

# 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 et personnalisez votre prompt avec votre nom et prénom. Utilisez la commande :

```
hostnamectl set-hostname PC-votrePrénom-votreNom
```

Les copies d'écrans feront apparaître ces éléments dans votre compte-rendu. et vous ne pourrez utiliser que votre compte-rendu pour le QCM à suivre.

- 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.
- Vous devez avoir un compte sur le site Docker: <https://hub.docker.com/>.

Vous travaillerez individuellement.

### 1.1 Obtenir de l'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 Obtenir des informations sur votre installation Docker

1. Quelle la version du daemon Docker installée ? de votre client Docker ?
2. Quel est le storage driver utilisé par Docker ? expliquez ce que c'est.

### 2.2 Premiers pas avec Docker.

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

```
docker run hello-world
```

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

- a) Retrouvez sur <https://hub.docker.com/> l'image hello-world.
- b) 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. Vérifiez l'état du container avec la commande "docker ps". Le container est-il "vivant" ?
6. Relancez le "docker run -d debian" en lui donnant comme argument bash -c "while ;; do echo "coucou"; sleep 1; done". Le container est-il vivant ? expliquez.
7. Stoppez et redémarrez le container.
8. Utilisez les options "-it" afin d'être dans le container après son lancement. Retrouvez son IP
9. Même opération mais nommant le container et son hostname DebianOne.
10. Lancez la commande "ip a" dans le container DebianOne sans faire de run -it.
11. Détachez-vous du container DebianOne puis rattachez-vous à lui de nouveau.
12. Lancez un processus bash supplémentaire dans le container DebianOne. Pour celà, utilisez la commande docker exec.
13. Lancez un container issu de cette image au travers de la commande "docker run -rm -it ...". A quoi sert le -rm ?
14. Listez uniquement les id des containers à l'arrêt. Ne listez ensuite que le dernier Container créé
15. Supprimez le container debianone et l'image debian.

## 2.3 Réseaux Docker

1. Lancez un container Nginx qui soit accessible sur le port 8010 de votre machine.
2. Que se passe-t-il si vous vous attachez à ce container ? qu'en déduisez-vous ?
3. Trouvez malgré tout l'adresse IP de ce container ?
4. lister les réseaux docker (docker ps) et expliquez les différents types de réseaux visibles.
5. Quel est le mécanisme réseau standard de communication entre un container et le réseau de la salle ?
6. Lancez un nouveau container à partir de l'image registry.iutbeziers.fr/debianiut:latest en utilisant au run le DNS de CloudFlare (1.1.1.1). Sniffez le trafic pour le vérifier.

## 3 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
```

### 3.1 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. Testez votre container avec le container serveur SNMP de l'exercice précédent.
7. A l'aide de la commande "docker history" retrouvez le contenu dy Dockerfile de l'image server de SNMP". Combien d'années se sont écoulées depuis la création de l'image.
8. Appelez l'enseignant pour validation.

## 4 Tips & Tricks

### 4.1 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.
3. Installez dive: voir (<https://github.com/wagoodman/dive> pour visualiser l'image en "couche" de Debian de docker.)
4. Comparer avec l'image registry.iutbeziers.fr/debianiut:latest
5. Faites valider par l'enseignant.