

Pequeñas notas sobre Virtualbox 3.2.10

Óscar Casal Sánchez

oscar.casal@gmail.com

Sobre el relator

- *Óscar Casal Sánchez*
 - *Administrador de sistemas en el Dpto Electrónica y Computación de la USC desde 2003*
 - *Miembro del GLUG y AGNIX*
 - *LPIC3*
 - *Colaborador de la Oficina de Software Libre de la USC (OSLUSC)*

Contenidos

- *Emulación, virtualización total y paravirtualización*
- *Introducción a Virtualbox*
- *Instalación en sistemas debian*
- *Tipos de discos duros y configuración*
- *Tipos de red y configuración*
- *“Teleporting”*

Emulación

- *Emular es cuando se usa software para simular hardware para que un sistema operativo invitado corra.*
- *Es la más costosa en términos de computación y la menos eficiente, ya que obliga a simular completamente el comportamiento de la máquina*
- *Ejemplos de ello son: qemu, Bochs, VirtualBox...*

Virtualización completa

- *Son aquellas soluciones que permiten que se ejecuten sistemas operativos invitados, sin tener que modificarlos, utilizando en el medio un Hypervisor o Virtual Machine Monitor (VMM) que permite compartir el hardware real*
- *El inconveniente sobre la emulación es que el sistema operativo debe estar soportado en la arquitectura virtualizada*
- *Ejemplos de ello son VMWare Server Vmware Workstation, Virtual PC y Virtual Server*

Paravirtualización

- *El rendimiento o eficiencia se acerca al rendimiento nativo*
- *Los sistemas virtualizados deben estar basados en sistemas operativos modificados para ejecutarse sobre un Hypervisor*
- *Ejemplos de ello son Microsoft Hyper-V y Vmware ESX Server*

Es Virtualbox una buena elección para virtualizar?

- *Ventajas*
 - *Fácil de usar*
 - *Sencillo instalar*
 - *Sencillo crear máquinas invitadas*
 - *Portada a múltiples SO anfitriones*
 - *Soporta multitud SO invitados*
 - *Funciona con casi cualquier tipo de hardware*
 - *...*

Es Virtualbox una buena elección para virtualizar?

- *Inconvenientes*
 - *Emula (veremos que no siempre)*
 - *No soporta roles de usuarios sobre las máquinas*
 - *Soporta **teleporting** a partir versión 3.1*
 - *No soporta cambio de hardware sobre la máquina invitada cuando esta está encendida*
 - *Casi inexistente soporte para hardware real de la máquina anfitriona en la máquina invitada*
 - *...*

Es Virtualbox una buena elección para virtualizar?

- *Conclusiones*
 - *Buen sistema de virtualización para usuario doméstico*
 - *Desaconsejable sistema de virtualización para un sistema en producción*

Instalación de virtualbox 3.2 en sistemas debian

- *Tipos de instalación de Virtualbox*
 - *Bajar los fuentes y compilarlos a mano*
 - *Descargar el paquete deb e instalarlo con dpkg*
 - *apt: el más práctico y con menor esfuerzo*

Instalación de virtualbox 3.0 en sistemas debian

- *Procedimiento descrito en [página de Virtualbox](#)*
- *Agregar usuario/s que se quiere que puedan usar virtualbox a grupo vboxusers*

Instalación de virtualbox 3.0 en sistemas debian

- **Problema:** Al instalar por apt nos arrastra los headers de nuestro kernel actual. Si actualizamos el kernel, entonces nuestro virtualbox dejará de funcionar ya que no se actualizan también los headers
- **Solución:** instalando el metapaquete de los headers de nuestro kernel (linux-headers-{generic, generic-pae, server, virtual, 386...}) lo resolveremos para “siempre”
 - `#apt-get install linux-headers-`(uname -r |cut -d -f3)``
 - Esperar dkms haga su trabajo
 - Alternativa:
`#/etc/init.d/vboxdrv setup`

Front-ends

- **VirtualBox** es la interfaz de usuario gráfica
- **VboxManage** es la interfaz de línea de comandos (más extensa que la GUI gráfica)
- **VboxSDL**: alternativa gráfica, pero muy limitada
- **VboxHeadless**: front-end que no produce salida gráfica, pero que actúa como servidor VRDP y que no necesita soporte gráfico

