

INSTALACIÓN DE KVM



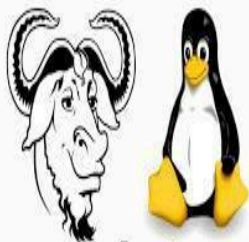
Gestor de máquinas virtuales

Archivo Editar Vista Ayuda



Nombre

QEMU/KVM



GNU/Linux

Nueva MV

Creación de una máquina virtual nueva
Etapa 1 of 5

Conexión: QEMU/KVM

Elija cómo le gustaría instalar el sistema operativo

- ☒ Medio de instalación local (Imagen ISO ó CDROM)
- ☐ Instalación de red (HTTP, HTTPS o FTP)
- ☐ Arranque por Red (PXE)
- ☐ Importar imagen de disco existente

Atrás

Adelante

DIGITAL
EDUCAS

Virtualización

#DigitalEducas

Ivana Sánchez Pérez

Indice

1. Introducción	2
2. Ejercicio 1	3
2.1 ¿Qué es KVM?	3
2.2 Ventajas e inconvenientes de KVM	4
2.3 Definir qemu, libvirt, libvirtd, virsh y virt-manager	5
3. Ejercicio 2	7
3.1 Instalación básica de KVM en Linux (Debian)	7
3.2 Comprobar el estado del servicio libvirtd	10
3.3 Editar el fichero /etc/libvirt/qemu.conf	11
3.4 Cómo modificar la red que trae por defecto	14
4. Ejercicio 3	16
4.1 Leer información virt-manager y realizar los cambios solicitados	17
4.2 Arrancar el virt-manager y verificar el entorno KVM	17
4.2.1 Iniciar virt-manager	17
4.2.2 Verificar el entorno KVM	18
5. Ejercicio 4	19
5.1 Crear una máquina virtual Alpine 3.1	19
5.2 Mostrar los detalles del hardware virtual	25

1. Introducción

En esta tarea se llevará a cabo la instalación y configuración del servidor KVM (Kernel-based Virtual Machine) en un sistema operativo Debian anfitrión. El propósito principal es aprender a desplegar y gestionar un entorno de virtualización basado en KVM, una de las soluciones de virtualización más eficientes y utilizadas en entornos Linux. KVM permite que un sistema operativo Linux actúe como un hipervisor, lo que facilita la creación y administración de máquinas virtuales.

Durante el desarrollo de la actividad, se realizarán diversas configuraciones necesarias para garantizar el correcto funcionamiento del servidor KVM. Además, se abordarán aspectos relacionados con la configuración de la red, almacenamiento y administración de las máquinas virtuales. Se documentará todo el proceso con capturas de pantalla, respondiendo a preguntas específicas para asegurar la comprensión de cada paso.

2. Ejercicio 1

2.1 ¿Qué es KVM?

KVM es una tecnología de virtualización de código abierto integrada a Linux, que permite que éste actúe como hipervisor (software que permite realizar una virtualización del hardware), es decir, que permite que una máquina física ejecute múltiples sistemas operativos virtualizados (máquinas virtuales) de forma aislada sobre un mismo hardware físico de forma simultánea.

Aunque se instala como un módulo del núcleo (kernel), sigue siendo un hipervisor de tipo 1.

Sus características principales son:

- Integración en el kernel de Linux: convierte el kernel de Linux en un hipervisor, lo que significa que las máquinas virtuales se ejecutan directamente sobre el hardware físico, lo que garantiza alto rendimiento.
- Soporte para varios sistemas operativos: KVM puede ejecutar diversos sistemas operativos invitados, como diferentes distribuciones de Linux, Windows, y otros, en máquinas virtuales.

- Uso de hardware de virtualización: Para un mejor rendimiento, KVM aprovecha las extensiones de virtualización presentes en los procesadores modernos, como Intel VT-x y AMD-V.
- Gestión de máquinas virtuales: KVM trabaja en conjunto con herramientas como QEMU (un emulador y virtualización), libvirt (una interfaz para gestionar máquinas virtuales) y otras para crear, gestionar y configurar las máquinas virtuales.

KVM es ampliamente utilizado en entornos empresariales y servidores de nube (Amazon, Oracle) debido a su eficiencia, estabilidad y capacidad de gestionar entornos virtualizados a gran escala.

