
TD2

Créer ses propres images Docker

Documentation utiles

Dockerfile reference : <https://docs.docker.com/engine/reference/builder/>

1 Construire sa propre image docker

Dans le TP précédent, un docker-compose permettait d'utiliser une image docker standard de *maria - db* et une image quasi standard de *apache + pphp* (on avait juste ajouté le support de PDO pour pouvoir utiliser les requêtes préparées).

Dans ce TP, je vous fourni un dockerfile de base. La suite de cette section consiste à modifier ce dockerfile.

1.1 Instruction FROM et RUN

Depuis un terminal

- A l'aide de la commande *git* récupérez le répertoire *tp2* du dépôt <https://gitlabinfo.iutmontp.univ-montp2.fr/coletta/r5.a.09-virtualisation-avancee>
- Construisez l'image du docker décrit dans le dockerfile fourni (docker build) de docker et lancer le docker (docker run)? Chercher les options requises pour nommer votre container lors du build , et l'exécuter en interactif .
- Depuis un second terminal, lister les docker en cours d'exécution
- Changez de distribution pour partir d'une distribution alpine, puis construisez l'image (en changeant son nom).
- Quelle est la taille du docker généré? comparé à l'image à base de Debian ?
Tapez la commande *docker* sans argument pour trouver l'argument permettant de lister les images.
- Testez différentes commandes de base depuis les deux images :
Qu'en déduisez vous?

1.2 Instruction COPY

Ici on repart de l'image à base de *debian*.

- A l'aide de l'instruction COPY, placer le fichier *cmd.sh* dans le docker
- Compilez le docker, relancez le pour vérifier la présence de ce script.

1.3 Instructions CMD et ENTRYPOINT

- Ajoutez une instruction CMD qui exécute le script */cmd.sh* ajouté au docker à la section 1.2
- Lancez le docker, que constatez vous?

Remarque : tous les dockers ne sont pas des serveurs (durée de vie infinie). Les dockers sont aussi utilisés pour des tâches one-shoot telles que configuration ou migration.

- Ajouter un ligne à la fin du fichier *CMD.sh* pour rendre à nouveau le docker interactif à son lancement. Comment mettre à jour.
- Copiez le script *entrypoint.sh* dans le container et ajoutez une instruction *ENTRYPOINT* au dockerfile.
- Lancer le docker en forçant une commande *bash*. Que constatez vous?

```
$ docker run -ti --rm mondocker bash
```

- Ajouter une instruction RUN qui inscrit la date dans un fichier *date.cmd.txt* du répertoire */tmp*

- o. Ajouter une ligne à la fin du *cmd.sh* et du *entrypoint.sh* qui écrit la date dans un fichier identifié du répertoire */tmp*. Dans quel ordre sont joués *ENTRYPOINT*, *CMD* et *RUN*? et à quelle fréquence?

Quelle instruction utiliseriez vous pour installer un paquet?

Quelle instruction utiliseriez vous pour configurer un logiciel au lancement du docker?

1.4 Instructions ENVIRONNEMENT

- p. Dans le dockerfile, ajoutez une variable d'environnement *\$nom* que vous initialisez à votre nom.
- q. Modifier le script *entrypoint.sh* pour qu'il affiche bonjour *nom*. Testez
- r. Relancez le docker en lui passant en ligne de commande une autre valeur aux variables *\$nom* et *\$prenom*

2 Revenons à notre docker compose de la semaine dernière

Notre docker de la semaine dernière est à l'état de développement, essayons de l'apporter vers la prod.

- s. Supprimer le montage de répertoire pour *app* (qui sont ok en cours de développement mais pas pour de la production) et remplacez les par des COPY.
- t. à l'aide de *curl* ou en appelant directement *php* en ligne de commande, faites en sorte que le script de création de la base soit lancée automatiquement : dans RUN, CMD ou ENTRYPOINT?
- u. Ajoutez l'affichage des erreurs dans PHP (pour le dev) puis conditionnez l'affichage des erreurs à la variable d'environnement "PROD=no".
- v. Variabilisez le fichier de config.php, testez en utilisant le mySql de l'IUT au lieu d'un docker mariadb
Voir rubrique Conf de <https://romainlebreton.github.io/R3.01-DeveloppementWeb/tutorials/tutorial2.html>

3 Construire un docker pour Django

- w. En vous basant sur la doc suivante, créez votre propre docker avec une mini-application en Django (framework MVC en python).

<https://docs.djangoproject.com/en/4.2/intro/tutorial01/#creating-a-project>

Vous pouvez partir d'une image docker *python*.

4 Publier mon image sur un registry

Information du registry :

URL bison.dep-info.iutmontp.univ-montp2.fr/cours/

Login : votre login LDAP

MDP : votre mdp LDAP

- x. En utilisant le commande docker login et docker push connectez vous au registry et publiez votre image
- y. Demander à un vos camarades de récupérer l'image sur le registry avec "docker pull" et de l'exécuter sur son poste