

# KVM y libvirt

Con la combinación de KVM y libvirt, tiene una manera fácil de crear y administrar máquinas virtuales.

Según la página oficial [homepage](#), libvirt es: Un kit de herramientas para interactuar con las capacidades de virtualización de versiones recientes de Linux (y otros sistemas operativos). Proporciona gestión de máquinas virtuales, redes virtuales y almacenamiento; tanto locales como remotos. Dado que libvirt actúa como un intermediario entre un hipervisor y las aplicaciones cliente, debe tener instalado un hipervisor compatible. Algunos ejemplos son: KVM / QEMU, Virtualbox, Xen y VMware ESX.

**Quote:** KVM (para Máquina Virtual basada en Kernel) es una solución de virtualización completa para Linux en hardware x86 que contiene extensiones de virtualización (Intel VT o AMD V). Consiste en un módulo de kernel que se puede cargar, `kvm.ko`, que proporciona la infraestructura de virtualización central y un módulo específico del procesador, `kvm-intel.ko` o `kvm-amd.ko`.

## Instalación

Libvirt se puede instalar utilizando un script slackbuild de [slackbuilds.org](#). Proporciona un demonio que interactúa entre aplicaciones y máquinas virtuales. También proporciona un shell de línea de comandos, `virsh`, que se puede usar para administrar máquinas virtuales y configurar el entorno libvirt. `Virsh` también se puede usar en shell scripts para iniciar y detener máquinas virtuales.

El kernel slackware tiene el módulo KVM habilitado. El script de inicio de libvirt verificará la CPU y modificará el controlador correcto. Las herramientas de espacio de usuario se suministran con QEMU, que está disponible en [slackbuilds.org](#). Anteriormente, se utilizaba una QEMU modificada, `qemu-kvm`. Sin embargo, desde la versión 1.3, QEMU incorpora esos cambios y `qemu-kvm` está restringido.

Una herramienta de gestión de escritorio gráfica, `virt-manager`, también está disponible en [slackbuilds.org](#). Esto proporciona una visión general de todas las máquinas virtuales y tiene un asistente agradable para crear nuevas máquinas virtuales de una manera fácil.

## Configuration

### Inicio automático

Si desea que el demonio libvirt se inicie automáticamente, agregue la siguiente sección a `/etc/rc.d/rc.local` :

```
# start libvirt
if [ -x /etc/rc.d/rc.libvirt ]; then
    /etc/rc.d/rc.libvirt start
fi
```

Asegúrese de que `/etc/rc.d/rc.libvirt` sea ejecutable.

## Gestión de grupos de almacenamiento

El almacenamiento en libvirt se maneja en términos de *grupos de almacenamiento* y *volúmenes de almacenamiento*. Un *grupo* es un contenedor genérico para varios objetos de almacenamiento. Puede ser un directorio local, una partición física o un recurso compartido de red. Un *almacenamiento volumen* es la representación virtual de un disco para un sistema invitado. En el invitado, este volumen se ve como un disco local. Una imagen iso de un cd o dvd de instalación también se considera un volumen.

Cuando se instala libvirt, se crea un grupo de almacenamiento predeterminado (denominado *default*) con el directorio local `/var/lib/libvirt/images`. Cualquier volumen recién creado se crea en este directorio.

### Cree un nuevo grupo de almacenamiento basado en directorio usando virsh

Los comandos de Virsh se pueden pasar como parámetros a *virsh* en la línea de comandos, o puede iniciar un shell de virsh interactivo llamando a *virsh* sin parámetros:

```
# virsh
Welcome to virsh, the virtualization interactive terminal.

Type:  'help' for help with commands
       'quit' to quit

virsh #
```

Para crear un nuevo grupo de almacenamiento basado en directorio, primero asegúrese de que el directorio de destino existe. Luego use el comando `pool-define-as`. La sintaxis básica para este comando es: `pool-define-as <pool-name> dir - - - - <directory-name>`. Por ejemplo, para crear el grupo *discos* para el directorio `/srv/virtualmachines/disks`, use el siguiente comando:

```
# virsh pool-define-as disks dir - - - - /srv/virtualmachines/disks
Pool disks defined
```

Para obtener ejemplos más complejos de este comando, consulte la página del manual de virsh.

