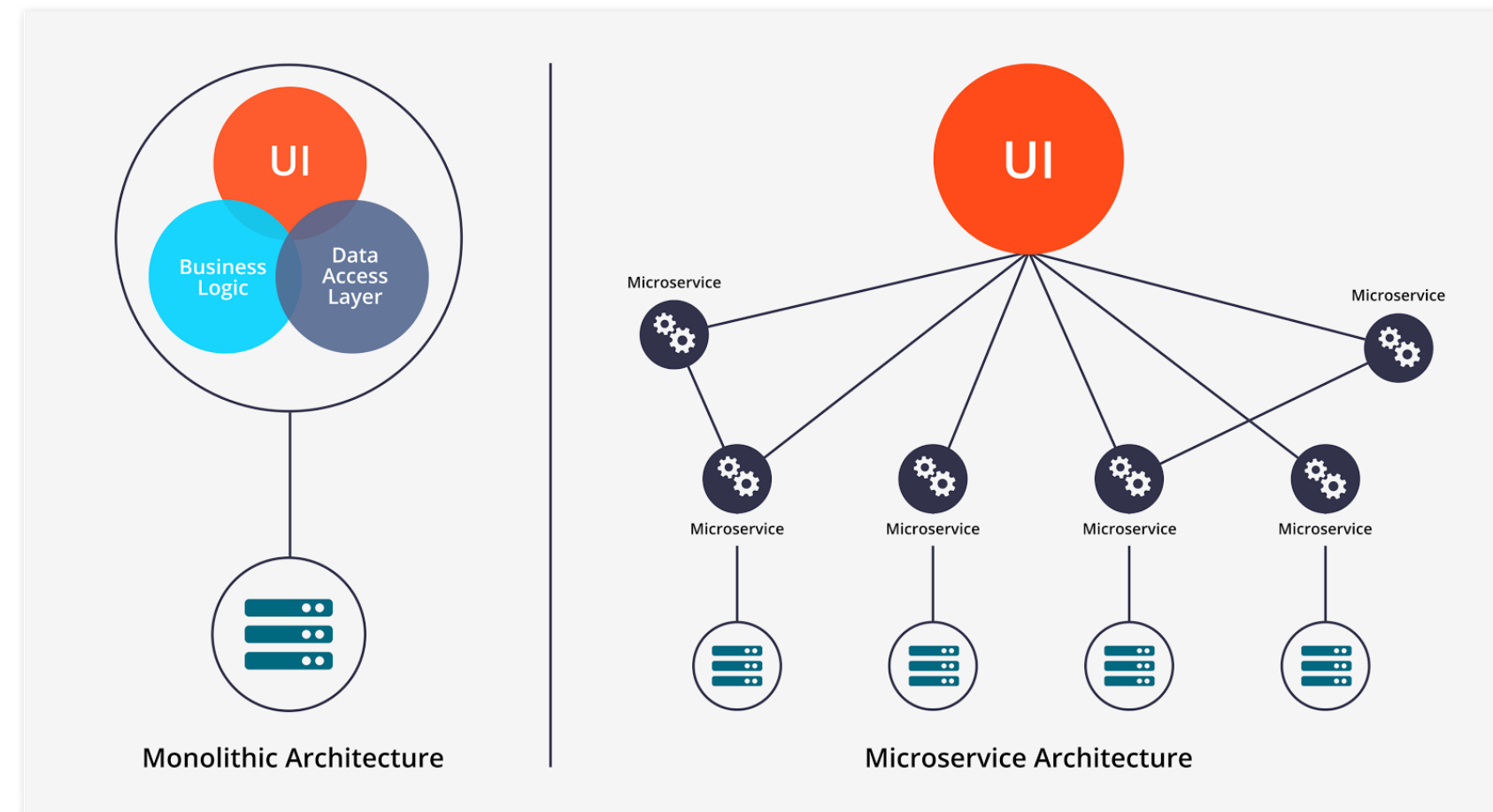
- Création d'un container à partir d'une image
- Démarrage d'un container avec un processus
- Le processus se termine et le container s'arrête
- Le container est détruit

Evolution de la virtualisation et des applications

Applications modernes: architecture micro services

Une architecture microservices est un ensemble complexe d'applications décomposé en plusieurs processus indépendants et faiblement couplés.

