

Virtualización con VirtualBox

1. Introducción y objetivos

El objetivo de esta práctica es familiarizarse con los entornos de virtualización, en concreto con algunas funcionalidades de VirtualBox similares (aunque de forma simplificada) a lo que se puede emplear en un entorno de servidor en el centro de datos. Practicaremos con la posibilidad de crear redes internas donde algunos hosts virtuales puedan actuar por ejemplo como routers o servidores (o balanceadores, o firewalls, o proxies, o IDS...), así como con el movimiento de máquinas virtuales entre hosts.

Esta práctica no pretende ser un tutorial paso-a-paso. Se plantean los objetivos de cada apartado pero puede haber múltiples formas de alcanzarlos, incluso se puede llevar a cabo la práctica en plataformas diferentes (Linux del laboratorio, macOS, Windows). Al final del guión de la práctica se incluyen anotaciones, pistas y consejos que pueden ayudar ante ciertas dificultades, aunque todo se puede resolver con un poco de trabajo y buscando documentación.

2. Networking virtual

VirtualBox permite diferentes formas de networking entre las máquinas virtuales y con el exterior. Se remite a su manual para más detalle¹.

En este apartado se le pide que cree un escenario como el que se muestra en la Figura 1. Los equipos vRouter1 y vRouter2 son máquinas virtuales Linux con dos interfaces Ethernet, actuando como routers. Los equipos PC1, PC2, PC3 y PC4 son máquinas virtuales con un interfaz de red, actuando como hosts normales. Todos ellos son guests en el mismo host VirtualBox.

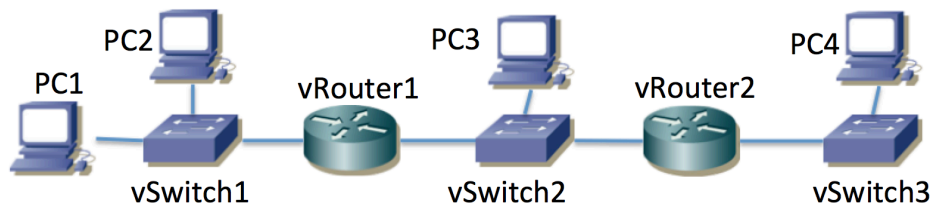


Figura 1 - Topología de red

Los equipos vSwitch1, vSwitch2 y vSwitch3 podrían ser equipos Linux actuando como puentes, sin embargo para este ejercicio se recomienda emplear la capacidad de crear redes internas independientes en VirtualBox². Cree tres redes internas independientes para actuar como cada una de esas LANs. Estas redes internas, para un uso “normal” se comportan casi como un conmutador, aunque tenga en cuenta que no tienen exactamente el mismo comportamiento y en ciertos escenarios complejos pueden dar problemas.

Configure direcciones IP y rutas en todos los hosts y routers y compruebe la conectividad y caminos con traceroute. Si la instalación de Linux que ha empleado para los PCs incluye servidor y cliente de ssh puede probar también a acceder de una máquina virtual a la otra.

No hace falta que ninguno de los PCs incluya un interfaz gráfico.

Punto de control: Muestre el escenario completo en funcionamiento al profesor. Por ejemplo pruebe a hacer traceroute entre cualquier par de PCs del escenario virtual.

¹ <https://www.virtualbox.org/manual/ch06.html>

² https://www.virtualbox.org/manual/ch06.html#network_internal

3. Migración de máquinas virtuales entre hosts

En este apartado debe reproducir un caso de movimiento de máquina virtual “en caliente” de un host a otro. VirtualBox implementa algo de ese estilo mediante la funcionalidad que llama “Teleporting”³.

Se sugiere crear un escenario como el mostrado en la Figura 2. En él los equipos Host1 y Host2 son ordenadores del laboratorio, interconectados por la LAN del mismo y “vm” es una máquina virtual ejecutándose gracias a VirtualBox en una de esas máquinas/hosts y que debemos mover sin detenerla a la otra.

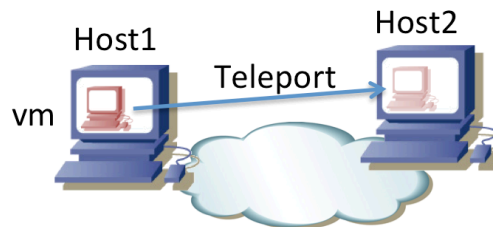


Figura 2 – Escenario de *teleporting*

Queremos que la máquina virtual tenga conectividad de red con el exterior. Puede hacer esto por ejemplo ofreciendo los host un acceso puentado a la LAN del laboratorio. Deberá entonces asignar una dirección IP a la máquina virtual que no colisione con otras máquinas del laboratorio. Consulte al profesor qué dirección puede emplear. Compruebe qué dirección MAC está empleando la máquina virtual y que efectivamente al comunicarse con otra máquina del laboratorio la comunicación es puentada (se mantienen las direcciones MAC). Compruebe qué sucede si antes del *teleport* la máquina virtual estaba haciendo un ping sostenido a otras máquinas de la red y qué sucede si lo que tenía es una conexión TCP establecida con otra máquina (por ejemplo un ssh o empleando *nc*). Analice el tráfico producido (puede emplear *wireshark* o *tcpdump*).

