

# LXC

- 01-Description
- 02-Installation
- 03-Exploitation
- 04-Gestion réseau et port forwarding
- 10-Scripts
- 05 - Transfert de Conteneur

# 01-Description

## A - Description :

Nom DNS interne : lxc01.v.sdem.fr

Nom DNS Externe : NC

Type de serveur (Physique, VMware, LXC, Docker...) : VM VMware

OS : Ubuntu 22.04 LTS

Nom : lxc01.v.sdem.fr

CPU : 6

RAM : 12Go

Disque LVM :

/ : 50Go # ext4

/LXC : 300Go # Formaté par le daemon LXD en BTRFS

### **Rôle :**

Serveur hébergeant des CT LXC

CT LXC :

### **Accès :**

[\*] Interne

[x] Externe

### **Réseaux :**

Adresse IP : 192.168.1.31

Masque : 255.255.255.0

Passerelle : 192.168.1.254

DNS : 192.168.1.50

VLAN : SDEM\_SERVEURS - 5

# 02-Installation

## A - Paramétrage :

### 1. VMware :

#### Message UUID dans Syslog :

Sur Ubuntu 20, le syslog est vite noyé avec des messages UUID, il y a un petit réglage à effectuer côté VMWARE :

<https://www.suse.com/support/kb/doc/?id=000016951>

Pour faire simple il faut arrêter la VM et ajouter un attribut sur les VM Ubuntu 20 :

```
disk.EnableUUID = "TRUE"
```

#### VMware tools :

```
# Installation des VMware Tools :  
apt install open-vm-tools
```

### 1. Gestion date et heure:

```
# Changer la date :  
date -s "02/18/2017 12:34:00"  
  
# Changer le fuseau horaire :  
rm /etc/localtime  
ln -s /usr/share/zoneinfo/Europe/Paris /etc/localtime  
# Pour que le changement de fuseau survive au redémarrage de la machine, il faut aussi le mettre dans le bon  
fichier.  
vi /etc/timezone
```

```
ZONE="Europe/Paris"

# Ce qui donne en une seule commande :
sudo su
rm /etc/localtime
ln -s /usr/share/zoneinfo/Europe/Paris /etc/localtime
cat > /etc/timezone << eof
ZONE="Europe/Paris"
eof
cat /etc/timezone
exit
date
```

## 3. Accès SSH :

L'algorithmme ssh-rsa est désactivé dans la version de SSH livrée dans Ubuntu 22.04 car il utilise la fonction cryptographique SHA-1 qui est jugée dangereuse depuis des années. La connexion SSH de guacamole vers ce serveur ne fonctionne pas. Il est nécessaire de réactiver temporairement la connexion RSA en attendant la mise à jour de Guacamole.

Source : <http://shaarli.guiguishow.info/?4DVyLA>

```
vi /etc/ssh/sshd_config
```

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

/etc/init.d/ssh restart
```

## 4. Prérequis :

```
# Mise à jour du serveur :
apt-get update && apt-get upgrade -y
```

```
# Installation des utilitaires :
```

```
apt-get install ntp postfix
```

```
# Installation de la supervision :
```

```
apt-get install nagios-nrpe-server nagios-plugins -y
```

## ***Ajout du disque de stockage BTRFS pour LXC :***

### **Disque LXC :**

Ce disque est dédié pour stocker les CT LXC.

On ne configure pas de base ce disque qui va être géré directement par LXC en BTRFS.

Il ne faut pas configurer le disque en ZFS car cela nécessite une configuration avancé du paramètre ashift.

Par défaut l'ashift est de 512byte =  $2^9$

Hors sur les serveurs les secteurs sont à minima 4k =  $2^{12}$

Ce qui fait qu'il va écrire sur les disques 8 fois plus et cela va également générer un problème de fragmentation du disque et à terme une panne accéléré du disque.

## **5. Installation de LXC :**

Certaines terminologies courantes utilisées avec LXC incluent :

chroot - Chroot, également appelé changement de racine ou changement de racine, est une section du système de fichiers isolée du reste du système de fichiers. Un programme exécuté dans cet environnement ne peut pas accéder aux fichiers en dehors de l'arborescence de répertoires désignée.

cgroups - Il s'agit d'une fonctionnalité du noyau qui permet d'agréger ou de partitionner des tâches (processus) et tous leurs enfants en groupes organisés hiérarchiquement pour isoler les ressources.

