

**CPE Lyon - 3ICS - Année 2021/22****Administration des systèmes Linux**

# TD3 – LXC/Docker

---

L'objectif de ce TD est :

- de découvrir les systèmes de conteneurisation (LXC)
- d'apprendre à créer un conteneur docker
- d'utiliser un docker de base
- de voir le cadre d'utilisation des conteneurs docker

Vous devrez répondre à toutes les questions et rendre un document numérique avec vos réponses.

1. Récupérez et déployez la VM (CentOS 7 Minimal) , ou installez une CentOS 7 ou 8 à partir de l'ISO

- La VM doit être une CentOS avec :
  - Une seule carte réseau branchée sur le VMNET8
  - 2 GO de RAM
  - CPU avec 2 cores.

2. Désactivez SELinux

3. Désactivez le pare-feu.

4. Origine des conteneurs

Les systèmes unix/bsd utilisent, dans certains cas, des « jails »

- [4A] Qu'est-ce qu'une « jails » ? Une jails est une architecture logicielle qui permet de compartimenter des processus et leurs descendants.
- [4B] Dans quels cadres utilise-t-on les « jails » ? Lorsqu'on le souhaite crée un environnement sécurisé et séparé du reste du système.
- [4C] Quel est l'équivalent des « jails » sous Linux ? Linux Containers
- [4D] Qu'est-ce qu'un CGROUP sous Linux ? cgroups (control groups) est une fonctionnalité du noyau Linux pour limiter, compter et isoler l'utilisation des ressources.

1. Les conteneurs

Docker, malgré son succès et ces déboires actuels, n'est pas la première technologie de conteneurisation.

- [5A] Quel est le principe de la conteneurisation ? Un conteneur est une structure de données, une classe, ou un type de données abstrait, dont les instances représentent des collections d'autres objets.
- [5B] Quelle est la différence essentielle entre un conteneur et une VM ? L'isolation lors de la virtualisation se fait au niveau matériel alors qu'avec la conteneurisation elle se fait au niveau du système d'exploitation
- [5C] Quels sont les différents types de conteneur existant ? Kubernetes, Docker et OpenShift

6. Installation et prise en main de LXC

Les conteneurs Docker sont basés sur la technologie des conteneurs LXC. Cette partie du TD doit vous permettre de voir l'intérêt des conteneurs LXC ainsi que leurs limites.

- Installez les packages nécessaires pour tester les conteneurs LXC `yum install epel-release yum install wget yum install lxc lxc-templates libvirt lxc-extra`
- Vérifiez votre configuration `lxc-checkconfig`
- [6A] Activez et démarrez le service `libvirtd`. Quel est son rôle ? Il remplit le rôle de virtualiser la gestion du système comme le stockage, les interfaces réseau, les états des VM, etc.
- [6B] Activez et démarrez le service `lxc`. Quel est son rôle ? LXC permet la création et la gestion des conteneurs.

## 7. Création d'un conteneur LXC

- [7A] Il existe 2 façons rapides de créer localement un conteneur : Quelles sont elles ? Les deux façons sont avec privilège ou sans. Avec la commande `lxc-create` accompagné de `sudo`.
  - Dans certains cas, il est possible qu'il y ait un message d'erreur provenant de la commande `lxc-create` : `Setting up the GPG keyring ERROR: Unable to fetch GPG key from keyserver`
- [7B] Dans quelles conditions pouvez-vous rencontrer ce message d'erreurs ? Lorsqu'on souhaite télécharger une image de conteneur protégée par une clé GPG et qu'on n'arrive pas à télécharger la clé pour vérifier l'image.
- [7C] Qu'est-ce qu'un serveur de clés ? Sur quel port écoute-t-il ? un serveur de clés est un serveur qui : reçoit des clés de chiffrement de leurs propriétaires ; emmagasine ces clés ; puis les distribue aux utilisateurs ou programmes qui veulent communiquer de façon chiffrée avec les propriétaires des clés. Port 11371
- [7D] Quelles solutions proposez-vous ? Désactivez la vérification des images. Avec `sed -i "/^DOWNLOAD_VALIDATE/s/true/false/" /usr/share/lxc/templates/lxc-download`
- [7E] Quelle est la commande permettant de déployer un conteneur LXC de type debian ? `lxc-create -n container1 -B lvm --vgname vglxc --fstype=ext4 --fssize=50G -t download -- -d debian -r buster -a amd64`
- [7F] Déployez un conteneur centos, version 7. Quelle commande avez-vous utilisée ? A quoi sert l'option `-n` ? `lxc-create -n container3 -t download -- -d centos -r 7 -a amd64` L'option `-n` est pour nommer le conteneur
- [7G] Démarrer le conteneur. Quelle commande avez-vous utilisée ? Quelle option vous permet de démarrer le conteneur en background ? `lxc-start -n container3`. Aucune option car il démarre de base en arrière plan.
- [7H] A quoi servent les commandes
  - `lxc-info` ? Récupérer des informations sur le conteneur
  - `lxc-execute` ? Exécuter une application dans le conteneur
  - `lxc-attach` ? Démarrer un service dans le conteneur
  - `lxc-top` ? Quel est l'empreinte mémoire de votre conteneur ? Voir les statistiques sur le conteneur.

