

Gérer les données dans Docker

Prérequis :

- ✓ Vous avez installé Docker et le logiciel fonctionne correctement.
- ✓ Vous connaissez les fonctionnalités et les commandes de base de Docker.

Source : <https://docs.docker.com/storage/storagedriver/select-storage-driver/>

Introduction

Le système de fichiers Docker

Pour comprendre les volumes Docker, il est essentiel de savoir comment fonctionne le système de fichiers Docker.

- ✓ Une image Docker est constituée de plusieurs couches,
- ✓ chacune d'elles en lecture seule.

Pour toute image lancée depuis un conteneur, Docker ajoute une nouvelle couche inscriptible au système.

Dans Docker, ce service est connu sous le nom d'« Union File System ».

Union File System :

Union File System



All layers except the topmost are readonly

Le système de fichiers Docker

Chaque fois qu'un fichier est modifié :

- ✓ Docker crée une copie de celui-ci à partir des couches en lecture seule,
 - ✓ puis l'ajoute à la couche inscriptible supérieure.
- Le fichier original (en lecture seule) n'est donc pas modifié.

Par défaut, tous les fichiers créés dans un conteneur sont stockés sur une couche conteneur inscriptible.

Cela signifie que :

- ✓ Les données ne persistent pas lorsque ce conteneur n'existe plus,
- ✓ et il peut être difficile d'extraire les données du conteneur si un autre processus en a besoin.

La couche inscriptible d'un conteneur est étroitement couplée à la machine hôte sur laquelle le conteneur est exécuté.

- ✓ Vous ne pouvez pas facilement déplacer les données ailleurs.

L'écriture dans la couche inscriptible d'un conteneur nécessite un pilote de stockage pour gérer le système de fichiers.

➤ Le pilote de stockage fournit un système de fichiers Union, utilisant le noyau Linux.

Cette abstraction supplémentaire réduit les performances par rapport à l'utilisation de volumes de données :

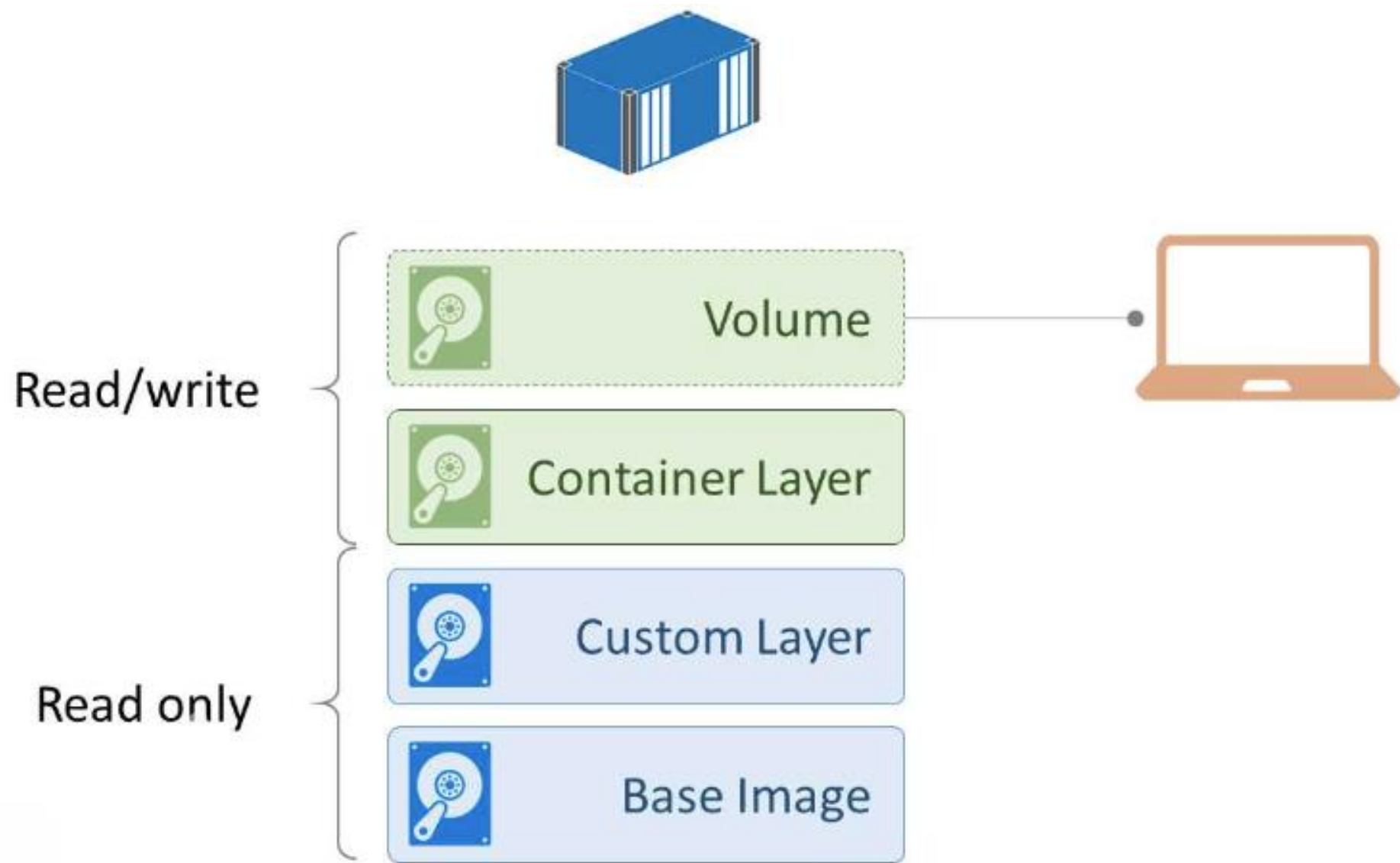
✓ En effet ces derniers écrivent directement sur le système de fichiers hôte.