```
# Installation de lxd et des outils associés :
```

```
# sudo apt install lxd lxc-utils iraf-dev zfsutils-linux -y
```

```
sudo apt install lxc lxd-installer
```

## ***Ajouter un utilisateur au groupe LXD à des fins de gestion :***

Pas besoin d'être l'utilisateur root pour gérer le démon LXD. Pour gérer le serveur LXD, ajoutez votre nom d'utilisateur au groupe lxd à l'aide de la commande adduser sur Ubuntu Linux:

```
sudo adduser adminme lxd
```

# Configurer les options de mise en réseau et de stockage LXD :

Maintenant que nous avons installé LXD, il est temps de configurer le serveur LXD. Vous devez configurer l'option de mise en réseau et de stockage telle que le répertoire, ZFS, Btrfs, etc., entrez :

```
sudo lxd init
```

Ensuite, vous devez répondre à une série de questions sur la façon de configurer le serveur LXD, y compris l'option de stockage et de mise en réseau.

Voici un exemple de session que j'ai configuré pour ZFS et le sous-réseau NAT 10.105.28.1/24.

N'hésitez pas à choisir les options selon vos besoins:

Prérequis pour la préparation du stockage en ZFS :

```
sudo lxd init
```

```
Would you like to use LXD clustering? (yes/no) [default=no]: # no
```

```
Do you want to configure a new storage pool? (yes/no) [default=yes]: # yes
```

```
Name of the new storage pool [default=default]: # lxc-storage01
```

```
Name of the storage backend to use (ceph, btrfs, dir, lvm, zfs) [default=zfs]: # btrfs
```

```
Create a new BTRFS pool? (yes/no) [default=yes]: # yes
```

```
Would you like to use an existing empty block device (e.g. a disk or partition)? (yes/no) [default=no]: # yes
```

```
Path to the existing block device: /dev/disk/by-id/scsi-36000c2920fa37484c1bbc95b339370e5 # Stockage sdb
```

```
Would you like to connect to a MAAS server? (yes/no) [default=no]: # no
```

```
Would you like to create a new local network bridge? (yes/no) [default=yes]: # yes
```

```
What should the new bridge be called? [default=lxdbr0]: # lxdbr0
```

```
What IPv4 address should be used? (CIDR subnet notation, "auto" or "none") [default=auto]: # 192.168.25.1/24
```

```
Would you like LXD to NAT IPv4 traffic on your bridge? [default=yes]: # yes
```

```
What IPv6 address should be used? (CIDR subnet notation, "auto" or "none") [default=auto]: # none
```

```
Would you like the LXD server to be available over the network? (yes/no) [default=no]: # no
```

```
Would you like stale cached images to be updated automatically? (yes/no) [default=yes] # yes
```

```
Would you like a YAML "lxd init" preseed to be printed? (yes/no) [default=no]: # no
```

```
# Redémarrer le serveur :
```

```
reboot
```

Vous êtes ensuite invité à configurer un pont réseau pour les conteneurs LXD. Cela active les fonctionnalités suivantes :

- Chaque conteneur obtient automatiquement une adresse IP privée.
- Chaque conteneur peut communiquer entre eux sur le réseau privé.
- Chaque conteneur peut initier des connexions à Internet.

- Chaque conteneur reste inaccessible depuis Internet par défaut ; vous ne pouvez pas initier une connexion à partir d'Internet et atteindre un conteneur à moins que vous ne l'activiez explicitement. Vous apprendrez à autoriser l'accès à un conteneur spécifique à l'étape suivante.

Nous pouvons vérifier les informations en tapant les commandes suivantes:

