

Documentación LXC

García Silva Adán Alberto

May 20, 2016

1 Porqué LXC

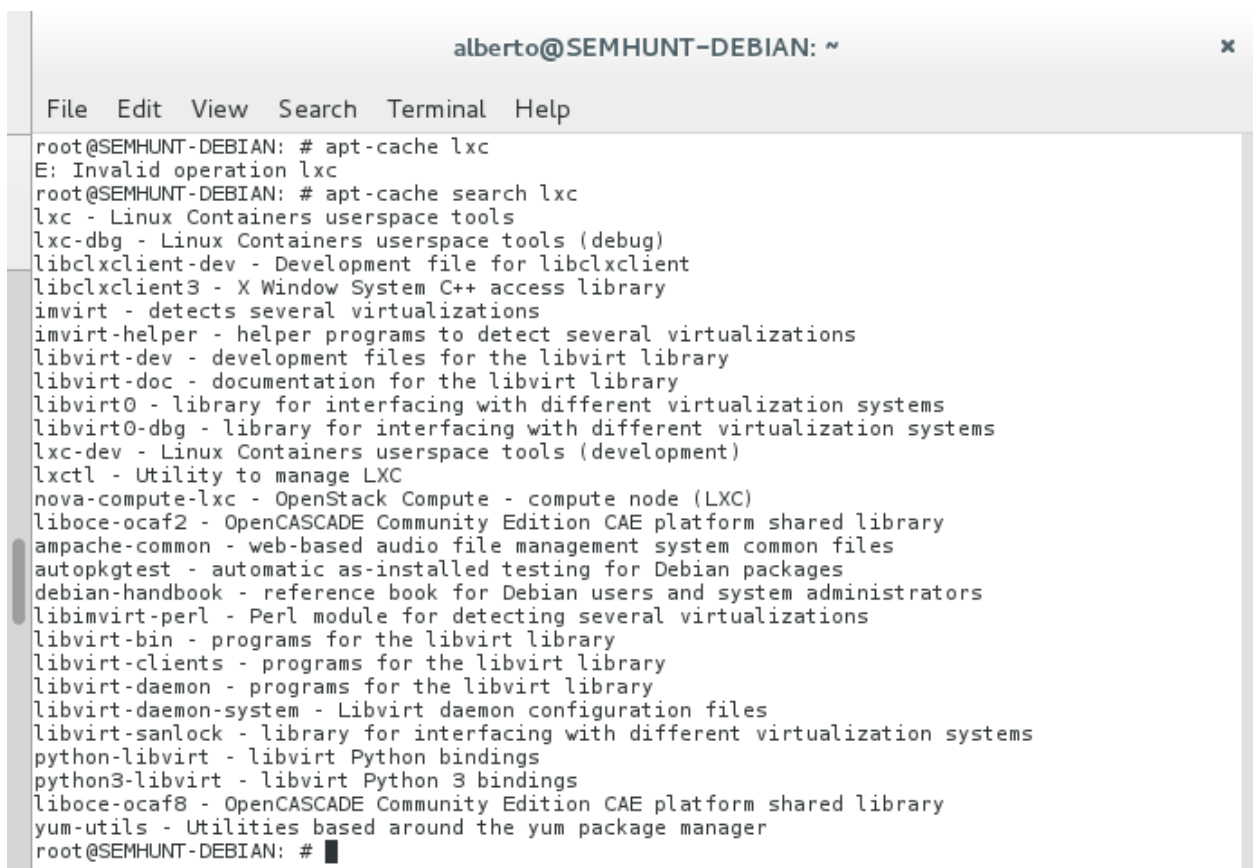
Hoy en día la tecnología de Linux Containers (LXC) te permite crear contenedores que son servidores linux aislados, en tu máquina Linux real, estos contenedores comparten el kernel con la máquina principal. Es como una virtualización muy ligera, tan ligera que realmente no hay virtualización en absoluto por llamarlo así, y por lo tanto no tiene un impacto negativo en el rendimiento. En este artículo explicaré el funcionamiento de Linux Containers en base a experiencia documentada y crearé un contenedor observando el comportamiento del mismo en el entorno.

La tecnología LXC utiliza Linux kernel control groups (cgroups) y NameSpaces (espacios de nombres) para proporcionar este aislamiento. Cabe mencionar que

