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.
```

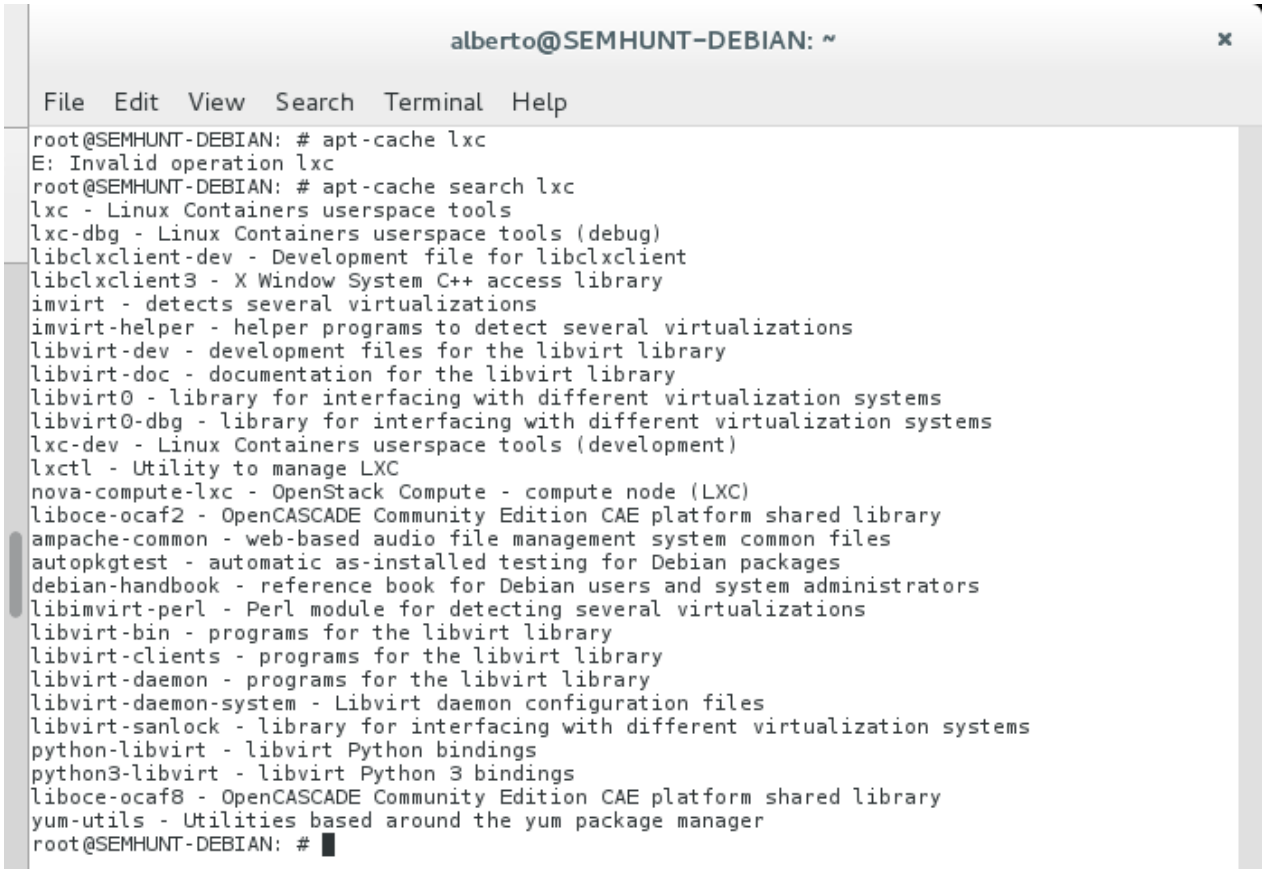

Figure 1: Búsqueda de Paquetes LXC

Posteriormente se procede a instalar lxc:

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

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.