```
lxc profile list
lxc profile show default

lxc network list
lxc network show lxdbr0

lxc storage list
lxc storage show lxc-storage01

sudo btrfs filesystem show
```

## ***Configurer VIM pour éditer la configuration des CT à la place de nano***

Par défaut, nano est utilisé pour l'éditer de la configuration des containers. La documentation suivante permet de modifier l'éditeur et utilise vim.

Source : <https://askubuntu.com/questions/1116800/change-default-editor-in-lxd-snap>

```
# Modification de l'éditeur :
echo 'export EDITOR=vim' >> ~/.profile
source ~/.profile

# Test :
lxc config edit $CTNAME
```



# 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

# 04-Gestion réseau et port forwarding

## A - Gestion du réseau LXC :

Afin de réserver les adresses IP en cas de plusieurs serveur LXC et éviter les conflits. Il est nécessaire de renseigner le fichier de zone DNS `"/etc/named/zone/lxc.morbihan-energies.fr"`.

La zone sert uniquement d'inventaire et n'est pas publiée.

Nom du CT	Adresse IP
BASE	192.168.25.150
WIKI	192.168.25.151
BASH	192.168.25.152
DASHBOARD	192.168.25.153
RACKTABLES	192.168.25.232
KANBOARD	192.168.25.156

## B - Gestion du forwarding de port :

La gestion du forwarding de port (SSH, HTTP, MySQL...) est recensé dans un tableur spécifique :

N:\05-Informatique\01-Systeme Réseaux\Network-Configuration.ods (Page LXC-Network)

Ce paramétrage permet à une personne de se connecter en SSH directement sur le CT sur un port spécifique sans passer par le serveur LXC.

# 10-Scripts

## A - check-disk.sh :

Ce script à pour objectif d'envoyer un mail sur hostmaster pour avoir le délai de l'espace disque du stockage BTRFS de LXC.

Crontab -e :

```
# Check disques :  
0 06 * * * cd /usr/local/bin/ && ./check-disk.sh 2>&1
```


Emplacement du script : /usr/local/bin/check-disk.sh

```
#!/bin/bash  
(  
    echo To: hostmaster@morbihan-energies.fr  
    echo From: hostmaster@morbihan-energies.fr  
    echo Subject: Usage du disque sur le serveur lxc01.v.sdem.fr  
    echo LXC01.V.SDEM.FR - Usage du disque :  
    echo  
    sudo df -h /dev/mapper/ubuntu--vg-ubuntu--lv  
    echo  
    echo LXC01.V.SDEM.FR - Taille du stockage BTRFS de LXC01 :  
    sudo btrfs filesystem show  
    echo  
)| /usr/sbin/sendmail -fhostmaster@morbihan-energies.fr hostmaster@morbihan-energies.fr
```

# 05 - Transfert de Conteneur

## A - Transfert de Conteneur :

Voici mon exemple de configuration pour utiliser l'API LXD et le protocole Simplestreams à des fins de migration :

 image.png and or type unknown

Encore une fois, vous avez besoin de deux serveurs avec LXD configuré. J'appellerai votre machine locale serveur1 et un serveur distant j'appellerai serveur2.

## 1. Configurer le serveur :

Ces étapes sont à faire sur les 2 Serveurs LXC.

Tapez la commande suivante pour activer un accès à distance via API en indiquant l'adresse IP du serveur LXC sur lequel on est connecté.

```
lxc config set core.https_address 192.168.1.6:8443
```

Définissez le mot de passe pour le démon server2 lxd : Remplacez le MOT DE PASSE ici par le mot de passe réel.