2 Instalación LXC

Buscamos los paquetes disponibles para LXC mediante apt, obteniendo lo siguiente:

```
sudo apt-cache search lxc.
```



```
alberto@SEMHUNT-DEBIAN: ~
File Edit View Search Terminal Help
root@SEMHUNT-DEBIAN: # apt-cache lxc
E: Invalid operation lxc
root@SEMHUNT-DEBIAN: # apt-cache search lxc
lxc - Linux Containers userspace tools
lxc-dbg - Linux Containers userspace tools (debug)
libclxclient-dev - Development file for libclxclient
libclxclient3 - X Window System C++ access library
imvirt - detects several virtualizations
imvirt-helper - helper programs to detect several virtualizations
libvirt-dev - development files for the libvirt library
libvirt-doc - documentation for the libvirt library
libvirt0 - library for interfacing with different virtualization systems
libvirt0-dbg - library for interfacing with different virtualization systems
lxc-dev - Linux Containers userspace tools (development)
lxcctl - Utility to manage LXC
nova-compute-lxc - OpenStack Compute - compute node (LXC)
liboce-ocaf2 - OpenCASCADE Community Edition CAE platform shared library
ampache-common - web-based audio file management system common files
autopkgtest - automatic as-installed testing for Debian packages
debian-handbook - reference book for Debian users and system administrators
libimvirt-perl - Perl module for detecting several virtualizations
libvirt-bin - programs for the libvirt library
libvirt-clients - programs for the libvirt library
libvirt-daemon - programs for the libvirt library
libvirt-daemon-system - Libvirt daemon configuration files
libvirt-sanlock - library for interfacing with different virtualization systems
python-libvirt - libvirt Python bindings
python3-libvirt - libvirt Python 3 bindings
liboce-ocaf8 - OpenCASCADE Community Edition CAE platform shared library
yum-utils - Utilities based around the yum package manager
root@SEMHUNT-DEBIAN: #
```

Figure 1: Búsqueda de Paquetes LXC

Posteriormente se procede a instalar lxc:

```
sudo apt-get install lxc bridge-utils libvirt-bin debootstrap.
```



```
alberto@SEMHUNT-DEBIAN: ~
File Edit View Search Terminal Help

root@SEMHUNT-DEBIAN: # apt-get install lxc bridge-utils libvirt-bin debootstrap
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  augeas-lenses dmeventd ebttables ethtool hdparm ipxe-qemu libapparmor1 libaugeas0
  libdevmapper-event1.02.1 libfdt1 libiscsi2 liblvm2cmd2.02 libnetcf1 libnumal librados2
  librbd1 libreadline5 libseccomp2 libspice-server1 libvdeplug2 libvirt-clients
  libvirt-daemon libvirt-daemon-system libvirt0 libx86-1 libxen-4.4 libxenstore3.0 lvm2
  netcat-openbsd pm-utils powermgmt-base qemu-kvm qemu-system-common qemu-system-x86
  qemu-utils rsync seabios sharutils vbtool
Suggested packages:
  augeas-doc apmd augeas-tools radvd auditd systemtap apparmor thin-provisioning-tools lua5.2
  cpufrequtils radeontool samba vde2 sgabios ovmf
The following NEW packages will be installed:
  augeas-lenses bridge-utils debootstrap dmeventd ebttables ethtool hdparm ipxe-qemu
  libapparmor1 libaugeas0 libdevmapper-event1.02.1 libfdt1 libiscsi2 liblvm2cmd2.02 libnetcf1
  libnumal librados2 librbd1 libreadline5 libseccomp2 libspice-server1 libvdeplug2
  libvirt-bin libvirt-clients libvirt-daemon libvirt-daemon-system libvirt0 libx86-1
  libxen-4.4 libxenstore3.0 lvm2 lxc netcat-openbsd pm-utils powermgmt-base qemu-kvm
  qemu-system-common qemu-system-x86 qemu-utils rsync seabios sharutils vbtool
0 upgraded, 43 newly installed, 0 to remove and 0 not upgraded.
Need to get 16.5 MB of archives.
After this operation, 59.4 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://security.debian.org/ jessie/updates/main libspice-server1 i386 0.12.5-1+deb8u2 [48
4 kB]
Get:2 http://http.us.debian.org/debian/ stable/main libdevmapper-event1.02.1 i386 2:1.02.90-2.2
[35.7 kB]
Get:3 http://http.us.debian.org/debian/ stable/main libiscsi2 i386 1.12.0-2 [55.8 kB]
Get:4 http://http.us.debian.org/debian/ stable/main dmeventd i386 2:1.02.90-2.2 [51.5 kB]
Get:5 http://http.debian.net/debian/ wheezy/main libx86-1 i386 1.1+ds1-10 [85.4 kB]
Get:6 http://security.debian.org/ jessie/updates/main lxc i386 1:1.0.6-6+deb8u2 [637 kB]
Get:7 http://http.debian.net/debian/ wheezy/main netcat-openbsd i386 1.105-7 [39.3 kB]
Get:8 http://http.us.debian.org/debian/ stable/main liblvm2cmd2.02 i386 2.02.111-2.2 [552 kB]
Get:9 http://http.us.debian.org/debian/ stable/main libnumal i386 2.0.10-1 [33.7 kB]
Get:10 http://http.us.debian.org/debian/ stable/main librados2 i386 0.80.7-2+deb8u1 [1,666 kB]
Get:11 http://http.us.debian.org/debian/ stable/main librbd1 i386 0.80.7-2+deb8u1 [381 kB]
Get:12 http://http.us.debian.org/debian/ stable/main libreadline5 i386 5.2+dfsg-2 [146 kB]
Get:13 http://http.us.debian.org/debian/ stable/main libseccomp2 i386 2.1.1-1 [26.8 kB]
Get:14 http://http.us.debian.org/debian/ stable/main libapparmor1 i386 2.9.0-3 [60.8 kB]
Get:15 http://http.us.debian.org/debian/ stable/main libvirt0 i386 1.2.9-9+deb8u2 [2,975 kB]
Get:16 http://http.us.debian.org/debian/ stable/main libvirt-clients i386 1.2.9-9+deb8u2 [529 k
B]
Get:17 http://http.us.debian.org/debian/ stable/main augeas-lenses all 1.2.0-0.2+deb8u1 [335 kB
]
Get:18 http://http.us.debian.org/debian/ stable/main libaugeas0 i386 1.2.0-0.2+deb8u1 [268 kB]
Get:19 http://http.us.debian.org/debian/ stable/main libnetcf1 i386 1:0.2.3-4.1 [47.7 kB]
Get:20 http://http.us.debian.org/debian/ stable/main libvirt-daemon i386 1.2.9-9+deb8u2 [1,855
kB]
52% [20 libvirt-daemon 103 kB/1,855 kB 6%] 421 kB/s 18s
```