Ahora prueba implementando el acceso de la VM a la red mediante un NAT ofrecido por el host y repita el experimento de *teleport* así como la comprobación de si se mantiene el funcionamiento de ping y ssh.

Punto de control: Muestre una migración de máquina virtual al profesor y conteste a sus preguntas.

4. Recomendaciones, pistas, consejos y comentarios

- Para los guest del apartado 2 puede emplear una instalación pequeña de Linux, por ejemplo una *Slitaz*⁴, un *Linux Core*⁵ o un *Tiny Core*⁶. Si escoge una pensada para tenga en cuenta que probablemente necesite activar el reenvío de paquetes con el comando *sysctl*⁷. Si toma un *Linux Core* pensado para actuar como router traerá ya activada esta opción (aunque la imagen del disco puede ser un poco más grande). En cualquier caso tenga en cuenta que algunos de estos Linux pequeños montan el sistema de ficheros en un RAM disk, así que los cambios a ficheros no permanecen tras un reinicio del mismo (pero esto probablemente no resulte trascendente en la primera parte de esta práctica). Un *Linux Core* o una *Slitaz* puede correrla como un LiveCD o hacer una instalación local. Consulte la web de la asignatura si hay disponible alguna instalación ya hecha, aunque siempre es un

³ <https://www.virtualbox.org/manual/ch07.html#teleporting>

⁴ <http://www.slitaz.org>

⁵ <http://brezular.com/2013/09/17/linux-core-appliances-download/>

⁶ <http://tinycorelinux.net>

⁷ `sysctl net.ipv4.ip_forward=1`

buen ejercicio hacer una propia (pero se le recomienda hacerlo fuera del horario de prácticas para aprovechar el tiempo del mismo).

- Para ahorrar recursos puede crear una máquina virtual y clonarla varias veces (clon enlazado o “*linked clone*”) en VirtualBox.
- Si emplea NAT en el host para la conectividad de un guest hacia el exterior y este último no toma dirección IP automáticamente será probablemente porque éste no tenga lanzado un cliente de DHCP (el host haciendo NAT suele tener un servidor de DHCP para los guests). Puede lanzar un cliente de DHCP en la mayoría de los Linux del guest haciendo simplemente (si tiene instalado *dhclient*):

```
# dhclient eth0
```

Según la versión de Linux o el usuario empleado tendrá que usar *sudo* y puede que el cliente de DHCP tenga también otro nombre. El nombre del interfaz es el que quiera configurar (y que esté en una LAN con un servidor DHCP, como es el caso de una red NAT en VirtualBox).

- Si emplea un guest con un sistema operativo de escritorio como puede ser un *Ubuntu Desktop* tenga en cuenta que normalmente hay servicios del sistema encargados de configurar automáticamente los interfaces de red. Puede que usted vaya a configurar un interfaz de red en línea de comandos (*ifconfig*) y se encuentre al rato con que el sistema le ha cambiado la configuración. Normalmente se puede desactivar ese comportamiento con el panel de control correspondiente a la configuración de red en ese sistema operativo de escritorio (también puede ahorrarse problemas de este tipo y ceder ante la tentación de configurar el interfaz de red con dicho panel de control).
- Para un *teleport* necesita que el disco de la máquina virtual a mover se encuentre en un volumen compartido entre los dos hosts. Recuerde que su directorio *home* en el laboratorio se encuentra en un volumen compartido por NFS con todas las máquinas del mismo. No emplee una máquina virtual con un disco grande si pretende guardarla en su directorio *home*; tenga en cuenta que el laboratorio implementa cuotas (puede ver su límite de usuario con el comando *quota*).
- Para hacer un *teleport* entre dos máquinas del laboratorio se recomienda que en cada una haga *login* con un usuario diferente (coördínesse con un compañero o pida una segunda cuenta al profesor). El motivo es que puede dar problemas hacer *login* con el mismo usuario en dos máquinas dado que el *home* del mismo es compartido y el interfaz gráfico guarda ficheros de configuración, por lo que podrían entrar en conflicto ambas máquinas al modificar los mismos ficheros.
- En caso de querer crear una máquina virtual con un disco de centenares de megabytes o incluso algún gigabyte, en las máquinas del laboratorio tiene un directorio en el disco local en el que puede escribir: */opt/rng/practica*

