

LXC ou la virtualisation allégée

- Objet : Créer des environnements cloisonnés avec LXC sous Debian jessie
- Niveau requis : [avisé](#)
- Commentaires : *Entre machine virtuelle et chroot, votre cœur balance : choisissez LXC (Linux Containers)*
- Suivi :
 - Création par [bendia](#) le 22/02/2014
 - Mis à jour par [greenmerlin](#) le 29/10/2016
 - Testé par [bendia](#) le 21/02/2014
 - Testé par [paskal](#) le 01/03/2014
- Commentaires sur le forum : [ici](#) ¹⁾

Introduction

LXC (LinuX Container) est un système permettant d'installer plusieurs distributions et applications séparées les unes des autres (comme des machines virtuelles) mais qui partagent malgré tout le même noyau du système hôte.

Il est donc à mi-chemin entre un *chroot* et une machine virtuelle.

Attention ! Sous Debian 7 Wheezy, LXC peut ne pas fournir une isolation suffisante permettant ainsi sous certaines conditions aux systèmes invités de compromettre le système hôte.

Cette situation est en cours d'évolution avec 3.12/lxc-beta2 :

<https://www.stgraber.org/2014/01/17/lxc-1-0-unprivileged-containers/>

Installation des paquets nécessaire

Installer le paquet `lxc` qui contient les outils permettant de créer, lancer, supprimer des conteneurs LXC

```
apt-get install lxc
```

D'autres paquets sont recommandés, et notamment *debootstrap* qui permettra d'installer un autre système Debian, *bridge-utils* quand a lui vous permettra de créer des interfaces de type "pont" permettant a votre conteneur de communiquer avec l'extérieur:

```
apt-get install bridge-utils libvirt-bin debootstrap
```

Notes sur les anciennes versions de debian

Ancienne version de debian



Si vous utilisez la version stable de debian (la jessie à l'heure ou j'écris ces lignes) vous pouvez directement vous rendre au paragraphe suivant

[ICI](#)

/etc/fstab



La modification de `/etc/fstab` n'est pas nécessaire sous *Jessie* si *systemD* et le paquet *libvirt-bin* sont installés. Le montage de *cgroup* sera effectué automatiquement.

LXC utilise le mécanisme des [Control Groups](#) (les *cgroups*), une fonctionnalité du noyau Linux pour limiter, compter et isoler l'utilisation des ressources (processeur, mémoire, utilisation disque, etc.). Voir les détails sur :

- [Wikipédia \(fr\)](#)

Il est nécessaire de lister ce système de fichier virtuel dans `/etc/fstab`

```
cgroup /sys/fs/cgroup cgroup defaults 0 0
```

et monter le système de fichier :

```
mount /sys/fs/cgroup
```

Vérifier la configuration du noyau

```
lxc-checkconfig
```

[retour de la commande](#)

```
Kernel config /proc/config.gz not found, looking in other places...
Found kernel config file /boot/config-3.2.0-4-amd64
--- Namespaces ---
Namespaces: enabled
Utsname namespace: enabled
Ipc namespace: enabled
Pid namespace: enabled
User namespace: enabled
Network namespace: enabled
Multiple /dev/pts instances: enabled

--- Control groups ---
Cgroup: enabled
Cgroup clone_children flag: enabled
Cgroup device: enabled
```

```
Cgroup sched: enabled
Cgroup cpu account: enabled
Cgroup memory controller: enabled
Cgroup cpuset: enabled
```

```
--- Misc ---
```

```
Veth pair device: enabled
Macvlan: enabled
Vlan: enabled
File capabilities: enabled
```

Note : Before booting a new kernel, you can check its configuration usage : `CONFIG=/path/to/config /usr/bin/lxc-checkconfig`

Création de votre premier conteneur

LXC vient avec des *Templates*, c'est à dire des scripts permettant l'installation simple d'un système. Il en existe pour Debian, Archlinux, Fedora, OpenSuse et Ubuntu.

Créons par exemple un système, tiens au hasard, **Une Debian** 😊

```
lxc-create -n myfirstcontainer -t debian
```



le -n définit le nom de votre conteneur

Cela va nous installer une Debian Stable super minimale dans la même architecture que le système hôte (i386, AMD64 ...) dans `/var/lib/lxc/myfirstcontainer/rootfs` et un fichier de configuration `/var/lib/lxc/myfirstcontayner/config`.

