

Introducción e Investigación de Contenedores LXC/LXD

Proyecto de Fin de FP Administración de Sistemas Informáticos en Red



12 de abril de 2023

Ies Francisco Romero Vargas

Índice

[1. Introducción 3](#_Toc137206219)

[1.1. ¿En qué consiste LXD? 3](#_Toc137206220)

[1.2. Características de LXD 3](#_Toc137206221)

[1.3. Ventajas e inconvenientes de usar LXD 4](#_Toc137206222)

[1.3.1. Ventajas 4](#_Toc137206223)

[1.3.2. Desventajas 4](#_Toc137206224)

[2. Finalidad del proyecto 4](#_Toc137206225)

[3. Objetivos 5](#_Toc137206226)

[4. Medios necesarios 5](#_Toc137206227)

[5. Planificación 5](#_Toc137206228)

[5.1. Presupuesto del proyecto 5](#_Toc137206229)

[5.2. Esquematización del proyecto. 6](#_Toc137206230)

[5.3. Asignaturas empleadas en el proyecto. 6](#_Toc137206231)

[6. Elaboracion del proyecto 6](#_Toc137206232)

[6.1. Instalación del software LXC y LXD 6](#_Toc137206233)

[6.2. Redes 9](#_Toc137206234)

[6.2.1. Creación de una nueva red 9](#_Toc137206235)

[6.2.2. Modificar una red ya existente 10](#_Toc137206236)

[6.2.3. Usar una red en un contenedor (perfiles) 11](#_Toc137206237)

[6.2.4. ACL 14](#_Toc137206238)

[6.3. Almacenamiento 15](#_Toc137206239)

[6.4. Configuración de instancias, opciones diversas. 24](#_Toc137206240)

[6.5. Máquinas virtuales 34](#_Toc137206241)

[6.6. Demostración de mini sistema de red 37](#_Toc137206242)

[6.6.1. Instalación de la máquina virtual host 37](#_Toc137206243)

[6.6.2. Instalación del sistema operativo 42](#_Toc137206244)

[6.6.3. Configuración adaptadores de red 55](#_Toc137206245)

[6.6.4. Instalación de software 57](#_Toc137206246)

[6.6.5. Configuración inicial de LXD 59](#_Toc137206247)

[6.6.6. Creación de la red y perfil aplicable 60](#_Toc137206248)

[6.6.7. Creacion de servidor WEB 62](#_Toc137206249)

[6.6.8. Servidor DNS 67](#_Toc137206250)

[6.7. DHCP 72](#_Toc137206251)

[6.8. MySQL + APACHE 79](#_Toc137206252)

[6.8.1. MariaDB 79](#_Toc137206253)

[6.8.2. Apache 82](#_Toc137206254)

[6.9. VSFTPD (FTP) 84](#_Toc137206255)

[6.10. Automatización de creación de máquinas host 86](#_Toc137206256)

[6.11. Esquema de red final 90](#_Toc137206257)

[7. Problemas encontrados 91](#_Toc137206258)

[8. Posibles mejoras 92](#_Toc137206259)

[9. Conclusión 92](#_Toc137206260)

[10. Enlaces y referencias de la documentación (Bibliografía) 93](#_Toc137206261)

# Introducción

## ¿En qué consiste LXD?

LXD o Linux Container Daemon, es una herramienta que se utiliza principalmente para la gestión de contenedores para distribuciones Linux.

Te permite gestionar los contenedores creados con LXC con un uso muy enfocado a la nube. Con la herramienta de LXD podemos crear varios contenedores y máquinas virtuales dentro de un mismo sistema operativo aprovechando mejor los recursos de nuestra máquina anfitriona, pudiendo virtualizar aplicaciones aisladas en ellos o un sistema operativo completo.

El objetivo principal de usar un sistema con contenedores LXD es la posibilidad de poder virtualizar un sistema entero en un contenedor con aplicaciones o servicios que tengan un ciclo de vida largo o medio en el tiempo, como un servidor DHCP o DNS.

## Características de LXD

* Todas las imágenes de Linux funcionaran y son compatibles con esta herramienta.
* Es fácil e intuitivo de utilizar, más que un cliente de Docker normal, cuenta con una api clara y sencilla y su cliente con una línea de comandos fácil de utilizar.
* Puedes gestionar contenedores que sean individuales o contenedores que estén repartidos en distintos entornos por lo cual son escalables.
* Gracias a que se ejecutan en espacios aislados, solo pueden acceder a unos recursos predefinidos, por lo cual son bastante seguras.
* Se puede establecer un límite para la memoria que quieres que consuma, la CPU e incluso el uso de la red más el almacenamiento, por lo cual su control de recursos es excepcional.
* Admite varios back- ends de almacenamiento, grupos y volúmenes por lo cual se puede gestionar el almacenamiento de una manera muy eficiente.
* Permite crear puentes, túneles de red y crear redes de una forma muy conveniente, por lo cual también tiene una buena gestión de red.
* Los contenedores que se crean tienen acceso a dispositivos USB, GPU y medios de almacenamiento masivo siempre y cuando la configuración lo permita.

## Ventajas e inconvenientes de usar LXD

Como todo software o sistema que exista, todo tiene que tener unas ventajas y unas desventajas, las principales que yo he encontrado para LXD son las siguientes:

### Ventajas

Puedes controlar todo desde una terminal, haciendo que el manejo y gestión de los contenedores y máquinas virtuales sea impecable, pudiendo usar la máquina virtual como si estuvieses dentro de la máquina en sí, abriendo diferentes interfaces de sistema.

Puedes tener un Sistema Operativo completo sin necesidad de ocupar todo el espacio en tu máquina anfitriona, por lo cual, adiós al desperdicio de almacenamiento y recursos. Esto no se podría hacer un con software de contenedores como es Docker.

### Desventajas

La única y grave desventaja que veo de este software de virtualización de contenedores y máquinas virtuales es la incapacidad de poder virtualizar sistemas Windows, por lo cual se cierra mucho el abanico de posibilidades a solo sistemas Linux.

Tienes que tener mucho conocimiento sobre redes si quieres sacarle partido en tu propia red local.

# Finalidad del proyecto

Ya que es un software que nos permite hacer muchas cosas, la finalidad de este proyecto puede servir como una guía para la gente novata que quiera aprender a hacer diferentes estructuras con contenedores pero que no tengan los conocimientos de cómo hacerlo, a su vez llevado al ámbito práctico, puede servir para montar nuestra propia estructura de servidores conectados entre sí. Un punto a tener en cuenta va a ser que mucha gente quiere tener algo con contenedores, pero el coste de ello los lleva a rendirse, gracias a LXD y que es gratuito ni siquiera se tienen que preocupar por pagar algo.

# Objetivos

Una vez puesto en marcha el proyecto vamos a ser capaces de crear nuestra propia organización de contenedores gestionados en una red creada por nosotros mismos que hace de puente entre tus clústeres por lo cual tendrán conectividad entre sí, dando lugar a la posibilidad de crear una estructura como nosotros queramos, para final de proyecto aparte de toda la información documentada de cómo se usa el software crearemos nuestra propia estructura personalizada.

# Medios necesarios

* Un ordenador con VMware Instalado
* Una máquina virtual con Debian 11 instalado
* LXD
* Software de almacenamiento Btrfs
* Cliente Spice "virt-viewer" para visualizar las máquinas virtuales.

# Planificación

* Encontrar la distribución de Linux en la cual voy a basar el proyecto (2 horas)
* Buscar información sobre LXD/LXC y comprobar que es verídica (15 horas)
* Elaborar un entorno con contenedores y máquinas virtuales haciendo uso de LXC/LXD y configuración de los servicios (16 horas)
* Documentar la información y hacer una presentación (40 horas)
* Resolución de errores o añadidos inesperados al proyecto (18 horas)

## Presupuesto del proyecto

Teniendo en cuenta que en el presupuesto del proyecto no incluyamos el precio del ordenador que hemos utilizado para montar toda la mini estructura de red, el presupuesto sería de 0 €.

Todo lo que utilizamos es software libre e incluso teniendo en cuenta que estoy utilizando VMware para realizar el proyecto, sería capaz perfectamente de hacer lo mismo con VirtualBox que es un software de virtualización gratuito. LXD sigue siendo gratuito y como solo trabaja con distribuciones Linux no hay ninguna licencia que podamos comprar.

## Esquematización del proyecto.

El proyecto se va a realizar de la siguiente manera:

* Introducción
* Investigación con capturas de pantalla de los diferentes apartados
* Elaboración práctica de algo funcional con LXD

## Asignaturas empleadas en el proyecto.

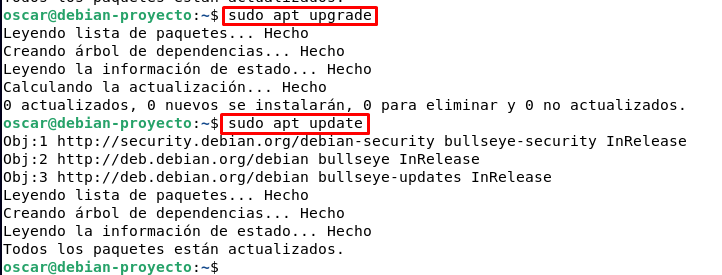
Teniendo en cuenta el estilo de proyecto que he realizado, las asignaturas que se han empleado han sido:

* ISO: instalación de sistemas, virtualización y scripting.
* ASO: instalación de sistemas en red y scripting.
* PAR: gestión de la estructura de red.
* IAW: página web en HTML y php.
* SRI: implementación de diversos servicios de red.
* GBD: gestión de la base de datos incluida en el servidor MySQL
* ASGBD: instalación del software gestor MariaDB y configuración.
* LMSGI: en la página web uso de HTML, css y JavaScript.
* SAD: implementación de seguridad mediante ACL de red.

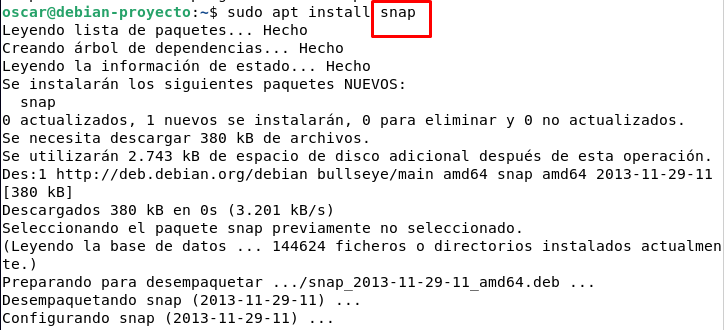
# Elaboracion del proyecto

## Instalación del software LXC y LXD

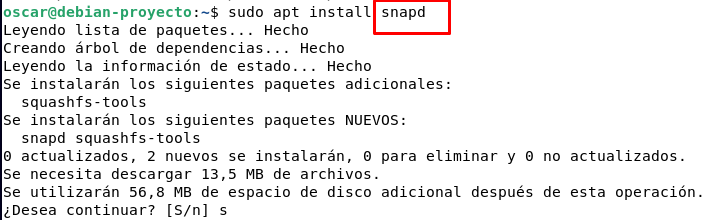
Lo primero que vamos a tener que hacer para poder instalar nuestro software va a ser actualizar los repositorios. Con las ordenes apt upgrade y apt update



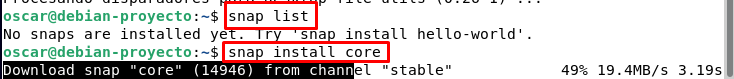
Ahora vamos a instalar lo necesario para poder instalar LXC y LXD, para ello vamos a instalar “snap”, una utilidad para empezar nuestra instalación de LXD y LXC.



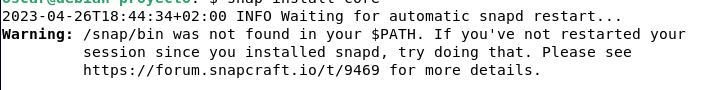
También su servicio



Si hacemos un snap list de lo que tenemos instalados, por defecto en Debian 11 no viene ningún snap instalado, así que instalaremos el core.

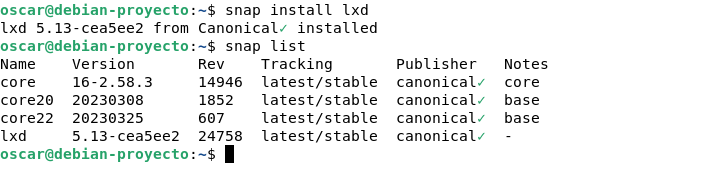


Una vez que tenemos el core instalado nos va a pedir que hagamos log out y log in para que se añada al path el /snap/bin



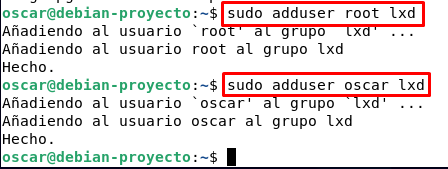
Simplemente le hacemos caso y cerramos sesión e iniciamos sesión.

Una vez instalado, usaremos snap para instalar LXD



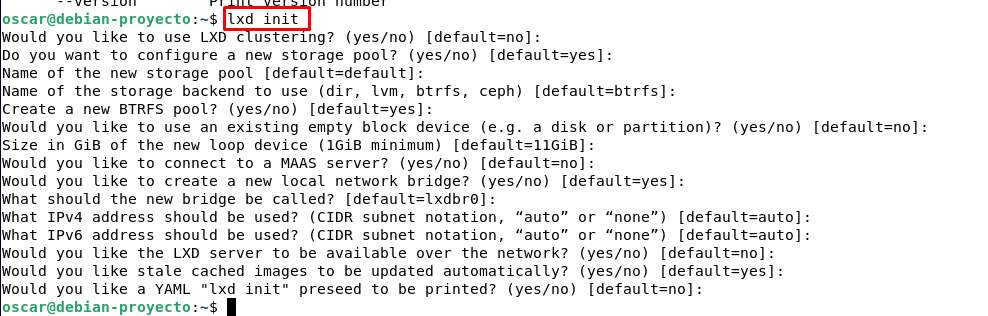
Te instala los paquetes core20 y core22 cuando se instala lxd.

Antes de empezar a trabajar con LXD nos hace falta meter el usuario al grupo LXD, meteré a los 2 tanto a mi usuario normal como a root.



Una vez que tenemos a los usuarios dentro del grupo ya podemos inicializar el software.

1 BTRFS



## Redes

LXD por defecto te crea una red a la cual irán conectadas las instancias



Su nombre es lxdbr0 y es un enlace tipo puente

También puedes crear tu propia red, que es lo que vamos a hacer nosotros en nuestro caso.

### Creación de una nueva red

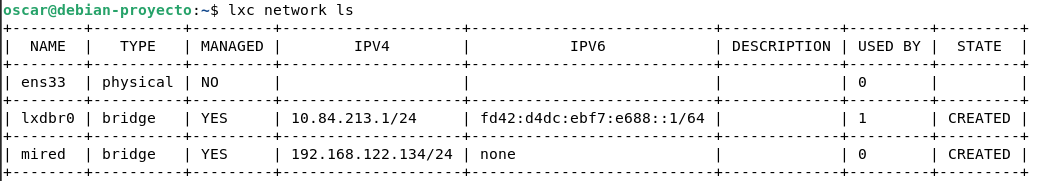
El comando para crear una nueva red es lxc network create [nombre de la red] [opciones]



El comando para crear mi red sería como el de la captura de pantalla, le asignamos una dirección, le asignamos que no tenga dirección ipv6, le ponemos el dhcp a true para que las máquinas que tengan esa red cojan la ip por dhcp y el nat para que haga la traducción de direcciones IP y podamos conectarnos desde la máquina que tiene los contenedores.

Si hacemos un listado de las redes, nos tiene que salir creada.

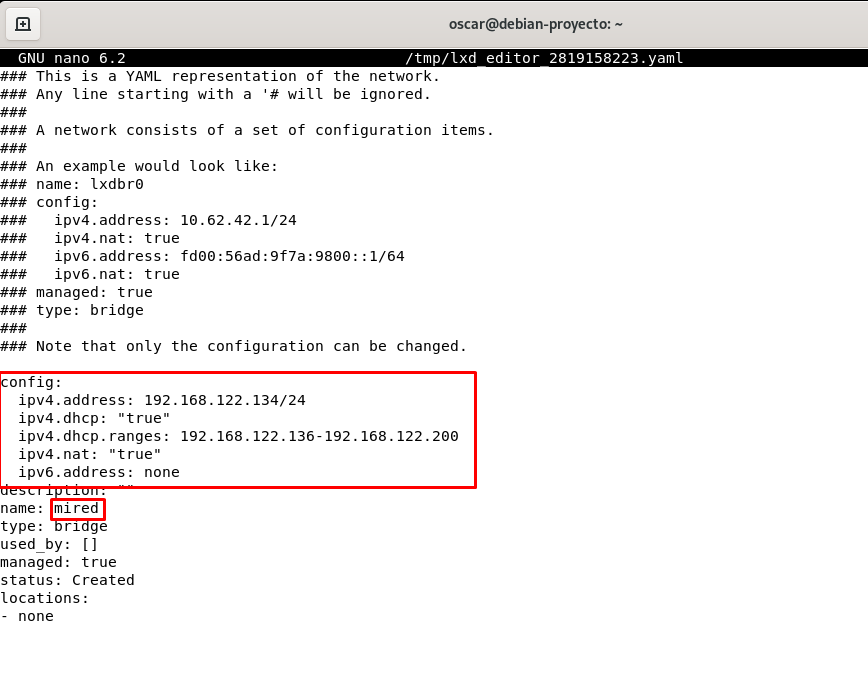
Con el comando “lxc network ls”



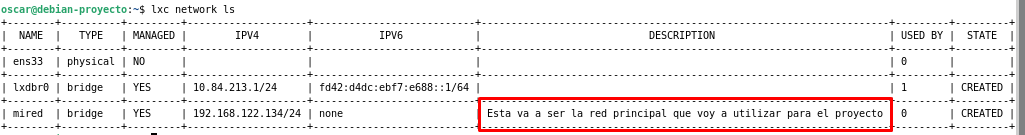
### Modificar una red ya existente

Para modificar una red usaremos el comando “lxc network edit <nombre de la red>”





Yo le voy a añadir una pequeña descripción a la red



### Usar una red en un contenedor (perfiles)

Ahora mismo la red que tenemos creada, no hay ninguna manera de aplicarla a las máquinas virtuales, para ello tenemos que hacer uso de la creación de perfiles para asignar esa red a alguna instancia.

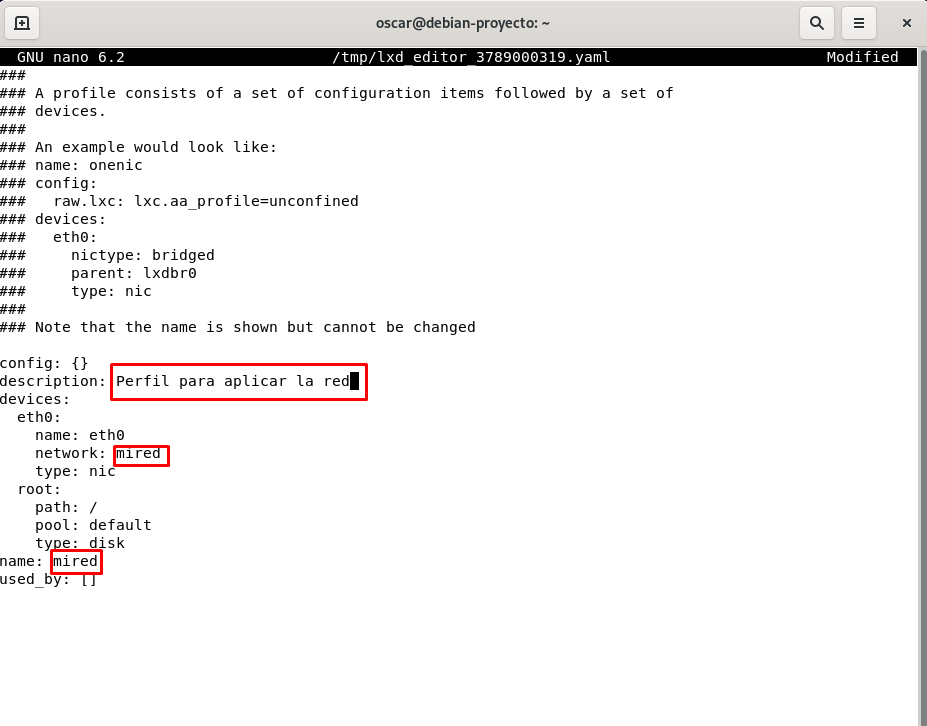
Para crear un perfil podemos usar dos de las siguientes formas:

Copiándolo del default y modificándolo(recomendable)

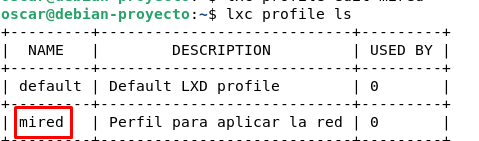
* El comando para copiar es “lxc profile copy <perfil a copiar> <nombre del perfil nuevo>”
* Y para editarlo es “lxc profile edit <nombre del perfil a editar>”



Y lo editamos poniendo el nombre de nuestra red y nuestra descripción personalizada.



De esta manera tenemos un perfil listo para aplicar.



Y la segunda manera que sería creándolo nosotros mismos a comando.

* Con el comando “lxc profile create <nombre del perfil a crear>” creamos el perfil

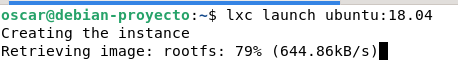


* Y lo mismo lo editamos con “lxc profile edit <nombre del perfil>”
* Los perfiles tienen contenido y se pueden ver con el comando “lxc profile show <nombre del perfil a ver>”



Estos perfiles ahora que están creados, son aplicables a las instancias para que sean capaces de pertenecer a la red que nosotros hemos creado, ya que, si no, no hay otra manera de aplicarlos, para ello primero crearemos una instancia rápida de prueba que ni siquiera usaremos.

* Con el comando “lxc launch <repositorio:nombre de la imagen>” seremos capaces de hacer una instancia con una imagen de ese sistema operativo descargado directamente de los repositorios de Linux, la primera vez que se hace una instancia tarda mucho porque tiene que descargar la imagen, pero una vez descargada, hacer una instancia con ese sistema operativo es cuestión de 3 o 4 segundos.

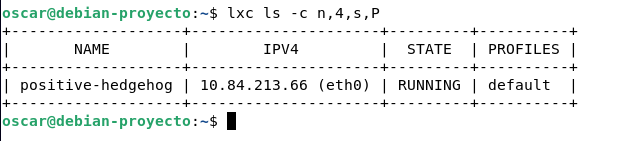


Una vez acabado si nosotros no le hemos dado un nombre a la instancia, el propio lxd creará un nombre aleatorio para ella.



Ahora tenemos la instancia con nombre positive-hedgehog con el perfil por defecto.

* Vamos a verlo con el comando “lxc ls -c n,4,s,P”

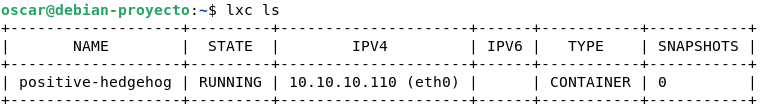


Como vemos el perfil que tiene es el de por defecto, ahora vamos a cambiarlo para poder asignarle la dirección ip que nosotros queremos que se asigne.

* Usaremos el comando “lxc profile add <nombre de la instancia> <nombre de la red>”



Ahora podemos ver que la instancia ha cogido la configuración de red del perfil.

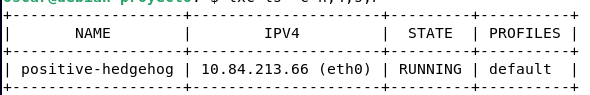


Para borrar un perfil de un contenedor es tan sencillo como usar el siguiente comando.

* “lxc profile remove <nombre del contenedor> <nombre del perfil a borrar>

De la siguiente manera se borrará el perfil de ese contenedor y por lo cual perderá la configuración de red que nosotros habíamos asignado con ese perfil.





Como vemos se ha borrado el perfil de red que nosotros habíamos aplicado en nuestro contenedor.

### ACL

El sistema LXD tiene la opción de configurar normas acl para nuestras redes.

La ACL o “Access Control List” en español listas de control de acceso es un método de seguridad en formato reglas que regulan el tráfico de red entre las instancias en referente a LXD y el acceso hacia y desde otras redes.

Lo primero que vamos a probar es a crear nuestra propia ACL, hay un comando en específico para poder hacerlo y es el siguiente:

* “lxc network acl create [nombre de la acl] [opciones de la acl]”

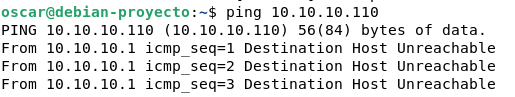


Ahora vamos a añadir alguna norma de ingreso a esa acl para que las conexiones no puedan pasar y solo admita algunas.

* “lxc network acl rule add <nombre de la acl> <destino> <propiedad> <propiedades>

Ese sería un comando para rechazar conexiones.





Como vemos está rechazando la conexión con el host ya que nosotros le hemos quitado en la ACL todas las conexiones

Ahora vamos a eliminar la acl que hemos creado, para ello se usará el siguiente comando:

* “lxc network acl rule remove <nombre de la acl> <propiedad> <propiedades>” esta la usaríamos si queremos borrar una regla
* “lxc network acl delete <nombre de la acl> esta es la que usaríamos para borrar una acl completamente



Por lo cual se quedaría eliminada completamente.

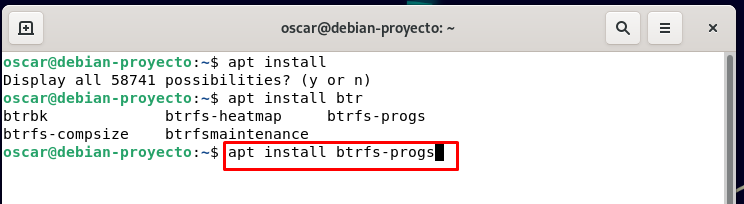
## Almacenamiento

LXD funciona almacenando todos sus datos en una especie de grupos de almacenamiento, que se dividen en volúmenes de almacenamiento que difieren dependiendo de su tipo de contenido. (como imágenes o instancias)

* Grupos de almacenamiento que existen:
  + Directorio – dir
  + Btrfs – Btrfs(B-tree File System) sistema de fichero copy-on-write(CoW))
  + Lvm- lvm
  + Zfs- zfs

BTRFS, es un sistema de archivos que funciona de manera local y almacena los datos en bloques diferentes después de modificarlos, en lugar de sobrescribir constantemente los datos que ya existen, esto nos da la ventaja de tener una mayor fortaleza contra la corrupción de datos en nuestras instancias o máquinas virtuales.

Como en mi caso voy a usar BTRFS lo primero que tenemos que hacer es instalar la utilidad.

