

SAÉ 1.03

Installation d'un Poste de Développement Partie 3 - Volumes

Introduction

Ce sujet est la suite directe de la **Partie 2 - Utilitaires**.

Vous avez forcément noté que les échanges entre les conteneurs nécessitaient de passer des commandes **docker container cp** entre les conteneurs et l'hôte.

Ça fonctionne bien mais ce n'est pas très pratique.

Heureusement, il existe une solution pour faciliter ces échanges : **les volumes**.

Volumes

Un conteneur est, par nature, un espace clos où les fichiers sont visibles uniquement de l'intérieur du conteneur.

Mais il est possible de partager des dossiers entre le conteneur et l'hôte. Physiquement, un tel dossier appartient à l'hôte et son contenu est visible dans le conteneur.

Dans le vocabulaire **docker**, ce partage s'appelle un **Volume**.

Hôte et conteneur peuvent tous deux modifier les fichiers contenus dans un volume. Les modifications apportées d'un côté sont immédiatement visibles de l'autre côté. Il ne s'agit pas d'une synchronisation de fichiers, ce sont physiquement les mêmes fichiers, vus de deux endroits différents.

Si ça peut vous aider à mieux comprendre, un volume est à Docker ce qu'une clé USB est à votre ordinateur : un espace de stockage qui est en dehors de la machine (ou du conteneur dans le cas de Docker) et qu'on vient en quelque sorte "brancher" pour que son contenu soit visible et accessible dans la machine (ou dans le conteneur dans le cas de Docker). Si on supprime l'ordinateur, la clé USB n'est pas impactée, et donc si on supprime le conteneur, les volumes ne sont pas non plus impactés¹. Évidemment, cette explication reste une analogie !

¹ Attention, lors de la suppression d'un conteneur, on verra qu'il existe une option permettant de supprimer en même temps les volumes attachés, mais c'est une action volontaire de l'utilisateur.

Il existe deux types de volumes : les volumes mappés et les volumes managés.

Volumes mappés

Commençons par celui qu'on n'utilisera pas à l'IUT mais qui est peut-être le plus pratique quand on fait du développement sur sa machine personnelle.

Vous vous demandez sans doute pourquoi on ne l'utilisera pas à l'IUT et aussi pourquoi on en parle quand même ici dans ce cas.

En fait, à l'IUT nous avons des impératifs sécuritaires qui contraignent beaucoup la façon dont on a configuré Docker.

Docker est un outil qui est généralement utilisé par des utilisateurs qui sont aussi administrateurs de leur machine. En effet, même si les conteneurs sont des environnements cloisonnés et qui, du fait de ce cloisonnement, apportent un gain non négligeable en termes de sécurité, la mise en place et la gestion de ces conteneurs demandent souvent des droits d'administration de la machine hôte.

Ces droits, vous n'en disposez évidemment pas sur les machines de l'IUT.

Cependant, comme vous pouvez aussi travailler sur un ordinateur chez vous, il est important que vous sachiez ce que sont les volumes mappés et ce qu'ils peuvent vous apporter.

Il est très courant d'utiliser des volumes mappés quand on développe sur sa machine personnelle. L'autre type de volume, qu'on va étudier un peu plus loin, est plutôt réservé au contexte des machines de production (serveurs) ou pour contourner certains inconvénients des volumes mappés.

Rappels : on l'a dit en introduction, un volume est un peu comme une clé USB qu'on viendrait brancher sur un conteneur. Le contenu d'un volume ne fait pas partie du conteneur. Ce dernier n'a seulement qu'un accès au contenu du volume. S'il en a les droits, il peut évidemment créer ou supprimer ce qu'il veut du volume, et même tout son contenu s'il le veut, mais le volume existera toujours à l'extérieur du conteneur. Les actions (créations, suppression, modifications) faites sur son contenu dans le conteneur ou à l'extérieur (dans l'hôte) sont visibles instantanément des deux côtés (hôte et conteneur).

Pourquoi "mappé" ?

