



UNIVERSITÉ
CAEN
NORMANDIE

La conteneurisation sous Linux

Virtualisation - CM



Maxime Lambert

2023 / 2024



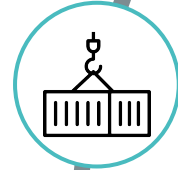
Plan



UNIVERSITÉ
CAEN
NORMANDIE



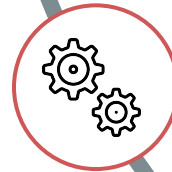
Les conteneurs



La plateforme
Docker



Le socle technologique
de Docker



- Vous devez déjà être familier avec les modèles de gestion de projet tels que le cycle en V ou les méthodologies agiles.
- L'analyse des besoins, les spécifications, la conception, le codage, les tests et la recette sont des termes que vous connaissez.
- Vous êtes-vous déjà confronté à la problématique du déploiement dans une infrastructure, de la maintenance et de l'évolution d'une solution logicielle ?
- La gestion des différents environnements d'exécution, les configurations et les versions applicatives peuvent être particulièrement complexes. De plus, pour des raisons de sécurité une certaine isolation est souhaitable entre les applications.
- La virtualisation à l'aide d'un hyperviseur apporte une première solution, gage d'une excellente isolation matérielle. Toutefois, cette isolation à un coût. Chaque machine virtuelle exécute un système d'exploitation, ce qui se traduit par une consommation conséquente de ressources matérielles. De plus, leur déploiement est relativement lourd.
- Les architectures à base de conteneurs visent à adresser ces limitations.

Les conteneurs



Les conteneurs

L'histoire du transport intermodal



UNIVERSITÉ
CAEN
NORMANDIE



- Avant le conteneur, il y avait beaucoup de manutention.
- Son but est de réduire les temps de transbordement **quelle que soit la marchandise**
- 1958, premier navire porte-conteneurs
- **Standardisation ISO** du conteneur en 1968
- Révolution du transport maritime

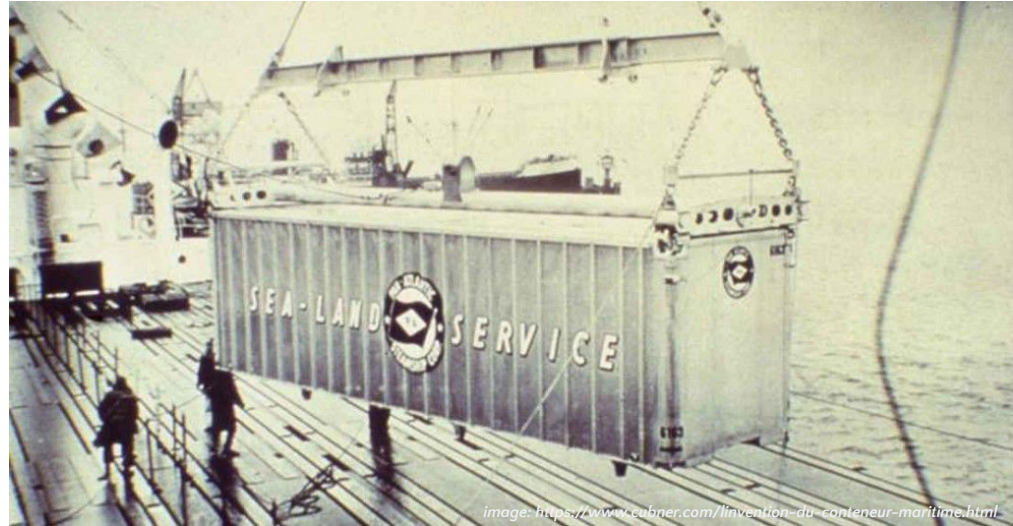


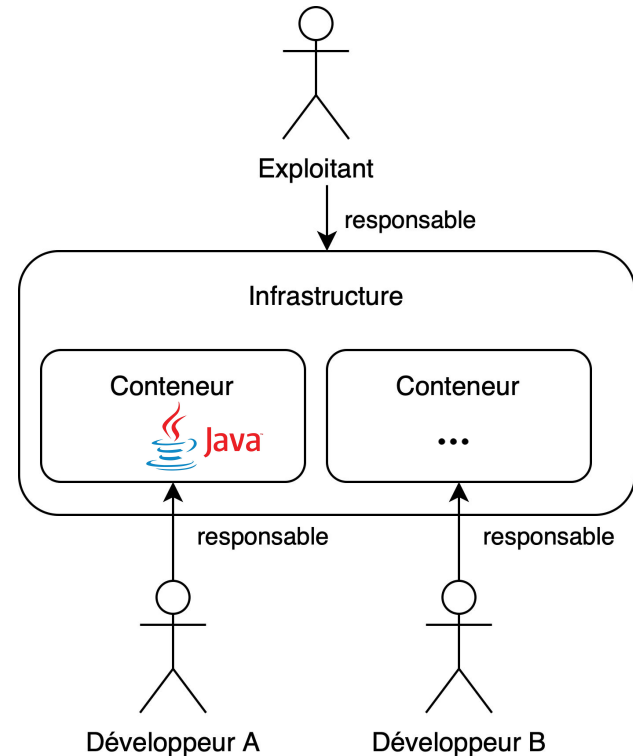
image: <https://www.eubner.com/l'invention-du-conteneur-maritime.html>

