

Máquinas Virtuales

Departamento de Sistemas Telemáticos y Computación (GSyC)

<http://gsync.urjc.es>

Octubre de 2013



©2013 GSyC
Algunos derechos reservados.
Este trabajo se distribuye bajo la licencia
Creative Commons Attribution Share-Alike 3.0

- 1 Máquinas Virtuales
 - Utilidad de las máquinas virtuales
 - Inconvenientes de las máquinas virtuales
- 2 Estructura de los laboratorios del GSyC
- 3 Configuración de VirtualBox
 - Imágenes de máquinas virtuales
 - Interfaces de red en VirtualBox
 - Configuración que seguiremos en prácticas
 - Cambio de host en el laboratorio
- 4 Algunos bugs conocidos
- 5 Configuración del teclado

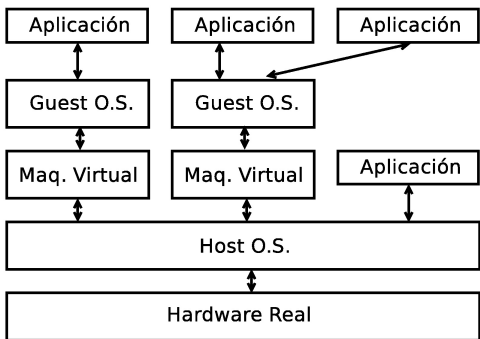
Máquinas Virtuales

Máquina Virtual: Software que crea una capa de abstracción, ofreciendo una máquina diferente a la máquina original

Las máquinas virtuales que nos interesan en administración de sistemas suelen ofrecer a un sistema operativo la percepción de una máquina física

- Las aplicaciones y los usuarios dentro de la máquina virtual se relacionan con la capa de abstracción y no con la plataforma real
- La máquina virtual puede implementar diversos dispositivos virtuales (disco, dispositivos de red, etc) diferentes a los de la plataforma real

- La tecnología sobre Máquinas Virtuales está muy madura. La terminología, no. Es frecuente encontrarse con el distintos nombres para el mismo concepto, o incluso el mismo nombre para cosas distintas
- *Guest*: Sistema Operativo de la máquina virtual
Host: Sistema Operativo de la máquina real



Este es uno de los modelos posibles

- La máquina virtual se comporta como una aplicación más en el *host*
- El *guest* percibe la máquina virtual como si fuera hardware real

Utilidad de las máquinas virtuales

Tecnología tradicional y actual, con muchas utilidades

- Ejecutar aplicaciones hechas para una plataforma sobre una plataforma diferente: p.e Microsoft Windows sobre Mac OS, Java Virtual Machine
- Ofrecer un entorno seguro donde experimentar (*sandbox*)
 - Docencia
 - Probar aplicaciones en desarrollo
 - Probar aplicaciones o webs no confiables
- Señuelos (*Honeypots*)
- Empresas de *hosting* pueden ofrecer servidores virtuales (alimentación y conectividad redundante, soporte 24/365, etc)

- Respaldo (*backup*) de máquinas enteras, no solo de datos. Ante un pequeño problema o un gran desastre, la máquina virtual se recupera inmediatamente
- Seguridad: Cortafuegos, perímetros de seguridad,...
- Portabilidad: Moviendo un directorio se puede mover la máquina virtual de una máquina real a otra
- Independencia del Hardware, p.e. homogeneizar un conjunto de máquinas diferentes
- ...

