

Introduction à Docker

Ce projet utilise *Docker for Desktop* pour sa réalisation. Une fois installé, utiliser la ligne de commande (*Powershell* ou *Bash*). Pour vous aider, consulter la section Get started sur *docker docs*. Référez-vous à la documentation officielle pour la suite.

Préparation

Cette section n'est pas à remettre. Il s'agit plutôt de prendre en note et s'habituer aux différentes commandes possibles.

Créer, démarrer, interrompre, arrêter et supprimer un conteneur

Référez-vous à la référence ligne de commande pour cette section. Trouver les commandes pour:

- Créer un conteneur
- Démarrer un conteneur
- Exécuter une commande dans un conteneur
- Interrompre un conteneur
- Se connecter (accès à un terminal) dans un conteneur
- Arrêter un conteneur
- Supprimer un conteneur
- Supprimer les conteneurs arrêtés

Volumes et liaisons hôte-invité

Il est possible de lier un ou des dossiers de l'hôte dans un conteneur. Il est aussi possible de créer un espace mémoire permanent appelé volume sur votre système pouvant être connecté/déconnecté sur un ou plusieurs conteneurs.

- Démarrer un conteneur en liant le dossier `/home/pi/config` au dossier `/config` d'un conteneur Ubuntu 20.04
- Créer un volume nommé DB dont la taille est de 512 Mo.
- Démarrer un conteneur dont l'image est Debian 11.1 avec le volume DB lié au dossier `/database`.

Création d'une image

Pour cette section, référez-vous à la section *Dockerfile reference*.

- Créer une image basée sur Ubuntu 21.04 et y installer l'éditeur de texte `vim`. Ensuite, créer un conteneur basé sur cette image, puis valider que vous pouvez démarrer l'éditeur `vim`. Pour quitter l'éditeur, faites la commande `:q!` (deux point, suivi de q puis ! et entré).

- Améliorer votre image dans créant un utilisateur **user** avec un dossier **/home/user**. Faites en sorte que vous être l'utilisateur **user** lorsque vous êtes en session interactive dans le conteneur.

Défi 1 - Image GPIO (facile)

Pour ce défi, crée une image permettant d'exécuter votre programme **gpio** conçu dans le précédent projet. Votre image doit:

- Être basé sur Ubuntu 21.04
- Accéder aux GPIO du Raspberry Pi
- Installé les logiciels nécessaires à la compilation de votre code
- Récupérer votre code via Git
- Compile votre code

Lorsque vous commencez une session interactive dans un conteneur basé sur votre image, vous devez être capable d'exécuter la commande **gpio** immédiatement.

Pour ce défi, récupérez le devoir à cette adresse: <https://classroom.github.com/a/3jav6yI6>
Vous devez me remettre tout simplement un fichier Dockerfile.

Défi 2 - Environnement de développement en conteneur (moyen)

Une des possibilités intéressantes en développement avec le principe de conteneur est d'avoir son environnement de développement directement dans un conteneur. Ainsi, peu importe l'ordinateur utilisé, il est possible de simplement récupérer son image et la démarrer. Pour cet exemple, vous allez développer un environnement de développement pour le langage C/C++ avec VSCode. Votre image ensuite se connectera à l'interface graphique de votre système pour afficher l'application. Voici les spécifications:

- Être basé sur Ubuntu 21.10
- Doit pouvoir s'exécuter sur un PC ou un Raspberry Pi (donc x64 ou arm)
- Contient les logiciels nécessaires pour compiler du code en C/C++
- Installer VSCode
- Contient un dossier **/workspace**. À l'exécution, un dossier de votre choix sur votre hôte sera lié à ce dossier.
- Créer un utilisateur **user**. Il devra avoir accès au dossier **/workspace** et être l'utilisateur par défaut du conteneur.
- L'image devra s'exécuter à l'aide de x11docker. Une présentation de l'outil se fera le 24 novembre 2021 en cours.

Pour ce défi, récupérez le devoir à cette adresse: <https://classroom.github.com/a/Dv4vDAGq>
Me remettre un Dockerfile

Défi 3 - Maintien d'une image Raspberry Pi à jour et configurée (avancée)

Ce dernier défi est plus compliqué que les autres. Il existe une image permettant de démarrer un émulateur exécutant une image de Raspbian directement sur un PC. Le nom de cette image est `lukechilds/dockerpi`. À partir de cette image Docker, améliorez là ou créez-en une nouvelle permettant :

- L'installation des dernières mises à jour
- La configuration du serveur de temps pour pointer vers `time.apple.com`
- L'installation de WiringPi
- La création d'un nouvel utilisateur avec un mot de passe
- L'ajout d'une clé publique SSH associé à votre nouvel utilisateur
- La désactivation de l'utilisateur `pi`
- La réécriture de l'image à mettre sur la carte SD

Pour ce défi, récupérez le devoir à cette adresse: <https://classroom.github.com/a/a5UCttkb>
Me remettre un Dockerfile