Les conteneurs

Les conteneurs en informatique

Dans une architecture à base de conteneurs:

- **Le développeur a la responsabilité du contenu du conteneur:** l'applicatif, ses dépendances et sa configuration
- **L'exploitant a la responsabilité de la gestion du conteneur** au sein de son infrastructure: déploiement, réseau, stockage et allocation de ressources,



Les conteneurs

Historique



UNIVERSITÉ
CAEN
NORMANDIE



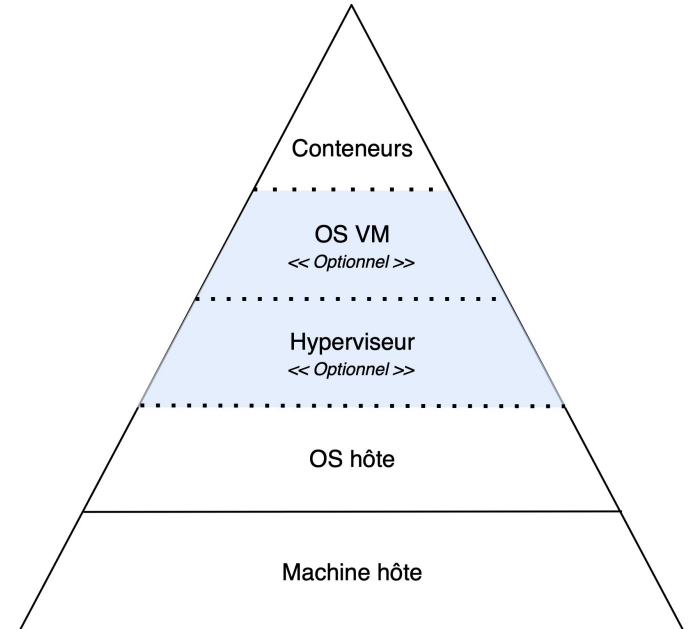
- Prémices à la fin des années 1970 avec l'appel système **chroot**
- Premiers conteneurs en 2000 avec les **BSD Jails**
- Les **espaces de noms (*namespaces*)**, apparus en 2002, sont la première brique nécessaire à la conteneurisation sur Linux.
- En 2008, ajout des **PID *namespaces*** et lancement du projet LXC (Linux Containers)
- Naissance de la société **Docker** en 2010 et projet rendu **Open Source** en 2013
- Création de l'**Open Container Initiative** en 2015

Les conteneurs

Lien avec la virtualisation



- La conteneurisation est également qualifiée de **virtualisation au niveau du système d'exploitation**.
- Un conteneur **ne contient pas un système d'exploitation**.
Il s'appuie sur le noyau de la machine hôte l'accueillant
- La portabilité d'un conteneur **est limitée à des OS de la même famille**.
- Machines virtuelles et conteneurs ne sont pas incompatibles.
Il est possible d'exécuter des conteneurs dans des VM.



Les conteneurs

Qu'est ce qu'un conteneur ?



UNIVERSITÉ
CAEN
NORMANDIE



- Un conteneur dispose de **son propre système de fichiers** et permet l'exécution de processus dans **un environnement isolé et contraint**.
- **L'isolation est possible grâce aux espaces de noms**. Il s'agit d'une fonctionnalité du noyau Linux permettant à des ensembles de processus de ne voir qu'un ensemble de ressources.
- **Le contrôle de la consommation des ressources** (cpu, mémoire, réseau...) **est assuré par la librairie CGroups** du noyau Linux.

Les conteneurs

Usages et caractéristiques



UNIVERSITÉ
CAEN
NORMANDIE



- Les conteneurs ont des cas d'utilisation similaires aux machines virtuelles. Toutefois, dans la philosophie qui leur est associée, **les conteneurs ont généralement vocation à être éphémères, et si possible, sans état.**
- De par leur nature, les applications au sein de conteneurs sont **moins consommatrices de ressources** que des VM.
- **Les conteneurs peuvent très facilement migrer sur n'importe quelle machine**, du moment qu'elle dispose d'un OS de la même famille que la machine ayant permis de créer le conteneur.
- Ne devant pas amorcer un OS complet, **les conteneurs sont plus rapides à démarrer et éteindre que des machines virtuelles.** C'est une fonctionnalité intéressante dans les environnements de *cloud computing* afin de pratiquer de la scalabilité à la demande.
- En dehors des applications conteneurisées mises en production, on retrouve fréquemment les conteneurs dans les environnements de développement (instanciation de bdd, *broker...*), de qualification (afin de jouer des tests automatisés ou manuels) et sur les pipelines d'intégration et de déploiement continu.