* Lo haremos con el comando de instalación y el paquete que vamos a instalar es btrfs-prog.
  + 
* Tipos de contenido: Existen 2 tipos de contenido que se pueden trabajar con BTRFS:
  + Filesystem: Este tipo de contenido se puede adjuntar tanto a contenedores como a las máquinas virtuales, y son un tipo que permite compartirse entre las instancias. Se usa para contenedores e imágenes de contenedores.
  + Block: Para crearlo es necesario añadir el argumento –type=block. No deben compartirse entre las instancias que tengamos instaladas, ya que eso puede ayudar al daño de los datos que tengamos, como solo se pueden adjuntar a máquinas virtuales solo se puede usar en las máquinas virtuales.
  + Para crearlo podemos usar el siguiente comando:
    - Lxc storage volume create [nombre\_pool] [nombre\_volumen] –type=block [opciones de configuración varias]



Cuando iniciamos LXD después de la instalación, nosotros podemos elegir qué tipo de ficheros 1 BTRFS

Configuramos varios parámetros de la instalación y uno de esos parámetros es el método de almacenamiento, que en nuestro caso usamos siempre el btrfs, aunque poner esta opción por defecto no nos quita la posibilidad de poder poner otro tipo de método de almacenamiento en nuestras instancias, ya que al crear una instancia existen varias formas de asignarle el método de almacenamiento que se le va a proveer con la opción “—storage”

* Se lo podemos aplicar a las instancias de las siguientes maneras:
  + Con un comando directamente a la instancia:
    - “lxc launch [nombre\_imagen][nombre\_instancia] –storage [nombre\_pool]”, ese comando es para poder crear una instancia en la piscina que tengamos creada previamente, si queremos el volumen es el siguiente:
  + “lxc storage volumen attach [nombre\_pool] [nombre\_volumen] [nombre de la instancia] [ruta donde la vas a poner en la máquina]
  + 
  + 

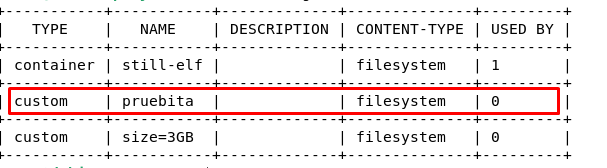
Realmente el pool de almacenamiento son como grupos a los que se le aplica un sistema de archivos que nosotros le hayamos puesto, para ello podemos incluso crear nuevos grupos de almacenamiento.

* Con el comando “lxc storage create [nombre\_nuevapool] btrfs” ponemos btrfs porque nosotros estamos usando ese driver.
  + 
  + 

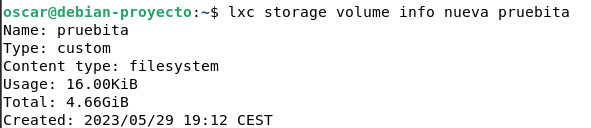
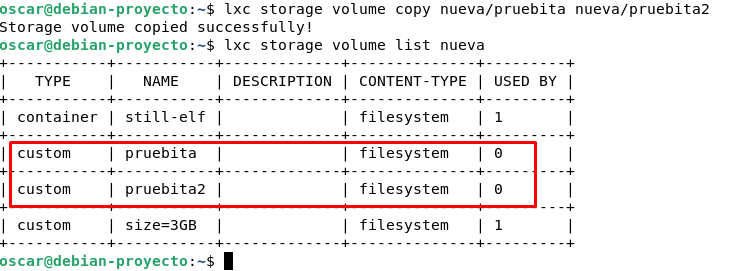
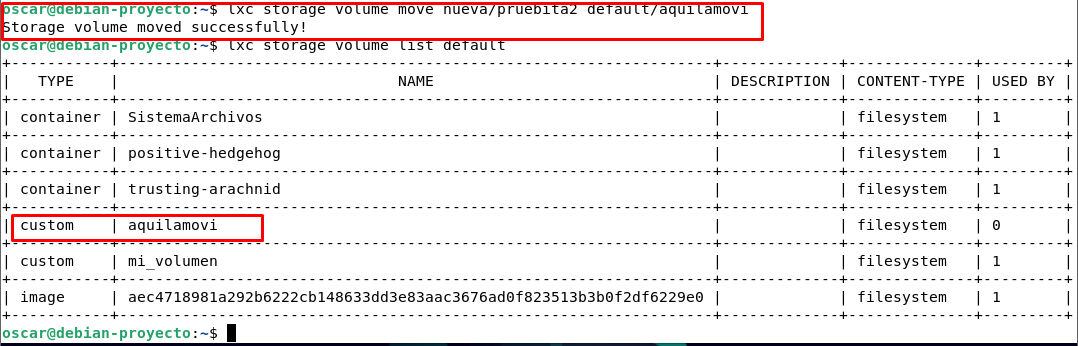
Ahora podemos mover de un grupo de almacenamiento a otro una instancia con un comando, será el siguiente.

* “lxc move [nombre\_instancia] –storage [pool\_destino]”
  + 
  + Como podemos ver si hacemos un listado de almacenamiento vemos que nueva está siendo usada por 1 dispositivo

También podemos crear un nuevo volumen de almacenamiento que pertenezca al grupo de almacenamiento que hemos creado nuevo

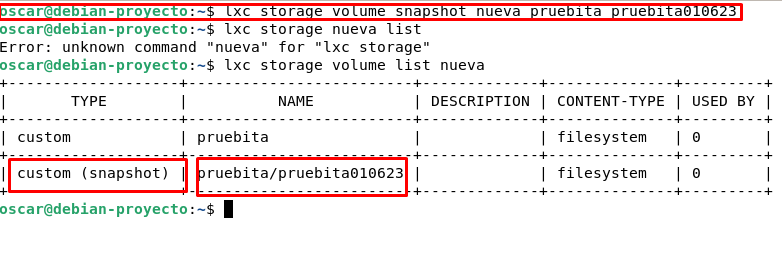
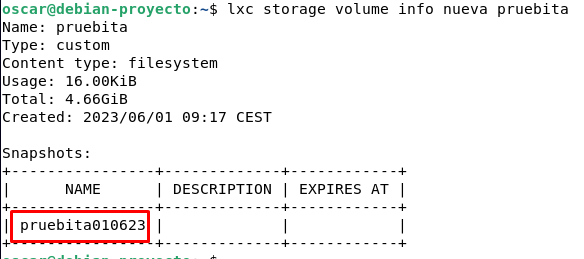
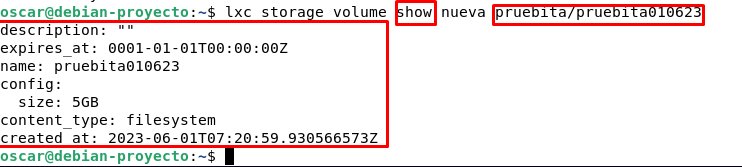
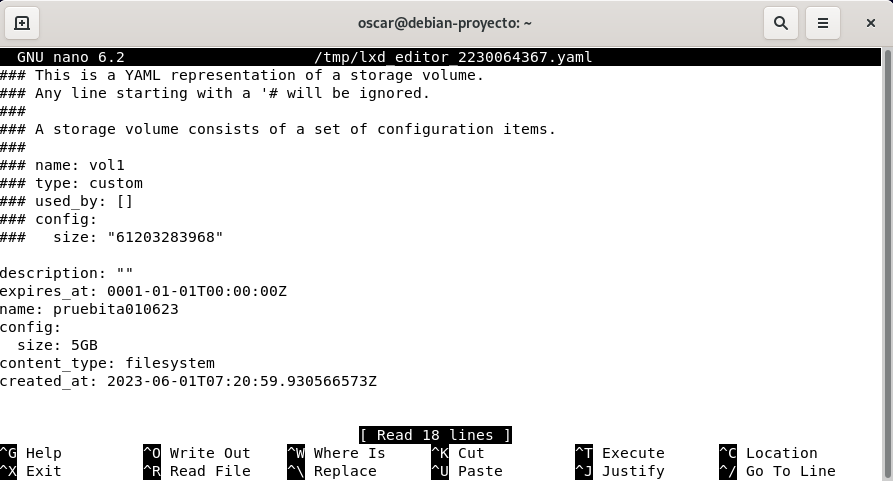
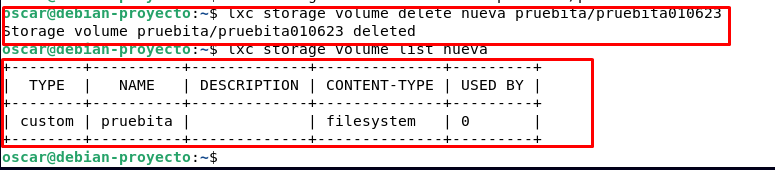
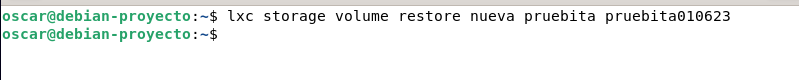
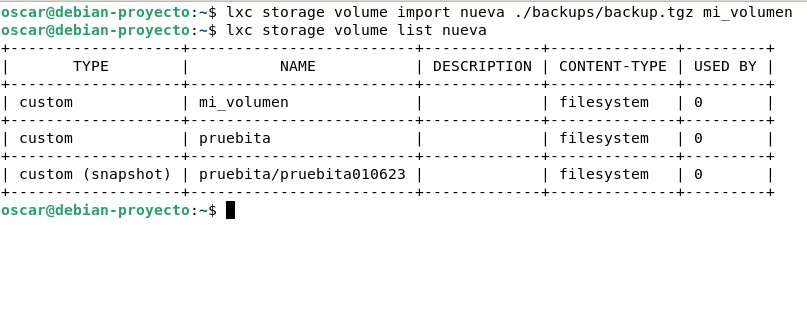
* “lxc storage volume create [nombre\_pool] [nombre\_volumen] [opciones configuración]”
  + 
  + 

Se pueden usar los volúmenes de almacenamiento para backups específicos del sistema con los siguientes comandos:

* Para los archivos comprimidos de copias de seguridad se usa el comando: “lxc config set storage.backups\_volume [nombre del pool]/[nombre del volumen]”
  + 
* Para los tarballs de las imágenes no es muy diferente “lxc config set storage.images\_volume [nombre del pool]/[nombre del volumen]”
  + 
* Incluso se puede cambiar el tamaño de los volúmenes, con el comando “lxc storage volume set [nombre\_pool]/[nombre\_volumen] size [tamaño que le quieras poner]
  + 
  + 
* Puedes copiar los volúmenes también con el comando lxc storage volumen copy [pool\_origen]/[volumen origen] [pool\_destino]/[volumen destino]”
  + 
* También puedes renombrarlas o moverlas de sitio con el comando “lxc storage volumen move [pool\_origen]/[volumen origen] [pool\_destino]/[volumen destino]”
  + 

Los volúmenes se pueden respaldar con copias de seguridad y lxd trae la utilidad necesaria para poder realizar esta tarea.

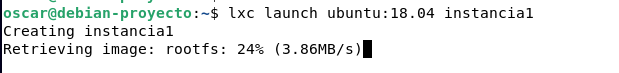
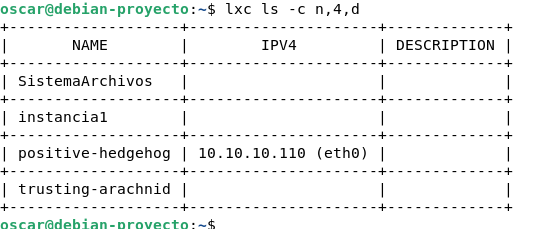
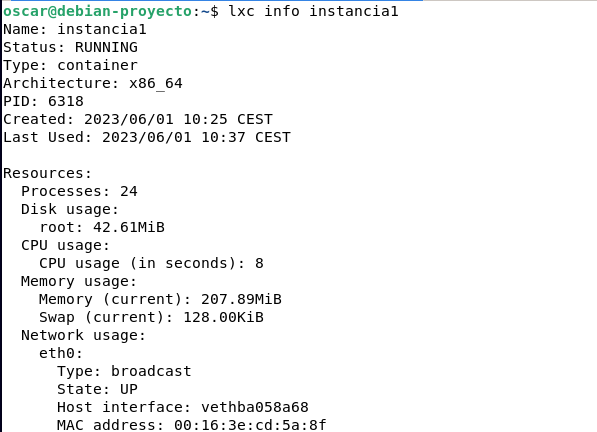
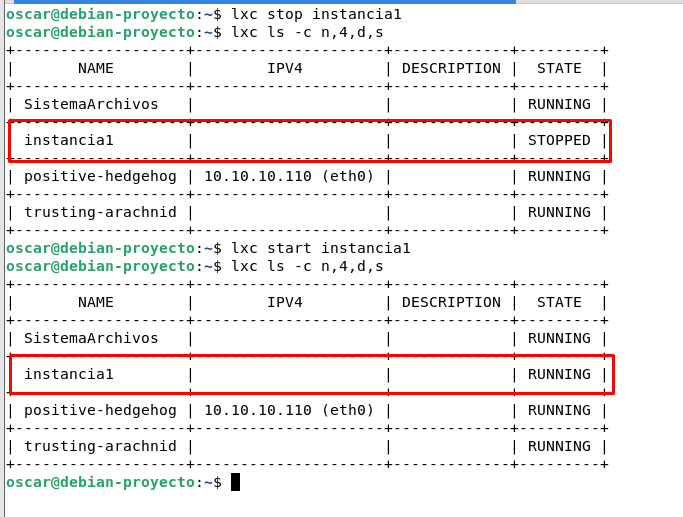
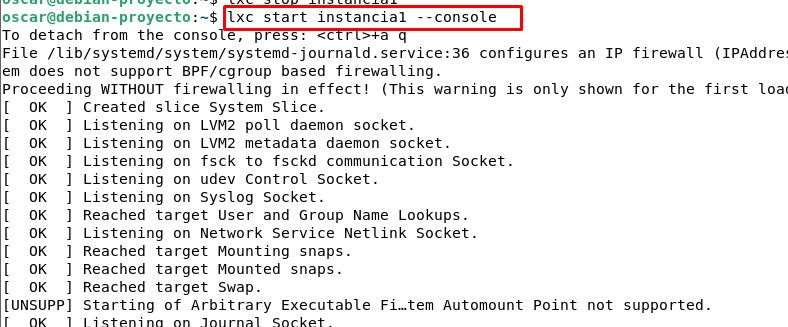
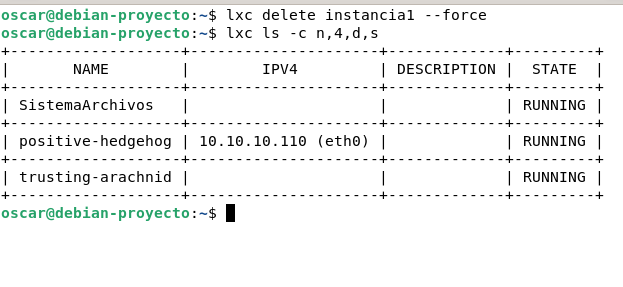
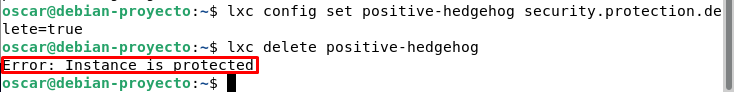
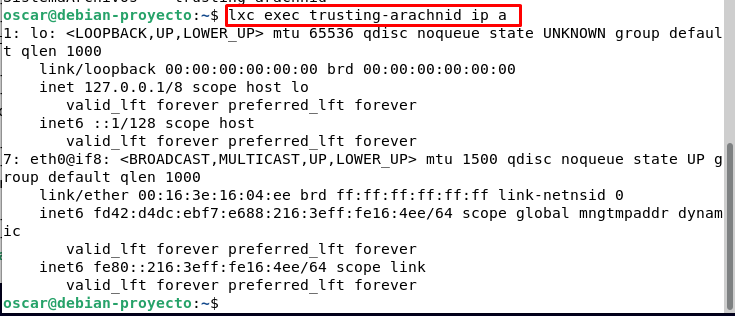
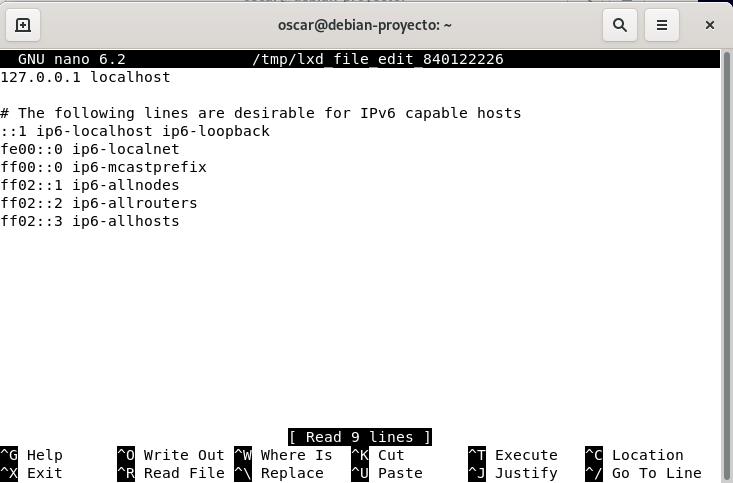
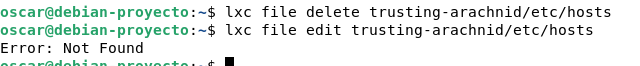
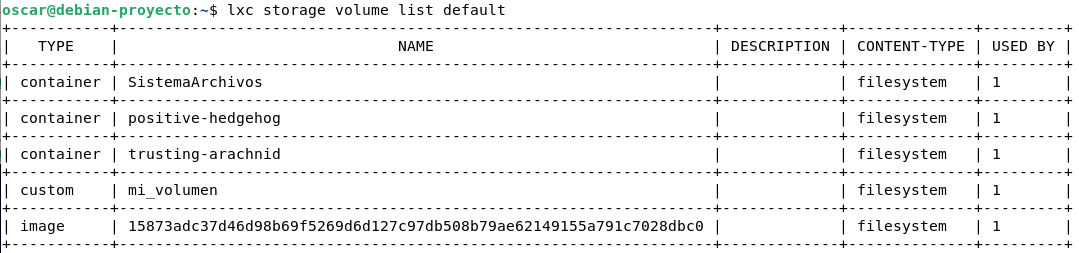
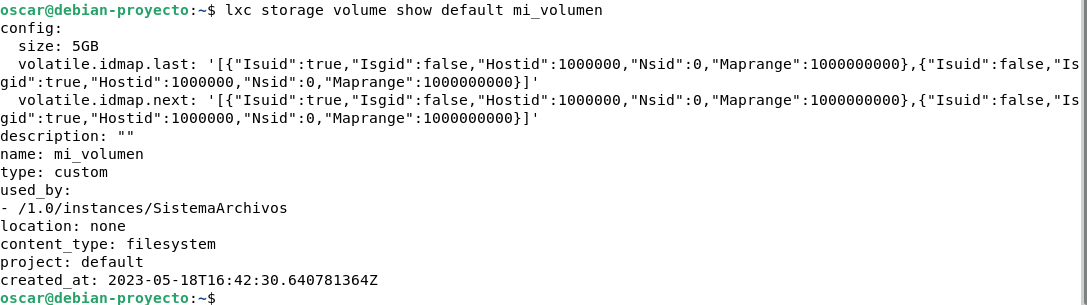
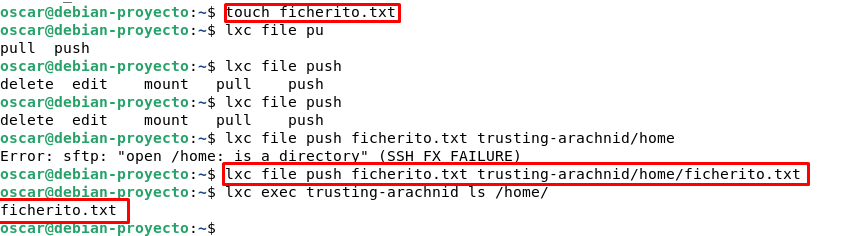
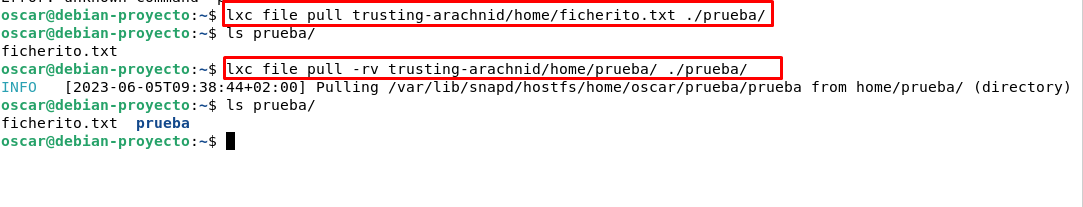
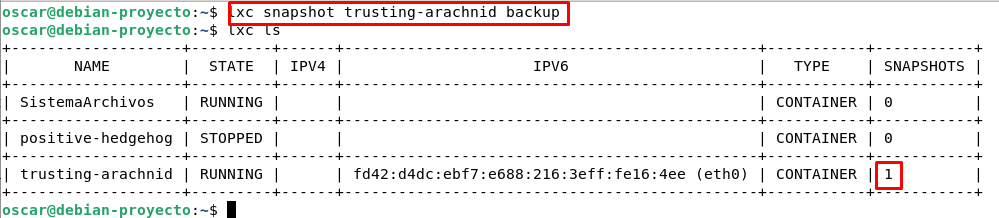
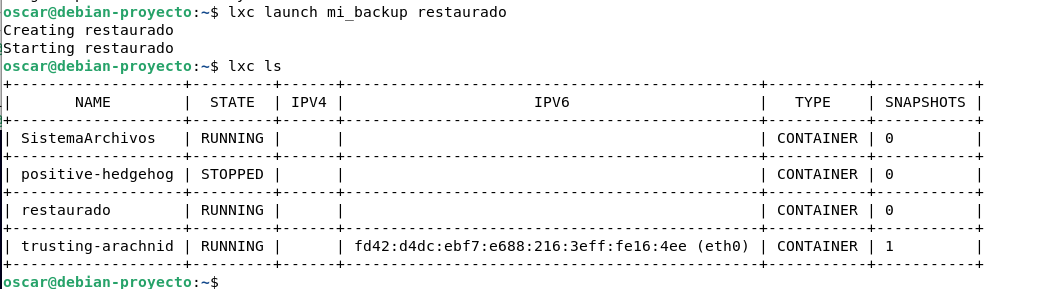
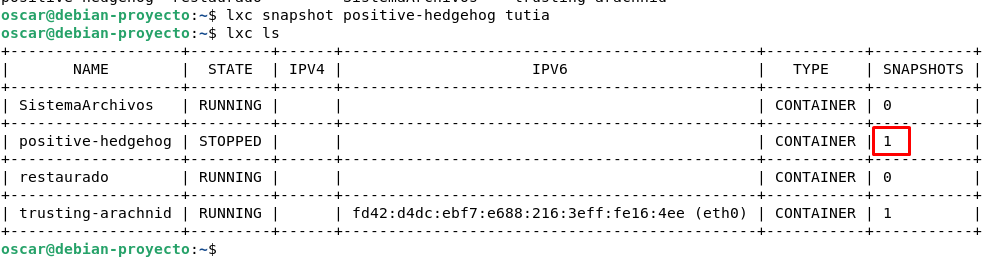
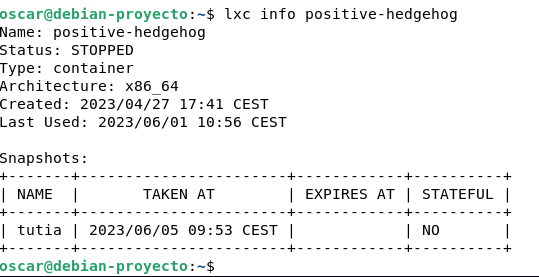
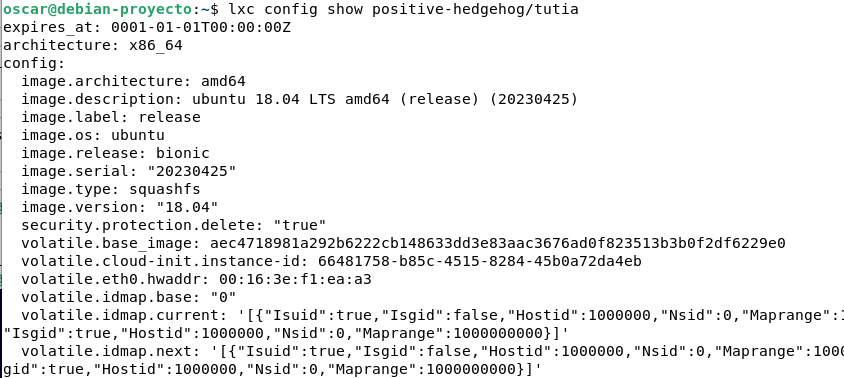
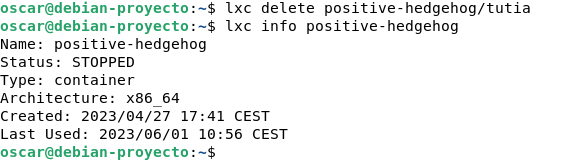
Tenemos dos formas de hacerlo:

* Snapshots de los volúmenes tiene el inconveniente que se guarda en el mismo pool de almacenamiento que el volumen original por lo cual es un poco menos seguro:
  + Usamos el comando “lxc storage volumen snapshot [nombre\_pool] [nombre\_volumen] [nombre\_snapshot]”: 
  + Puedes mostrar las instantáneas que tiene un volumen con el comando “lxc storage volume info [nombre\_pool] [nombre\_volumen]”: 
  + Puedes ver la información de esa snapshot con el mismo comando agregando /[nombre\_snapshot] despues del volumen y cambiando info por show: 
  + Tambien tenemos que tener en cuenta que se pueden editar esas snapshots con el comando “lxc storage volume edit [nombre\_pool] [nombre\_volumen]/[nombre\_snapshot]”: 
  + Tambien podemos borrar una snapshot con el comando “lxc storage volume delete [nombre\_pool] [nombre\_volumen]/[nombre\_snapshot]: 
  + Podemos restaurar una snapshot con el comando “lxc storage volume restore [nombre\_ pool] [nombre\_volumen] [nombre\_snapshot]”: no se ve realmente que se haya restaurado porque no hay forma de ver los cambios sin meternos en la máquina cambiar algo y restaurar pero ciertamente se ha restaurado.
* Mediante un export del fichero del volumen:
  + Para exportarlo simplemente usamos el comando “lxc storage volume export [nombre\_pool] [nombre\_volumen] [ruta de exportacion]”: 
  + Ahora podemos restaurarla importando ese mismo fichero con el comando “lxc storage volume import [nombre\_pool] [ruta al fichero] [nombre\_volumen]”: 

Esto es lo más esencial que tenemos que saber del almacenamiento de LXD.

## Configuración de instancias, opciones diversas.

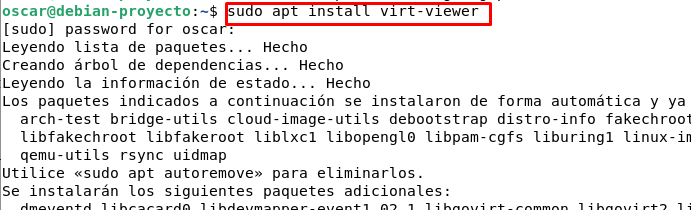
A pesar de ser lo más básico de LXD he decidido dejarlo para casi lo último ya que así lo podremos ver más en profundidad.