```
lxc config set core.trust_password PASSWORD
```

## 2. Configurer un local nommé serveur1 :

Tapez la commande suivante sur le serveur1 pour ajouter le serveur2 et inversement :

```
lxc remote add lxcXX.v.sdem.fr 192.168.1.6
```

Vous pouvez lister vos télécommandes et vous verrez « server2 » répertorié comme suit :

```
lxc remote list
```

image.png  
image not found or type unknown

## 3. Transfert du conteneur :

### *Instantané 1*

Création d'une capture instantanée pour le conteneur transféré.

```
lxc snapshot {CT_NAME} {NOM_SNAP}
```

### *Transfert 1*

```
lxc copy {CT_NAME}/{NOM_SNAP} server2:{CT_NAME} --verbose
```

## 4. Script de Transfert de Conteneur

:

Voici un script permettant d'automatiser avec Salt le transfert d'un conteneur entre 2 serveurs LXC.

```
#####  
# Auteur : mkermorvant #  
# Date de modification : 20240115 #  
# #  
# Description : Script de migration de conteneur lxc #  
# #  
#####  
#!/bin/bash #  
# #
```

```
#####
# Variables :
echo""
read -p "Entrez le nom de votre Serveur Source: " SOURCE
read -p "Entrez le nom actuel de votre conteneur: " CT_NAME
read -p "Entrez le nom de votre Snapshot: " SNAP_NAME
echo ""
echo "Début de la Migration du Conteneur."
#####
# 1 - Création de la Snapshot
echo "Création de la Snapshot."
echo""
salt ${SOURCE} cmd.run "lxc snapshot ${CT_NAME} ${SNAP_NAME}"
# 2 - Information supplémentaire
salt ${SOURCE} cmd.run "lxc remote list"
read -p "Entrez le nom de votre Remote Serveur: " REMOTE_SERV
# 3 - Migration du conteneur
salt ${SOURCE} cmd.run "lxc copy ${CT_NAME}/${SNAP_NAME} ${REMOTE_SERV}:${CT_NAME} --verbose"
echo ""
# 4 - Suppression du Conteneur de Base
read -r -p "Voulez-vous supprimer le conteneur ? [y/N] " response

case "$response" in
    [yY][eE][sS][yY])
        echo "Suppression de votre premier conteneur."
        salt ${SOURCE} cmd.run "lxc stop ${CT_NAME}"
        salt ${SOURCE} cmd.run "lxc delete ${CT_NAME}"
        echo ""
        echo "Le transfère de votre conteneur LXC est maintenant fini !"
        ;;
    *)
        echo "Le transfère de votre conteneur LXC est maintenant fini !"
        ;;
esac
```

## **B - Création de Conteneur :**

# 1. Script de Création de conteneur LXC :

```
#####  
# Auteur : mkermorvant #  
# Date de modification : 20240108 #  
# #  
# Description : Script de création de conteneur lxc #  
# #  
#####  
# #  
#!/bin/bash #  
# #  
#####  
# Variables :  
echo ""  
read -p "Entrez le nom de votre Serveur LXC: " SERVER  
read -p "Entrez la distribution à installer: " DISTRIB  
read -p "Entrez la version à installer: " VERSION  
read -p "Entrez le nom de votre conteneur: " CONTENEUR  
read -p "Donner le nom de votre snapshot: " SNAP  
echo ""  
echo "Début de la création du conteneur."  
#####  
# 1 - Création du conteneur  
salt ${SERVER} cmd.run "lxc launch ${DISTRIB}:${VERSION} ${CONTENEUR}"  
echo ""  
# 2 - Création d'une Snapshot  
echo "Création d'une Snapshot."  
salt ${SERVER} cmd.run "lxc snapshot ${CONTENEUR} ${SNAP}"  
echo ""  
# 3 - Infos Conteneur  
salt ${SERVER} cmd.run "lxc info ${CONTENEUR}"  
echo ""  
echo "Votre conteneur ${CONTENEUR} est maintenant créé et opérationnelle"  
echo ""
```