Les conteneurs

Open Container Initiative (OCI)



UNIVERSITÉ
CAEN
NORMANDIE



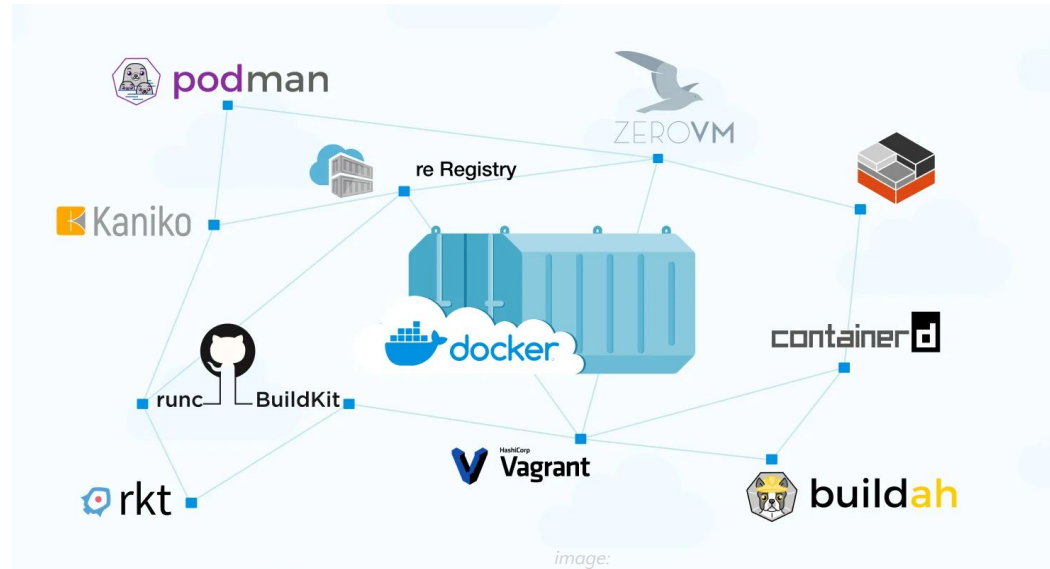
- L'Open Container Initiative est une structure de gouvernance ouverte dont **l'objectif est de créer des normes industrielles ouvertes** pour les formats et les moteurs d'exécution des conteneurs¹.
- 3 spécifications clés:
 - **image-spec**: définit ce qu'est une image OCI. Une image OCI est composée d'un manifeste d'image, d'un index d'image (facultatif), d'un ensemble de couches de système de fichiers et d'une configuration.
 - **runtime-spec**: vise à préciser la configuration, l'environnement d'exécution et le cycle de vie d'un conteneur.
 - **distribution-spec**: définit un protocole API pour faciliter et normaliser la distribution de contenu.

¹Définition traduite depuis le site de l'Open Container Initiative

Les conteneurs

Les outils sur le marché

- L'écosystème gravitant autour des conteneurs est riche. De nombreux logiciels existent sur le marché. Aujourd'hui, il est conseillé pour l'interopérabilité de choisir des logiciels conformes aux normes conçues par l'Open Container Initiative.
- Pour n'en citer que quelques-uns:
 - **Docker:** Plateforme de conteneurisation la plus utilisée
 - **LXD:** daemon pour la gestion de VM et de conteneurs sur Linux
 - **Buildah:** spécialisé dans la construction d'images OCI
 - **Podman:** moteur de conteneurs sans démon pour développer, gérer et exécuter des conteneurs OCI sur Linux



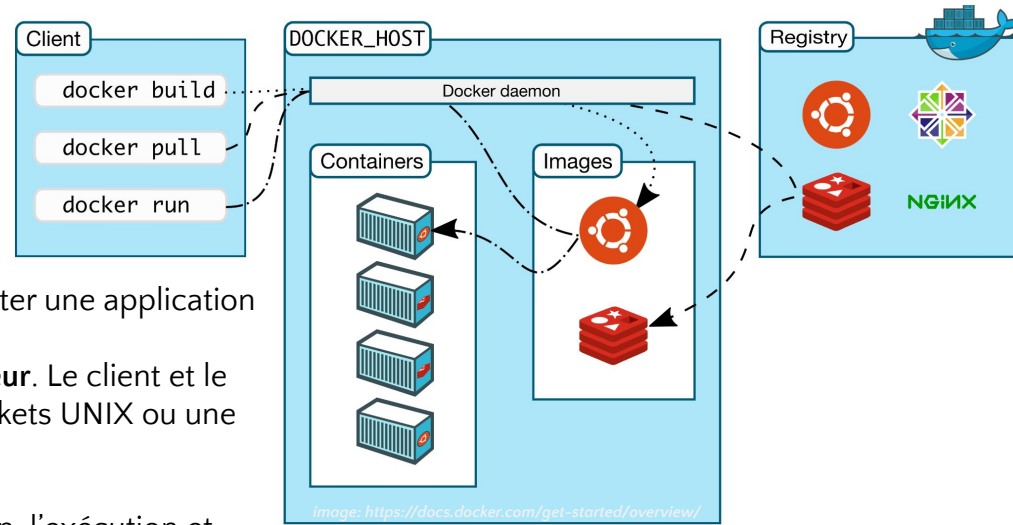
La plateforme Docker



La plateforme Docker

L'architecture de Docker

- Docker est une **plateforme ouverte** pour le développement, la livraison et l'exécution d'applications
- Le conteneur devient l'unité pour distribuer et tester une application
- Docker est basé sur une **architecture client-serveur**. Le client et le démon communiquent via une API REST, des sockets UNIX ou une interface réseau.
- Le démon Docker est en charge de la construction, l'exécution et la distribution des conteneurs.
- Il existe 2 principaux clients:
 - `docker cli` exposant un interface en ligne de commande
 - `docker-compose` permettant de définir et d'exécuter une application constituée d'un ensemble de conteneurs.
- Le registre est un serveur assurant le stockage et la distribution des images.