Mapper un volume c'est indiquer à Docker qu'un dossier qu'on possède (par exemple, un dossier qu'on a créé pour un projet dans **Documents**) doit être visible dans un conteneur donné. On appelle cette action Monter un volume, et on en reparle un peu plus loin. Ce volume est dit *mappé* car il n'a pas été créé spécifiquement pour servir de volume par et pour Docker. C'est vous qui avez décidé de rendre visible, dans un conteneur, le contenu

d'un dossier existant, créé par vous pour cette occasion, mais qu'a priori rien ne différencie d'un autre dossier voisin.

Il n'est pas *managé* car c'est vous qui le gérez et non pas Docker.

Préparation

Note : Ne rien faire pour ce TP, juste lire ce paragraphe, car nous n'utiliserons pas les volumes mappés.

Pour préparer le mappage d'un volume, la seule chose à faire serait de créer un dossier quelque part dans votre arborescence et d'y placer ce que vous voulez partager avec un ou plusieurs conteneurs.

Le mappage en lui-même est décrit dans le paragraphe *Montage de volumes*, plus loin.

Volumes managés

Ce qui vient d'être dit à propos des volumes mappés et aussi valable en grande partie pour les volumes managés.

Ce qui fait la différence est que ces volumes, même s'ils sont sur la machine hôte, ne seront pas accessibles directement par l'explorateur de fichier sur les ordinateurs de l'IUT. Vous ne disposez pas des droits² suffisants pour le faire. Mais pas d'inquiétude, nous allons pouvoir y accéder par d'autres moyens, sinon on ne serait pas en train d'en parler ici !

Par contre, chez vous il sera possible d'y accéder si vous le souhaitez, à condition d'être **root**. Pour information, sous Linux vous les trouverez dans **/var/lib/docker/volumes**.

A l'IUT, ne cherchez pas à y accéder.

Pourquoi "Managé" ?

A contrario d'un volume mappé, un volume managé est créé pour et par Docker lui-même. Ce n'est pas vous qui créez un dossier sur disque pour servir de volume. Vous devez demander à Docker, par une commande spécifique qu'on va voir un peu plus loin, de créer un volume. D'ailleurs c'est bien un volume que vous créez et non pas un dossier. Le terme *dossier* est un terme associé à l'OS et à l'arborescence. Le terme *volume* est ici un terme de l'environnement Docker³.

Un tel volume est donc dit *managé* car c'est Docker qui le gère, qui le *manage*.

² On parlera des droits Unix prochainement en R1.04

³ Le terme *volume* existe aussi dans d'autres domaines du monde informatique.

Préparation

Note : Faites les exercices indiqués car nous allons utiliser les volumes managés à l'IUT.

Pour créer un volume managé, comme il a été évoqué précédemment, il faut explicitement demander à Docker de le faire à l'aide d'une commande spécifique.

Voici la syntaxe de la commande :

```
docker volume create <nom_du_volume>
```

Le **<nom_du_volume>** doit être adapté et doit être unique. Utilisez un nom explicite avec des lettres minuscules et/ou majuscules mais sans espace. Vous pouvez utiliser des - (tirets) ou des _ (underscores), ainsi que des chiffres, mais évitez les autres symboles et les caractères accentués.

Exercice : créez un volume nommé de votre nom de login et préfixé de **vol_**. Exemple pour Mathieu Lucas : **vol_mlucas**.

Pour lister les volumes existants, utilisez la commande suivante :

```
docker volume ls
```

Exercice : Vérifiez la présence de votre volume personnel tout nouvellement créé.

Pour supprimer un volume, utilisez la commande suivante :

```
docker volume rm <nom_complet_du_volume>
```

Le nom doit être donné in extenso, c'est une question de sécurité pour éviter de supprimer un mauvais volume par erreur. Attention, il n'y a aucune confirmation.

Exercice : Supprimez votre volume personnel.

Comme toutes les commandes Docker qui font des actions de suppression, Docker affichera le nom du volume supprimé comme confirmation du bon déroulement de l'action.

Notez qu'un volume ne peut être supprimé que s'il n'est pas monté dans un conteneur.

