User Manual for LXC

KostiTeam

October 2, 2016

Contents

1 The Beginning

All commands used are launched/done as root

1.1 Install LXC

1.1.1 Arch-linux

#pacman -S lxc arch-install-scripts

1.1.2 Debian

#apt-get install lxc

1.2 Create a container

#lxc-create -t download -n cntName: create a container which name is *cntName*, by displaying a list, where the user can choose the OS he wants for the container.

#lxc-create -t download -n cntName – -d debian -r jessie -a i386: create the *cntName* container, by downloading an OS of *debian* distribution, *jessie* release, and *i386* architecture (32 bits).[1]

#lxc-ls: display the list of all the created containers; –fancy for more details.

1.3 Start a container

#lxc-start -n cntName -d: Start the *cntName* container in daemon mode *(-d)*. By default, there isn't any user's account. So, it is necessary to connect as root.

#lxc-attach -n cntName: connect as root to the *cntName* Container.
Once an user's account is created, and the container launched, it is possible to launch a terminal on a session

#lxc-console -n cntName -t 0: open a login screen on the *tty0* terminal of the *cntName* container.[2]
Or, if the container is stopped, it is also possible to launch a container on a login screen by taking off the *-d* to the lxc-start command.

[1] create a container which has a 64 bits architecture on a 32 bits host computer **dosen't work**.

[2] **BUG:** on some OS, -t **must** be equal to 0

1.4 Configure a container

1.4.1 configuration file

A container's configuration file can be found at /var/lib/lxc/<*container name*>/config.
Here is a gateway configuration example:
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 = 192.168.1.1/24
lxc.network.ipv6 = fc00:0:0:1::1/64
```

Here, the container has two interfaces (eth0 and eth1); each one has its own ipv4 (*lxc.network.ipv4*), ipv6(*lxc.network.ipv6*), and MAC(*lxc.network.hwaddr*) adresses. lxc.network.flags indicate what to do with the interface (*up* activate the interface.).
lxc.network.type indicate the type of network virtualisation to use (RTFM).
lxc.network.link indicate the bridge to use for the package trafic between containers, this will be explained later.

1.4.2 ifconfig, ip

IP commands/ifconfig for ipv4,ipv6

1.5 Join containers with bridges

To join the continers to other containers, or to the host, LXC uses bridges. When a container is launched, each "virtual" interfaces of the container will create an interface on the host. Join those interfaces to a same bridge will allow to comunicate between severals interfaces (if there is a proper network configuration)

1.5.1 The "manual" method

After launching the containers, it is possible to connect them with the bridges:
#brctl addbr br0: create a bridge which name is *br0*
#ifconfig br0 up: activate the *br0* interface
#brctl addif br0 VETH12345: Join the *br0* bridge and the *VETH12345* physical interface[3]

1.5.2 The automatic method

It is also possible to join containers by the configuration file.[4] Be carefull, if you tried to launch a container, which depends on a bridge which is not created, it will return an error.
#lxc.network.link allow an interface to connect to a specific bridge.

Exemple:
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 eth1 sera reliée (indirectement) au pont lxcbr0. the *eth1* interface will be connected to the *lxcbr0* bridge.

[3] See also the other doc to have more details
[4] **Warning :** the bridge **must** be created manually, it can't be created with the config file.