

Sistemas invitados

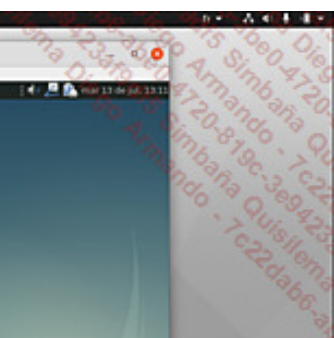
1. Hipervisor y adiciones

Por sistema invitado, entendemos el sistema operativo instalado en la máquina virtual. Dependiendo del sistema operativo anfitrión, habrá disponibles distintos programas. Solamente para Linux, encontrará:

- ✓ **Qemu**, todavía presente en el mercado, pero muy lento,
- ✓ **KVM**, descendiente de qemu, pero que, al usar virtio y un acceso al núcleo, propone un rendimiento nativo,
- ✓ **Virtualbox**, de Oracle, que hace lo mismo, pero disponible de manera idéntica en Windows y macOS,
- ✓ **VMWare Workstation**, de VMWare, que también es multiplataforma,
- ✓ **Xen**, hipervisor de tipo 1, pero que necesita una instalación particular.

Casi nunca se piensa en ello, pero Qemu y KVM disponen, por supuesto, de comandos accesibles en shell, pero también, gracias a libvirt, dispone de una interfaz gráfica muy eficiente, que permite incluso administrar máquinas virtuales de otros hipervisores (como Virtualbox, Xen, VMware...).

El autor, por ejemplo, usa muchas máquinas virtuales (hasta ocho), en Virtualbox o KVM, desde macOS o una máquina virtual Linux. Esto provoca, a veces, situaciones interesantes en las que Virtualbox ejecuta una máquina virtual que a su vez opera KVM que ejecuta un sistema invitado...



Para obtener un funcionamiento óptimo de los sistemas operativos invitados, hay algunos elementos que se tienen que tener en cuenta:

- La memoria deber ser la necesaria para el sistema y para el entorno gráfico. Un anfitrión con 16 GB de memoria puede hacer funcionar máquinas virtuales de 4 GB (para Gnome o KDE). El autor ha podido arrancar 6 máquinas virtuales de 1 a 2 GB sin problemas.
- El número de procesadores debería de ser inferior o igual al número de procesadores del sistema anfitrión, aunque este último puede gestionar correctamente los recursos.
- Para las máquinas virtuales profesionales, los discos deberían estar provisionados en el momento de su creación, y sobre soportes rápidos.
- El piloto gráfico debe ser el aconsejado por defecto. Algunos hipervisores proponen el soporte de la 3D acelerada, para ejecutar algunos juegos o herramientas gráficas de mucho rendimiento.
- Y sobre todo, los "guest additions" deben de ser añadidos si se encuentran disponibles en el sistema invitado.

Un sistema invitado no sabe que se está ejecutando en una máquina virtual. Se comporta como si estuviera instalado en una máquina física corriente. La mayoría de los periféricos presentados por el hipervisor están emulados. El sistema invitado enumera una lista de material del BIOS, los datos DMI o UEFI. Esta "falsa" información la da el hipervisor que ha configurado la "falsa" BIOS de la máquina virtual en consecuencia. El núcleo va a intentar cargar los drivers deducidos del material encontrado. Por ejemplo, si la tarjeta de red emulada es de tipo Intel Pro 1000 Ethernet card, el núcleo Linux cargará el módulo e1000 y el sistema invitado lo considerará como una tarjeta ethernet verdadera. Sin embargo, esta tarjeta la emula el hipervisor, causando una sobrecarga en el anfitrión e impactando el rendimiento de la máquina virtual.

A veces o el núcleo no dispone de los pilotos por defecto, o el hipervisor propone métodos de paravirtualización y, por lo tanto, material "virtual" optimizado hasta ofrecer, incluso, casi el mismo rendimiento que el sistema anfitrión. En ese caso, a menudo hay que instalar drivers adicionales.

Las adiciones de los drivers y los servicios proveídos, generalmente, por el hipervisor, se instalan en el sistema invitado para mejorar su funcionamiento y rendimiento. Se trata normalmente de módulos del núcleo para la gestión de la comunicación con el hipervisor o el sistema anfitrión: los drivers gráficos, la gestión de la memoria y del procesador, los pilotos de las tarjetas de red virtuales, etc. A veces se añaden algunos servicios. Generalmente se proveen pilotos X11 o Wayland.

