Departamento de
Automática y Computación
*Automatika eta*
*Konputazio Saila*

Campus de Arrosadía
*Arrosadiko Campusa*
31006 Pamplona - *Iruñea*
*Tfno. 948 169113, Fax. 948 168924*
*Email: ayc@unavarra.es*

# Virtualización con Contenedores

## 1. Introducción y objetivos

El objetivo de esta prácticas en tomar un contacto básico con soluciones para el despliegue y configuración de contenedores en Linux, en especial la parte de red.
Crearemos pequeños escenarios de red donde las máquinas virtuales serán contenedores y se encontrarán en diferentes hosts. Interconectaremos dichos contenedores mediante varias alternativas topológicas diferentes.
Se probarán en concreto contenedores con LXD y con Docker, pero en ningún momento se pretende llevar a cabo una evaluación de todas sus posibilidades y opciones de configuración.

## 2. Contenedores con LXD

Las herramientas para gestión de contenedores suelen incluir opciones para la operación sobre la red de interconexión de los mismos. En esta práctica reduciremos al mínimo la gestión de la red a través de perfiles y otras opciones de automatización y la llevaremos a cabo de forma manual para tener más claro su funcionamiento. Crearemos varios escenarios de comunicación entre contenedores.
En vez de crear contenedores directamente en la instalación de Linux de las máquinas del laboratorio crearemos máquinas virtuales Linux en VirtualBox y será ahí donde crearemos los contenedores. Esto también permite que hagan gran parte de la práctica en sus propios ordenadores, aunque no corran Linux nativo, con tal de que puedan instalar VirtualBox (u otro hypervisor).
Cree dos máquinas virtuales en VirtualBox que actuarán como los hosts en los que se configuren los contenedores. Instale en ellas por ejemplo Ubuntu Server[1], emplee un disco local y no su *home* para guardarlas. A partir de aquí, en esta práctica se llamará "host" a esas máquinas virtuales que hospedan a los contenedores; si se desea hacer referencia al host físico en el que corre VirtualBox se hará específicamente para distinguirlo de los otros. Esas máquinas virtuales deberán tener un interfaz en una red interna por la que puedan comunicarse en capa 2. Pueden crear una sola máquina y después clonarla (cambie las direcciones MAC de los interfaces al clonarla).
En las máquinas virtuales permita acceso a modo promiscuo para el tráfico entre las máquinas virtuales en el interfaz asignado a la red interna.

### *Instalación de LXD*

Puede consultar la web de LXD[2] para tener una idea general sobre el proyecto y se recomienda su tutorial básico sobre instalación y creación de contenedores[3].
Al hacer la configuración inicial del demonio *lxd* en cada host (sudo lxd init) se recomienda:

> *Name of the storage backend to use (dir or zfs):* dir
> *Would you like LXD to be available over the network (yes/no)?* no
> *Do you want to configure the LXD bridge (yes/no)?* no

[1] https://www.ubuntu.com/server
[2] https://linuxcontainers.org
[3] https://linuxcontainers.org/lxd/getting-started-cli/

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Departamento de
Automática y Computación
*Automatika eta*
*Konputazio Saila*

Campus de Arrosadía
*Arrosadiko Campusa*
31006 Pamplona - *Iruñea*
*Tfno. 948 169113, Fax. 948 168924*
*Email: ayc@unavarra.es*

***Escenario puenteado a través de LAN***

En este escenario tendremos dos hosts, cada uno de ellos con contenedores configurados, tal que todos esos contenedores estén en la misma LAN, que será la LAN de los hosts. Se ha intentado representar esto en la Figura 1.