Verifique que la agrupación exista con el comando `pool-list`. La opción `-all` muestra grupos activos e inactivos:

```
# virsh pool-list --all
Name                State      Autostart
-----
default             active    yes
disks                inactive  no
```

Ahora, cree el grupo real con el comando `pool-build` :

```
# virsh pool-build disks
Pool disks built
```

Cuando se construye pool, se puede iniciar con el comando `pool-start` :

```
# virsh pool-start disks
Pool disks started
```

Ahora se puede utilizar pool. En este punto, el grupo siempre debe iniciarse manualmente. Para que libvirt inicie el grupo cuando se inicie el demonio, debe verificar el indicador de inicio automático con el comando `grupo-inicio` automático :

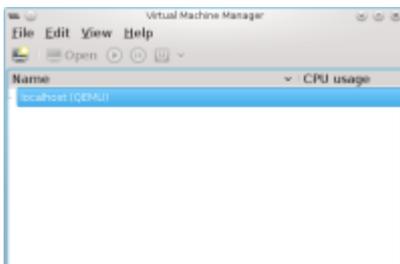
```
# virsh pool-autostart disks
Pool disks marked as autostarted
```

Muestra información sobre el grupo con el comando `pool-info` :

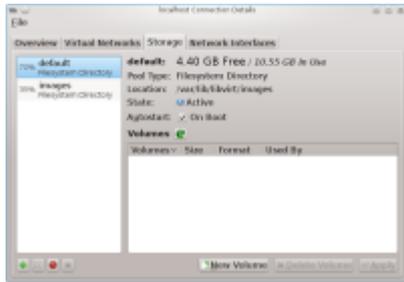
```
# virsh pool-info disks
Name:          disks
UUID:          4ae08c3d-4622-9f2a-cfa9-9dea4d1eb465
State:         running
Persistent:    yes
Autostart:     yes
Capacity:      697.92 GiB
Allocation:    250.89 GiB
Available:     447.04 GiB
```

## Cree un nuevo grupo de almacenamiento basado en directorio usando `virt-manager`

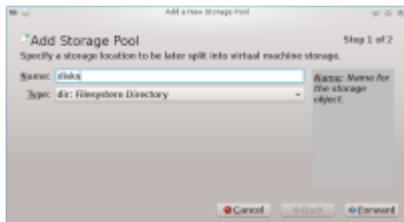
En primer lugar, asegúrese de que el directorio de destino existe. Entonces empieza `virt-manager` .



Seleccione la máquina host (el valor predeterminado es `localhost` ). Seleccione *Editar*, *Detalles de conexión* en el menú, o haga clic con el botón derecho en la máquina y seleccione *Detalles* , o haga doble clic en la máquina. Aparece la ventana *Detalles de la conexión* . Seleccione la pestaña *Almacenamiento* .



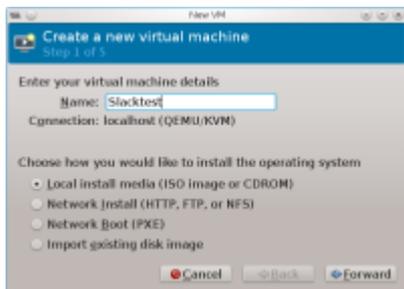
Presione el botón + en la parte inferior izquierda. Aparece la ventana *Agregar grupo de almacenamiento* .



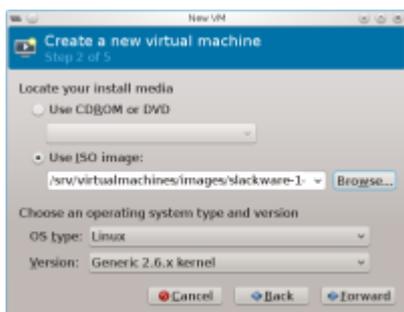
Ingrese el nombre del nuevo grupo. El tipo predeterminado es *dir* , que es el tipo correcto. Presione **Forward** e ingrese el directorio del sistema en el campo de entrada *Target Path* . Presione **Finalizar** para crear el grupo.

