

TP Noté N°3: Docker et VMs

I - Travail Attendu

Dans ce TP nous attendons la production d'un rapport au format PDF envoyé à [<jbbesnard@paratools.fr>](mailto:jbbesnard@paratools.fr) le vendredi suivant le TP avant 23h59. Soyez concis et efficaces mais ne renvoyez pas non plus un document en texte brut – trouvez le juste milieu. Le travail peut être fait seul ou en binôme.

Récupération des exercices:
git clone

II - Introduction

Dans le cadre des travaux dans des environnements divers, il est parfois important de pouvoir déployer des versions précises des logiciels, voir même des systèmes d'exploitation et de leurs environnements. Le but de ce TP est de vous familiariser avec les méthodes de virtualisation pour savoir lancer une VM sous linux dans votre environnement et également une introduction aux conteneurs. Le but de ce TP est d'agir comme un cheminement de découverte afin de vous permettre de vous approprier ces technologies; Ainsi, c'est quasiment un cours que vous allez vous faire à vous même, les questions et exercices seront là pour souligner les points à comprendre dans le fonctionnement des ces abstractions. Enfin, le but est également de vous donner une vision pratique de ces outils très utiles dans le cadre de la validation et plus généralement du développement des applications. Certaines questions vont vous demander de la recherche documentaire sur internet ou via les LLM. Ce qui vous est demandé est de comprendre les choses, et vous pouvez poser toutes les questions que vous souhaitez pendant le TP, n'hésitez pas !

Exercice 1: Bases techniques

Ici nous attendons des réponses courtes et précises, utilisez tous les moyens pour vous documenter. Internet, wikipedia, LLM, ... Le but est que vous sachiez rapidement acquérir une vision critique des technologies de virtualisation et de conteneurisation.

- 1) Connaissances de base:
 - a) Expliquez ce qu'est une machine virtuelle;
 - b) Expliquez la différence avec un conteneur;
 - c) Réalisez un schéma comparant les deux configurations, il doit faire apparaître:
 - i) Le(s) noyau(x) Linux
 - ii) L'application en cours d'exécution (conteneur ou VM)
 - iii) Les limites du conteneur et de la VM
 - d) Quels sont les avantages / inconvénients des deux méthodes ?
 - i) Quelle est l'approche la plus sécurisée ?
 - ii) Quelle est l'approche la plus performante ?
 - iii) Quelles sont les raisons d'utiliser l'une ou l'autre ?
 - e) Donnez trois scénarios justifiant l'utilisation de VM et/ou conteneurs
- 2) Exercez votre sens critique, préciser ces affirmations sont-elles correctes (justifiez brièvement pourquoi) ?
 - a) Une VM est plus légère qu'un conteneur

- b) Le conteneur permet de changer de noyau
- c) Je peux changer de distribution Linux avec un conteneur
- d) On ne peut lancer qu'une VM par noeud physique
- e) Si je développe un module noyaux (code dans le kernel) un conteneur est le bon choix pour faire mes modifications
- f) La VM doit avoir la même architecture que le système hôte

III - Lancement d'une VM

Il faut tout d'abord vous donner les moyens de lancer une VM, en fonction de si vous êtes sous Linux ou Windows ces étapes peuvent varier. Voici l'approche basique pour Windows et différentes distributions Linux (retenez l'approche pour votre système).

Installation de VirtualBox sous Windows/ MacOS / Linux :

Téléchargement de VirtualBox :

Allez sur le site officiel de VirtualBox : [VirtualBox Download](#).
Téléchargez la version correspondant à votre système d'exploitation Windows.

Installation de VirtualBox :

Lancez le fichier d'installation téléchargé.
Suivez les instructions à l'écran pour installer VirtualBox en acceptant les paramètres par défaut.

Lancement de VirtualBox :

Une fois l'installation terminée, lancez VirtualBox à partir du menu Démarrer ou de l'icône sur le bureau.

Installation de QEMU sur les distributions Linux principales :

Installation sur Ubuntu / Debian :

```
sudo apt-get update  
sudo apt-get install qemu
```

Installation sur Fedora :

```
sudo dnf install qemu
```

Installation sur CentOS / RHEL :

```
sudo yum install qemu
```

Installation sur Arch Linux :

```
sudo pacman -S qemu
```

Vérification de l'installation :

Une fois l'installation terminée, vous pouvez vérifier si QEMU est installé en tapant la commande suivante dans le terminal :

```
qemu-system-x86_64 --version
```

Cela devrait afficher la version de QEMU installée sur votre système.

