

CONTENEDORES LXC EN UBUNTU

CONTENIDO

Introducción	1
Preparar el host	2
Otras configuraciones de <code>lxd</code>	2
Comandos LXD	3
Ayuda de LXD	3
Imágenes	3
Instancias	3
Información de instancias	4
Manipular instancias	4
Ejecutar comandos en una instancia	4
Pasar archivos entre instancia y host	5
Configurar un contenedor como privilegiado	5
Snapshots	5
Configuraciones de instancia	5
Configuraciones útiles	6
Redes	6
Configuraciones útiles de red	7
Manipular conexiones de red de una instancia	7
Perfiles LXD	8
Pools	8
Añadir un dispositivo de almacenamiento	9
Añadir una pool	9
Otros comandos con pools	9
LXD con GUI	10

INTRODUCCIÓN

Los contenedores LXC (*Linux Container*) son una alternativa ligera a las máquinas virtuales (VM) tradicionales, aunque su funcionamiento y usos es parecido. Los contenedores suponen un nivel menor de virtualización, ya que utilizan el núcleo (sistema operativo) del sistema anfitrión, por lo que las imágenes e instancias de contenedores ocupan menos espacio, se lanzan en cuestión de segundos, utilizan menos memoria y pueden ser más rápidas en ejecutarse.

Trabajar directamente con LXC puede ser tedioso y poco intuitivo, por lo que trabajaremos con LXD (*Linux container Daemon*), un gestor de contenedores por línea de comandos que nos resultará más cómodo de usar.

[Aquí](#) hay una guía de los primeros pasos con LXD en Ubuntu.

PREPARAR EL HOST

Los contenedores LXC solo funcionan con Linux, ya que se diseñaron para compartir el núcleo Linux, por lo que la máquina host deberá tener una distribución de Linux instalada, y los contenedores serán todas máquinas Linux.

En nuestro caso, crearemos una máquina Ubuntu Server 24.04, con suficientes recursos para alojar potencialmente un buen número de contenedores (≥ 50 GB de almacenamiento, ≥ 4 GB de memoria, ≥ 2 núcleos de CPU).

Una vez instalado el sistema operativo (o clonada), es recomendable hacer SSH a la VM para realizar los siguientes pasos.

Una vez iniciado la VM, instalamos LXD:

```
# snap install lxd
```

Una vez instalado, para poder llevar a cabo los siguientes pasos y administrar contenedores, es necesario pertenecer al grupo `lxd` o bien ser `root`. Si se quiere usar desde un usuario normal, será necesario añadirle al grupo `lxd` (si no lo está):

```
# usermod <usuario> -aG lxd
```

Primero de todo, es necesario definir una configuración inicial de LXD. Crearemos una configuración inicial mínima mediante:

```
$ lxd init --minimal
```

Ya podemos crear nuestros contenedores.

OTRAS CONFIGURACIONES DE LXD

Se puede definir la configuración inicial de forma interactiva y guiada:

```
$ lxd init
```

Para exportar la configuración actual en YAML (a la salida estándar):

```
$ lxd init --dump
```

Para importar una configuración (desde la entrada estándar, en YAML):

```
$ lxd init --preseed
```

COMANDOS LXD

Aunque pueda resultar confuso, LXD proporciona el comando `lxc` para administrar los contenedores. No se debe confundir con `lxd`, que solo hemos usado para establecer la configuración inicial.

AYUDA DE LXD

Cuando no sabemos cómo se escribe una línea, podemos acudir a la ayuda de `lxc` para saber qué comandos y argumentos acepta:

```
$ lxc --help
```

Aunque también podemos usar la ayuda dentro de otros comandos:

```
$ lxc config --help
$ lxc config device --help
$ lxc config device add --help
```

IMÁGENES

Los contenedores se crean a partir de una imagen, que suele tener el nombre de un repositorio y la distribución o versión (`<repo>:<distro>`). Se pueden buscar las imágenes disponibles de un repositorio determinado con:

```
$ lxc image list images:
```

En nuestras prácticas y actividades usaremos la imagen `ubuntu:24.04`, pero se pueden usar otras imágenes por motivos de espacio:

- Los contenedores Ubuntu (`ubuntu:24.04`) ocupan 600-1000MB recién instalados.
 - [Networking en Ubuntu](#)
- Los contenedores Debian (`images:debian/12`) ocupan 200-600MB recién instalados.
 - [Networking en Debian](#)
- Los contenedores Alpine (`images:alpine/3.20`) ocupan 6-10MB recién instalados.
 - [Networking en Alpine](#)

Al buscar ayuda para instalar y configurar un servicio, es necesario incluir la distribución Linux en la búsqueda, ya que en general no se siguen los mismos pasos en una distribución que en otra.

