Le Protocole MQTT

Nicolas Vadkerti

2 décembre 2019

https://github.com/SlaynPool/CR_LXC-LXD/

1 Container LXC sosu Linux

1.1 Installation de LXC

Pour installer LXC, on doit suivre le TP, càd :

```
apt-get purge lxd
apt-get install lxc  lxc-templates lxc-utils bridge-utils debootstrap yum libvirt0
    libpam-cgfs
```

Listing 1 – Installation de LXC

Pour vérifier que l'installation fonctionne coorectement on peut utiliser les commandes lxc-...

```
root@ubuntu:~# lxc-checkconfig
Kernel configuration not found at /proc/config.gz; searching...
Kernel configuration found at /boot/config-4.15.0-29-generic
--- Namespaces ---
Namespaces: enabled
Utsname namespace: enabled
Ipc namespace: enabled
Pid namespace: enabled
User namespace: enabled
Network namespace: enabled

--- Control groups ---
Cgroups: enabled

Cgroup v1 mount points:
/sys/fs/cgroup/systemd
/sys/fs/cgroup/rdma
/sys/fs/cgroup/memory
/sys/fs/cgroup/cpuset
/sys/fs/cgroup/net_cls,net_prio
/sys/fs/cgroup/devices
/sys/fs/cgroup/cpu,cpuacct
/sys/fs/cgroup/hugetlb
/sys/fs/cgroup/blkio
/sys/fs/cgroup/pids
/sys/fs/cgroup/freezer
/sys/fs/cgroup/perf_event

Cgroup v2 mount points:
/sys/fs/cgroup/unified

Cgroup v1 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, not loaded
Macvlan: enabled, not loaded
Vlan: enabled, not loaded
Bridges: enabled, loaded
Advanced netfilter: enabled, not loaded
CONFIG_NF_NAT_IPV4: enabled, loaded
```

```
CONFIG_NF_NAT_IPV6: enabled, not loaded
CONFIG_IP_NF_TARGET_MASQUERADE: enabled, loaded
CONFIG_IP6_NF_TARGET_MASQUERADE: enabled, not loaded
CONFIG_NETFILTER_XT_TARGET_CHECKSUM: enabled, loaded
CONFIG_NETFILTER_XT_MATCH_COMMENT: enabled, not loaded
FUSE (for use with lxcfs): enabled, not loaded

--- Checkpoint/Restore ---
checkpoint restore: enabled
CONFIG_FHANDLE: enabled
CONFIG_EVENTFD: enabled
CONFIG_EPOLL: enabled
CONFIG_UNIX_DIAG: enabled
CONFIG_INET_DIAG: enabled
CONFIG_PACKET_DIAG: enabled
CONFIG_NETLINK_DIAG: enabled
File capabilities:

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

Listing 2 – Verification de LXC

Il ne semble pas avoir de soucis, ce qui est plutot bon signe pour la suite du TP

Si l'on regarde dans /usr/share/lxc/templates/ on peut voir toutes les distributions que l'on va pouvoir contenariser :

```
root@ubuntu:~# ls  /usr/share/lxc/templates/
lxc-alpine     lxc-fedora-legacy  lxc-sabayon
lxc-altlinux   lxc-gentoo   lxc-slackware
lxc-archlinux  lxc-local    lxc-sparclinux
lxc-busybox    lxc-oci      lxc-sshd
lxc-centos     lxc-openmandriva   lxc-ubuntu
lxc-cirros     lxc-opensuse   lxc-ubuntu-cloud
lxc-debian     lxc-oracle   lxc-voidlinux
lxc-download   lxc-plamo
lxc-fedora     lxc-pld
```

Listing 3 – listes des distributions contenarisables

1.2 Création d'un container LXC Debian stretch.

Pour cela, il faut :