Iniciando Virtualbox

- *Licencia y registro*
- *Front-end “Virtualbox”*
 - *Preferencias*
 - *Administrador de medios virtuales*
 - *Diferencias entre importar/exportar servicios virtualizados, copiar VDI y clonar el “disco duro”*

Máquina guest

- Creación máquina ejemplo desde GUI
- Gestión/modificación hardware
- Instalación *guest additions* para soporte de:
 - Integración puntero del ratón
 - Mejor soporte de vídeo (3D)
 - Sincronización de tiempo
 - Carpetas compartidas
 - Conexión escritorio remoto
 - Autenticación windows automática
 - Siempre que se actualice la versión de VirtualBox, debería de actualizarse la versión de las guest additions

Arranque máquina guest

- *VirtualBox GUI*
- *VirtualBox --startvm <vmname|UUID>*
- *VBoxManage startvm <uuid|name> --type gui|sdl|vrdp|headless>*
- *VBoxHeadless:*
 - *comando interesante si se tiene un servidor remoto que no tiene X, para poder iniciar las MV.*
 - *Igual que usar VBoxManage con opción --type vrdp)*
 - *Opción para grabar vídeo a archivo*
- *VBoxSDL --startvm <name>*

Práctica: creación/arranque máquina guest por comandos

VBoxManage createvm --name Server2 --register

VBoxManage modifyvm Server2 --memory 256 --acpi on --boot1 dvd --nic1 nat

VBoxManage createhd --filename Server2.vdi --size 8000 --register

VBoxManage storagectl Server2 --name "SATA Controller" --add sata

VBoxManage storageattach Server2 --storagectl "SATA Controller" --port 0 --device 0 --type hdd --medium Server2.vdi

VBoxManage storagectl Server2 --name "IDE Controller" --add ide

VBoxManage storageattach "Server2" --storagectl "IDE Controller" --port 0 --device 0 --type dvddrive --medium /full/path/ubuntu.iso

VBoxHeadless --startvm Server2 o

VBoxManage list [--long] vms #Sirve para ver VM's existentes

VBoxManage list runningvms #Sirve para ver VM's corriendo

Disposición de los ficheros de VirtualBox en disco

- Se guarda todo dentro de `$HOME/.VirtualBox`. Dentro encontraremos lo siguiente:
 - Fichero **VirtualBox.xml** donde se guardan las configuraciones globales de discos duros, imágenes de DVD y de disquette, de máquinas creadas, servicios de red como DHCP...
 - Directorio **HardDisks** donde se guardan los discos `.vdi` creados donde están realmente instaladas las máquinas huésped
 - Directorio **Machines** donde se almacena un directorio para cada máquina huésped creada. Dentro un fichero xml con su configuración, el directorio Logs y Snapshots donde se guardan las instantáneas de las máquinas

Imágenes disco virtuales y tipos discos

- El método más común para proveer almacenamiento de disco a las MV es a través de un controlador de disco virtual, en las cuales el disco es un fichero virtual, aunque también podría serlo una partición real
- Las imágenes de disco virtuales pueden ser (ver comando VBoxManage createhd):
 - Expansión dinámica:
 - El tamaño del disco es tratado con el tamaño mínimo
 - Crece automáticamente a medida que el sistema huésped va creciendo de tamaño
 - Ventaja: necesidad menor de espacio en disco hosts
 - Inconveniente: peor rendimiento si disco está creciendo continuamente
 - Problema: si posteriormente se libera espacio en el guest, este espacio no es liberado en el disco real del sistema host. Para solucionarlo tenemos un comando que permite borrar los bloques que sólo contienen ceros (ver comando zerofree en máquinas GNU/Linux)

`VBoxManage modifyhd <uuid>|<filename> --compact`

Imágenes disco virtuales y tipos discos

- *Tamaño de disco fijo:*
 - *El archivo imagen del disco es creado con el tamaño especificado*
 - *Esto reserva todo el espacio solicitado*
 - *Inconveniente: necesidad de mayor espacio en disco en máquina host*
 - *Ventaja: mejor rendimiento*
- *Otros tipos: Split2G, ESX*
- *Para ver información sobre un disco ya existente:*
VBoxManage showhdinfo <uuid|filename>