Il existe quelques options pour ce *template*, notamment pour le choix de la version. Par exemple pour installer une Debian Sid :

```
lxc-create -n sid-container -t debian -- -r sid
```

Personnaliser un fichier template

Si les templates fournis ne répondent pas à vos besoin, ils peuvent néanmoins constituer un point de départ à la confection de votre modèle.

Par exemple, les modifications suivantes vous permettront d'avoir les locales françaises :

```
sed -i -e "s/locale-gen en_US.UTF-8 UTF-8/locale-gen fr_FR.UTF-8/g"
/usr/share/lxc/templates/lxc-debian
sed -i -e "s/update-locale LANG=en_US.UTF-8/update-locale
LANG=fr_FR.UTF-8/g" /usr/share/lxc/templates/lxc-debian
```

Le fichier de configuration

Un fichier par défaut est créé lors de la création du conteneur, par exemple :

[/var/lib/lxc/myfirstcontainer/config](#)

```
# Template used to create this container: /usr/share/lxc/templates/lxc-
debian
# Parameters passed to the template: -r jessie
# For additional config options, please look at lxc.container.conf(5)
lxc.network.type = none

lxc.rootfs = /var/lib/lxc/proxies/rootfs
lxc.start.auto = 1
# Common configuration
lxc.include = /usr/share/lxc/config/debian.common.conf

# Container specific configuration
lxc.mount = /var/lib/lxc/proxies/fstab
lxc.utsname = proxies
lxc.arch = i386
lxc.autodev = 1
lxc.kmsg = 0
```

Ce fichier est à adapter suivant les besoins.

Gérer et administrer vos conteneur

Démarrer votre conteneur

```
lxc-start -n myfirstcontainer -d
```

- le **-n** pour indiquer le nom du conteneur,
- le **-d** pour le lancer en arrière plan (non attaché à la console)



Sans le **-d**, il ne sera pas possible de quitter proprement le conteneur.
Utilisez `lxc-start` au premier plan seulement dans un but de debugage.

Démarrage automatique de votre conteneur (au boot de votre bécane)

Simple comme Debian

```
systemctl enable lxc
```

ensuite éditez le fichier `/var/lib/lxc/$containername/config`

[/var/lib/lxc/proxies/config](#)

```
lxc.start.auto = 1
```

puis on vérifie par un

```
lxc-ls --f
```

qui vous renvoie ça

```
NAME      STATE    IPV4  IPV6  AUTOSTART
-----
myfirstcontainer  STOPPED  -     -     YES
```

Arrêter le conteneur

Arrêter le conteneur sans arrêt propre par SysV :

```
lxc-stop -n myfirstcontainer
```

Pour un arrêt propre :

```
lxc-halt -n myfirstcontainer
```

Afficher la liste des conteneurs et leurs états

```
lxc-ls -f
```

Mettre en pause un conteneur

```
lxc-freeze -n myfirstcontainer
```

Sortir de pause un conteneur

```
lxc-unfreeze -n myfirstcontainer
```

Détruire un conteneur

```
lxc-destroy -n myfirstcontainer
```

Afficher l'état d'un conteneur et son PID

```
lxc-info -n myfirstcontainer
```

[retour de la commande](#)

```
state:    RUNNING
pid:      19879
```

Se connecter au conteneur

```
lxc-console -n myfirstcontainer
```

Le login/mot de passe est *root/root* à [changer à la première connexion](#) 😊 <code config séquence interactive>Type <Ctrl+a q> to exit the console, <Ctrl+a Ctrl+a> to enter Ctrl+a itself

```
Debian GNU/Linux 8 myfirstcontainer tty1
```

```
myfirstcontainer login: root Mot de passe : Dernière connexion : dimanche 29 novembre 2016 à 16:45:20 CET sur tty1 Linux proxies 3.16.0-4-586 #1 Debian 3.16.36-1+deb8u2 (2016-10-19) i686
```

The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

Quitter le conteneur

```
Ctrl+A+Q
```

TP associés

- [TP : Configurer des conteneurs LXC en mode Bridge](#)
- [TP : Utilisation de LXC en mode utilisateur non-privilégié](#)

Liens

- [LXC sur Le Wiki Debian](#)
- [Un topic dans le forum DF](#) 😊
- [10 articles intéressants pour lxc](#)
 - Dont: [Lancer des conteneurs sans privilèges root](#) (en)

1)

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

From:

<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:

<http://debian-facile.org/doc:autres:vm:lxc>



Last update: **01/10/2018 08:57**