```
root@ubuntu:~# lxc-create -t debian -n debian-j1 -- -r stretch -a amd64
ebootstrap is /usr/sbin/debootstrap
Checking cache download in /var/cache/lxc/debian/rootfs-stretch-amd64 ...
gpg: key 7638D0442B90D010: 4 signatures not checked due to missing keys
gpg: key 7638D0442B90D010: "Debian Archive Automatic Signing Key (8/jessie) <
    ftpmaster@debian.org>" not changed
gpg: Total number processed: 1
gpg:              unchanged: 1
Downloading debian minimal ...
I: Retrieving InRelease
I: Retrieving Release
...

root@ubuntu:~# lxc-info debian-j1
Name:           debian-j1
State:          STOPPED
root@ubuntu:~# lxc-start debian-j1
root@ubuntu:~# lxc-info debian-j1
Name:           debian-j1
State:          RUNNING
PID:            19922
CPU use:        0.12 seconds
BlkIO use:      132.00 KiB
Memory use:     13.07 MiB
KMem use:       2.03 MiB
Link:           veth19GBKG
 TX bytes:      858 bytes
 RX bytes:      780 bytes
```

Listing 4 – premier container

Comme on peux le voir, nous venons de lancer notre premier container via lxc.

Si l'on recreer un nouveau container avec un nom different, la vitesse est bien plus rapide. En effet, il ne retelecharge pas toutes l'image, mais copie simplement la précédente. Le cache est ici : `var/cache/lxc/debian/rootfs-stretch-amd64`

Pour déployer d'autre distributions comme centos on peut aussi utiliser LXC

```
lxc-create -t centos -n centos
```

Listing 5 – LXC centos

```
# Lister les containers
$ lxc-ls
centos    debian-j1 debian-j2

# Demarrer un containers
$ lxc-start debian-j1

# Stopper Un containers
lxc-stop debian-j1

# Redemarrer en mode Deamon

lxc-start -d debian-J1

# S'attacher a ce container

root@ubuntu:~# lxc—attach debian—j1
root@debian-j1:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
9: eth0@if10: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 00:16:3e:aa:9b:c3 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.0.3.212/24 brd 10.0.3.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::216:3eff:feaa:9bc3/64 scope link
       valid_lft forever preferred_lft forever
root@debian-j1:~#


# Voir le PID du Container et les processus fils

root@ubuntu:~# lxc—info debian—j1
Name:           debian-j1
State:          RUNNING
PID:            6196
IP:             10.0.3.212
CPU use:        0.25 seconds
BlkIO use:      2.02 MiB
Memory use:     15.79 MiB
KMem use:       2.16 MiB
Link:           veth23209A
 TX bytes:      1.55 KiB
 RX bytes:      1.86 KiB
 Total bytes:   3.42 KiB

root@ubuntu:~# ps —ejH 6196

 6192  6192  6192 ?        Ss     0:00   [lxc monitor] /var/lib/lxc debian-j1
 6196  6196  6196 ?        Ss     0:00     /sbin/init
 6252  6252  6252 ?        Ss     0:00       /lib/systemd/systemd-journald
 6295  6295  6295 ?        Ss     0:00       /sbin/dhclient -4 -v -pf /run/dhclient.eth0.pid -lf /var/lib/dhcp/dhclient.eth0.leases -I -df /var/lib/dhcp/dhclient6.eth0.leases eth0
 6331  6331  6331 pts/3    Ss+    0:00       /sbin/agetty --noclear --keep-baud console 115200,38400,9600 vt220
 6332  6332  6332 ?        Ss     0:00       /usr/sbin/sshd -D
```

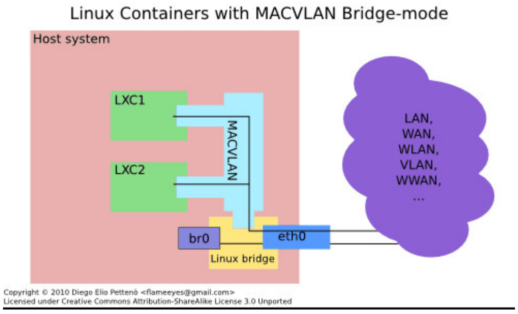
Listing 6 – Les commandes utiles

