

LXD

Table of Contents

- [LXD](#)
- [Installation et configuration](#)
 - [Pré-requis](#)
 - [Installation](#)
 - [Configuration](#)
- [Création et manipulation des containers](#)
 - [Création](#)
 - [Manipulation](#)
 - [Exportation / Importation](#)
- [Travailler dans les containers](#)

LXD est un logiciel libre développé par Canonical pour simplifier la manipulation de conteneurs de logiciels à la manière d'un hyperviseur de VM. C'est une surcouche logicielle de LXC. Il fait partie du projet global Linux Containers de gestion de containers.

Il a l'avantage d'être beaucoup plus léger qu'une machine virtuelle classique, car il ne virtualise pas un OS complet mais partage de nombreuses ressources avec l'OS hôte. On parle d'environnements virtuels (voir la page virtualisation pour plus d'information). Il s'appuie sur des fonctions comme les espaces de noms et les groupes de contrôle du noyau pour isoler le container du reste de la machine. En conséquence, **LXD** ne peut pas accueillir des systèmes autres que Linux (il n'est pas capable d'accueillir un container Windows donc) et ces systèmes ont été spécialement préparés pour fonctionner dans un container. Il n'est donc pas n'ont plus possible d'utiliser des ISO de n'importe quelle distribution Linux.

LXD permet donc de faire tourner plusieurs serveurs sur une même machine et de les isoler les uns des autres et de la machine hôte pour une plus grande sécurité.

Pour un particulier, il peut être intéressant d'utiliser un container pour démarrer un serveur ouvert sur l'internet. Ainsi, si il est mal configuré et qu'un hacker arrivait à s'introduire sur votre serveur, il serait bloqué à l'intérieur du container et n'aurait pas accès aux fichiers persos, qui eux restent sur l'hôte.

Installation et configuration

Pré-requis

Avoir installé Ubuntu 16.04 ou une version ultérieure

Installation

Pour l'installer, il suffit d'installer le paquet lxd.

```
sudo apt install -y lxd
```

Configuration

La configuration de LXD se réalise en entrant la commande suivante dans une console :

```
sudo lxd init
```

Il faut alors répondre aux questions du configurateur. On peut simplement valider toutes les réponses par défaut proposées.

- Les images des containers seront alors stockée sur le disque (option dir à la question Name of the storage backend to use (dir or zfs) [default=zfs]:)
- LXD sera accessible uniquement en local (option no à la question Would you like LXD to be available over the network (yes/no) [default=no]?)
- Une connexion réseau par pont (bridged network) sera configurée sur l'ordinateur. En validant les options proposées dans les boîtes de dialogue de cette étape, on va créer un sous-réseau privé virtuel sur la machine. Les différents containers pourront communiquer entre eux sur ce réseau. Ils pourront accéder à l'extérieur grâce au mécanisme NAT (Network Address Translation) mais ne seront pas accessibles depuis l'extérieur. Pour faire simple, vos containers sont sur un réseau interne à votre ordinateur exactement comme vos ordinateurs sont sur votre réseau local derrière votre box Internet / Firewall.

Si besoin, on pourra reconfigurer votre sous-réseau privée virtuel en entrant la commande :

```
sudo dpkg-reconfigure -p medium lxd
```

Enfin, afin de ne pas à avoir à passer en mode super-utilisateur à chaque commande, il faut ajouter son user au groupe lxd

```
sudo adduser nom_utilisateur lxd
```

Création et manipulation des containers

Création

Maintenant que l'hyperviseur est correctement configuré, on peut créer un premier container Debian

nommé MonContainer en entrant la commande :

```
lxc launch images:debian/stretch MonContainer
```

LXD va aller chercher sur le serveur d'image images: l'image correspondant à la distribution Debian dans sa version Stretch. Il existe 3 serveurs d'images:

- **ubuntu:** (contenant les images des versions stables d'Ubuntu)
- **ubuntu-daily:** (Contenant les dernières versions d'Ubuntu)
- **images:** (Contenant les images de diverses distributions Linux)

Pour obtenir la liste des images disponibles sur un serveur d'images, par exemple pour le serveur ubuntu-daily: taper la commande

```
lxc image list ubuntu-daily:
```



Il est possible de créer des conteneurs lxc à partir de rootfs préchargés en utilisant le script salt_tarball

https://raw.githubusercontent.com/saltstack/salt/develop/salt/templates/lxc/salt_tarball

```
lxc-update-config -c salt_tarball
lxc-create -n smail -t /tmp/rootfs/salt_tarball -- --network_link
vnet0 --imgtar /tmp/rootfs/rootfs.tar.xz
```

Manipulation

Pour obtenir la liste des containers présents sur la machine, taper :

```
lxc list
```

Pour arrêter l'exécution de votre container MonContainer

```
lxc stop MonContainer
```

Et reprendre son exécution

```
lxc start MonContainer
```

Et enfin pour le supprimer définitivement

```
lxc delete MonContainer
```

Exportation / Importation

Comme les machines virtuelles, les containers sont portables. Lorsqu'on a configuré un container, on peut l'exporter pour le réimporter sur une autre machine hôte.

Pour l'exportation, la commande est

```
lxc publish MonContainer --alias=MyContainer --force  
lxc image export MyContainer
```

On se retrouve alors avec un fichier de ce genre 0bf450c17cb7034077c83487e0.tar.gz. On peut le passer sur un autre système hôte et l'importer dans un serveur d'images local avec la commande suivante :

```
lxc image import 0bf450c17cb7034077c83487e0.tar.gz
```

Taper ensuite

```
lxc image list
```

pour connaître les infos sur votre image importée et les utiliser pour créer un nouveau container

Travailler dans les containers

Pour exécuter une commande dans le container MonContainer, taper par exemple :

```
lxc exec MonContainer -- apt get update
```

Lorsqu'on souhaite exécuter plusieurs tâches, il est alors préférable de “rentrer” dans le container comme suit :

```
lxc exec MonContainer -- /bin/bash
```

On peut aussi y entrer en temps que utilisateur ubuntu

```
lxc exec MonContainer -- sudo --login --user ubuntu
```



Les images Unubuntu ont un user ubuntu préconfiguré avec les droits de sudo sans



password.

On peut faire ce qu'on veut dans le container sans risque d'endommager ou corrompre le système hôte. Ce qui se passe dans le container reste dans le container.

Il est toutefois possible de partager des fichiers et/ou des répertoires entre l'hôte et le container. Par exemple, pour copier le fichier truc.pdf dans le container :

```
lxc file push truc.zip MonContainer/tmp/
```

On peut aussi récupérer ce fichier truc.zip sur le système hôte:

```
lxc file pull MonContainer/etc/truc.zip .
```

Enfin, pour monter un repertoire de partage entre le système hôte et un container:

```
lxc config device add MonContainer MonContainerDevice disk source=/tmp/  
path=/host/tmp
```

Ceci monte le dossier /tmp/ de l'hôte vers le dossier /host/tmp du container.

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=virtualisation:lxd-presentation>

Last update: **2025/05/02 14:14**