* Creamos un contenedor con el comando “lxc launch [repositorio]:[distribución] [nombre de la máquina virtual] <opciones>”:
* Para listar los contenedores puedes usar “lxc list <opciones>” o “lxc ls <opciones>”:la opción -c especifica columnas y tú le dices que columnas le vas a asignar para que te haga una lista, por ejemplo, el de arriba es n de name, 4 de ipv4 y d de description.
* Puedes acceder a un contenedor con el comando “lxc exec [nombre\_instancia] bash”: 
* Puedes mostrar toda la información de una instancia con el comando “lxc info [nombre\_instancia]”: 
* Podemos parar e iniciar una instancia con los comandos “lxc [start/stop] [nombre\_instancia]”: 
* Podemos conectarnos a la terminal cuando la iniciamos con el mismo comando con el cual iniciamos la instancia pero agregandole –console: 
* Podemos eliminar una instancia una vez que esté parada con el comando “lxc delete [nombre\_instancia]” si no está parada puedes ponerle la flag “—force””: 
* Se puede habilitar la seguridad para la eliminación de instancias con el comando “lxc config set [instancia] security.protection.delete=true”: 
* Se puede ejecutar un comando sin tener que entrar a la instancia como bash con el comando “lxc exec [instancia] [comando a ejecutar]”: 
* Podemos editar ficheros de las instancias desde nuestra máquina anfitriona, con el comando “lxc file edit lo [nombre\_instancia]/[ruta\_fichero]” lo editará con el editor de ficheros predeterminado de tu máquina linux: 
* Tambien al igual que editar, también se pueden eliminar archivos desde la máquina anfitriona a las instancias con el comando “lxc file delete [nombre\_maquina]/[ruta\_fichero]”: 
* Para ver cualquier elemento referenciado al almacenamiento se usan los siguientes:
  + Listar volúmenes: “Lxc storage volume list [nombre\_pool]” 
  + Para ver un volumen en concreto con más detalle: “lxc storage volume show [pool] [volumen]” 
* Podemos también transferir archivos desde máquina host a una instancia con el comando “lxc file push [ruta\_fichero\_local] [nombre\_instancia]/[ruta\_fichero\_contenedor]”: 
* Podemos hacer lo mismo con directorios pero tenemos que agregarle el valor “--recursive/-r” para que funcione como un directorio. 
* Podemos tambien descargar ficheros y directorios con el comando “lxc file pull [contenedor]/[ruta\_fichero\_origen] [ruta\_fichero\_destino]”: 
* Se pueden hacer backups de los contenedores lxd, el procedimiento para hacer un backup completo y exportarlo como imagen tarball es el siguiente:
  + Creamos una snapshot: 
  + Ahora publicamos la imagen que acabamos de crear para exportarlo: 
  + Una vez publicado vamos a exportar la imagen al tarball para tenerla disponible siempre que queramos restaurar el backup: 
  + Y ahora podemos lanzar la instancia backup como si fuera una normal y va a ser una instancia con las mismas características que de la que hemos hecho un backup, esto puede ser útil para hacer una instancia que queramos repetir todo el rato: 
* Como hemos visto antes se pueden crear instantáneas de las instancias en el momento justo, estas se pueden hacer con el comando “lxc snapshot [instancia] [nombre\_snapshot]”: 
* Podemos listar las instantáneas de cada instancia con el comando “lxc info [instancia]”: Estas mismas instantáneas las podemos editar con el comando “lxc config edit [instancia]/[instantánea]”: 
* Tambien podemos ver la configuración de cierta instantánea con el comando “lxc config show [instancia]/[instantánea]”: A su vez tambien las podemos borrar con el comando “lxc delete [instancia]/[instantánea]”: 
* Podemos programar para que haga instantáneas diariamente por ejemplo con el comando lxc config set [instancia] snapshots.schedule @daily”: 
* Para restaurar una snapshot el comando es “lxc restore [instancia] [snapshot]”: 

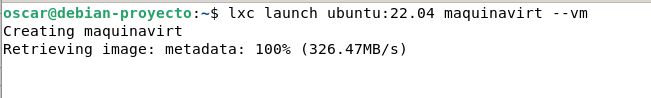
## Máquinas virtuales

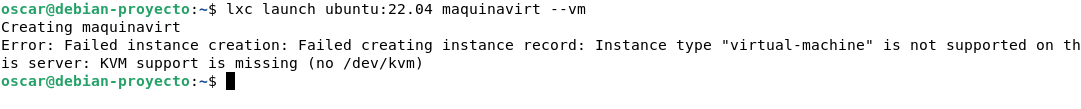
Una máquina virtual es distinta de un contenedor, digamos que el contenedor consume los recursos necesarios para funcionar en base a tu sistema host y una máquina virtual consume unos recursos fijos prefijados por ti en la configuración, una ventaja de las máquinas virtuales es que podemos visualizarlas con un cliente Spice desde lxd de una manera bastante sencilla.

Hay diferentes clientes Spice en Linux, los más conocidos son virt-viewer y spice-gtk-client, yo voy a usar virt-viewer porque es simplemente más fácil y no tienes que configurar parámetros extra, lo primero que vamos a hacer es instalarlo.

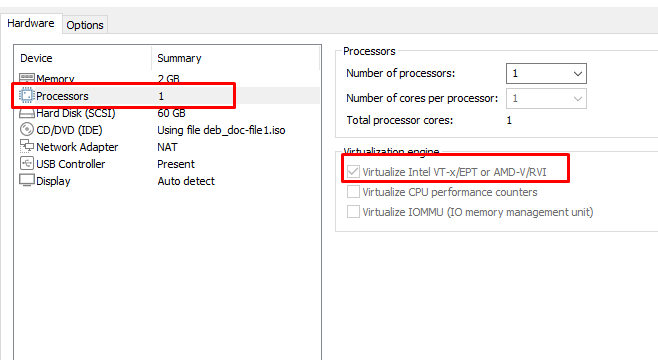


Una vez instalado simplemente vamos a instalar un contenedor de carácter máquina virtual porque si no, te buscará la versión de contenedor, simplemente el comando es “lxc launch [repositorio]:[imagen] --vm”:

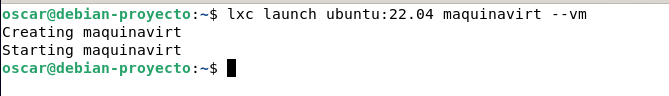




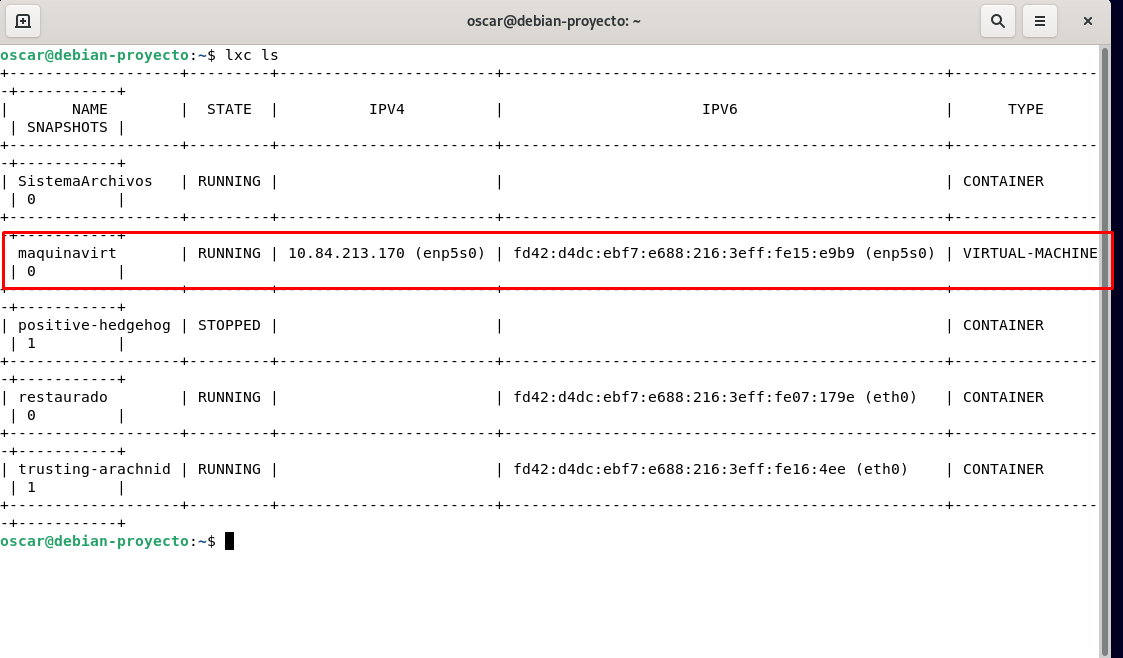
Me dio un error de que no es compatible, buscando información encontré que era problema de que no está habilitada la virtualización anidada en la máquina que he creado de VMware, para habilitarlo marcamos esta pestaña en la configuración de la máquina virtual.



Una vez eso está hecho ya te deja crear una VM en lxd.

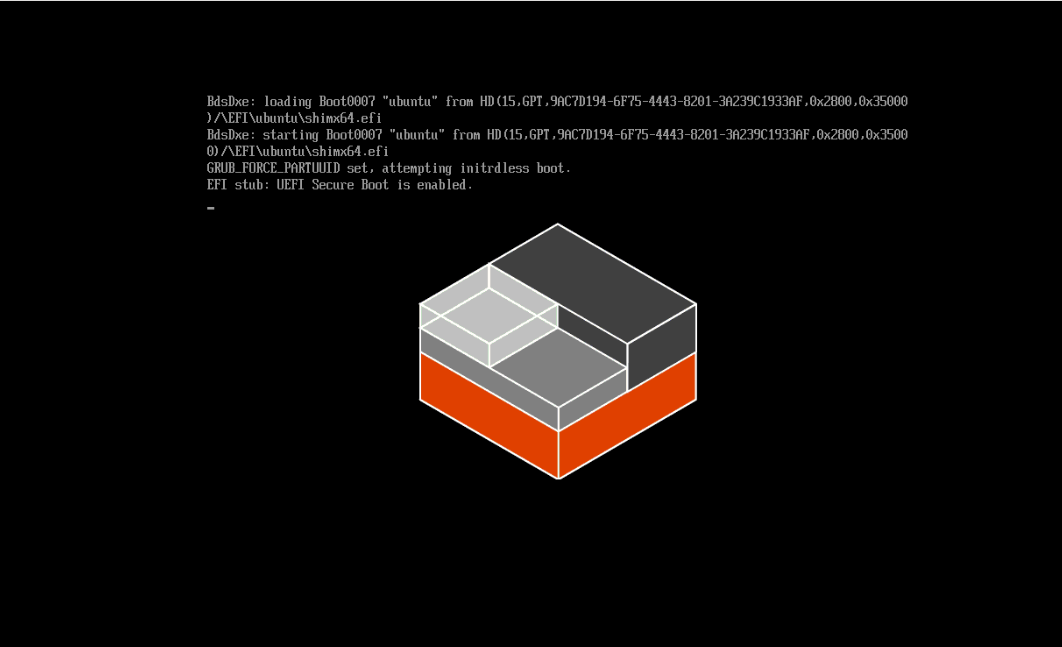


Ahora si vemos el tipo de sistema que es saldrá que es una máquina virtual.



Ahora por último vamos a meternos en ella en entorno gráfico.





He elegido una distribución mala, si hubiera escogido una de escritorio hubiera podido manejarla en gráfico.

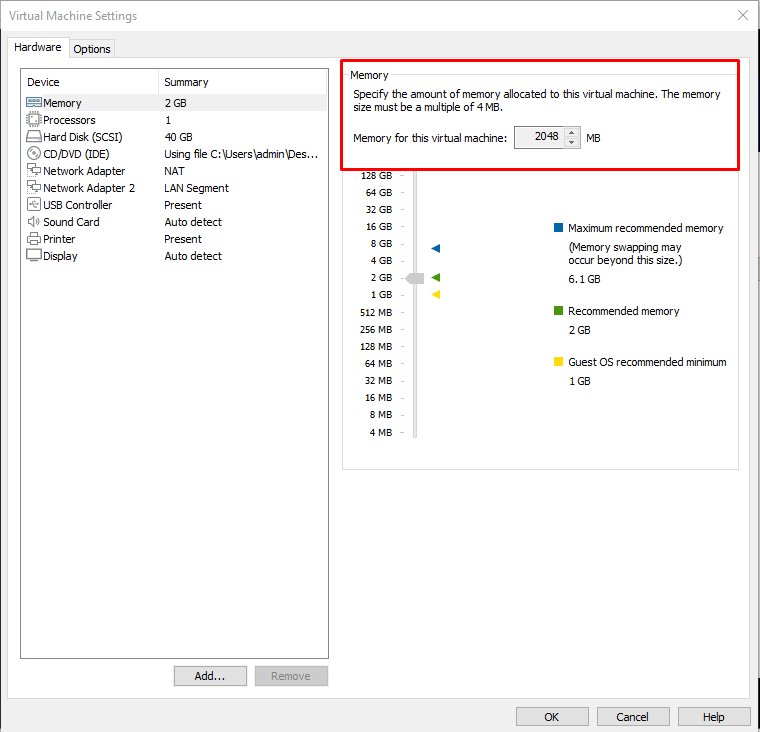
## Demostración de mini sistema de red

Ahora que ya está todo explicado voy a montar una mini estructura sencilla utilizando algunos de los conceptos que he estado explicando anteriormente para demostrar que realmente el software funciona y que es una gran alternativa a Docker o a cualquier software de virtualización

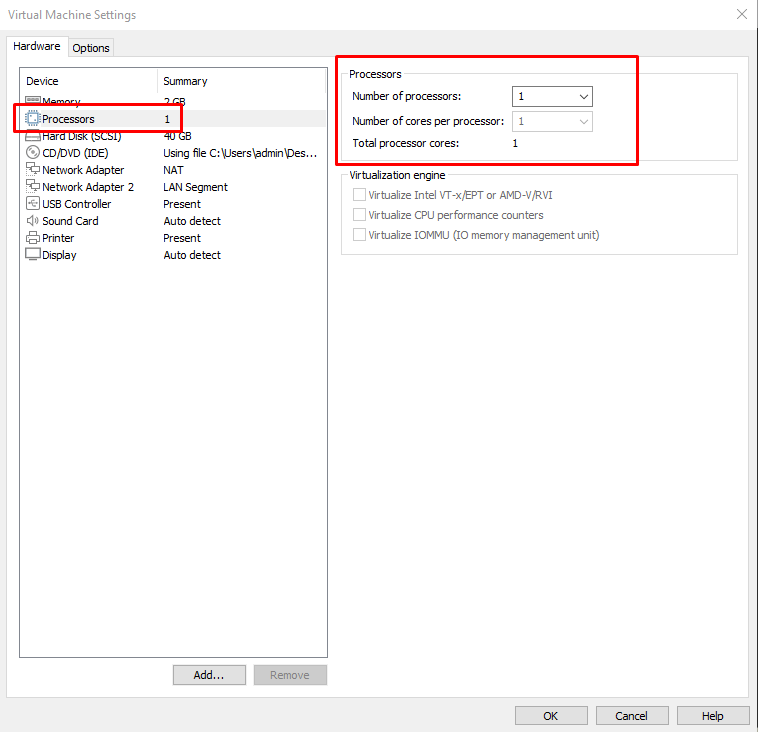
### Instalación de la máquina virtual host

Primero vamos a crear una máquina virtual con la configuración que nosotros veamos óptima, las características hardware que van a tener mi máquina van a ser las siguientes.

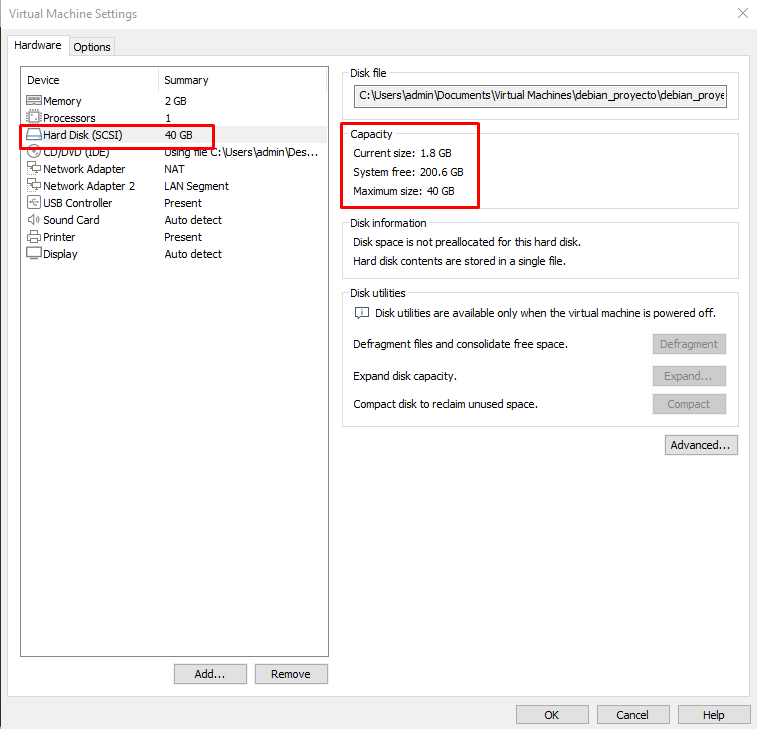
He de mencionar que he cogido prestado un portátil de la junta y por eso el Setup va a ser un poco más ahorrativo



La cantidad de memoria que le vamos a asignar al equipo host de LXD van a ser de 2 GB ya que lo considero suficiente para que funcionen todos los contenedores y máquinas virtuales que va a contener.

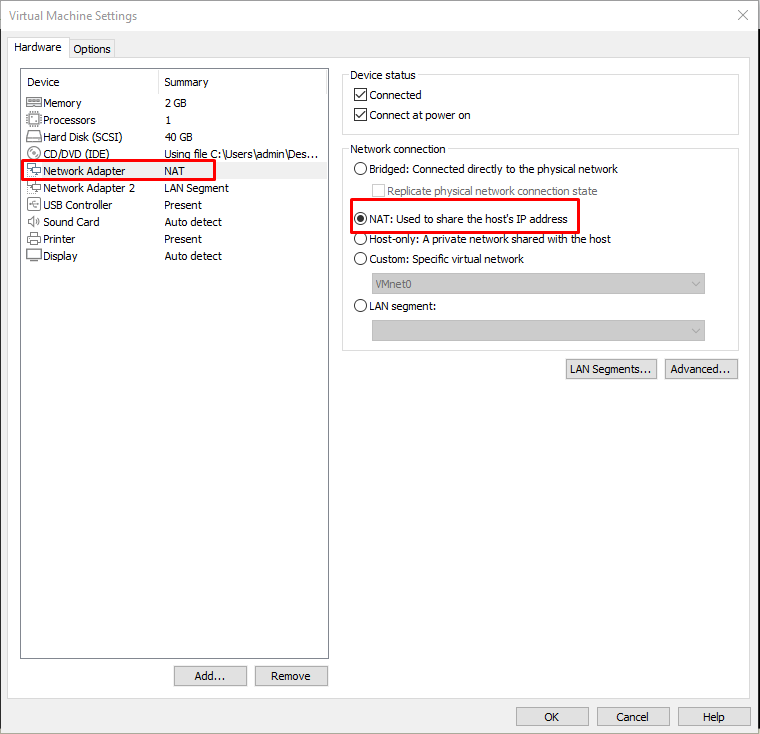


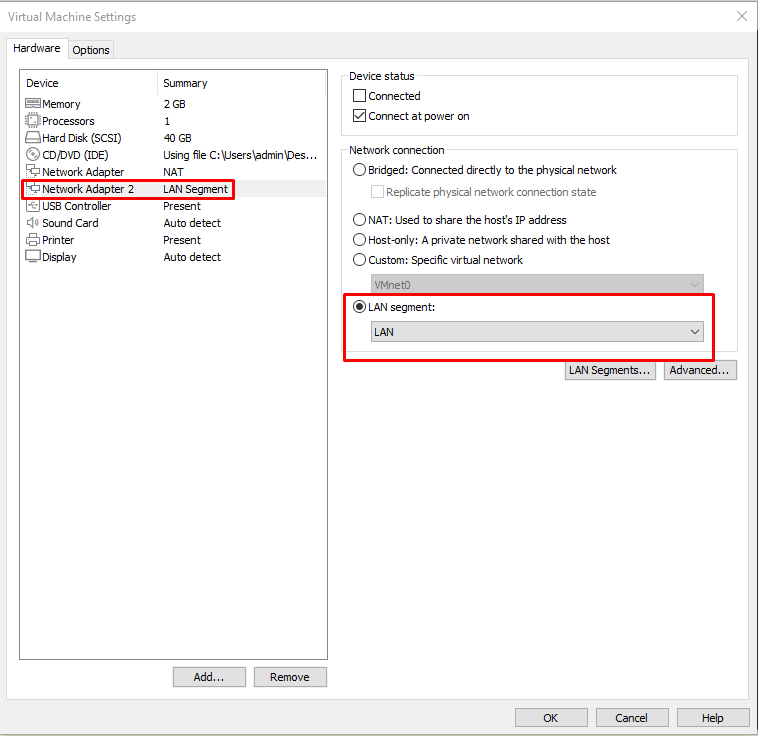
1 procesador para esta máquina será más que suficiente teniendo en cuenta que el sistema operativo que vamos a usar será Debian con un entorno gráfico XFCE



Nosotros usaremos un disco duro de 40GB de almacenamiento ya que al tener muchas imágenes de contenedores y máquinas virtuales vamos a necesitar más almacenamiento que el que te pedía de recomendado, que en su momento eran 20GB

Teniendo en cuenta que una vez que instalemos el Debian no volveremos a usar esta característica hardware, lo dejaremos deshabilitado



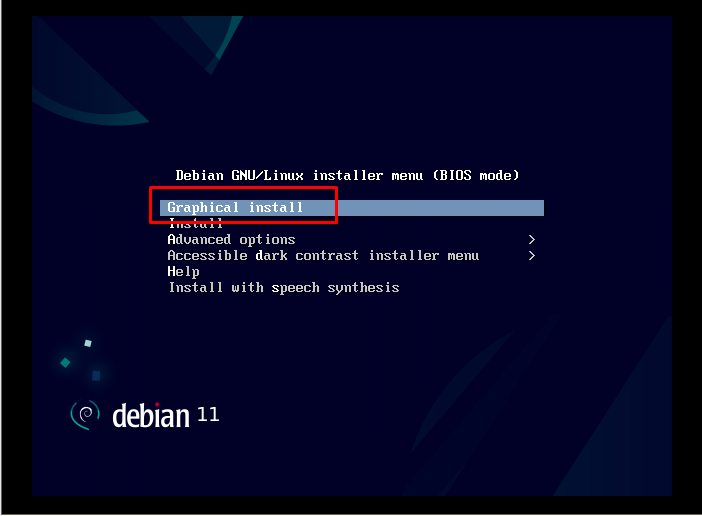


Nosotros tendremos 2 adaptadores de red, uno en NAT para poder tener conectividad a internet para descargar los paquetes etc y otro en LAN Segment para realizar la estructura de red, una vez que tengamos los paquetes completamente instalados haré una migración de la estructura de red a este LAN segment.

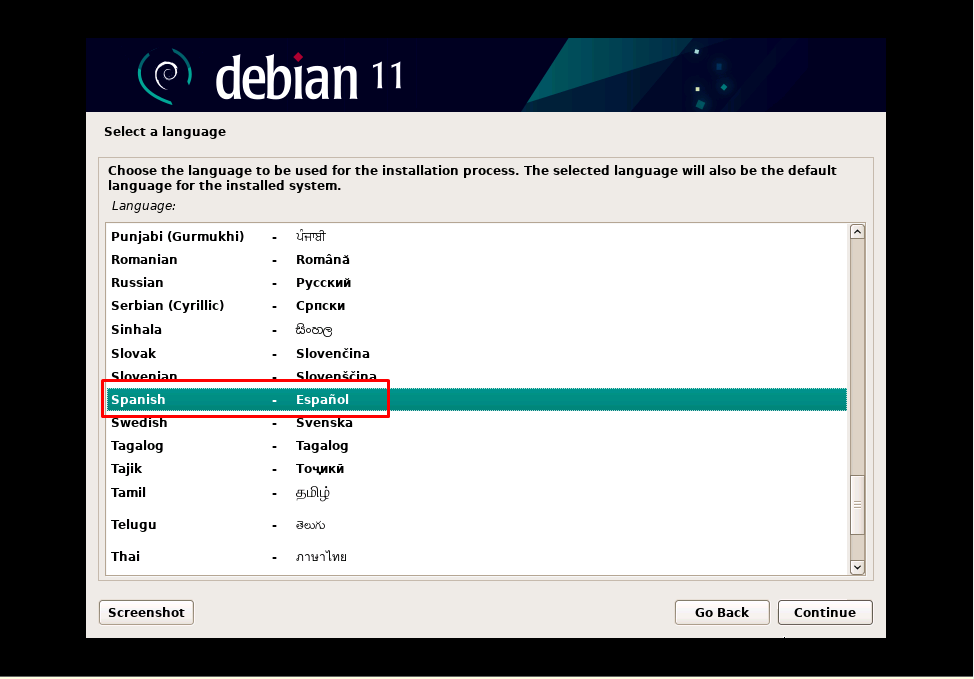
El resto de opciones de hardware en la máquina virtual lo dejaremos por defecto ya que se ajustan a las características hardware de tu maquina real.