Ces processus communiquent les uns avec les autres en utilisant des API. L'API REST est souvent employée pour relier chaque microservice aux autres.



Présentation de docker

Objectifs

- Connaître le "langage" de docker
- Connaître l'architecture docker
- Comprendre comment marche un conteneur , une image

Présentation de docker

Historique du projet et présentation de Docker INC

- Création en 2008 de DotCloud par Solomon Hykes
- DotCloud proposait une plateforme PaaS avec une technologie de conteneurs
- Création en 2013 de Docker par Solomon Hykes
- Docker est distribué en tant que projet open source proposant une surcouche qui automatise et simplifie le déploiement d'applications dans des conteneurs
- Deux versions du produit: Community Edition & Enterprise Edition
- Les version Docker CE/EE implémentent les nouvelles fonctionnalités du projet Moby
- Moby est un projet open source lancé par Docker afin d'alimenter les versions Docker CE & Docker EE

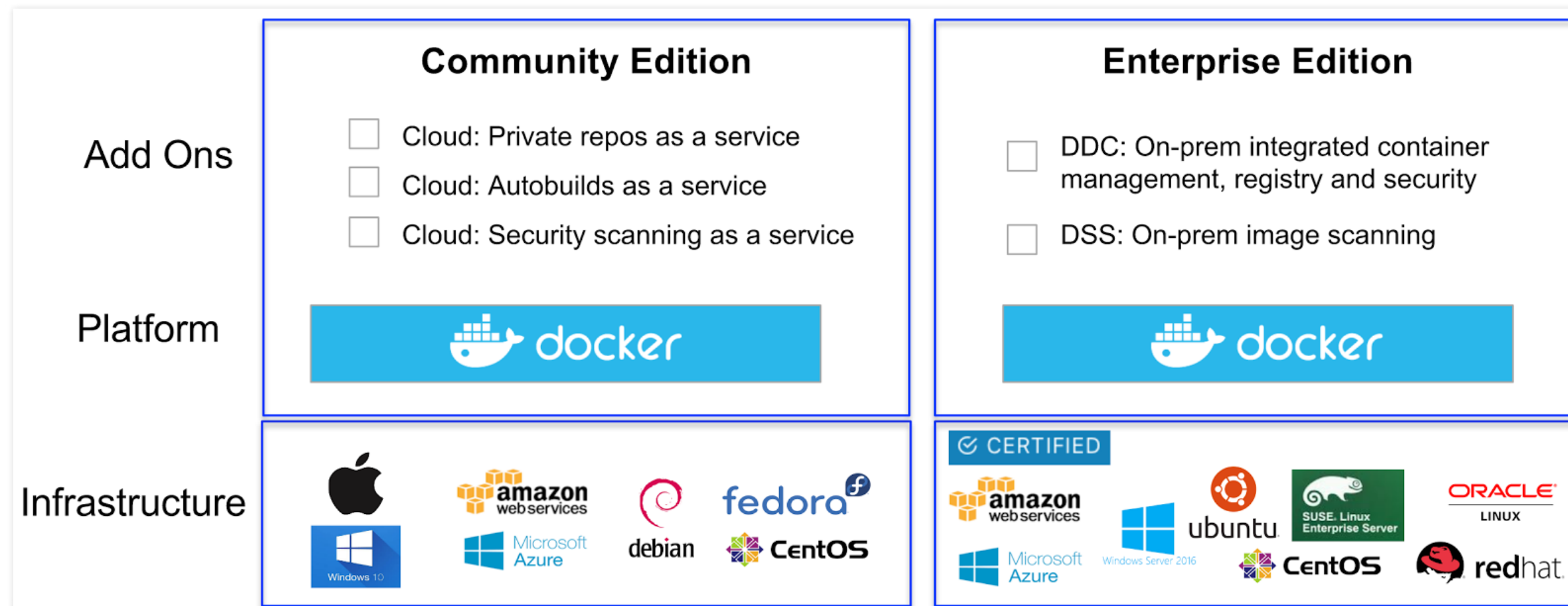
Présentation de docker

Quelques chiffres...