Résoudre les problèmes grâce aux volumes

Un volume de conteneur vous permet de conserver vos données,

➤ même en cas de suppression du conteneur Docker.

Il s'agit également d'une solution pratique pour l'échange de données entre l'hôte et le conteneur.



Cas d'utilisation d'un volume de conteneur Docker :

- ✓ transférer des données vers un conteneur Docker ;
- ✓ enregistrer des données depuis un conteneur Docker ;
- ✓ partager des données entre vos conteneurs Docker.

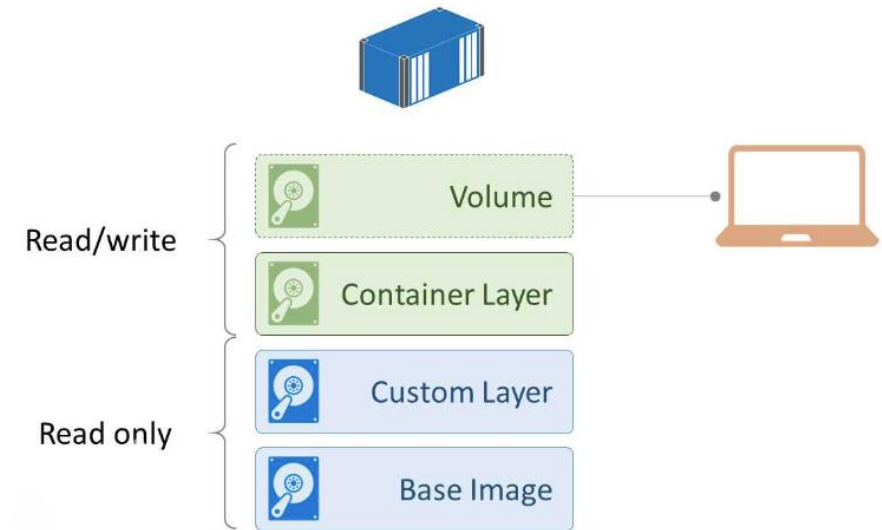
Les volumes Docker n'appartiennent pas à l'« Union File System » (avec son accès en lecture seule et sa couche inscriptible).

- ✓ Un volume consiste plutôt en un dossier que se partagent le conteneur et l'ordinateur hôte.
- ✓ Plusieurs conteneurs peuvent également se partager un même volume.

Union File System



All layers except the topmost are readonly



Types de volumes

Docker propose deux options permettant aux conteneurs de stocker des fichiers sur la machine hôte, de sorte que les fichiers soient conservés même après l'arrêt du conteneur :

- ✓ les volumes
- ✓ et les montages de liaison (Bind Mounts).

Docker prend également en charge les conteneurs stockant des fichiers en mémoire (RAM) sur la machine hôte.

- Ces fichiers ne sont pas conservés.