Las adiciones a veces están presentes de manera nativa en el sistema invitado, o en forma de paquetes, o con el núcleo linux vanilla, como es el caso para los módulos de tipo virtio. En Ubuntu por ejemplo, los paquetes están disponibles para los sistemas invitados en Virtualbox:

```
$ dpkg -l | grep -i "^ii.*virtualbox" | awk '{ print $2}'
virtualbox-guest-dkms
virtualbox-guest-utils
virtualbox-guest-x11
```

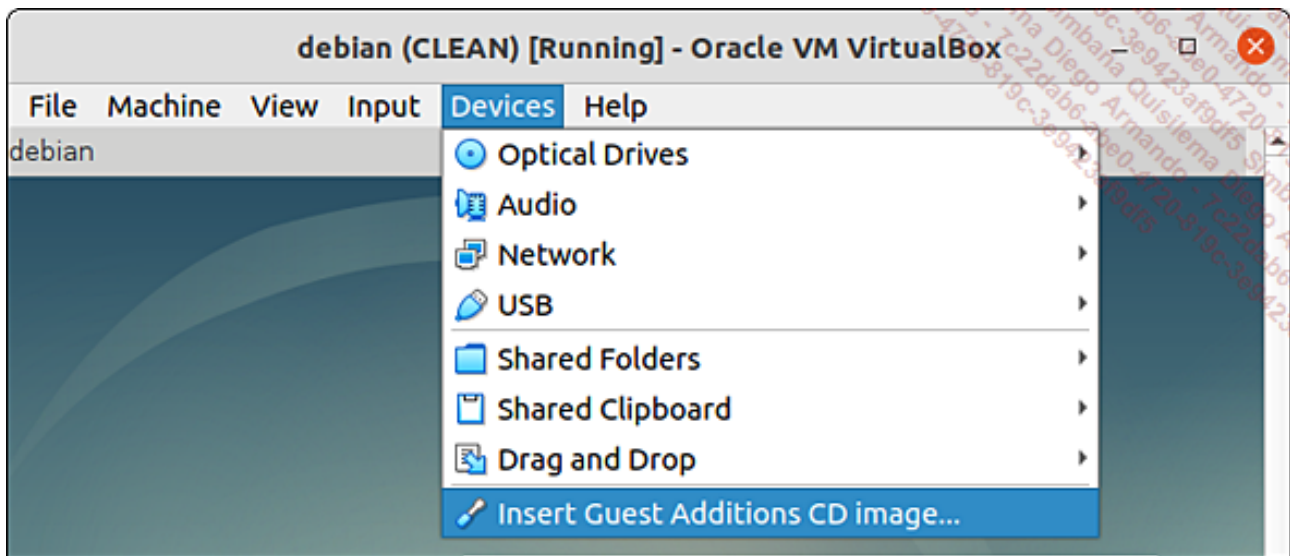
Para KVM y virtio (y los otros hipervisores que lo soportan), encontraremos módulos que permitirán una comunicación de tipo FIFO directa entre el núcleo del anfitrión y el del sistema invitado en `/lib/modules/$(uname -r)`. Los módulos serán, de hecho, cargados automáticamente por el núcleo, si es necesario:

```
seb@ubuntu:/lib/modules/5.3.0-29-generic$ find -type f -iname "*virtio*.ko"
./kernel/net/vmw_vsock/vmw_vsock_virtio_transport.ko
./kernel/net/vmw_vsock/vmw_vsock_virtio_transport_common.ko
./kernel/net/9p/9pnet_virtio.ko
./kernel/drivers/block/virtio_blk.ko
./kernel/drivers/net/virtio_net.ko
./kernel/drivers/net/caif/caif_virtio.ko
./kernel/drivers/scsi/virtio_scsi.ko
./kernel/drivers/crypto/virtio/virtio_crypto.ko
./kernel/drivers/gpu/drm/virtio/virtio-gpu.ko
./kernel/drivers/nvdim/virtio_pmem.ko
./kernel/drivers/nvdim/nd_virtio.ko
./kernel/drivers/char/hw_random/virtio-rng.ko
./kernel/drivers/rpmsg/virtio_rpmsg_bus.ko
./kernel/drivers/virtio/virtio_input.ko
```