Figure 2: Instalación LXC

CGROUPS es una característica del kernel necesaria para que LXC corra apropiadamente. Comenzaremos agregandolos siguiente al archivo `/etc/fstab` y montarlos:

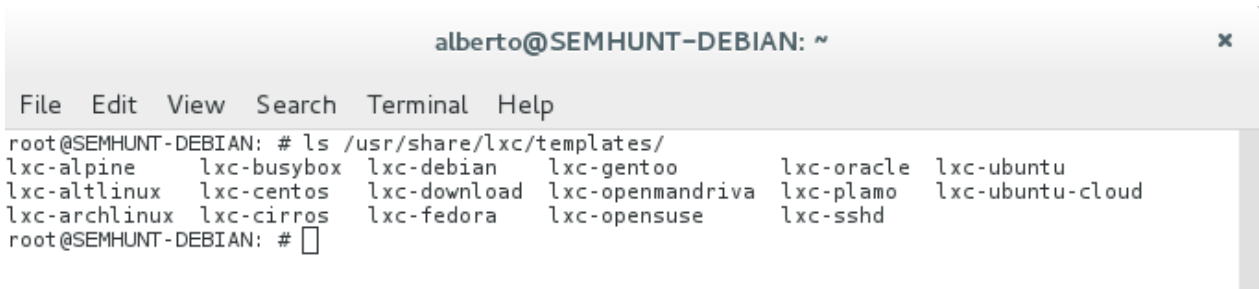
```
cgroup /sys/fs/cgroup cgroup defaults 0 0.
```

```
sudo mount /sys/fs/cgroup.
```

3 Creación de un Contenedor

Podemos revisar que templates se encuentran disponibles en la version LXC instalada.

```
ls /usr/share/lxc/templates.
```

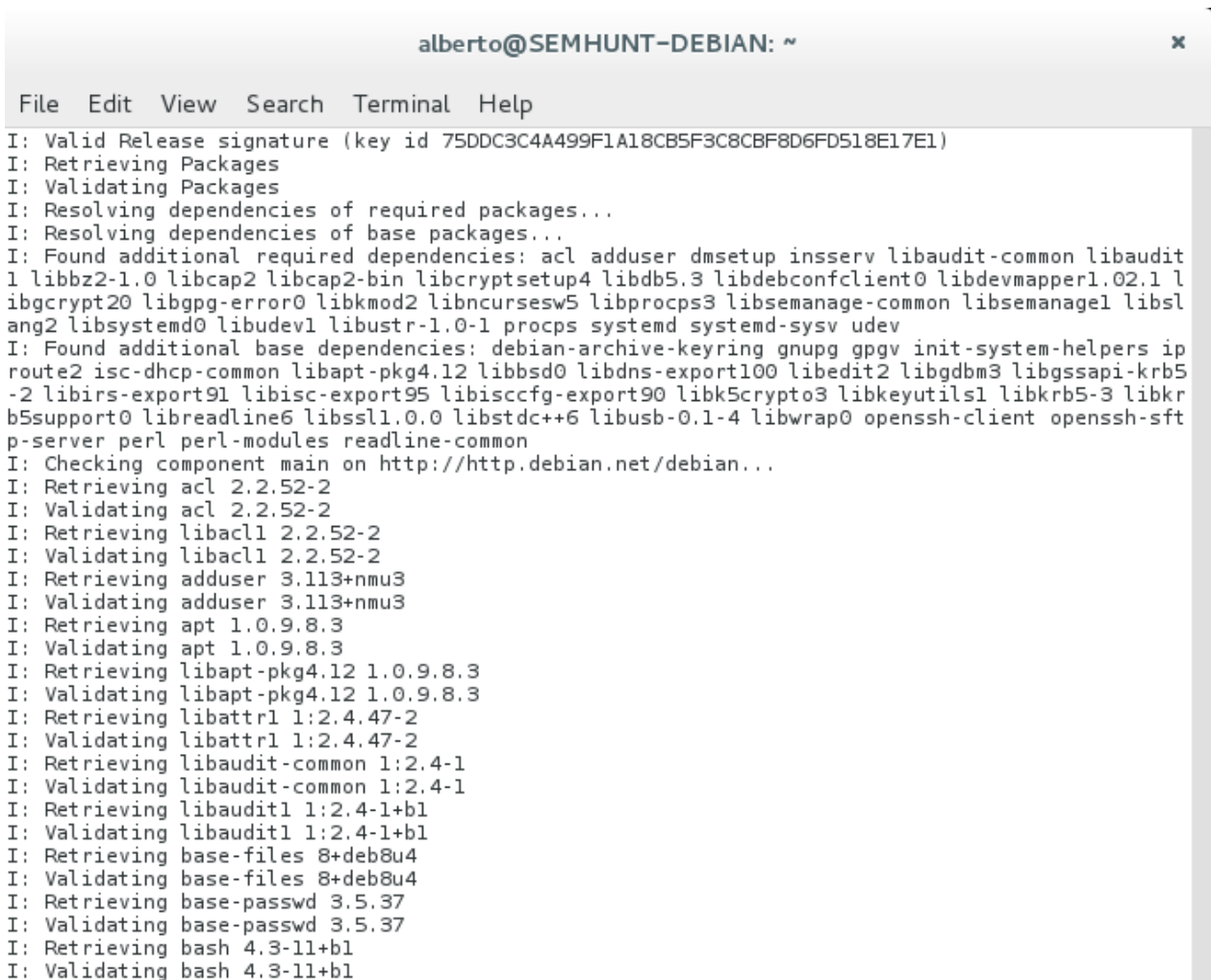


```
alberto@SEMHUNT-DEBIAN: ~  
File Edit View Search Terminal Help  
root@SEMHUNT-DEBIAN: # ls /usr/share/lxc/templates/  
lxc-alpine      lxc-busybox    lxc-debian     lxc-gentoo     lxc-oracle     lxc-ubuntu  
lxc-altlinux   lxc-centos     lxc-download   lxc-openmandriva lxc-plamo      lxc-ubuntu-cloud  
lxc-archlinux  lxc-cirros     lxc-fedora     lxc-opensuse   lxc-sshd  
root@SEMHUNT-DEBIAN: #
```