2.2 Ventajas e inconvenientes de KVM

❖ Ventajas:

- Al ser un proyecto de código abierto es gratuito y altamente personalizable.
- Previene los problemas que pudieran ocasionarse por los drivers, ya que al tener todos los sistemas operativos emulados una base de hardware común, no surgirían conflictos entre drivers.
- Flexibilidad: Los sistemas operativos se pueden ejecutar de manera independiente a los demás, según el tipo de tarea que queramos realizar y sin necesidad de hardware adicional.
- Escalabilidad: Permite crear un gran número de máquinas virtuales en un solo servidor físico.
- Los sistemas operativos que usemos pueden ser pausados, lo cual permite guardar lo que tenemos frente a futuros cambios provisional, ya que en ese caso bastaría con restaurarlo.
- El traslado a nuevos equipos de hardware es rápido.
- Ofrece un rendimiento excepcional al ser parte del núcleo de Linux.

❖ Desventajas:

- Al ser esta herramienta propiedad de Linux, sólo está disponible para los sistemas Linux.
- Hacer uso de una KVM requiere un hardware muy solvente y de alto rendimiento para el ordenador host, que sea capaz de soportar la carga de los sistemas operativos ejecutados. En nuestros servidores cloud, esto no representa ningún problema, ya que tienen capacidad suficiente para gestionar estas instalaciones.
- Resulta poco intuitivo hacer uso de una KVM, por lo que debemos dedicarle tiempo a familiarizarnos para aprender a usarla. La configuración inicial y la gestión avanzada de KVM pueden requerir un mayor conocimiento técnico en comparación con otras soluciones de virtualización.
- Dependencia del hardware: El rendimiento de KVM depende en gran medida del hardware subyacente. Un hardware antiguo o de baja calidad puede limitar el número y el rendimiento de las máquinas virtuales.
- La gestión de almacenamiento de KVM puede ser más compleja que en otras soluciones, especialmente cuando se trata de configuraciones de almacenamiento distribuido (sistema de almacenamiento definido por software que permite el acceso a los datos en cualquier momento, desde cualquier lugar y por los usuarios que tengan permiso).
- El uso de un hardware común para la KVM aumenta el riesgo de pérdidas en caso de accidente o avería, ya que todos los sistemas operativos funcionan con el mismo hardware. No obstante, esto puede evitarse gracias a las copias de seguridad.

2.3 Definir qemu, libvirt, libvirtd, virsh y virt-manager

- **Qemu:** Es un emulador y virtualizador de máquinas de código abierto. Permite emular una arquitectura distinta en el sistema anfitrión, lo que hace posible ejecutar aplicaciones y sistemas operativos diseñados para otra arquitectura completamente distinta. Está basado en la traducción dinámica de binarios, es decir, en la conversión del código binario de la arquitectura fuente en código entendible por la arquitectura huésped.
- **Libvirt:** Es un paquete diseñado para la administración y gestión de entornos de virtualización, que proporciona una API (interfaz de programación de aplicaciones) de virtualización independiente del hipervisor, que puede interactuar con las capacidades de virtualización de una variedad de sistemas operativos. Dicho de otra manera, es una interfaz gráfica para crear, gestionar y configurar máquinas virtuales KVM.
- **Libvirt:** Es el componente demonio (proceso que se ejecuta continuamente en segundo plano y realiza funciones necesarias para otros procesos) del lado del servidor del sistema de administración de virtualización Libvirt. Este demonio se ejecuta en servidores host y realiza las tareas de administración necesarias para los huéspedes virtualizados. Esto incluye actividades como iniciar, detener y migrar invitados entre servidores host, configurar y manipular redes y administrar el almacenamiento para que lo utilicen los invitados.
- **Virsh (Virtual Shell):** Es una interfaz CLI para administrar máquinas virtuales invitadas. Permite crear, enumerar, editar, iniciar, reiniciar, detener, suspender, reanudar, apagar y eliminar máquinas virtuales.
- **Virt-manager:** Es una interfaz de usuario de escritorio para administrar máquinas virtuales a través de Libvirt o lo que es lo mismo, es una herramienta gráfica para crear y administrar máquinas virtuales invitadas.

