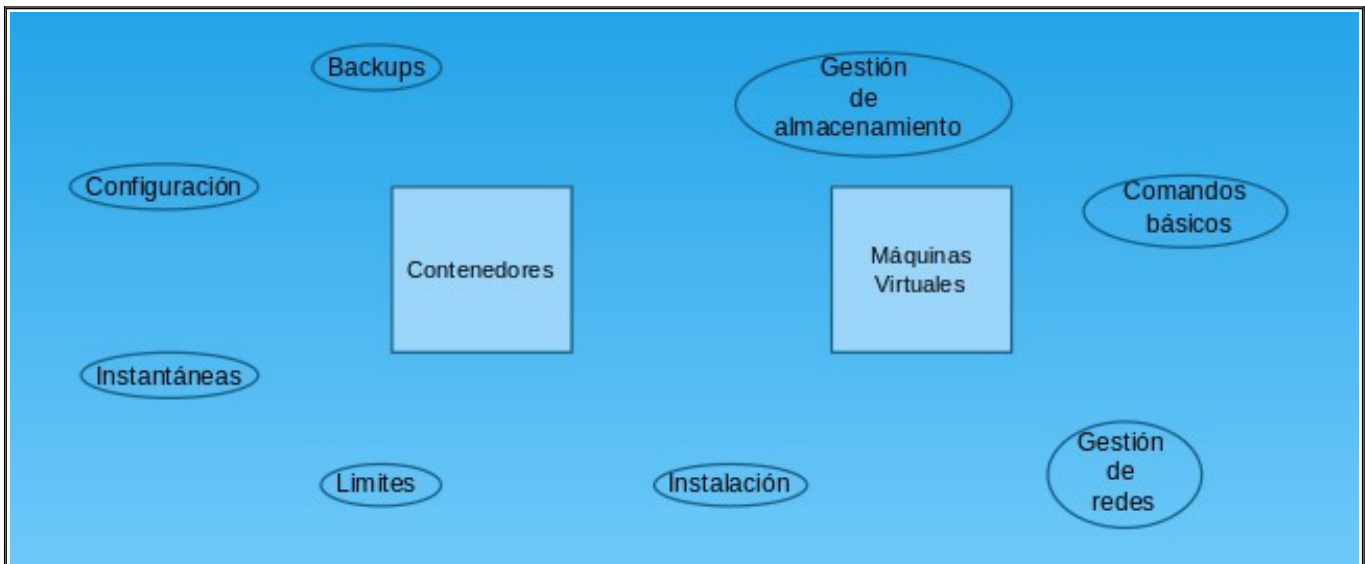


# INTRODUCCIÓN

## A

# LXD



**María Jesús Bohíguez Pérez**

**IES Gonzalo Nazareno**

**Administración de Sistemas Informáticos en Red**

## INDICE

1. Introducción.....	3
1.1 ¿Qué es LXD?.....	3
1.2 ¿Cuáles son sus principales características?.....	3
1.3 Ventajas e inconvenientes de usar LXD.....	4
1.4 Contenedores y máquinas virtuales.....	4
2. Objetivos conseguidos o que se quieren conseguir.....	4
3. Instalación y configuración inicial LXD.....	5
4. Gestión de redes.....	6
5. Gestión de almacenamiento.....	8
5.1 Copias de seguridad de los volúmenes.....	10
6. Configuración LXD.....	12
6.1 Comandos Básicos.....	12
6.2 Transferir archivos a/desde contenedores LXD.....	13
6.3 Crear backups contenedores LXD.....	14
6.4 Instantáneas de las instancias.....	16
6.5 Límites instancias.....	17
7. Máquinas Virtuales - Comandos Básicos.....	18
8. Conclusión.....	23
9. Bibliografía.....	23

# 1. Introducción

## 1.1 ¿Qué es LXD?

Linux Container Daemon (LXD), es una herramienta de gestión de los contenedores del sistema operativo Linux.

Permite crear contenedores de sistemas Linux ideales para su uso en la nube. Con esta herramienta tenemos la posibilidad de crear múltiples contenedores dentro del mismo. Así aprovechamos de una mejor manera los recursos al compartirlos en diferentes sistemas operativos o aplicaciones que corren cada una en un contenedor.

El objetivo principal de usar contenedores de sistema con LXC/LXD es el de virtualizar todo un sistema operativo Linux con aplicaciones que tengan ciclos de vida medios o largos, lo que contrasta con los contenedores Docker, donde el enfoque está puesto en contenedores de corta duración que contienen una sola aplicación. Por ejemplo: Si deseamos virtualizar un servidor de comunicaciones completo que requiere de múltiples aplicaciones ejecutándose en conjunto, pero ahorrando recursos de cómputo, un contenedor LXD es ideal. Mientras que si deseamos abstraer una aplicación web para facilitar la automatización de sus ciclos de vida cortos al requerir múltiples actualizaciones frecuentes, utilizaríamos contenedores Docker.

## 1.2 ¿Cuáles son sus principales características?

Las características de LXD son las siguientes:

- ✓ Basado en imágenes de Linux: Funciona con todas las imágenes de Linux, todas son compatibles con esta herramienta.
- ✓ Usabilidad intuitiva: LXD tiene una Api clara y sencilla y un cliente con línea de comandos sencillo también.
- ✓ Escalabilidad: Puedes gestionar contenedores individuales aunque también puedes gestionar contenedores repartidos en distintos entornos.
- ✓ Seguridad: Estos contenedores se ejecutan en espacios aislados y solo pueden acceder a determinados recursos predefinidos.
- ✓ Control de recursos sofisticado: Se puede establecer límites para la memoria principal, la CPU, el uso de la red, el almacenamiento, etc...
- ✓ Gestión de almacenamiento masivo: Admite varios backends de almacenamiento, grupos de almacenamiento y volúmenes de almacenamiento.
- ✓ Gestión de red: Permite crear puentes y túneles de red.
- ✓ Acceso al hardware del sistema: Los contenedores pueden acceder a dispositivos USB, GPU y medios de almacenamiento masivo siempre y cuando la configuración lo permita.

## 1.3 Ventajas e inconvenientes de usar LXD

Algunas de las ventajas e inconvenientes de esta herramienta son las siguientes:

- Ventajas:
  - o Puedes controlarlo todo desde una terminal.
  - o Puedes tener un Sistema Operativo completo sin necesidad de ocupar mucho espacio en tu anfitriona.
- Inconvenientes:
  - o Únicamente puedes crear contenedores con Sistema Operativo Linux.