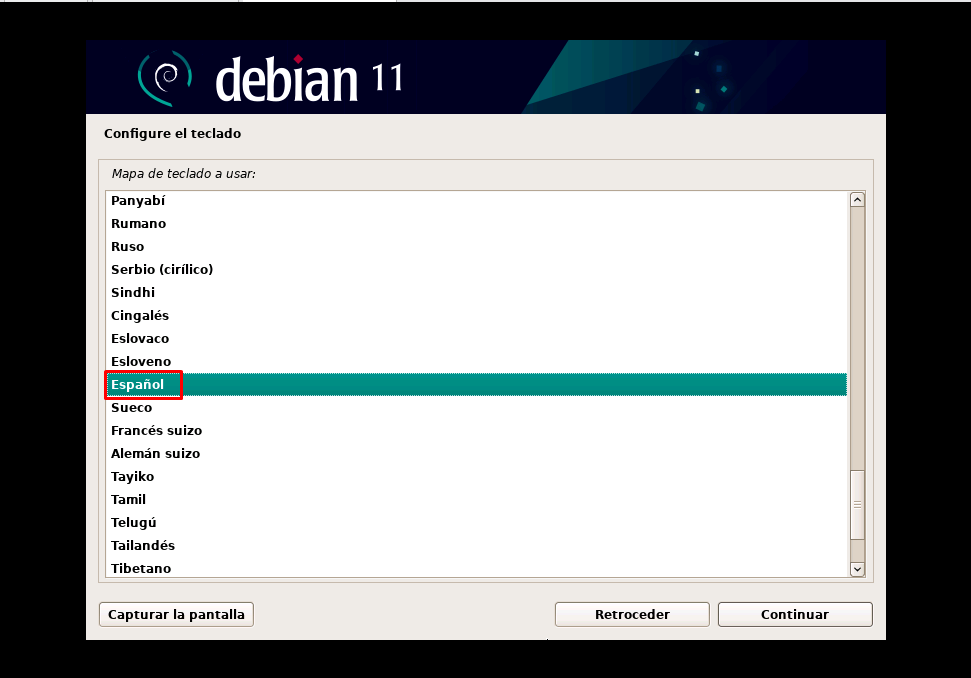
### Instalación del sistema operativo

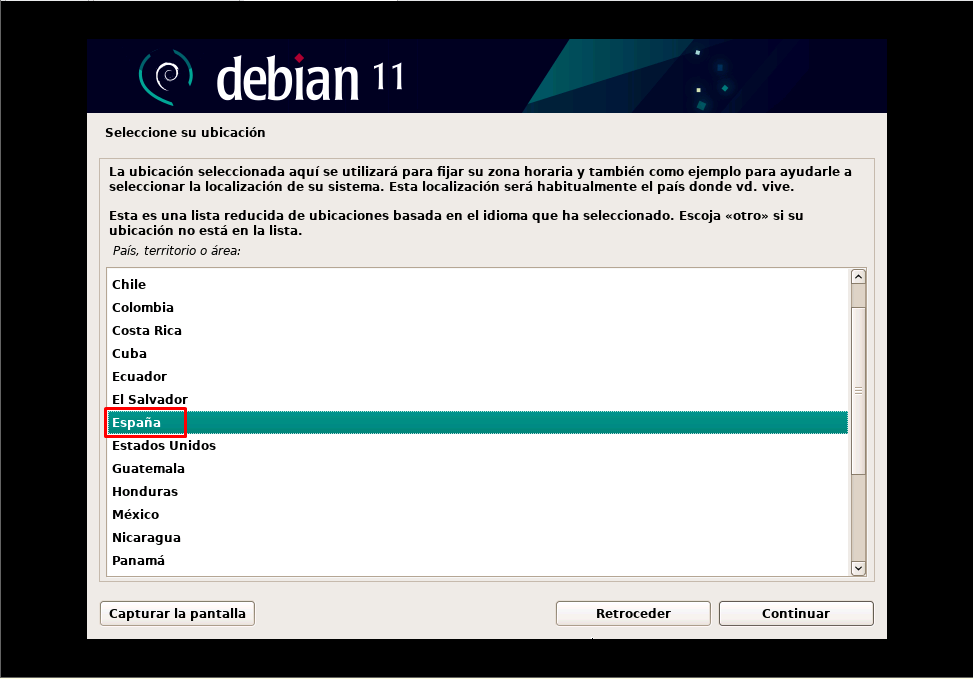
Lo primero que vamos a realizar va a ser la instalación del sistema operativo, empezaremos arrancando la máquina virtual y lo que nos vamos a encontrar va a ser la instalación del Debian, nosotros usaremos la instalación gráfica.



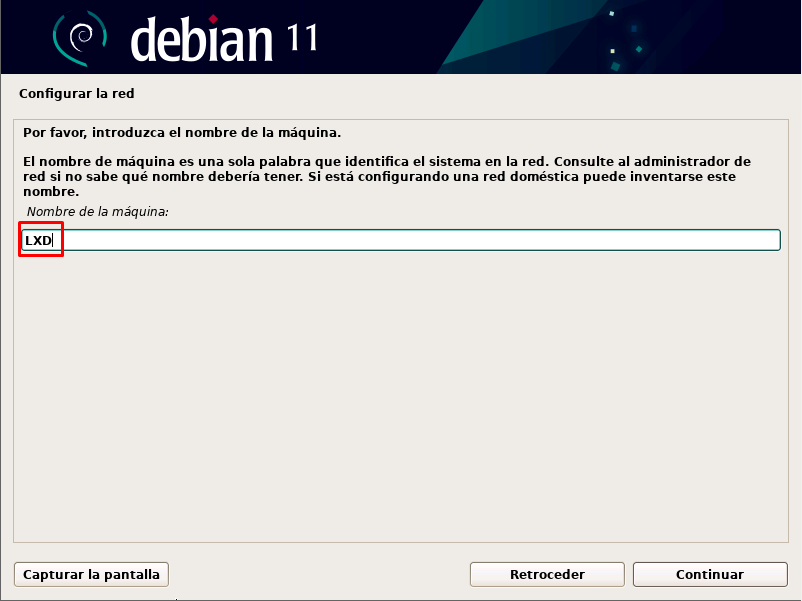
El idioma que elegiremos para nuestra máquina será el español, al igual que la distribución de teclado y de ubicación elegiremos España.



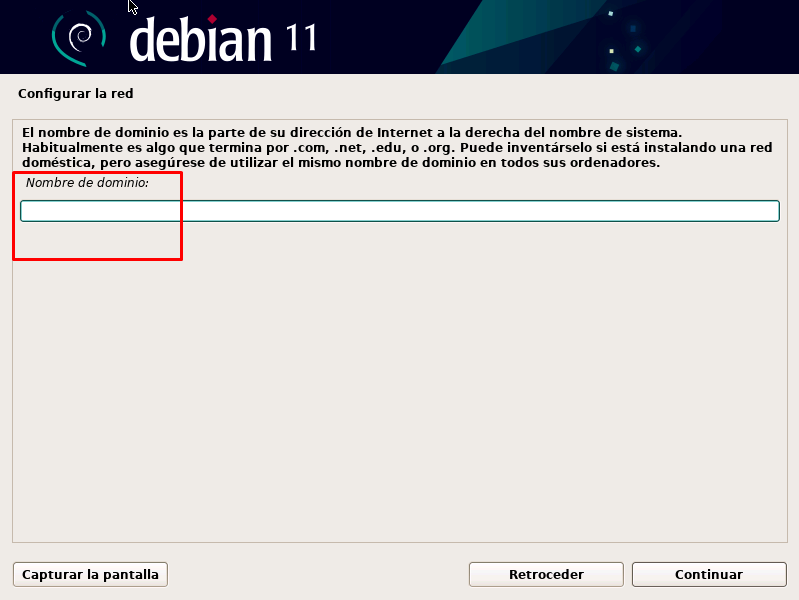




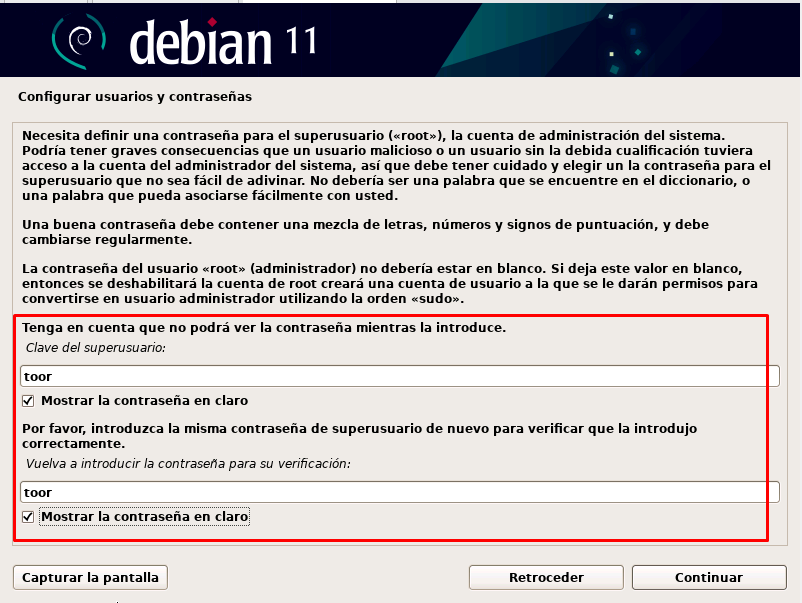
El nombre de máquina que le pondremos será LXD y no agregaremos ningún tipo de proxy ni pertenecerá a ningún dominio.



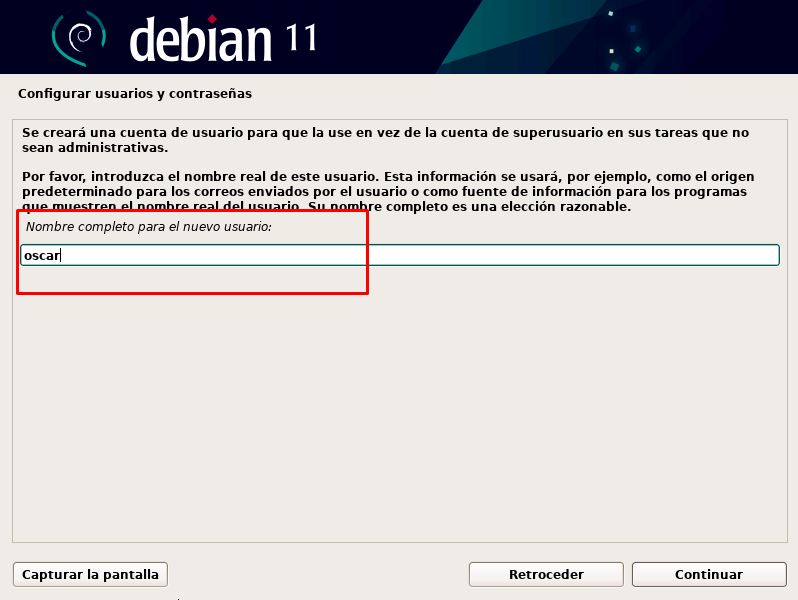
Nombre de dominio vacío

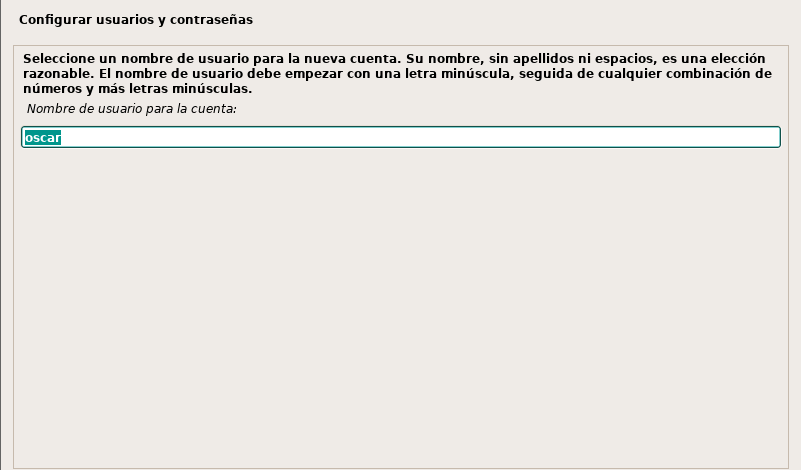


Nos pedirá una contraseña de superusuario, nosotros le pondremos la clásica toor que es root al revés para evitar ningún tipo de confusión en el futuro.

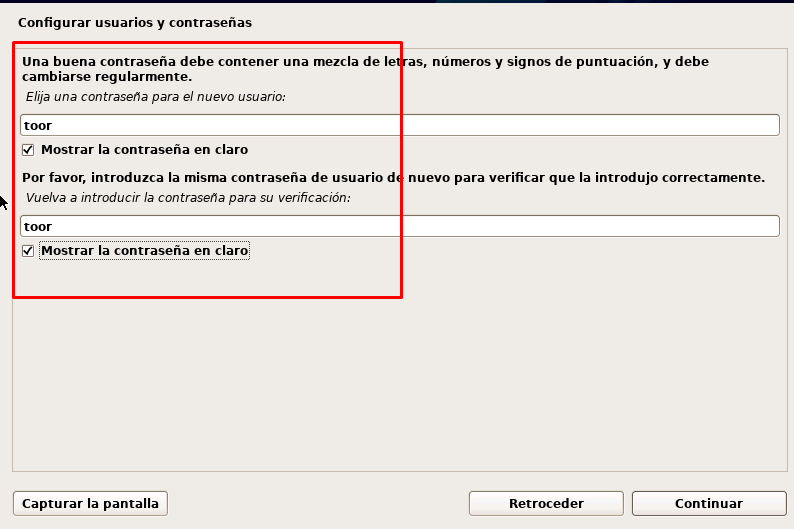


Ahora nos pedirá crear un usuario para usar cuando no tengamos que hacer tareas administrativas, básicamente un usuario sin permisos de root, yo voy a poner mi nombre, así no habrá ninguna confusión.

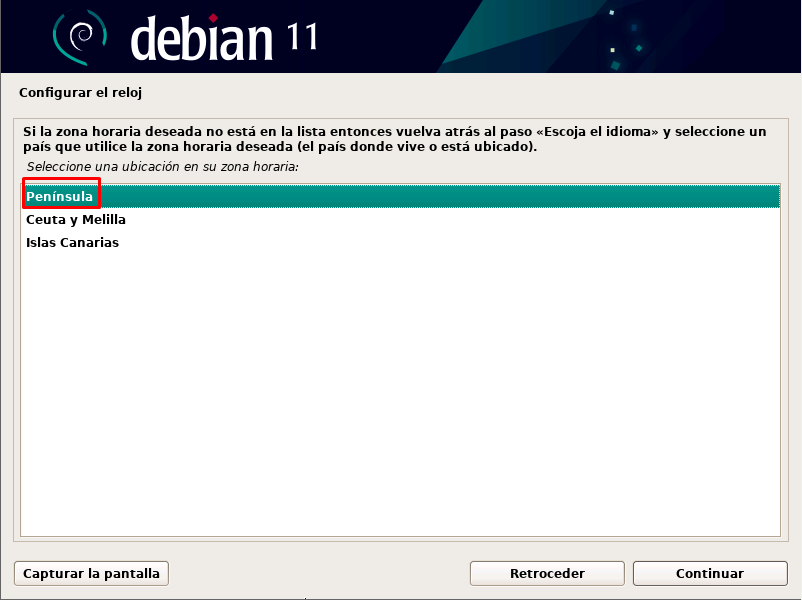




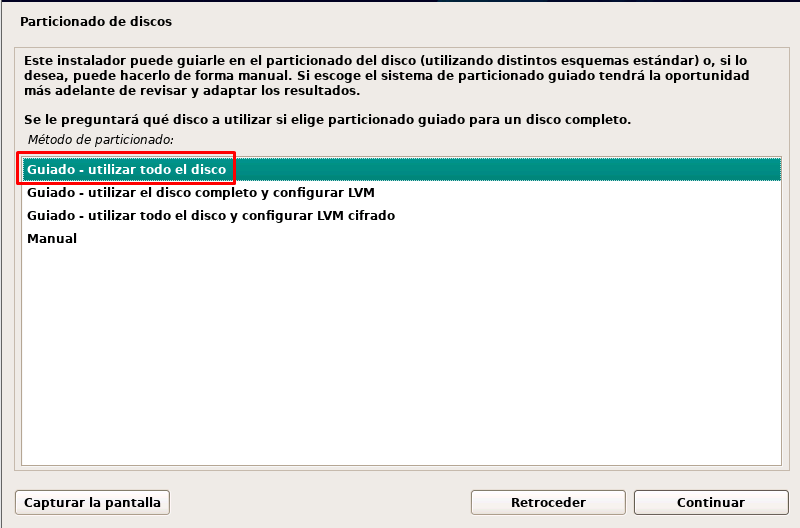
Ahora me pedirá una contraseña para este usuario, voy a ser súper original y voy a volver a poner la contraseña toor.



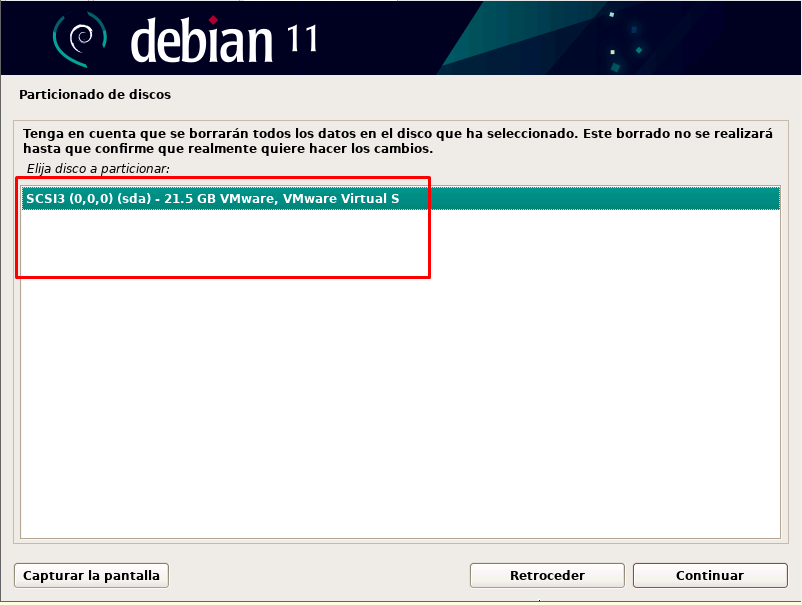
Nos pedirá una zona horaria, nosotros le pondremos la zona horaria de la península.



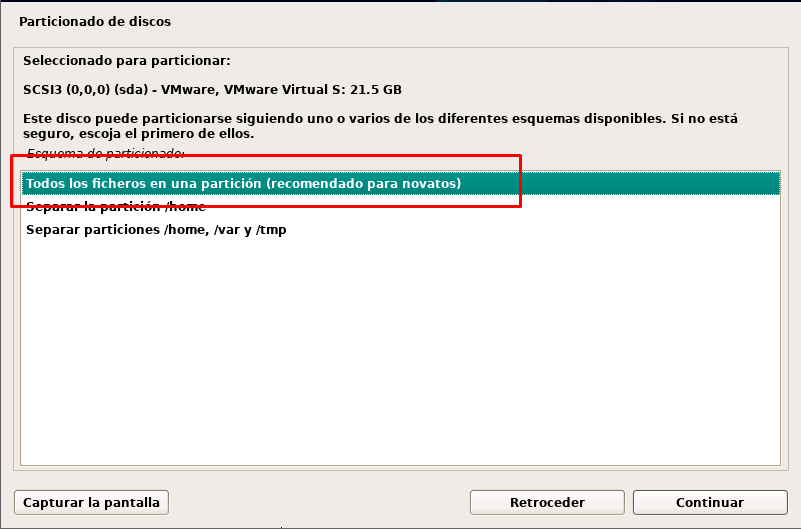
A pesar de que somos administradores y obviamente unos expertos en el particionado de discos, la opción guiada es una muy buena opción para configurar el particionado de un disco si no tenemos ningún Setup del disco planificado. Así que usaremos la opción de particionado guiado.



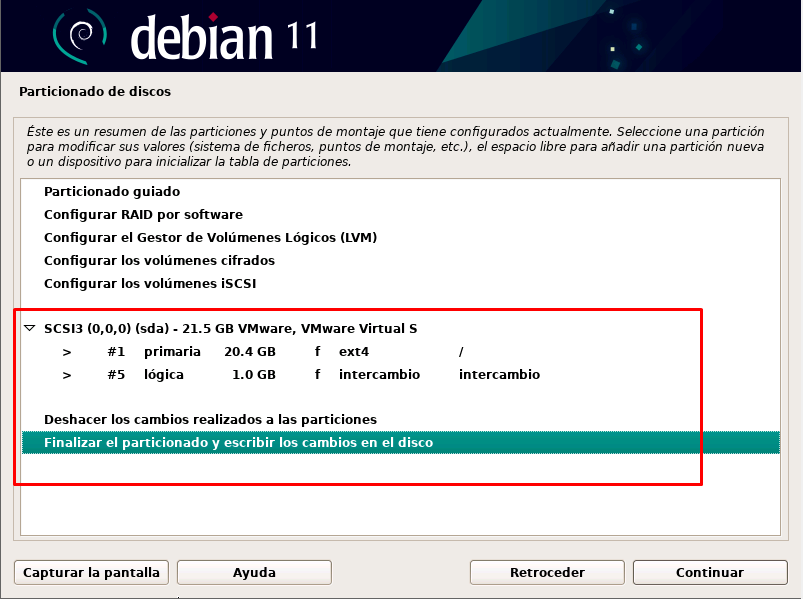
Seleccionamos nuestro disco duro



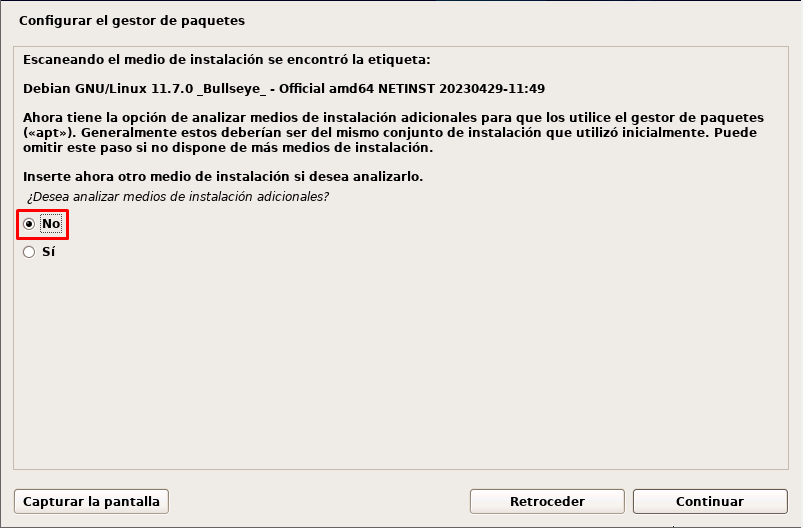
Nosotros no somos novatos, pero si es cierto que la opción de que todos los ficheros en la misma partición es algo muy cómodo y no hay porque sentirse avergonzado por usarlo.



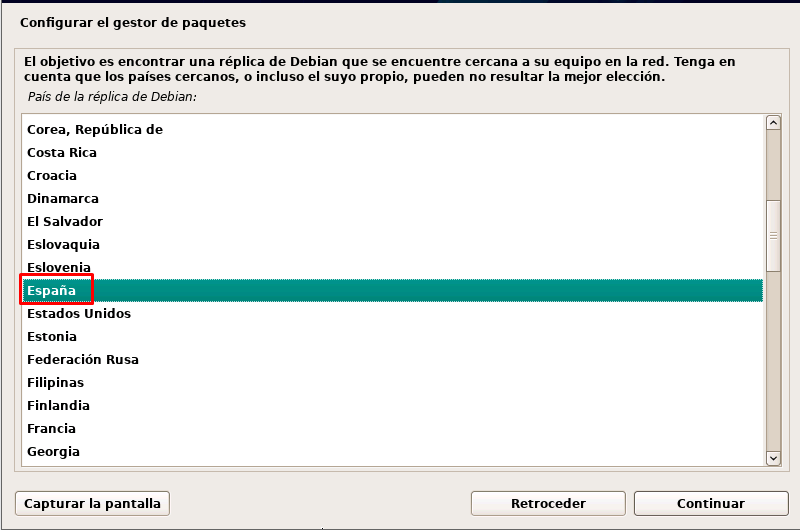
Más o menos el esquema de particionado se quedaría de la siguiente manera.



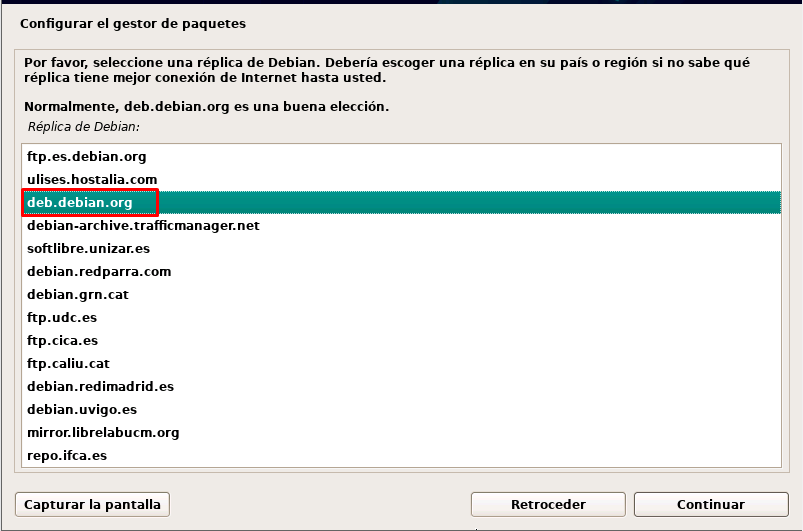
Ahora da el paso de la instalación en el que configuramos todos los parámetros referentes al gestor de paquetes, lo primero que tenemos que hacer es rechazar la encuesta para formar parte del desarrollo de Debian, (si tú quieres ser parte adelante, no influye en nada) yo en mi caso le daré que no.



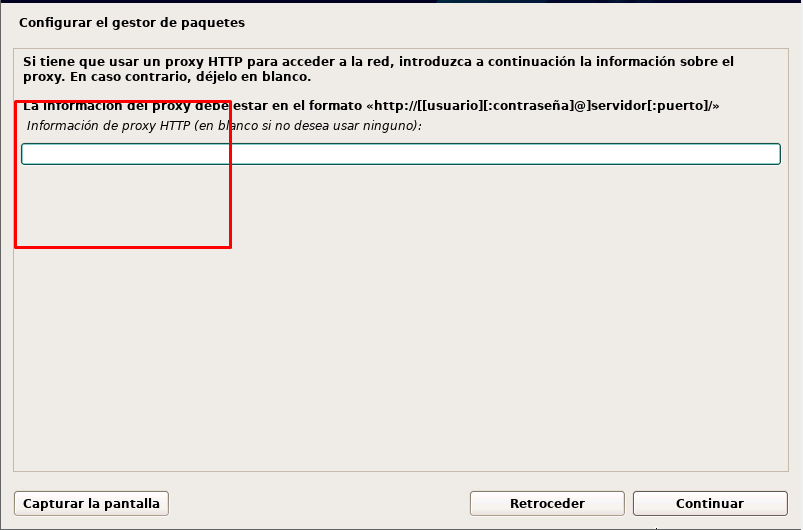
Ahora tenemos que seleccionar la región en la que va a hacer la búsqueda para tener en cuenta que sede de Debian está más cerca de tu localización.



Ahora escogemos el deb.debian.org, a mí me gusta mucho esa réplica, no suele fallar.



El servidor proxy como dijimos anteriormente, no lo vamos a configurar así que lo dejaremos en blanco.



Ahora nos saldrá la pestaña que nos permitirá seleccionar el entorno de escritorio que va a usar y los programas por defecto que va a traer instalado.

Nosotros le pondremos como hemos comentado anteriormente el entorno de escritorio XFCE para que no ocupe tanto espacio nuestra máquina virtual y que no gaste demasiada memoria RAM en su utilización.