Figure 3: Templates LXC

Ahora se deberá ejecutar la siguiente sentencia para cada contenedor que se desee crear. Hagámos nuestro primer contenedor:

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



```
alberto@SEMHUNT-DEBIAN: ~  
File Edit View Search Terminal Help  
I: Valid Release signature (key id 75DDC3C4A499F1A18CB5F3C8CBF8D6FD518E17E1)  
I: Retrieving Packages  
I: Validating Packages  
I: Resolving dependencies of required packages...  
I: Resolving dependencies of base packages...  
I: Found additional required dependencies: acl adduser dmsetup insserv libaudit-common libaudit  
1 libbz2-1.0 libcap2 libcap2-bin libcryptsetup4 libdb5.3 libdebconfclient0 libdevmapper1.02.1 l  
ibgcrypt20 libgpg-error0 libkmod2 libncursesw5 libprocps3 libsemanage-common libsemanage1 libsl  
ang2 libsystemd0 libudev1 libustr-1.0-1 procps systemd systemd-sysv udev  
I: Found additional base dependencies: debian-archive-keyring gnupg gpgv init-system-helpers ip  
route2 isc-dhcp-common libapt-pkg4.12 libbsd0 libdns-export100 libedit2 libgdbm3 libgssapi-krb5  
-2 libirs-export91 libisc-export95 libiscfg-export90 libk5crypto3 libkeyutils1 libkrb5-3 libkr  
b5support0 libreadline6 libssl1.0.0 libstdc++6 libusb-0.1-4 libwrap0 openssh-client openssh-sft  
p-server perl perl-modules readline-common  
I: Checking component main on http://http.debian.net/debian...  
I: Retrieving acl 2.2.52-2  
I: Validating acl 2.2.52-2  
I: Retrieving libacl1 2.2.52-2  
I: Validating libacl1 2.2.52-2  
I: Retrieving adduser 3.113+nmu3  
I: Validating adduser 3.113+nmu3  
I: Retrieving apt 1.0.9.8.3  
I: Validating apt 1.0.9.8.3  
I: Retrieving libapt-pkg4.12 1.0.9.8.3  
I: Validating libapt-pkg4.12 1.0.9.8.3  
I: Retrieving libattr1 1:2.4.47-2  
I: Validating libattr1 1:2.4.47-2  
I: Retrieving libaudit-common 1:2.4-1  
I: Validating libaudit-common 1:2.4-1  
I: Retrieving libaudit1 1:2.4-1+b1  
I: Validating libaudit1 1:2.4-1+b1  
I: Retrieving base-files 8+deb8u4  
I: Validating base-files 8+deb8u4  
I: Retrieving base-passwd 3.5.37  
I: Validating base-passwd 3.5.37  
I: Retrieving bash 4.3-11+b1  
I: Validating bash 4.3-11+b1
```

