

Construction Docker

Le but de ce TP va être de construire des images Docker selon les consignes et de s'assurer que les images sont construites comme attendu.

Attention : certaines commandes comme le démarrage d'un container peuvent prendre parfois jusqu'à 30 secondes, voir un peu plus, soyez patient :)

1) **Connection sur la machine distante ou on doit travailler**

Pour se connecter sur la machine distante, il faut utiliser une connexion ssh. C'est une connexion sécurisée qui permet de ne pas faire passer le mot de passe en clair sur le réseau.

Vous pouvez utiliser le programme « putty » sur windows ou « ssh » sur linux.

La connexion se fait sur une machine dont l'adresse IP est 195.83.161.135. Pour que chaque étudiant ait sa propre machine, les ports ssh de connexion diffèrent. Lors de la mise en place du TP, l'enseignant vous a donné un numéro. Ce numéro est important car il va être utilisé pour le port ssh de connexion et le mot de passe.

Pour résumer:

IP: 195.83.161.135
port: 122<votre numéro sur 2 digits>
login: student
mot de passe est: student

Ex:

Si votre numéro est 86 alors le port sera 12286

Une fois sur votre machine distante, vous pouvez l'administrer totalement. Vous devez absolument changer le mot de passe par défaut le plus rapidement possible.

Pour pouvoir noter le TP, il faut que vous mettiez dans un fichier nommé /root/info.txt votre nom et prénom. Attention, pas de fichier = zéro !

Tout ce qui est noté sur fond bleu ... (xxx réponse à fournir)...correspond à une question à laquelle vous devez répondre dans le document de rendu.

A la fin de la séance, vous devez rendre ce document dans la zone de rendu « Rendu séance », même si vous n'avez pas fini le TP.

2) **Construction de container:**

Maintenant que vous savez bien manipuler les images et les containers, il est temps de construire ses propres images.

Pour construire notre première image, nous allons créer et remplir un fichier de description avec des informations qui vont permettre de construire cette image.

Comment doit s'appeler le fichier de description (1ère réponse à fournir) ?

Vous allez maintenant éditer le fichier, et trouvez les bonnes informations pour que les pré-requis suivants soit respectés:

- l'image de base est Ubuntu
- Mettre le mode non interactif sinon pour les commandes suivantes, vous allez être embêté
- Mettre à jour les packages existants

Copiez/Collez le contenu du fichier (2ème réponse à fournir).

Construisez votre image, indiquez la commande et l'output dans le document de rendu **(3ème réponse à fournir).**

Lancer un container à partir de votre image de manière non-interactive, ça doit se lancer mais quitter aussitôt.

Refaire de même mais en interactif.

Maintenant il est temps de construire un vrai conteneur. Pour cela, rajouter dans le fichier de description:

- l'installation de Apache2
- La suppression des packages inutiles
- une commande pour faire que le port du serveur web qui sera géré par Apache2 soit visible de l'extérieur du container. Attention visible ne veut pas dire qu'il sera forcément visible, juste qu'on le rend visible et que si le Host fait ce qu'il faut, le port sera bien visible du Host.
- Lancer le serveur apache2 avec la commande `apachectl` (pas en tâche de fond)

Copiez/Collez le contenu du fichier (4ème réponse à fournir).

Builder le docker, et essayer maintenant de lancer un container à partir de cette image. Pour voir si cela fonctionne (si le serveur Apache est bien lancé) il y a plusieurs manières de le faire:

- * regarder les logs du container
- * lancer le container et regarder si il y a bien un process sur le port 80 en écoute dans ce docker (en exécutant une autre commande sur le host)
- * lancer le container de manière interactive et se connecter sur le port 80 depuis le host pour voir si un process répond.
- * lancer le container et mapper le port 80 du container sur le port 8080 du host et vérifier qu'en se connectant au port 8080 du host on a bien une réponse du serveur Apache2.

Pour chacune des possibilités énoncées ci-dessus, indiquez les réponses que vous avez trouvé:

- * regarder les logs du container (5ème réponse à fournir).
- * alncer le container et regarder si il y a bien un process sur le port 80 en écoute dans ce docker (en executant une autre commande sur le host) (6ème réponse à fournir).
- * lancer le container de manière interactive et se connecter sur le port 80 depuis le host pour voir si un process répond (7ème réponse à fournir).
- * lancer le container et mapper le port 80 du container sur le port 8080 du host et vérifier qu'en se connectant au port 8080 du host on a bien une réponse du serveur Apache2 (8ème réponse à fournir).

Du coup comme cela ne fonctionne pas bien, expliquez ce qu'il manque et donnez la réponse dans le document de rendu (9ème réponse à fournir).

Une fois corrigé, essayez de reconstruire l'image et de la lancer en détaché. Ensuite pour chacune des 4 possibilités, ci-dessus, refaite les opérations de check, et indiquez les réponses dans le document de rendu:

- * regarder les logs du container (10ème réponse à fournir).
- * lancer le container de manière interactive et regarder si il y a bien un process sur le port 80 en écoute (11ème réponse à fournir).
- * lancer le container de manière interactive et se connecter sur le port 80 pour voir si un process répond (12ème réponse à fournir).
- * lancer le container et mapper le port 80 du container sur le port 8080 du host et vérifier qu'en se connectant au port 8080 du host on a bien une réponse du serveur Apache2 (13ème réponse à fournir).

Faite mal même chose mais avec un container qui utiliser MySQL. Indiquez tout ce que vous avez fait pour que cela fonctionne. Attention il faut que le port du serveur MySQL soit aussi visible du host (port 3307) et que vous puissiez communiquer avec le serveur MySQL depuis le host avec un mysql client. (14ème réponse à fournir).

Comme vous êtes arrivé jusque là facilement, pour la dernière question, voici le challenge, construire une image complexe embarquant plusieurs outils :

- mysql
- php
- Apache
- Wordpress

L'idée étant de pouvoir construire une image Docker qui contient un wordpress fonctionnel. Attention, le répertoire qui héberge les fichiers wordpress (php) doivent être dans un volume partagé avec le host, pareil pour les fichiers de la base de donnée, de manière à pouvoir relancer le docker et récupérer les données du Docker.

A vous de jouer. Précisez dans la 15ème réponse à fournir. Comment vous avez fait. Pensez à donner le contenu de votre fichier Dockerfile.