## 1.4 Contenedores y máquinas virtuales

LXD proporciona soporte para contenedores y de sistema y para máquinas virtuales, las diferencias son las siguientes:

- Al ejecutar un contenedor del sistema, lo que hace LXD es simular una versión virtual de un sistema operativo completo.
- Al ejecutar una máquina virtual, lo que hace LXD es utilizar el hardware del sistema host, pero la máquina virtual proporciona el kernel.

El enfoque basado en contenedores ofrece un rendimiento superior al de las máquinas virtuales.

# 2. Objetivos conseguidos o que se quieren conseguir

Con este proyecto he conseguido aprender mucho sobre esta herramienta de visualización, tanto con contenedores como con máquinas virtuales.

Dentro de LXD hay mucho que aprender y muchos parámetros de cada apartado, poder verlos, aprenderlos y explicarlos todos es totalmente imposible pero he intentado informarme de lo que puede ser más usado a la hora de trabajar.

En este proyecto vais a ver:

- Gestión de redes
- Gestión de almacenamiento:
- Crear, editar y eliminar contenedores y máquinas virtuales
- Comandos para gestionar y configurar algunos parámetros
- Copias de seguridad y recuperación de esas mismas copias de seguridad en caso necesario (instantáneas)

# 3. Instalación y configuración inicial LXD

Antes de instalar algún paquete actualizamos el equipo:

```
root@debian:~# apt update
```

```
root@debian:~# apt upgrade
```

Instalamos snap ya que esta herramienta una de las opciones que tiene para instalarla es con snap.

Instalamos LXD:

```
root@debian:~# snap install lxd
2022-11-13T19:40:18+01:00 INFO Waiting for automatic snapd restart...
Warning: /snap/bin was not found in your $PATH. If you've not restarted your
session since you installed snapd, try doing that. Please see
https://forum.snapcraft.io/t/9469 for more details.
```

```
lxd 5.7-c62733b from Canonical✓ installed
```

```
root@debian:~# █
```

```
root@debian:~# snap list
```

Name	Version	Rev	Tracking	Publisher	Notes
core20	20221027	1695	latest/stable	canonical✓	base
lxd	5.7-c62733b	23889	latest/stable	canonical✓	-
snapd	2.57.4	17336	latest/stable	canonical✓	snapd

```
root@debian:~# █
```

Antes de iniciar LXD vamos a añadir nuestro usuario a el grupo lxd de la siguiente forma:

- `sudo usermod -a -G lxd [usuario]`

Iniciamos LXD:

```
mjbp@debian:~$ lxd init
Would you like to use LXD clustering? (yes/no) [default=no]:
Do you want to configure a new storage pool? (yes/no) [default=yes]:
Name of the new storage pool [default=default]:
Name of the storage backend to use (dir, lvm, btrfs, ceph, cephobject) [default=btrfs]:
Create a new BTRFS pool? (yes/no) [default=yes]:
Would you like to use an existing empty block device (e.g. a disk or partition)? (yes/no) [default=no]:
Size in GiB of the new loop device (1GiB minimum) [default=30GiB]:
Would you like to connect to a MAAS server? (yes/no) [default=no]:
Would you like to create a new local network bridge? (yes/no) [default=yes]:
What should the new bridge be called? [default=lxdbr0]:
What IPv4 address should be used? (CIDR subnet notation, "auto" or "none") [default=auto]:
What IPv6 address should be used? (CIDR subnet notation, "auto" or "none") [default=auto]:
Would you like the LXD server to be available over the network? (yes/no) [default=no]:
Would you like stale cached images to be updated automatically? (yes/no) [default=yes]:
Would you like a YAML "lxd init" preseed to be printed? (yes/no) [default=no]:
mjbp@debian:~$ █
```

Al iniciar lxd tenemos la opción de crear una red por defecto a la que irán conectada nuestros contenedores pero en caso de no quererlo así, podremos crearla con el siguiente comando:

```

root@webserver: ~
root@debian:~# lxc network create lxdnetwork
Network lxdnetwork created
root@debian:~# 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
2: eno1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 84:a9:3e:3a:98:d6 brd ff:ff:ff:ff:ff:ff
    altname enp2s0
    inet 192.168.1.102/24 brd 192.168.1.255 scope global dynamic noprefixroute eno1
        valid_lft 80986sec preferred_lft 80986sec
    inet6 fe80::86a9:3eff:fe3a:98d6/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
7: lxdnetwork: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether 00:16:3e:bb:3d:76 brd ff:ff:ff:ff:ff:ff
    inet 10.192.171.1/24 scope global lxdnetwork
        valid_lft forever preferred_lft forever
    inet6 fd42:77af:6d53:fd2c::1/64 scope global
        valid_lft forever preferred_lft forever
root@debian:~# █

```

## 4. Gestión de redes

Por defecto LXD al iniciarlo crea una red en la que iran conectados todas las instancias automaticamente, en caso de no crearse por defecto, podremos hacerlo con el siguiente comando.

- `lxc network create [nombre_red] --type=[nombre_tipo] [opciones de configuración]`

Crear nuestras propias redes:

- `lxc network create [nombre_red] [opciones]`

```

mjbp@debian:~$ lxc network ls
+-----+-----+-----+-----+-----+-----+-----+-----+
| NAME | TYPE | MANAGED | IPV4 | IPV6 | DESCRIPTION | USED BY | STATE |
+-----+-----+-----+-----+-----+-----+-----+-----+
| eno1 | physical | NO | | | | 0 | |
+-----+-----+-----+-----+-----+-----+-----+-----+
| lxdbr0 | bridge | YES | 10.40.111.1/24 | fd42:a512:2286:5ed2::1/64 | | 3 | CREATED |
+-----+-----+-----+-----+-----+-----+-----+-----+
mjbp@debian:~$ lxc network create red1 ipv4.address=192.168.16.1/24 ipv6.address=None ipv4.dhcp=true ipv4.dhcp.ranges="192.168.16.52-192.168.16.102" i
pv4.nat=true
Network red1 created
mjbp@debian:~$ lxc network ls
+-----+-----+-----+-----+-----+-----+-----+-----+
| NAME | TYPE | MANAGED | IPV4 | IPV6 | DESCRIPTION | USED BY | STATE |
+-----+-----+-----+-----+-----+-----+-----+-----+
| eno1 | physical | NO | | | | 0 | |
+-----+-----+-----+-----+-----+-----+-----+-----+
| lxdbr0 | bridge | YES | 10.40.111.1/24 | fd42:a512:2286:5ed2::1/64 | | 3 | CREATED |
+-----+-----+-----+-----+-----+-----+-----+-----+
| red1 | bridge | YES | 192.168.16.1/24 | none | | 0 | CREATED |
+-----+-----+-----+-----+-----+-----+-----+-----+
mjbp@debian:~$ █