```

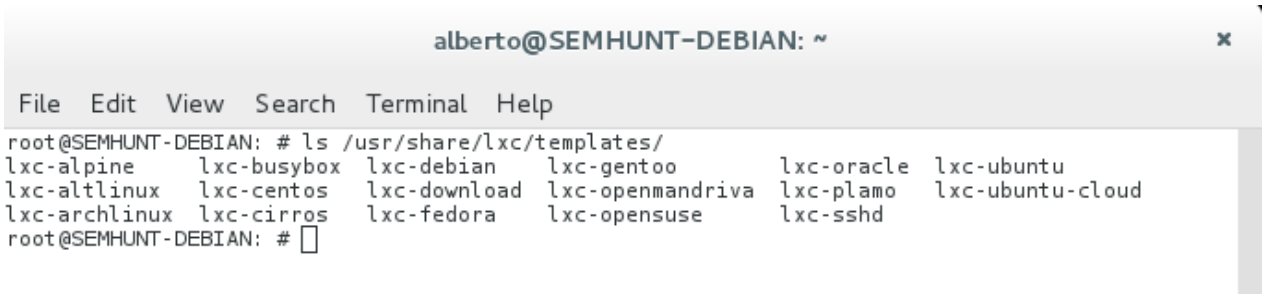

Figure 3: Templates LXC

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.
```

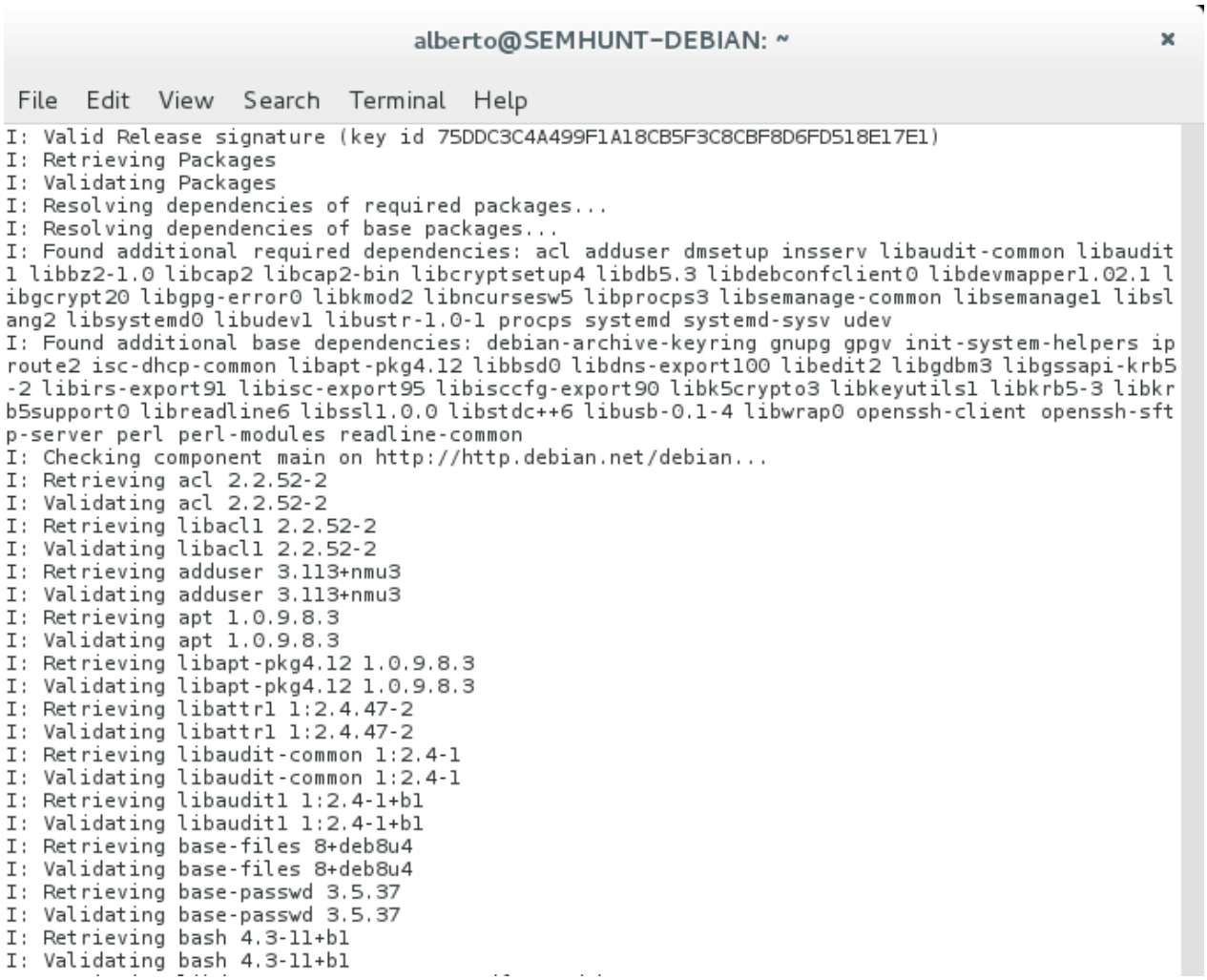