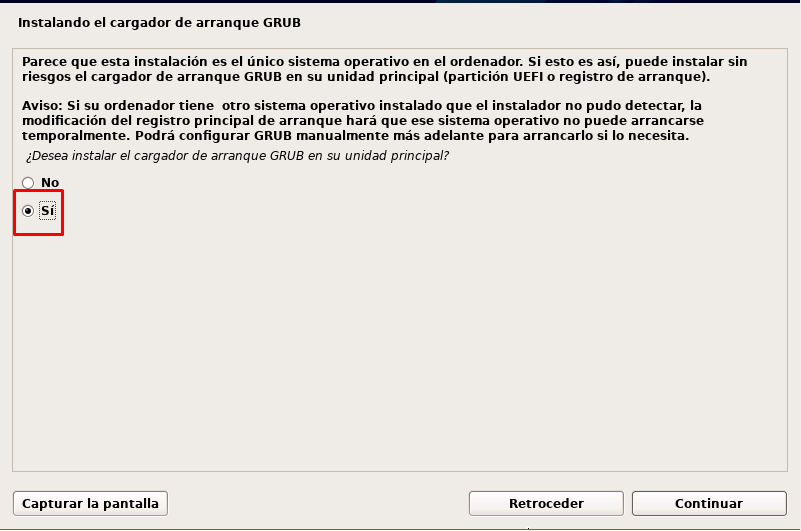
Pondremos que por defecto también traiga SSH server por si acaso hay alguna vez que queramos configurar la máquina por Putty, por ejemplo.

Las utilidades estándar del sistema vienen por defecto marcado así que no tendremos que tocar nada.

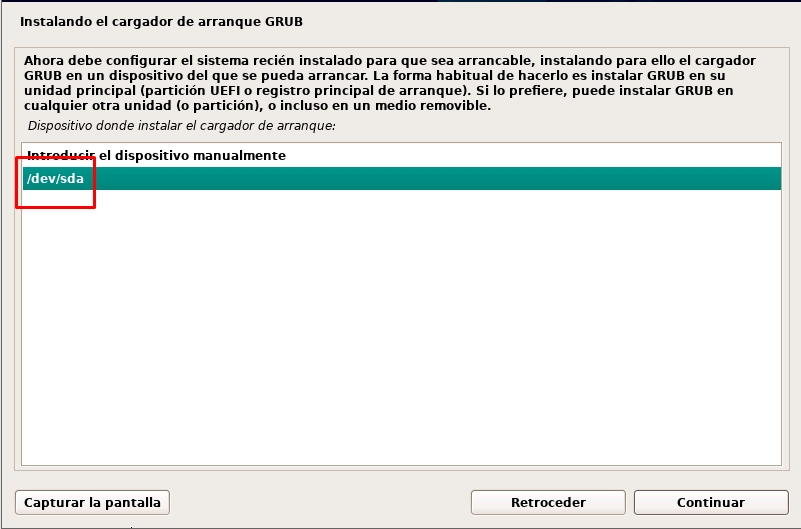
Se quedaría más o menos así:



Por último, seleccionamos el que el grub se instale.



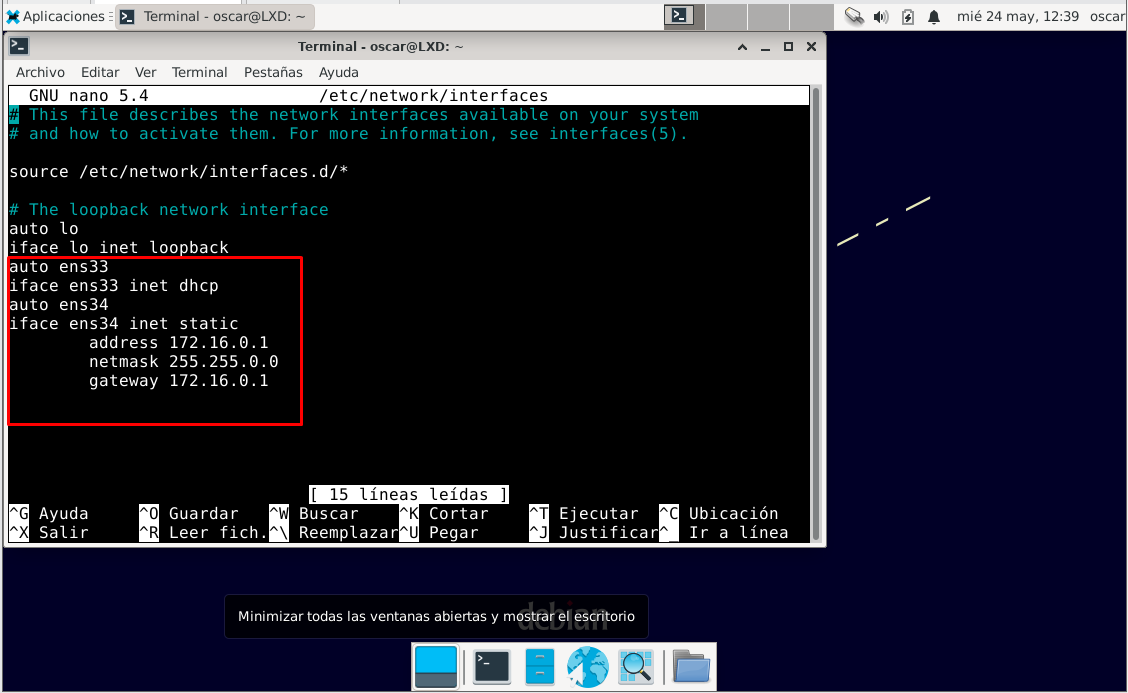
Y seleccionamos el disco donde se va a instalar el gestor de arranque.



Una vez seleccionado esto, cuando acabe tendremos la máquina donde vamos a alojar todo el sistema de LXD.

### Configuración adaptadores de red

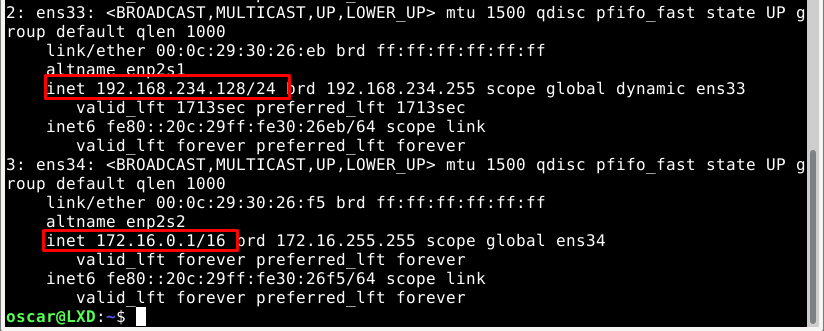
Lo primero que vamos a hacer con la máquina virtual es configurar los dos adaptadores de red que vamos a usar, nos meteremos al fichero /etc/network/interfaces y pondremos la siguiente configuración, donde ens33 es nuestro adaptador que se encarga de hacer NAT con nuestra máquina real para poder tener conectividad a Internet y ens34 es nuestro adaptador de nuestra red en LAN segment que solo tendremos acceso con otra máquina que esté en LAN segment



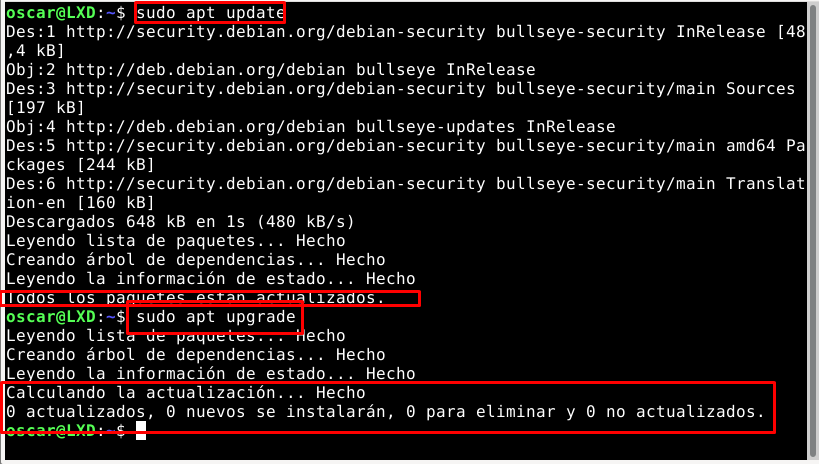
Una vez tengamos eso haremos ifdown, ifup de las 2 interfaces para poder tener la dirección IP para cada uno, por alguna extraña razón que desconozco los ficheros venían vacíos por defecto así que tuve que escribir la interfaz desde 0.



A pesar del error la interfaz se levanta correctamente y tendremos direccionamiento IP para ambas interfaces.



Ahora haremos una actualización del repositorio y de los paquetes actualmente instalados, teniendo en cuenta que es una instalación de Debian nueva no debería de tener ningún tipo de actualización lista.

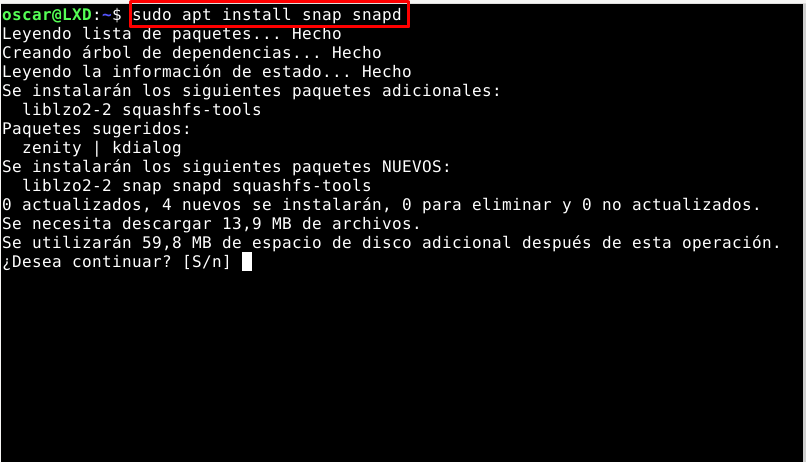


Una vez que tenemos ya todo eso configurado vamos a pasar a la instalación y configuración de LXD.

### Instalación de software

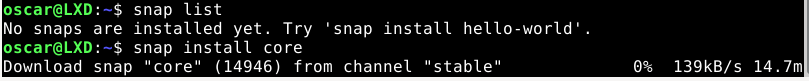
Vamos a instalar el software, realmente seguiremos los pasos que especifiqué al principio del trabajo ya que así es como lo vamos a configurar.

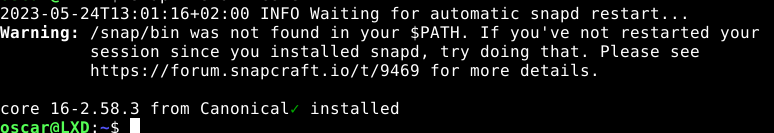
Lo primero que hacemos es instalar Snap y SnapD



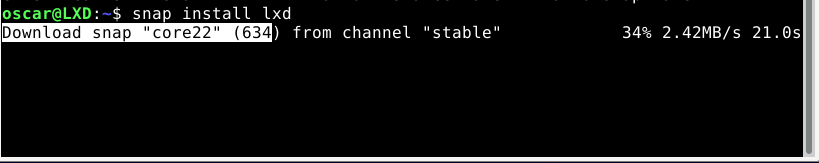
Esperamos a que se instale y conseguimos el siguiente paquete.

Ahora el siguiente paquete es el core, este lo instalaremos a través de snap con el comando snap install core lo instalaremos, una vez que acabe nos pedirá que hagamos log out y log in para poder tenerlo en funcionamiento.



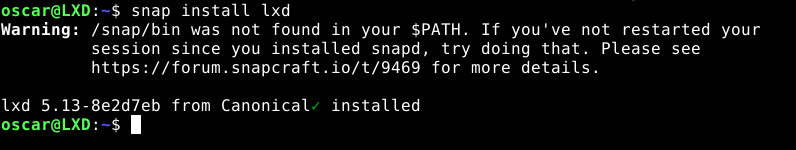


Una vez que hayamos instalado el core y cerrado e iniciado sesión de nuevo, podremos instalar el paquete LXD.



Primero instalará la dependencia del core22 y luego ya instalará el paquete correspondiente.

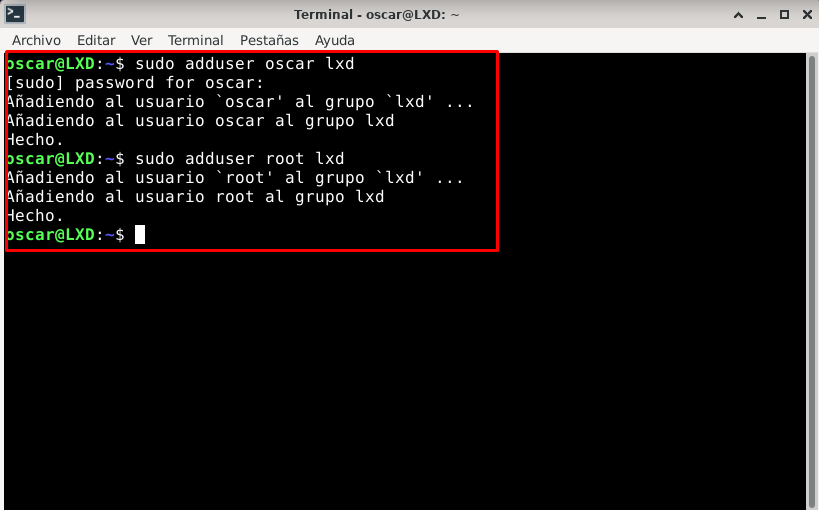
Una vez instalado nos saldrá lo siguiente.



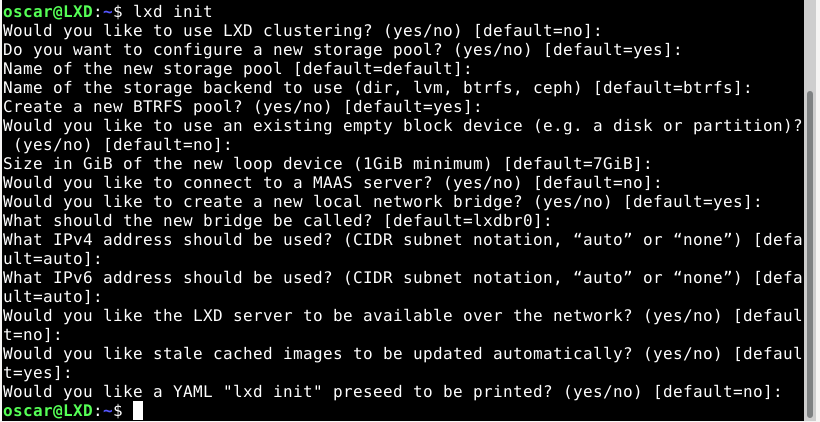
Simplemente haremos un reboot del equipo y empezará a funcionar todo correctamente.

### Configuración inicial de LXD

Vamos a hacer la configuración inicial para que el servicio ya comience a funcionar en nuestro mini Setup añadiremos nuestro usuario al grupo LXD, por si acaso también añadiremos al usuario root por si queremos tocar o hacemos algo con root.



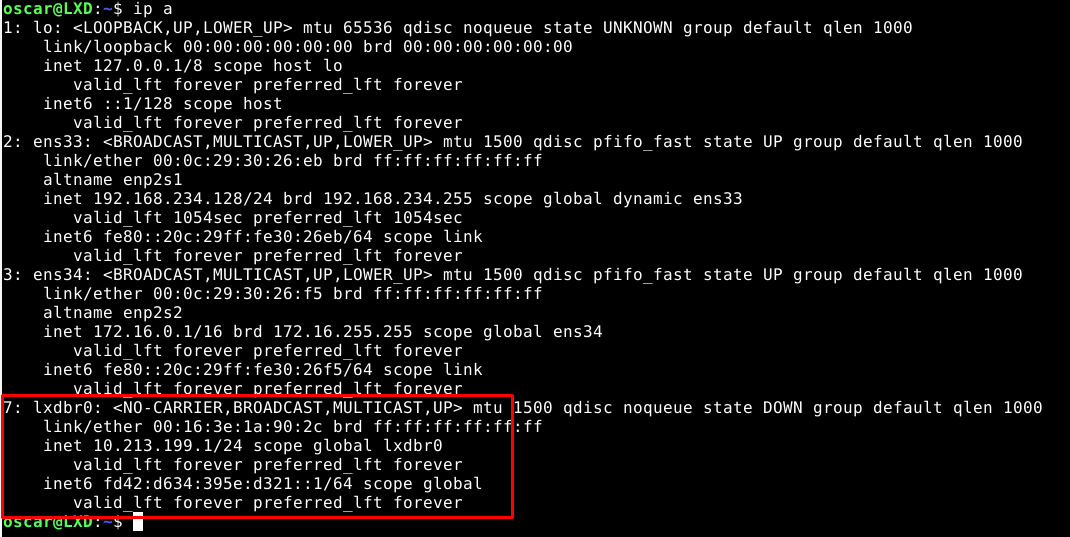
Una vez que ya lo tenemos añadido al grupo vamos a hacer lxd init para iniciar el servicio.

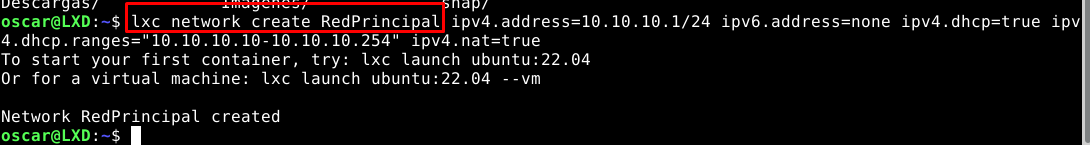


Pondré toda la configuración por defecto ya que luego crearemos nuestra propia red como ya especificamos en las explicaciones previas.

### Creación de la red y perfil aplicable

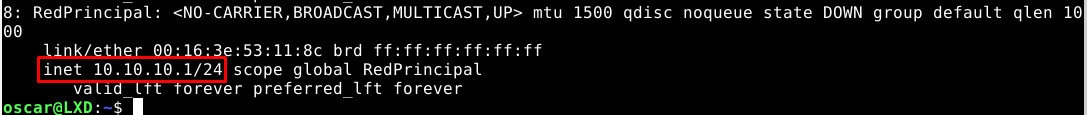
Como podemos ver después de haber hecho el init nos habrá creado una interfaz lxdbr0 que va a ser la interfaz que haga de bridge para las máquinas por defecto, pero a nosotros esta interfaz no nos interesa del todo ya que tiene un rango de direcciones bastante extraño

cómo podemos ver tiene la 10.213.199.1, a mí la red 10.213.199.0/24 no me gusta mucho teniendo en cuenta que es un poco críptica así que vamos a crear una nueva red que se acomode a nuestros gustos.



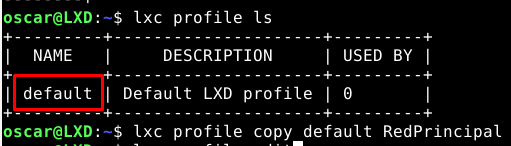
El comando que he utilizado es: “lxc network create RedPrincipal ipv4.address=10.10.10.1/24 ipv6.address=none ipv4.dhcp=true ipv4.dhcp.ranges=”10.10.10.10-10.10.10.254” ipv4.nat=true”

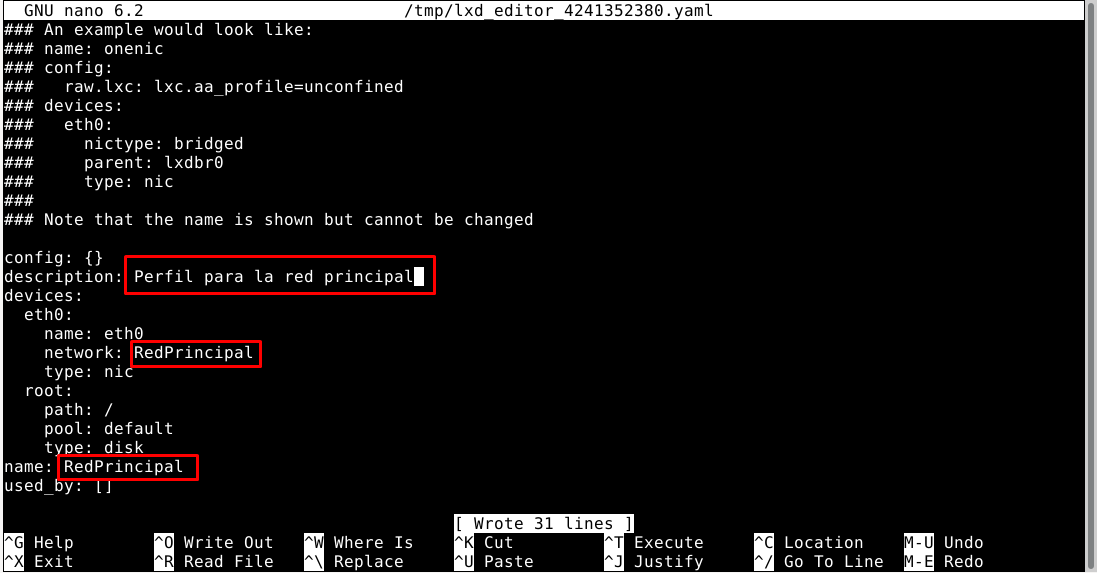
Ahora que hemos creado eso, nos habrá aparecido un adaptador virtual bridge con la dirección IP que nosotros le habremos asignado



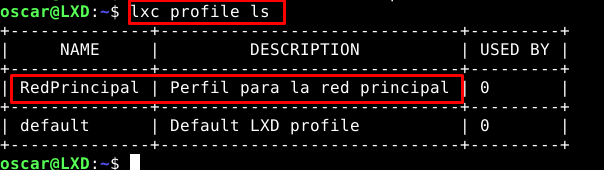
Una vez que tenemos ya nuestra red creada vamos a crear el perfil que vamos a utilizar para aplicárselo a los contenedores.

Teniendo en cuenta que se crea un perfil por defecto que viene con todo configurado sin que tengamos que hacer nada, vamos a copiar el perfil default a un perfil nuestro y simplemente le cambiaremos la red que aplica.





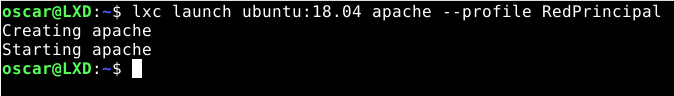
Así solo tenemos que poner las características para nuestra red, nada complicado.



### Creacion de servidor WEB

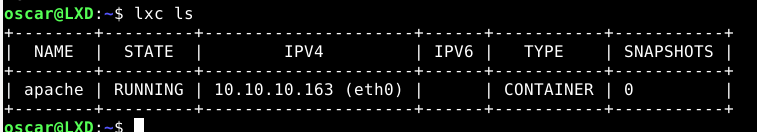
La primera instancia que vamos a hacer va a ser un contenedor de un servidor web, lo primero que vamos a hacer es lanzar una nueva instancia que va a coger el perfil nuevo que nosotros hemos creado anteriormente.

Vamos a utilizar el comando “lxc launch ubuntu:18.04 apache --profile RedPrincipal”



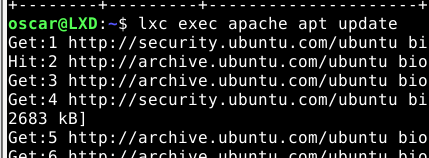
La primera vez que lances una instancia va a tardar mucho si no tienes la imagen descargada, pero luego se harán mucho más rápido, recomiendo usar distribuciones nuevas, porque tienen la nueva versión de los repositorios, si usas versiones antiguas, los repositorios no existirán y no podrás instalar nada a menos de que los exportes o le cambies los repositorios a los nuevos.

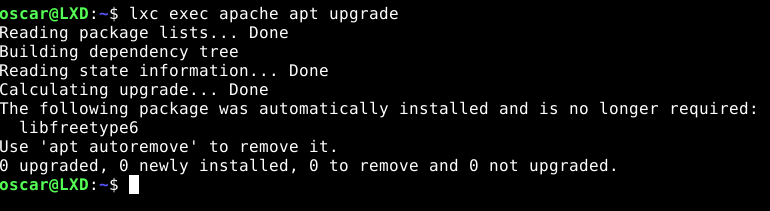
Ahora que ya tenemos la máquina vamos a comprobar que se ha creado correctamente



#### Configuración de red y paquetes

Lo primero que vamos a hacer con nuestro contenedor Linux va a ser actualizarlo, le pasaremos un apt update y un apt upgrade

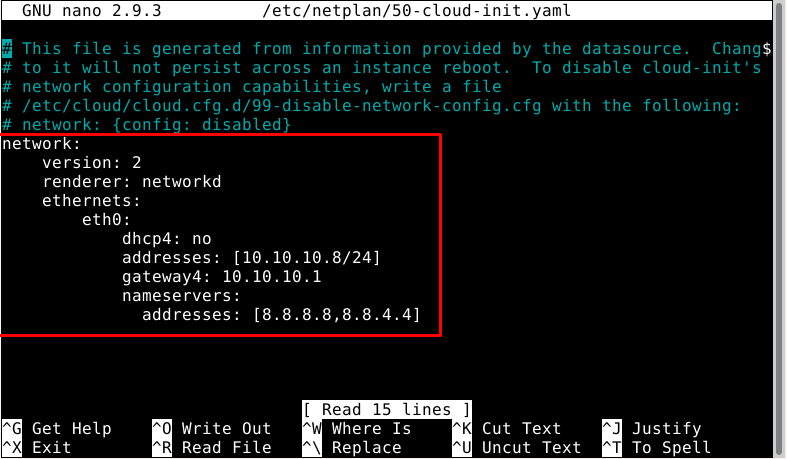




Ahora lo primero que vamos a hacer es cambiarle la dirección IP a estática, para ello nos metemos en el contenedor.



Ahora una vez que estamos dentro vamos a cambiar el Netplan para que podamos poner la configuración en static.