Trabajando con discos duros: Tipos especiales

• Tipos

• **Imágenes normales (default):**

- No hay ningún tipo de restricción en como los guest leen y escriben a disco
- No puede ser usado por más de una máquina en ejecución
- Cuando se coge una instantánea, deja de escribirse al disco normal y los cambios serán escritos en esta instantánea.
 - `VBoxManage createhd --filename <filename> --size <MB> --type normal`
 - `VBoxManage modifyhd <uuid>|<filename> --size <MB> --type normal`

• **Discos duros “Write-through”:**

- Su estado no es salvado cuando se hace un snapshot
- Tampoco es restaurado cuando se restaura un snapshot
 - `VBoxManage createhd --filename <filename> --size <MB> --type writethrough`
 - `VBoxManage modifyhd <uuid>|<filename> --type writethrough`

• **Imágenes inmutables:**

- Los cambios hechos se hacen temporalmente. cuando apagamos la máquina y la volvemos a encender estos cambios se han perdido. No es así si hacemos un reinicio desde dentro de la máquina huésped
- Imágenes inmutables no son reseteadas si se ha hecho una instantánea mientras la máquina estaba en ejecución. En tal caso, en cada inicio vuelve al estado de la última instantánea realizada
- A diferencia de las imágenes normales, un disco inmutable puede ser usado en varias VM sin restricciones
- Los contenidos del disco son de sólo lectura, y cualquier cambio es escrito a una separada imagen de disco diferencial que es reseteado en cada arranque
- No se puede poner la propiedad de inmutable a un disco al crearlo
 - `VBoxManage modifyhd <uuid>|<filename> --type immutable`

Trabajando con discos duros: Tipos especiales

Discos duros “shareable”:

- Son una variante de los discos duros write-through
- La diferencia con los write-through es que ellos pueden ser compartidos entre varias MV corriendo a la vez
- Perfecto para el uso de sistemas de ficheros tipo cluster entre MV y aplicaciones similares que necesitan acceso a disco concurrente
- Únicamente puede ser usado con imágenes de tamaño de disco fijo.

```
VBoxManage createhd --filename <filename> --size <MB> --type shareable
```

```
VBoxManage modifyhd <uuid>|<filename> --type shareable
```

Trabajando con discos duros: clonando

- *Puede resultar interesante clonar un disco existente:*
 - *Para distribuir el hd*
 - *Para cambiar el tipo de formato, lo cual nos permite importar máquinas procedentes de otros sistemas de virtualización*
 - *--format VDI|VMDK|VHD*
 - *Para cambiar el tipo de disco. Ya que cuando creamos un hd, a posteriori, esta propiedad no puede ser modificada*
 - *--variant Standard, Fixed, Split2G, Stream, ESX*
 - *Se hace con el comando: VBoxManage clonehd*

Práctica: creación MV sobre disco real

- Vamos a crear una MV sobre un disco duro real. En nuestro caso vamos usar /dev/sdb que es un lápiz usb (sino, usar el correspondiente dispositivo)

```
VBoxManage createvm --name Ubuntu4 --register
```

```
VBoxManage modifyvm Ubuntu4 --memory 256 --acpi on --boot1 dvd --nic1 nat
```

```
VBoxManage internalcommands createrawvmdk -filename  
/root/.VirtualBox/HardDisks/sdb_HDD.vmdk -rawdisk /dev/sdb -register
```

```
VBoxManage storageattach "Ubuntu4" --storagectl "IDE Controller" --port 0  
--device 0 --type hdd --medium /root/.VirtualBox/HardDisks/sdb_HDD.vmdk
```

```
VBoxManage storagectl Ubuntu4 --name "IDE Controller" --add ide
```

```
VBoxManage storageattach "Ubuntu4" --storagectl "IDE Controller" --port 0  
--device 1 --type dvddrive --medium /full/path/ubuntu.iso
```

```
VBoxHeadless --startvm Ubuntu4
```

- Al acabar la instalación apagamos la máquina y cambiamos orden de arranque de los dispositivos para que arranque del disco duro

```
VBoxManage modifyvm Ubuntu4 --boot1 disk
```