Inconvenientes de las máquinas virtuales

Inconveniente principal: pérdida de rendimiento

Aunque no siempre

- La máquina *real* tal vez no existe (p.e. java)
- Existe, pero es una máquina de propósito específico.
Un guest sobre un host de propósito general puede ser más eficiente

Estructura de los laboratorios del GSyC

- Para las prácticas de esta asignatura, tendrás una cuenta en los laboratorios Linux del Departamento GSyC
- La misma cuenta la usarás en las prácticas de muchas asignaturas del Departamento, durante toda la carrera/todo el máster

Campus de Fuenlabrada

Servidor: `bil0`

Clientes: `epsilonNN`, `zetaNN`, `thetaNN`, `iotaNN`, `kappaNN`

Para conocer la dirección IP de la máquina en la que estás trabajando puedes usar `hostname -i`

- Las direcciones IP de cada máquina pueden consultarse en el fichero `/etc/hosts` de cualquier equipo
 - Este fichero equivale a
 - `%SystemRoot%\system32\drivers\etc\hosts` (MS Windows)
 - `/private/etc/hosts` (Mac OS)
- La misma cuenta permite entrar en todas las máquinas
- Cada usuario verá el mismo *home* en todas las máquinas de Fuenlabrada
- Cada usuario verá el mismo *home* en todas las máquinas de Móstoles, distinto al de Fuenlabrada
- Los servidores están dimensionados para mover ficheros del orden de KBytes o MBytes, no GBytes
- Los ficheros de los directorios `/tmp/` y `/var/tmp` son locales a cada ordenador
- Como en todo linux,
 - El directorio `/tmp` se borra cada vez que se reinicia el ordenador
 - El directorio `/var/tmp` se borra cada vez que al administrador le parece oportuno, sin que debamos esperar aviso previo

Imágenes de máquinas virtuales

- Una de las ventajas de las máquinas virtuales es que pueden clonarse (copiarse) de un *host* a otro. Para ello basta copiar un fichero o ficheros: la *imagen de la máquina*
- VirtualBox llama a estas imágenes *servicio virtualizado*. En VirtualBox 4, es un fichero `.ova` ¹

¹En VirtualBox 3 eran 3 ficheros: `.ovf` `.vmdk` `.mf`

Para clonar una máquina virtual de un *host* a otro

- En el *host* origen, exportamos la imagen indicando dónde queremos guardar el *.ova*
- Llevamos este ficheros al *host* destino
- Importaremos el *.ova*, esto generará automáticamente una nueva copia del disco duro virtual, en el directorio especificado en

Archivo|Preferencias|General|
Carpeta predeterminada de máquinas

- Podemos borrar el *.ova*, pero normalmente será preferible conservarlo por si queremos en el futuro otra máquina *como nueva*

Observa que entonces tenemos 3 copias del disco duro virtual

- 1 La del *host* origen, en formato *.vmdk*
- 2 La *que viaja*, incluida dentro del fichero *.ova*
- 3 La del *host* destino, en formato *.vmdk*

Ejemplo típico de uso de máquinas virtuales

- Un profesor instala una máquina virtual partiendo de cero
Crea una máquina, especifica su tamaño de disco, de memoria, etc
- El profesor exporta la máquina virtual como fichero .ova (que dentro lleva un .vmdk) y la deja en algún lugar, como p.e. el directorio `/var/lib/vms` de las máquinas de sus alumnos
- Cada alumno instala la máquina virtual en su VirtualBox partiendo de la imagen creada por el profesor
- Ahora el alumno podría exportar de nuevo la máquina virtual y llevársela en un *pendrive* al pc de su casa

Algo muy parecido podría hacerlo un administrador con los equipos de sus usuarios, o un administrador que quiera conservar un servidor recién instalado para recuperarlo rápidamente si hay problemas

Instalación de una m.v. partiendo de cero

Si ya contamos con una imagen de la m.v, podemos omitir estos pasos. Pero en otro caso:

- 1 Lanzamos VirtualBox desde la shell
- 2 Pulsamos el botón *nueva*, que ejecuta el asistente para crear una nueva máquina virtual
- 3 Indicamos el nombre de la m.v. (p.e. pc01, ro01, auditor01)
Sistema operativo Linux, Versión Ubuntu
- 4 El tamaño de memoria base dependerá de lo que tenga el *host* y necesite en *guest*. Como referencia:
 - OpenWrt: 32 Mb
 - Ubuntu Server: 192 Mb
 - Ubuntu con gráficos, BackTrack con gráficos: 512 Mb