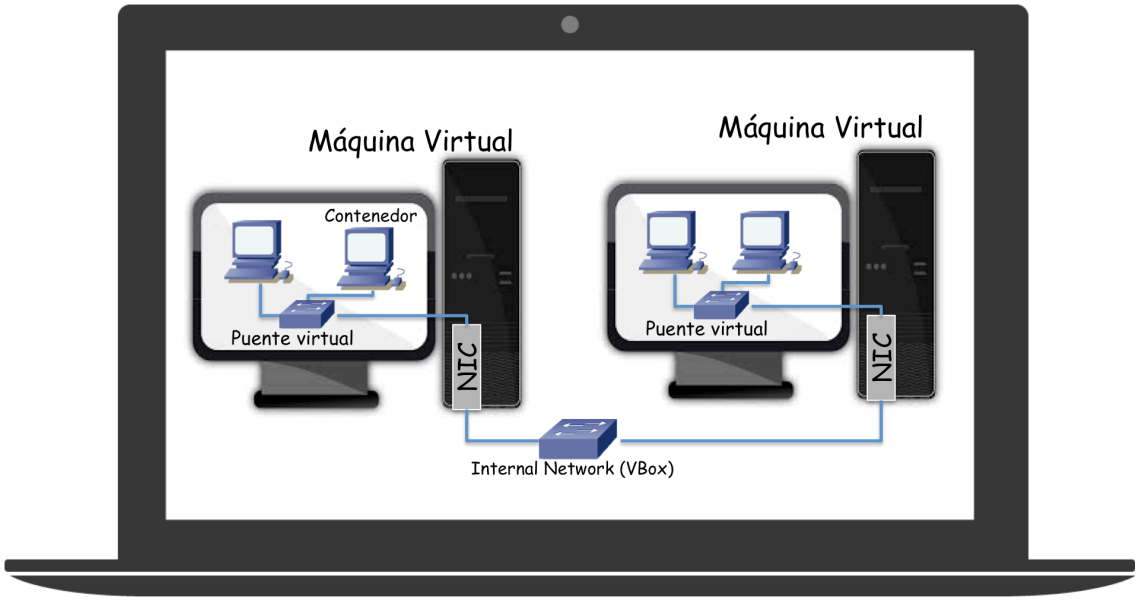


**Figura 1 - Escenario puenteado a través de LAN**

Verá que LXD ha creado un puente en el host. Normalmente se llama lxdbr0. Un puente en el kernel de Linux nos permite por ejemplo construir un puente con un PC que tenga múltiples interfaces físicos, aunque en este caso lo que vamos a hacer es que el puente tenga un interfaz físico (la NIC del host, que en nuestro caso en realidad es virtual, dada por VirtualBox) e interfaces virtuales que lo conectan a los contenedores.

Los puentes en Linux se gestionan con el comando *brctl*. Puede consultar su documentación[4] para más detalle. Ahí verá cómo de sencillo es crear un puente o incluir un interfaz en el mismo. Si no tiene ese comando en su host deberá instalar las *bridge utilities*.

En uno de los hosts añada al puente creado por LXD el interfaz físico de la internal network. Por ejemplo, si su interfaz se llama eth0 el comando sería:

*# brctl addif lxdbr0 eth0*

Puede ver los interfaces que hay en el puente con el comando:

*# brctl show lxdbr0*

A continuación lance un contenedor en ese host (por ejemplo una Ubuntu, pero no tiene mucha importancia la distribución de Linux en concreto).

Si ahora mira los interfaces que hay conectados a ese puente encontrará un nuevo interfaz. Este es un interfaz virtual creado para comunicar al host con el contenedor. Lo que ha creado el host (debido a la plantilla por defecto de creación de contenedores en LXD) es un par de interfaces de tipo veth los cuales son interfaces lógicos emparejados, tal que lo que se envía por uno se recibe por el otro. Uno de esos dos interfaces es el que está viendo en el host (lo verá con el comando *ifconfig*) y el otro se ha movido al *network*

[4] https://wiki.linuxfoundation.org/networking/bridge

Campus de Arrosadía
*Arrosadiko Campusa*
31006 Pamplona - *Iruñea*
*Tfno. 948 169113, Fax. 948 168924*
*Email: ayc@unavarra.es*