Figure 4: Creación de Contenedor LXC

```
alberto@SEMHUNT-DEBIAN: ~  
File Edit View Search Terminal Help  
I: Configuring openssh-client...  
I: Configuring openssh-sftp-server...  
I: Configuring openssh-server...  
I: Configuring libc-bin...  
I: Configuring systemd...  
I: Base system installed successfully.  
Download complete.  
Copying rootfs to /var/lib/lxc/myfirstcontainer/rootfs...Generating locales (this might take a  
while)...  
en_US.UTF-8... done  
Generation complete.  
insserv: warning: current start runlevel(s) (empty) of script `checkroot.sh' overrides LSB defa  
ults (S).  
insserv: warning: current stop runlevel(s) (S) of script `checkroot.sh' overrides LSB defaults  
(empty).  
insserv: warning: current start runlevel(s) (empty) of script `checkroot.sh' overrides LSB defa  
ults (S).  
update-rc.d: error: umountfs Default-Start contains no runlevels, aborting.  
insserv: warning: current start runlevel(s) (empty) of script `hwclock.sh' overrides LSB defaul  
ts (S).  
insserv: warning: current stop runlevel(s) (0 6 S) of script `hwclock.sh' overrides LSB default  
s (0 6).  
update-rc.d: error: cannot find a LSB script for hwclockfirst.sh  
Creating SSH2 RSA key; this may take some time ...  
2048 f7:20:80:6e:6f:e0:5d:4b:89:11:57:27:f5:8c:3d:41 /etc/ssh/ssh_host_rsa_key.pub (RSA)  
Creating SSH2 DSA key; this may take some time ...  
1024 60:81:f8:41:2e:f5:d0:09:7e:7d:6a:7b:09:91:3e:39 /etc/ssh/ssh_host_dsa_key.pub (DSA)  
Creating SSH2 ECDSA key; this may take some time ...  
256 7d:34:61:6c:86:ab:ab:4c:23:29:c0:1f:fc:30:b6:0d /etc/ssh/ssh_host_ecdsa_key.pub (ECDSA)  
Creating SSH2 ED25519 key; this may take some time ...  
256 6f:f1:7c:1b:67:d8:cf:c3:f3:3f:84:dc:7a:81:31:ff /etc/ssh/ssh_host_ed25519_key.pub (ED25519)  
Failed to read /proc/cmdline. Ignoring: No such file or directory  
invoke-rc.d: policy-rc.d denied execution of start.  
  
Current default time zone: 'America/Mexico_City'  
Local time is now:      Wed May 18 18:50:14 CDT 2016.  
Universal Time is now:  Wed May 18 23:50:14 UTC 2016.  
  
Root password is 'PELYSTgC', please change !  
root@SEMHUNT-DEBIAN: #
```

Figure 5: Creación de Contenedor LXC

Una vez que se crea el contenedor (esto toma algún tiempo porque tiene que descargar todos los paquetes base e instalar un sistema base), chroot en ella y crear algunos dispositivos TTY . He descubierto que el comando lxc-console no funcionaba si yo no he creado estos dispositivos en primer lugar:

```
sudo chroot /var/lib/lxc/myfirstcontainer/rootfs.
```

```
mknod -m 666 /dev/tty1 c 4 1.
```

```
mknod -m 666 /dev/tty2 c 4 2.
```

```
mknod -m 666 /dev/tty3 c 4 3.
```