INSTANCIAS

Una **instancia** puede referirse a un contenedor o a una máquina virtual, aunque generalmente nos referiremos a **contenedor** en este documento.

Para crear una instancia, ejecuta uno de los siguientes comandos:

```
$ lxc init <image> <instancia>
$ lxc launch <image> <instancia>
```

El comando `init` solo crea la instancia, mientras que `launch` la crea y la arranca. `<image>` es la imagen de la instancia e `<instancia>` es el nombre que daremos al nuevo contenedor.

También se pueden crear máquinas virtuales, tan solo añadiendo la opción `--vm` al comando.

INFORMACIÓN DE INSTANCIAS

Para ver una lista de las instancias creadas:

```
$ lxc list
```

Para ver información detallada de una instancia:

```
$ lxc info <instancia>
```

MANIPULAR INSTANCIAS

Para iniciar una instancia:

```
$ lxc start <instancia>
```

Para detener (apagar) una instancia (con la opción `-f`, se puede forzar el apagado):

```
$ lxc stop [-f] <instancia>
```

Para iniciar/detener todas las instancias a la vez:

```
$ lxc start --all  
$ lxc stop --all
```

Para reiniciar una instancia:

```
$ lxc restart <instancia>
```

Para eliminar una instancia (debería estar apagada):

```
$ lxc delete <instancia>
```

Para renombrar una instancia (debe estar apagada):

```
$ lxc rename <instancia> <nuevo_nombre>
```

Para copiar una instancia:

```
$ lxc copy <instancia> <copia_instancia>
```

Para importar una instancia a partir de un fichero `.tar.gz`:

```
$ lxc import <fichero> <instancia>
```

EJECUTAR COMANDOS EN UNA INSTANCIA

Para ejecutar un comando en una instancia:

```
$ lxc exec <instancia> -- <comando> <argumentos...>
```

Los dos guiones (`--`) son opcionales, pero si se va a haber opciones en los argumentos de `<comando>`, estos previenen que se interpreten como opciones de `lxc`.

Para abrir un shell interactivo de una instancia:

```
$ lxc shell <instancia>
```

O, de forma extendida:

```
$ lxc exec <instancia> bash
```

`bash` es el intérprete interactivo shell por defecto de Ubuntu y Debian. Para instancias de Alpine, el intérprete deberá ser `sh` (ya que no incluyen intérprete `bash`).

Para cerrar el shell de la instancia, puedes salir con `exit`.

PASAR ARCHIVOS ENTRE INSTANCIA Y HOST

Para traer un archivo desde una instancia hasta el host:

```
$ lxc file pull <instancia><ruta origen> <ruta destino>
```

Por ejemplo:

```
$ lxc file pull first/etc/hosts .
$ lxc file pull first/var/log/syslog - | less
```

Para llevar un archivo del host a una instancia:

```
$ lxc file push <ruta origen> <instancia><ruta destino>
```

CONFIGURAR UN CONTENEDOR COMO PRIVILEGIADO

Los contenedores se crean por defecto sin permisos de administrador (*unprivileged*), para establecer que un contenedor tenga privilegios:

```
$ lxc config set <instancia> security.privileged true
```

SNAPSHOTS

Para crear una *snapshot* del estado actual de una instancia:

```
$ lxc snapshot <instancia> <snapshot>
```

Donde `<snapshot>` es el nombre de la nueva *snapshot*.

Para restaurar una instancia a una *snapshot*:

```
$ lxc restore <instancia> <snapshot>
```

Se pueden ver las *snapshots* creadas en una instancia mediante `lxc info`.