Si vous exécutez Docker sous Linux :

- ✓ tmpfs mount est utilisé pour stocker les fichiers dans la mémoire système de l'hôte.

Si vous exécutez Docker sous Windows, :

- ✓ le canal nommé (Named Pipe) est utilisé pour stocker les fichiers dans la mémoire système de l'hôte.

Choisissez le bon type de support

Quel que soit le type de montage que vous choisissiez d'utiliser :

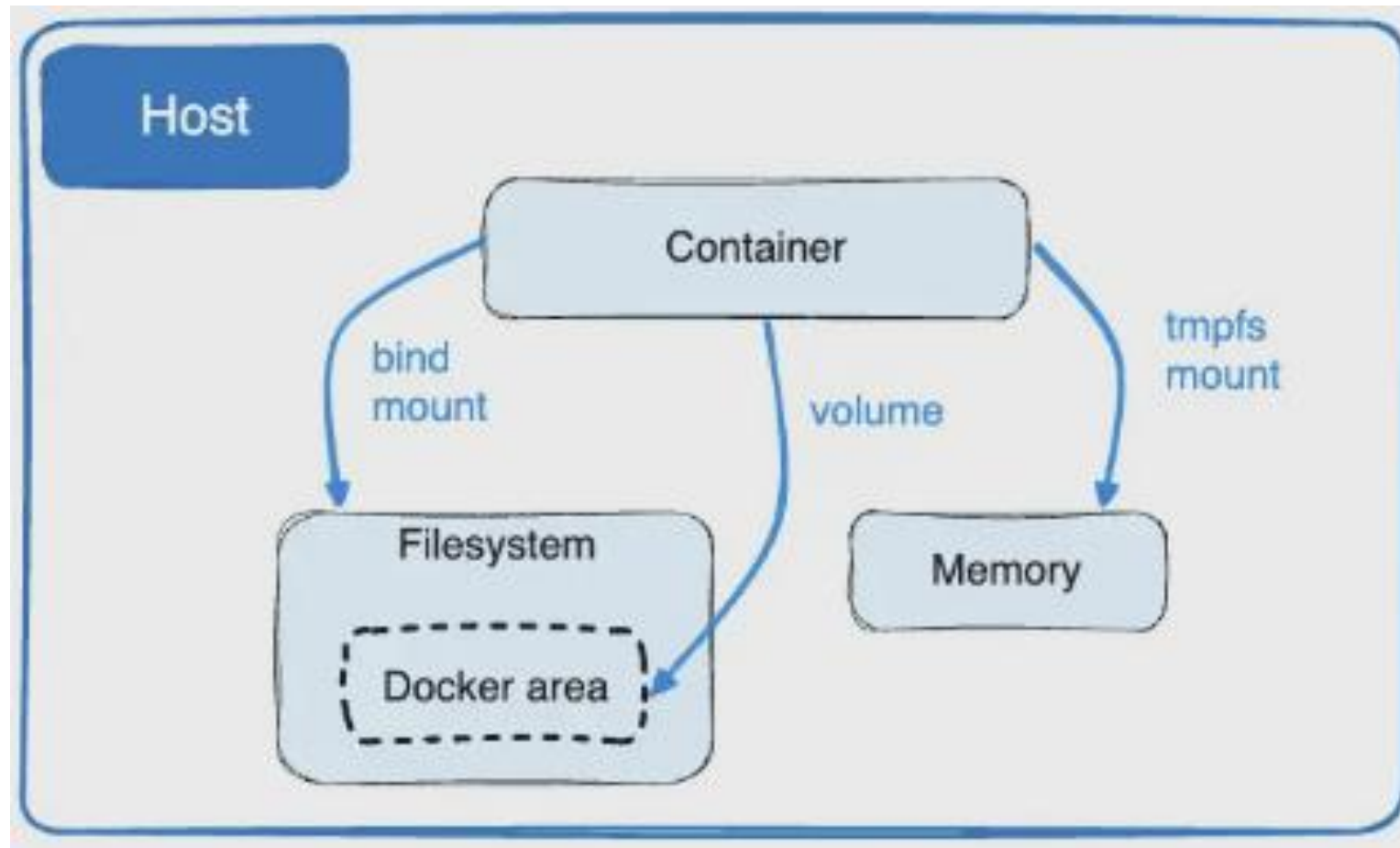
- ✓ les données sont identiques depuis l'intérieur du conteneur.