La plateforme Docker

Les innovations de Docker



UNIVERSITÉ
CAEN
NORMANDIE



- En adoptant **une structure en couches** (*Union FS*), Docker a optimisé la taille des conteneurs.
- Avec Docker, la persistance des données dans un conteneur est gérée via **la notion des volumes**.
- Le partage des conteneurs entre les différents systèmes est rendu possible au moyen **des registres et de la notion d'image**.

La plateforme Docker

client - l'interface de commandes docker



UNIVERSITÉ
CAEN
NORMANDIE



- Docker dispose d'une interface en ligne de commandes (*CLI*). C'est le moyen privilégié d'interagir avec le démon lorsque l'on travaille sur des conteneurs autonomes ou afin d'administrer les différents objets Docker (conteneurs, images, volumes, réseaux...)
- Les commandes suivent la syntaxe suivante: `docker <<objet>> <<commande>> <<options>>`
 - exemple pour afficher la liste des conteneurs: `docker container ls`
- La ligne de commande Docker est configurable via des variables d'environnement (`DOCKER_HOST`, `DOCKER_CONFIG`...)
- Pour l'utilisation, se référer à la documentation en ligne¹ et celle obtenue avec la commande `help`.

¹<https://docs.docker.com/engine/reference/commandline/cli/>

La plateforme Docker

Les objets fondamentaux - Image



UNIVERSITÉ
CAEN
NORMANDIE



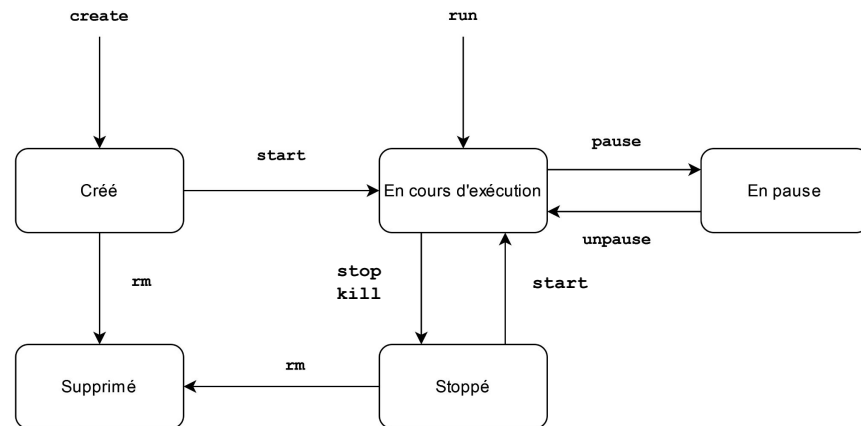
- Une image Docker est **un modèle en lecture seule contenant des instructions** pour créer un conteneur Docker.
- Les instructions d'une image sont de trois natures différentes:
 - **des modifications ordonnées** d'un système de fichiers
 - **des paramètres d'exécution** destinés à être utilisés lors de l'instanciation d'un conteneur
 - **des méta-données** (optionnelles) utiles notamment à des fins de documentation
- **Chaque instruction crée une nouvelle couche** dans l'image.
- **Une image est une méthode** afin de construire un conteneur de manière reproductible mais pas forcément déterministe.
- Vous pouvez construire vos propres images ou utiliser des images publiées par d'autres personnes sur un registre.

La plateforme Docker

Les objets fondamentaux - Conteneur



- **Un conteneur Docker est une instance exécutable d'une image Docker.**
- **Toutes les modifications apportées à un conteneur ne seront jamais répercutées sur son image de base.** Elles sont stockées dans la couche "conteneur" accessible en lecture/écriture.
- **Au démarrage d'un conteneur, des instructions préalablement définies dans l'image permettent d'exécuter une ou des commandes ;** qui peuvent, si l'utilisateur le souhaite, être surchargées au moment de l'instanciation du conteneur.
- **Un conteneur possède un cycle de vie** constitué de 5 états : créé, en cours, arrêté, en pause, supprimé



La plateforme Docker

Le Registre



UNIVERSITÉ
CAEN
NORMANDIE



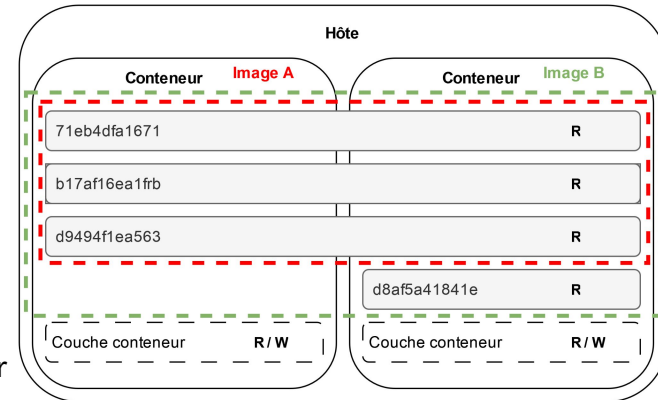
- Le registre est un composant essentiel dans l'architecture de Docker. Il s'agit d'un serveur assurant le stockage et la distribution des images.
- L'application en elle-même est open source sous licence Apache 2.0. Si vous souhaitez maîtriser de bout en bout votre chaîne de déploiement, il est possible de déployer votre propre registre.
- Le registre par défaut est Docker Hub. Il appartient à la société Docker et différentes offres de tarifications sont proposées. A ce jour la société Docker propose une offre "Personal" qui convient pour les projets open sources, les petites entreprises, et l'éducation. La limitation principale concerne le nombre d'images (200) que nous pouvons récupérer par tranche de 6 heures.
- En fonction du registre utilisé, il peut être nécessaire de s'authentifier pour déposer ou récupérer des images.
- Pour pointer vers un autre registre que Docker Hub, il faut préfixer votre image par l'adresse de votre registre. Par exemple:
 - **prof:5000/fr/iut/caen/mapremiereimage:latest**