- [7I] A quoi sert l'option -B de la commande lxc-create ? **Connecter le conteneur au système mère pour rendre persistente les données dans un répertoire**
- [7J] Que contient le fichier /etc/lxc/default.conf ? **Les paramètres des conteneurs.**
- [7K] Ou est déployer le conteneur créé ? **Sous /var/lib/lxc**
- [7L] Que contient le répertoire « rootfs » de ce répertoire ? **Le stockage du conteneur**
- [7M] Que contient le fichier config de ce répertoire ? **Les paramètres de configuration du conteneurs**
- [7N] Dans la VM, la commande ip addr fait apparaître un périphérique nommé virbr0. Qu'est ce que ce périphérique. A quoi sert-il ? **C'est l'interface utilisé par LXC pour faire le NAT des conteneur**
- [7O] Si votre conteneur LXC est bien démarré, la commande ip addr doit aussi afficher un périphérique dont le nom commence par veth. Qu'est ce que ce périphérique ? Quel est son rôle ? **C'est l'interface réseau du conteneur qui est coinnecté au réseau interne des conteneurs et de la machine**
- [7P] Lors de la création du conteneur, aucun mdp ne vous a été demandé. Proposez une solution pour changer le mot de passe **On lance le terminal du conteneur grâce à la commande lxc-attach puis on rentre les commandes voulu, passwd dans notre cas**
- [7Q] Vous avez maintenant 2 options pour vous connecter sur le conteneur. Quels sont elles ? **lxc-attach -n container\_centos**
- [7R] « Loggez » vous sur le conteneur et :
  - Donnez l'IP **192.168.122.5/24**
  - Installez le service ssh, démarrez-le. Quelles commandes avez vous utilisées ? **yum install openssh-server et systemctl start sshd**
  - Connectez-vous sur le conteneur en ssh depuis un autre terminal de votre VM
- [7S] Pouvez vous joindre le conteneur depuis Windows ? Si non pourquoi ? **Oui**
- [7T] Imaginez que vous deviez déployer très souvent des conteneurs avec des services Apache et MySQL. Quels sont les problèmes que vous allez rencontrer ? Quels solutions proposez vous ? **Il faudra installer à la main à chaque fois les services. Utiliser Docker pour créer des images personnalisés**
- Pour les fans de Debian : déployez un conteneur Debian 10, installer le service ssh et Apache, vérifiez que vous pouvez atteindre le serveur Web

## 8. Création d'un conteneur Docker

Docker permet de « versionner » les conteneurs créés et d'automatiser l'installation des services et de fixer rapidement la configuration réseau ainsi que les partages entre le conteneur et l'hôte.

- Installez, activez et démarrez le service docker.

- [8A] Il existe 2 solutions pour créer un conteneur docker. Quelles sont elles ? Avec une interface graphique grâce à Docker Hub ou en ligne de commande avec la commande `docker run <options> name`
- [8B] La commande `ip addr` affiche un périphérique nommé `docker0` ? A quoi correspond-t-il ? C'est l'interface réseau qui permet d'utiliser un réseau de type bridge

## 9. Utilisation d'un Dockerfile

Pour créer un conteneur vous allez utiliser un Dockerfile. Vous pouvez le faire à partir d'une image disponible sur le dockerhub (<https://hub.docker.com/>). Il existe tout un ensemble de conteneur prêt à l'emploi (<https://hub.docker.com/search?q=&type=image>) avec des services installés et configurés. (redis , mondo, nodejs...)