## Creando una nueva máquina virtual usando virt-manager

Seleccione el host en el que desea crear la nueva máquina virtual. Esto será *localhost* por defecto. Haga clic con el botón derecho en el host y seleccione *Nuevo* , o haga clic en el botón **Crear** para iniciar el asistente de creación.



Paso 1: Asigne un nombre a la nueva máquina, seleccione el método de instalación del sistema operativo y presione **Forward**.



Paso 2: Dependiendo del método de instalación, se le pedirá que ingrese más detalles, como el nombre de la imagen de DVD iso utilizada.

También elige el tipo de sistema operativo y la versión. Al elegir *Linux*, elija *Mostrar todas las opciones de SO* en el indicador de Versión para obtener más opciones. Al elegir *kernel genérico 2.6.x*, el asistente asumirá un disco duro tipo IDE. Puedes cambiar esto en el paso 5.

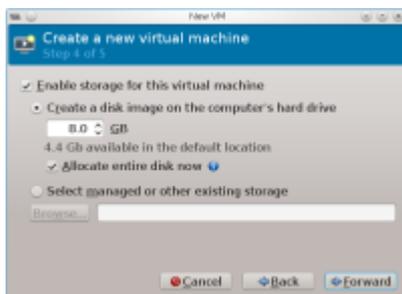


Si elige *kernel genérico 2.6.25 o posterior con virtio*, la nueva máquina utilizará el controlador del kernel de virtio para emular un controlador de disco duro. La imagen de instalación de Slackware se iniciará, pero debe crear un initrd con los controladores del kernel de Virtio incluidos, de lo contrario su nueva máquina no arrancará.

Paso 3: Seleccione la cantidad de RAM y CPUs.



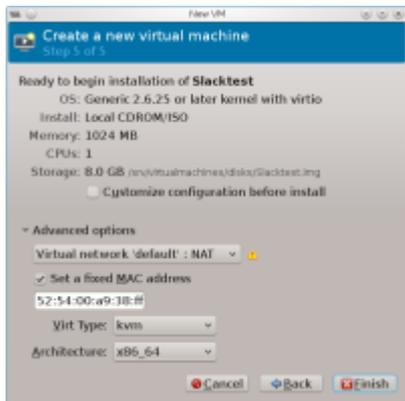
Paso 4: Configurar el almacenamiento para la nueva máquina. Esta opción está seleccionada de forma predeterminada. Si deselecciona esta opción, tendrá una máquina sin disco que aún puede iniciar distribuciones en vivo.



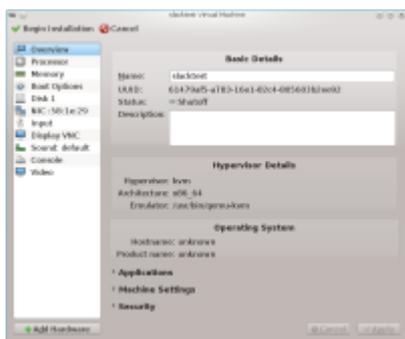
Puede dejar que el asistente cree una nueva imagen de disco sobre la marcha. Ingrese el tamaño y se creará una nueva imagen de disco (tipo *raw*) en el grupo de almacenamiento predeterminado. Si selecciona *Asignar todo el disco ahora*, el tamaño completo del nuevo disco se asigna de antemano. De lo contrario, se crea una imagen más pequeña que crecerá cuando los datos se escriban en el disco.

Como alternativa, puede seleccionar otra imagen de disco (volumen) de un grupo de almacenamiento. También puede usar esta opción para crear un nuevo volumen de almacenamiento. Este método le dará más flexibilidad, ya que tiene más opciones sobre el tipo de disco que puede crear. Por último, puede usar esta opción para explorar el sistema de archivos local y seleccionar una imagen de disco que creó anteriormente.