La plateforme Docker

La structure en couches - Principe



- L'illustration ci-après montre, schématiquement, le concept des couches des images et conteneurs.
- Dans cet exemple, nous avons sur notre machine hôte 2 images:
 - Une image "A" constituée de 3 couches : 71e, b17, d94
 - Une image "B" constituée de 4 couches : 71e, b17, d94, d8a
- On constate que les couches 71e, b17, d94 sont communes entre les images.
- Un des atouts de Docker est de pouvoir partager ces couches communes entre des images et les conteneurs. Toutes les modifications d'un conteneur seront effectuées dans sa "couche conteneur".
- Une couche est identifiée grâce à un condensé (*digest*). Il s'agit d'une valeur générée par une fonction de hachage.



La plateforme Docker

Le Dockerfile



- Pour construire une image, il existe deux méthodes:
 - Le commit d'un conteneur existant. Cela va sauvegarder les modifications de la couche conteneur dans une nouvelle image. Cette méthode est plutôt conseillée à des fins de débogage.
 - **L'écriture d'un fichier en utilisant un langage dédié (DSL). Ce fichier se nomme le Dockerfile.**
- Un Dockerfile est simplement une suite d'instructions. Chaque ligne débute par l'intitulé de l'instruction et chacune va correspondre à une nouvelle couche dans l'image.
- Pour passer du Dockerfile à une image exploitable, il y a une étape de construction (*build*) qui consiste à répéter l'opération suivante:
 - créer un conteneur temporaire à partir de la couche précédente (ou de l'instruction **FROM**)
 - Exécuter l'instruction à l'intérieur de ce conteneur temporaire
 - Sauvegarder les modifications du système de fichier dans une nouvelle couche

```
1 FROM ubuntu:latest
2
3 RUN apt update && apt install -y nano
4
5 CMD ["sh"]
```

Exemple de Dockerfile

- Vous pouvez spécifier quel programme sera exécuté au démarrage d'un conteneur à l'aide des instructions `ENTRYPOINT` et `CMD`.
- La principale différence c'est que l'instruction `CMD` est ignorée si des paramètres sont passés à l'instanciation d'un conteneur alors qu'ils seront traités comme des arguments avec l'instruction `ENTRYPOINT`.
- Ces instructions peuvent être écrites sous deux formes :
 - `exec`:
 - `<INSTRUCTION> [<COMMANDE>, <PARAMETRES>]`
 - exemple: `CMD ["echo", "hello-world"]`
 - `shell`:
 - `<INSTRUCTION> <COMMANDE> <PARAMETRES>`
 - exemple: `ENTRYPOINT echo "hello-world"`
- Avec la forme `shell`, il est possible de faire des substitutions de commandes, d'utiliser des variables d'environnement ou encore de rediriger des entrées sorties. Ce n'est pas possible avec la forme `exec`.
- Il est conseillé de privilégier la forme `exec` car elle permet de prendre en compte des arguments passés en paramètres. Généralement, on utilise `ENTRYPOINT` pour lancer un programme et `CMD` pour spécifier des paramètres par défaut pouvant être surchargés par le client.

La plateforme Docker

Le cache



- Précédemment, nous avons vu que chaque instruction d'un Dockerfile était exécutée dans un conteneur temporaire afin d'ajouter une nouvelle couche à l'image en cours de construction.
- Un mécanisme de cache existe afin de raccourcir les temps de reconstruction d'une image. Tant que les instructions¹ du Dockerfile ne sont pas modifiées, les images intermédiaires construites précédemment seront ré-utilisées.
- Si une couche change, toutes les couches en aval doivent être reconstruites également.
- Pour tirer parti efficacement du cache il faut privilégier les instructions les plus génériques en haut du Dockerfile.
- N.B: Il est possible de désactiver le cache lors de la construction d'une image.

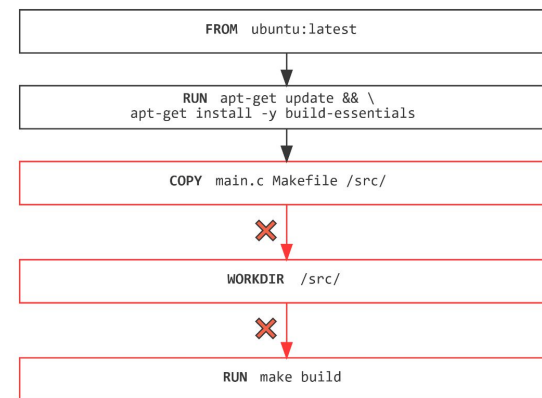


image: <https://docs.docker.com/build/cache/>

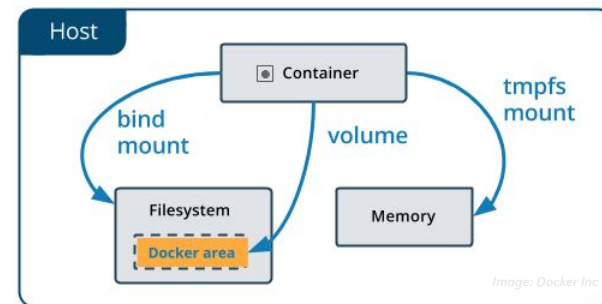
¹pour les instructions COPY et ADD, la date de dernière modification est également prise en compte.