L'autre solution consiste à créer un conteneur à partir d'une image locale ou distante de base et à installer et paramétrer les services nécessaires grâce à des commandes placées dans le fichier Dockerfile.

- Le lien [https://hub.docker.com/\\_/httpd](https://hub.docker.com/_/httpd) vous permettra de déployer un docker avec un service Apache fonctionnel
- [9A] Quel est le contenu du fichier Dockerfile qui est proposé ?

```
FROM httpd:2.4
COPY ./public-html/ /usr/local/apache2/htdocs/
```

- [9B] A quoi sert la directive FROM ? Définit l'image de base qui sera utilisée par les instructions suivantes.
- [9C] A quoi sert la directive COPY ? Permet de copier des fichiers depuis notre machine locale vers le conteneur Docker.
- [9D] A quoi correspond `./public-html/` ? Au dossier local
- [9E] A quoi correspond `/usr/local/apache2/htdocs/` ? Au dossier du conteneur où il faut copier le contenu du dossier local
- Créez un fichier `info.php` contenant la commande `phpinfo()` ; (cf TD serveur web) dans le répertoire `public-html`.
- [9F] Quelle est la commande utilisée pour créer l'image docker ? Créez une image nommée « apachy »

```
docker build -t apachy .
```

- [9G] Quelle commande permet de lister les images présentes dans le dépôt local (Repository)

```
docker images
```

- [9H] Que contient votre dépôt local ?

| REPOSITORY | TAG    | IMAGE ID     | CREATED       | SIZE  |
|------------|--------|--------------|---------------|-------|
| apachy     | latest | 6879ed60afa5 | 5 seconds ago | 143MB |
| httpd      | 2.4    | ad17c88403e2 | 10 days ago   | 143MB |

- [9I] Votre dépôt ne devrait contenir que 2 images. Quelle commande devez vous utiliser pour supprimer une image ?

```
docker rmi name
```

- [9J] Démarrez votre conteneur nommé « apachy » A quoi servent les options :

--name ? Définir un nom

--d Exécuter en arrière plan et donner l'id du conteneur

--it Allouer un pseudo TTY (terminal virtuel) et garder un STDIN ouvert

--p Faire une redirection de port

- [9K] Pouvez vous vérifier que votre conteneur est lancé ? Comment ?

```
docker ps
Puis regarder son statut qui doit être up
```

- [9L] La commande ip addr affiche un périphérique particulier. Lequel ? veth716fa31@if23
- [9M] Quelle commande permet d'afficher l'IP du conteneur ?

```
docker inspect -f '{{.Name}} - {{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' $(docker ps -aq)
```

- [9N] Essayez de joindre le serveur web du conteneur, quelle réponse obtenez vous ? Je tombe sur la page par défaut de Apache It's works
- [9M] Récupérez le fichier info.php qui doit être présent sur le serveur web du conteneur « apachy ». Qu'observez-vous ? ``

## 1. Création d'une image personnalisée avec un Dockerfile

Vous allez installer les packages nécessaires pour avoir un serveur web dynamique.

- [10A] A quoi sert la commande RUN présente dans un fichier Dockerfile Exécute des commandes Linux lors de la création de l'image

- Créez une image local nommée Web a partir d'une image Fedora 31 ([https://hub.docker.com/\\_/fedora?tab=tags](https://hub.docker.com/_/fedora?tab=tags))
- Installez Apache dans l'image
- Installez les packages php php-cli php-gd php-mbstring php-mysqlnd php-pdo php-sodium php-xml via le Dockerfile
- [10A] Lancez l'image. Quelle démarche avez-vous utilisée pour vérifier :
  - Que l'image est lancée ? `docker ps`
  - Que le serveur web fonctionne ? `sudo docker exec -d web_c apt-get install net-tools` puis `sudo docker exec web_c netstat` On vérifie que le conteneur écoute le protocole web
  - Que le moteur php dans le conteneur répond correctement. Création d'un fichier `info.php` dans le serveurs web et accéder à la page web

Dockerfile web:

```
FROM fedora:31

RUN dnf install -y -q httpd-manual php php-cli php-gd php-mbstring php-mysqlnd
php-pdo php-sodium php-xml

EXPOSE 80

ENTRYPOINT systemctl start httpd.service && systemctl enable httpd.service
```