

R202: TP Docker

Jean-Marc Pouchoulon

Mai 2025

1 Pré-requis, recommandations et notation du TP.

Les pré-requis sont les suivants:

- Avoir un PC sous Linux.
- Avoir installé Docker ou utiliser une OVA prête à l'emploi. Merci **de ne pas** utiliser les packages fournis par les distributions qui sont souvent moins récents que les packages fournis par Docker.
- Avoir installé docker-compose, il est présent sur les OVA à votre disposition.
- Vous devez avoir un compte sur le site Docker: <https://hub.docker.com/>.

Vous travaillerez individuellement.

1.1 Installation de Docker et obtenir de l'aide.

1.1.1 Changements à réaliser sur votre VM

Relancez docker:

```
systemctl reload docker
```

1.1.2 Rappel: Installation de Docker sous Linux.

Vous travaillerez avec une VM en utilisant l'OVA Debian.

1.1.3 Aide sur Docker.

```
man docker-run  
man docker-create
```

Accéder à la Documentation Docker:

<https://docs.docker.com/>

Documentation sur les commandes Docker:

<https://docs.docker.com/engine/reference/commandline/>

La complétion avec la touche tab fonctionne aussi.

Connectez-vous à votre compte DockerHub via docker login.

2 Docker sous Linux.

2.1 Installation de Docker sous Linux

L'installation est faite sur la machine virtuelle mais vous pouvez retrouver la procédure d'installation sous <https://docs.docker.com/engine/installation/linux/docker-ce/ubuntu/>

1. Quelle la version de Docker installée? Retrouvez des informations sur le daemon docker.

2. Vérifiez que votre installation fonctionne bien avec la commande :

```
docker run hello-world
```

3 Compétences à acquérir lors du TP.

Compétence principale:

- "Containériser" des logiciels pour la mise en œuvre de projets au cours de votre apprentissage.

4 Compétences à acquérir lors du TP.

Compétence principale:

- "Containériser" des logiciels pour la mise en œuvre de projets au cours de votre apprentissage.

- a) Que vous exp

5 Compétences à acquérir lors du TP.

Compétence principale:

- "Containériser" des logiciels pour la mise en œuvre de projets au cours de votre apprentissage.

lique le retour de cette commande (au delà de "tout s'est bien passé" reformulez en Français..)?

- b) Retrouvez sur <https://hub.docker.com/> l'image hello-world.
 - c) Expliquez les mécanismes en jeux pour la création du container helloworld. Quel est le fichier sur DockerHub qui permet de créer le container ?
3. Recherchez les images officielles Debian à l'aide de docker search. Récupérez-les ainsi que les images officielles busybox (une distribution légère).
 4. Créez votre premier container à partir de l'image Debian officielle et en utilisant la commande "docker run -d debian" sans argument.
 5. En utilisant la commande "docker ps" vérifiez que le container est "vivant" ? expliquez ?
 6. Relancez le "docker run" en lui donnant comme argument bash -c "while ;; do echo "coucou" ; sleep 1; done".
 7. Stoppez et redémarrez le container.
 8. Supprimez le container.
 9. Utilisez les options "-it" afin d'être dans le container après son lancement.
 10. Même opération mais nommant le container et son hostname DebianOne.
 11. Détachez-vous du container DebianOne puis rattachez-vous à lui de nouveau.
 12. Lancez un processus bash supplémentaire dans le container DebianOne. Pour cela utilisez la commande docker exec.
 13. Listez le container restant. Ne listez ensuite que le dernier ContainerId.
 14. Affichez les logs du container DebianOne.
 15. Utilisez un volume pour donner à votre container l'accès à un répertoire de l'hôte. Quels sont les avantages de l'utilisation d'un volume ? un inconvénient ? A l'aide de la commande docker volume affichez les volumes présents sur votre hôte.
 16. Supprimez le container et son image.

17. Supprimez tous les containers avec un oneliner sous bash. Idem pour les images.
18. Supprimez les images et les containers non utilisés avec la commande "docker system prune".
- 19.

6 Création d'images Docker

Dans cette partie nous allons apprendre à créer une image Docker.

Récupérez les fichiers pour cet exercice via git:

```
git clone https://github.com/pushou/tpdocker.git
```

6.1 Build d'une image Docker Debian

1. Construisez l'image "debian:vosinitiales" à partir du Dockerfile du repository et de la commande "docker build..."
2. Expliquez ce que font les différentes commandes "RUN, ENV, FROM" de ce Dockerfile.
3. Quel est l'intérêt de faire tous les apt-get en une seule fois pour la taille de l'image Docker. (indice: voir AUFS et Docker).
4. A partir de l'image "debian:vosinitiales" générez une image contenant un client snmp comme fait manuellement dans le précédent TP"
5. Sur la machine distante Ouvrez deux terminaux. Dans le premier terminal, passez les commandes suivantes:

```
export DOCKER_HOST=tcp://localhost:2376
socat -v -ls TCP-LISTEN:2376,reuseaddr,fork UNIX-CLIENT:/var/run/docker.sock
```

Dans le second terminal, passez faites des commandes Docker. Expliquez ce que vous voyez dans le premier terminal.

6. Lancez un container issu de cette image au travers de la commande "docker run -rm -it ...". A quoi sert le -rm ?
7. Testez votre container avec le container serveur SNMP de l'exercice précédent.
8. A l'aide de la commande "docker history" retrouvez le contenu de votre Dockerfile"
9. Appelez l'enseignant pour validation.

7 Tips & Tricks

7.1 Connexion à distance au daemon Docker

Docker est bâtie sur une architecture modulaire qui lui permet de se connecter à un daemon Docker sur une machine distante en TLS ou SSH.

1. Créez un contexte pour vous connectez à distance au daemon Docker de votre VM (Vous pouvez utiliser une autre VM Linux ou installer Docker)

```
docker context create ubuntuvm --docker "host=ssh://student@VOTRE_IP"
```

2. Vérifier le bon fonctionnement du contexte en créant un container sur la VM.