Paso 5: Esto le dará una visión general de la nueva máquina



Si marca *Personalizar la configuración antes de instalar* , aparecerá la página de configuración de hardware de virt-manager. Aquí puede hacer cambios en la máquina antes de instalar.



La opción de red predeterminada en el paso 5 es *Red virtual 'default': NAT* . Esta red virtual utilizará el dispositivo *virbr0* bridge para conectarse al host local mediante NAT. Libvirt iniciará dnsmasq para proporcionar DHCP para esta red.

Finalmente, puede cambiar el tipo de máquina virtual (KVM o QEMU) y la arquitectura (x86\_64, i686, arm, etc.). Presione Finalizar para crear la máquina virtual. Será creado y arrancará.

## Redes

Una máquina virtual se puede configurar de dos maneras: conectada a una red virtual o conectada a un dispositivo físico compartido (puente).

### Redes virtuales

Las redes virtuales son implementadas por libvirt en forma de conmutadores virtuales. Una máquina virtual tendrá su tarjeta de red conectada a este conmutador virtual. El conmutador virtual aparece en el host como una interfaz de red.

La red virtual puede operar en 3 configuraciones diferentes:

1. NAT: el conmutador virtual está conectado a la LAN del host en modo NAT. El host puede conectarse a las máquinas virtuales, pero otras máquinas en la red del host no pueden conectarse. Esta es la configuración por defecto.
2. Enrutado: el conmutador virtual está conectado a la LAN del host sin NAT.
3. Aislado: el conmutador virtual no está conectado al host. Las máquinas virtuales pueden verse

entre sí y al host, pero el tráfico de la red no pasa fuera del host.

## Dnsmasq y iptables

DHCP y DNS en las redes virtuales son manejados por Dnsmasq. Para cada red virtual que se inicia, libvirt inicia una instancia separada de Dnsmasq.

Al iniciar una red virtual, libvirt también agregará reglas de iptables para manejar el enrutamiento y NAT entre el host y la red virtual. También habilitará `ip_forward`.

## Red por defecto

Después de la instalación, una red (apropiadamente llamada *por defecto* ) ya está creada. Se configura como tipo NAT y esta es la red predeterminada que está vinculada a nuevas máquinas virtuales. La red está configurada para iniciarse automáticamente cuando se inicia libvirt, por lo que está disponible inmediatamente cuando se ejecuta libvirt.

La red es visible en el host como bridge `virbr0`.



No agregue interfaces físicas a este bridge. Es manejado por libvirt.

## Creación y mantenimiento

Las redes virtuales están visibles en la pestaña *Redes virtuales* de la ventana *Detalles de conexión del host* en Virtual Machine Manager. La tecla + se puede usar para agregar una nueva red virtual. Ingrese el nombre, la dirección de red IPv4, el rango DHCP y el tipo de red.

Una vez que se crea la red virtual, se puede utilizar en las pantallas de mantenimiento de la máquina virtual.

## Carpetas compartidas usando VirtFS

Es posible compartir carpetas entre el sistema huésped y el host mediante VirtFS (carpeta de Plan 9 compartidas sobre Virtio). Esto está disponible desde la versión 2.6.36 del kernel en adelante. VirtFS es un sistema de archivos de acceso directo, lo que significa que se puede acceder directamente a un directorio en el host desde el invitado a través de la capa de virtualización.

## Preparar kernel host

Asegúrese de que las siguientes opciones estén configuradas en el núcleo del host:

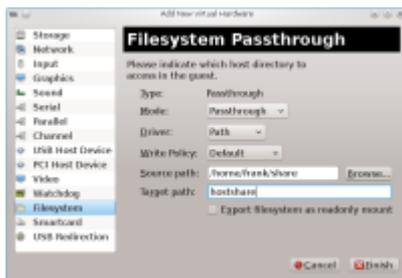
```
CONFIG_NET_9P=y
CONFIG_NET_9P_VIRTIO=y
```

```
CONFIG_NET_9P_DEBUG=y (Optional)
CONFIG_9P_FS=y
CONFIG_9P_FS_POSIX_ACL=y
```

En el sistema invitado, se necesitan los módulos `9p` , `9pnet` y `9pnet_virtio` . Estos deberían estar disponibles en el kernel estándar de Slackware.