Para borrar una *snapshot*:

```
$ lxc delete <instancia>/<snapshot>
```

CONFIGURACIONES DE INSTANCIA

Para modificar la configuración de una instancia abriendo un editor de texto:

```
$ lxc config edit <instancia>
```

Para modificar la configuración por defecto de todas las instancias de un perfil:

```
$ lxc profile edit <perfil>
```

Añadiendo o cambiando líneas dentro de la sección `config`.

Para modificar rápidamente una configuración de una instancia/perfil:

```
$ lxc config set <instancia> <config> <valor>
$ lxc profile set <perfil> <config> <valor>
```

Para ver la configuración de una instancia (con la opción `-e` incluye las configuraciones heredadas de perfiles):

```
$ lxc config show [-e] <instancia>
```

CONFIGURACIONES ÚTILES

Algunas configuraciones útiles de un contenedor son:

CONFIGURAR QUE UN CONTENEDOR ARRANQUE/NO ARRANQUE AL INICIO

```
boot.autostart: true|false
```

Por defecto, todos los contenedores arrancan al inicio. Esto tiene sentido para servidores DHCP, enrutadores, etc., pero puede no ser siempre la opción deseada. El valor `false` evita este comportamiento.

PRIVILEGIOS

```
security.privileged: true|false
```

Los contenedores se crean por defecto sin permisos de administrador (*unprivileged*). El valor `true` establece privilegios de administrador para el contenedor (puede ser inseguro).

REDES

Para ver una lista de las redes disponibles para usar en las instancias:

```
$ lxc network list
```

La configuración inicial mínima de LXD crea una red virtual de tipo puente llamada `lxdbr0`. Esta red se comporta como una red privada NAT autogestionada, donde los contenedores reciben automáticamente una IP por DHCP y pueden comunicarse con el host pero no son accesibles desde la red a la que pertenece el host.

El perfil `default` viene configurado con una única interfaz de red `eth0` conectada a `lxdbr0`, por lo que cualquier nueva instancia bajo el perfil `default` estará conectada a esta red interna.

Para crear una red llamada `<red>` para usar en las instancias:

```
$ lxc network create <red>
```

Por defecto, se crea una red de tipo puente (`bridge`), que permitirá simular una red interna.

Para modificar la configuración de una red en un editor:

```
$ lxc network edit <red>
```

Para modificar rápidamente una configuración de una red:

```
$ lxc network set <red> <config> <valor>
```

CONFIGURACIONES ÚTILES DE RED

Algunas configuraciones útiles de una red son:

DIRECCIÓN DEL GATEWAY DE LA RED

```
ipv4.address: <CIDR>|auto|none
ipv6.address: <CIDR>|auto|none
```

Se puede configurar la dirección IP y máscara de la red (<CIDR>), que se genere una red aleatoria (auto) o desactivar IPv4/IPv6 (none).

Si <CIDR> es una dirección IP de host, entonces será el *gateway* virtual de la red, salvo que se configure otra IP en `ipv4.dhcp.gateway/ipv6.dhcp.gateway`.

DHCP

```
ipv4.dhcp: true|false
ipv6.dhcp: true|false
```

Para redes administradas, esto permite la asignación automática de direcciones IP (DHCP) a las instancias de la red. Por defecto es `true`, salvo que `address` esté desactivado.

NAT

```
ipv4.nat: true|false
ipv6.nat: true|false
```

Para redes administradas, se puede configurar que se lleven a cabo traducciones de direcciones de red (NAT) a los paquetes salientes de la red. Por defecto es `true`, salvo que `address` esté desactivado.

MANIPULAR CONEXIONES DE RED DE UNA INSTANCIA

Para añadir una conexión de red a una instancia:

```
$ lxc config device add <instancia> <interfaz> nic network=<red>
```

Donde `nic` es el tipo de dispositivo que se está creando (dispositivo de red), <interfaz> es el nombre que tendrá la nueva interfaz dentro de la instancia, y <red> es el nombre de la red (creada previamente con `lxc network create`) a la que estará conectada la nueva interfaz de la instancia.

Una vez creada la conexión deseada, al iniciar la instancia se deberá configurar la nueva interfaz, ya sea estática o por DHCP.

Los perfiles de las instancias suelen definir una o más interfaces de red, las cuales convivirán con las definidas en la instancia actual, salvo que se cree una con el mismo nombre que una definida en el perfil (en este caso la nueva la sobrescribiría).