*namespace*[5][6] empleado por el contenedor. Si crea una Shell en el contenedor podrá ver el interfaz parejo mediante el comando *ifconfig* o el comando *ip*.
Repita la configuración en el segundo host. Asigne dirección IP a los contenedores que se encuentren en los dos hosts, todas en la misma subred IP. Debería poder hacer ping de un contenedor a otro siguiendo el escenario puenteado tal y como se describe en la Figura 1. Puede ver el tráfico con *tcpdump,* por ejemplo en uno de los hosts, y verá las tramas Ethernet con las direcciones MAC de los interfaces virtuales de los contenedores. Esto es tráfico que está circulando por la internal network creada en VirtualBox. En un escenario donde los hosts no son máquinas virtuales sino el host nativo de un servidor este tráfico estaría circulando por uno de los conmutadores hardware de nuestra infraestructura de red (elimine de la Figura 1 el portátil que engloba a todo el escenario y se ha quedado con un switch hardware con dos PCs que corren los contenedores).
En este punto no hemos configurado dirección IP a los hosts (las máquinas virtuales en VirtualBox). Si quisiéramos darles dirección IP deberíamos asignarla al interfaz del puente (lxdbr0), dado que el interfaz físico está conectado al mismo. Esto es similar a lo que hacemos en un switch Cisco capa 2/3 donde no asignamos dirección IP al interfaz físico sino que lo hacemos al interfaz lógico de la VLAN porque el interfaz físico está funcionando en modo conmutación en capa 2 (en muchos switches capa 2/3 se puede cambiar el modo de funcionamiento de ese interfaz para que pase a ser un interfaz capa 3, que sería como si sacáramos nuestro interfaz del host del puente).

**Punto de control:** Muestre el escenario en funcionamiento al profesor.

***Escenario enrutado***

Este escenario está representado en la Figura 2. El cambio en el dibujo respecto al escenario puenteado es sutil. Ahora cada host actúa como un router, de forma que sus contenedores están en una subred IP diferente de los contenedores del otro host. La interconexión entre los hosts la hemos mantenido con una internal network de VirtualBox (es decir, en capa 2) pero valdría cualquier conectividad en capa 3 (es decir, podría haber routers entre los hosts). A nivel lógico capa 3 las subredes existentes se ven en la Figura 3.

[5] http://man7.org/linux/man-pages/man8/ip-netns.8.html
[6] http://man7.org/linux/man-pages/man7/namespaces.7.html

Departamento de
Automática y Computación
*Automatika eta*
*Konputazio Saila*

Campus de Arrosadía
*Arrosadiko Campusa*
31006 Pamplona - *Iruñea*
*Tfno. 948 169113, Fax. 948 168924*
*Email: ayc@unavarra.es*