La plateforme Docker

La gestion du stockage des données



- **Par défaut, la persistance des données s'effectue dans la couche conteneur.** Ce qui induit quelques limitations:
 - Si le conteneur est supprimé, les données sont perdues.
 - La couche conteneur est étroitement liée à l'hôte.
 - L'écriture dans la couche conteneur passe par une couche d'abstraction. Les performances sont donc moindre qu'une écriture directe sur le système de fichier de l'hôte
- Il existe 3 méthodes pour écrire en dehors de la couche conteneur:
 - Volumes: un objet Docker appartenant aux systèmes de fichiers de l'hôte, créé et géré par Docker.
 - Montages liés (*bind mounts*): Similaire aux volumes mais avec des fonctionnalités limités
 - Montages temporaires (*tmpfs mounts*): Persistance en mémoire ne pouvant être partagée entre plusieurs conteneurs. Utile pour enregistrer des états non persistants ou des informations sensibles.



La plateforme Docker

Les volumes



UNIVERSITÉ
CAEN
NORMANDIE



- Les volumes sont des objets Docker stockés dans un répertoire sur la machine hôte (sur Linux `/var/lib/docker/volumes`).
- Lorsqu'un volume est monté dans un conteneur, cela monte un répertoire dans ce conteneur.
- Plusieurs conteneurs peuvent monter simultanément le même volume.
- Lors de la suppression d'un conteneur, le volume qui lui est associé n'est pas supprimé.
- Un volume possède un nom défini explicitement ou il est généré aléatoirement par défaut.
- Il est possible d'utiliser des pilotes pour les volumes, par exemple REX-Ray. Ces derniers permettent de stocker les données sur des hôtes distants ou des fournisseurs de services *cloud*.

La plateforme Docker

Les montages liés (*bind mounts*)



UNIVERSITÉ
CAEN
NORMANDIE



- Un montage lié consiste à monter un fichier ou un répertoire de l'hôte à l'intérieur d'un conteneur.
- Les fonctionnalités sont similaires aux volumes à quelques détails près:
 - **Le chemin vers le fichier ou le répertoire est obligatoirement un chemin absolu.** Cela implique que le système de fichiers de la machine hôte possède une certaine structure.
 - **Les performances en écriture comme en lecture sont meilleures** que dans la couche conteneur ou un volume.
 - **Attention, avec ce type de montage, on assiste à une diminution de l'isolation du conteneur.** Le conteneur peut venir modifier le système de fichiers de l'hôte. Cela peut-être l'objectif recherché, mais cela expose l'hôte et les autres processus de la machine à un risque potentiel.
 - Il n'est pas possible d'administrer les montages liés via l'interface en ligne de commande de Docker.

La plateforme Docker

Les pilotes réseau



UNIVERSITÉ
CAEN
NORMANDIE



- Les conteneurs doivent être conçus pour traiter une tâche spécifique. Pour tirer pleinement parti de leur potentiel, ils doivent coopérer entre eux ou avec des services tiers. Nous allons nous intéresser aux fonctionnalités réseau de Docker.
- Docker gère la partie réseau de manière agnostique, peu importe que les conteneurs proviennent d'hôtes sous des OS Linux, Windows ou les deux.
- Sans plugins tiers, Docker est livré avec 6 pilotes réseau, voici un aperçu des plus utilisés:
 - **bridge**: le pilote par défaut. A privilégier lorsque plusieurs conteneurs sur un même hôte doivent communiquer entre eux.
 - **host**: supprime l'isolation du réseau entre le conteneur et l'hôte.
 - **overlay**: connecte plusieurs démons Docker entre eux. Idéal lorsque plusieurs conteneurs sur des hôtes différents doivent communiquer entre eux.
 - **none**: Désactive les fonctionnalités réseaux (excepté l'adresse de loopback)
 - **macvlan**: permet d'assigner une adresse MAC à un conteneur. Utile pour faire apparaître le conteneur comme un hôte physique sur le réseau.
- Au démarrage de Docker, un réseau de type "bridge" est créé. A moins de spécifier le contraire, tous les conteneurs créés seront rattachés par défaut à ce réseau. Ils pourront communiquer entre eux via leur adresse IP.

La plateforme Docker

Le réseau depuis un conteneur



UNIVERSITÉ
CAEN
NORMANDIE



- Le type de réseau utilisé (bridge, overlay, macvlan) est transparent depuis le conteneur.
- Le conteneur ne possède à son niveau qu'une interface réseau dont les principaux éléments sont:
 - une adresse IP
 - une passerelle
 - un table de routage
 - un service DNS
- **Par défaut, aucun port n'est exposé.** Pour rendre accessible le port d'un conteneur à des services en dehors de Docker ou à des conteneurs qui ne sont pas connectés au même réseau ; il est possible de mapper un port de l'hôte sur un port du conteneur.
- Le démon Docker se comporte comme un serveur DHCP en attribuant une IP à chaque conteneur se connectant à un réseau.
- Les conteneurs disposant de la configuration réseau par défaut hérite d'une copie du fichier `/etc/resolv.conf` Alors que les conteneurs avec un réseau spécifique utilisent le serveur DNS embarqué de Docker.

