

# Les volumes

## Utilisation des volumes

Les volumes sont des répertoires accessibles depuis le conteneur et l'hôte.

**Les volumes sont persistants**, contrairement aux fichiers des conteneurs docker.

Les volumes sont très utiles avec Docker, ils peuvent servir à stocker, par exemple :

- Les données d'une base de données : il existe des images Docker pour la plupart des SGBD
- Les fichiers de configuration d'une application
- Les fichiers utilisateurs d'une application web (uploads)
- etc.

Une application dockerisée correctement configurée doit stocker ses données dans des volumes.

L'avantage des volumes est la sauvegarde : seules les données **utiles** s'y trouvent et il est possible de dupliquer une application à partir d'une image docker et des volumes associés.

Un volume est accessible depuis le container à partir de son point de montage. Exemple :

```
debian@debian:~$ pwd
/home/debian
debian@debian:~$ mkdir test && echo "Les volumes Docker c'est pratique !" > test/contenu.txt
debian@debian:~$ docker run -ti -v /home/debian/test:/toto ubuntu
root@2b038b1f509b:/# cd /toto
root@2b038b1f509b:/toto# cat contenu.txt
Les volumes Docker c'est pratique !
root@2b038b1f509b:/toto# echo "Les volumes Docker c'est super !" > contenu.txt
root@2b038b1f509b:/toto# exit
exit
debian@debian:~$ cat test/contenu.txt
Les volumes Docker c'est super !
debian@debian:~$
```

Le dossier `/home/debian/test` de l'hôte est monté dans `/toto` au sein du conteneur avec le paramètre `-v`. L'hôte et le conteneur ont accès au répertoire.

```
-v /home/debian/test:/toto
```

Généralement, les images Docker d'applications indiquent les répertoires dans lesquels elles stockent leurs données utiles, par exemple :

- **MariaDB** stocke les fichiers de bases de données dans `/var/lib/mysql` ([documentation](#)).
- **Odoo** stocke des données dans `/var/lib/odoo` ([documentation](#)).
- **Redmine** stocke les fichiers uploadés dans `/usr/src/redmine/files` ([documentation](#)).

## Volumes mappés vs. Volumes managés

Il existe deux types de volumes Docker : mappés et managés. Ils diffèrent par leur mode de gestion et leur usage :

- **Les volumes mappés** relient un dossier de l'hôte à un dossier du conteneur, ce qui permet un accès direct aux fichiers et facilite le développement. Les fichiers dépendent du système de fichiers de l'hôte, ce qui réduit la portabilité et peut poser des problèmes en production.
- **Les volumes managés** sont créés et gérés par Docker. Ils sont stockés dans un emplacement interne et peuvent être utilisés par plusieurs conteneurs. Ils offrent une meilleure isolation et une meilleure portabilité. Leur usage est idéal pour la persistance des données (bases de données, application, etc.), mais ils nécessitent des commandes Docker pour y accéder.

L'exemple ci-dessus est un volume mappé.

## Gestion des volumes managés

Les volumes managés docker sont gérés avec la commande `docker volume`.

```
docker volume create mon_volume # créer un volume
docker volume ls # lister les volumes
docker volume rm mon_volume # supprimer un volume
```

Lors de l'exécution d'un conteneur, il faut aussi utiliser `-v` pour monter un volume managé. À l'inverse d'un volume mappé, il faut indiquer le nom du volume plutôt que son chemin.

```
docker run -v mon_volume:/home/thibaud/volume ubuntu
# Le contenu du volume mon_volume sera monté dans /home/thibaud/volume au sein du conteneur
```

Le volume est indépendant du conteneur auquel il est attaché :

```
debian@debian:~$ docker volume create mon_volume
mon_volume
debian@debian:~$ docker run -ti --rm -v mon_volume:/home/toto ubuntu
root@592d0e235bd4:/# cd /home/toto/
root@592d0e235bd4:/home/toto# echo "Je suis dans un volume !" > contenu.txt
root@592d0e235bd4:/home/toto# ls
contenu.txt
root@592d0e235bd4:/home/toto# exit
exit
debian@debian:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
debian@debian:~$ docker volume ls
DRIVER      VOLUME NAME
local      mon_volume
debian@debian:~$ docker run -ti -v mon_volume:/var/test debian
root@213af518f0de:/# cd /var/test/
root@213af518f0de:/var/test# cat contenu.txt
Je suis dans un volume !
root@213af518f0de:/var/test#
```

- Le volume `mon_volume` est attaché à un conteneur `ubuntu` dans `/home/toto` et le fichier `contenu.txt` y est créé.
- Lorsque le conteneur est arrêté, il est automatiquement supprimé puisqu'il a été lancé avec `--rm`.
- Le volume `mon_volume` persiste et son contenu est préservé.
- Un nouveau conteneur `debian` est lancé et le volume `mon_volume` y est attaché dans `/var/test` : le fichier `contenu.txt` est disponible.

Une subtilité des volumes managés est **la priorité des données** si l'un des deux répertoires est vide lors du lancement d'un conteneur :

| Volume | Point de montage (conteneur) | Action |
|--------|------------------------------|--------|
|        |                              |        |

|             |             |                                                                                   |
|-------------|-------------|-----------------------------------------------------------------------------------|
| Si vide     | Si non vide | Les données initialement dans le conteneur sont copiées dans le volume.           |
| Si non vide | Si vide     | Le contenu du volume est monté dans le conteneur, comme dans l'exemple ci-dessus. |
| Si non vide | Si non vide | Le contenu du répertoire dans le conteneur est écrasé par celui du volume.        |

---

Revision #35

Created 14 August 2025 20:33:26 by Thibaud FRICHET

Updated 5 September 2025 08:55:37 by Thibaud FRICHET