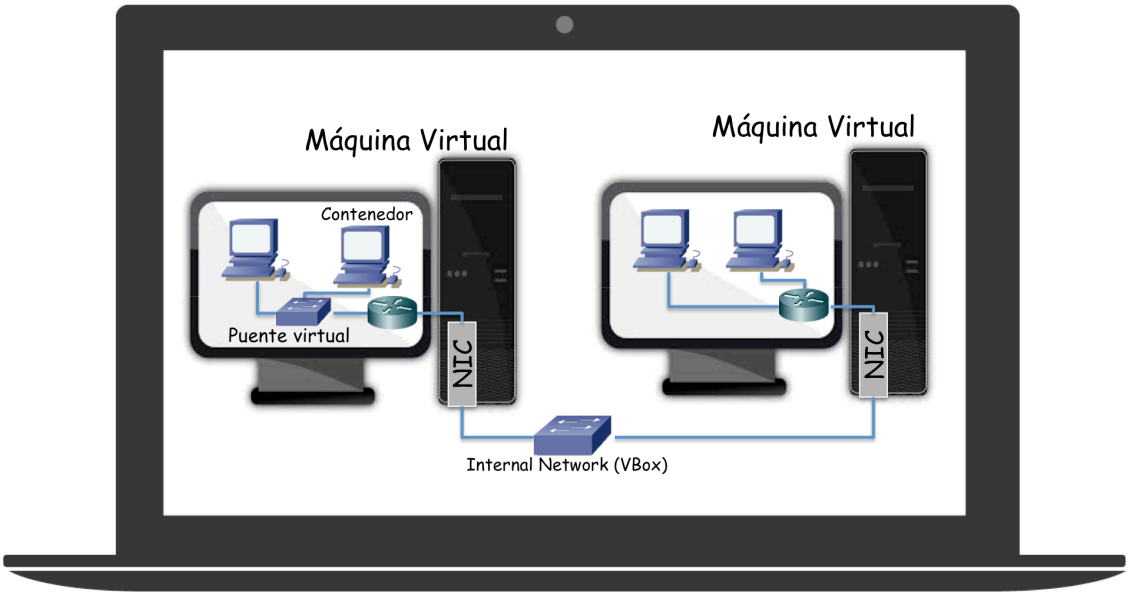


**Figura 2 - Escenario enrutado**

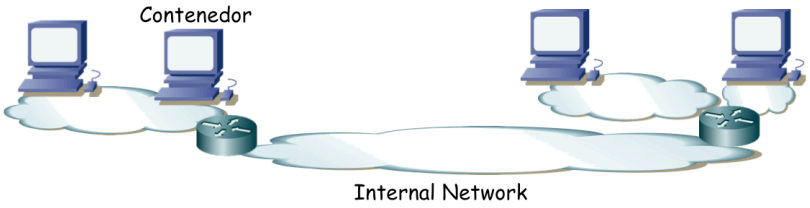


**Figura 3 - Topología capa 3 del escenario enrutado**

Partiremos de los hosts sin la configuración del escenario puenteado.
Asigne dirección IP a los interfaces físicos de los hosts en la internal network en una subred propia para ella. Deberían alcanzarse el uno al otro (es decir, que pueden hacerse ping). Los hosts van a enrutar así que hay que activar el reenvío de paquetes IP en ellos. Por simplicidad pondremos una ruta por defecto de cada uno apuntando al otro.
En uno de los hosts configuramos en el puente la dirección IP que queremos que tenga el router en la subred de los contenedores. Así mismo a sus contenedores en ese puente les damos dirección IP de esa misma subred y ponemos como siguiente salto en su ruta por defecto la dirección IP del bridge.
En el otro host podríamos hacer una configuración similar y tendríamos resuelto el escenario pero vamos a hacerla ligeramente diferente para practicar con el entorno virtualizado. No vamos a utilizar el bridge. Saque el interfaz del contenedor del bridge (*brctl delif*). A continuación, en el host, asigne dirección IP a ese interfaz. Ese es el interfaz del router virtual de un enlace punto a punto con el contenedor así que puede emplear una subred con prefijo de 30 bits. Ahora dentro del contenedor configure la dirección IP del otro extremo de ese veth con la otra dirección IP de la subred y la ruta por defecto apuntando a la dirección IP del router.
Con eso ya debería funcionar el ping entre los contenedores de los dos hosts. Con *traceroute* o con *ping –R* podrá ver los saltos que dan los paquetes. También puede observarlos con *tcpdump* en los hosts y ver las direcciones MAC que aparecen en esas tramas.

Departamento de
Automática y Computación
*Automatika eta*
*Konputazio Saila*

Campus de Arrosadía
*Arrosadiko Campusa*
31006 Pamplona - *Iruñea*
*Tfno. 948 169113, Fax. 948 168924*
*Email: ayc@unavarra.es*

En la Figura 2 aparece un segundo contenedor en el segundo host. Si ha creado un segundo contenedor tendrá que sacar también su interfaz del puente de LXD y configurar una nueva subred IP para ese enlace.

**Punto de control:** Muestre el escenario en funcionamiento al profesor.

***Escenario Ethernet sobre GRE***

En este caso vamos a extender la LAN capa 2 entre los hosts para que los contenedores se encuentren todos en la misma LAN/subred IP pero lo vamos a hacer sin puentearla con la LAN externa a los hosts. Lo que haremos será introducir las tramas Ethernet en un túnel GRE para hacerlas llegar de un host al otro.