```

- Modificar una red:
  - `lxc network edit [nombre_red]`
- Aplicar una red a un contenedor, no podemos hacerlo directamente, tendremos que hacerlo con la ayuda de los perfiles.
  - Podemos crear el perfil de dos formas:
    - Copiandolo del default:
      - `lxc profile copy default [nombre_perfil]`
    - Creandolo nuevo:
      - `lxc profile create [nombre_perfil]`
  - Modificamos el perfil creado anteriormente para así poder añadir la nueva red al perfil creado:
    - `lxc profile edit perfil1`

```

GNU nano 4.8 /tmp/lxd_editor_3422138638.yaml
### This is a YAML representation of the profile.
### Any line starting with a '#' will be ignored.
###
### A profile consists of a set of configuration items followed by a set of
### devices.
###
### An example would look like:
### name: onenic
### config:
###   raw.lxc: lxc.aa_profile=unconfined
### devices:
###   eth0:
###     nictype: bridged
###     parent: lxdbr0
###     type: nic
###
### Note that the name is shown but cannot be changed

config: {}
description: Perfil red1
devices:
  eth0:
    name: eth0
    network: red1
    type: nic
name: perfil1
used_by: []

```

- Ver contenido de un perfil:
  - `lxc profile show perfil1`
- Aplicar un perfil a una instancia:
  - `lxc profile add [nombre_instancia] [nombre_perfil]`
- Ver perfiles aplicados a un contenedor
  - `lxc ls -c n,4,s,P`

```

mjbp@debian:~$ lxc start c1
mjbp@debian:~$ lxc profile add c1 perfil1
Profile perfil1 added to c1
mjbp@debian:~$ lxc ls -c n,4,s,P
+-----+-----+-----+-----+
| NAME |          IPV4          | STATE | PROFILES |
+-----+-----+-----+-----+
| c1   | 192.168.16.53 (eth0) | RUNNING | default |
|      |                       |         | perfil1  |
+-----+-----+-----+-----+
| c2   |                       | STOPPED | default  |
+-----+-----+-----+-----+
mjbp@debian:~$ █

```

- Borrar un perfil de un contenedor:
  - `lxc profile remove c1 perfil1`
- Configurar ACL: Son reglas de tráfico que permiten controlar el acceso a la red entre diferentes instancias conectadas a la misma red y el acceso hacia y desde otras redes.
  - Creamos las ACL:
    - `lxc network acl create [nombre_acl] [configuración_acl]`
  - Agregar ACL:
    - `lxc network acl rule add [nombre_acl] [dirección/propiedad] [propiedades]`
  - Eliminar ACL:
    - `lxc network acl rule remove [nombre_acl] [dirección/propiedad] [propiedades]`
  - Donde:
    - Dirección/propiedad:
      - name: Nombre único de la red ACL en el proyecto
      - description: Descripción de la red
      - ingress: Reglas de tráfico de salida
      - egress: Reglas de tráfico de salida

- config: opciones configuración (clave/valor)
- Propiedades:
  - action: Acción a realizar para el trafico (allow, reject, drop)
  - state: Estado de la regla (enabled, disabled o logged)
  - description: Descripción de la regla
  - source: Rangos ips
  - destinación: Rangos ips
  - source\_port: Rangos de puertos
  - .....

## 5. Gestión de almacenamiento

LXD almacena sus datos en grupos de almacenamiento, divididos en volúmenes de almacenamiento de diferentes tipos de contenido (como imágenes o instancias)

- Grupos de almacenamiento:
  - Directorio - dir
  - Btrfs - btrfs
  - Lvm - lvm
  - Zfs - zfs

**Btrfs**, es un sistema de archivos local que almacena los datos en un bloque diferente después de modificarlos en lugar de sobrescribir los datos existentes, esto hace que se reduzca el riesgo de corrupción de datos.

En nuestro caso usaremos Btrfs, lo primero que haremos sera instalarlo:

- `sudo apt install btrfs-progs`
- Tipos de contenido
  - Filesystem: Se pueden adjuntar tanto a contenedores como a máquinas virtuales, y se pueden compartir entre instancias. Se usa para contenedores e imágenes de contenedores.
  - Block: Para crearlo es necesario añadirle **--type=block**. No deben compartirse entre instancias, ya que eso puede provocar daños en los datos. Se pueden adjuntar solo a máquinas virtuales. Se usa para máquinas virtuales.
    - `lxc storage volume create [nombre_pool] [nombre_volumen] -type=block [opciones_configuración]`

Cuando iniciamos lxd después de su instalación, configuramos varios parámetros, uno de esos parámetros es el grupo de almacenamiento que en nuestro caso usamos Btrf, eso no significa que no podamos usar los demás en diferentes instancias, ya que al crear las instancias o al asociar el volumen a cada instancia podremos configurar el grupo de almacenamiento (**--storage**)

- Esta configuración se puede establecer de cualquiera de las siguientes maneras:
  - Directamente en una instancia:
    - `lxc launch [nombre_imagen] [nombre_instancia] --storage [nombre_pool]`



- A través de un perfil:
    - `lxc profile device add [nombre_perfil] root disk path=/pool=[nombre_pool] lxc launch [nombre_instancia] -profile [nombre_perfil]`
- Crear grupo de almacenamiento (Nosotros nos centraremos en el driver Btrfs)
  - Crear un grupo respaldado en bucle:
    - `lxc storage create [nombre_pool] btrfs`
  - Usamos el sistema de archivos Btrfs existente en `/some/path`:
    - `lxc storage create pool2 btrfs source=/some/path`
  - Crear un grupo en `/dev/sdX`:
    - `lxc storage create pool3 btrfs source=/dev/sdX`
- Cambiar el tamaño de un grupo de almacenamiento
  - Por ejemplo si queremos hacer crecer un grupo Btrfs respaldado por bucle en 5 Gigabytes tendremos que seguir los siguientes pasos:
    - `sudo truncate -s +5G <LXD_lib_dir>/disks/<nombre_pool>.img`
    - `sudo losetup -c <loop_device>`
    - `sudo btrfs filesystem resize max <LXD_lib_dir>/storage-pools/<nombre_pool>/`
  - Reemplace las siguientes variables:
    - `<LXD_lib_dir>`
      - `/var/snap/lxd/common/mntns/var/snap/lxd/common/lxd`
      - `/var/lib/lxd/`
    - `<nombre_pool>`
      - El nombre del grupo de almacenamiento
    - `<loop_device>`
      - El dispositivo de bucle montado que esta asociado con la imagen del grupo de almacenamiento (`/dev/loop8`). Para encontrar nuestro dispositivo de bucle:
        - `/var/snap/lxd/common/mntns/var/snap/lxd/common/lxd//disks/[nombre_pool].img losetup -l`
      - Para enumerar todos los dispositivos de bucle montados:
        - `losetup -j`
- Mover volúmenes de almacenamiento de instancias a otro grupo.
  - Tendremos que asegurarnos antes de que la instancia esta detenida. Una vez nos aseguramos de esto, vamos a mover la instancia a un grupo diferente con el siguiente comando:
    - `lxc move [nombre_instancia] --storage [target_pool_name]`
- Crear volumen grupo de almacenamiento Btrfs
  - `lxc storage volume create [nombre_pool] [nombre_volumen] [opciones_configuración]`
- Adjuntar un volumen a una instancia
 

Después de crear un volumen de almacenamiento personalizado, puede agregarlo a una o más instancias como un dispositivo de disco.