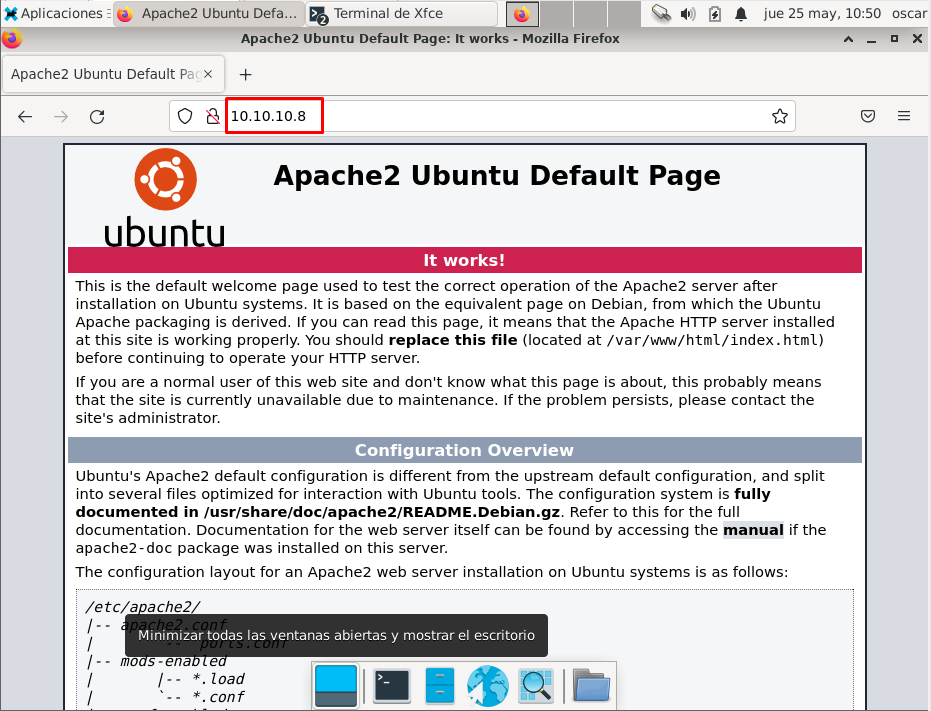
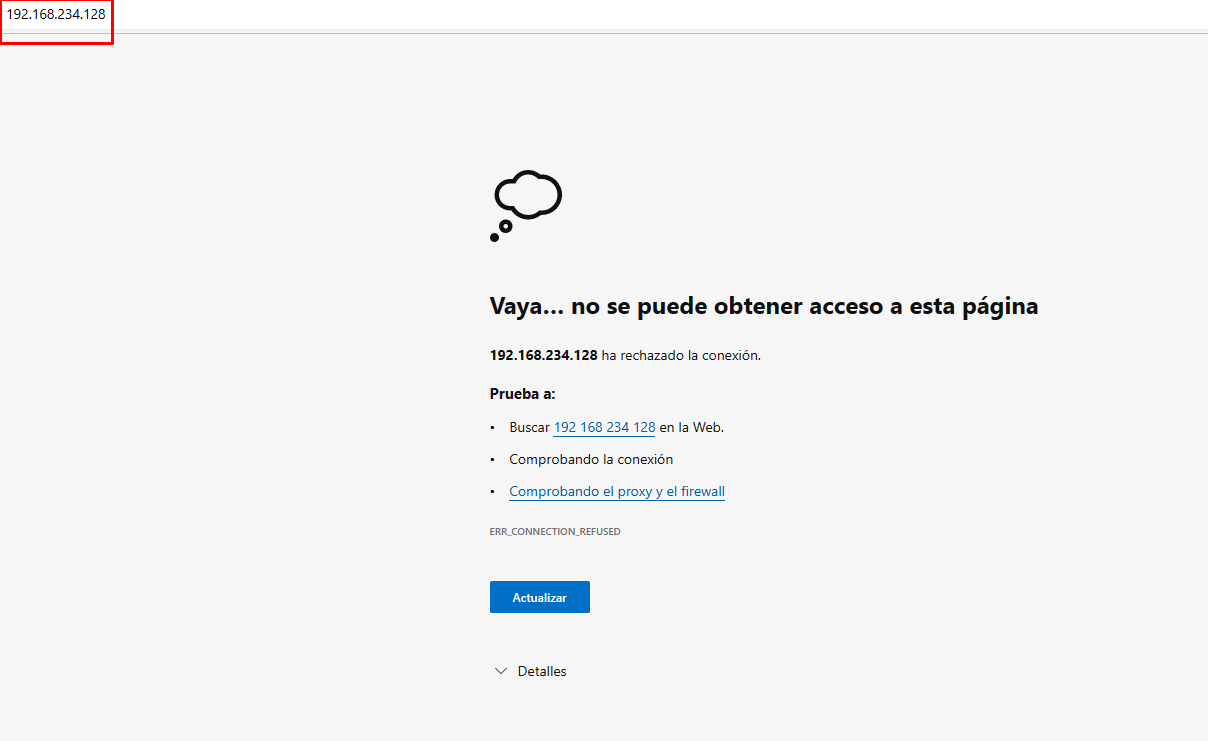
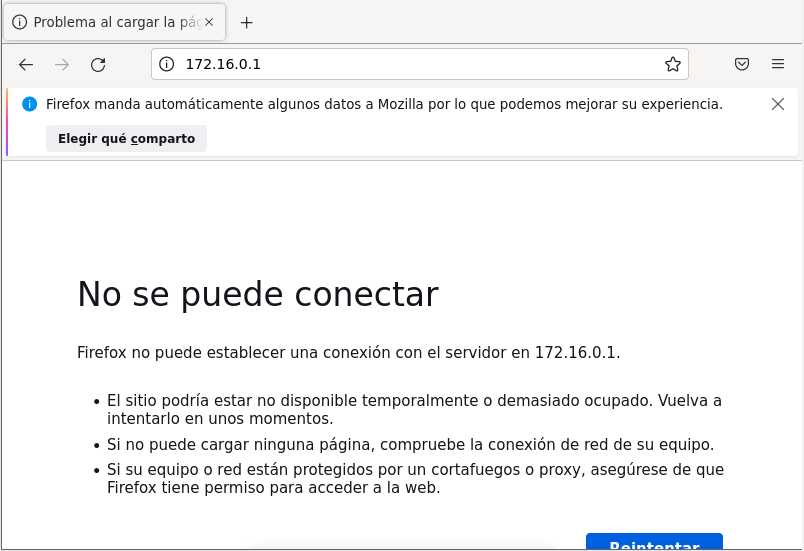
Con ese comando ya se aplica la configuración de red que le habíamos asignado en el fichero y ahora vamos a instalar apache2 para montar nuestro servidor WEB

#### Instalación del servidor WEB y redirección PROXY



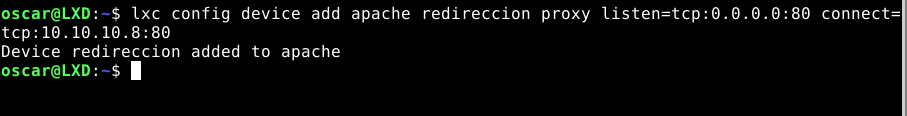
Una vez que se acaba de instalar tenemos el servidor WEB montado en el contenedor, pero tiene un pequeño problema, este contenedor no es accesible desde fuera de la red, es accesible desde el host de LXD, porque tiene una interfaz en la red, pero no va a ser accesible desde la máquina real ya que tiene una NAT creada, y por supuesto desde otros equipos de la red del host real tampoco va a ser accesible, no por lo menos teniendo un sistema virtualizado.

Vamos al menos a conseguir que desde nuestra máquina real se le pueda hacer ping y que también se le pueda hacer ping desde el LAN segment que no tiene nada que ver con el contenedor.

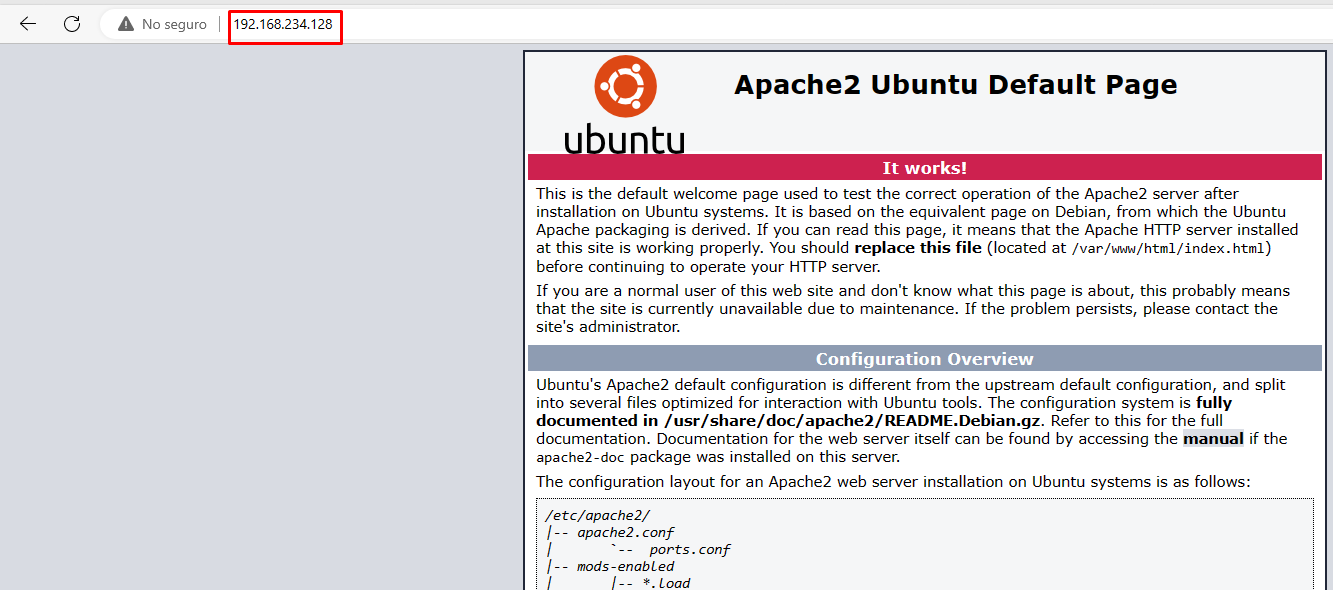
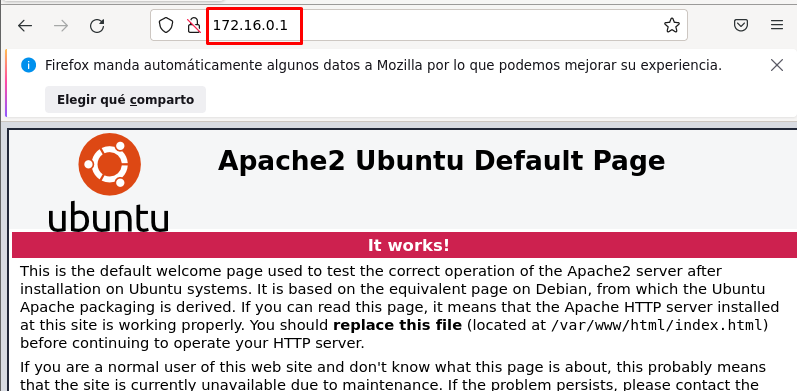
* Máquina virtual con hostname “LXD”:
  + 
* Máquina real:
  + 
* Máquina virtual en el LAN segment:
  + 

Como podemos ver no nos deja acceder desde la maquina real ni desde la que tiene LAN segment.

La solución a esto va a ser añadir como una especie de configuración de proxy que nos permite LXD con el siguiente comando.



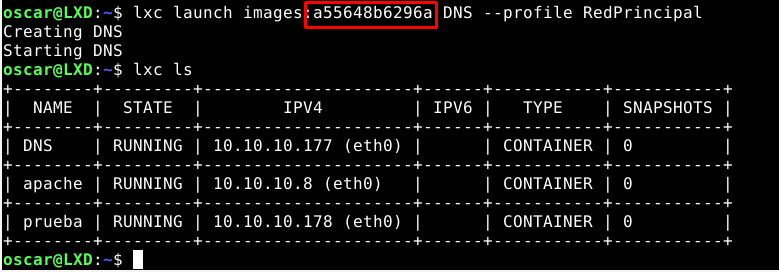
Con ese comando lo que le estamos diciendo a LXD es que añada un dispositivo de configuración para el contenedor apache que se llame redirección y que funcione como proxy para que escuche todas las peticiones tcp que pasen por el puerto 80 y las redirija al contenedor también al puerto 80, con esto al menos tendremos conectividad a través de las direcciones IP de la máquina virtual host de LXD.

* Maquina real:
  + 
* Maquina en LAN segment:
  + 

Ahora tenemos conectividad, bueno tenemos nuestro prototipo de servidor a través de un contenedor redireccionado con un proxy.

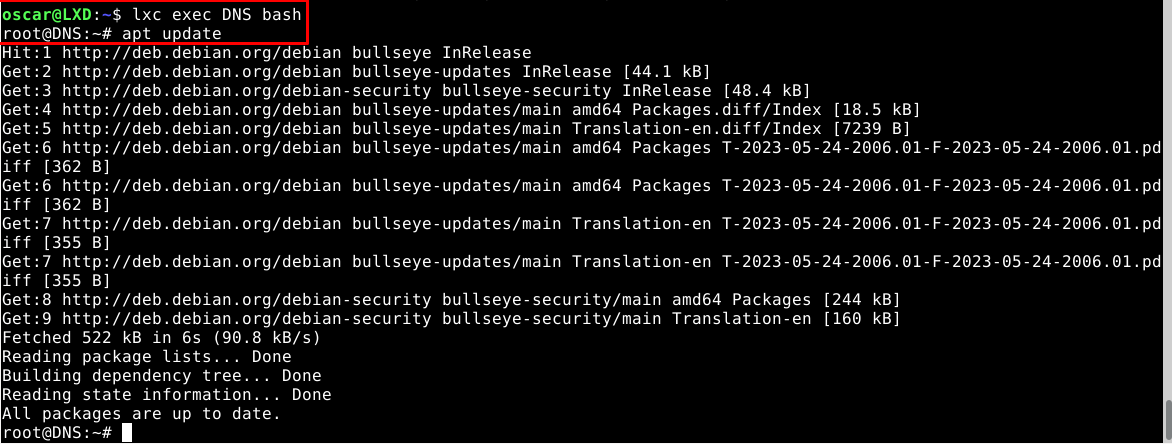
### Servidor DNS

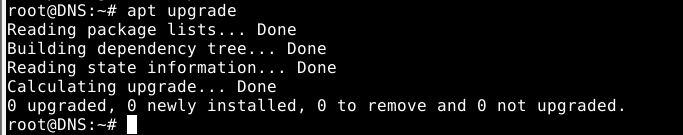
Primero vamos a crear la instancia que va a usarse como servidor DNS, la imagen es un Debian 11 de contenedor para hacer las cosas un poquito más complicadas y tener diferentes distribuciones conviviendo en el mismo sistema



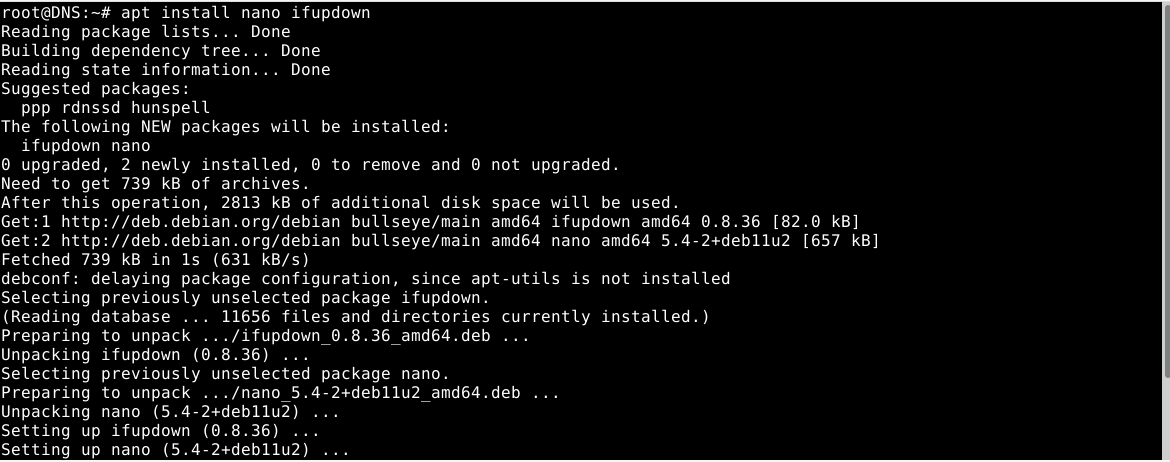
#### Configuración de red y paquetes

Vamos a hacer un apt update y apt upgrade como en la anterior máquina:





Ahora le pondremos la dirección IP estática, teniendo en cuenta que es un Debian de contenedor, hay algunas características que no están incluidas, una de ellas es que el editor nano no está incluido, así que lo instalaremos y la más importante de todas, el paquete ifupdown que es el que agrega el /etc/network/interfaces tampoco está instalado, por lo cual instalaremos esos 2 paquetes.



Una vez que ya los tenemos instalados simplemente configuraremos la ree

No sé por qué no funciona el /etc/network/interfaces, así que he buscado una alternativa a la configuración de red estándar por el fichero /etc/systemd/network/eth0.network

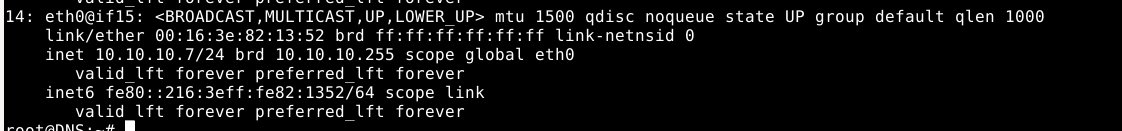


Una vez que tenemos eso así vamos a hacerle una copia al /etc/network/interfaces por si acaso.

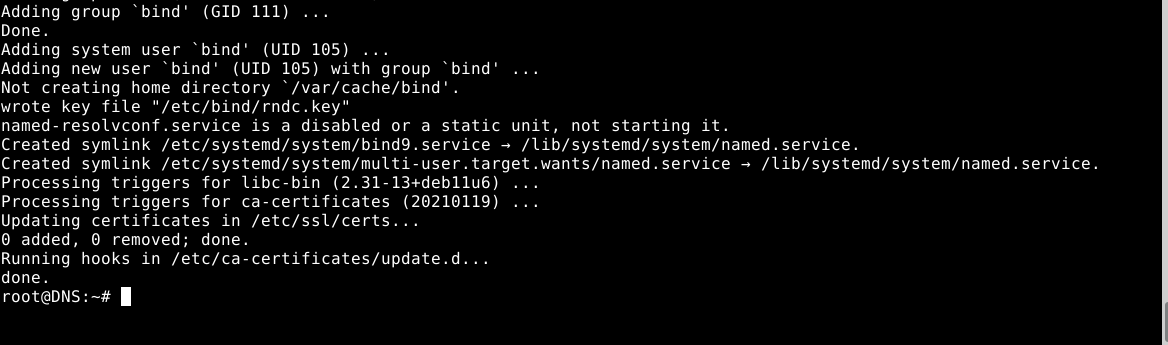


Y dejaremos el servicio en enable y reiniciaremos el contenedor.

Y por lo que nosotros podemos ver la ip se ha cambiado correctamente.

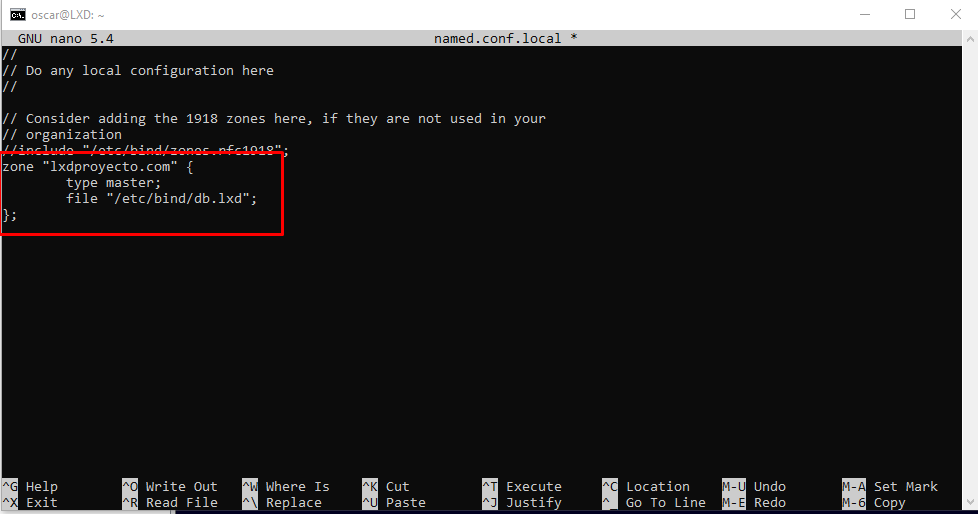


Ahora vamos a instalar el servicio DNS bind9.

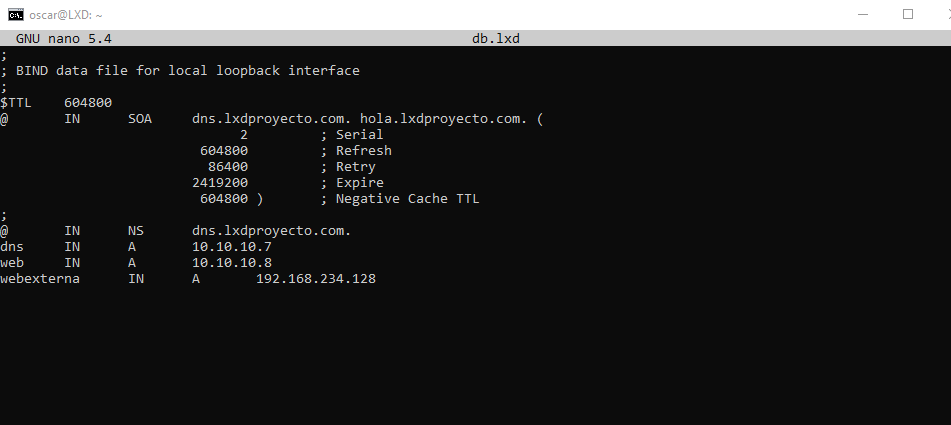


Una vez instalado vamos a pasar a configurarlo, primero vamos a crear nuestra zona DNS.

Para hacer esto más fácil vamos a entrar por ssh para copiar y pegar ciertas cosas.



Una vez que tenemos nuestra zona vamos a crear los ficheros para poder configurar correctamente las resoluciones de nombres.



Ese va a ser nuestro fichero de configuración por ahora de nuestro servidor de nombres, ahora con una máquina de prueba vamos a ver si es capaz de resolver todo correctamente.

Hay una cosa a tener en cuenta, realmente nuestro equipo que contiene todas las máquinas LXD actúa como servidor, y si pones de servidor DNS la máquina que es el DNS todos los contenedores que tengan DHCP van a coger como servidores DNS la resolución de esta máquina.

Si vemos un nuevo contenedor prueba que he creado veremos que este contenedor tiene su ip por dhcp, pero aun así resuelve lo que nosotros hemos especificado en el servidor DNS.

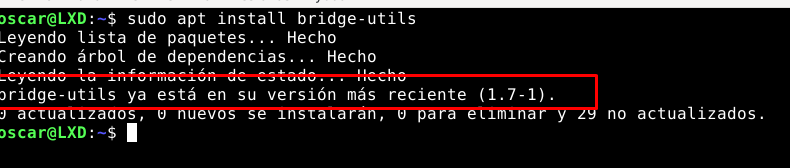


Como vemos resuelve bastante bien.

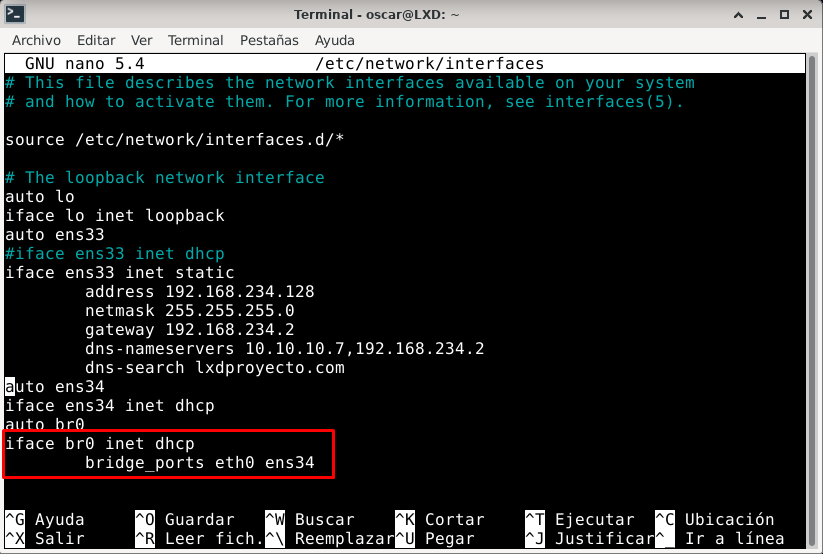
## DHCP

Ahora vamos a instalar un servidor DHCP para nuestros clientes, pero antes de instalar el servidor DHCP nos encontraremos con el grave problema de que realmente estos contenedores no son accesibles en la red local, a pesar de tener la opción NAT no son accesibles desde la red, por lo cual lo que nosotros tenemos que hacer es que los contenedores sean parte de nuestra red local, esto realmente no es necesario para que todo funcione pero si queremos que nuestro sistema sea incluso más funcional es bastante recomendado.

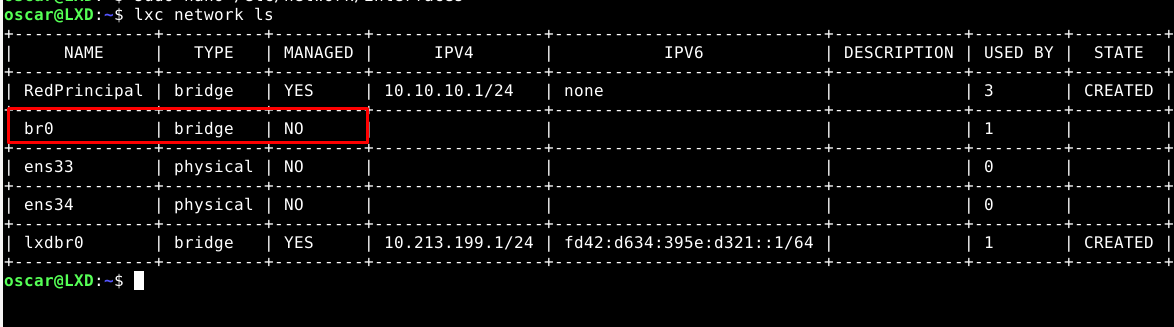
Para ello primero nos instalamos el paquete de utilidades “bridge-utils” con este nos permitirá hacer puentes de red en nuestra máquina Debian como si fuera un router.



Una vez que estén instalados, en el /etc/network/interfaces podemos crear nuestro puente con la siguiente sintaxis:

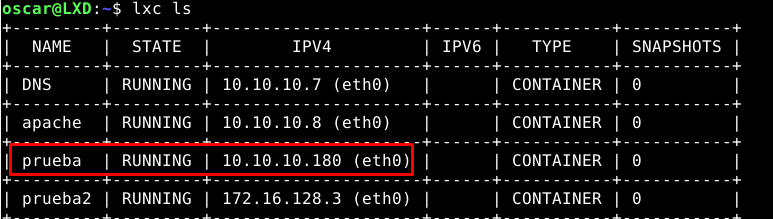


Una vez que eso está configurado, la interfaz será del tipo puente, y si lo vemos en el networking de LXD se verá como tal



Ahora podemos asignar esa interfaz a un contenedor con un solo comando, un poco largo.