2.4 Requisitos de hardware y software

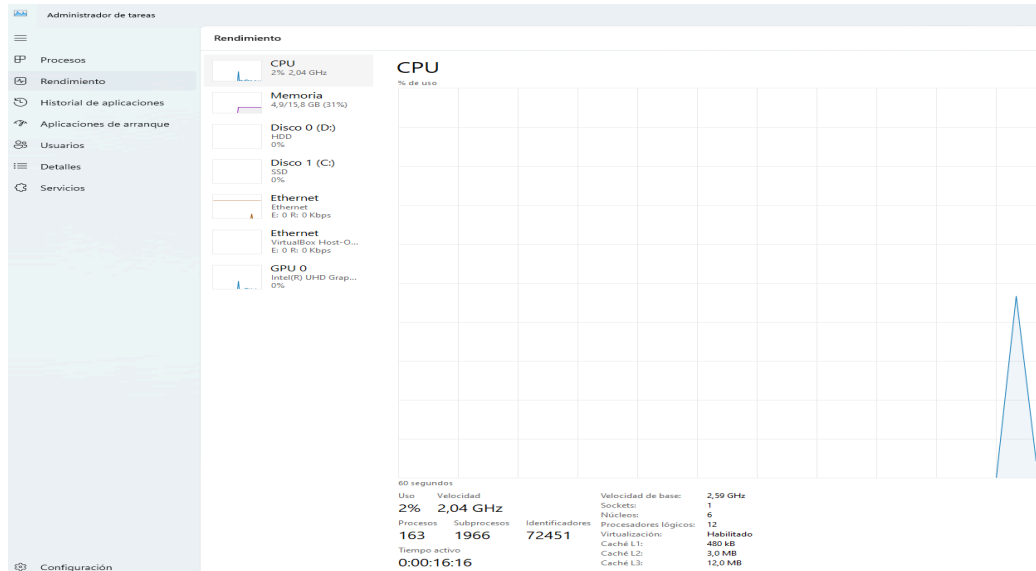
- **Requisitos de hardware:**
 - Velocidad de CPU: 3154MHz.

- **Procesador:** Es esencial que soporte las extensiones de virtualización de hardware y que tenga varios núcleos, pues a mayor número más máquinas virtuales ejecutará de forma simultánea.
 - Intel: Debe contar con la tecnología Intel VT-x.
 - AMD: Debe contar con la tecnología AMD-V.
 - **Espacio disco duro:** Se requiere suficiente espacio para almacenar los sistemas operativos invitados, aplicaciones y datos, por lo que se recomienda al menos 20 GB de espacio libre, pero puede variar según las necesidades de almacenamiento de cada máquina virtual.
 - **Memoria RAM:** La cantidad necesaria dependerá del número y tipo de máquinas virtuales que se ejecuten. Como regla general, se recomienda asignar al menos 2 GB de RAM por máquina virtual, más la RAM necesaria para el sistema operativo host. Ante esto, la memoria del sistema deberá tener como mínimo 64 GB.
 - **Tarjeta de red:** Tiene que tener una capacidad con suficiente ancho de banda, pues de esta forma garantizará una buena conectividad de red para las máquinas virtuales.
- **Requisitos software**
 - Sistema operativo RHEL 7.0 de 64 bits con soporte habilitado para virtualización.
 - Es necesario un puente de red para configurar una interfaz de red par KVM.
 - Herramientas de gestión: Virt-manager, Libvirt.

3. Ejercicio 2

3.1 Instalación básica de KVM en Linux (Debian)

Antes de instalar KVM nos aseguraremos que la virtualización está habilitada en la BIOS/UEFI y que el procesador soporta la citada virtualización de hardware (Intel VT-x o amd-v). Para ello, en nuestra máquina física, entraremos en “Administración de sistemas” (Ctrl+shift+esc) y dentro pincharemos en el desplegable de la izquierda en “Rendimiento”.



Para utilizar KVM es necesario instalar los siguientes paquetes de software:

- KVM: el propio módulo del kernel que habilita la virtualización.
- QEMU: un emulador y virtualizador que trabaja junto a KVM para gestionar las máquinas virtuales.
- LIBVIRT: Un conjunto de herramientas para gestionar hipervisores.
- VIRT-MANAGER: interfaz gráfica para gestionar las máquinas virtuales.