  - Se aplica las siguientes restricciones(Tipo Contenido **Block**):
    - Los volúmenes de almacenamiento de tipo de contenido **block** no se pueden adjuntar a contenedores, sino solo a máquinas virtuales.

- Para evitar la corrupción de datos, los volúmenes de almacenamiento de tipo de contenido **block** nunca deben adjuntarse a más de una maquina virtual a la vez.
  - Comando - Tipo de contenido **Filesystem**
    - lxc storage volume attach [nombre\_pool] [nombre\_volumen\_filesystem] [nombre\_instancia] [location]
  - Comando - Tipo de contenido **Block**
    - lxc storage volume attach [nombre\_pool] [nombre\_volumen\_block] [nombre\_instancia]
- Usar el volumen para copias de seguridad o imágenes
  - Para usar un volumen personalizado para almacenar los archivos comprimidos de copia de seguridad:
    - lxc config set storage.backups\_volume [nombre\_pool]/[nombre\_volumen]
  - Para usar un volumen personalizado para almacenar los tarballs de imágenes:
    - lxc config set storage.images\_volume [nombre\_pool]/[nombre\_volumen]
- Cambiar el tamaño de un volumen de almacenamiento:
  - lxc storage volume set [nombre\_pool] [nombre\_volumen] size [nuevo\_tamaño]
- Copiar volumen de almacenamiento personalizado
  - lxc storage volume copy [source\_pool\_name]/[source\_volumen\_name] [target\_pool\_name]/[target\_volume\_pool]
  - Si agregamos la opción **-volume-only** hace que solo se copie el volumen y omita las instantáneas.
  - Si agregamos la opción **-refresh** actualiza la copia
- Mover o cambiar el nombre de los volúmenes de almacenamiento personalizados
  - lxc storage volume move [source\_pool\_name]/[source\_volume\_name] [target\_pool\_name]/[target\_volume\_name]

## 5.1 Copias de seguridad de los volúmenes

- Existen diferentes formas de realizar una copia de seguridad de los volúmenes de almacenamiento personalizados:
  - Usando instantáneas para la copia de seguridad
 

Las instantáneas son rápidas y ahorran espacio pero se almacenan en el mismo grupo de almacenamiento que el volumen, eso hace que no sean demasiado confiables.

    - Crea una instantánea de un volumen de almacenamiento personalizado
      - lxc storage volume snapshot [nombre\_pool] [nombre\_volumen] [nombre\_snapshot]
      - Si se añade **-reuse** reemplaza la instantánea anterior
    - Mostrar las instantáneas de un volumen de almacenamiento:
      - lxc storage volume info [nombre\_pool] [nombre\_volumen]
    - Mostrar información sobre una instantánea:

- `lxc storage volume show [nombre_pool] [nombre_volumen]/[nombre_snapshot]`
- Editar una instantánea de un volumen:
  - `lxc storage volume edit [nombre_pool] [nombre_volumen]/[nombre_snapshot]`
- Eliminar una instantánea:
  - `lxc storage volume delete [nombre_pool] [nombre_volumen]/[nombre_snapshot]`
- Programar instantánea de un volumen:
  - `lxc storage volume set [nombre_pool] [nombre_volumen] snapshots.schedule @daily`
  - `lxc storage volume set [nombre_pool] [nombre_volumen] snapshots.schedule "0 6 * * *"`
- Restaurar una instantánea de un volumen:
 

Antes de restaurar alguna instantánea debemos de parar todas las instancias que usen ese volumen.

  - `lxc storage volume restore [nombre_pool] [nombre_volumen] [nombre_snapshot]`
  - También puedes restaurar la instantánea en un nuevo volumen, ya sea en el mismo grupo de almacenamiento o en otro diferente:
    - `lxc storage volume copy [source_pool_name]/[source_volume_name]/[source_snapshot_name] [target_pool_name]/[target_volume_name]`
- Usar archivos de exportación para la copia de seguridad
 

Los archivos de exportación se pueden almacenar en diferentes discos, eso hace que sea mas fiable.

  - Exportar un volumen de almacenamiento
    - Para exportarlo a un archivo comprimido, usaríamos el siguiente comando:
      - `lxc storage volume export [nombre_pool] [nombre_volumen] [file_path]`
    - En caso de no especificar una ruta de archivo, el archivo de exportación se guarda como **backup.tar.gz** en el directorio de trabajo
  - Restaurar volumen de almacenamiento desde un archivo de exportación
    - `lxc storage volume import [nombre_pool] [file_path] [nombre_volumen]`

# 6. Configuración LXD

## 6.1 Comandos Básicos

- Crear contenedor :
  - `lxc launch [ubuntu:16.04] [nombre_contenedor]`
- Listar todos los contenedores:
  - `lxc list`
- Acceder a un contenedor:
  - `lxc exec [nombre_contenedor] bash`
- Dentro del contenedor podríamos configurar la fecha y la hora del sistema:
  - `timedatectl set-timezone Europe/Madrid`
- Filtrar instancias por nombre, tipo, estado o miembro del clúster donde se encuentra la instancia.
  - `lxc list type=container`
  - `lxc list status=running`
  - `lxc list location=server1`
  - `lxc list ubuntu.*`
- Mostrar información detallada sobre una instancia
  - `lxc info [nombre_instancia]`
- Mostrar las ultima líneas de registro para la instancia
  - `lxc info [nombre_instancia] --show-log`
- Iniciar una instancia
  - `lxc start [nombre_instancia]`
- Conectarnos a la consola a la misma vez que iniciamos la instancia
  - `lxc start [nombre_instancia] -- console`
- Detener una instancia
  - `lxc stop [nombre_instancia]`
- Eliminar una instancia
  - `lxc delete [nombre_instancia]`
- Evitar la eliminación de una instancia accidentalmente. Para poder hacer esto, tenemos dos opciones:
  - Si queremos que pida confirmación antes de eliminar las instancias, debemos ejecutar el siguiente comando:
    - `lxc alias add delete "delete -i"`
  - Si queremos configurarlo solo para instancias específicas, usaremos el siguiente comando:
    - `lxc config set [nombre_instancia] security.protection.delete=true`
- Ejecutar un comando en la instancia sin tener que acceder a la consola
  - `lxc exec [nombre_contenedor] - [comando]`
- Limitar el uso de memoria y el numero de CPU al crear un contenedor:
  - `lxc launch images:debian/11 [nombre_instancia] -c limits.cpu=1 -c limits.memory=512MB`
    - Comprobamos limite de cpu:

- nproc
  - Comprobamos el limite de memoria:
    - free --mega
- Limitando recursos de contenedor creado en caliente
  - Comandos base para limitar funciones y recursos:
    - lxc config set [nombre\_instancia] [opciones]
  - Ejemplo:
    - Limitar CPU de un contenedores
      - lxc config set [nombre\_instancia] limits.cpu 1
    - Limitar memoria de un contenedores
      - lxc config set [nombre\_instancia] limits.memory 2048 MB
- Editar archivos de una instancia
  - Para editar un archivo desde nuestra anfitriona tendremos que usar el siguiente comando:
    - lxc file edit [nombre\_instancia]/[ruta\_archivo]
  - Por ejemplo si queremos editar el fichero /etc/hosts:
    - lxc file edit c1/etc/hosts
- Eliminar archivos de una instancia
  - Para eliminar un archivo de una instancia desde nuestra anfitriona, usaremos el siguiente comando:
    - lxc file delete [nombre\_instancia]/[ruta\_archivo]
- Ver volúmenes de almacenamiento:
  - lxc storage volume list [nombre\_pool]
- Mostrar información detallada sobre un volumen personalizado específico:
  - lxc storage volume show [nombre\_pool] [nombre\_volumen]
- Mostrar información detallada sobre un volumen de instancia específico:
  - lxc storage volume show [nombre\_pool] [type\_volumen]/[nombre\_volumen]

## 6.2 Transferir archivos a/desde contenedores LXD

Podemos transferir archivos a nuestros contenedores y también podemos transferirnos esos archivos/directorios a nuestra maquina anfitriona.

- Cargar un archivo a un contenedor desde nuestra maquina anfitriona:

```
mjbp@debian:~$ echo "Hola, Este fichero se lo envio a c1 desde mi maquina fisica" > prueba.txt
mjbp@debian:~$ ls -l
total 177616
drwxr-xr-x 3 mjbp mjbp      4096 nov 20 13:47 Descargas
drwxr-xr-x 2 mjbp mjbp      4096 nov 20 13:47 Documentos
drwxr-xr-x 2 mjbp mjbp      4096 oct 16 12:07 Escritorio
-rw-r--r-- 1 mjbp mjbp 181834299 dic  6 10:53 f63aed81a129603cc3e62be9522b38faf683dbb1f483daba0c206adf9fb3de7b.tar.gz
drwxr-xr-x 3 mjbp mjbp      4096 dic  6 11:11 Imágenes
drwxr-xr-x 2 mjbp mjbp      4096 oct 16 12:07 Música
drwxr-xr-x 2 mjbp mjbp      4096 oct 16 12:07 Plantillas
-rw-r--r-- 1 mjbp mjbp        60 dic  6 11:24 prueba.txt
drwxr-xr-x 2 mjbp mjbp      4096 oct 16 12:07 Público
drwx----- 3 mjbp mjbp      4096 nov 13 20:29 snap
drwxr-xr-x 2 mjbp mjbp      4096 oct 16 12:07 Videos
mjbp@debian:~$ █
```

- Cargar un directorio a un contenedor desde nuestra maquina anfitriona:

```
mjbp@debian:~$ ls -l prueba/
total 4
-rw-r--r-- 1 mjbp mjbp 60 dic  6 11:24 prueba.txt
mjbp@debian:~$ lxc file push prueba/ c2/home/ --recursive --verbose
INFO [2022-12-06T11:28:57+01:00] Pushing /var/lib/snapd/hostfs/home/mjbp/prueba
a to /home/prueba (directory)
INFO [2022-12-06T11:28:57+01:00] Pushing /var/lib/snapd/hostfs/home/mjbp/prueba
a/prueba.txt to /home/prueba/prueba.txt (file)
mjbp@debian:~$ lxc shell c2
root@c2:~# pwd /home/
/root
root@c2:~# ls -l /home/
total 0
drwxr-xr-x 1 1000 1000 20 Dec  6 10:28 prueba
root@c2:~# ls -l /home/prueba/
total 4
-rw-r--r-- 1 1000 1000 60 Dec  6 10:28 prueba.txt
root@c2:~#
```

- Descargar un directorio desde el contenedor a nuestra maquina anfitriona:

```
mjbp@debian:~$ lxc file push prueba.txt c1/home/
mjbp@debian:~$ lxc shell c1
root@c1:~# ls -l /home/
total 4
-rw-r--r-- 1 1000 1000 60 Dec  6 10:26 prueba.txt
root@c1:~# cat prueba.txt
cat: prueba.txt: No such file or directory
root@c1:~#
```

```
mjbp@debian:~$ ls -l prueba
ls: no se puede acceder a 'prueba': No existe el fichero o el directorio
mjbp@debian:~$ lxc file pull c2/home/prueba /home/mjbp --recursive --verbose
INFO [2022-12-06T11:34:22+01:00] Pulling /var/lib/snapd/hostfs/home/mjbp/prueba
a from home/prueba (directory)
INFO [2022-12-06T11:34:22+01:00] Pulling /var/lib/snapd/hostfs/home/mjbp/prueba
a/prueba.txt from home/prueba/prueba.txt (file)
mjbp@debian:~$ ls -l /home/mjbp/prueba/
total 4
-rw-r--r-- 1 mjbp mjbp 60 dic  6 11:34 prueba.txt
mjbp@debian:~$ ls
Descargas
Documentos
Escritorio
f63aed81a129603cc3e62be9522b38faf683dbb1f483daba0c206adf9fb3de7b. tar.gz
Imágenes
Música
Plantillas
prueba
Público
snap
Videos
mjbp@debian:~$
```

## 6.3 Crear backups contenedores LXD

Se puede crear una copia de seguridad de todas los contenedores/maquinas de lxd y también podremos importar todo o solamente algún contenedor en concreto A la hora de crear backups es importante tener en cuenta los diferentes objetos que LXD almacena/administra:

- Instancias (registros de bases de datos y sistemas de archivos)
- Imágenes (registros de bases de datos, archivos de imágenes y sistemas de archivos)
- Redes (registro de bases de datos y archivos de estado)
- Perfiles (registros de base de datos)