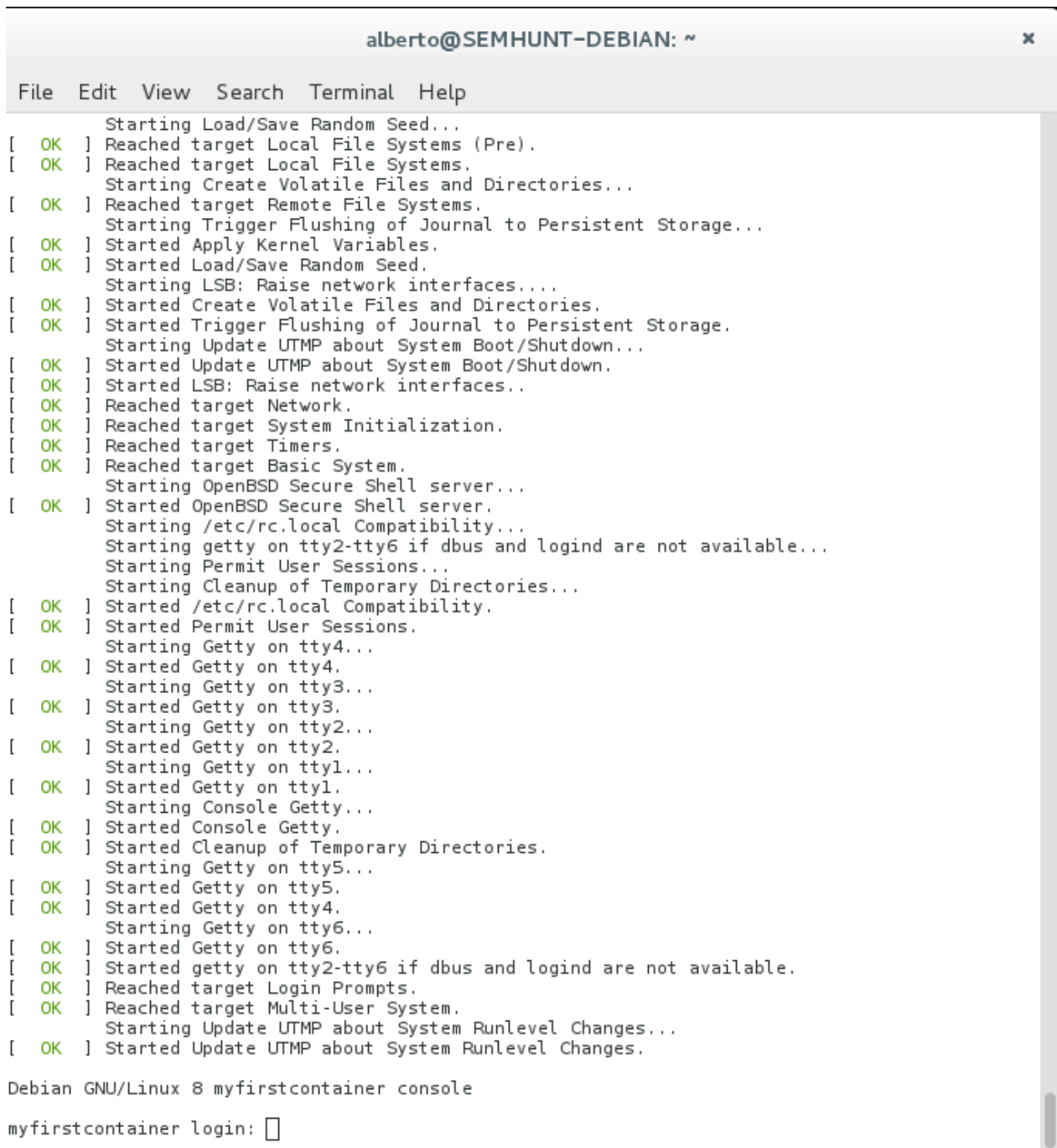
```
mknod -m 666 /dev/tty4 c 4 4.
```

```
mknod -m 666 /dev/tty5 c 4 5.
```

```
mknod -m 666 /dev/tty6 c 4 6.
```

Ahora iniciamos el contenedor:

```
sudo lxc-start -n myfirstcontainer.
```



The screenshot shows a terminal window titled 'alberto@SEMHUNT-DEBIAN: ~'. The terminal output displays the boot sequence of a container. It starts with 'Starting Load/Save Random Seed...' and proceeds through various system initialization steps, including reaching targets for Local File Systems, Remote File Systems, and System Initialization. The process continues with starting the OpenBSD Secure Shell server, updating UTMP, and starting getty on multiple ttys. The boot sequence concludes with 'Reached target Login Prompts.' and 'Reached target Multi-User System.' The prompt 'myfirstcontainer login: ' is visible at the bottom.

```
alberto@SEMHUNT-DEBIAN: ~
File Edit View Search Terminal Help
Starting Load/Save Random Seed...
[ OK ] Reached target Local File Systems (Pre).
[ OK ] Reached target Local File Systems.
Starting Create Volatile Files and Directories...
[ OK ] Reached target Remote File Systems.
Starting Trigger Flushing of Journal to Persistent Storage...
[ OK ] Started Apply Kernel Variables.
[ OK ] Started Load/Save Random Seed.
Starting LSB: Raise network interfaces...
[ OK ] Started Create Volatile Files and Directories.
[ OK ] Started Trigger Flushing of Journal to Persistent Storage.
Starting Update UTMP about System Boot/Shutdown...
[ OK ] Started Update UTMP about System Boot/Shutdown.
[ OK ] Started LSB: Raise network interfaces..
[ OK ] Reached target Network.
[ OK ] Reached target System Initialization.
[ OK ] Reached target Timers.
[ OK ] Reached target Basic System.
Starting OpenBSD Secure Shell server...
[ OK ] Started OpenBSD Secure Shell server.
Starting /etc/rc.local Compatibility...
Starting getty on tty2-tty6 if dbus and logind are not available...
Starting Permit User Sessions...
Starting Cleanup of Temporary Directories...
[ OK ] Started /etc/rc.local Compatibility.
[ OK ] Started Permit User Sessions.
Starting Getty on tty4...
[ OK ] Started Getty on tty4.
Starting Getty on tty3...
[ OK ] Started Getty on tty3.
Starting Getty on tty2...
[ OK ] Started Getty on tty2.
Starting Getty on tty1...
[ OK ] Started Getty on tty1.
Starting Console Getty...
[ OK ] Started Console Getty.
[ OK ] Started Cleanup of Temporary Directories.
Starting Getty on tty5...
[ OK ] Started Getty on tty5.
[ OK ] Started Getty on tty4.
Starting Getty on tty6...
[ OK ] Started Getty on tty6.
[ OK ] Started getty on tty2-tty6 if dbus and logind are not available.
[ OK ] Reached target Login Prompts.
[ OK ] Reached target Multi-User System.
Starting Update UTMP about System Runlevel Changes...
[ OK ] Started Update UTMP about System Runlevel Changes.

Debian GNU/Linux 8 myfirstcontainer console

myfirstcontainer login: 
```