La plateforme Docker

client - Compose

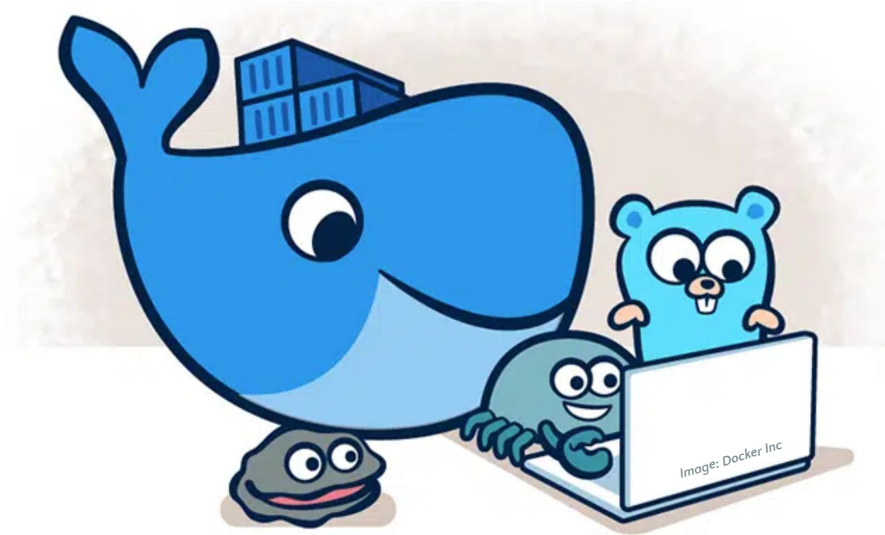


- Compose est un client s'appuyant sur un fichier de configuration dans laquelle toutes les dépendances (images, réseaux, volumes, variables d'environnement...) d'une application multi-conteneurs sont définies.
- **Ce client est le logiciel d'orchestration de la plateforme Docker.** Il existe des alternatives, par exemple Kubernetes qui est un projet open-source initié par Google en 2015. Mais il est très fréquent de rencontrer cette solution dans des projets grâce à sa simplicité de mise en œuvre.

```
1 version: "3"
2
3 services:
4   bdd:
5     image: 'postgres:15.1'
6     environment:
7       - POSTGRES_USER: 'myuser'
8       - POSTGRES_PASSWORD: 'secret1234'
9     ports:
10      - "5432:5432"
11   frontend:
12     build: ./webapp
13     image: sample/webapp
14     ports:
15       - "80:80"
16     environment:
17       - BACKEND_URL: http://backend:8080
18   backend:
19     image: mybackend:latest
20     ports:
21       - "8080:8080"
22
```

Exemple de fichier docker-compose.yml

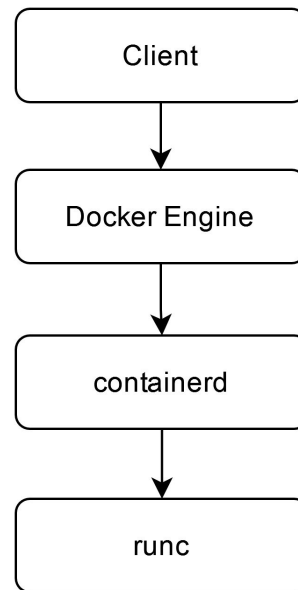
Le socle technologique de Docker



Le socle technologique de Docker

Linux

- Les conteneurs s'appuient sur deux principaux services exposés par le noyau Linux:
 - Les groupes de contrôle (*Cgroups*)
 - Les espaces de noms (*Namespaces*)
- Docker utilise le **démon containerd** afin de gérer l'ensemble du cycle de vie d'un conteneur. Du transfert de son image depuis un registre jusqu'à l'exécution d'un conteneur et la supervision bas-niveau de son stockage et du réseau.
- containerd est disponible sous Linux et Windows et **permet d'abstraire les appels systèmes ou fonctionnalités spécifiques** d'un système d'exploitation.
- **runC est le programme responsable du contexte d'exécution d'un conteneur.** Docker l'utilise afin de créer les conteneurs, mettre en place les points de montages...



Le socle technologique de Docker

Les groupes de contrôle



UNIVERSITÉ
CAEN
NORMANDIE



- *CGroups* est une fonctionnalité du noyau Linux permettant de quantifier l'utilisation des ressources d'une machine par des processus regroupés dans des groupes hiérarchisés.
- Il est ainsi possible de prioriser, limiter ou contrôler l'utilisation de ressources sur chacun des groupes.
- Qu'est-ce qu'une ressource ? Du temps CPU, de la bande passante, une quantité de mémoire vive...
- Les groupes de contrôle sont constitués de différents sous-systèmes, en voici quelques-uns:
 - freezer: démarrage et arrêt d'un ensemble de tâches.
 - memory: gestion de l'usage de la ram et du swap
 - pids: mise en place de limitation vis-à-vis du nombre de processus...
 - ...