Instalaremos los paquetes necesarios para la virtualización y realizar bridges con el comando “sudo apt install qemu-kvm libvirt-daemon-system libvirt-clients bridge-utils libguestfs-tools genisoimage virtinst libosinfo-bin virt-manager”. Así que, primero actualizaremos el sistema (sudo apt update) y posteriormente pondremos nuestro comando.


```
ivana@debian: ~  
ivana@debian:~$ su  
Contraseña:  
root@debian:/home/ivana# apt install -y qemu-kvm libvirt-clients libvirt-daemon-system \br/>idge-utils libguestfs-tools genisoimage virtinst libosinfo-bin virt-ma  
nager  
_leyendo lista de paquetes... Hecho  
_creando árbol de dependencias... Hecho  
_leyendo la información de estado... Hecho  
Nota, seleccionando «qemu-system-x86» en lugar de «qemu-kvm»  
qemu-system-x86 ya está en su versión más reciente (1:7.2+dfsg-7+deb12u7).  
libvirt-clients ya está en su versión más reciente (9.0.0-4+deb12u1).  
libvirt-daemon-system ya está en su versión más reciente (9.0.0-4+deb12u1).  
bridge-utils ya está en su versión más reciente (1.7.1-1).
```

Una vez acabe la instalación, procederemos a verificar que el módulo kvm esté cargado con el comando “sudo lsmod | grep kvm”.

```
ivana@debian: ~  
ivana@debian:~$ su  
Contraseña:  
root@debian:/home/ivana# lsmod | grep kvm  
kvm_intel          380928  0  
kvm                1146880  1 kvm_intel  
irqbypass         16384   1 kvm  
root@debian:/home/ivana#
```

Pero utilizando el comando “sudo systemctl status libvirtd”, observamos que nos da un fallo. Esto se debe a que VirtualBox ya está utilizando la virtualización de hardware de la CPU, lo que significa que no se puede utilizar KVM directamente.

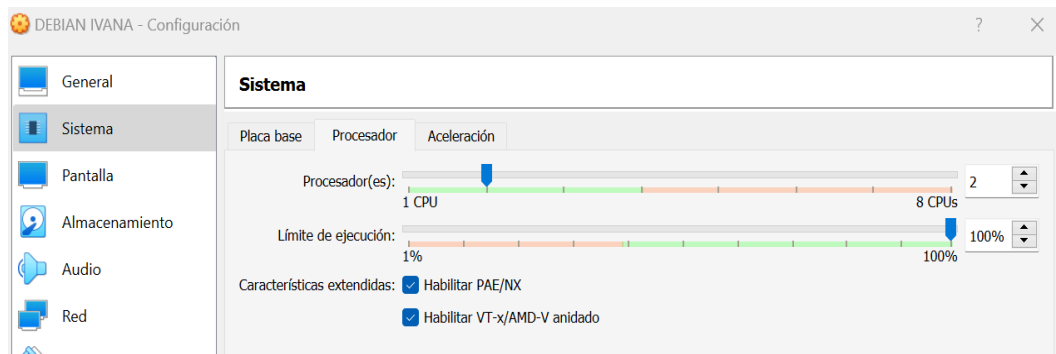
```
ivana@debian: ~  
root@debian:/home/ivana# systemctl status libvirtd  
● libvirtd.service - Virtualization daemon  
   Loaded: loaded (/lib/systemd/system/libvirtd.service; enabled; preset: ena  
   Active: active (running) since Mon 2024-09-30 23:47:11 CEST; 1min 29s ago  
TriggeredBy: ● libvirtd-admin.socket  
              ● libvirtd.socket  
              ● libvirtd-ro.socket  
   Docs: man:libvirtd(8)  
          https://libvirt.org  
  Main PID: 4692 (libvirtd)  
    Tasks: 19 (limit: 32768)  
  Memory: 12.1M  
    CPU: 115ms  
   CGroup: /system.slice/libvirtd.service  
           └─4692 /usr/sbin/libvirtd --timeout 120  
  
sep 30 23:47:11 debian systemd[1]: Starting libvirtd.service - Virtualization d  
sep 30 23:47:11 debian systemd[1]: Started libvirtd.service - Virtualization da  
sep 30 23:47:11 debian libvirtd[4692]: libvirt version: 9.0.0, package: 9.0.0-4  
sep 30 23:47:11 debian libvirtd[4692]: hostname: debian  
sep 30 23:47:11 debian libvirtd[4692]: No es posible abrir /dev/kvm: No existe  
lines 1-20/20 (END)
```

Para solucionar el problema, recurriremos a la virtualización anidada, la cual tendremos que habilitar en nuestro terminal de Debian. Es como dar un permiso que nos permita crear una máquina virtual dentro de otra. En este caso crear KVM dentro de VirtualBox.