## Configuración de invitado

En virt-manager, edite la máquina virtual invitada y agregue nuevo hardware. Seleccione *Sistema de archivos* y complete los campos requeridos como se muestra a continuación.



\* Modo. Selecciona uno de los siguientes:

- Passthrough: se accede al recurso compartido de host con los permisos del usuario invitado.
  - Asignado: se accede al recurso compartido de host con los permisos del hipervisor (proceso QEMU).
  - Squash: similar a *passthrough* excepto que se ignoran las fallas de operaciones privilegiadas como *chown* . Esto hace que se pueda utilizar un modo similar al paso a través de personas que ejecutan el hipervisor como no root.
- Controlador: use *Ruta* .
- Política de escritura (solo disponible para el controlador ruta ): use *Predeterminado* .
- Ruta de origen = directorio en el host que se comparte.
- Ruta de destino = etiqueta de montaje que está disponible en el sistema huésped. Esto no tiene que ser una ruta existente.

La opción *Exportar sistema de archivos como montaje de solo lectura* hace lo que sugiere.

## Mounting the share

Para montar el sistema de archivos con la etiqueta `hostshare` en el invitado en `/mnt/share` , inicie sesión en el invitado y use:

```
# mount -t 9p -o trans=virtio,version=9p2000.L hostshare /mnt/share
```

Ahora la carpeta `/mnt/share` está disponible y los cambios son visibles en el host y el invitado. Cuando se produzcan errores de permiso, intente compartir el directorio del host con un modo diferente.

## Acceso remoto



Nota del Autor. Trabajo en progreso

## Temas avanzados

### Monte la imagen qcow usando nbd

Las imágenes de disco sin procesar se pueden montar fuera de la máquina virtual mediante un dispositivo de bucle invertido. Para montar otros tipos de imágenes como qcow, se puede usar el comando `qemu-nbd`, que viene con `qemu-kvm`. Se basa en el módulo del núcleo `nbd` (dispositivo de bloqueo de red).

Comience cargando el módulo del kernel. El único parámetro es el número máximo de particiones a las que se puede acceder. Si se omite este parámetro, el valor predeterminado es 0, lo que significa que no se asignarán particiones.

```
# modprobe nbd max_part=8
```

Esto creará varios dispositivos nuevos `/dev/nbdxx`. Ahora la imagen del disco se puede conectar a uno de ellos:

```
# qemu-nbd -c /dev/nbd0 slackware.img
```

Esto creará dispositivos adicionales `/dev/nbd0pxx` para las particiones en el disco. Las particiones se numeran secuencialmente comenzando con 1. Puede usar el dispositivo `nbd0` para acceder a todo el disco, o los dispositivos `nbd0pxx` para acceder a las particiones:

```
# fdisk /dev/nbd0  
# mount /dev/nbd0p1 /mnt/hd
```



Asegúrese de que la máquina virtual no se esté ejecutando cuando monte la imagen de disco. Montar el disco de una máquina en marcha lo dañará.

Para eliminar la conexión:

```
# qemu-nbd -d /dev/nbd0
```

### Configurar el arranque PXE en libvirt

Para habilitar el arranque PXE para las máquinas invitadas, se necesitan un servidor de arranque PXE y un servidor TFTP. Libvirt se puede configurar para manejar tanto internamente. Estas opciones de configuración no están disponibles en `virt-manager`, por lo que se debe usar `virsh` para configurar esto.

1. Cree un directorio `/tftpboot` y rellénelo con los archivos necesarios para el servicio de

arranque tftp. Consulte el artículo [PXE: Instalación de Slackware en la red](#) por AlienBOB para obtener más detalles.