Figure 4: Creación de Contenedor LXC

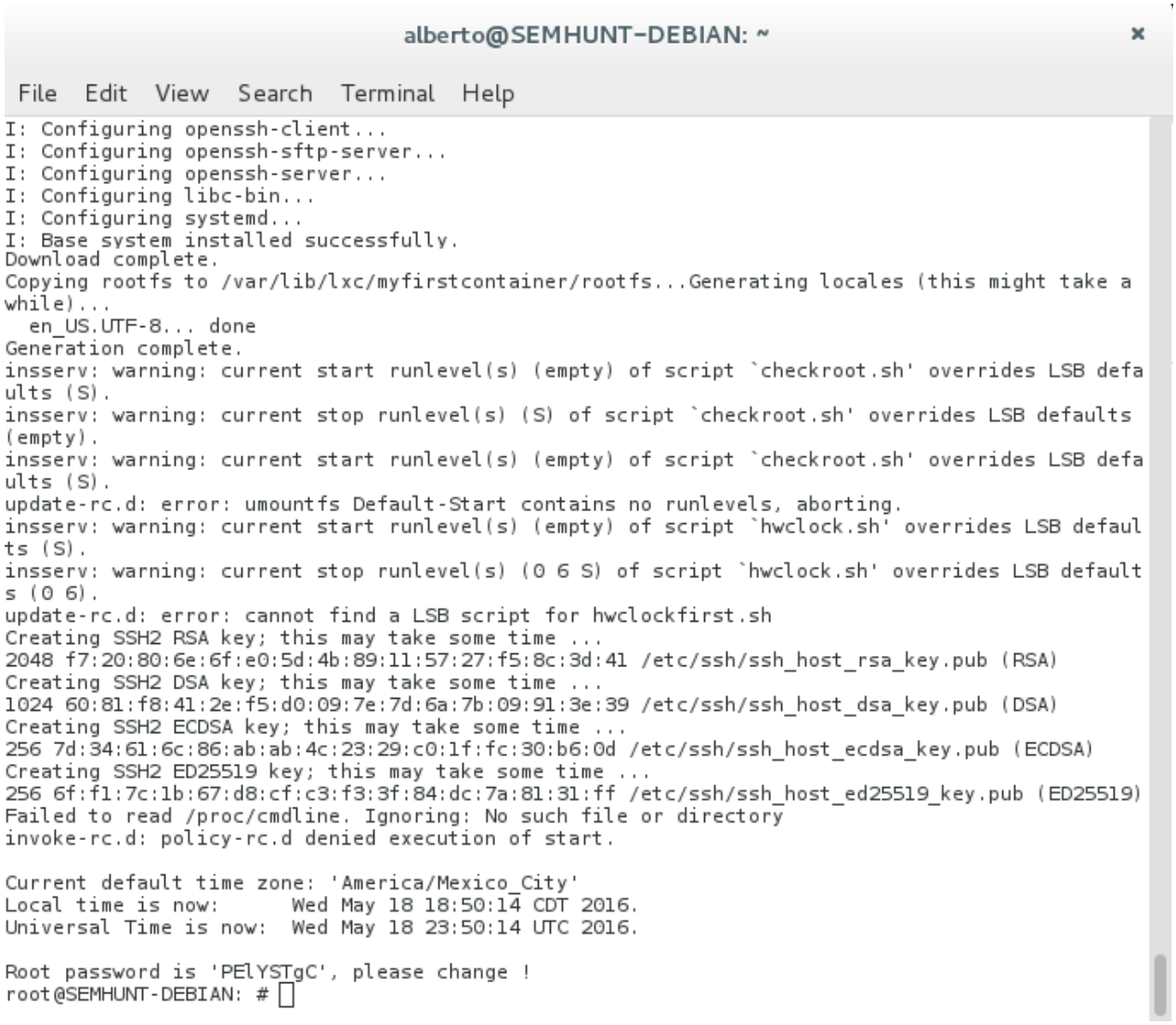

Figure 5: Creación de Contenedor LXC

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.