Il est exposé sous forme :

- de répertoire,
- ou de fichier individuel

dans le système de fichiers du conteneur.

Un moyen simple de visualiser la différence entre les volumes, les montages de liaison et les montages tmpfs consiste à réfléchir à l'emplacement des données sur l'hôte Docker.



Les volumes :

Les volumes sont :

- ✓ stockés dans une partie du système de fichiers hôte,
- ✓ gérés par Docker (/var/lib/docker/volumes/ sous Linux).

Les processus non Docker ne doivent pas modifier cette partie du système de fichiers.

- Les volumes constituent le meilleur moyen de conserver les données dans Docker.

Les Bind Mounts :

Les montages de liaison peuvent être stockés n'importe où sur le système hôte.

- Il peut même s'agir de fichiers ou de répertoires système importants.

Les processus non Docker sur l'hôte Docker ou un conteneur Docker :

- ✓ peuvent les modifier à tout moment.

Les montages tmpfs :

Les montages tmpfs sont :

- ✓ stockés uniquement dans la mémoire du système hôte
- ✓ et ne sont jamais écrits dans le système de fichiers du système hôte.

Plus de détails sur les types de montage

Volumes :

- créés et gérés par Docker.

Vous pouvez créer un volume :

- ✓ explicitement à l'aide de la commande docker

```
docker volume create monvolume
```

- ✓ ou Docker peut créer un volume lors de la création d'un conteneur ou d'un service.

Volumes :

Lorsque vous créez un volume :

- ✓ il est stocké dans un répertoire sur l'hôte Docker.
- ✓ ce répertoire est monté dans le conteneur.

C'est similaire à la façon dont fonctionnent les montages de liaison, sauf que :

- ✓ les volumes sont gérés par Docker
- ✓ et sont isolés des fonctionnalités de base de la machine hôte.

Volumes :

- ✓ Un volume donné peut être monté simultanément dans plusieurs conteneurs.

Lorsqu'aucun conteneur en cours d'exécution n'utilise un volume :

- ✓ le volume est toujours disponible pour Docker
- ✓ et n'est pas supprimé automatiquement.

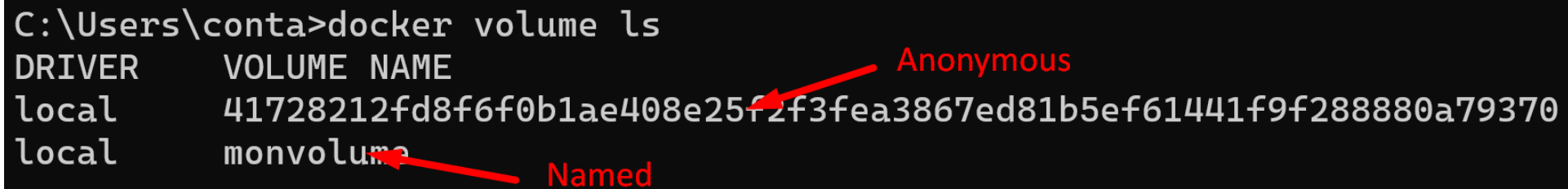
Vous pouvez supprimer les volumes inutilisés à l'aide de la commande :

```
docker volume prune
```

Volumes :

- ✓ Lorsque vous montez un volume, il peut être nommé ou anonyme.

```
C:\Users\conta>docker volume ls
DRIVER      VOLUME NAME
local       41728212fd8f6f0b1ae408e25f2f3fea3867ed81b5ef61441f9f288880a79370
local       monvolume
```



Les volumes anonymes ne reçoivent pas de nom explicite lorsqu'ils sont montés pour la première fois dans un conteneur :

- donc Docker leur donne un nom aléatoire dont l'unicité est garantie au sein d'un hôte Docker donné.

Outre le nom, les volumes nommés et anonymes se comportent de la même manière.

Volumes :

Les volumes prennent également en charge l'utilisation de pilotes (driver) de volume, qui vous permettent, entre autres possibilités, de stocker vos données sur des hôtes distants ou des fournisseurs de cloud.

