

Manuel d'utilisation LXC

KostiTeam

October 2, 2016

Contents

1	Bien débiter	3
1.1	Installer LXC	3
1.1.1	Arch-linux	3
1.1.2	Debian	3
1.2	Créer un container	3
1.3	Démarrer un container	3
1.4	Configurer un container	3
1.4.1	Fichier de configuration	3
1.4.2	ifconfig, ip	4
1.5	Relier les containers avec les ponts (ou bridges)	4
1.5.1	La methode manuelle	5

1 Bien débiter

Toutes les commandes exécutées sont effectuées en root (super-utilisateur).

1.1 Installer LXC

1.1.1 Arch-linux

```
#pacman -S lxc arch-install-scripts
```

1.1.2 Debian

```
#apt-get install lxc
```

1.2 Créer un container

`#lxc-create -t download -n name`: créer un container de nom *name* en proposant la liste des images d'OS possibles à télécharger.

`#lxc-create -t download -n name -d debian -r jessie -a i386`: créer un container de nom *name* en téléchargeant une image de distribution *debian*, de release *jessie* et d'architecture *i386* (32 bits).¹

`#lxc-ls`: obtenir la liste des containers créés `-fancy` pour plus de détails.

1.3 Démarrer un container

`#lxc-start -n name -d`: démarrer le container de nom *name* en daemon (`-d`). Par défaut, aucun compte utilisateur n'est créé. Il faut donc se connecter en root.

`#lxc-attach -n name`: se connecter en root sur le container *name*. Une fois un compte utilisateur créé, il est possible de lancer un terminal sur une session.

`#lxc-console -n name -t 0`: ouvrir un écran de login sur le terminal `tty0` du container *name*.²

1.4 Configurer un container

1.4.1 Fichier de configuration

Le fichier de configuration d'un container *name* est `/var/lib/lxc/name/config`. Voici un exemple de configuration de passerelle:

¹Créer un container d'architecture 64 bits sur un host 32 bits **ne fonctionne pas**.

²**BUG**: sur certaines distributions, `-t` différent de 0 ne fonctionne pas

RTFM: lxc.container.conf

/var/lib/lxc/passerelle/config

```
# Distribution configuration
lxc.include = /usr/share/lxc/config/debian.common.conf
lxc.arch = x86_64
```

```
# Container specific configuration
lxc.rootfs = /var/lib/lxc/passerelle/rootfs
lxc.rootfs.backend = dir
lxc.utsname = passerelle
```

```
# Network configuration
lxc.network.type = veth
lxc.network.name = eth0
lxc.network.link = lxcbr0
lxc.network.flags = up
lxc.network.hwaddr = 00:16:3e:5b:0e:8f
lxc.network.ipv4 = 172.16.1.1
lxc.network.ipv6 = fec00:0:0:2::1
```

```
lxc.network.type = veth
lxc.network.name = eth1
lxc.network.link = lxcbr0
lxc.network.flags = up
lxc.network.hwaddr = 00:16:3e:5b:0e:8f
lxc.network.ipv4 = 192.168.1.1
lxc.network.ipv6 = fc00:0:0:1::1
```

Ici, le container possède deux interfaces (eth0 et eth1) avec chacune leurs adresses ipv4 (lxc.network.ipv4), ipv6 (lxc.network.ipv6) et MAC (hwaddr). lxc.network.flags indique quelle action effectuer (up active l'interface). lxc.network.type indique quelle type de virtualisation de réseau utiliser (RTFM). lxc.network.link indique l'interface à utiliser pour le vrai trafic, cette notion sera vu plus en détail par la suite.

1.4.2 ifconfig, ip

Commandes ip/ifconfig pour ipv4,ipv6

1.5 Relier les containers avec les ponts (ou bridges)

Pour relier les containers entre eux, ou même au host, LXC utilise les bridge (pont en français). Lors de la création d'un container, l'activation de chacune

de ses interfaces va créer une interface sur la machine host. Relier ces interfaces à un même pont permet de relier les containers "physiquement".

1.5.1 La methode manuelle

Il est possible de relier les containers apres les avoir lancer:

#brctl addbr br0: crée un bridge de nom *br0*

#ifconfig br0 up: active l'interface *br0*

#brctl addif br0 VETH12345: relie le bridge *br0* a l'interface virtuelle correspondante à l'interface physique *VETH12345* Voir l'autre doc pour plus de details sur le nom de l'interface

1.6 La methode automatique

Ou, il est possible de relier les containers par le fichier de configuration **attention** : le pont ne peut pas etre **créé** par le fichier de configuration, il faut le créer à la main présenté precedement:

#lxc.network.link permet de préciser le pont auquel relier l'interface créée.

donc, ici:

```
lxc.network.type = veth
lxc.network.name = eth1
lxc.network.link = lxcbr0
lxc.network.flags = up
lxc.network.hwaddr = 00:16:3e:5b:0e:8f
lxc.network.ipv4 = 192.168.1.1
lxc.network.ipv6 = fc00:0:0:1::1
```

l'interface vas etre reliée au pont lxcbr0