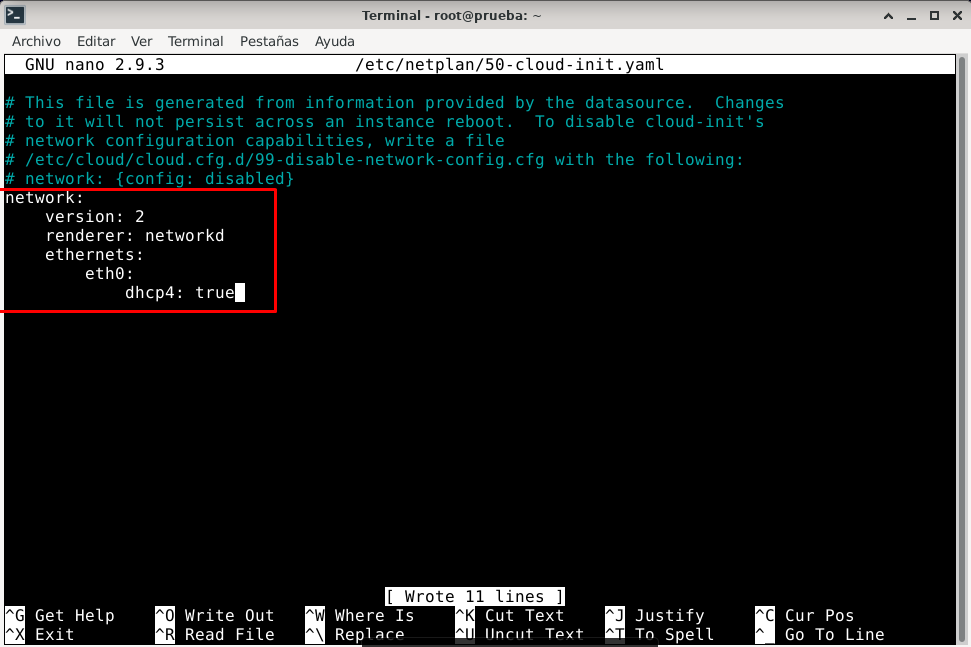
Primero vamos a añadir una máquina que se llame prueba y le vamos a pasar el comando.



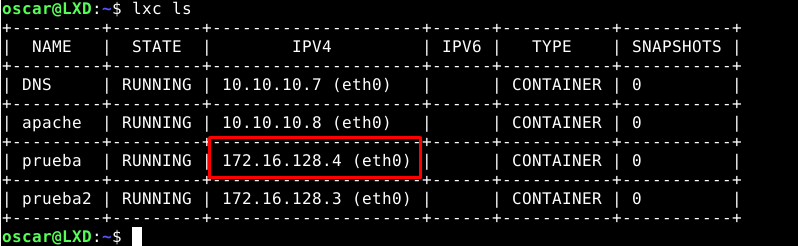
Y veremos que esa máquina cogerá dirección IP de las que distribuye la red RedPrincipal, pero con el comando añadirá la interfaz y cogerá dirección IP del dhcp del adaptador host-only, ya que tuve que cambiar el adaptador a host-only para mejor funcionamiento.



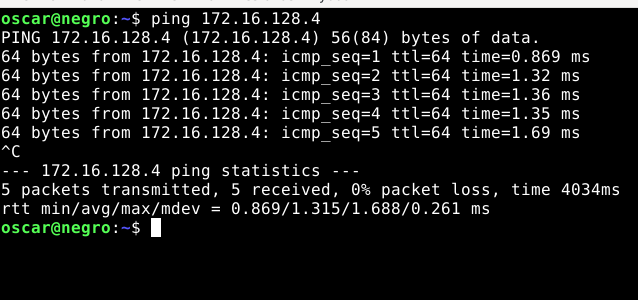
Ahora lo que vamos a hacer es en prueba asignarle la dirección por dhcp.



Hacemos Netplan apply y veamos si ha cogido dirección IP de nuestro dhcp.



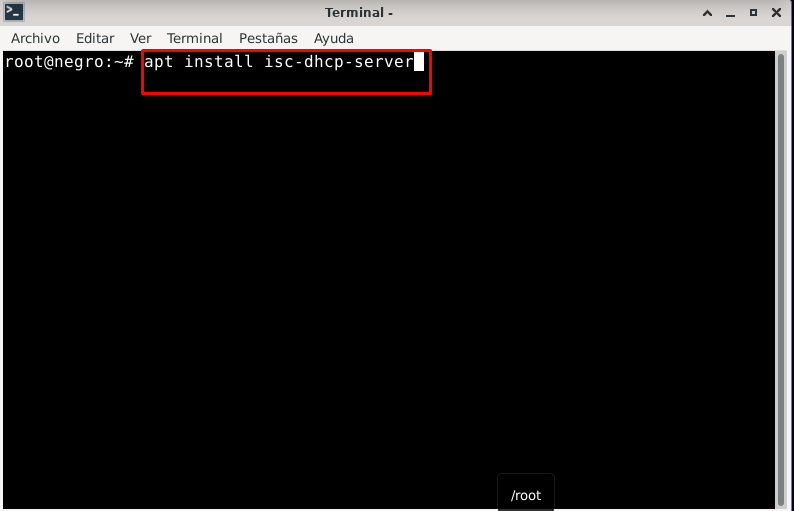
Si hacemos ping desde una máquina que esté en la misma red como vemos seremos capaces de alcanzar el contenedor.

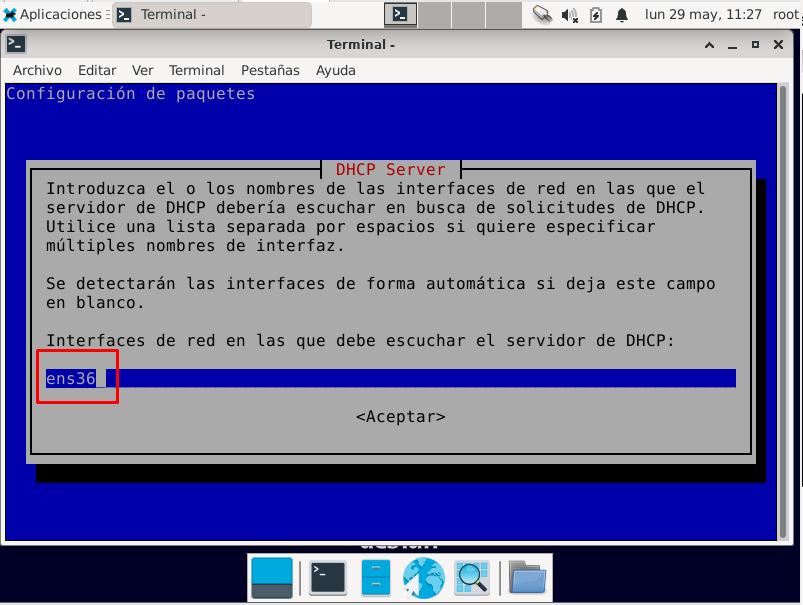


Ahora que vemos que ya es integrable dentro de nuestra red, vamos a crear un servidor dhcp que reparta direcciones y así no usaremos el de LXD por defecto que es solo de comunicación interna.

Vamos a utilizar una máquina virtual aparte simulando el router y le instalaremos un servidor DHCP

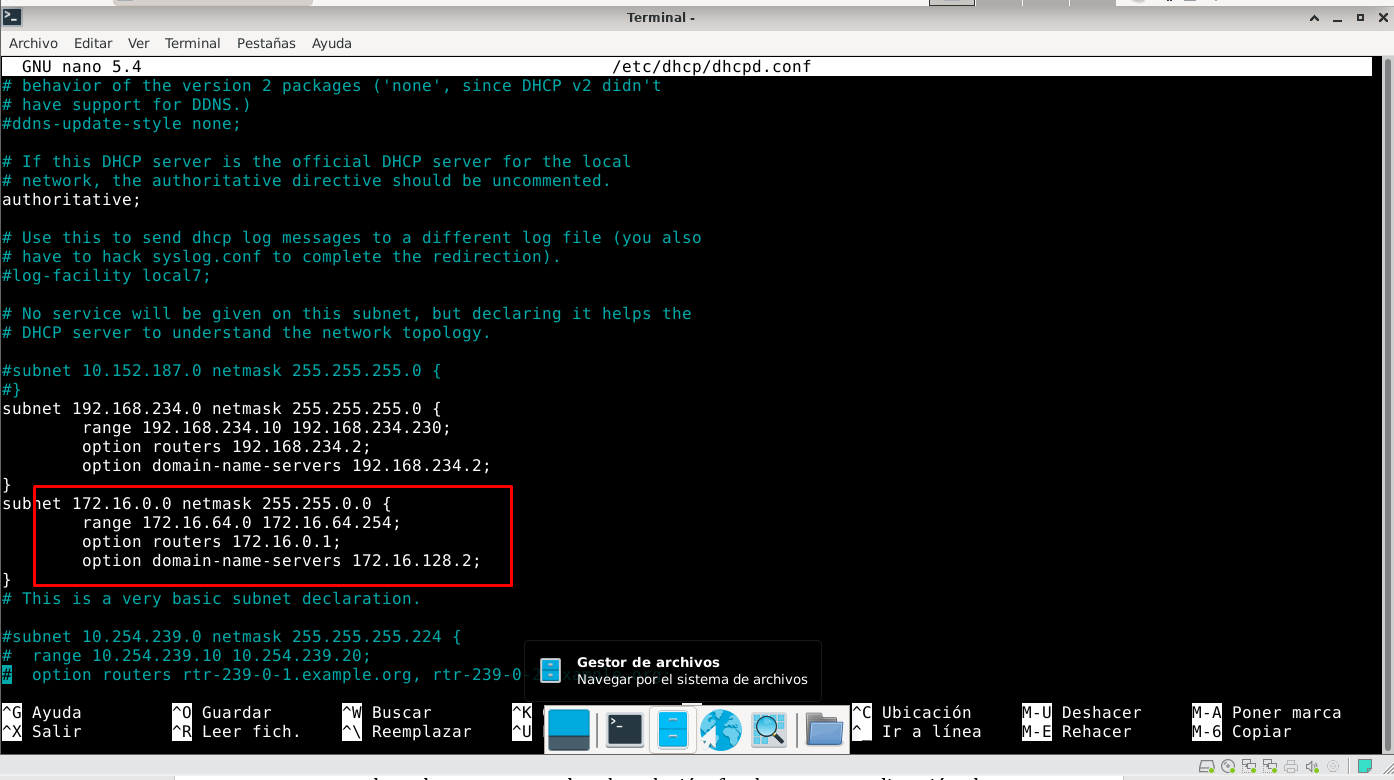
Nosotros usaremos el paquete isc-dhcp-server y lo que haremos será un dpkg-reconfigure para empezar a ponerlo todo desde el principio.



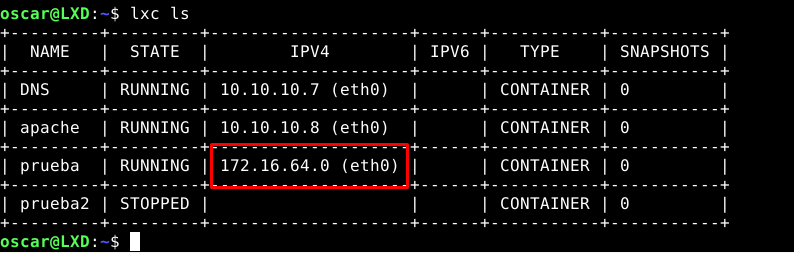


En la captura me equivoqué, mi adaptador es el ens33.

Ahora tocaremos el archivo de configuración de dhcp.



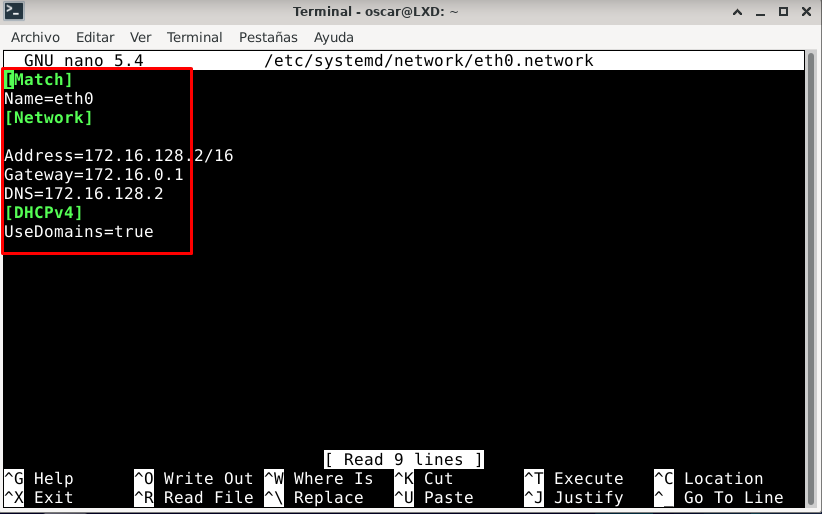
Ahora que ya lo tenemos simplemente vemos que algún contenedor resuelva, en teoría prueba que está configurado como dhcp tiene que servir



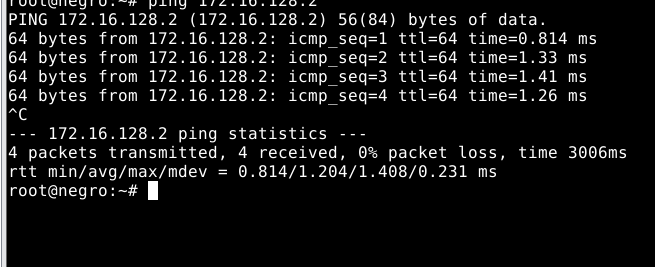
En el dhcp hemos puesto que el servidor que ponga a resolver los DNS tiene que ser la 172.16.128.2, así que cambiaremos la configuración del DNS para que ahora sea parte de la red y empiece a resolver adecuadamente todo.



Ahora le vamos a poner a DNS la dirección estática que hemos asignado en el DHCP

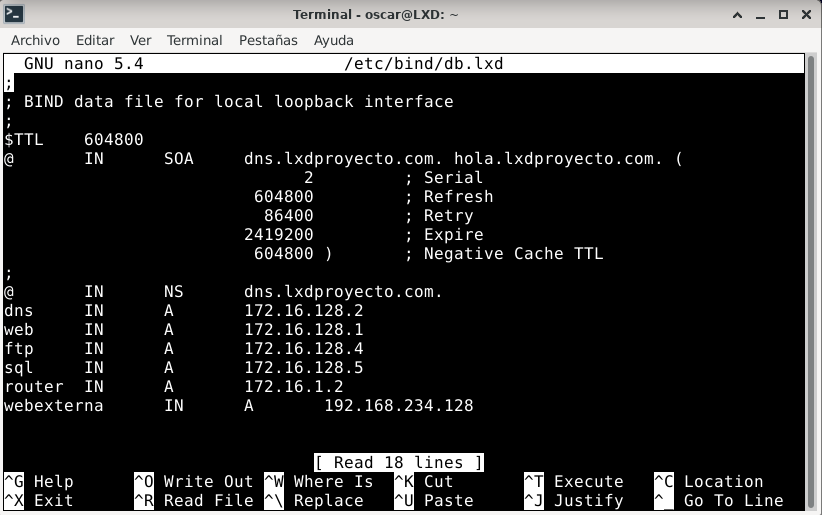


Comprobamos que haga ping desde nuestro “router”

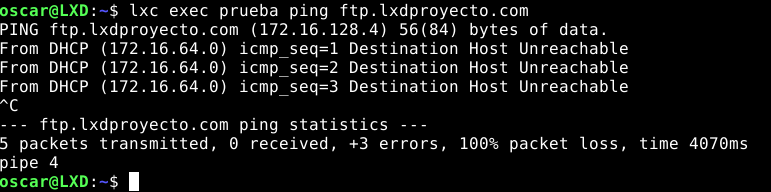


Hay ping

Ahora cambiamos los registros del DNS ya que las direcciones IP tienen que ser asignadas de nuestra red, vamos a comprobar que resuelve.

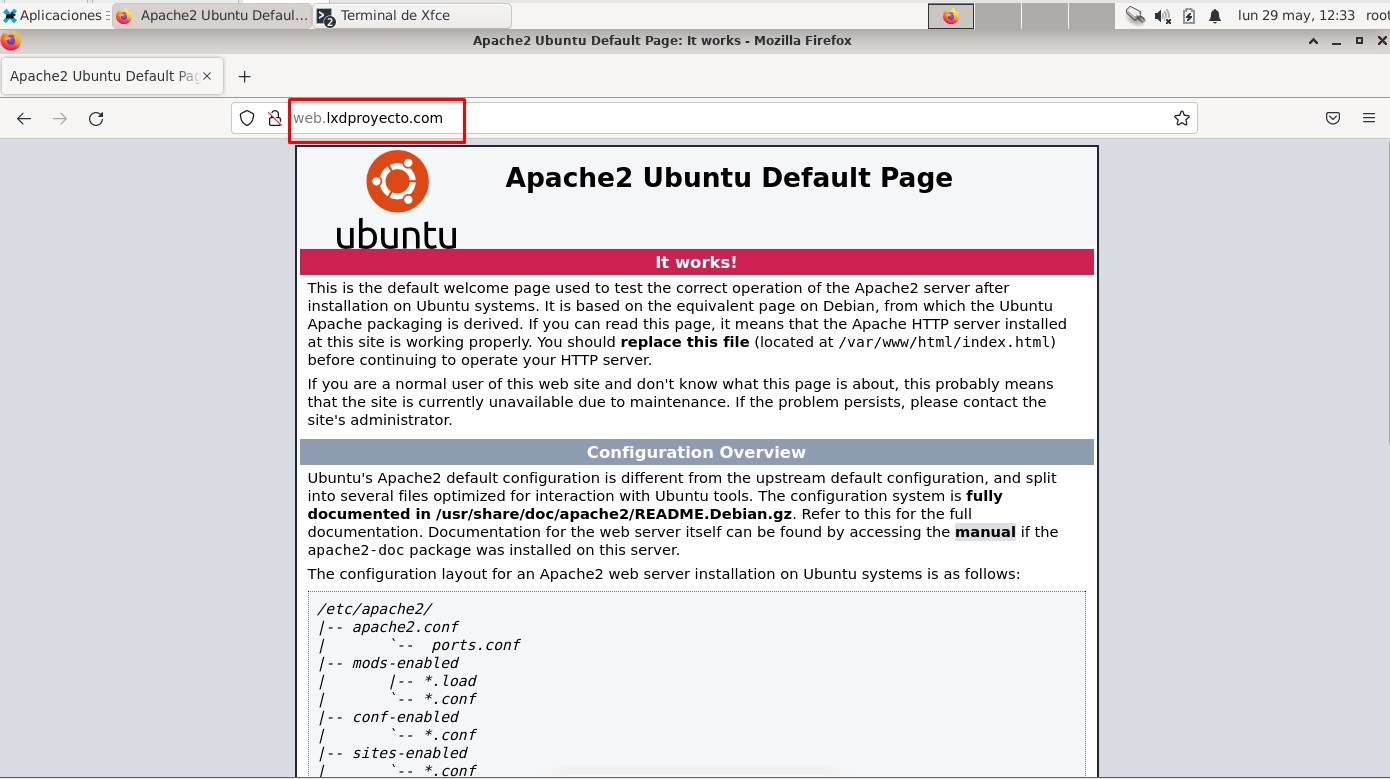


Ahora hacemos la prueba con la máquina prueba



Por lo que vemos el DNS resuelve, teniendo en cuenta que ftp realmente no existe todavía es normal que no tenga conectividad, pero nuestro objetivo funciona.

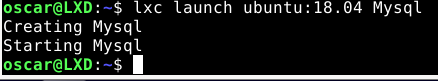
Ahora voy a configurarle a apache la dirección estática también, no pondré capturas porque no lo veo necesario, solo pondré captura de como se ve que se puede acceder por el nombre.



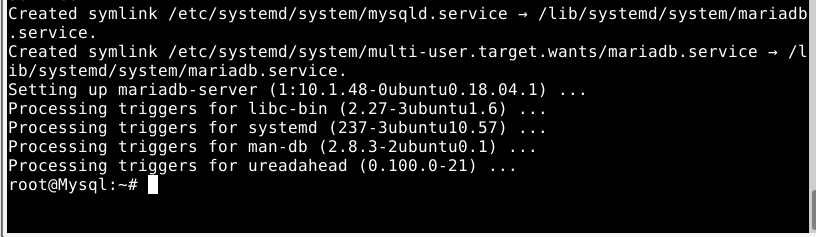
## MySQL + APACHE

### MariaDB

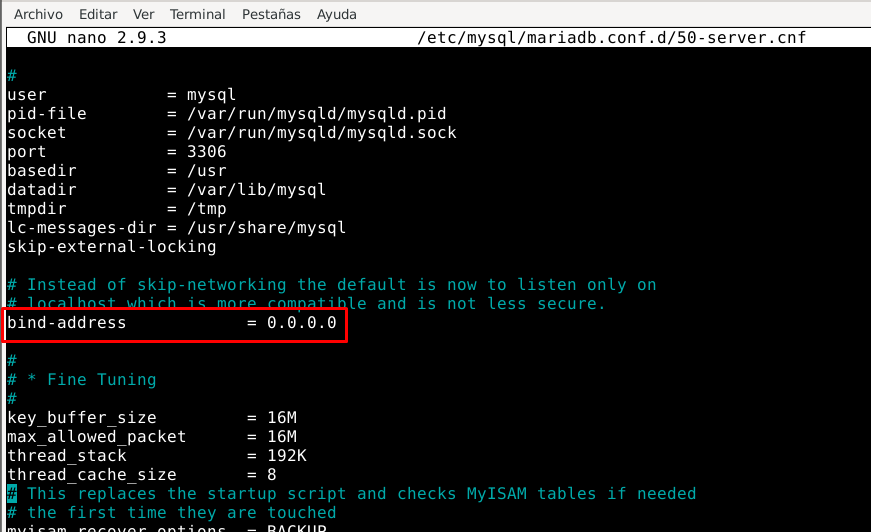
Vamos a crear un contenedor más para alojar MySQL.



Ahora vamos a meternos dentro del contenedor y actualizar los paquetes e instalar el paquete MariaDB server.



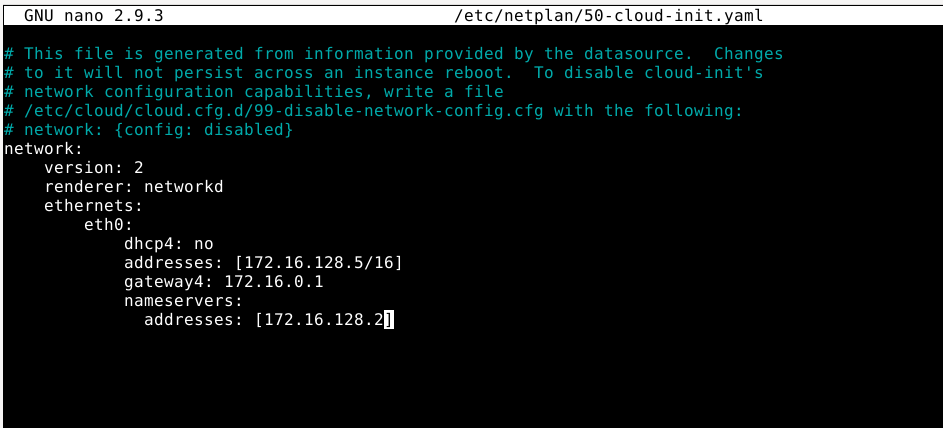
Una vez instalado vamos a tocar los campos de la configuración. Lo primero vamos a habilitar conexiones remotas a la base de datos.

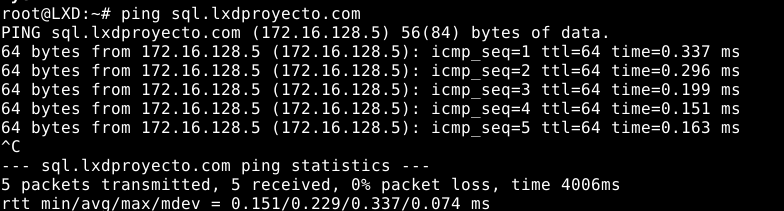


Lo dejaremos a 0.0.0.0 y ahora dentro de la base de datos daremos privilegios a root que se conecte de todos los hosts

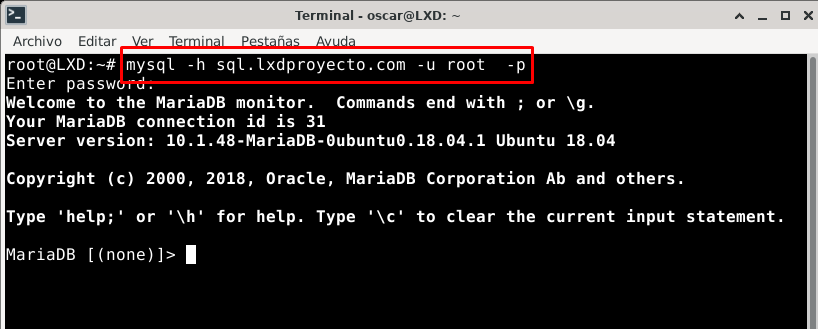


Ahora vamos a ponerle la dirección IP estática que le corresponde a nuestro servidor de MariaDB.





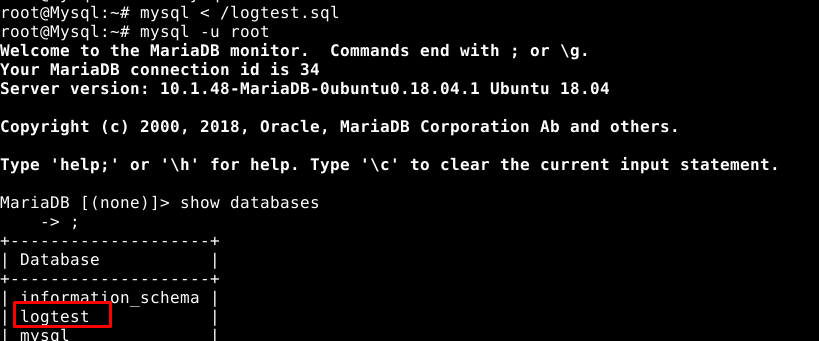
Como vemos hace ping y vamos a ver si podemos administrarlo desde fuera del equipo local.



Ahora le pasamos la base de datos del proyecto de GitHub de la página web que tengo hecha



Ahora la importamos.



Una vez importada ya está lista para usar

### Apache

Una vez que ya tengo el servidor de MySQL, voy a meter en apache mi página que hice para aplicaciones web, como un añadido al proyecto y un guiño a esa página que me costó tanto hacer, para ello vamos a meter el fichero en la máquina virtual.

El fichero lo descargué de mi GitHub previamente.



Así esta copiado en la ruta que he especificado de apache.

Ahora vamos a modificar los archivos de la página web para que haga las consultas en la base de datos que nosotros hemos añadido previamente.



Haremos eso con todos los archivos que requieran de una configuración de base de datos.

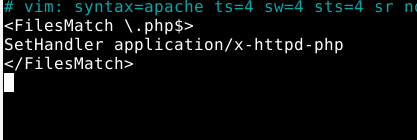
Una vez que hemos hecho esto nos hemos encontrado con un gran problema que tenemos que solucionar, la página en php no la puede procesar apache, se ve tal que así.



Lo que vamos a hacer va a ser crear un contenedor nuevo, en el que nosotros pondremos un nuevo servidor apache, pero instalando algunas dependencias



Una vez instaladas vamos a hacer ciertas cosas dentro del archivo de configuración /etc/apache/apache2.conf, las siguientes líneas irán dentro.



Una vez con eso habilitaremos y deshabilitaremos ciertos módulos:



Una vez que tenemos eso vamos a poner la página web en ese servidor.