Linux distribution	Recommended storage drivers	Alternative drivers
Ubuntu	overlay2	devicemapper ¹ , zfs, vfs
Debian	overlay2	devicemapper ¹ , vfs
CentOS	overlay2	devicemapper ¹ , zfs, vfs
Fedora	overlay2	devicemapper ¹ , zfs, vfs
SLES 15	overlay2	devicemapper ¹ , vfs
RHEL	overlay2	devicemapper ¹ , vfs

Bind Mounts :

- Disponibles depuis les débuts de Docker :

les supports de liaison ont des fonctionnalités limitées par rapport aux volumes.

Lorsque vous utilisez un montage lié :

- ✓ un fichier ou un répertoire sur la machine hôte est monté dans un conteneur.
- ✓ Le fichier ou le répertoire est référencé par son chemin complet sur la machine hôte.

Il n'est pas nécessaire que le fichier ou le répertoire existe déjà sur l'hôte Docker :

- ✓ Il est créé à la demande s'il n'existe pas encore.

Bind Mounts :

- Les montages Bind sont très performants,
- mais ils reposent sur le système de fichiers de la machine hôte disposant d'une structure de fichiers spécifique (cf. Windows).

Si vous développez de nouvelles applications Docker, envisagez plutôt d'utiliser des volumes nommés.

- ✓ Vous ne pouvez pas utiliser les commandes Docker CLI pour gérer directement les montages de liaison.

Bind Mounts – mise en garde :

- Les montages de liaison permettent d'accéder aux fichiers sensibles.

L'un des effets secondaires de l'utilisation des montages liés, pour le meilleur ou pour le pire, est que vous pouvez modifier le système de fichiers hôte via des processus exécutés dans un conteneur, notamment :

- ✓ la création,
- ✓ la modification
- ✓ ou la suppression

de fichiers ou de répertoires système importants.

Il s'agit d'une capacité puissante qui peut avoir des implications en matière de sécurité, notamment en affectant les processus non Docker sur le système hôte.

tmpfs mounts :

Un montage tmpfs n'est pas conservé sur le disque :

- ✓ que ce soit sur l'hôte Docker
- ✓ ou dans un conteneur.

Il peut être utilisé par un conteneur :

- ✓ pendant la durée de vie du conteneur,
- ✓ pour stocker un état non persistant
- ✓ ou des informations sensibles.

Named pipes :

un montage npipe peut être utilisé pour la communication :

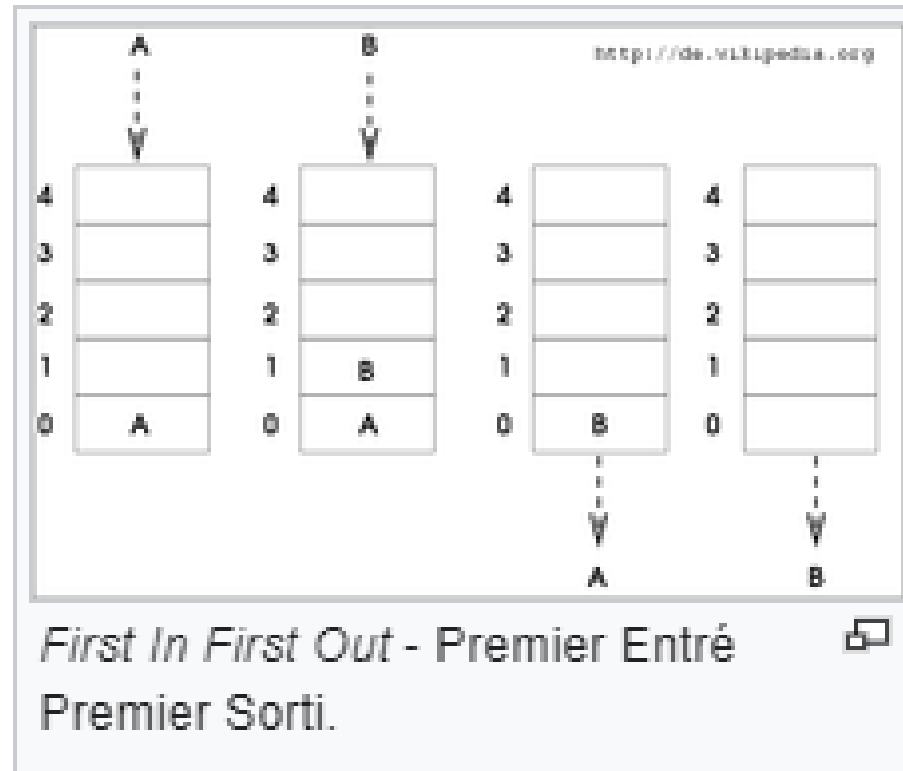
- ✓ entre l'hôte Docker et un conteneur.
- ✓ Entre deux ou plusieurs conteneurs.

