

Setting up LXC containers with Debian 7 Wheezy from scratch

LXC - container based virtualisation from scratch on Debian Wheezy - a slightly messy guide to start with

Forewords: I was trying finish this for nearly half a year. Sorry if some of the things are out of date for now.

For years I've been looking for the most effective virtualisation fitting my requirements. I'd tried VMware, Xen, even used - as user - Virtuozzo until I recently came across LXC^[1]. Even though LXC is not new, a lot of new shiny projects are aiming to utilise it a bit better, for example Docker^[2]

What I want is a lightweight semi-virtualisation: a private, virtual network, with virtual containers for a specific task, with backup, clone and export options. This is exactly what LXC is good for.

I'm using Debian Wheezy even though 3.10 kernel is highly recommended for LXC.

DISCLAIMER: This is not a copy-paste tutorial, the process is way more complicated than that.

Install required tools

```
apt-get install lxc bridge-utils debootstrap
```

```
bash
```

Mount cgroups

add to `/etc/fstab`:

```
cgroup /cgroup cgroup
defaults 0 0
```

```
mkdir /cgroups
mount cgroups
```

bash

Create virtual network

This is not the best way to create the network interface & make it persistent, but all other ways have failed for me so far for this specific case. The internal network will be 192.168.42.0/24, the host will have 192.168.42.1 and the first guest will be 192.168.42.10. The external address will be indicated with 10.0.0.1. Please adjust these according to your needs.

Add these to `/etc/rc.local`:

```
# script to setup a natted network for lxc guests
CMD_BRCTL=/sbin/brctl
CMD_IFCONFIG=/sbin/ifconfig
CMD_IPTABLES=/sbin/iptables
CMD_ROUTE=/sbin/route
NETWORK_BRIDGE_DEVICE_NAT=lxc-bridge-nat
HOST_NETDEVICE=eth0
PRIVATE_GW_NAT=192.168.42.1
PRIVATE_NETMASK=255.255.255.0
PUBLIC_IP=10.0.0.1
LXC_GUEST_NETWORK=192.168.42.0/24
LXC_GUEST1_IP=192.168.42.10
LXC_GUEST1_EXT_SSH_PORT=2222

${CMD_BRCTL} addbr ${NETWORK_BRIDGE_DEVICE_NAT}
${CMD_BRCTL} setfd ${NETWORK_BRIDGE_DEVICE_NAT} 0
${CMD_IFCONFIG} ${NETWORK_BRIDGE_DEVICE_NAT} $
{PRIVATE_GW_NAT} netmask ${PRIVATE_NETMASK} promisc up
${CMD_IPTABLES} -t nat -A POSTROUTING -o ${HOST_NETDEVICE} -j
MASQUERADE
${CMD_IPTABLES} -t nat -A POSTROUTING -d ${LXC_GUEST_NETWORK}
-o eth0 -j SNAT --to-source ${PUBLIC_IP}
${CMD_IPTABLES} -t nat -A PREROUTING -d ${PUBLIC_IP} -p tcp -
m tcp --dport ${LXC_GUEST1_EXT_SSH_PORT} -j DNAT --to-
destination ${LXC_GUEST1_IP}:22

echo 1 > /proc/sys/net/ipv4/ip_forward
```

bash

A useful source of this setup: [https://wiki.debian.org/LXC/SimpleBridge%5B%5E3%5D](https://wiki.debian.org/LXC/SimpleBridge%5B%5E3%5D_)

optinal: logical volume for guest

If you're running LVM on the host (it can make things easy & secure), you can create a new lv per guest:

```
LXC_GUEST1_NAME=lxc-1
lvcreate -L 16G -n ${LXC_GUEST1_NAME} vg0
mkfs.ext4 /dev/vg0/${LXC_GUEST1_NAME}
mkdir /lxc/${LXC_GUEST1_NAME}
echo -e "/dev/vg0/${LXC_GUEST1_NAME} /lxc/${LXC_GUEST1_NAME} ext4 defaults 0 0n" >> /etc/fstab
mount -a
```

bash

debootstrap the container (install the bare operating system)

```
LXC_GUEST1_NAME=lxc-1
mkdir -p /lxc/${LXC_GUEST1_NAME}
debootstrap --verbose --include ifupdown,locales,netbase,net-
tools,iproute,openssh-server,vim wheezy /lxc/$
{LXC_GUEST1_NAME} http://http.debian.net/debian/
```

bash

Edit devices, inittab, configuration for the container

Edit `/lxc/${LXC_GUEST1_NAME}/etc/inittab` as follows:

bash

```
# /etc/inittab: init(8) configuration.
# $Id: inittab,v 1.91 2002/01/25 13:35:21 miquels Exp $

# The default runlevel.
id:2:initdefault:

# Boot-time system configuration/initialization script.
# This is run first except when booting in emergency (-b)
mode.
si::sysinit:/etc/init.d/rcS

# What to do in single-user mode.
~:S:wait:/sbin/sulogin

# /etc/init.d executes the S and K scripts upon change
# of runlevel.
#
# Runlevel 0 is halt.
10:0:wait:/etc/init.d/rc 0
# Runlevel 1 is single-user.
11:1:wait:/etc/init.d/rc 1
# Runlevels 2-5 are multi-user.
12:2:wait:/etc/init.d/rc 2
13:3:wait:/etc/init.d/rc 3
14:4:wait:/etc/init.d/rc 4
15:5:wait:/etc/init.d/rc 5
# Runlevel 6 is reboot.
16:6:wait:/etc/init.d/rc 6
# Normally not reached, but fallthrough in case of emergency.
z6:6:respawn:/sbin/sulogin

# What to do when CTRL-ALT-DEL is pressed.
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now
```

```
# /sbin/getty invocations for the runlevels.
#
# The "id" field MUST be the same as the last
# characters of the device (after "tty").
#
# Format:
#   :::
#
1:2345:respawn:/sbin/getty 38400 console
```

Create the devices needed in the container

```
ROOT=/lxc/${LXC_GUEST1_NAME}
DEV=${ROOT}/dev
mv ${DEV} ${DEV}.old
mkdir -p ${DEV}
mknod -m 666 ${DEV}/null c 1 3
mknod -m 666 ${DEV}/zero c 1 5
mknod -m 666 ${DEV}/random c 1 8
mknod -m 666 ${DEV}/urandom c 1 9
mkdir -m 755 ${DEV}/pts
mkdir -m 1777 ${DEV}/shm
mknod -m 666 ${DEV}/tty c 5 0
mknod -m 600 ${DEV}/console c 5 1
mknod -m 666 ${DEV}/tty0 c 4 0
mknod -m 666 ${DEV}/full c 1 7
mknod -m 600 ${DEV}/initctl p
mknod -m 666 ${DEV}/ptmx c 5 2
```

bash

Create lxc configuration file in `/var/lib/lxc/${LXC_GUEST1_NAME}/config`

```
# name
lxc.utsname = ${LXC_GUEST1_NAME}

# network
lxc.network.type = veth
lxc.network.flags = up
```

apache

```
lxc.network.link = lxc-br
lxc.network.name = eth0
lxc.network.hwaddr = 00:FF:12:34:56:78
lxc.network.ipv4 = ${LXC_GUEST1_IP}/24
# lxc.network.ipv6 =

# pts
lxc.tty = 2
lxc.pts = 1024

# fs
lxc.rootfs = /lxc/${LXC_GUEST1_NAME}

# devices
lxc.cgroup.devices.deny = a

# /dev/null and zero
lxc.cgroup.devices.allow = c 1:3 rwm # dev/null
lxc.cgroup.devices.allow = c 1:5 rwm # dev/zero

# consoles
lxc.cgroup.devices.allow = c 5:1 rwm # dev/console
lxc.cgroup.devices.allow = c 5:0 rwm # dev/tty
lxc.cgroup.devices.allow = c 4:0 rwm # dev/tty0

# /dev/{,u}random
lxc.cgroup.devices.allow = c 1:9 rwm # dev/urandom
lxc.cgroup.devices.allow = c 1:8 rwm # dev/random
lxc.cgroup.devices.allow = c 136:* rwm # dev/pts/*
lxc.cgroup.devices.allow = c 5:2 rwm # dev/pts/ptmx

# rtc
lxc.cgroup.devices.allow = c 254:0 rwm

# mount points
lxc.mount.entry = devpts /lxc/${LXC_GUEST1_NAME}/dev/pts
devpts newinstance,ptmxmode=0666,nosuid,noexec 0 0
lxc.mount.entry = proc /lxc/${LXC_GUEST1_NAME}/proc proc
nosuid,noexec,nodev 0 0
lxc.mount.entry = sysfs /lxc/${LXC_GUEST1_NAME}/sys sysfs
nosuid,nodev,noexec 0 0
```

```
lxc.mount.entry = tmpfs /lxc/${LXC_GUEST1_NAME}/dev/shm tmpfs  
nosuid,nodev,noexec,size=64m 0 0
```

Boot the container

```
lxc start -n ${LXC_GUEST1_NAME} -d
```

apache

For initial setup, you might need the option without "-d", that will land you in the console of the container.

Links

1. <http://lxc.sourceforge.net/>
2. <http://www.docker.io/>

Created by Peter Molnar <mail@petermolnar.net>, published at 2014-03-26 14:21 UTC, last modified at 2021-10-31 15:57 UTC , to canonical URL <https://petermolnar.net/article/lxc-on-debian-wheezy/> , licensed under CC-BY-4.0 .