Vous avez maintenant installé VirtualBox sous Windows et QEMU sur votre distribution Linux. Vous pouvez commencer à utiliser ces outils pour créer et gérer des machines virtuelles selon vos besoins.

Exercice 2 : Lancement de ma VM

Récupérez l'image de la VM pour le cours soit via la clef USB ou bien sur <https://france.paratools.com/atoi.qcow2> (qemu) ou bien <https://france.paratools.com/atoi.vdi> (virtualbox).

- 1) Expliquez la sortie de la commande **qemu-system-x86_64** quelle est la raison de cette erreur ?
- 2) Maintenant nous allons lancer la VM:
 - a) Si sous Linux Expliquez le contenu et les arguments de cette commande:
qemu-system-x86_64 -hda atoi.qcow2 -m 1024 -netdev user,id=eth0,hostfwd=tcp::10022-:22 -device e1000,netdev=eth0
 - b) Sous VirtualBox, lancez la VM en exposant le port 22 et en activant le support du réseau depuis l'hôte. Expliquez votre configuration.
- 3) Que voyez-vous s'afficher ?
- 4) Sur votre machine (directement sous Linux ou avec Putty/WSL sous Windows) ouvrez un Shell et connectez vous en ssh:
ssh atoi@localhost -p 10022 (mdp: toto)
 - a) Dessinez un schéma de cette connexion
 - b) Sur quel port le serveur SSH tourne dans la VM ?
- 5) Essayez de lancer qemu sans affichage (pour ceux sous Linux):
 - a) Démarrer le système sans affichage: -nographic
 - i) CTRL + a puis x pour quitter
 - ii) CTRL + a puis c pour ouvrir la console qemu (qemu-monitor)
 - b) Comment pouvez-vous accéder à la VM dans ce cas ?

Vous avez maintenant la commande de base pour lancer une VM sur votre machine, soit dans qemu ou bien virtualbox. Notez que cette VM contient Docker pour les exercices suivants, au cas où votre OS ne soit pas Linux.

Exercice 3 : Briques pour Créer sa Propre VM

Le but de cet exercice est de vous pointer les étapes pour si-besoin créer votre VM. Nous ne sommes pas obligés de procéder à une installation complète si vous avez déjà un système opérationnel. Cependant, il est important de savoir le faire et donc si vous ne l'avez jamais fait procéder à l'installation.

Si Qemu

- 1) Création d'une image vide (cas qemu) expliquez la commande:
qemu-img create -f qcow2 mydebian.qcow2 5G
- 2) Lancez l'image avec:
qemu-system-x86_64 -m 1024 -hda ./mydebian.qcow2
 - a) Raison de l'erreur ?
- 3) Que rajouter pour avoir la configuration réseau fonctionnelle ? (voir Exo 2)
- 4) Téléchargez l'image d'installation de Debian par le réseau:
<https://cdimage.debian.org/debian-cd/current/amd64/iso-cd/debian-12.2.0-amd64-netinst.iso>
- 5) Ajoutez-la à la VM (quel paramètre utilisez vous pour insérer le cdrom ?)
- 6) Avez-vous réussi à lancer (screenshot de Grub dans le rapport) ?

Si Virtualbox

- 1) Comment créez-vous une image disque ?
- 2) Comment lancer cette image ?
- 3) Avez-vous configuré internet dans l'image ?
- 4) Comment insérer un CDROM (celui de netinstall voir lien dans qemu) ?
- 5) Avez-vous réussi à lancer (screenshot de Grub dans le rapport) ?

IV - Les Conteneurs

Nous avons maintenant fait le tour des VMs, concentrons nous sur les conteneurs. Vous aurez besoin de Docker et avez soit la possibilité de l'installer sur votre OS (Linux, OSX, dans certains cas Windows) voir <https://docs.docker.com/get-docker/> ou bien vous pouvez utiliser la VM que vous avez récupérée lors de l'exercice 2, elle contient Docker. Voici une liste des commandes de base Docker avec une brève explication de chaque commande, vous êtes encouragés à les essayer :

- **docker version** : Affiche la version de Docker installée sur votre système, ainsi que les informations sur le client et le serveur Docker.
- **docker info** : Affiche des informations détaillées sur la configuration de Docker, telles que le nombre de conteneurs en cours d'exécution, d'images disponibles, etc.
- **docker pull [nom_image]** : Télécharge une image Docker depuis Docker Hub ou un autre registre d'images. Par exemple, `docker pull nginx` téléchargera l'image Nginx.
- **docker images** : Affiche la liste des images Docker téléchargées localement sur votre système.

