

# 03-Exploitation

## A - Commandes :

### 1. Gestion du stockage :

#### *Gestion du storage pool BTRFS*

```
# Afficher le storage pool :  
lxc storage list  
lxc storage show lxc-storage01  
lxc storage list info lxc-storage01  
  
# Affiche l'espace disque du volume BTRFS :  
sudo btrfs filesystem show
```

#### *Gestion des fichiers des CT :*

```
Accès au stockage des CT :  
# Emplacement :  
/var/snap/lxd/common/mntns/var/snap/lxd/common/lxd/storage-pools/lxc-storage01/containers/CT/rootfs  
  
# Lien symbolique du CT :  
ln -s /var/snap/lxd/common/mntns/var/snap/lxd/common/lxd/storage-pools/lxc-storage01/containers/CT/rootfs/  
/home/adminme/CT
```

### 2. Commandes LXC :

#### *Gestion LXC*

# Gestion des profils :

lxc profile list

lxc profile show default

# Gestion des réseaux :

lxc network list

lxc network show lxdbr0

# Afficher les informations d'un CT :

lxc info CT-NAME

lxc info --show-log CT-NAME

lxc show CT-NAME

# Se connecter dans un CT :

lxc exec CT-NAME bash

# Editer la configuration d'un CT :

lxc config edit CT-NAME

# Démarrage d'un serveur d'un CT :

lxc exec CT-NAME service -- ssh start

#####

Source : <https://www.cyberciti.biz/faq/create-snapshots-with-lxc-command-lxd/>

### Gestion des snapshots d'un CT :

# Créer un snapshot :

lxc snapshot CT-NAME Snapname

# Vérifier les snapshots :

lxc info CT-NAME

# Restaurer un snapshot :

lxc restore CT-NAME Snapname

# Supprimer un snapshot :

lxc delete CT-NAME/Snapname

lxc delete CT-NAME/Snapname -i # Force la confirmation

lxc copy CT-NAME01/base030423 CT-NAME02

lxc delete CT-NAME/base090323

```
# Liste de l'image LXD intégrée pour diverses distributions Linux :  
# Pour répertorier toutes les images LXD, exécutez :  
lxc image list images: | grep -i ubuntu
```

## ***Gestion des conteneurs***

```
# Création d'un conteneur BASE qui servira de template :  
# Syntaxe : lxc launch images:{distro}/{version}/{arch} {container-name-here}  
lxc launch ubuntu:22.04 BASE  
  
# Lister les conteneurs :  
lxc list  
  
# Répertorier les instances de conteneur Linux  
# Nous pouvons répertorier les conteneurs existants en utilisant la syntaxe suivante et la commande grep /  
commande egrep en tant que pipe :  
lxc list --fast  
lxc list | grep RUNNING  
lxc list | grep STOPPED  
lxc list | grep -i opensuse  
lxc list "*c1*"  
lxc list "*c2*"
```

## **Comment exécuter/exécuter la commande spécifiée dans un conteneur**

Nous exécutons ou exécutons des commandes dans des conteneurs à l'aide de la commande `exec` comme suit :

```
lxc exec containerName -- command  
lxc exec containerName -- /path/to/script  
lxc exec containerName --env EDITOR=/usr/bin/vim -- command  
### run date, ip a, ip rm and other commands on various containers ###  
lxc exec cenots-8-c2 -- date  
lxc exec cenots-8-c2 -- ip a  
lxc exec ubuntu-focal-c5 -- ip r  
lxc exec fedora-31-c9 -- dnf -y update  
lxc exec debian-10-www -- cat /etc/debian_version
```

# Comment obtenir l'accès au shell bash dans un conteneur

Vous voulez obtenir un accès login/shell dans un conteneur nommé debian-10-www, entrez :  
Maintenant, vous pouvez exécuter des commandes ou installer des packages. Par exemple, utilisez la commande apt / la commande apt-get à l'intérieur du conteneur : tapez simplement exit pour revenir à l'hôte :

```
lxc exec {container-name} {shell-name}
lxc exec debian-10-www bash
lxc exec CT-NAME bash
lxc exec alpine-c1 sh
# cat /etc/*issue*
# apt update
# apt upgrade
# exit
```