Con respecto a VMWare, cada distribución proveerá las open-vm-tools, como en Ubuntu:

```
# apt-cache search open-vm-tools
open-vm-tools - Open VMware Tools for virtual machines hosted on VMware (CLI)
open-vm-tools-desktop - Open VMware Tools for virtual machines hosted on VMware (GUI)
open-vm-tools-dev - Open VMware Tools for virtual machines hosted on VMware
(development)
```

Si la distribución no facilita los pilotos o los paquetes necesarios, el editor de la solución de virtualización proveerá generalmente las adiciones en forma de imagen ISO que se podrá montar directamente dentro de la máquina virtual, como aquí en Virtualbox:



Instalar las adiciones de Virtualbox

Las distribuciones 31, RHEL8 o CentOS8 no incluyen las adiciones de Virtualbox por defecto, habrá que instalarlas en consola. Para ello, primero tendrá que instalar las herramientas de desarrollo y las fuentes.

He aquí el procedimiento.

➔ Solamente en RHEL8 o CentOS 8, hay que instalar el depósito:

```
# yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

➔ Después se instalarán las dependencias:

```
# yum install gcc kernel-devel kernel-headers dkms make bzip2 perl libxcrypt-compat
```

➔ Se indicará dónde se sitúan las fuentes del núcleo actual:

```
# export KERN_DIR=/usr/src/kernels/$(uname -r)
```

➔ Después de haber montado el soporte de las adiciones, se lanzará el script de instalación:

```
# ./VBoxLinuxAdditions.run
```

➔ Finalmente, se reinicia el sistema:

```
# reboot
```

En el cloud, las imágenes propuestas (los AMI en AWS, por ejemplo) ya están modificadas y presentan los pilotos y las adiciones necesarias por defecto.

2. El acceso a la consola o a la interfaz

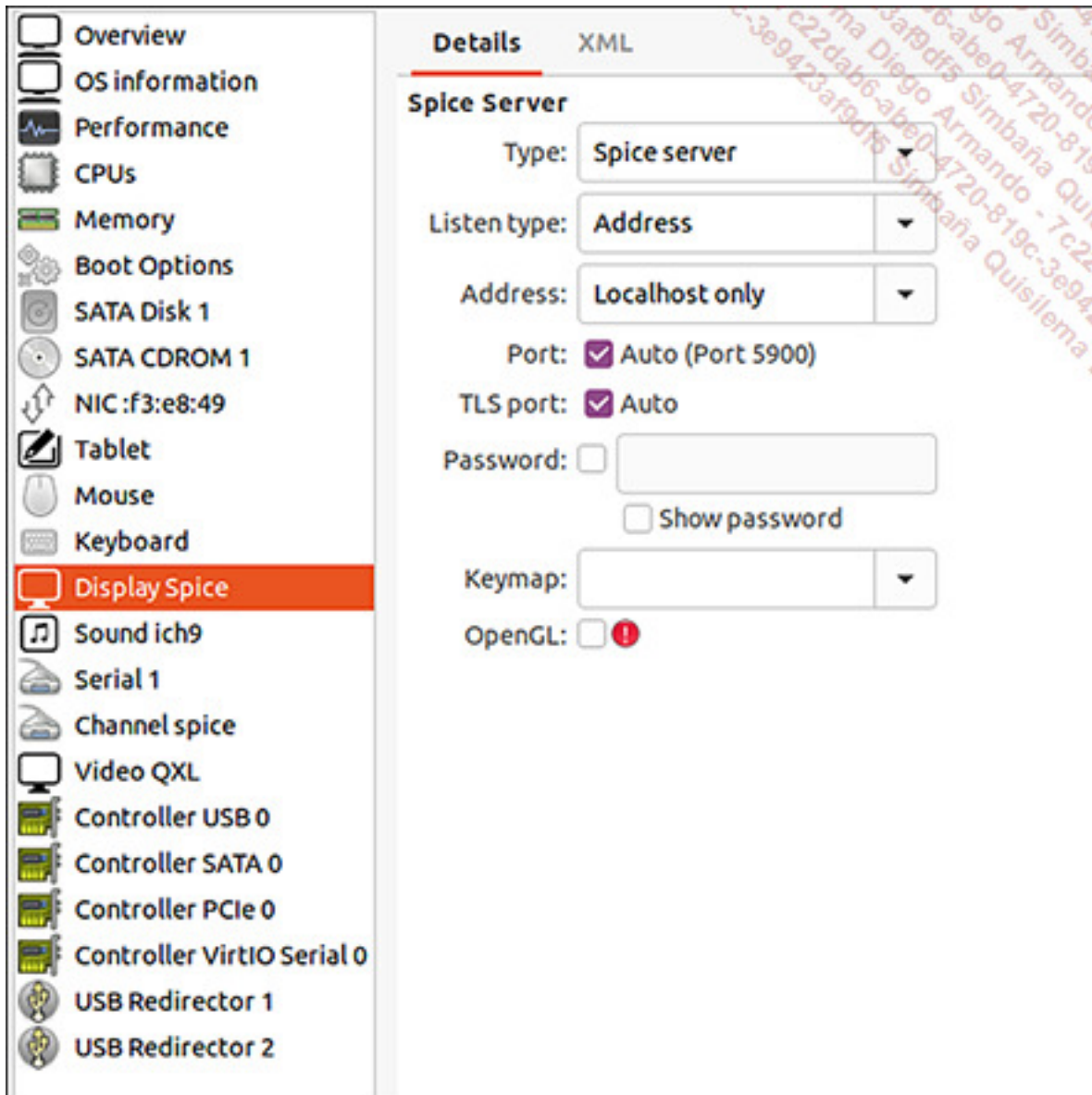
a. Spice y KVM

Los usuarios acostumbrados a utilizar VMWare o Virtualbox no suelen tomar en cuenta el acceso a la consola o a la interfaz gráfica, porque se accede a la máquina virtual a través de una ventana estándar o en modo pantalla completa. Sin embargo, esto no tiene nada de sencillo. El hipervisor intercepta la salida emulada del piloto gráfico del sistema invitado para transcribirlo en una ventana del sistema anfitrión. Esta posibilidad no existe siempre. Por ejemplo AWS no lo permite directamente. Una de las razones es que el cloud es vehiculado por Internet y, según el ancho de banda de su conexión, una visualización del escritorio en HD y en tiempo real podría ser muy lento. AWS propone comandos que le permiten obtener una imagen, en JPEG, del escritorio.

En la práctica, los hipervisores proponen generalmente un acceso a la consola a través de los protocolos RDP, VNC o Spice. RDP y VNC ya se han visto en el capítulo X Window. Spice es el protocolo propuesto por defecto por KVM. No solamente se encarga de la visualización, sino que también propone interfaces de paravirtualización, asociadas a virtio.

En las propiedades de una máquina virtual KVM creada desde Virt Manager, encontrará Spice (KVM también soporta VNC). Pero si observa con más detalle verá también que el

protocolo Spice se puede paravirtualizar gracias a virtio, especialmente para las interfaces USB, los puertos de comunicación, el audio bidireccional o las pantallas múltiples usados por el cliente Spice para dialogar y montar en caliente distintos periféricos.



KVM usa Spice para la visualización remota

El archivo de configuración de esta máquina virtual, situado en `/etc/libvirt/qemu/fedora31.xml`, contiene las líneas siguientes, que corresponden a las posibilidades ofrecidas por el protocolo Spice:

```

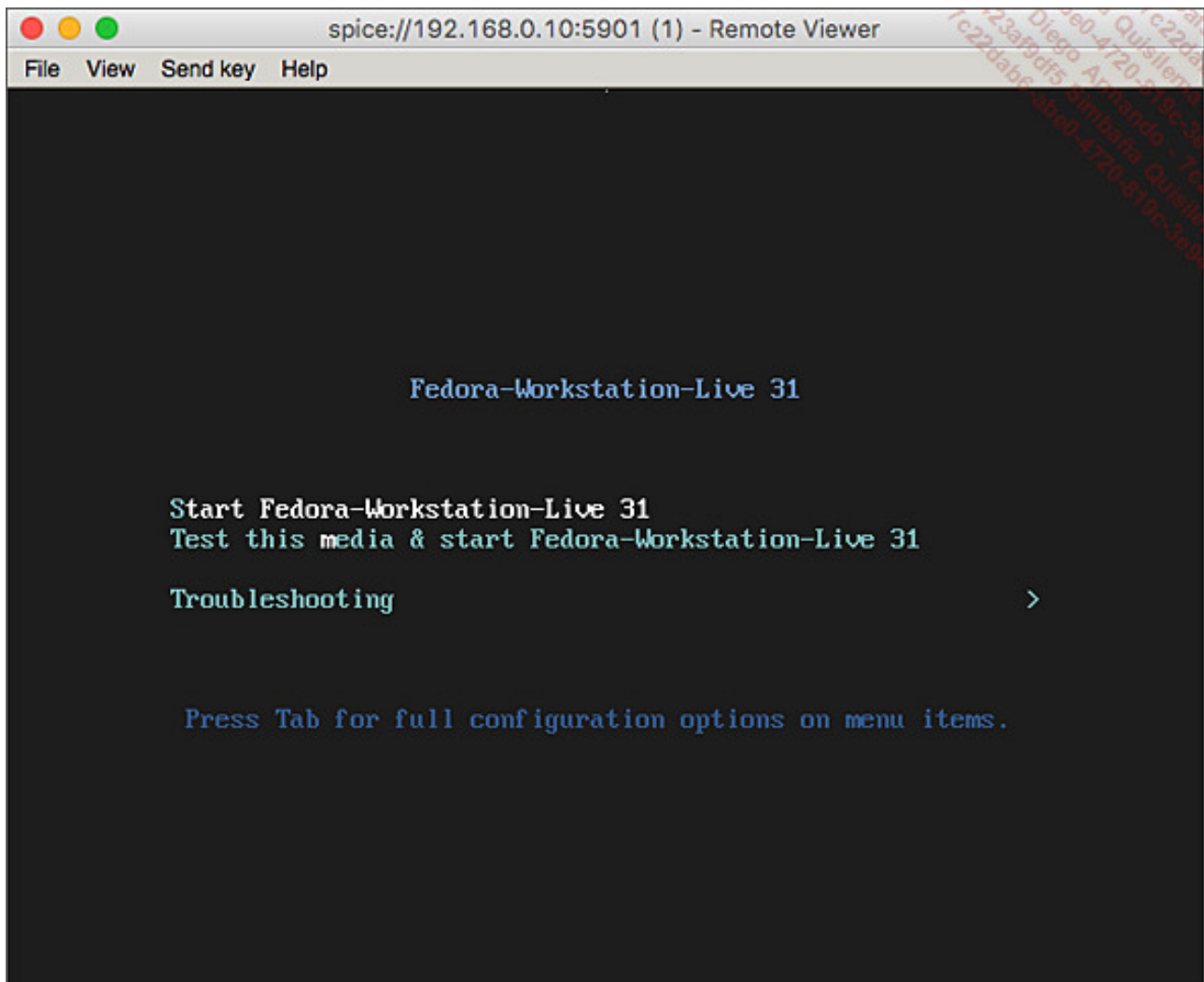
<channel type='spicevmc'>
  <target type='virtio' name='com.redhat.spice.0' />
  <address type='virtio-serial' controller='0' bus='0' port='2' />
</channel>
<graphics type='spice' autoport='yes'>
  <listen type='address' />
  <image compression='off' />
</graphics>
<redirdev bus='usb' type='spicevmc'>
  <address type='usb' bus='0' port='2' />
</redirdev>
<redirdev bus='usb' type='spicevmc'>
  <address type='usb' bus='0' port='3' />
</redirdev>

```

b. Cliente Spice

Los clientes Spice están disponibles en <https://www.spice-space.org/download.html>. Los hay para la mayoría de los sistemas operativos, incluso Android. Virt Manager incluye un cliente integrado, lo que le permite obtener la visualización de la consola gráfica.

En el cliente hay que especificar el protocolo. En el caso del ejemplo de la máquina virtual Fedora, la cadena de conexión sería: `spice://192.168.0.10:5901`



Visualización de una máquina virtual KVM a través de Spice

c. Otros casos

Como ya se indicó, AWS (todavía) no propone la visualización. Google Cloud sí la propone desde 2019 gracias al protocolo RDP, así como, naturalmente, Microsoft Azure.

De todas maneras, en el cloud, se instalan servidores y, en nuestro caso, sistemas invitados Linux. Desde el inicio de este libro, ha recibido formación para utilizar Linux desde la línea de comandos y ha podido descubrir que, a parte del entorno de la ofimática, no se necesita para nada la interfaz gráfica.

El acceso a la consola se realiza a través del protocolo SSH. Puede volver al ejemplo de instanciación de un servidor Ubuntu en AWS: se le solicita la clave pública asociada a la

clave privada que utilizará para conectarse al servidor.

Si realmente quiere usar un entorno gráfico en una instancia Linux en el cloud, deberá instalar un servidor RDP o VNC.