Le socle technologique de Docker

Les espaces de noms



UNIVERSITÉ
CAEN
NORMANDIE



- Les *namespaces* permettent de dupliquer certaines structures, habituellement considérées uniques pour le noyau, dans le but de les isoler d'un groupe de processus à un autre.
- On dénombre 8 espaces de noms: cgroup, IPC, network, mount, PID, time, user et UTS.
- La notion d'espace de noms est relativement nouvelle et a été intégrée progressivement au sein du noyau Linux. Aussi, toutes les structures ne sont pas encore conteneurisables (exemple: journaux d'événements, ou module de sécurité).
- Le noyau tient à jour un compteur de références pour chaque *namespace*. Un namespace est libéré dès qu'il n'est plus référencé par un processus.

Le socle technologique de Docker

Les espaces de noms



UNIVERSITÉ
CAEN
NORMANDIE



- **L'espace de noms** `mount`: isole la liste des points de montage
- **L'espace de noms** `UTS`: isole le nom d'hôte de la machine et son domaine NIS
- **L'espace de noms** `PID`: isole la liste des processus et virtualise leurs numéros
- **L'espace de noms** `IPC`: isole les objets IPC et les files de messages POSIX
- **L'espace de noms** `network`: isole toutes les ressources associées aux réseaux
- **L'espace de noms** `user`: isole la liste des utilisateurs, des groupes, leurs identifiants, les *capabilities*, la racine et le trousseau de clefs du noyau
- **L'espace de noms** `cgroup`: Filtre l'arborescence des groupes de contrôle en changeant la racine vue par le processus.

Référence: Chapitre 23. Les espaces de noms - Pierre-Olivier MERCIER.
Conteneurs et technologies du DevOps

Le socle technologique de Docker

Union Filesystem



UNIVERSITÉ
CAEN
NORMANDIE



- *Union Mount* est un service du système de fichiers qui permet de fusionner virtuellement le contenu de plusieurs points de montage en un seul sans pouvoir modifier les points de montage initiaux.
- Il existe plusieurs implémentations dont OverlayFS qui est le driver par défaut pour Docker.
- Les points de montage sont qualifiés de branches. Chaque branche se voit attribuer une priorité. Elle sera utilisée pour déterminer quel fichier devra apparaître dans la vue fusionnée dans le cas où il y aurait des fichiers avec le même nom dans plusieurs branches sources.
- **Les branches en priorité faible seront en lecture seule et la branche avec la priorité la plus élevée sera en lecture/écriture.**
- Lorsqu'une modification doit être effectuée depuis la vue fusionnée sur un fichier dans une branche en lecture seule. Alors ce fichier sera copié depuis la branche en lecture seule vers la branche ayant la plus haute priorité (celle en lecture/écriture). Cette technique d'optimisation est appelée le **copy-on-write (CoW)**.

Conclusion



UNIVERSITÉ
CAEN
NORMANDIE



- Au cours de ces deux dernières séances vous avez découvert la conteneurisation ainsi que ses principaux cas d'usage.
- C'est une technologie qui s'inscrit parfaitement dans le *Cloud computing* et qui est très implantée dans l'industrie.
- Vous avez été initié à l'utilisation de Docker, la plateforme de conteneurisation la plus utilisée à ce jour. N'hésitez pas à en tirer profit lors de vos prochains projets à l'IUT (en particulier lorsque vous devez mettre en place une base de données sur votre poste, c'est très pratique !)
- Nous avons vu que l'isolation d'un conteneur était "à la carte" (système de fichiers, réseaux, utilisateurs) au contraire des machines virtuelles qui imposent une isolation forte par le biais du déploiement d'un OS complet.
- Étant donné le temps imparti, nous avons survolé certains sujets, notamment en ce qui concerne l'orchestration et la sécurité. Ces derniers pourront sûrement être adressés plus en profondeur lors du module de virtualisation avancé au semestre 5.

MERCIER Pierre-Olivier. *Conteneurs et technologies du DevOps*. Gentilly: Editions ALPO, 2022. 209 p.

CLOUX Pierre-Yves, GARLOT Thomas, KOHLER Johann. *Docker et conteneurs: architectures, développement, usages et outils*. 3e édition. Malakoff: Dunod, 2022. 317 p.

Docker Inc. *Docker Documentation [en ligne]*. 2023. [Consulté le 6 janvier 2023]. Disponible à l'adresse: <https://docs.docker.com/>

BARJONET Claude. *Le monde mis en boîte ou l'histoire du conteneur [en ligne]*. Publié le 4 septembre 2014. [Consulté le 6 janvier 2023]. Disponible à l'adresse: <https://www.lesechos.fr/2014/09/le-monde-mis-en-boite-ou-l-histoire-du-conteneur-308918>

The Linux Foundation. *Open Container Initiative [en ligne]*. 2023. [Consulté le 13 janvier 2023]. Disponible à l'adresse: <https://opencontainers.org>

HEINZ Martin. *Deep Dive into Docker Internals - Union Filesystem [en ligne]*. Publié le 1 mars 2021. [Consulté le 15 janvier 2023]. Disponible à l'adresse: <https://martinheinz.dev/blog/44>

Merci de votre attention

Maxime Lambert

maxime.lambert@unicaen.fr



UNIVERSITÉ
CAEN
NORMANDIE



GRAND OUEST
NORMANDIE