Tenga en cuenta que ese directorio es local, no está compartido con otras máquinas. También vigile su uso del disco pues si lo llena empezará a tener problemas de uso del ordenador. No debería necesitar crear máquinas virtuales de más de un par de gigabytes. Por defecto VirtualBox guarda las máquinas virtuales que crea o importa en un directorio en el *home* del usuario. Desde las preferencias de VirtualBox, en la sección General puede cambiar ese comportamiento y cambiarlo por ejemplo a que se guarden en un subdirectorio de */opt/rng/practica* para no tener problemas de espacio (aunque entonces solo las tendrá disponibles en esa máquina). También cuando importa una VM desde un fichero *.ova* tiene en la ventana de importación la opción de modificar dónde se va a guardar la VM. Otra causa habitual de agotamiento de cuota es que su carpeta de descargas del navegador se encuentra también en su *home*. Si hace la descarga de imágenes grandes hágalo en */opt/rng/practica*

- La prueba de *teleport* se puede llevar a cabo sin emplear los ordenadores del laboratorio, empleando por ejemplo un ordenador portátil, aunque puede resultar más complicada (pero también más instructiva). Si dispone de un ordenador con una CPU suficientemente moderna puede instalar dos máquinas virtuales que incluyan una instalación de VirtualBox cada una y mover una máquina virtual desde una de esas máquinas virtuales a la otra (Figura 3). En este caso la virtualización es doble pero puede llevar a cabo el ejercicio con un solo ordenador (esto se ha probado sin demasiados problemas empleando un ordenador del laboratorio). En la configuración de red de VirtualHost de los HostVirtualX (Figura 3) configure que puedan usar el modo promiscuo en el interfaz. Pruebe tanto el caso en que los HostVirtualX tienen configurado dar acceso al guest vm a la red con un NAT como con un bridge. Puede poner un tercer host para poder hacer ping o ssh desde vm hasta él mientras teleporta esa máquina.

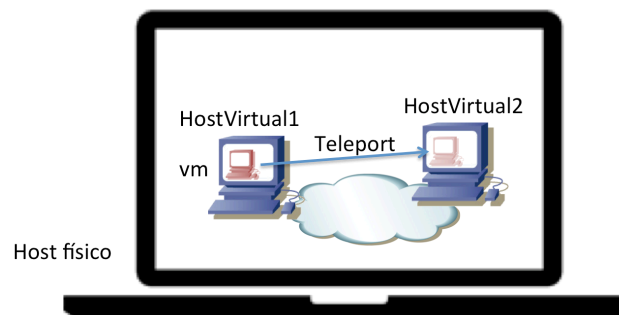


Figura 3 - *Teleport* completamente virtualizado

- En todos los casos de *teleport* recuerde que el disco de la máquina virtual debe encontrarse en un volumen accesible por los dos hosts. En el caso del laboratorio esto se puede lograr sin más que teniendo dicho disco en su *home*, dado que ese directorio está montado de un servidor NFS en todos los tlmXY (si está dando acceso al fichero de disco de una VM a otro usuario deberá tener cuidado con los permisos de acceso – comando `chmod`) En el caso de un *teleport* en un escenario completamente virtualizado (por ejemplo en su portátil) tendrá que instalar una tercera máquina virtual que actúe como servidor NFS donde hospede la imagen del disco de la máquina virtual (Figura 4) o puede hacer que uno de los HostVirtualX actúe también como servidor NFS (simplemente resulta un poco más confuso). Cómo configurar un servidor NFS depende de la versión de Unix⁸ (en un Ubuntu puede que tenga que indicar el directorio que desee compartir en el fichero `/etc/exports` y activar algún servicio del sistema o del kernel; en el cliente normalmente deberá usar el comando `mount`)

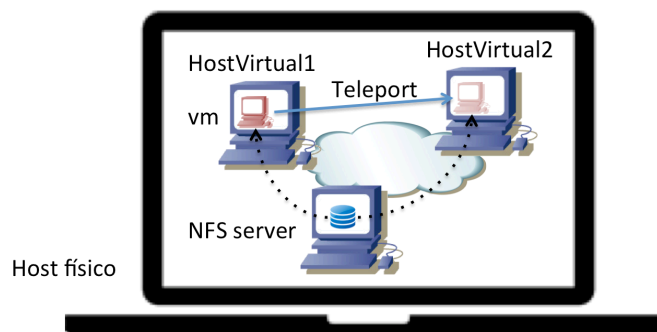


Figura 4 - *Teleport* con disco en servidor NFS independiente

- En el caso de la Figura 4, lo más sencillo es implementar la infraestructura de comunicación entre las tres máquinas virtuales (NFS server, HostVirtual1 y HostVirtual2)