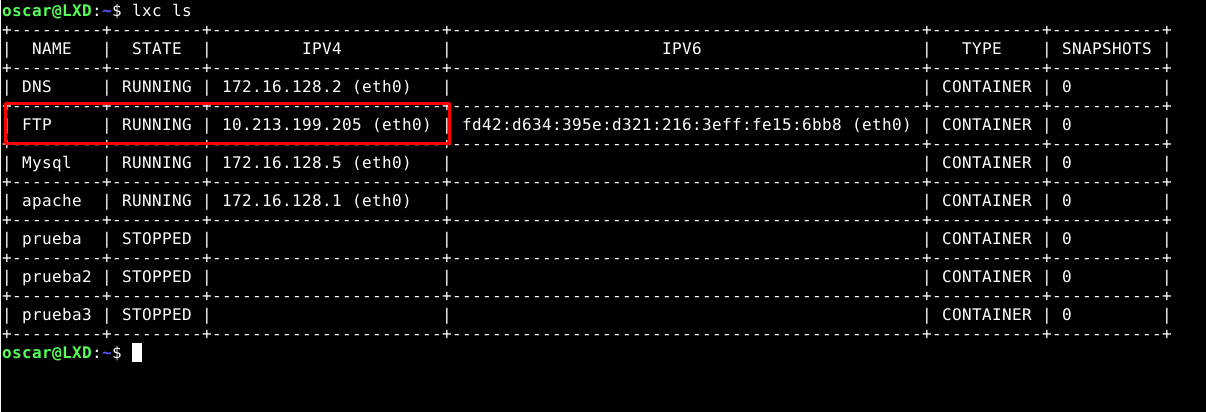
No entiendo muy bien porque no funciona el php de la página, pero el resto de servicios están completamente funcionales, la conexión con la base de datos se hace correctamente así que no sé muy bien de donde viene el problema, se pondrá como mejoría a proyecto.

## VSFTPD (FTP)

Vamos a tener un servidor FTP montado en un contenedor ya como último servicio que voy a aplicar en el trabajo para la mini demostración de red.

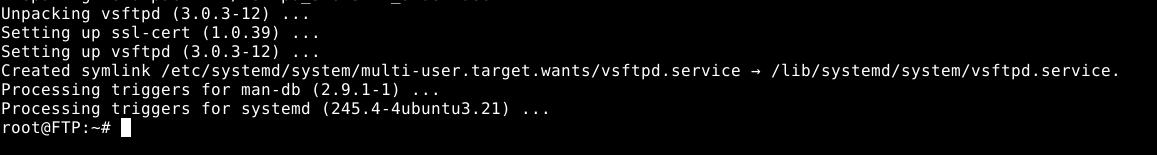
Primero vamos a crear la máquina en la que nosotros vamos a instalar el servidor FTP, a esta la vamos a poner con el sistema operativo Ubuntu 20.04 y le vamos a poner un limitador de CPU y de memoria para que solo llegue a usar 1 CPU y 320 MB de memoria RAM.



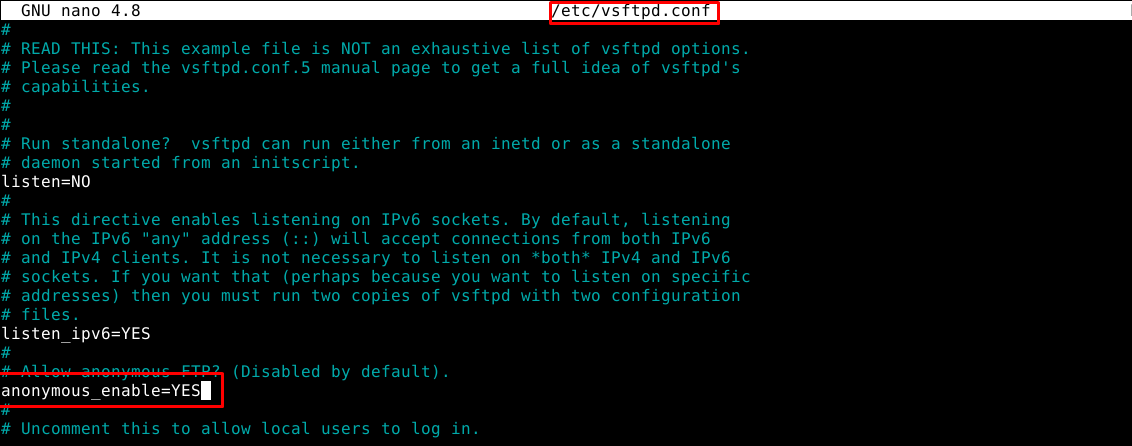


Ahora lo que haremos será entrar dentro del contenedor y configurar todo lo que sería nuestro servidor FTP.

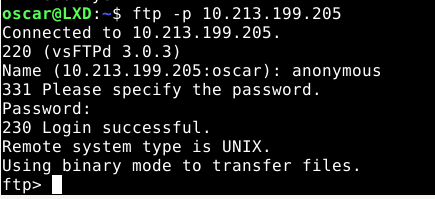
Vamos a instalar el servicio vsftpd que es el servicio ftp que es compatible con Linux.



Como solo vamos a hacer una prueba de que el ftp funciona y que se puede acceder desde cualquier equipo de nuestra red, lo que vamos a hacer es simplemente habilitar el modo para anónimos.



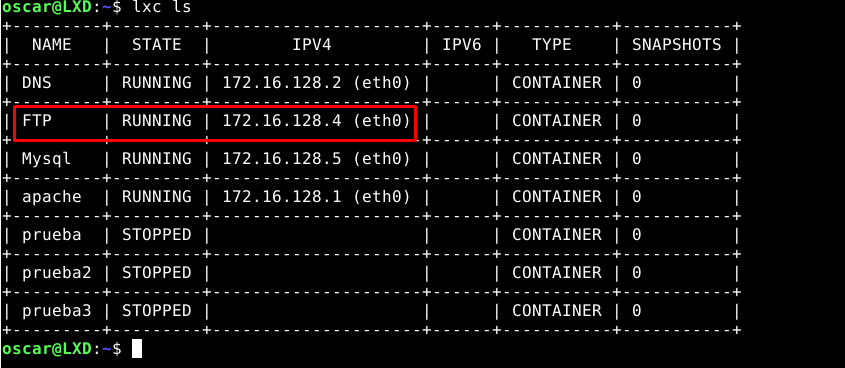
Este software autentica a los usuarios con usuarios de sistema de manera que si tú estás iniciado sesión en tu Linux con un usuario con la misma contraseña que el que está autorizado en el ftp simplemente te podrás conectar a ello.



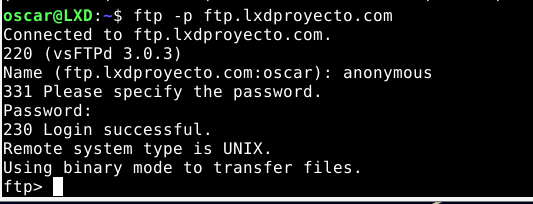
Como vemos se puede entrar con Anonymous vamos a probar a cambiar la ip y lo haremos que forme parte de la red.



Ahora le ponemos la dirección de red estática.



Una vez hecho eso vamos a comprobar si resuelve con el nombre de domino

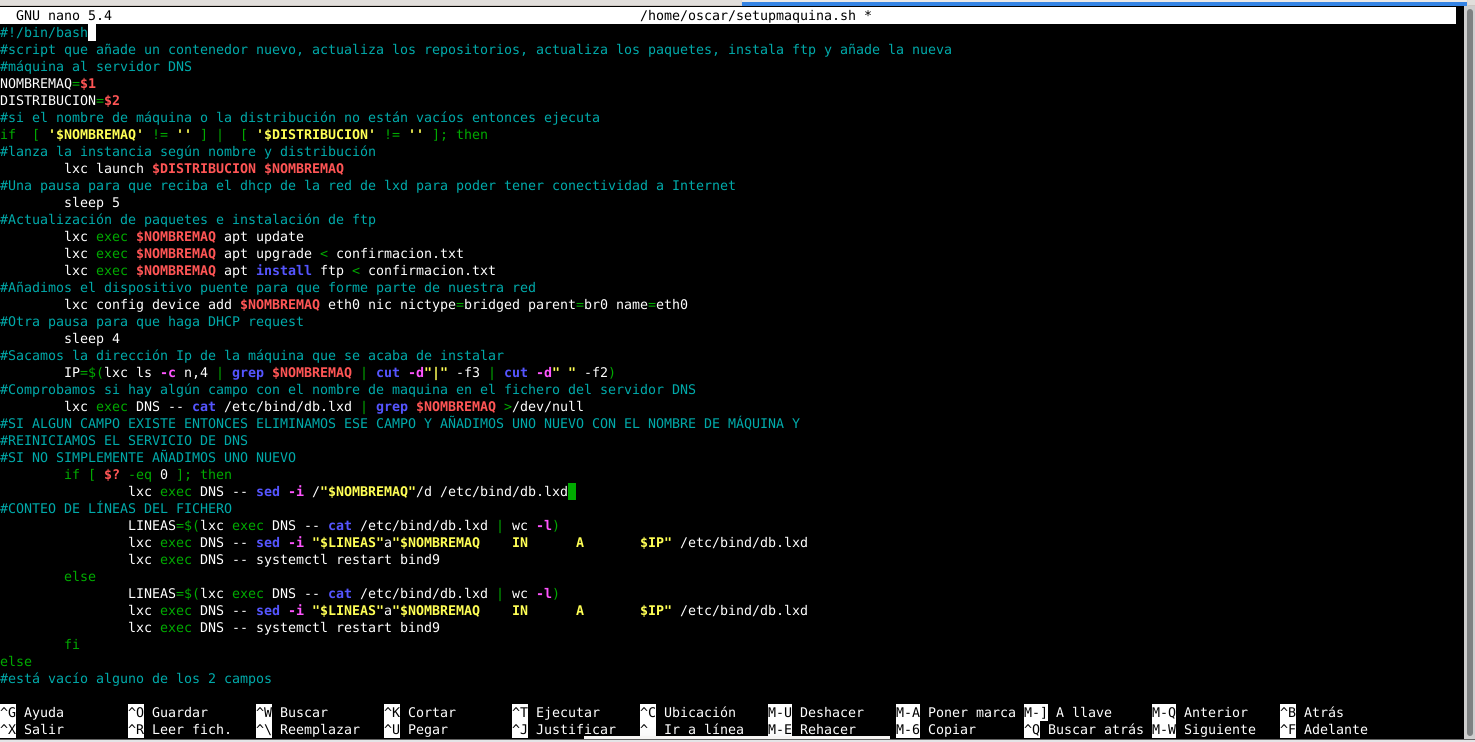


Efectivamente es capaz de resolver por nombre de proyecto.

## Automatización de creación de máquinas host

Vamos a crear un script que nos monte una máquina ya configurada con dhcp y con el servicio ftp instalado.

Simplemente combinaremos los comando y ciertas técnicas utilizadas en scripting de bash para poder crear un script de automatización de la instalación de una instancia.

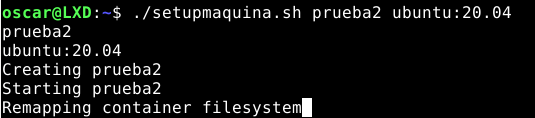


Como es muy difícil de leer a menos de que le pongas zoom dejaré un fichero en el github y en la propia entrega para que no sea muy complicado la lectura en la imagen. Si queréis mirarlo desde aquí simplemente hacer mucho zoom.

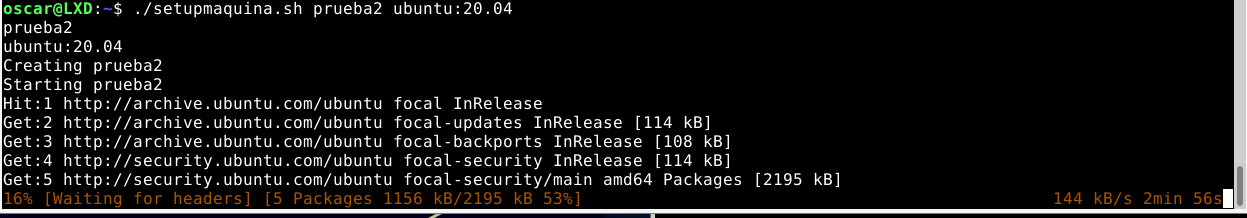
Ese es el script que le pasamos como parámetros el nombre que va a tener la máquina y el tipo de distribución de la que va a coger una imagen y el resto simplemente le actualiza los repositorios y los paquetes que tiene instalados, una vez que eso está ya configurado instala el servicio de ftp para poder conectarse al ftp que tenemos creado y le añade la interfaz puente a la máquina para que sea capaz de coger dirección IP de la red y poder trabajar con los DNS de la red, aparte añade su dirección ip al servidor DNS



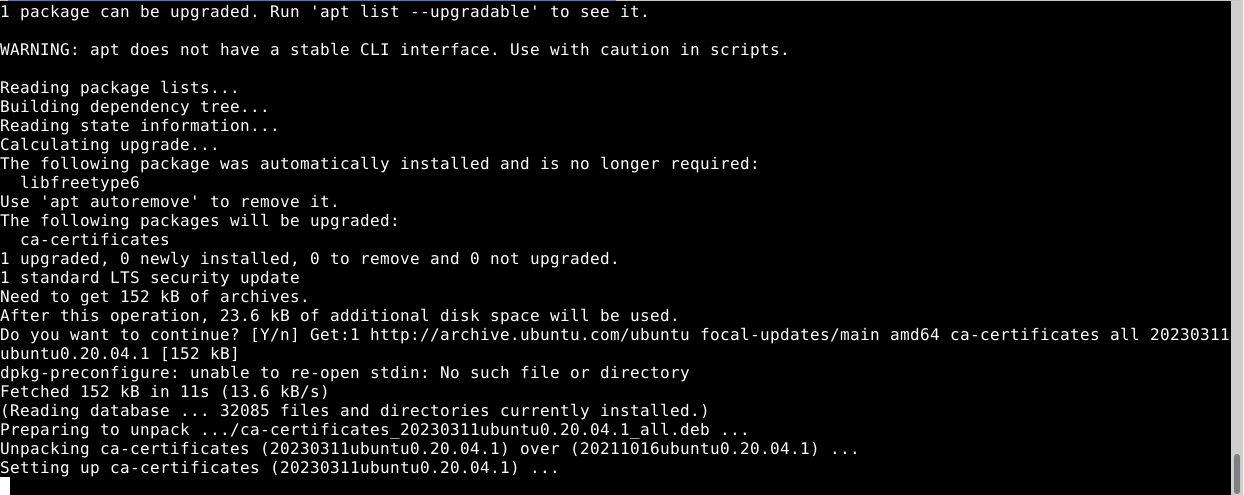
Ahora empieza a trabajar.



Actualiza repositorios

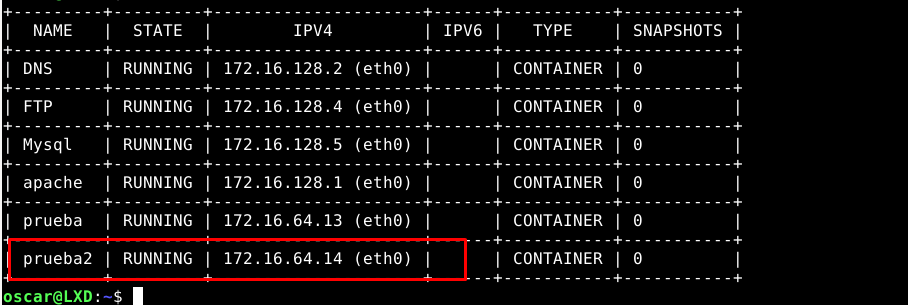


Actualiza los paquetes ya instalados

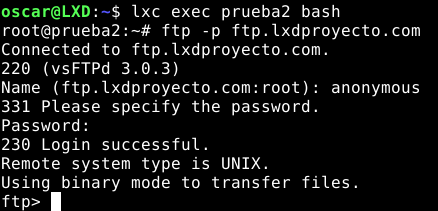


Instala FTP como servicio y pone el adaptador de red conectado a la máquina.

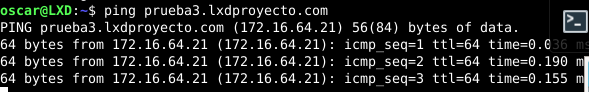


Si hacemos lxc ls podremos ver que la máquina está perfectamente configurada. 

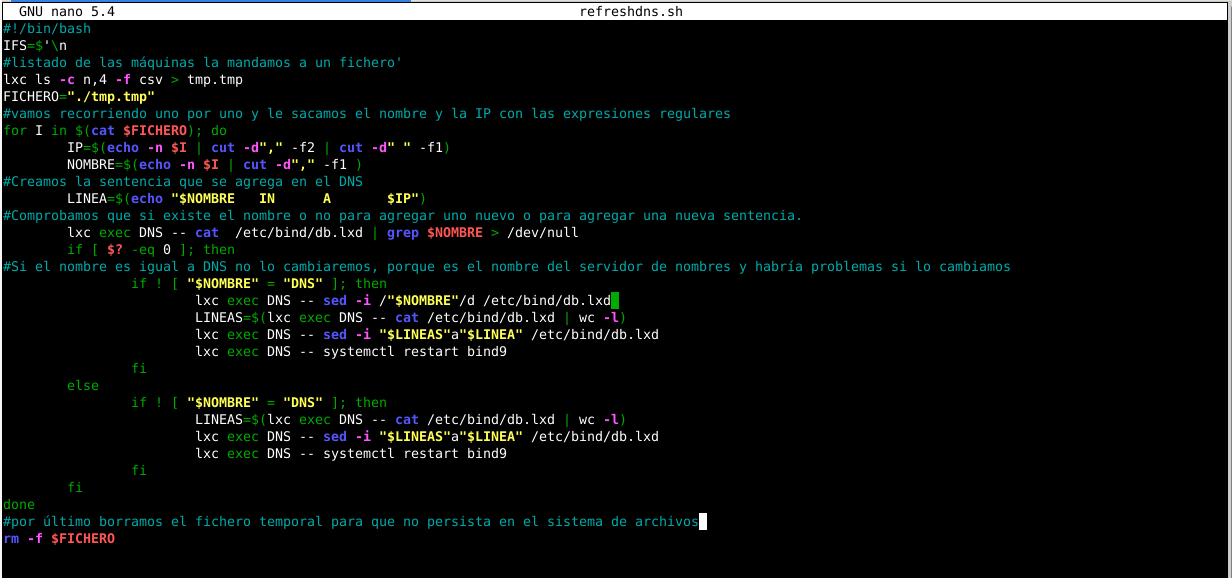
Y podemos acceder desde ella al servidor FTP por ejemplo



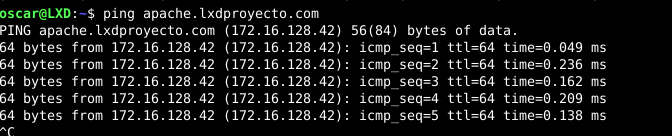
He añadido otra máquina llamada prueba 3 con el script ahora vamos a probar a hacerle ping por nombre.



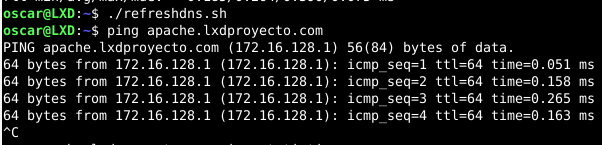
Y por último ya para que sea más cómodo aún como las máquinas están por dhcp cogerán dirección IP dinámica por lo cual el nombre DNS referente a la dirección IP puede cambiar lo que significa que tenemos que hacer otro script que sirva para refrescar el DNS dependiendo de las máquinas que tengamos de contenedores, básicamente el script es así.



Y con ese fichero incluso si cambiamos la dirección de uno de los estáticos simplemente usaremos el script y se pondrá automáticamente, por ejemplo, yo he cambiado la dirección estática de apache que era la 172.16.128.1 y la he cambiado a la 43, ejecuto el script y se refresca sola y ahora el ping se hace bien.

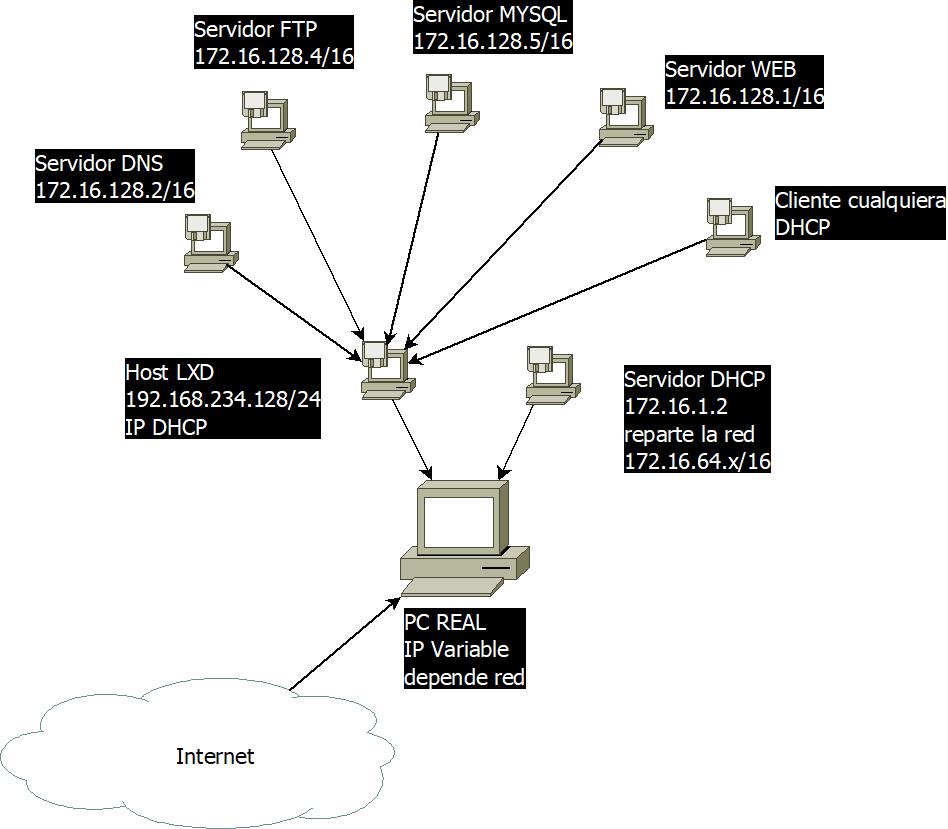


Si lo volvemos a cambiar al 1 y usamos el script vemos que funciona



## Esquema de red final

El esquema de red se nos quedaría de la siguiente manera.



Tenemos nuestra máquina real que está conectada a internet y su dirección va por dhcp luego un servidor dhcp que es otra máquina Debian con una ip estática dentro del segmento de red 172.16.0.0/16 y va a repartir la dirección ip del resto de máquinas y luego nuestro host LXD con un adaptador que hace NAT con nuestro PC REAL dándole conectividad a internet y el adaptador de red que conecta con nuestra red local. Y por último los contenedores servidores con IP estática y los clientes que tendrán sus direcciones por DHCP

# Problemas encontrados

* No era posible ejecutar el lxd init: la solución añadir al $PATH lo siguiente: “/snap/bin”
* No se puede conectar desde la máquina real ni una del mismo segmento de red a un contenedor: la solución fue hacer una redirección de direcciones y puertos con un proxy gracias a una utilidad de LXD
* No se podía configurar la red del contenedor Debian: la solución fue cambiarlo desde otro sitio que no era el /etc/network/interfaces, (/etc/systemd/network/eth0.network)
* No se puede agregar más de una redirección de puerto por proxy de la dirección 0.0.0.0: no tiene solución así que tuve que buscar una alternativa para poder acceder a la red de manera más eficiente, solucionado gracias a la creación de una interfaz puente que deja pasar el tráfico de red entre 2 redes.
* El servidor apache no es capaz de procesar código en php: no solucionado.
* No dejaba crear máquinas virtuales dentro del Debian porque dice que no es compatible con la virtualización: arreglado habilitando la virtualización anidada dentro de la máquina de VMware.

# Posibles mejoras

* Hacer que los equipos sean parte de la red local, ya que los contenedores solo son visibles entre sí y la máquina host de los contenedores. Esta mejora la he conseguido gracias a que pude hacer un puente entre las 2 interfaces para que se vieran en capa 2 y por ende también en capa 3
* Crear toda la estructura con nombres de dominio, así no tendría que usar ninguna dirección IP, esto se ha conseguido gracias a la implementación del servidor DNS y que sea automático gracias al DHCP que reparte el servidor DNS de manera automática.
* Conseguir implementar la página web que ya tenía de IAW con un servidor MySQL y apache. He conseguido implementar la página web, pero el php no funciona.
* Añadir un servidor FTP que funcione perfectamente. Esto se ha conseguido gracias a vsftpd
* Automatización de creación de las instancias e incluirlas en la red local aisladas de internet.

# Conclusión

La conclusión que he podido sacar con este proyecto es que LXD es un software tan amplio y tan extenso que no se pueden cubrir todos los puntos ya que no acabaría nunca, te da muchas posibilidades de hacer estructuras de red ya venga siendo para hacer un home server en tu casa para guardar archivos con un ftp un dns y un dhcp o si simplemente tienes Linux y quieres virtualizar otros Linux para tener más máquinas virtuales.  
He reflejado la cantidad de posibilidades que este software te puede brindar y el aprendizaje de redes que he tenido que adoptar ha sido muy superior a mi nivel inicial, ya que no sabía si quiera hacer un puente y espero que este trabajo pueda servir de guía a otro novato en esto de los contenedores LXD como yo lo fui cuando empecé a hacerlo.

# Enlaces y referencias de la documentación (Bibliografía)

La información la he sacado principalmente de la página principal de LXD y de ciertas páginas externas. Aquí los enlaces a las páginas:

<https://linuxcontainers.org/lxd/introduction/>

<https://www.cyberciti.biz/faq/how-to-install-lxd-on-debian-11-linux/>

<https://www.youtube.com/watch?v=wrUG3ioyS6M>

<https://www.youtube.com/watch?v=fZveDIePeSg>

<https://www.youtube.com/watch?v=mu34G0cX6Io>

<https://images.linuxcontainers.org>

[Systemd: Setup a static IP Address (michlstechblog.info)](https://michlstechblog.info/blog/systemd-setup-a-static-ip-address/)

[MariaDB Server Documentation - MariaDB Knowledge Base](https://mariadb.com/kb/en/documentation/)

[Netplan static IP on Ubuntu configuration - Linux Tutorials - Learn Linux Configuration](https://linuxconfig.org/how-to-configure-static-ip-address-on-ubuntu-18-04-bionic-beaver-linux)

[isc-dhcp-server - Community Help Wiki (ubuntu.com)](https://help.ubuntu.com/community/isc-dhcp-server)

[BIND 9 Administrator Reference Manual — BIND 9 9.18.8 documentation](https://bind9.readthedocs.io/en/v9.18.8/)

<https://ubuntu.com/blog/lxd-virtual-machines-an-overview>

<https://discuss.linuxcontainers.org/t/cannot-create-instance-type-virtual-machine/11630>