Si se quiere conectar directamente la instancia a una interfaz puente existente en el host (<interfaz_host>), en lugar de a una red gestionada por LXD, la línea será:


```
$ lxc config device add <instancia> <interfaz> nic nictype=bridged
parent=<interfaz_host>
```

Para modificar una interfaz de red ya existente de una instancia:

```
$ lxc config device set <instancia> <interfaz> <clave>=<valor>...
```

Para eliminar una interfaz de red de una instancia:

```
$ lxc config device remove <instancia> <interfaz>
```

Aunque las interfaces que vengan definidas en el perfil de la instancia seguirán existiendo y no se podrán borrar con este comando, sino modificando el perfil.

Estos comandos `lxc device` también se pueden usar para crear y eliminar dispositivos de almacenamiento y cualquier tipo de dispositivo por bloques de una instancia o un perfil.

PERFILES LXD

Los nuevos contenedores se crean a partir de un perfil, que va a definir sus configuraciones predeterminadas. Es deseable personalizar el perfil por defecto (`default`) o crear nuevos perfiles.

Para modificar un perfil:

```
$ lxc profile edit <perfil>
```

Para crear un perfil nuevo:

```
$ lxc profile create <perfil>
```

Para copiar un perfil en otro:

```
$ lxc profile copy <perfil> <perfil-nuevo>
```

Para crear un nuevo contenedor a partir de un cierto perfil (si no se especifica, se usa el perfil `default`):

```
$ lxc launch <imagen> <instancia> -p <perfil>
```

Para añadir un perfil a un contenedor existente:

```
$ lxc profile add <instancia> <perfil>
```

Por ejemplo, podemos modificar el perfil `default` (o crear uno para este fin) para reemplazar su interfaz de red por defecto (`lxdbr0`) por una de las redes que hemos creado:

```
devices:
  eth0:
    name: eth0
    network: lan1
    type: nic
```

POOLS

En caso de estar usando una máquina virtual (o física) que no cuenta con suficiente espacio para alojar los contenedores, se puede usar almacenamiento externo para ello.

AÑADIR UN DISPOSITIVO DE ALMACENAMIENTO

Primero, añadiremos un nuevo dispositivo de almacenamiento virtual SCSI (o conectamos un dispositivo de almacenamiento físico) a la máquina. Una vez hecho, podremos buscarlo con:

```
# fdisk -l
```

La unidad que buscamos se llama `/dev/sdX`, donde *X* es una letra minúscula, y tendrá el mismo tamaño que el dispositivo conectado, y en caso de ser nuevo, no tendrá ninguna partición. Es posible que no se muestre en la lista hasta hacer un reinicio.

AÑADIR UNA POOL

Sabiendo el nombre de la unidad que hemos añadido, podremos crear una *pool* en esa unidad para almacenar los contenedores. Las *pools* pueden ser de distintos tipos: volúmenes lógicos (LVM), ZFS, Btrfs, etc.

En este caso, asumiendo que no usaremos la unidad para ninguna otra cosa y que no contiene datos ni particiones, crearemos una *pool* especificando la unidad:

```
$ lxc storage create <pool> <tipo> source=<unidad>
```

Donde *<pool>* es el nombre de la *pool* (por ejemplo, *pool1*), *<tipo>* es el tipo de volumen que se quiere crear (*lvm|zfs|btrfs*) y *<unidad>* es el nuevo dispositivo conectado al sistema (por ejemplo, */dev/sdb*).

OTROS COMANDOS CON POOLS

Para crear un contenedor en una *pool* determinada:

```
$ lxc launch ubuntu:22.04 -s <pool> <instancia>
```

Para modificar la *pool* en la que se crearán las nuevas instancias por defecto:

```
$ lxc profile device set default root pool <pool>
```

Para mover un contenedor a otra *pool*:

```
$ lxc move <instancia> -s <pool>
```

Para ver la información de una *pool*, como el espacio ocupado o los contenedores que aloja:

```
$ lxc storage info <pool>
```

LXD CON GUI

Si se prefiere, y se trabaja en un sistema de escritorio, se pueden gestionar los contenedores con LXD mediante una interfaz gráfica (GUI) usando el proyecto LXDUI. [Este tutorial](#) explica cómo configurarlo en Ubuntu.