2. Detenga la red predeterminada y edite la definición de red:

```
# virsh net-destroy default
# virsh net-edit default
```

3. Esto abrirá la configuración de red en una sesión vi. Agregue los parámetros *tftp* y *bootp* en la sección *ip* y guarde el archivo:

```
<ip address='192.168.122.1' netmask='255.255.255.0'>
  <tftp root='/tftpboot' />
  <dhcp>
    <range start='192.168.122.2' end='192.168.122.254' />
    <bootp file='pxelinux.0' />
  </dhcp>
</ip>
```

4. Ahora reinicie la red:

```
# virsh net-start default
```

Ahora el servidor DHCP libvirt permitirá a los invitados arrancar con PXE.

## Troubleshooting

### Eliminar solicitud de contraseña en virt-manager

Cuando inicie virt-manager como un usuario regular, es posible que se le solicite la contraseña de root, incluso cuando haya configurado los permisos de socket de Unix correctos (notificación: “la política del sistema impide la administración de sistemas virtualizados locales”). Esto se debe a que las versiones anteriores de libvirt usaban PolicyKit de forma predeterminada. Desactive el uso de PolicyKit editando `/etc/libvirt/libvirtd.conf`. Descomente las siguientes opciones y cámbielas a ninguna :

```
auth_unix_ro = "none"
auth_unix_rw = "none"
```

### Mejorar el movimiento del ratón

En el modo gráfico, el movimiento del mouse puede ser errático y difícil de cambiar en la configuración de su VM. Para resolver esto, agregue una tableta virtual.

En *virt-manager*, abra la máquina virtual haciendo doble clic en la máquina y seleccione la pantalla de información del hardware (el botón azul **i**). Ahora, presione + **Add Hardware** y seleccione *Input*. Seleccione *Type EvTouch USB Graphics Tablet* y presione **Finish**. La próxima vez que se inicie la máquina virtual, el movimiento del mouse se sincronizará con el cursor del escritorio.

## Cambiar la resolución de la pantalla a más de 1024x768

La tarjeta de video emulada predeterminada es de tipo Cirrus. Esto tiene una resolución máxima de 1024x768. El tipo *vga* puede lograr una resolución más alta, pero para que funcione, la configuración de X en el sistema operativo invitado también debe cambiarse.

Para cambiar esto, abra la máquina virtual y seleccione la pantalla de información de hardware. Seleccione la tarjeta de video y cambie el tipo a *vga* .

Iniciar la máquina virtual. Si existe un archivo `/etc/X11/xorg.conf` , cámbielo. De lo contrario, cree un nuevo archivo de texto en `/etc/X11/xorg.conf.d` con la extensión `conf` , por ejemplo `/etc/X11/xorg.conf.d/monitor.conf` .

Agrega o cambia las siguientes secciones:

```
Section "Monitor"
    Identifier "Monitor0"
    HorizSync 30 - 80
    VertRefresh 40 - 90
EndSection

Section "Device"
    Identifier "Card0"
    Driver "vesa"
Endsection

Section "Screen"
    Identifier "Screen0"
    Device "Card0"
    Monitor "Monitor0"
    SubSection "Display"
        Viewport 0 0
        Modes "1440x900"
    EndSubSection
EndSection
```

Puede cambiar la resolución de la pantalla (la línea *Modes* ) según sus necesidades.

## Recursos

- Páginas oficiales de [libvirt](#), [virt-manager](#), [QEMU](#), [KVM](#).
- Red Hat [Guía de administración de virtualización](#).

## Sources

- Esta página escrita originalmente por [Frank Donkers](#)

- Traducido por: [Victor](#) 2019/02/09 00:53 (UTC)

[howtos](#), [kvm](#), [libvirt](#), [virtualization](#), [virt-manager](#), [qemu](#), [author](#) [fdonkers](#)

From:

<https://docs.slackware.com/> - **SlackDocs**

Permanent link:

[https://docs.slackware.com/es:howtos:general\\_admin:kvm\\_libvirt](https://docs.slackware.com/es:howtos:general_admin:kvm_libvirt)

Last update: **2019/02/09 00:56 (UTC)**