- 8B+ image téléchargés
- 3000 contributeurs
- 500,000+ applications
- 32000 étoiles sur github
- 100000+ projets tiers utilisant Docker

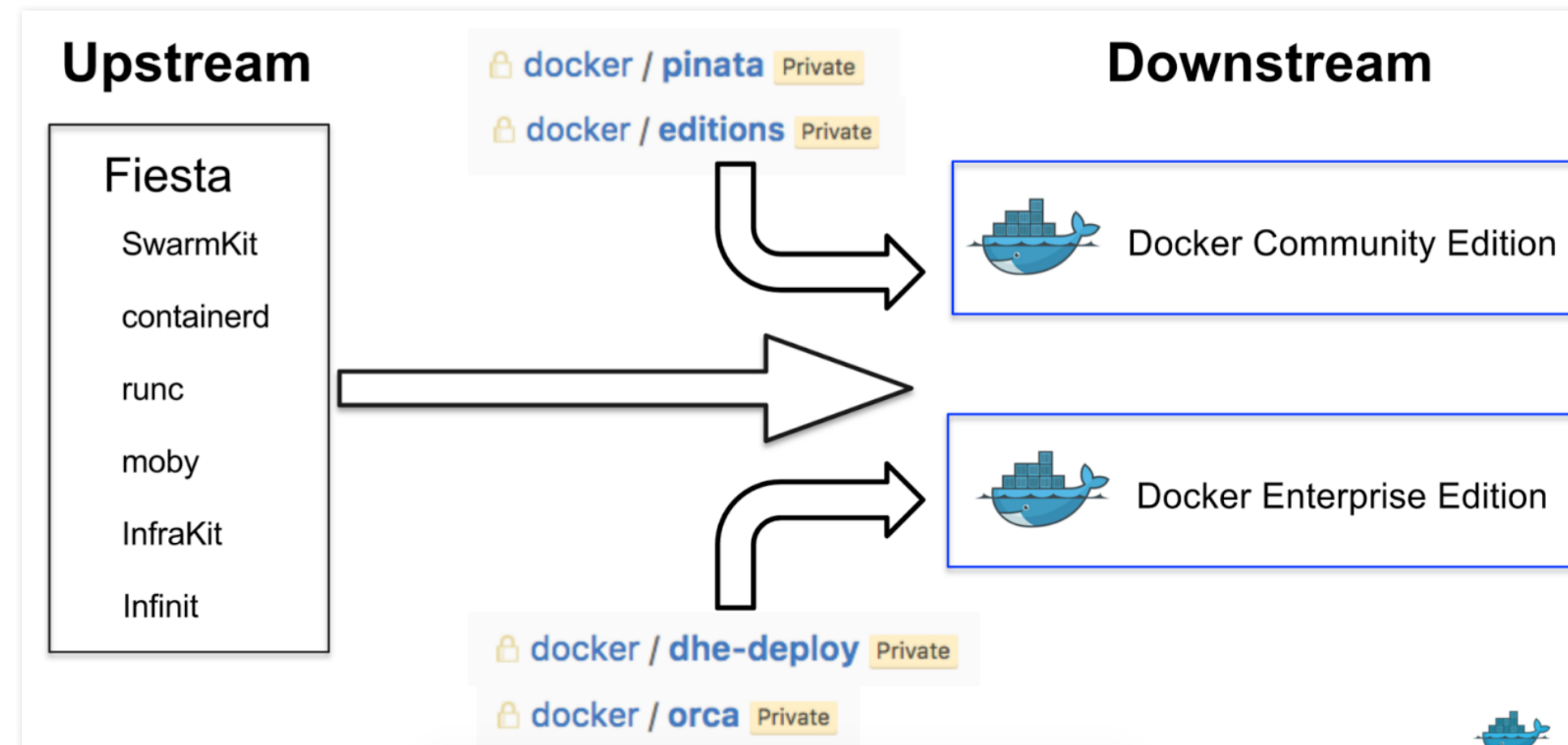
Présentation de docker

Positionnement de Docker et composants de l'offre



Présentation de docker

Positionnement de Docker et composants de l'offre



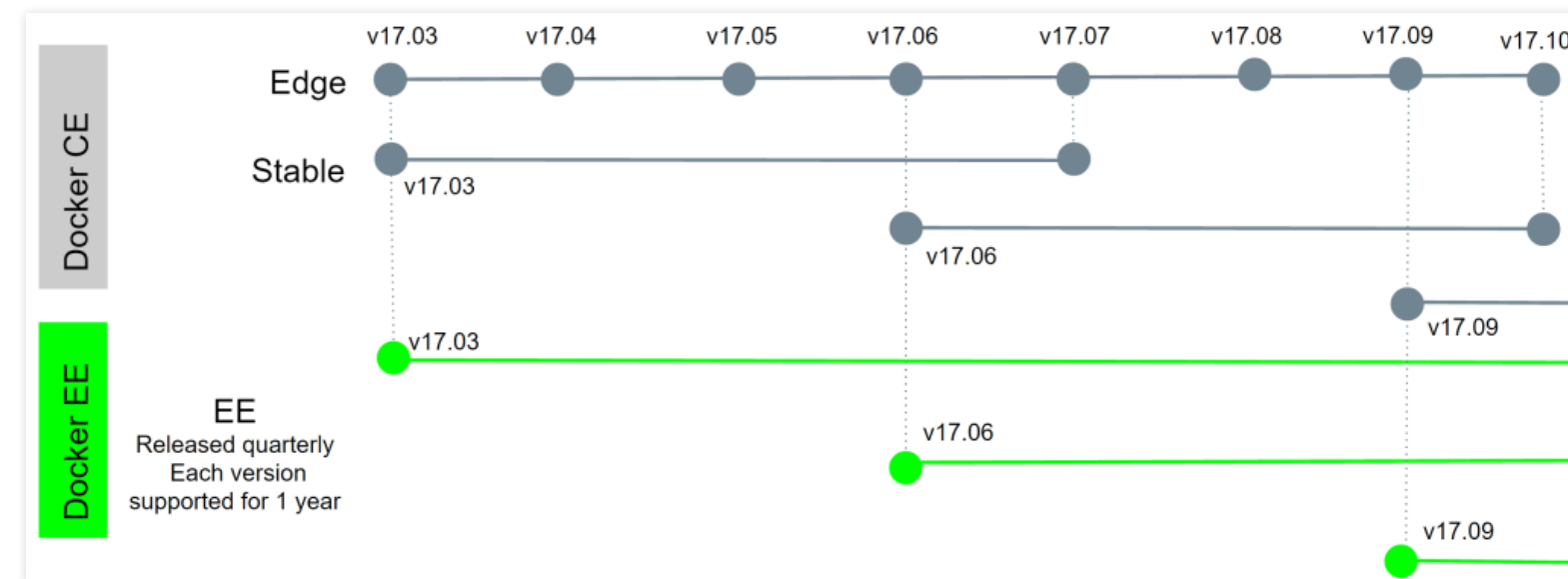
Présentation de docker

Positionnement de Docker et composants de l'offre

	Docker Enterprise Edition		
	Basic	Standard	Advanced
Container engine and built in orchestration	X	X	X
Image management (private registry, caching)		X	X
Integrated container app management		X	X
Multi-tenancy, RBAC, LDAP/AD		X	X
Integrated secrets mgmt, image signing, policy		X	X
Security Scanning			X

Présentation de docker

Positionnement de Docker et composants de l'offre



Séparation des tâches :

- Les développeurs se concentrent sur la création de leurs applications
- Les administrateurs système se concentrent sur le déploiement

Portabilité des applications :

- Construire dans un environnement, puis l'expédier à un autre

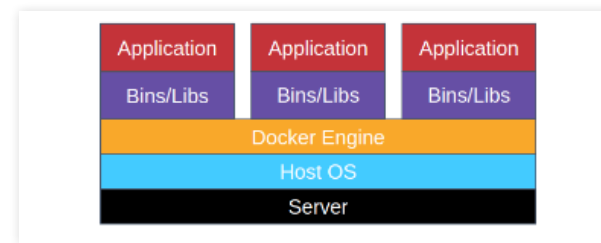
Évolutivité :

- Création de nouveaux conteneurs si nécessaire et de manière rapide
- Exécuter plusieurs applications sur un même hôte

Cycle de développement rapide

Présentation de docker

Architecture de l'application Docker



Docker est une nouvelle technologie à conteneurs. Cette technologie est considérée comme une solution de virtualisation du système d'exploitation. Ce dernier fournit un ensemble d'espaces d'utilisateurs qui sont isolés les uns des autres et offre un niveau d'abstraction tel que les applications pensent faire partie de l'espace utilisateur unique de l'hôte.