- 5 Activamos *Crear disco virtual* y seguimos el asistente
El tamaño del disco dependerá de lo que tenga el *host* y necesite en *guest*. Como referencia:
 - OpenWrt: 64 Mb
 - Ubuntu, Backtrack: 8Gb
- 6 En la ventana *resumen* revisamos todo y pulsamos *Terminar*

Una vez que hemos especificado los componentes de la máquina, habrá que instalar el sistema operativo, que normalmente tendremos en un cdrom/dvd o en una imagen iso de un cdrom/dvd

- En el apartado de configuración de la máquina virtual, en Almacenamiento | controlador IDE
(pulsamos en el icono que representa un CD con el signo +)
(pulsamos en el CD recién creado, llamado "vacío")
(Vamos a "dispositivo cd/dvd")
Aquí indicamos si usaremos el lector físico del *host* (anfitrión) o una imagen iso del cdrom/dvd

Instalación de una m.v. partiendo de una imagen

Desde VirtualBox

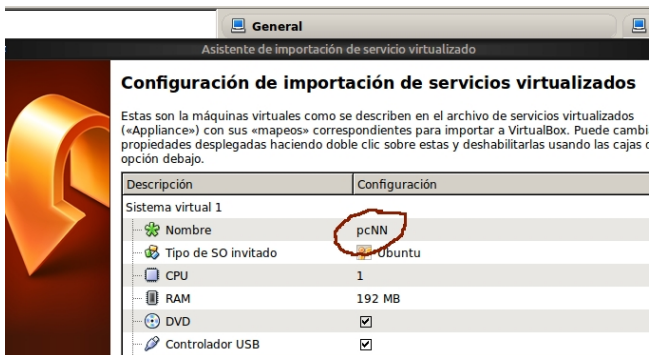
- Comprobamos en

Archivo|Preferencias|General|
Carpeta predeterminada de máquinas

que el disco duro virtual quedará copiado en el lugar adecuado.

- En casa, el lugar por omisión es válido:
~/VirtualBox VMs
 - En el laboratorio, es imprescindible que sea
/var/tmp/tulogin
- Archivo|Importar servicio virtualizado|Seleccionar

(Elegimos el fichero .ova)
 - En la ventana *Configuración de importación de servicios virtualizados* podemos cambiar algunos parámetros del *guest* (nombre, memoria, disco....)



General

Asistente de importación de servicio virtualizado

Configuración de importación de servicios virtualizados

Estas son las máquinas virtuales como se describen en el archivo de servicios virtualizados («Appliance») con sus «mapeos» correspondientes para importar a VirtualBox. Puede cambiar propiedades desplegadas haciendo doble clic sobre estas y deshabilitarlas usando las cajas de opción debajo.

Descripción	Configuración
Sistema virtual 1	
Nombre	pcNN
Tipo de SO invitado	Ubuntu
CPU	1
RAM	192 MB
DVD	<input checked="" type="checkbox"/>
Controlador USB	<input checked="" type="checkbox"/>

- Si dos máquinas van a compartir segmento de red, es necesario cambiar su dirección MAC
- Si la imagen original se llama p.e. pcNN, haciendo clic sobre este nombre en la pantalla de configuración de importación, podemos cambiarlo. P.e. para llamarla pc01

Este es el nombre de la máquina visto desde VirtualBox.
Para cambiar el nombre visto desde dentro de la propia máquina,
hay que

- O bien editar `/etc/hostname` y reiniciar
- O bien ejecutar la orden
`hostname <NUEVO_NOMBRE>`
(Pero este cambio no es persistente)

Fragmentación de ficheros

Si necesitas trocear una imagen de gran tamaño en ficheros que quepan en un *pendrive* o cdrom

- Empaquetar y comprimir un directorio:

```
tar -cvzf mi_imagen.tgz mi_directorio
```

- Mostrar contenido:

```
tar -tzf mi_imagen.tgz
```

- Trocear:

```
#      tamaño      fichero      prefijo
split -b 500MB  mi_imagen.tgz  mi_imagen.tgz.
```