- **docker run [options] [nom_image]** : Crée et lance un conteneur à partir de l'image spécifiée. Par exemple, `docker run -d -p 8080:80 nginx` lancera un conteneur Nginx en arrière-plan sur le port 8080 de l'hôte.
- **docker ps** : Affiche la liste des conteneurs en cours d'exécution sur votre système.
- **docker ps -a** : Affiche la liste de tous les conteneurs, y compris ceux qui ne sont pas en cours d'exécution.
- **docker stop [ID_conteneur ou nom_conteneur]** : Arrête un conteneur en cours d'exécution. Vous pouvez spécifier soit l'ID du conteneur, soit son nom.
- **docker rm [ID_conteneur ou nom_conteneur]** : Supprime un conteneur arrêté de votre système. Vous pouvez spécifier soit l'ID du conteneur, soit son nom.
- **docker rmi [nom_image]** : Supprime une image Docker de votre système. Vous devez d'abord supprimer tous les conteneurs associés à cette image.
- **docker exec -it [ID_conteneur ou nom_conteneur] [commande]** : Exécute une commande à l'intérieur d'un conteneur en cours d'exécution. L'option `-it` permet une interaction en mode `tty`.

Exercice 4: Mon premier conteneur

- 1) Lancez un conteneur debian:
docker run -ti --rm debian
 - a) quel utilisateur êtes-vous dans le conteneur (commande `whoami`) ?
 - b) Que contient le système de fichier / ?
 - c) Est-il différent de celui de votre machine ?
- 2) Maintenant lancez:
docker run -ti --rm rancher/cowsay cowsay "Hello Docker"
 - a) Expliquez les différents arguments
 - b) Que représente **rancher/cowsay** (voir <https://hub.docker.com/r/rancher/cowsay>)
 - c) Que fait la commande **cowsay "Salut Linux"**
 - d) Expliquez où se situe le binaire `cowsay` ?
 - e) Quelles sont les différentes étapes lors du lancement du conteneur (en considérant que le conteneur n'est pas présent) ?
 - f) Pourquoi la commande est plus rapide lors du second lancement ?

Exercice 5: Mon premier serveur Web

- 1) Lancez la commande suivante:
docker run -ti -p8080:80 nginx
 - a) Expliquez l'effet des différents arguments
 - b) Que voyez-vous en vous connectant sur <http://localhost:8080> (en ayant le conteneur lancé)
 - i) Note si vous êtes dans la VM vous pouvez utiliser la commande **curl <http://localhost:8080>** pour voir le contenu de la page
 - c) Trouvez le nom de votre conteneur en lançant:
docker ps

- d) Connectez vous dans votre conteneur avec:
docker exec -ti [NOM CONTENEUR] bash
 - e) Quelle est le chemin de la racine du serveur WEB (regardez la configuration de NGINX)
cat /etc/nginx/conf.d/default.conf
 - f) Si vous vous rendez dans ce dossier et exécutez la commande suivante:
echo "<h1>PAWNED</h1>" > index.html
 - i) Le contenu de <http://localhost:8080> a-t-il changé ?
 - ii) Vous pouvez quitter le shell dans le conteneur avec CTRL+D ou exit
- 2) Nous voulons maintenant exposer notre propre site web, pour ce faire nous voulons remplacer le dossier racine du serveur web en insérant notre dossier à la place.
Pour cela nous allons utiliser un volume (option -v) la syntaxe est:
docker run -ti -p8080:80 -v SOURCE:DESTINATION nginx
Cela va insérer le dossier SOURCE à DESTINATION dans le conteneur
- a) Lancez le conteneur en remplaçant le site par celui dans ./ex5/html du dépôt du TP
 - b) Quel est le mot de passe de la page secrète ?

Exercice 6: Redis et Script

- 1) lancer un serveur Redis (https://hub.docker.com/_/redis) en exposant le port 6379 du conteneur sur le port 6379 local
 - a) Quelle commande utilisez-vous ?
 - b) Qu'est ce que redis ?
 - c) Connectez-vous au redis avec la commande
nc localhost 6379
 - d) Effectuez des commandes simples:
SET lol 123
GET lol
- 2) Réalisez un script python qui compte le nombre d'occurrence dans guerre et paix Tome 1 (voir ./ex6/gp.txt). Pour ce faire, nous utiliserons le Redis pour compter avec la commande HINCR (<https://redis.io/commands/hincrby/>). Pour terminer on affichera le compte final (voir <https://redis.io/commands/hkeys/>)