```
# Pour limiter l'utilisation Memoire d'un Container
# Dans /var/lib/lxc/$CONTAINER/config
lxc.cgroup.memory.limit_in_bytes = 512M

# Pour limiter le Swap du container
memory.memsw.limit_in_bytes= 1G

# Pour verifier, on peut utiliser la commande stress dans le container, et verifier que
    l'on ne peut allouer plus de 512M
stress --vm 1 --vm-bytes 600M


# Installer apache2
root@ubuntu:~# lxc-attach  debian-j1
root@debian-j1:~# apt install apache2
Lecture des listes de paquets... Fait
...
# Faire un autostart
vim /var/lib/lxc/$CONTAINER/config
lxc.start.auto = 1


# Figer, relancez le container
root@ubuntu:~# lxc-freeze debian-j1
root@ubuntu:~# lxc-info debian-j1
Name:           debian-j1
State:          FROZEN
PID:            1105
IP:             10.0.3.212
CPU use:        0.77 seconds
BlkIO use:      21.57 MiB
Memory use:     48.55 MiB
KMem use:       4.82 MiB
Link:           vethG27FGY
 TX bytes:      2.10 KiB
 RX bytes:      16.55 KiB
 Total bytes:   18.64 KiB
root@ubuntu:~# lxc-unfreeze debian-j1
root@ubuntu:~# lxc-info debian-j1
Name:           debian-j1
State:          RUNNING
PID:            1105
IP:             10.0.3.212
CPU use:        0.77 seconds
BlkIO use:      21.57 MiB
Memory use:     48.55 MiB
KMem use:       4.82 MiB
Link:           vethG27FGY
 TX bytes:      2.10 KiB
 RX bytes:      16.88 KiB
 Total bytes:   18.98 KiB

# Cloner un container
Il faut que la machine soit stoper puis

$: lxc-copy -n debian-j1 -N clonedebian
```

Listing 7 – Les commandes utiles

1.3 Comment le container est-il connecté au réseau ?

Le container est relié au reseau grâce à un Bridge Virtuel. On se rend compte grâce à la commande ip a que lxc monte des interfaces réseaux virtuels Le fichier relatif aux reseaux est /etc/default/lxc-net :



Copyright © 2010 Diego Elio Pettenò <flameeyes@gmail.com>
Licensed under Creative Commons Attribution-ShareAlike License 3.0 Unported

Figure 1 – Network Schema

```
root@ubuntu:~# cat /etc/default/lxc-net
# This file is auto-generated by lxc.postinst if it does not
# exist.  Customizations will not be overridden.
# Leave USE_LXC_BRIDGE as "true" if you want to use lxcbr0 for your
# containers.  Set to "false" if you’ll use virbr0 or another existing
# bridge, or mavlan to your host’s NIC.
USE_LXC_BRIDGE="true"

# If you change the LXC_BRIDGE to something other than lxcbr0, then
# you will also need to update your /etc/lxc/default.conf as well as the
# configuration (/var/lib/lxc/<container>/config) for any containers
# already created using the default config to reflect the new bridge
# name.
# If you have the dnsmasq daemon installed, you’ll also have to update
# /etc/dnsmasq.d/lxc and restart the system wide dnsmasq daemon.
LXC_BRIDGE="lxcbr0"
LXC_ADDR="10.0.3.1"
LXC_NETMASK="255.255.255.0"
LXC_NETWORK="10.0.3.0/24"
LXC_DHCP_RANGE="10.0.3.2,10.0.3.254"
LXC_DHCP_MAX="253"
# Uncomment the next line if you’d like to use a conf-file for the lxcbr0
# dnsmasq.  For instance, you can use ’dhcp-host=mail1,10.0.3.100’ to have
# container ’mail1’ always get ip address 10.0.3.100.
#LXC_DHCP_CONFILE=/etc/lxc/dnsmasq.conf