Con la máquina parada y desde una línea de comando de nuestra máquina física, nos desplazamos hasta la carpeta de VirtualBox con el comando “cd” y seguidamente pondremos el comando “VBoxManage modifyvm “Nombre_de_tu_VM” --nested-hw-virt on”.

```
C:\Users\IVANA>cd "C:\Program Files\Oracle\VirtualBox\  
C:\Program Files\Oracle\VirtualBox>VBoxManage modifyvm "DEBIAN IVANA" --nested-hw-virt on  
C:\Program Files\Oracle\VirtualBox>
```

Posteriormente nos ponemos encima de nuestra máquina virtual Debian y hacemos clic en “Configuración”, y dentro de Sistema, pulsamos la pestaña de “Procesador” Y habilitamos el VT-x/AMD-v y el PAE/NX.



El siguiente paso es agregar nuestro usuario al grupo de administradores de KVM. Para ello lo primero que hay que hacer es verificar que el grupo esté ya creado.

```
ivana@debian: ~  
ivana@debian:~$ su  
Contraseña:  
root@debian:/home/ivana# grep kvm /etc/group  
kvm:x:104:  
root@debian:/home/ivana#
```

Seguidamente agregaremos el usuario al grupo de administradores de KVM con el comando “gpasswd”, pues tanto “usermod” como “adduser” me han dado problemas. y con el comando “newgrp” aplicaremos los cambios de forma inmediatamente sin necesidad de reiniciar.

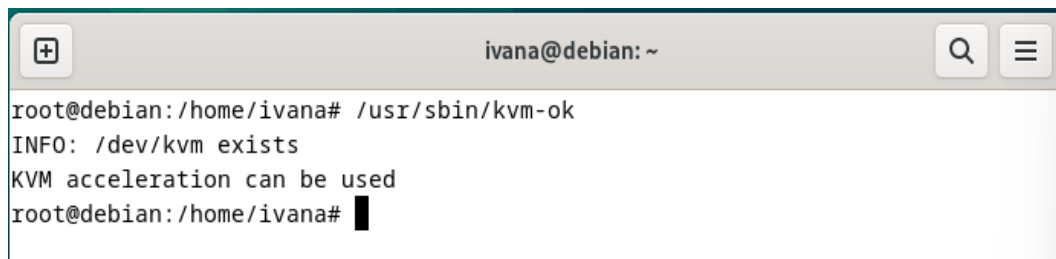
```
ivana@debian: ~  
root@debian:/home/ivana# gpasswd -a ivana libvirt  
Añadiendo al usuario ivana al grupo libvirt  
root@debian:/home/ivana# gpasswd -a ivana libvirt-qemu  
Añadiendo al usuario ivana al grupo libvirt-qemu  
root@debian:/home/ivana# newgrp libvirt  
root@debian:/home/ivana# newgrp libvirt-qemu  
root@debian:/home/ivana#
```

```
ivana@debian: ~  
vana@debian:~$ groups  
vana cdrom floppy audio dip video plugdev users kvm netdev bluetooth lpadmin scanner libvirt libvirt-qemu  
vana@debian:~$
```

3.2 Comprobar el estado del servicio libvirtd

Comprobamos de nuevo si KVM está activo instalando el paquete “cpu-checker” y poder utilizar el comando “kvm-ok”.

```
root@debian:/home/ivana# apt install cpu-checker
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
```

A terminal window titled 'ivana@debian: ~' with search and menu icons in the top right. The terminal shows the command '/usr/sbin/kvm-ok' being executed. The output indicates that '/dev/kvm' exists and KVM acceleration can be used. The prompt returns to the root user at the ivana home directory.

```
root@debian:/home/ivana# /usr/sbin/kvm-ok
INFO: /dev/kvm exists
KVM acceleration can be used
root@debian:/home/ivana#
```