- Volúmenes de almacenamiento (registro de base de datos y sistemas de archivos)
  - Una copia de seguridad completa incluiría la totalidad de /var/lib/lxd o en nuestro caso la totalidad de /var/snap/lxd/common/lxd
  - Si ejecutamos el siguiente comando, nos informara sobre la red actual, el grupo de almacenamiento y el perfil:
- lxc init -dump

## Exportar contenedores en tarballs

Creamos un snapshot del contenedor

```
mjbp@debian:~$ lxc list
```

NAME	STATE	IPV4	IPV6	TYPE	SNAPSHOTS
mvubuntu	STOPPED			VIRTUAL-MACHINE	1
prueba1	RUNNING	10.40.111.224 (eth0)	fd42:a512:2286:5ed2:216:3eff:fe21:8ed (eth0)	CONTAINER	0
prueba2	RUNNING	10.40.111.98 (eth0)	fd42:a512:2286:5ed2:216:3eff:fe63:b2d5 (eth0)	CONTAINER	0

```
mjbp@debian:~$
```

Listamos los contenedores para ver si tiene un snapshot creado:

```
mjbp@debian:~$ lxc snapshot prueba1 backup
mjbp@debian:~$ lxc list prueba1
```

NAME	STATE	IPV4	IPV6	TYPE	SNAPSHOTS
prueba1	RUNNING	10.40.111.224 (eth0)	fd42:a512:2286:5ed2:216:3eff:fe21:8ed (eth0)	CONTAINER	1

Publicamos una imagen del snapshot:

```
mjbp@debian:~$ lxc publish prueba1/backup --alias prueba-backup
Instance published with fingerprint: 14978ebe534bc3934ea0a03bf877e1ada001b67bf6f226145350350154f87419
mjbp@debian:~$
```

Exportamos la imagen a un tarball:

```
mjbp@debian:~$ lxc image export prueba-backup
Image exported successfully!
mjbp@debian:~$
```

Listamos el contenedor exportado a un tarball:

```
mjbp@debian:~$ ls -alht 14978ebe534bc3934ea0a03bf877e1ada001b67bf6f226145350350154f87419.tar.gz
-rw-r--r-- 1 mjbp mjbp 114M dic 15 23:07 14978ebe534bc3934ea0a03bf877e1ada001b67bf6f226145350350154f87419.tar.gz
mjbp@debian:~$
```

Importamos el tarball:

```
mjbp@debian:~$ lxc launch prueba-backup prueba
Creating prueba
Starting prueba
mjbp@debian:~$ lxc list
```

NAME	STATE	IPV4	IPV6	TYPE	SNAPSHOTS
mvubuntu	STOPPED			VIRTUAL-MACHINE	1
prueba	RUNNING	10.40.111.149 (eth0)	fd42:a512:2286:5ed2:216:3eff:fe47:c61c (eth0)	CONTAINER	0
prueba2	RUNNING	10.40.111.98 (eth0)	fd42:a512:2286:5ed2:216:3eff:fe63:b2d5 (eth0)	CONTAINER	0

## 6.4 Instantáneas de las instancias

Podemos crear snapshot de nuestras instancias, veremos como crear un snapshot, como programar instantáneas de instancias y también veremos como podemos restaurarla

- Creamos una instantánea:
  - `lxc snapshot [nombre_instancia] [nombre_instantánea]`
  - En el caso de querer reemplazar una instantánea ya existente usaremos en el comando anterior la opción **--reuse**
  - Las instantáneas no expiran a no ser que lo configuremos y le establezcamos la opción **snapshots.expiry**. En caso de establecer un tiempo de caducidad para todas las instantáneas, podremos usar la bandera **-no-expiry** para una instantánea concreta.
- Ver, editar o eliminar instantaneas:
  - Ver las instantáneas de una instancia:
    - `lxc infor [nombre_instancia]`
  - Ver/Modificar instantáneas haciendo referencia directa a esa instantánea:
    - `lxc info [nombre_instancia]/[nombre_instantánea]`
  - Mostrar información de configuración sobre una instantánea:
    - `lxc show [nombre_instancia]/[nombre_instantánea]`
  - Editar instantánea:
    - `lxc config edit [nombre_instancia]/[nombre_instantánea]`
  - Eliminar una instantánea:
    - `lxc delete [nombre_instancia]/[nombre_instantánea]`
- Programar instantáneas de instancias
  - Para configurar instantáneas diarias usaremos este comando:
    - `lxc config set [nombre_instancia] snapshots.schedule @daily`
  - Para configurar una instantánea todos los días a las 6 de la mañana, usaremos el siguiente comando:
    - `lxc config set [nombre_instancia] snapshots.schedule "0 6 * * *`
  - Si queremos crear instantaneas de las instancias que no se estan ejecutando, usaremos **snapshots.schedule.stopped**.
- Restaurar una instantánea de una instancia
  - `lxc restore [nombre_instancia] [nombre_instantanea]`
  - En el caso de que la instancia tenga estado, puedes agregar el **--stateful** que es un indicador para restaurar el estado.



## 6.5 Limites instancias

Cuando creamos las instancias o incluso después podemos ponerle algunos limites, tanto de CPU, como de memoria, como en red y también en disco.

En que situaciones podría ser útil estos limites:

- Por ejemplo si nos ponemos en la situación de tener un servidor grande en el que tiene varios clientes en el con sitios web, y un contenedor recibe una rafaga repentina de tráfico, esto podria ralentizar las otras instancias. Si tenemos limites, solo un contenedor se ralentizará, el resto funcionara con normalidad.

Vamos a ver como serian los comandos para configurar estos limites en LXD:

- Limitar CPU
  - `lxc config set [nombre_instancia] limits.memory 100MB`
- Limitar Memoria RAM
  - `lxc config set [nombre_instancia] limits.cpu 2`
  - Si queremos fijar núcleos de CPU específicos
    - `lxc config set [nombre_instancia] limits.cpu 0-0`
    - Esto haria que el contenedor solo use la primera CPU, para usar el segundo, debemos poner 1-1. Si queremos usar todos los núcleos del primero al tercero, debemos poner 0-3.
  - Si queremos limitar una cantidad de tiempo de CPU que puede usar un contenedor
    - `lxc config set [nombre_instancia] limits.cpu.allowance 10ms/100ms`
    - Esto permitiría que el contenedor use 10 milisegundos de tiempo de CPU de cada 100 milisegundos, por lo que alrededor del 10% de un núcleo de CPU
- Limitar Red
  - Agregamos un dispositivo Ethernet Virtual
    - `lxc config device add [nombre_instancia] eth0 nic name=eth0 nictype=bridged parent=lxdbro`
  - Establecemos limites en la entrada (descarga) y en la salida (carga) de la red
    - `lxc config device set [nombre_instancia] eth0 limits.ingress 1Mbit`
  - Para la salida:
    - `lxc config device set [nombre_instancia] eth0 limits.egress 1Mbit`
- Limitar Uso de Disco
  - Antes de limitar los recursos relacionados con el disco, debemos de agregar un dispositivo de disco raiz a nuestro contenedor:
    - `lxc config device add [nombre_instancia] root disk pool=default path=/`
  - Limitar espacio en disco:
    - `lxc config device set [nombre_instancia] root size 7GB`