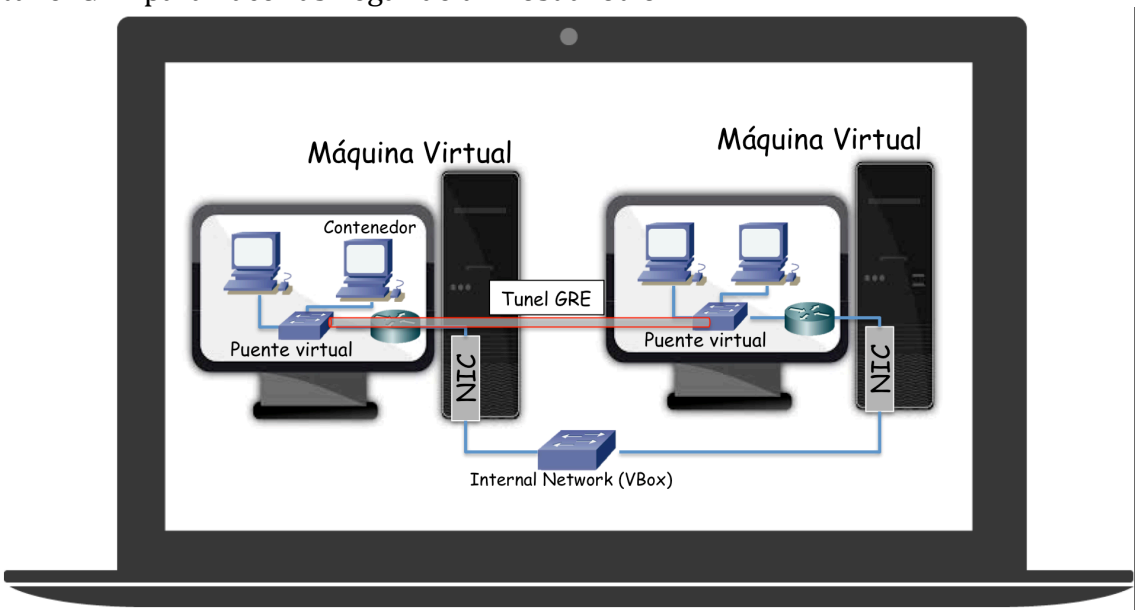


**Figura 4 - Escenario Ethernet sobre GRE**

Muchos túneles GRE se emplean para transportar IP sobre IP, pero eso no impide transportar otros protocolos sobre GRE y Ethernet es uno de ellos.

Empezamos de nuevo con los hosts sin configurar.

Dado que la comunicación entre los hosts es con paquetes IP quiere decir que de nuevo podría haber routers entre ellos (aunque en este ejemplo tengamos solo la internal network) y también significa necesitamos asignar dirección IP a los interfaces externos de los hosts. Asigne direcciones a los interfaces de los hosts para que haya conectividad IP entre ellos. Para el resto de ejemplo supondremos que un host es 172.16.1.1 y el otro 172.16.1.2.

En cada host hay que crear un interfaz virtual que haga referencia al túnel GRE. El tipo de interfaz en el Kernel es “gretap” y lo crearíamos de la siguiente forma:

*# ip link add eltunelgre type gretap remote 172.16.1.2 local 172.16.1.1*

Evidentemente en el otro host las direcciones IP en el comando estarán en el orden contrario, dado que determinan los extremos local y remoto del túnel.

Con esto se habrá creado un nuevo interfaz de nombre *eltunegre*. Levante el interfaz y añádalo al bridge de LXD en cada host.

No hace falta activar el reenvío de paquetes IP en los hosts pues no van a reenviar paquetes IP. El puente virtual actúa en capa 2 y uno de sus interfaces será el túnel. Ese interfaz especial introduce la trama Ethernet que se le entraga en un paquete IP dirigido

Campus de Arrosadía
*Arrosadiko Campusa*
31006 Pamplona - *Iruñea*
*Tfno. 948 169113, Fax. 948 168924*
*Email: ayc@unavarra.es*

al otro extremo del túnel. En ningún momento el host está haciendo *forwarding* de un paquete IP por una tabla de rutas.
Ahora configure la dirección IP del interfaz de cada contenedor de forma que estén en la misma subred aunque estén en hosts diferentes. Con eso ya deberían alcanzarse el uno al otro sin saltos IP enrutados aparentes y las tramas Ethernet se transportarán sobre GRE si las miramos con *tcpdump* en el interfaz de uno de los hosts.