Reiniciamos el servicio libvirt con “systemctl restart libvirtd” y volvemos a verificamos su estado con “systemctl status libvirtd”.

```
root@debian:/home/ivana# systemctl restart libvirtd
root@debian:/home/ivana# systemctl status libvirtd
● libvirtd.service - Virtualization daemon
   Loaded: loaded (/lib/systemd/system/libvirtd.service; enabled; preset: ena>
   Active: active (running) since Tue 2024-10-01 01:19:42 CEST; 14s ago
 TriggeredBy: ● libvirtd.socket
               ● libvirtd-ro.socket
               ● libvirtd-admin.socket
   Docs: man:libvirtd(8)
         https://libvirt.org
 Main PID: 2762 (libvirtd)
    Tasks: 19 (limit: 32768)
  Memory: 9.5M
     CPU: 97ms
   CGroup: /system.slice/libvirtd.service
           └─2762 /usr/sbin/libvirtd --timeout 120

oct 01 01:19:42 debian systemd[1]: Starting libvirtd.service - Virtualization d>
oct 01 01:19:42 debian systemd[1]: Started libvirtd.service - Virtualization da>
lines 1-17/17 (END)
```

3.3 Editar el fichero /etc/libvirt/qemu.conf

Para editar el fichero /etc/libvirt/qemu.conf en Debian podemos utilizar un editor de texto como nano o vim con privilegios de superusuario.

Primero nos aseguramos de que libvirt y qemu estén correctamente instalados con: `sudo apt list --installed | grep libvirt` y `sudo apt list --installed | grep qemu`.

```
root@debian:/home/ivana# sudo apt list --installed | grep libvirt

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

qir1.2-libvirt-glib-1.0/stable,now 4.0.0-2 amd64 [instalado, automático]
libvirt-clients/stable,now 9.0.0-4+deb12u1 amd64 [instalado]
libvirt-daemon-config-network/stable,now 9.0.0-4+deb12u1 all [instalado, automático]
```

```
root@debian:/home/ivana# sudo apt list --installed | grep qemu

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

ipxe-qemu/stable,now 1.0.0+git-20190125.36a4c85-5.1 all [instalado, automático]
libvirt-daemon-driver-qemu/stable,now 9.0.0-4+deb12u1 amd64 [instalado, automático]
qemu-block-extra/stable,now 1:7.2+dfsg-7+deb12u7 amd64 [instalado, automático]
qemu-system-common/stable,now 1:7.2+dfsg-7+deb12u7 amd64 [instalado, automático]
qemu-system-data/stable,now 1:7.2+dfsg-7+deb12u7 all [instalado, automático]
qemu-system-gui/stable,now 1:7.2+dfsg-7+deb12u7 amd64 [instalado, automático]
qemu-system-x86/stable,now 1:7.2+dfsg-7+deb12u7 amd64 [instalado]
qemu-utils/stable,now 1:7.2+dfsg-7+deb12u7 amd64 [instalado, automático]
```

Y además que ambos están funcionando (“virsh capabilities”).

```
ivana@debian: ~  
root@debian:/home/ivana# virsh capabilities  
<capabilities>  
  
<host>  
  <uuid>2fe3f24e-c460-9d44-be29-430336b2d028</uuid>  
  <cpu>  
    <arch>x86_64</arch>  
    <model>Westmere</model>  
    <vendor>Intel</vendor>  
    <signature family='6' model='142' stepping='12' />  
    <counter name='tsc' frequency='2111998000' scaling='no' />  
    <topology sockets='1' dies='1' cores='1' threads='1' />  
    <maxphysaddr mode='emulate' bits='39' />  
  </cpu>  
</host>
```

Posteriormente abrimos el terminal en Debian y ejecutamos el siguiente comando: “sudo nano /etc/libvirt/qemu.conf” para que se nos abra el archivo y asignarle qemu al usuario root. Reiniciamos el servicio y volvemos a comprobar systemctl restart libvirt.