Compartir carpetas entre máquina invitada y máquina host

- *Necesario una carpeta compartida (para ello se instalará samba)*
- *Hacer accesible esta carpeta a la máquina invitada*

```
VboxManage sharedfolder add <vmname>|  
<uuid> --name nameFolder --hostpath  
/full/path/folder <readonly>
```

- *Una vez arrancada la MV montar la carpeta compartida*

```
$sudo mount -t vboxsf share nameFolder
```

- *Para eliminar una carpeta compartida en una máquina*

```
VboxManage sharedfolder remove  
<vmname>|<uuid> --name nameFolder
```

Red: Tipos de hardware virtual de red en Virtualbox

- Hay diferentes tipos de tarjetas de red que van desde velocidades de 100Mb a 1000Mb. Se pueden ver desde este [enlace](#)
 - AMD PCNet PCI II
 - AMD PCNet FAST III (por defecto)
 - Intel PRO/1000 MT Desktop
 - Intel PRO/1000 T Server
 - Intel PRO/1000 MT Server
 - Adaptador de red Paravirtualizado (virtio-net):
 - Esta tarjeta de red es especial ya que no virtualiza el hardware de red común
 - Emplea una interfaz de software especial para MV virtualizadas dada por el huésped, evitando así:
 - Complejidad de emular el hardware
 - Mejorando el rendimiento de red.
 - En GNU/Linux driver soportado por kernels mayores o igual a 2.6.25
 - En Windows los drivers pueden ser bajados de la [página de KVM](#)
 - A partir versión 3.1 Virtualbox
- Podemos elegir un tipo de tarjeta u otro desde interfaz de Virtualbox o con el comando:
`VBoxManage modifyvm <uuid|name> [--nictype<1-N> Am79C970A|Am79C973|82540EM|82543GC|82545EM|virtio]`

Red: tipos de modos de red

- ***Not attached***: *Virtualbox avisa al guest de que una tarjeta de red está presente, pero que no hay conexión.*
- ***NAT*** (*Network Address Translation*):
 - *Se tiene conexión a Internet*
 - *Puedes conectarte a otros equipos de tu red Interna*
 - *Los equipos de la red no tienen acceso a ti, incluido el equipo host de la VM*
 - *Ping no trabaja sobre NAT*
 - *Ciertas limitaciones cuando se está usando compartición de archivos sobre Windows*

Red: tipos de modos de red

- ***Bridged networking:***

- *Nuestra VM es un equipo más de la red interna*
- *Perfecto para correr servidores en máquinas guest*
- *Se tiene conexión “directa” con todos los equipos*
- *Todos los equipos tienen conexión directa contigo*

- ***Internal networking:***

- *Cuando se quiere que tengan conexión entre sí las máquinas virtuales seleccionadas*
- *Se quiere que ni el equipo host, ni la red interna de la empresa ni el mundo exterior tenga conexión con este equipo*

Red: tipos de modos de red

- ***Host-only networking***

- *Usada para crear una red que contiene al host y a un conjunto de máquinas virtuales*
- *No se necesita físicamente una interfaz de red*
- *Usa una interfaz de red virtual similar a una interfaz loopback, la cual es creada en el host*

Red: NAT

- *Es el modo por defecto*
- *El más simple modo de acceder a la red externa desde una máquina virtual*
- *No requiere configuración en la red de la máquina host ni en la del guest.*
- *Es como estar detrás de un router con sus ventajas e inconvenientes*
- *Ventajas:*
 - *Las máquinas invitadas están protegidas de accesos indeseados*

Desventajas “o puede que no”:

- *La máquina virtual es invisible e inalcanzable desde resto máquinas, incluido el host*
- *No se puede correr un servidor, a menos que se haga un “port forwarding”*

Red: Nat IP address

- *La máquina virtual recibe su dirección IP y configuración procedente del DHCP de VirtualBox*
- *La primera tarjeta de red pertenece a la red privada 10.0.2.0, la segunda a la 10.0.3.0 y así las siguientes*
- *Por defecto, la primera tarjeta de red tendrá los siguientes datos:*
 - *IP: 10.0.2.15*
 - *Gateway: 10.0.2.2*
 - *Servidor de nombres: 10.0.2.3*
- *Si queremos cambiar la red asignada a la VM*
VBoxManage modifyvm "Name VM" --natnet[1-N]
"172.10/16"