```

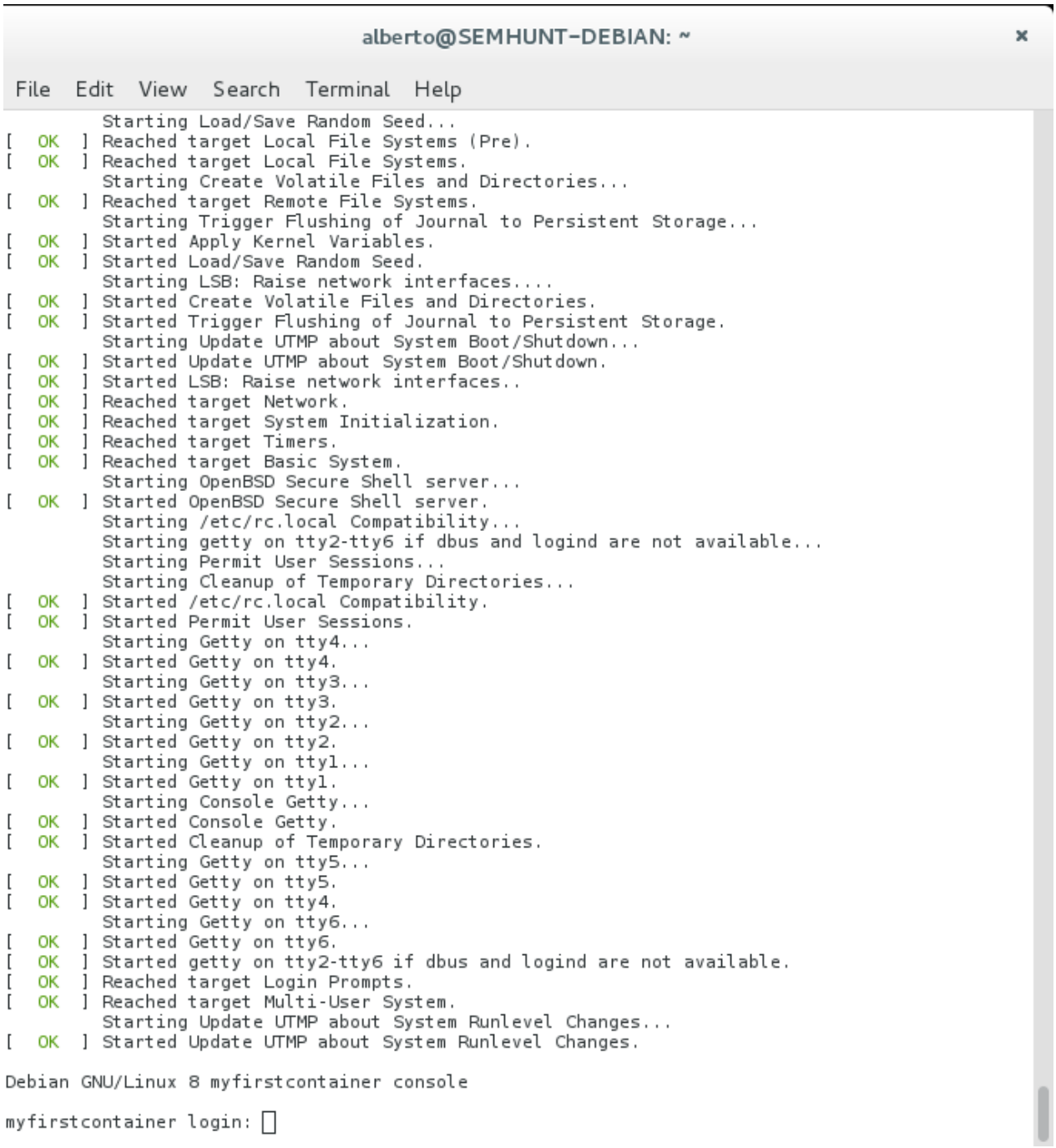

Figure 6: Arrancando el Contenedor

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.