(Observa que el segundo parámetro es igual al primero, pero añadiendo un punto)

- Habremos generado

```
mi_imagen.tgz.aa  mi_imagen.tgz.ab  mi_imagen.tgz.ac
```

En la máquina destino (no importa si en el *host* el S.O. es distinto)

- Unir los fragmentos

```
cat mi_imagen.tgz.* > mi_imagen.tgz
```

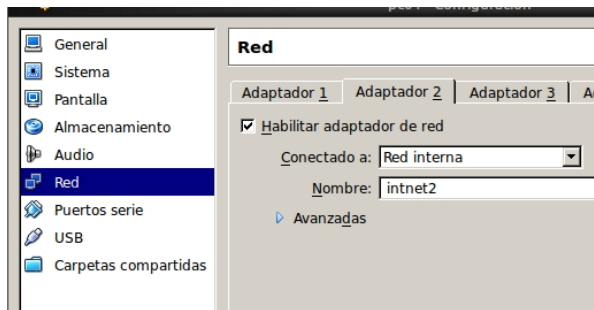
(En MS Windows para este paso podemos emplear Hjsplit, Free File Splitter o cualquier otro programa similar)

- Descomprimir y desempaquetar:

```
tar -xvzf mi_imagen.tgz
```

(En MS Windows podemos usar 7-Zip o similares)

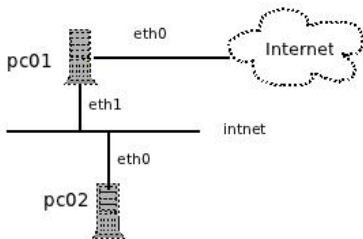
Interfaces de red de VirtualBox



- Cada máquina virtual puede tener hasta 4 interfaces *aka* adaptadores de red
adaptador 1 será eth0, *adaptador 2* será eth1, etc
- Cada interfaz puede conectarse a 5 tipos de segmento de red:
No conectado, NAT, Adaptador puente, Red interna, Solo anfitrión

- **Not attached / No conectado**
Emula una tarjeta con el cable de red desconectado
- **Network Address Translation (NAT)**
Configuración por defecto. El *guest* tiene acceso al exterior (típicamente internet) a través de NAT. El *host* no tiene acceso al *guest*
Podemos usar varios *guest*, pero cada uno tiene su propio NAT y está aislado en su propio segmento de red
- **Bridged networking / Adaptador puente**
Interfaz en el *guest* conectado virtualmente al mismo *hub* (real) que el *host*
El *guest* está en el segmento de red *normal* del *host*
- **Internal networking / Red interna**
Red entre diferentes *guests* en un mismo *host*
Sin acceso al *host* ni al exterior
- **Host-only networking / Sólo anfitrión**
Red entre el *guest* y el *host*, sin acceso al exterior
Permite tener varios *guests* en el mismo segmento de red

Supongamos que deseamos configurar dos *guest* de esta forma:



- En pc01, el interfaz eth0 estaría conectado a NAT. Dentro de la máquina virtual, lo configuraríamos para obtener sus parámetros por DHCP.
- En pc01, el interfaz eth1 lo conectaríamos a una red interna. El nombre por omisión de este segmento de red es *intnet* (atención, significa *internal net*, no *internet*). Podemos ponerle el nombre que queramos al segmento.

- Dentro de la máquina virtual pc01, configuraríamos estáticamente los parámetros de eth1
- En pc02, conectaríamos eth0 a una red interna, con el mismo nombre que la red interna de eth1 en pc01 (en este ejemplo, *intnet*)
- Dentro de pc02, configuraríamos estáticamente los parámetros de eth0

Uso de VirtualBox en los laboratorios del GSyC

Un disco duro virtual será típicamente un fichero de varios GBytes almacenado en

~/VirtualBox VMs

- En tu PC esto no será un problema
- En el laboratorio sí, el rendimiento sería muy pobre. Por tanto cambiaremos la ubicación por omisión de los discos duros virtuales

- En el *host*

```
mkdir /var/tmp/tulogin
```

(Donde *tulogin* es tu usuario del laboratorio, p.e. mgarcia, jperez...)