Présentation de docker

Conteneurs, images et dépôts

Conteneurs

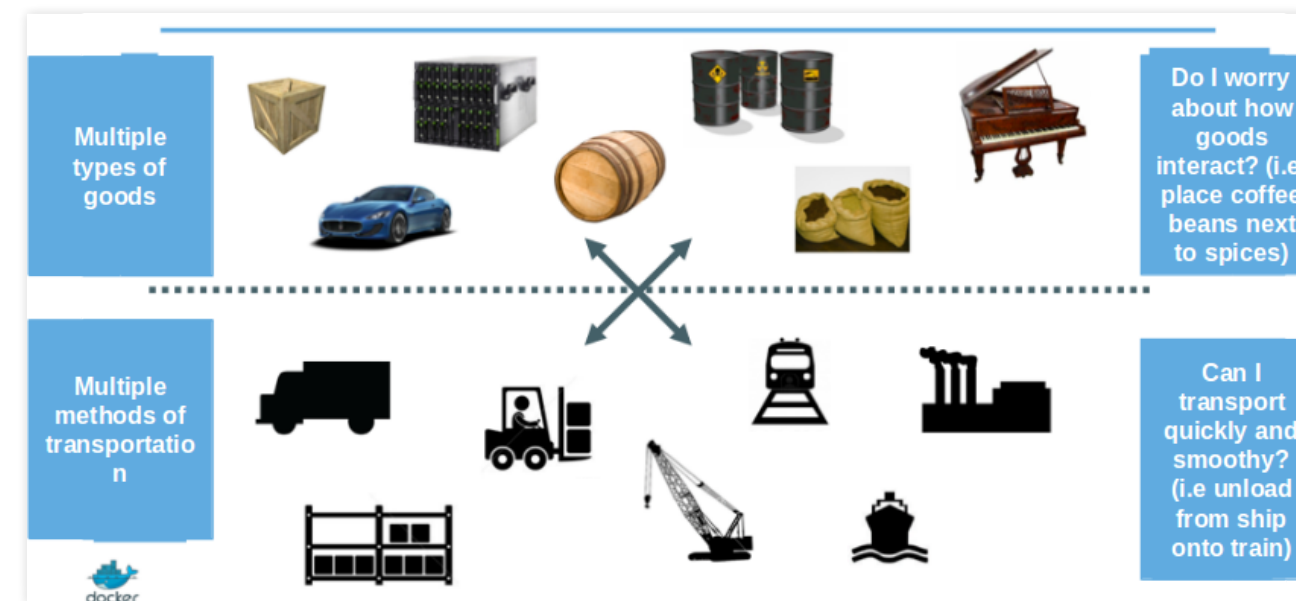
- Plate-forme d'application isolée
- Contient tout ce qui est nécessaire pour exécuter votre application
- Un conteneurs possède un processeur, une mémoire, des dispositifs ainsi que des stacks réseaux
- Un conteneur est une instance d'exécution d'une image

Images

- Construites par vous ou par d'autres utilisateurs
- Peut être partagée avec d'autres utilisateurs
- Forment un ensemble de couches
- Contient toutes les dépendances pour votre application