**Punto de control:** Muestre el escenario en funcionamiento al profesor, así como una traza de tráfico con paquetes con encapsulado GRE.

***Escenario Ethernet sobre VXLAN***

En este caso configuraremos un escenario muy similar al anterior pero ahora la comunicación de las tramas Ethernet entre los hosts se hará sobre UDP, en concreto empleando VXLAN. Esta solución emplea un grupo multicast para el tráfico BUM (Broadcast, Unknown unicast, Multicast, más información en los materiales de teoría).
Empezamos de nuevo con los hosts sin configurar.
Dado que la comunicación entre los hosts es con paquetes IP quiere decir que de nuevo podría haber routers entre ellos (aunque en este ejemplo tengamos solo la internal network) y también significa que necesitamos asignar dirección IP a los interfaces del host. Asigne direcciones a los interfaces de los hosts para que haya conectividad IP entre ellos.
Creamos el interfaz encargado del tráfico VXLAN:

> *# ip link add vxlan0 type vxlan id <VNI> group <ip_multicast> dev <interfazInternalNetwork> dstport 4789*

donde el significado de los parámetros es:

> *"id <grupoDePracticas>"* : VXLAN permite crear LANs sobre UDP. Cada LAN necesita su identificador numérico que se llama el VNI (VXLAN Network Identifier) y es un número de 24 bits (muchos más que las VLANs de 802.1Q)
> *"group <ip_multicast>"* : VXLAN emplea un grupo multicast. Seleccione uno dentro del rango privado (239.0.0.0/8).
> *"<interfazInternalNetwork>"* : El nombre del interfaz (ej: eth0) conectado a la internal network (por donde podrá comunicarse con el resto de hosts con contenedores de esta VXLAN)

El valor del puerto 4789 es el reservado por IANA para VXLAN[7].
Añada el interfaz VXLAN creado (en este caso lo hemos llamado vxlan0) al puente de LXD y configure la dirección IP de un contenedor.
Repita la configuración en el otro host, asignando al contenedor una dirección IP en la misma subred que el otro. Prueba a comunicar uno con otro (un ping por ejemplo) y vea en uno de los hosts el intercambio de paquetes en el interfaz en la internal network. Es interesante que lance *tcpdump* antes de empezar la comunicación para poder ver los paquetes ARP sobre el grupo multicast.

**Punto de control:** Muestre el escenario en funcionamiento al profesor, así como una traza de tráfico con paquetes con encapsulado VXLAN.

## 3. Contenedores con Docker

Para completar esta práctica se plantea un ejercicio más abierto. Fundamentalmente se desea comparar la solución de contenedores basada en LXD con la solución de Docker[8].

[7] https://tools.ietf.org/html/rfc7348

Departamento de
Automática y Computación
*Automatika eta*
*Konputazio Saila*

Campus de Arrosadía
*Arrosadiko Campusa*
31006 Pamplona - *Iruñea*
*Tfno. 948 169113, Fax. 948 168924*
*Email: ayc@unavarra.es*

Docker ofrece gran cantidad de documentación y formas sencillas de probarlo incluso en sistemas operativos macOS y Windows. La recomendación es acudir a la documentación y en especial a los tutoriales disponibles en su web para tener una idea de la filosofía detrás de Docker y de cómo emplearlo. Existen muchos más recursos en Internet sobre Docker, por ejemplo en Katacoda[9]. Por supuesto muchos tutoriales estarán orientados a herramientas para la gestión de los contenedores y no a llevar a cabo manualmente todos los pasos como se ha hecho en esta práctica para LXD.

Lo que se le pide en este apartado es un breve documento (no más de 4 páginas) explicando las diferencias entre las dos soluciones y algunos casos que haya probado. Nos interesan principalmente los aspectos relacionados con el *networking* entre contenedores. Recuerde poner referencias para todas las afirmaciones que haga y que no sean por su propia experiencia.

[8] https://www.docker.com/
[9] https://www.katacoda.com/courses/docker