Il s'agit d'un un mécanisme inter-process communication (IPC) ou FIFO utilisé pour transmettre des messages entre les processus d'un système.

En clair :

faire circuler des flux d'informations entre les processus émetteurs et récepteurs.

FIFO (First in, first out) :



Named pipes - exemple :

1^{er} terminal :

```
C:\Users\conta>docker volume create fifo-ipc-exemple
```

```
C:\Users\conta>docker run --rm -it -v fifo-ipc-exemple:/var/run/fifo-ipc-exemple debian:11-slim mkfifo /var/run/fifo-ipc-exemple/message
```

```
C:\Users\conta>docker run --rm -it -v fifo-ipc-exemple:/var/run/fifo-ipc-exemple debian:11-slim cat /var/run/fifo-ipc-exemple/message
```

2^{ème} terminal :

```
C:\Users\conta>docker run --rm -it -v fifo-ipc-exemple:/var/run/fifo-ipc-exemple debian:11-slim bash -c "echo 'Bonjour.' > /var/run/fifo-ipc-exemple/message"
```

Résultat (sur le 1^{er} terminal) :

```
C:\Users\conta>docker run --rm -it -v fifo-ipc-exemple:/var/run/fifo-ipc-exemple debian:11-slim cat /var/run/fifo-ipc-exemple/message  
Bonjour.
```

Les montages et les volumes de liaison peuvent tous deux être montés dans des conteneurs à l'aide de l'indicateur `-v` ou `--volume`,

➤ mais la syntaxe de chacun est légèrement différente.

Pour les montages tmpfs, vous pouvez utiliser l'indicateur `--tmpfs`.

Il est recommandé d'utiliser l'indicateur `--mount` pour :

- ✓ les conteneurs et les services,
- ✓ les montages de liaison,
- ✓ les volumes
- ✓ ou les montages tmpfs,

car la syntaxe est plus claire.

Les cas d'utilisation

Les volumes constituent le moyen privilégié pour conserver les données dans les conteneurs et les services Docker.

Exemples de cas d'utilisation des volumes :

- ✓ Partage de données entre plusieurs conteneurs en cours d'exécution.
- ✓ Lorsqu'il n'est pas garanti que l'hôte Docker ait une structure de répertoires ou de fichiers.
- ✓ Lorsque vous souhaitez stocker les données de votre conteneur sur un hôte distant ou un fournisseur cloud, plutôt que localement.
- ✓ Lorsque vous devez sauvegarder, restaurer ou migrer des données d'un hôte Docker à un autre.
- ✓ Lorsque votre application nécessite des I/O hautes performances.
- ✓ Lorsque votre application nécessite un comportement de système de fichiers entièrement natif (ex: moteur de base de données).

Les montages de liaison (Bind Mounts) sont appropriés pour les types de cas d'utilisation suivants :

- ✓ Partager des fichiers de configuration de la machine hôte vers des conteneurs.
- ✓ Partager du code source ou créer des liens entre un environnement de développement sur l'hôte Docker et un conteneur.
- ✓ Lorsque la structure des fichiers ou des répertoires de l'hôte Docker est garantie d'être cohérente avec les montages de liaison requis par les conteneurs.

Les montages tmpfs sont à utilisés dans les cas où :

- ✓ vous ne souhaitez pas que les données soient conservées sur la machine hôte ou dans le conteneur.

Cela peut être pour des raisons de sécurité ou pour protéger les performances du conteneur lorsque votre application doit écrire un grand volume de données d'état non persistantes.

**Tips (Conseils)
pour l'utilisation de montages
ou de volumes de liaison**

Si vous utilisez des montages ou des volumes de liaison, gardez les points suivants à l'esprit :

- ✓ Si vous montez un volume vide dans un répertoire du conteneur dans lequel existent des fichiers ou des répertoires :

ces fichiers ou répertoires sont propagés (copiés) dans le volume.

- ✓ Si vous démarrez un conteneur et spécifiez un volume qui n'existe pas déjà :

un volume vide est créé pour vous.

- C'est un bon moyen de préremplir les données dont un autre conteneur a besoin.

Questions ??

Prochain chapitre: Les volumes.