Alors mappé ou managé ?

À l'IUT, pas le choix mais chez vous c'est à vous de voir. Un petit résumé de la situation :

- Machine de développement : pas obligatoire mais souvent en mode **mappé**
- Serveur de production : généralement en mode **managé**
- Chez vous : plutôt en mode **mappé** si vous êtes **root** mais **managé** OK aussi.
- A l'IUT : obligatoirement **managé** pour des raisons de configuration (sécurité).

Montage de volumes

Comme le dossier partagé appartient en premier lieu à l'hôte, on appelle l'action de le rendre visible dans le conteneur : **monter le volume** dans le conteneur.

Nous allons décrire le mécanisme de montage pour les deux types de volumes : managés et mappés. La syntaxe diffère très légèrement. Nous allons donc placer un pictogramme spécifique pour chaque type.



désignera ce qui concerne les volumes managés, puisque c'est ce qu'on utilise à l'IUT



désignera ce qui concerne les volumes mappés, puisque c'est ce que vous pourrez utiliser chez vous. **Notez quand même que les volumes managés sont aussi possibles chez vous et qu'il n'y a aucune contre-indication à en faire usage aussi chez vous !**

Pour monter un volume, on utilise un **point de montage** dans le conteneur. Un point de montage est simplement un dossier de l'arborescence du conteneur. Ce dossier peut exister avant le montage mais ce n'est pas une obligation. S'il n'existe pas dans le conteneur, il sera créé au moment du montage.

Ainsi, un dossier partagé apparaît forcément à des endroits différents de l'arborescence, qu'on se place du point de vue de l'hôte ou de celui du conteneur.

Monter un volume ne peut se faire qu'au lancement d'un conteneur. On ne peut pas attacher un volume à un conteneur déjà créé (qu'il soit actif ou pas).

Syntaxe



En mode *managé*, l'option à ajouter à la commande **docker container run** pour monter un volume est la suivante. Mais il faut avoir créé un volume au préalable !

```
-v nom_du_volume:chemin_absolu_dans_le_conteneur[:ro]
```

où **nom_du_volume** est le nom d'un volume créé au préalable par une commande **docker volume create**, et **chemin_absolu_dans_le_conteneur** est le chemin du dossier partagé vu depuis le conteneur. Comme indiqué, c'est un chemin obligatoirement absolu.

Il est possible d'ajouter un **:ro** collé derrière, pour monter un volume en lecture seule (**ro** = **read only**). Sans cet ajout, par défaut, le montage se fait en lecture-écriture.

ATTENTION : le nom de l'image (ex : **qrcode**, **html2pdf**) doit toujours être placé à la fin de la commande **docker container run...**



En mode *mappé*, l'option à ajouter à la commande **docker container run** pour monter un volume est la suivante :

```
| -v chemin_absolu_d_un_dossier_hote:chemin_absolu_dans_le_conteneur[:ro]
```

où **chemin_absolu_d_un_dossier_hote** et **chemin_absolu_dans_le_conteneur** sont les deux chemins du dossier partagé. Le 1^{er} est celui vu de l'hôte et le second est celui vu du conteneur. Comme indiqué, ces chemins sont obligatoirement absolus.



Accès dans l'hôte à un volume mappé

Chez vous, vu de l'hôte, vous accédez directement avec votre explorateur de fichiers ou avec un éditeur comme VSC par exemple. Vous pouvez créer, modifier, supprimer des fichiers dans le dossier correspondant à la partie à gauche du **:** (deux-points) dans le **-v chemin_absolu_d_un_dossier_hote:chemin_absolu_dans_le_conteneur**

Accès dans un conteneur à un volume

Que vous soyez à l'IUT ou chez vous, que vous ayez un volume mappé ou managé, vu du conteneur, le volume est accessible par le chemin du dossier qui a servi de point de montage.

C'est la partie droite du **:** (deux-points) du **-v ...:chemin_absolu_dans_le_conteneur**

Accès aux volumes par VSC

Ce qui suit nécessite que vous ayez lancé un conteneur quelconque. Si ce n'est pas le cas, faites-le avant de poursuivre.