Figure 6: Arrancando el Contenedor

De igual form se puede arrancar un contenedor en segundo plano (no entra en él):

```
lxc-start -n minigentoo -d.
```

Puedo ver su estado ahora:

```
sudo lxc-info -n minigentoo (me informa de que está funcionando, pero estoy fuera).
```

Ahora bien, puedo ver los contenedores creados y su estado:

```
sudo lxc-ls -f.
```



```

alberto@SEMHUNT-DEBIAN: ~
File Edit View Search Terminal Help
root@SEMHUNT-DEBIAN: # lxc-ls -f
NAME          STATE     IPV4  IPV6  AUTOSTART
-----
myfirstcontainer  RUNNING -    -      NO
root@SEMHUNT-DEBIAN: # ls /var/lib/lxc/
myfirstcontainer
root@SEMHUNT-DEBIAN: # lxc-info -n myfirstcontainer
Name:          myfirstcontainer
State:         RUNNING
PID:           17652
CPU use:       4.41 seconds
BlkIO use:     0 bytes
root@SEMHUNT-DEBIAN: #

```

Figure 7: Contenedores creados

```

alberto@SEMHUNT-DEBIAN: ~
File Edit View Search Terminal Help
top - 21:09:34 up 14 min, 1 user, load average: 0.06, 0.42, 0.53
Tasks: 12 total, 1 running, 11 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.9 us, 0.3 sy, 0.0 ni, 97.8 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem: 4024336 total, 1175756 used, 2848580 free, 61628 buffers
KiB Swap: 8143868 total, 0 used, 8143868 free. 602664 cached Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
    1 root        20   0   4936   3468  2732  S   0.0   0.1   0:00.31 systemd
   20 root        20   0  12148   3036  2796  S   0.0   0.1   0:00.25 systemd-j+
   69 root        20   0   7872   4612  4136  S   0.0   0.1   0:00.02 sshd
   75 root        20   0   2196   1528  1420  S   0.0   0.0   0:00.00 agetty
   76 root        20   0   2196   1576  1464  S   0.0   0.0   0:00.00 agetty
   77 root        20   0   2196   1548  1432  S   0.0   0.0   0:00.00 agetty
   78 root        20   0   2196   1492  1384  S   0.0   0.0   0:00.00 agetty
   79 root        20   0   5376   3108  2676  S   0.0   0.1   0:00.09 login
  121 root        20   0   5200   3192  2864  S   0.0   0.1   0:00.01 bash
  140 root        20   0   4828   2424  2144  R   0.0   0.1   0:00.00 top
  141 root        20   0   2196   1568  1444  S   0.0   0.0   0:00.00 agetty
  142 root        20   0   2196   1592  1476  S   0.0   0.0   0:00.00 agetty

```

Figure 8: Procesos del Contenedor

4 Configuración de Red

Estaré usando libvirt para administrar el puente de red que vamos a crear para nuestros contenedores . Comienzo por habilitar la configuración de red por defecto que viene con libvirt:

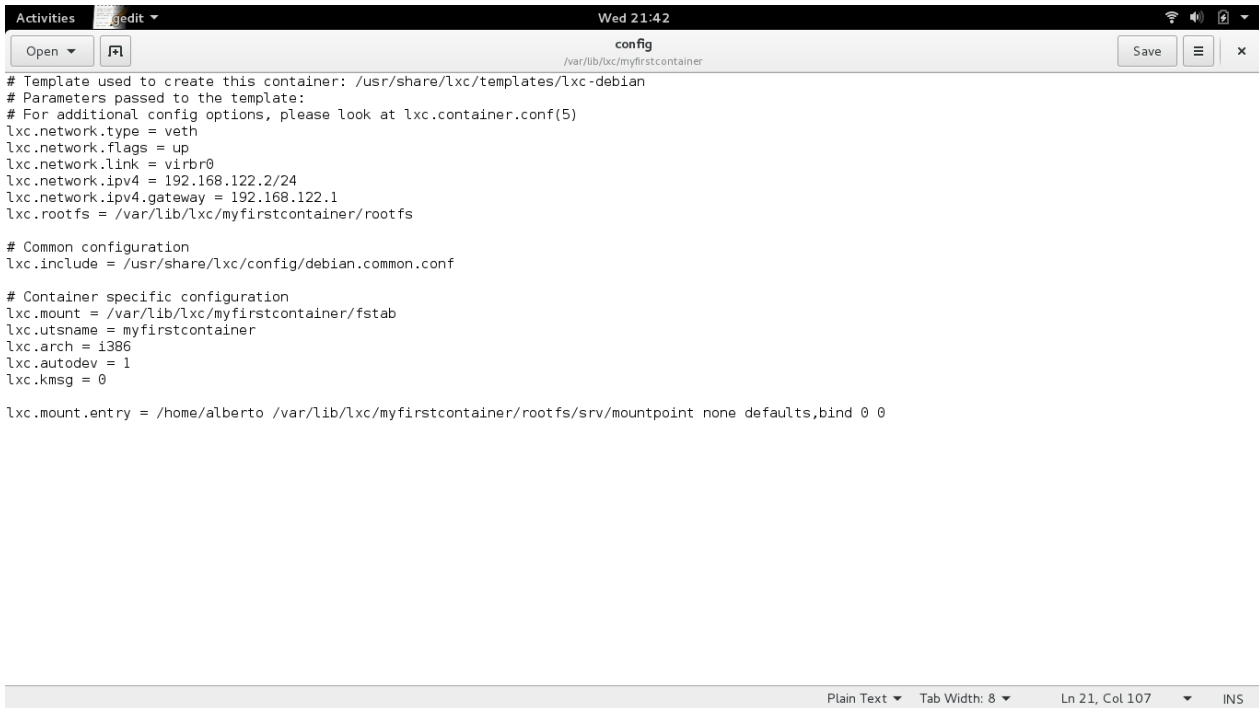
```
sudo virsh -c lxc:/// net-define /etc/libvirt/qemu/networks/default.xml.
```

```
sudo virsh -c lxc:/// net-start default.
```

```
sudo virsh -c lxc:/// net-autostart default.
```