⁸ <http://www.tldp.org/HOWTO/NFS-HOWTO/server.html>

mediante una Internal Network del VirtualBox de su Host físico. Esta opción funciona en casi su totalidad, pero puede encontrarse con algunos problemas debido a que esa Internal Network no se comporta exactamente como un conmutador Ethernet. Tenga en cuenta que ahora los HostVirtualX que se conectan a esa Internal Network envían tramas tanto con su dirección MAC como con la dirección MAC de vm. Una alternativa que da un funcionamiento completo del Teleport (pudiendo mantener conexiones TCP establecidas en la vm que se migra) es implementar esa comunicación mediante un puente, por ejemplo un puente con un Linux.

La Figura 5 intenta mostrar este último escenario, donde los 3 hosts (virtuales) se interconectan a través de un cuarto host que es un Linux con 3 interfaces en el que se ha configurado que haga de puente entre ellos. Para crear un puente y añadir interfaces al mismo se suelen emplear la *bridge-utils*, de las cuales necesitará el comando *brctl*. Puede crear el puente con "*brctl addbr*" y añadir a él interfaces con "*brctl addif*" (volveremos a usar este comando en otras prácticas).

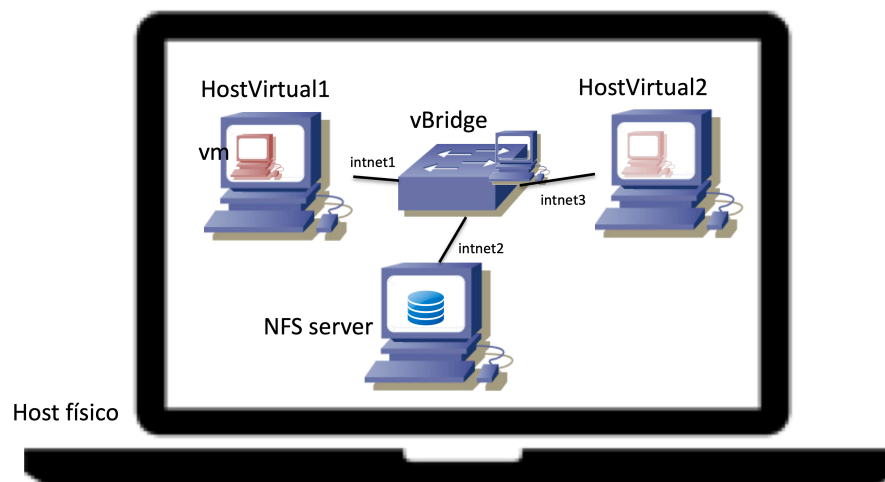


Figura 5 - Teleport en un escenario con puente implementado con un Linux

- En la configuración de los interfaces de red, en la parte "Advanced" (Figura 6), cuando haga estos escenarios con virtualización dentro de virtualización o máquinas virtuales haciendo de puentes es recomendable que configure el "Promiscuous Mode" para que permita a las máquinas virtuales ver tramas que no son para ellas. También en esa sección tiene el control de la dirección MAC del interfaz de la máquina virtual, la cual está en la configuración de la máquina en VirtualBox y debe ser la misma en el origen y destino del Teleport.

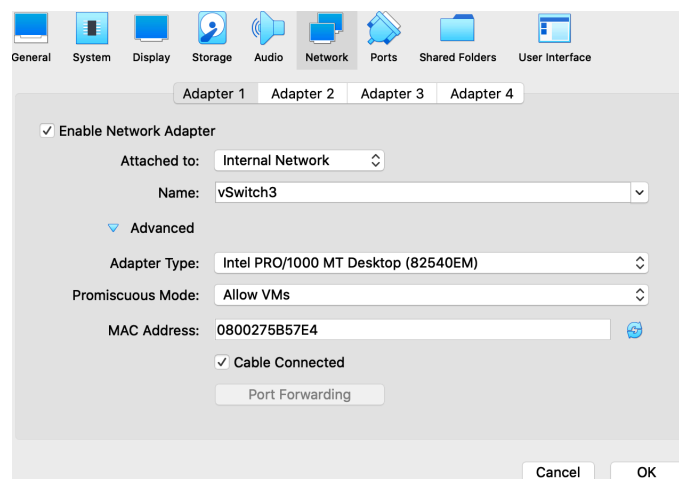


Figura 6 - Configuración de interfaces de red de VM en VirtualBox