Il est possible que les actions dans les menus de VSC soient en **français** dans votre interface. Nous vous les indiquons ici en **anglais**, à vous de traduire si besoin, pour trouver la bonne entrée dans le menu.

Installation du plugin Dev Containers

Dans VSC, installer le plugin (l'extension) **Dev Containers** de Microsoft.

Connexion de VSC à un conteneur actif

Dans VSC :

- Cliquez sur l'icône  en bas à gauche (c'est vraiment tout en bas du bas du bas de la fenêtre, à gauche !)
- Choisissez **Attach to Running Container**
- Choisissez un conteneur auquel vous attacher. Attention, certains conteneurs ne seront pas attachables car ils doivent contenir le nécessaire pour que VSC puisse y installer des outils à lui. C'est le cas de **clock**, choisissez plutôt un conteneur issu de l'image **qrcode** par exemple.
- Une fois connecté au conteneur, rendez-vous dans l'Explorateur à l'aide de ce bouton : 

Puis choisissez :  et déplacez-vous dans le dossier de votre choix, peut-être par exemple **/work** s'il existe.

Ainsi, avec VSC, vous bénéficiez d'un explorateur de fichiers avec une interface graphique. Il accepte le glisser-déposer de fichiers/dossiers.

Détachement d'un conteneur

Pour vous détacher d'un conteneur dans VSC sans l'arrêter, refaites la même action  mais en choisissant **Close Remote Connection** cette fois-ci.

On vient de voir comment connecter VSC à un conteneur.

C'est pratique et ça vous servira plus tard mais pour les exercices suivants, cette technique ne doit pas être utilisée pour remplacer par des glisser-déposer.

Échanges entre conteneurs

Bénéficier d'un espace partagé entre hôte et conteneur par le biais d'un volume est très pratique mais ce qui l'est encore plus est de pouvoir le faire aussi entre plusieurs

conteneurs, sans même que l'hôte soit impliqué dans cet échange, et c'est ce que permettent les volumes car on peut monter un volume sur plusieurs conteneurs !

QRCode

Reprenez le 1^{er} exercice de la **SAÉ - Partie 2 - Utilitaires** et apportez le changement nécessaire lors de la création du conteneur **qrcode** pour que soit monté un volume (à créer au préalable) de votre choix, sur le point de montage **/work** dans le conteneur.

Attention, on n'utilise plus les commandes **docker container cp** mais on ne fait pas non plus de glisser-déposer avec VSC. C'est justement le but des volumes que de permettre des échanges entre les conteneurs sans passer par ces autres méthodes.

Poursuivez l'exercice jusqu'au moment où il vous faut récupérer le **code.png** en utilisant une commande **docker container cp**. Ne lancez pas cette commande, vérifiez juste avec VSC que vous voyez bien ce fichier dans le conteneur.

Suite du workflow

Reprenez la suite du workflow de la **SAÉ - Partie 2 - Utilitaires** en partageant aussi dans le conteneur **html2pdf** le même volume que vous avez créé, et en ne lançant aucun des **docker container cp** de et vers les deux conteneurs, tout se faisant désormais beaucoup plus simplement et directement grâce au volume monté dans les deux conteneurs.

Le résultat final peut alors être extrait du volume en passant par VSC.



Chez vous, en supposant que vous ayez utilisé le mode *mappé*, la récupération du résultat pourrait aussi se faire depuis le dossier partagé sur l'hôte. Mais à l'IUT, en mode managé, vous devez passer la VSC.

Epilogue

Les volumes sont très utiles pour partager des fichiers entre l'hôte et les conteneurs ou entre les conteneurs.

On l'a évoqué en introduction lors de l'analogie faite avec les clés USB, les conteneurs ont un énorme avantage : **la persistance des données**.

Même en utilisant des conteneurs éphémères (**--rm**), le contenu des volumes est préservé puisque les données sont, avant tout, physiquement sur l'hôte, le conteneur n'ayant qu'un accès partagé avec l'hôte.