# Uncomment the next line if you want lxcbr0’s dnsmasq to resolve the .lxc
# domain.  You can then add "server=/lxc/10.0.3.1’ (or your actual $LXC_ADDR)
# to your system dnsmasq configuration file (normally /etc/dnsmasq.conf,
# or /etc/NetworkManager/dnsmasq.d/lxc.conf on systems that use NetworkManager).
# Once these changes are made, restart the lxc-net and network-manager services.
# ’container1.lxc’ will then resolve on your host.
#LXC_DOMAIN="lxc"
```

Listing 8 – lxc-net default

Vu du Container, cela ressemble a ceci :

```
9: eth0@if10: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group
    default qlen 1000
    link/ether 00:16:3e:aa:9b:c3 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.0.3.212/24 brd 10.0.3.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::216:3eff:feaa:9bc3/64 scope link
       valid_lft forever preferred_lft forever
```

Listing 9 – Configuration Réseau

2 Création des CGROUPS

2.1 Les xterm consomateurs

```
top - 17:19:34 up 19 min,  2 users,  load average: 1,42, 0,58, 0,32
Tasks:  85 total,   3 running,  82 sleeping,   0 stopped,   0 zombie
%Cpu(s): 23,3 us, 76,7 sy,  0,0 ni,  0,0 id,  0,0 wa,  0,0 hi,  0,0 si,  0,0 st
MiB Mem :    987,3 total,    156,4 free,     97,4 used,    733,6 buff/cache
MiB Swap:    488,0 total,    481,0 free,      7,0 used.    723,7 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
25488 user      20   0    5272    748    688 R  49,8   0,1   0:50.58 md5sum
25490 user      20   0    5272    752    688 R  49,8   0,1   0:28.06 md5sum
24206 user      20   0   17024   5124   4028 S   0,3   0,5   0:00.58 sshd
```

Listing 10 – top

On peut voir que les deux processus prennent 50 % du CPU de la vm environ.

Pour limiter l'utilisation de ou augmenter l'utilisation du CPU, on peut utiliser les cgroupes comme ceci

```
#!/bin/bash


cgcreate -g cpu,cpuset:/quatrevingtpourcentcpu
cgcreate -g cpu,cpuset:/vingtpourcent
cpucgset -r cpu.shares=2 vingtpourcent
cpucgset -r cpu.shares=8 quatrevingtpourcent
cpucgget -r cpu.shares quatrevingtpourcent
cpucgget -r cpu.shares vingtpourcent
cpucgexec -g cpu:/quatrevingtpourcentcpu
xterm -bg orange -e "md5sum /dev/urandom"  &
cgexec -g cpu:/vingtpourcentcpu xterm -bg blue -e "md5sum /dev/urandom"  &
top -d2
```

Listing 11 – Cgroupe set

Pour utiliser cgexec, il faut etre root, et comme root ne peut utiliser de session graphique en temps normal, il faut modifier etc/pam.d/su et /etc/pam.d/su-l en rajoutant "latex veut pas" pour que ca fonctionne.

On

```
top - 17:35:13 up 34 min,  2 users,  load average: 0,56, 0,41, 0,56
Tasks:  86 total,   4 running,  82 sleeping,   0 stopped,   0 zombie
%Cpu(s): 28,3 us, 70,9 sy,  0,0 ni,  0,0 id,  0,0 wa,  0,0 hi,  0,8 si,  0,0 st
MiB Mem :    987,3 total,    144,6 free,     99,5 used,    743,3 buff/cache
MiB Swap:    488,0 total,    481,2 free,      6,8 used.    721,4 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
25672 root      20   0    5272    752    688 R  79,8   0,1   0:31.67 md5sum
25676 root      20   0    5272    688    624 R  19,9   0,1   0:02.67 md5sum
    1 root      20   0  105164  10112   7500 S   0,0   1,0   0:03.00 systemd
```

Listing 12 – top