```

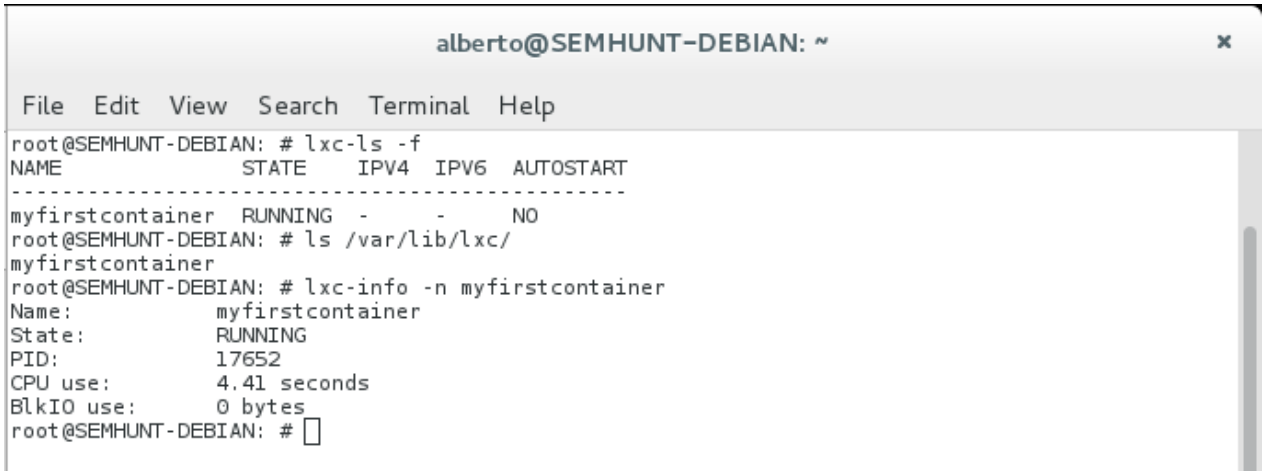

Figure 7: Contenedores creados

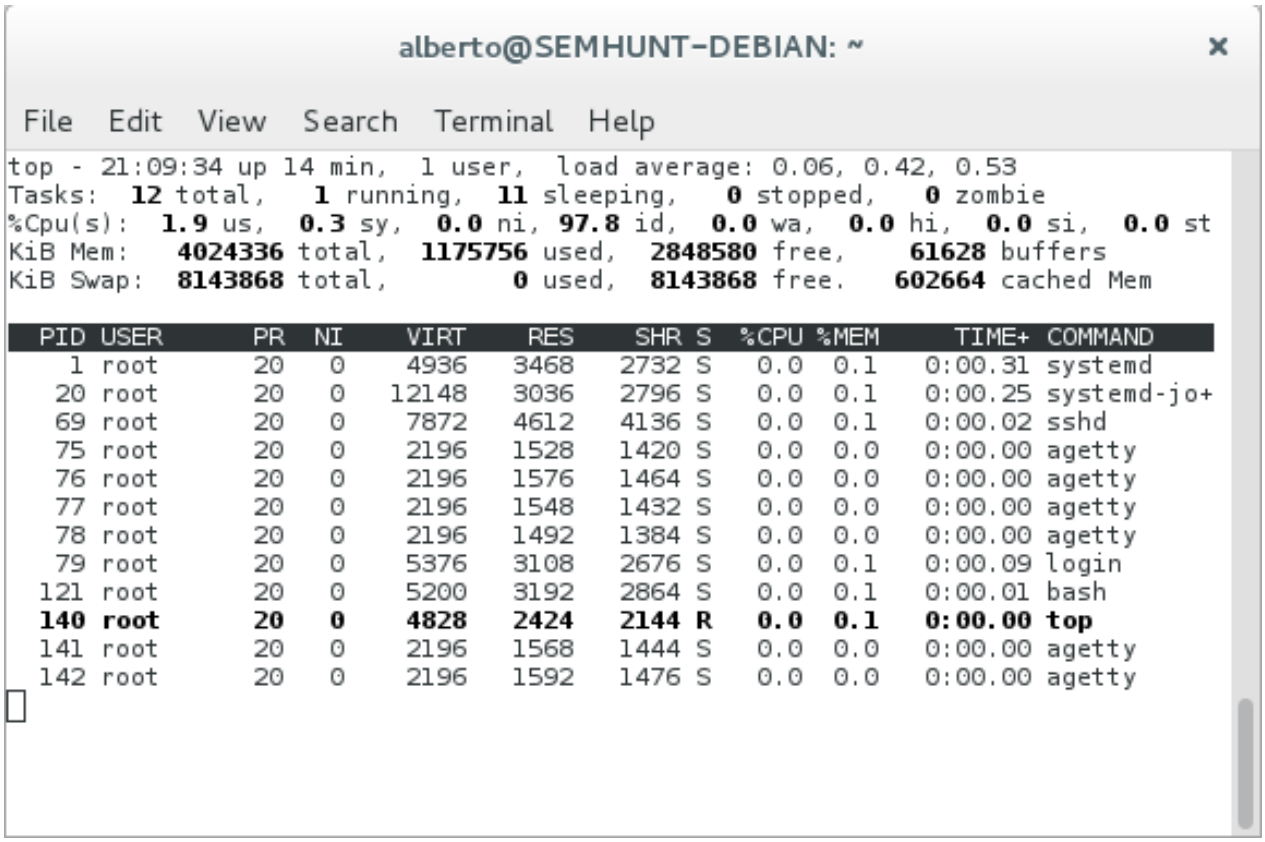

Figure 8: Procesos del Contenedor

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 ka interface de red a mi contenedor, hice esto agregando las siguientes lineas a /var/lib/lxc/myfirstcontainer/config

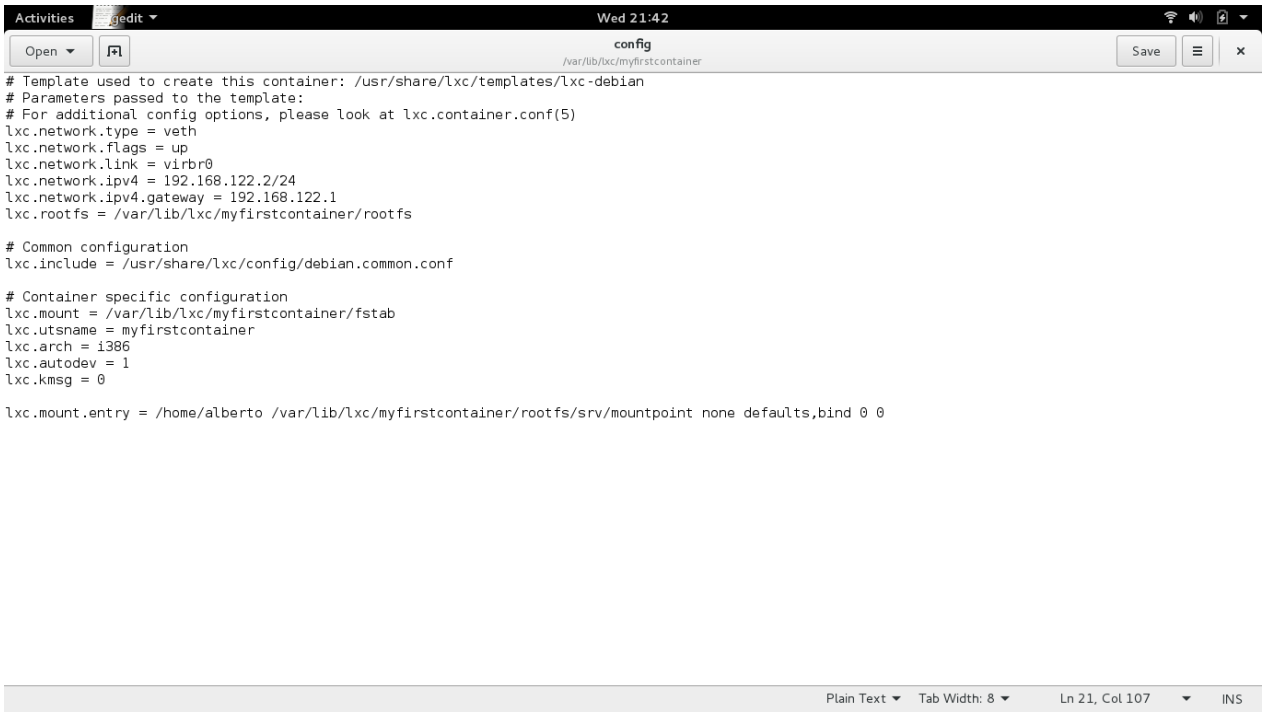

Figure 9: Configuracion de Red

5 Iniciación Automática del Contenedor.

Para iniciar automaticamente el contenedor desde que se inicia el boot, lo que teneos que hacer es colocar una ligaa los archivos de configuracion 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 extremadamamente 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 puedemos hacer un rsync de eso a otra máquina, si quieremos 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 caracteristica 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.