La red por defecto usando libvirt es 192.168.122.0/24 . Si desea cambiarlo , ejecute virsh -c lxc : /// net- edición por defecto y adaptar los ajustes.

Ahora, estando el puente definido, necesito asignar la interface de red a mi contenedor, hice esto agregando las siguientes lineas a `/var/lib/lxc/myfirstcontainer/config`



```
# Template used to create this container: /usr/share/lxc/templates/lxc-debian
# Parameters passed to the template:
# For additional config options, please look at lxc.container.conf(5)
lxc.network.type = veth
lxc.network.flags = up
lxc.network.link = virbr0
lxc.network.ipv4 = 192.168.122.2/24
lxc.network.ipv4.gateway = 192.168.122.1
lxc.rootfs = /var/lib/lxc/myfirstcontainer/rootfs

# Common configuration
lxc.include = /usr/share/lxc/config/debian.common.conf

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

lxc.mount.entry = /home/alberto /var/lib/lxc/myfirstcontainer/rootfs/srv/mountpoint none defaults,bind 0 0
```

Figure 9: Configuración de Red

5 Iniciación Automática del Contenedor.

Para iniciar automáticamente el contenedor desde que se inicia el boot, lo que tenemos que hacer es colocar una ligaa los archivos de configuración de nuestro contenedor a la siguiente ruta `/etc/lxc/auto/`

```
sudo ln -s /var/lib/lxc/myfirstcontainer/config /etc/lxc/auto/myfirstcontainer.
```

6 Conclusión.

Jamás había escuchado el término de Linux Containers, resolví muchos problemas creando máquinas virtuales lo que me tomaba mas tiempo y costo, el tener contenedores bien administrados me da ideas infinitas de poder hacer muchas cosas usando LXC, como por ejemplo tener una Base de Datos en un contenedor y un aplicativo web en otro contenedor.

Es extremadamente conveniente poder tener varios sistemas heterogéneos funcionando en un servidor, pero no tener que preocuparse de configuraciones o versiones de lenguajes potencialmente conflictivas. Incluso el servir diferentes apps en la web a través de un proxy y simplifica las cosas.

Tomemos en cuenta que un contenedor tiene su propio sistema de ficheros y que es simplemente un directorio en nuestras máquinaa, así que podemos hacer un rsync de eso a otra máquina, si queremos copiar nuestro contenedor a otra máquina.

LXC está muy bien. El aislamiento es bueno.

Ahora bien, resolviendo los puntos de esta tarea puedo comentar lo siguiente:

¿Pueden detectar como se ven diferentes procesos init desde el anfitrión?

R: No, lo que si pude detectar es que el contenedor maneja el mismo PID de systemd, pero por la característica de LXC, este es un entorno con su propio espacio de procesos. Ahora, revisando los procesos de la maquina anfitrión, procesos como ssh, apache, etc, no comparten el mismo PID.

¿Pueden detectar cuantos y cuales procesos se ven desde el invitado?

R: No, de hecho podría atreverme a decir que es un ambiente aislado y esto hace que, el sistema operativo virtualizado, funcione a velocidad nativa, cosa que no ocurriría si hubiese que emular una máquina completa y tener varias instancias de kernels. En este sentido, podríamos considerarlos a los clásicos chroot, permitiendo más aislamiento y más flexibilidad. ¿Qué otras llamadas al sistema importantes creen que pasan para inicializar el contenedor?

R: Creo que las llamadas al sistema que realiza nuestro contenedor se realizan a los procesos de nuestro contenedor y a su vez directamente al kernel de la máquina anfitrión.