Red: NAT: “port forwarding”

- *Por qué usar “port forwarding” y no tarjeta de red en modo bridged?*
 - *El port forwarding nos aporta:*
 - *Mayor seguridad:*
 - *Únicamente los puertos con NAT serán accesibles desde otro equipo incluido la máquina host*
 - *Desconocimiento de que al servicio que se accede está realmente en una máquina virtual y no en la máquina host*
 - *Ahorro de IP's en la red interna, ya que una máquina host puede tener varias MV cada una con un/os servicios levantados, pero para el exterior accesibles desde un único equipo*
 - *Ideal en ambientes de desarrollo para que la gente pruebe en MV*
 - *El port forwarding para puertos < 1024 en la máquina host únicamente es posible si se es administrador de la máquina host*

Red: NAT: “port forwarding”

- *Practica: redirigir puerto 1301 de host a puerto 22 máquina guest*
- *Nota: Para que tenga efecto, la/ máquina/s guest deben ser paradas y también la gui de Virtualbox en el caso de que la tengamos abierta deberá ser cerrada. Sino, hasta haber reiniciado ambas, no tendrá efecto.*
 - *Configuramos puerto a usar en la máquina host*
`VBoxManage setextradata <namevm>`
`"VBoxInternal/Devices/pcnet/0/LUN#0/Config/SSH/HostPort" 1301`
 - *Configuramos puerto a usar en la máquina guest*
`VBoxManage setextradata <namevm>`
`"VBoxInternal/Devices/pcnet/0/LUN#0/Config/SSH/GuestPort" 22`
 - *Configuramos protocolo*
`VBoxManage setextradata <namevm>`
`"VBoxInternal/Devices/pcnet/0/LUN#0/Config/SSH/Protocol" TCP`
 - *Comprobamos el resultado*
`VBoxManage getextradata <namevm> enumerate`
 - *Probamos a acceder al puerto 1301 de la máquina host*

Red: NAT: “port forwarding”

- *En el caso de que la tarjeta de red fuera una intel tendríamos que sustituir la cadena pcnet por e1000*

Red: Internal Networking

- *Beneficios de este tipo de red:*
 - ***Seguridad:*** ya que el tráfico no pasa por la máquina host como ocurre en el modo bridge, en el cual se podría poner un sniffer y ver todo el tráfico
 - ***Velocidad:*** internal networking es más eficiente que el modo bridge ya que Virtualbox puede transmitir los datos sin tener que pasar estos por la pila de red del sistema host

Red: Internal Networking: práctica1

- 2 máquinas guest (Ubuntu3 y Ubuntu4) conectadas entre sí en modo "Internal Networking"
 - Clonamos Ubuntu3
`VBoxManage clonevdi Ubuntu3.vdi Ubuntu4.vdi`
 - Ligamos el hd creado a una VM's ya creada
`VBoxManage modifyvm Ubuntu4 -hda Ubuntu4.vdi`
 - Creamos la red para las 2 lo que permitirá que se conecten entre ellas, pero no con el host
`VBoxManage modifyvm Ubuntu3 -nic1 intnet`
`VBoxManage modifyvm Ubuntu3 -intnet1 redinterna`
`VBoxManage modifyvm Ubuntu4 -nic1 intnet`
`VBoxManage modifyvm Ubuntu4 -intnet1 redinterna`
 - Arrancamos las VM's y les asignamos a las 2 una IP dentro de la misma subred (por ejemplo 192.168.1.100 y 192.168.1.101)
 - Probamos que hay conexión entre las 2 máquinas, pero no con el host, ni con el resto equipos de la red interna, ni con el mundo exterior

Red: Internal Networking: práctica2

- *Posible escenario real*
 - *Un equipo con servidor web: queremos que acceda todo el mundo*
 - *1 tarjeta en modo bridged (IP de la red de la corporación)*
 - *1 tarjeta en modo internal networking (llamada “redinterna” con IP 192.168.1.100)*
 - *Un equipo con servidor MYSQL: queremos que únicamente se pueda acceder desde el servidor web*
 - *1 tarjeta en modo internal networking (llamada “redinterna” con IP 192.168.1.101)*