# 7. Maquinas Virtuales – Comandos Básicos

Si queremos usar la consola VGA con salida grafica para nuestra maquina virtual, tendremos que instalar un cliente Spice, podrian ser (virt-viewer o spice-gtk-client):

- `sudo apt install virt-viewer`

LXD admite imágenes de escritorio que nos permite iniciar una máquina virtual de escritorio sin necesidad de configuración adicional

- Crear máquina virtual
  - `lxc launch ubuntu:22.04 [nombre_mv] --vm`
- Iniciar una maquina virtual y crearla al mismo tiempo:
  - `lxc launch images:ubuntu:20.04 [nombre_vm] --vm -c limits.memory=4GiB -c limits.cpu=4 --console=vga`

Realmente los demás comandos son prácticamente igual que para gestionar contenedores. Cambia un poco mas a la hora de crear, ya que hay que indicarle que es una maquina virtual pero por lo demás es prácticamente igual. También indicándole la consola, esta funcionalidad nos da la opción de ver la maquina con entorno gráfico, no solo por comandos.

Comandos para acceder a la consola

- Consola interactiva:
  - `lxc console [nombre_mv]`
- Mostrar salida del registro, se usa la bandera `--show-log`:
  - `lxc console [nombre_mv] --show-log`
- También nos podemos conectarnos nada mas iniciar la mv:
  - `lxc start [nombre_mv] --console`
  - `lxc start [nombre_mv] --console=vga`

**Pruebas** --> Creamos una maquina virtual en la que instalamos Apache2 y configuramos la pagina web, en esta misma maquina virtual crearemos una instantánea del estado actual. Una vez tengamos la instantánea creada, podríamos recuperarla en una maquina nueva pero en nuestro caso restauraremos la anterior, simplemente lo que haremos sera devolverla al estado que estaba cuando hicimos la instancia:

Instalamos Virt-Viewer para poder visualizar gráficamente las maquinas virtuales:

```

root@debian:~# apt install virt-viewer
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
dmeventd libaio1 libcacard0 libdevmapper-event1.02.1 libgiovann-2.0 libglib2.0-0 libglib2.0-common libglib2.0-glib-2.0-0 libglib2.0-glib-2.0-common libspice-client-glib-2.0-0 libspice-client-gtk-3.0-5 libusbredirhost1 libusbredirparser1 libvirt-glib-1.0-0 libvirt0 lvm2
spice-client-glib-usb-acl-helper thin-provisioning-tools
Se instalarán los siguientes paquetes NUEVOS:
dmeventd libaio1 libcacard0 libdevmapper-event1.02.1 libgiovann-2.0 libglib2.0 libglib2.0-common libglib2.0-glib-2.0-0 libglib2.0-glib-2.0-common libspice-client-glib-2.0-0 libspice-client-gtk-3.0-5 libusbredirhost1 libusbredirparser1 libvirt-glib-1.0-0 libvirt0 lvm2
libphodav-2.0-common libspice-client-glib-2.0-0 libspice-client-gtk-3.0-5 libusbredirhost1 libusbredirparser1 libvirt-glib-1.0-0 libvirt0 lvm2
0 actualizados, 21 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 7,700 kB de archivos.
Se utilizarán 38,2 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S

```

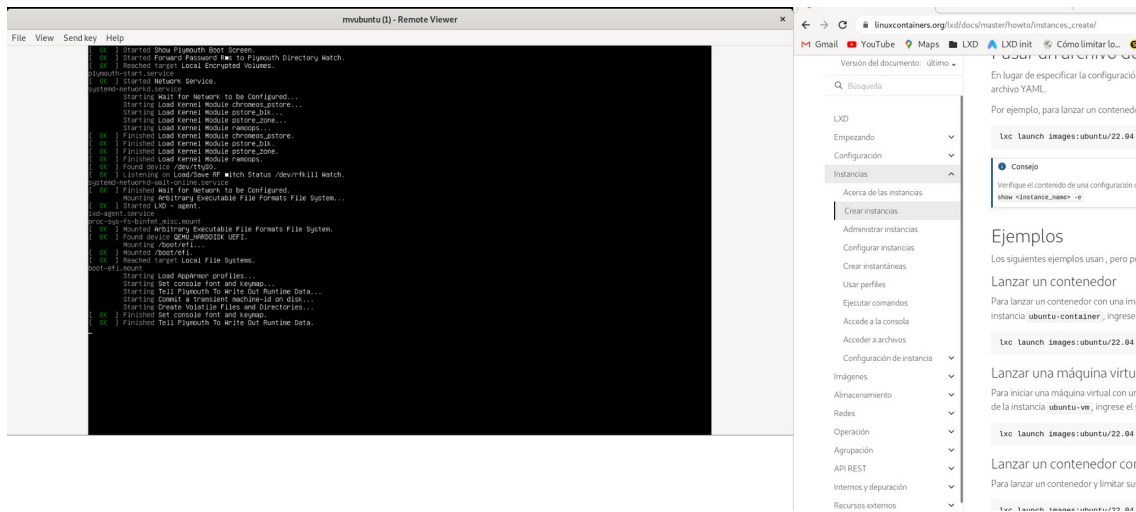
## Creamos la maquina virtual:

```

mjb@debian:~$ lxc launch images:ubuntu/20.04/desktop mvubuntu --vm -c limits.memory=4GiB --console=vga
Creating mvubuntu
Retrieving image: Unpack: 100% (2.19GB/s)

```

En la misma instalación al indicar el parametro `-console` nos abra la maquina virtual:



Tenemos la maquina virtual ya creada:

```

mjb@debian:~$ lxc list
+-----+-----+-----+-----+-----+-----+
| NAME | STATE | IPV4 | IPV6 | TYPE | SNAPSHOTS |
+-----+-----+-----+-----+-----+-----+
| mvubuntu | RUNNING | 10.40.111.52 (eth0) | fd42:a512:2286:5ed2:216:3eff:fed9:f640 (eth0) | VIRTUAL-MACHINE | 0 |
+-----+-----+-----+-----+-----+-----+
mjb@debian:~$