- En VirtualBox:

Archivo|Preferencias|General|

Carpeta predeterminada de máquinas

Indicamos

```
/var/tmp/tulogin
```

Muy importante: asegúrate de cambiar esta preferencia y mantenerla siempre. De lo contrario, cargarás mucho el servidor, perjudicandote a tí y a tus compañeros

Problema: las imágenes son ficheros grandes

Con lo visto hasta ahora, ya podríamos hacer las prácticas de la asignatura. Pero sería poco práctico.

Supongamos una práctica que consista en configurar en red 3 máquinas virtuales

- Cada alumno tendría que guardar en su cuenta del laboratorio 3 imágenes (con sus 3 discos duros virtuales completos)
- Para trabajar en casa, tendría que llevarse las 3 imágenes con sus 3 discos duros virtuales completos
- Para que 70 alumnos entreguen su práctica, el profesor tendría que manejar 210 discos duros virtuales completos

Si en la asignatura se hacen 2 o 3 prácticas, seguimos multiplicando...

Solución: almacenar solo los ficheros importantes

Administrar un Unix/Linux consiste en editar diversos ficheros de texto

- Solamente manejaremos estos ficheros, que estarán guardados en la cuenta de bilo y respaldados en la nube de Dropbox
- Las máquinas virtuales serán *de usar y tirar*, tomarán la configuración de estos ficheros
- Dentro de las máquinas virtuales, los ficheros de configuración serán enlaces simbólicos a ficheros en un directorio, que a su vez estará montado por red desde un directorio en el laboratorio

Ejemplo:

Para configurar los interfaces de red de una máquina, hay que editar `/etc/network/interfaces`

- Dentro de la máquina virtual `pc01`, este fichero será un enlace simbólico que apuntará a `/media/nube/interfaces`
- El directorio `/media/nube` de `pc01`, estará montado desde el directorio `~/Dropbox/pc01` del laboratorio
- Por tanto, `/etc/network/interfaces` en la máquina virtual `pc01` y `~/Dropbox/pc01/interfaces` en el laboratorio serán el mismo fichero, podrá editarse indistintamente cualquiera de los dos

Para montar el directorio remoto, aquí emplearemos sshfs

Ventaja principal:

- Basta con tener acceso por ssh a la máquina remota para poder montar un directorio

Pero esta idea de tener los ficheros importantes por separado y luego colocarlos automáticamente en su sitio puede aplicarse de muchas otras formas

- Tanto en máquinas físicas como virtuales
- En un entorno docente, doméstico, de oficina, granja de servidores...
- Mediante nfs, o scp, o rsync, o unison, o smb/cifs, o vboxsf...
- Con la oportuna atención a la seguridad si se trata de un sistema en producción

Montar un directorio con sshfs

Punto de montaje: directorio local donde veremos el directorio remoto

- Montar el *home* remoto:

```
sshfs usuario@maquina: /punto/de/montaje
```

- Montar un directorio remoto cualquiera

```
sshfs usuario@maquina:/un/directorio /punto/de/montaje
```

(Siempre path absoluto, no soporta ~)

- Desmontar:

```
fusermount -u /punto/de/montaje
```

No siempre es necesario tener privilegios de root (es configurable)
En conexiones lentas puede ser conveniente añadir la opción `-C` para que comprima el tráfico

```
sshfs -C usuario@maquina:/path /punto/de/montaje
```


Cambio de host en el laboratorio

La m.v. está en un directorio local del pc donde trabajas, no en tu cuenta de pantuflo/bilo

Si te sientas en un puesto del laboratorio distinto al del día anterior:

- 1 Sal de VirtualBox y borra la máquina vieja

```
rm -rf ~/.VirtualBox
rm -rf /var/tmp/tulogin/* # si este directorio existe
```