## Autres commandes

```
# Comment afficher des informations sur les serveurs et conteneurs LXD ?
lxc info
lxc info {container-name}
lxc info opensuse15-1-c10
lxc info -n CT-ID

# Vérifier la config :
lxc checkconfig

# Extraire un fichier du conteneur
lxc file pull {container-name}/{path/to/file} {/path/to/local/dest}
lxc file pull ubuntu-xenial-c3/var/www/nginx/app/config.php .

# Copier et récupérer des fichiers d'un CT :
# Il est possible de transférer des fichiers vers un CT : PUSH
# Mais également de récupérer des fichiers depuis un CT : PULL
# Pousser un fichier vers le conteneur
lxc file push {/path/to/file} {container-name}/path/to/dest/dir/
lxc file push config.php ubuntu-xenial-c3/var/www/nginx/app/
# PUSH d'un fichier vers un CT :
lxc file push /home/adminme/file.txt CT-NAME/tmp/
# PUSH d'un dossier vers un CT :
```

```
lxc file push -r /home/adminme/DOSSIER CT-NAME/tmp/  
# PULL d'un fichier depuis un CT :  
lxc file pull /home/adminme/file.txt CT-NAME/tmp/  
# PULL d'un dossier depuis un CT :  
lxc file pull -r CT-NAME/tmp/DOSSIER /home/adminme/  
  
# Obtenir de l'aide sur les commandes lxc  
lxc --help  
lxc {command} --help  
lxc list --help  
  
# Gestion d'in Conteneur :  
lxc start CT-ID  
lxc restart CT-ID  
lxc stop CT-ID  
lxc delete CT-ID  
lxc stop ubuntu-xenial-c3 && lxc delete ubuntu-xenial-c3  
  
# Configurer l'autostart d'un CT :  
lxc config set CT-ID boot.autostart true
```

# B - Procédure de création d'un CT :

## 1. Création d'un CT :

```
# Déclarer la variable :  
CTNAME=MONCT01  
# Création du conteneur :  
lxc launch ubuntu:22.04 $CTNAME  
  
# Configurer l'autostart d'un CT :
```

```
lxc config set $CTNAME boot.autostart true
```

## 2. Paramétrage :

Editer le fichier de configuration afin de paramétrer une adresse IP fixe et le forward de port.

### ***Configuration réseau et forward de port :***

Il est nécessaire de bien renseigner le fichier de zone DNZ "lxc.morbihan-energies.fr" afin d'attribuer une adresse IP sur le réseau LXC (non routé).

```
# Editer le fichier de configuration :  
lxc config edit $CTNAME
```

Ajouter les lignes suivantes :

```
devices:  
  eth0:  
    ipv4.address: 192.168.25.XXX  
    name: eth0  
    network: lxdbr0  
    type: nic  
  port52201:  
    connect: tcp:0.0.0.0:22  
    listen: tcp:192.168.1.31:52201  
    type: proxy  
  port53080:  
    connect: tcp:0.0.0.0:80  
    listen: tcp:192.168.1.31:53080  
    type: proxy
```

### ***Création du compte d'admin :***

```
# Se connecter dans le conteneur :  
lxc exec $CTNAME bash  
  
# Créer un utilisateur :  
useradd adminme
```

```
# Ajouter un utilisateur au groupe sudo :
usermod -aG sudo adminme

# Modifier le mot de passe :
passwd adminme
passwd root

# Création de l'arborescence pour le déploiement des clés SSH :
mkdir /home/adminme/.ssh
vi /home/adminme/.ssh/authorized_keys
chown -R adminme:adminme /home/adminme/.ssh

# Autoriser un sudo sans demande de mot de passe :
sudo visudo

# Ajouter les lignes ci-dessous :
%sudo ALL=(ALL:ALL) NOPASSWD:ALL
```

## ***Configuration de la connexion SSH :***

```
# Autoriser la connexion par mot de passe :
vi /etc/ssh/sshd_config

#Ajouter les lignes suivantes à la fin du fichier de conf :
-----
# Autoriser connexion par mot de passe :
PasswordAuthentication yes

# JKT-20220517 - Connexion Guacamole :
# Source : http://shaarli.guiguishow.info/?4DVyLA
HostkeyAlgorithms +ssh-rsa
PubkeyAcceptedAlgorithms +ssh-rsa
-----
```

```
# Redémarrer le service SSH :  
/etc/init.d/ssh restart
```

## B - CT-BASE :

### 1. Description :

#### *SSS-Titre 1*

#### **Titre 1**

Ce Conteneur a pour objectif d'être un Template qui pourra être déployé en dupliquant un snapshot.

### 1. Installation :

Cette procédure n'est pas encore terminée

```
# Prérequis :  
apt update && apt upgrade -y  
apt install software-properties-common bash-completion wget git -y  
  
# Installation de nginx :  
  
# Installation de php-fpm :  
$ sudo apt install php-fpm  
  
vi parametres.php  
vi /etc/nginx/sites-enabled/  
vi /etc/nginx/sites-enabled/default
```

# 1. Exploitation :

Cette procédure n'est pas encore terminée

---

Revision #1

Created 8 February 2024 08:14:16 by Jérôme

Updated 3 March 2026 08:05:51 by Jérôme