```

Nos conectamos al contenedor y hacemos un `ip` a para ver que realmente tiene la `ip`:

```

mjb@debian:~$ lxc exec mvubuntu -- /bin/bash
root@mvubuntu:~# 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
2: enp5s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:16:3e:d9:f6:40 brd ff:ff:ff:ff:ff:ff
    inet 10.40.111.52/24 brd 10.40.111.255 scope global dynamic enp5s0
        valid_lft 3378sec preferred_lft 3378sec
    inet6 fd42:a512:2286:5ed2:216:3eff:fed9:f640/64 scope global dynamic mngtmpa
        valid_lft 3381sec preferred_lft 3381sec
    inet6 fe80::216:3eff:fed9:f640/64 scope link
        valid_lft forever preferred_lft forever
root@mvubuntu:~# █

```

Sin entrar en la maquina virtual, instalamos apache2:

```

mjb@debian:~$ lxc exec mvubuntu -- apt install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap libcurl4 liblua5.2-0
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom ufw
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap libcurl4 liblua5.2-0
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
Need to get 2056 kB of archives.
After this operation, 8660 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y█

```

Comprobamos que efectivamente esta instalado:

```

mjb@debian:~$ lxc exec mvubuntu -- apt policy apache2
apache2:
  Installed: 2.4.41-4ubuntu3.12
  Candidate: 2.4.41-4ubuntu3.12
  Version table:
 *** 2.4.41-4ubuntu3.12 500
    500 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages
    500 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages
    100 /var/lib/dpkg/status
 2.4.41-4ubuntu3 500
    500 http://archive.ubuntu.com/ubuntu focal/main amd64 Packages
mjb@debian:~$ █

```

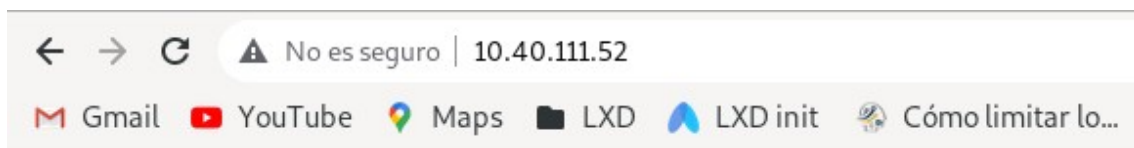
En nuestra maquina física hemos descargado una imagen para ponerla en la pagina web de la maquina virtual, la hemos metido en un directorio y vamos a pasar el directorio a la maquina virtual:

```
mjbp@debian:~/Descargas$ lxc file push imagen/ mvubuntu/var/www/html/ --recursive --verbose
INFO [2022-12-13T12:48:04+01:00] Pushing /var/lib/snapd/hostfs/home/mjbp/Descargas/imagen to /var/www/html/imagen (directory)
INFO [2022-12-13T12:48:04+01:00] Pushing /var/lib/snapd/hostfs/home/mjbp/Descargas/imagen/descarga.jpeg to /var/www/html/imagen/descarga.jpeg (file)
mjbp@debian:~/Descargas$
```

Editamos el fichero inde.html para que se pueda visualizar la imagen:

```
mjbp@debian:~$ lxc exec mvubuntu -- ls /var/www/html
imagen index.html
mjbp@debian:~$
```

Entramos en el navegador y vemos que podemos visualizar la pagina web:



**Hola**

**BIENVENIDOS a mi web con apache2**

Esta web esta en una maquina virtual creada con LXD



Creamos el snapshot de la maquina virtual:

```
mjbp@debian:~$ lxc list
+-----+-----+-----+-----+-----+-----+
| NAME | STATE | IPV4 | IPV6 | TYPE | SNAPSHOTS |
+-----+-----+-----+-----+-----+-----+
| mvubuntu | RUNNING | 10.40.111.52 (enp5s0) | fd42:a512:2286:5ed2:216:3eff:fed9:f640 (enp5s0) | VIRTUAL-MACHINE | 0 |
+-----+-----+-----+-----+-----+-----+
mjbp@debian:~$ lxc snapshot mvubuntu mvubuntu-backup
mjbp@debian:~$ lxc list
+-----+-----+-----+-----+-----+-----+
| NAME | STATE | IPV4 | IPV6 | TYPE | SNAPSHOTS |
+-----+-----+-----+-----+-----+-----+
| mvubuntu | RUNNING | 10.40.111.52 (enp5s0) | fd42:a512:2286:5ed2:216:3eff:fed9:f640 (enp5s0) | VIRTUAL-MACHINE | 1 |
+-----+-----+-----+-----+-----+-----+
mjbp@debian:~$ █
```

Snapshots:

```
+-----+-----+-----+-----+-----+
| NAME | TAKEN AT | EXPIRES AT | STATEFUL |
+-----+-----+-----+-----+
| mvubuntu-backup | 2022/12/13 12:58 CET | | NO |
+-----+-----+-----+-----+
mjbp@debian:~$ █
```

Podemos restaurar el snapshot para así volver al estado del snapshot creado:

```
mjbp@debian:~$ lxc restore mvubuntu mvubuntu-backup
mjbp@debian:~$ lxc list
+-----+-----+-----+-----+-----+-----+
| NAME | STATE | IPV4 | IPV6 | TYPE | SNAPSHOTS |
+-----+-----+-----+-----+-----+-----+
| mvubuntu | RUNNING | 10.40.111.52 (eth0) | fd42:a512:2286:5ed2:216:3eff:fed9:f640 (eth0) | VIRTUAL-MACHINE | 1 |
+-----+-----+-----+-----+-----+-----+
mjbp@debian:~$ █
```

También con el siguiente comando, crearemos un snapshot diario automáticamente:

```
mjbp@debian:~$ lxc config set mvubuntu snapshots.schedule @daily
```

## 8. Conclusión

Con este proyecto podemos ver que LXD permite hacer y configurar muchísimas cosas, permitiéndonos tener en nuestro equipo varias máquinas ya sean contenedores o máquinas virtual, pudiéndolas crear, editar, gestionar y hacer todo como si de una máquina física se tratase.

LXD tiene infinidad de opciones pero lamentablemente es imposible verlas todas.

LXD puede llegar a ser una muy buena herramienta de trabajo, a la misma vez que útil y eficiente.

## 9. Bibliografía

- <https://www.adictosaltrabajo.com/2018/07/11/amaras-lxd-por-encima-de-todas-las-cosas/>
- <https://www.youtube.com/watch?v=TmGvbXfwJEA>
- <https://www.dz-techs.com/es/introduction-ubuntu-lxd-containers>
- <https://www.cyberciti.biz/faq/how-to-backup-and-restore-lxd-containers/>
- <https://superadmin.es/blog/devops/backup-contenedores-lxd/>
- Página principal donde he cogido prácticamente toda la información:
  - <https://linuxcontainers.org/lxd/introduction/>