- 2 Vuelve a indicar en

```
Archivo|Preferencias|General|
Carpeta predeterminada de máquinas
```

que la carpeta predeterminada de máquinas tiene que ser
`/var/tmp/tulogin`

- 3 Vuelve a importar el servicio virtualizado (O copia
`/var/tmp/tulogin` desde el host anterior)

Observaciones

- Recuerda que el administrador puede borrar tu máquina virtual en cualquier momento, no guardes dentro nada de valor, todos tus ficheros deben estar en el directorio compartido
- En esta asignatura, cada máquina Ubuntu tendrá por omisión un usuario de nombre `user` y contraseña `user` autorizado a ejecutar la orden `sudo`
 - Recuerda que en el caso de Ubuntu, se espera que no empleemos el usuario `root`
- En las máquinas sin gráficos, podemos usar varias consolas pulsando `Alt F1`, `Alt F2`, etc
- Si dejamos la máquina virtual desatendida algunos minutos, puede saltar el salvapantallas y quedarse en negro. En tal caso, llevamos el foco a la máquina virtual (haciendo clic dentro) y pulsamos cualquier tecla

Algunos errores posibles

Si has empezado a importar una máquina virtual, te has equivocado en algo y has vuelto a empezar, VirtualBox puede mostrará un error indicando que ese disco duro ya está registrado y no puede importarse de nuevo

Soluciones

1 En

Archivo | Administrador de medios virtuales

Elimina esa imagen de la lista de medios conocidos, o elimínala por completo (una ventana te informará). Fíjate si es la imagen *que viaja* o la del *host destino*

- ### 2 Alternativa más drástica: Cerrar VirtualBox y borrar todo el directorio `~/VirtualBox` (esto elimina toda la configuración y todas las máquinas)

Si intentamos usar dos instancias de un *guest* en el mismo *host* nos dará un error indicando que ambos discos tienen el mismo identificador. En este caso, hay que clonar el disco

Ejecutamos desde la shell

```
VBoxManage clonehd <filename> <outputfilename>
```

Algunos Bugs conocidos

- En la versión de OpenWrt para VirtualBox que usamos en el laboratorio, nunca ejecutes `reboot`. Usa `halt` en su lugar. De lo contrario, la máquina montará el sistema de ficheros en modo solo lectura
- Para borrarlo todo y volver a empezar, elimina los directorios `~/VirtualBox` y `/var/tmp/tulogin`
Pero recuerda volver a indicar `/var/tmp/tulogin` en
Archivo|Preferencias|General|
Carpeta predeterminada de máquinas

Configuración del teclado

- El teclado habitual en los PCs españoles es el pc105 (O el pc102 si no tiene teclas *menú*, *windows*)
- El equivalente en los PCs estadounidenses es el pc104 y el pc101, respectivamente
- Programadores y usuarios normalmente trabaja con versiones del SO adaptadas a su idioma
- Un administrador frecuentemente se encontrará con un SSOO en inglés
 - Normalmente podrá configurarlo para que admita su propio idioma (si no en menús y documentación, sí en la configuración del teclado)
 - Pero mientras lo configura, tendrá que saber manejarse mínimamente con el teclado desconfigurado

Si el ordenador tiene X Window (Gráficos), podemos configurarlo con

- `setxkbmap us` Fija el teclado en la disposición pc104
- `setxkbmap es` Fija el teclado en la disposición pc105

En Debian/Ubuntu podemos usar

- `dpkg-reconfigure console-setup`
- O más globalmente
`sudo dpkg-reconfigure locales`

En gnome:

- Sistema | Preferencias | Teclado | Distribuciones |
| Añadir | Español | Subir (hasta colocarlo el primero)

Necesitarás cambiar esto por ejemplo si entras desde casa al laboratorio por VNC y lanzas una máquina virtual

En OpenWrt esto no está disponible, la mejor opción es entrar por ssh (lo que exige que la red ya funcione)

Si nuestro teclado es español, pero el sistema operativo espera un teclado norteamericano:

Obtener	Pulsar

Esc	Esc
:	Ñ
;	ñ
/	-
!	!
	shift Ç
'	, (apóstofre, coma)