Red: Internal Networking: práctica2

- *Servidor web es Ubuntu3*
 - *La interfaz de internal networking ya nos sirve la anterior, tendremos que crear la interfaz de bridge*

VBoxManage modifyvm Ubuntu3 -nic2 bridge

- *Servidor BBDD es Ubuntu4*
 - *Podemos dejarlo como estaba ya antes configurado de la anterior práctica*

Red: Host-only networking

- *Es un híbrido entre el modo bridged y el modo internal networking ya que:*
 - *Las máquinas virtuales pueden hablar entre ellas y con el host*
 - *No pueden hablar con el resto del mundo, al igual que en la “internal networking” ya que no existe una interfaz de red física presente*
 - *Se crea en el equipo Host una nueva interfaz de red llamada por defecto “vboxnet0” con una ip asignada (por defecto 192.168.56.1), que será la IP con la que desde los guest podemos acceder al equipo Host*
 - *Las IP's en los equipos guest pueden ser asignadas:*
 - *Por el DHCP de Virtualbox*
 - *Estáticamente siempre estén dentro de la subred en la que se encuentra “vboxnet0” del equipo host.*

Red: Host-only networking: practica

- *Dos equipos (Ubuntu3 y Ubuntu4) con una única interfaz de red en modo host-only*
 - *En Ubuntu3 de la anterior práctica tenemos configuradas 2 tarjetas de red, así que podemos:*
 - *Desactivar cable de red*
`VBoxManage modifyvm Ubuntu3 --nic2 null`
 - *Eliminar la tarjeta de red*
`VBoxManage modifyvm Ubuntu3 --nic2 none`
 - *Pasamos a configurar las tarjetas de red en modo host-only*
`VBoxManage modifyvm Ubuntu3 --nic1 hostonly`
`VBoxManage modifyvm Ubuntu4 --nic1 hostonly`
 - *Arrancamos las máquinas guest y hacemos que cojan las IP's por DHCP*

Nota sobre interfaces de red en máquinas guest

- En linux cada vez que hacemos un cambio de red (agregamos quitamos una interfaz) su mac cambia, así que puede ocurrir que ya no existe eth0 porque está asignada a la MAC de la anterior tarjeta de red existente y nuestra interfaz actual es la eth5.*
- Solución: la más elegante es editar el fichero /etc/udev/rules.d/70-persistent-net.rules y poner en eth0 la MAC de nuestra actual interfaz de red*

“Teleporting”: descripción

- *Disponible desde la versión 3.1 de VirtualBox*
- *Se define como mover una máquina virtual sobre la red desde un host a otro, mientras la máquina virtual está corriendo*
- *No importa que SO origen y destino sean SO diferentes*
- *“Teleporting” ocurre sobre cualquier red TCP/IP. Únicamente necesitan estar de acuerdo en el puerto TCP/IP*

“Teleporting”: requisitos

- *Son necesarios unos prerequisites para que funcione:*
 - *La host destino se debe configurar una máquina virtual en VirtualBox con exactamente la misma configuración hardware que en el host origen*
 - *Las dos máquinas virtuales en origen y destino deben tener el mismo soporte de almacenamiento (discos duros, disquettes, imágenes CD/DVD)*
 - *Ni la máquina origen ni la destino pueden tener snapshots*

“Teleporting”: configuración

- *En la máquina destino escribimos:*

*VBoxManage modifyvm <targetvmname>
--teleporter on --teleporterport <port>*

- *Arrancamos la máquina en el target y veremos que pone “Teleportando”*

- *En la máquina origen, arrancamos la MV y cuando queramos “teletransportarla” a la máquina destino ejecutaremos:*

*VboxManage controlvm <sourcevmname>
teleport --host <targethost> --port <port>*

“Teleporting”: configuración

- *Para testear, se puede usar en el mismo host poniendo targethost = 127.0.0.1*
- *También podríamos colocar la opción --teleporterpassword <password>, en este caso, a no ser que en la máquina origen se coloque el mismo, no tendrá éxito.*

Documentación consultada

- *<http://www.virtualbox.org>*
- *Documentación de usuario de Virtualbox*