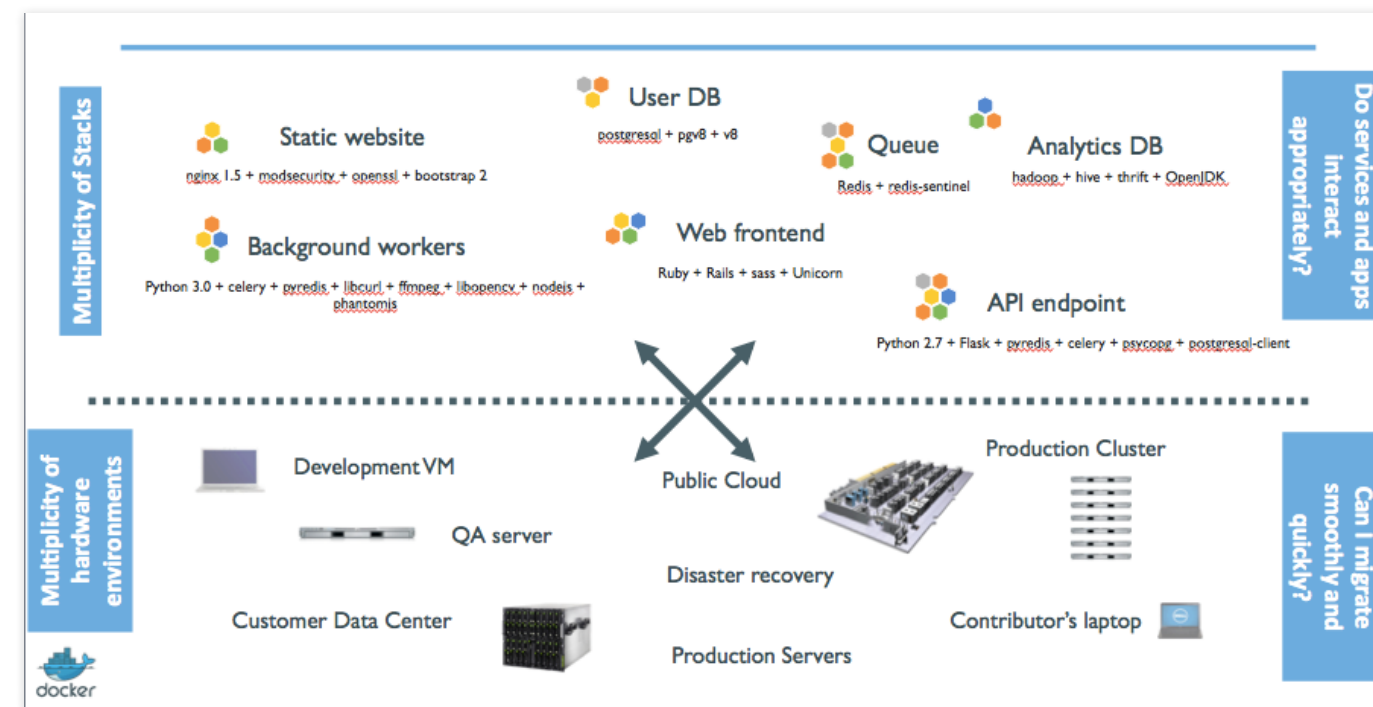
Présentation de docker

Conteneurs, images et dépôts



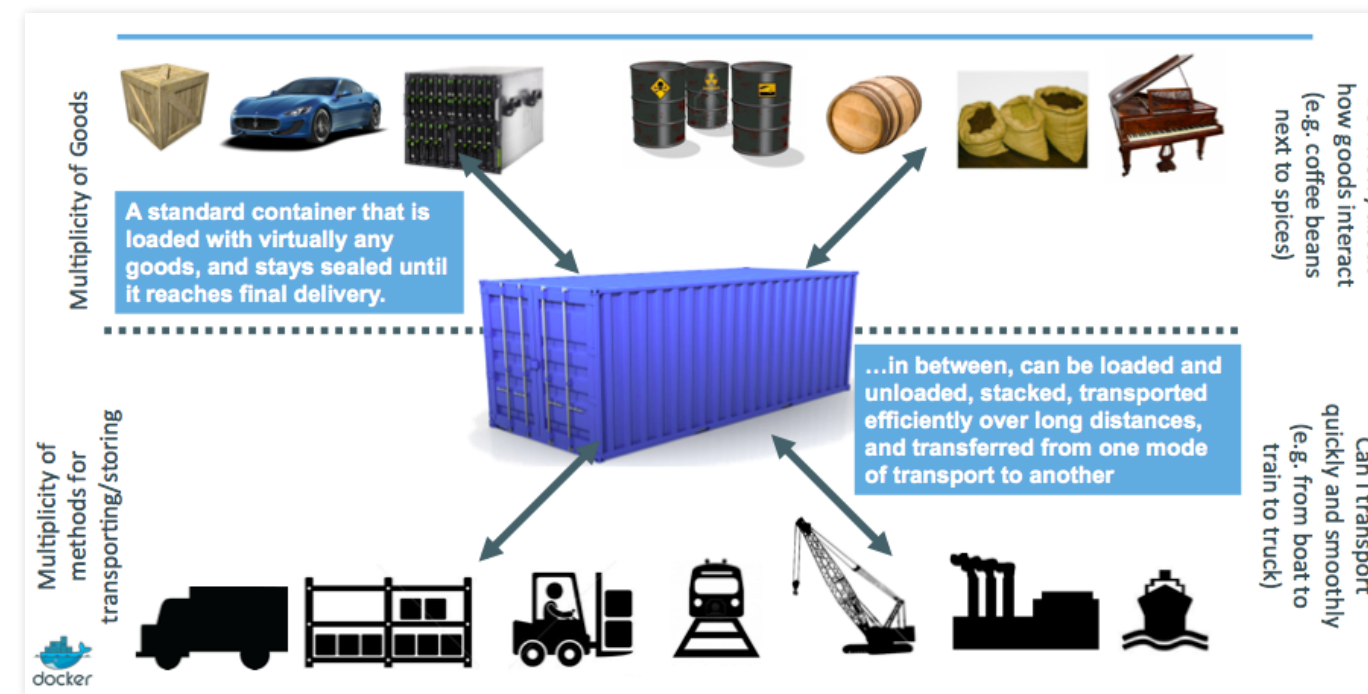
Présentation de docker

Conteneurs, images et dépôts



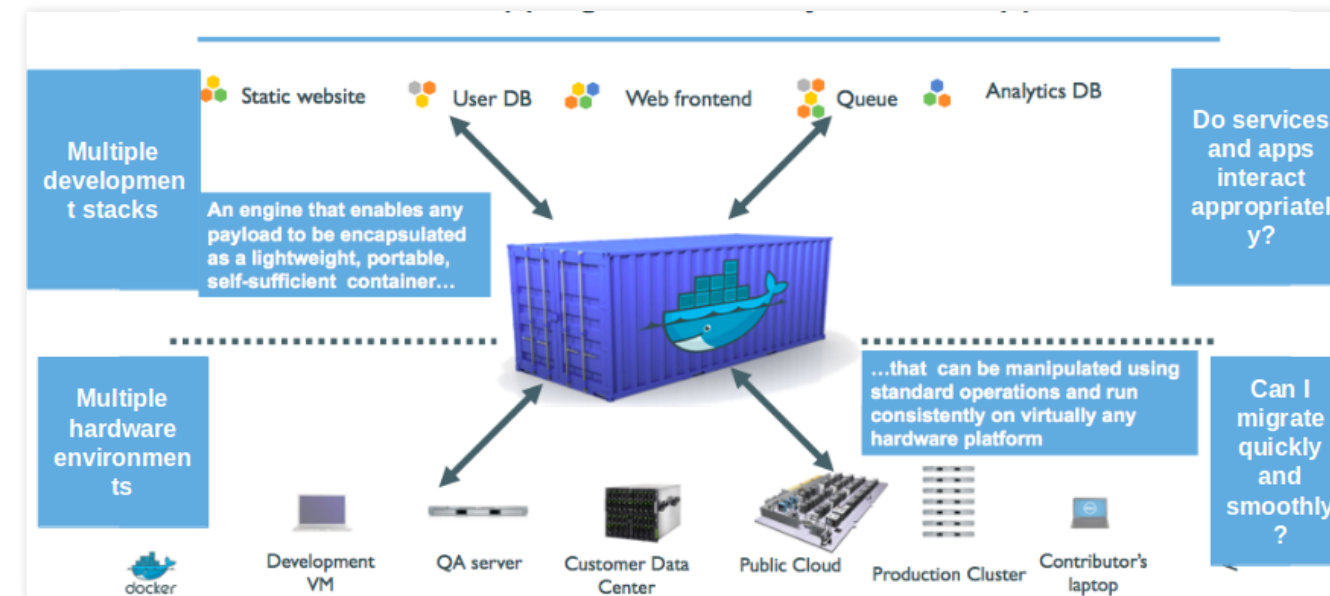
Présentation de docker

Conteneurs, images et dépôts



Présentation de docker

Conteneurs, images et dépôts



Présentation de docker

Conteneurs, images et dépôts

Dépôts

- Les dépôts Docker ont le rôle de stocker des images docker par catégorie.
- L'ensemble des dépôts forme un registre (exemple: Docker hub)

