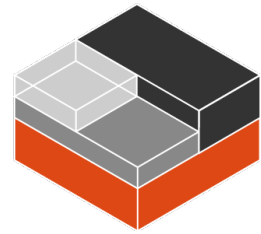


# Centro de Procesamiento de Datos



## Práctica 1. LXD

LXD permite ejecutar procesos y mantener recursos aislados de forma eficiente y compartiendo el núcleo del sistema operativo. Los dispositivos están virtualizados de forma que cada contenedor ve sus propios dispositivos. Los procesos que se ejecutan son nativos del equipo host y no es necesario crear un hardware virtualizado completo, por lo que el funcionamiento global es más eficiente.

ubunut

### Primeros pasos con contenedores LXD

Creamos un contenedor ubuntu que llamaremos c0:

```
lxc launch ubuntu: c0
```

Podemos comprobar que está funcionando

```
lxc list
```

Ver qué imágenes de contenedor tenemos

```
lxc image list
```

Accedemos mediante shell al contenedor

```
lxc exec c0 bash
```

Comprobamos la versión de Ubuntu

```
lsb_release -a
```

Copiamos el contenedor creando uno nuevo

```
lxc copy c0 c1
```

Comprobamos que se ha creado

```
lxc list
```

Activamos el contenedor

```
lxc start c1
```

Accedemos al segundo contenedores

```
lxc exec c1 bash
```

Paramos y borramos el segundo contenedor

```
lxc stop c1  
lxc delete c1
```

Mostrar la lista de imágenes disponibles

```
lxc image list images:
```

Existen 3 repositorios con imágenes:

ubuntu: (para imágenes estables de Ubuntu)

ubuntu-daily: (para imágenes diarias de Ubuntu)

images: (para un conjunto de otras distribuciones)

Creamos un contenedor basado en Centos 7 y que se llame c2

```
lxc launch images:centos/7 c2
```

Accedemos mediante shell al contenedor c2

```
lxc exec c2 bash
```

Comprobamos la versión de Centos

```
More /etc/redhat-release
```

Comprobación de la red

Activamos 2 contenedores y comprobamos el acceso entre ambos verificando la ip asignada a cada contenedor

```
lxc list
```

Compartiendo un directorio local con el contenedores

```
mkdir disco1
```

```
lxc config device add c1 disco1 disk source=/home/usuario/prueba1 path=/mnt/disco1
```

Si queremos acceso de lectura y escritura hay que comprobar los permisos del propietario del contenedor

```
sudo ls -l /var/lib/lxd/containers
```

Limitando recursos en un contenedor

```
lxc config set c1 limits.memory 512MB
```

```
lxc config show c1
```

Podemos comprobar con `lxc exec c1 /bin/bash`

`free -m`

Algunos parámetros:

boot.autostart

limits.cpu

limits.cpu.allowance

limits.memory

## Operaciones con contenedores LXD

### Creando un contenedor

Los contenedores se crean a partir de imágenes y según donde están: remotos, built-in y locales

### Órdenes básicas

Creamos un contenedor:

```
#la última versión
lxc launch ubuntu: c0

#versión específica
lxc launch ubuntu:16.04 c1

#versión 32 bits
lxc launch ubuntu:14.04/i386 c2

#versión de alpine linux (ligera y segura)
lxc launch images:alpine/3.8 c2a

#versión centos
lxc launch images:centos/7 mi_centos1
#versiones no testeadas
lxc launch ubuntu-daily:devel c3
```

### Lista de contenedores

```
lxc list
lx image list images:
```

Ya que LXD está orientado a definir contenedores con un sistema operativo, podemos encontrar: ubuntu, debian, CentOS, Oracle, gentoo, plamo, Alpine, ...

### Parar un contenedores

```
lxc stop c1
#parada forzada
lxc stop c1 --force
```

### Iniciamos un contenedor

```
lxc start c1
```

### Reiniciar

```
lxc restart c1
lxc restart c1 --force
```

#### Pausar

```
lxc pause c1
```

#### Borrar

```
lxc delete c1
```

#### Copiar un contenedor

```
Lxc copy <contenedor_origen> <contenedor_destino>
```

#### Mover un contenedor

```
Lxc move
```

#### Información

```
lxc info c1
```

#### Executamos un shell

```
lxc exec c1 bash
```

#### Leer un fichero del contenedor

```
lxc file pull <contenedor>/<path> <dest>
```

#### Grabar un fichero en el contenedor

```
lxc file push <source> <contenedor>/<path>
```

#### Editar un fichero

```
lxc file edit <contenedor>/<path>
```

#### Snapshots:

```
lxc snapshot <contenedor> <nombre_contenedor>
```

#### Restaurar

```
lxc restore <contenedor> <nombre_contenedor>
```

#### Renombrar

```
lxc move <contenedor>/<nombre_contenedor> <contenedor>/<nuevo_nombre_contenedor>
```

#### Copiar un contenedor desde un snapshot

```
lxc copy <contenedor>/<nombre_contenedor> <nuevo_contenedor>
```

#### Borrar snapshot

```
lxc delete <contenedor>/<nombre_contenedor>
```

#### Importar

```
lxc image import <fichero> --alias mi_alias  
#después se puede utilizar dicho alias  
lxc launch mi_alias <contenedor>
```

#### Publicar

```
lxc publish <contenedor> --alias nombre
```

#### Interfaz en modo bridge

```
lxc config device add c1 eth1 nic nictype=bridged parent=lxdbr0
```

#### Accediendo remotamente a los contenedores

Creamos las claves RSA y utilizamos ssh-copy-id