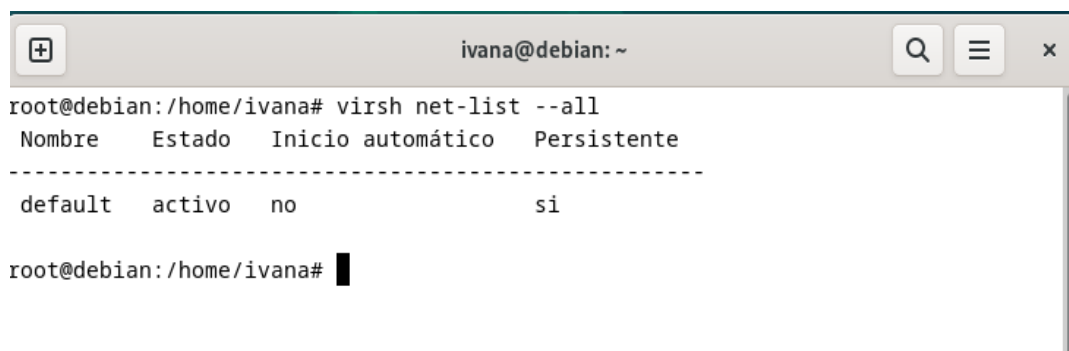
```
ivana@debian: ~  
GNU nano 7.2 /etc/libvirt/qemu.conf *  
# name.  
#  
# Some examples of valid values are:  
#  
#     user = "qemu"    # A user named "qemu"  
#     user = "+0"      # Super user (uid=0)  
#     user = "100"     # A user named "100" or a user with uid=100  
#  
#user = "root"  
  
# The group for QEMU processes run by the system instance. It can be  
# specified in a similar way to user.  
#group = "root"  
  
# Whether libvirt should dynamically change file ownership  
# to match the configured user/group above. Defaults to 1.  
# Set to 0 to disable file ownership changes.  
#dynamic_ownership = 1  
  
# Whether libvirt should remember and restore the original  
  
^G Ayuda      ^O Guardar    ^W Buscar     ^K Cortar     ^T Ejecutar   ^C Ubicación  
^X Salir      ^R Leer fich. ^_ Reemplazar  ^U Pegar      ^J Justificar ^/ Ir a línea
```

Reiniciamos el servicio y volvemos a comprobar con “systemctl status libvirtd”.

3.4 Cómo modificar la red que trae por defecto

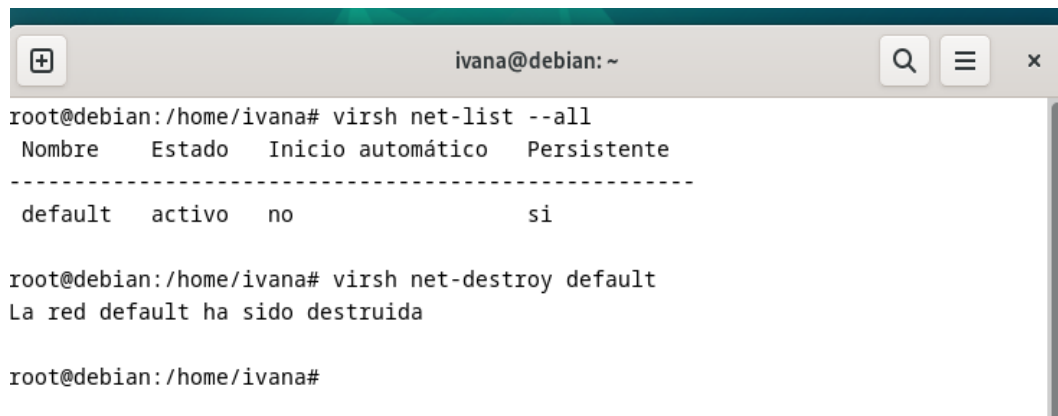
Para modificar la configuración de la red que trae por defecto libvirt en Debian (NAT), seguiremos los siguientes pasos:

- Abrir el terminal
- Listar las redes configuradas (incluyendo la red default), con el comando: “virsh net-list –all”.



```
ivana@debian: ~  
root@debian:/home/ivana# virsh net-list --all  
Nombre      Estado  Inicio automático  Persistente  
-----  
default     activo  no                  si  
  
root@debian:/home/ivana#
```

- Detener la red por defecto antes de hacer ningún cambio con “virsh net-destroy default”.

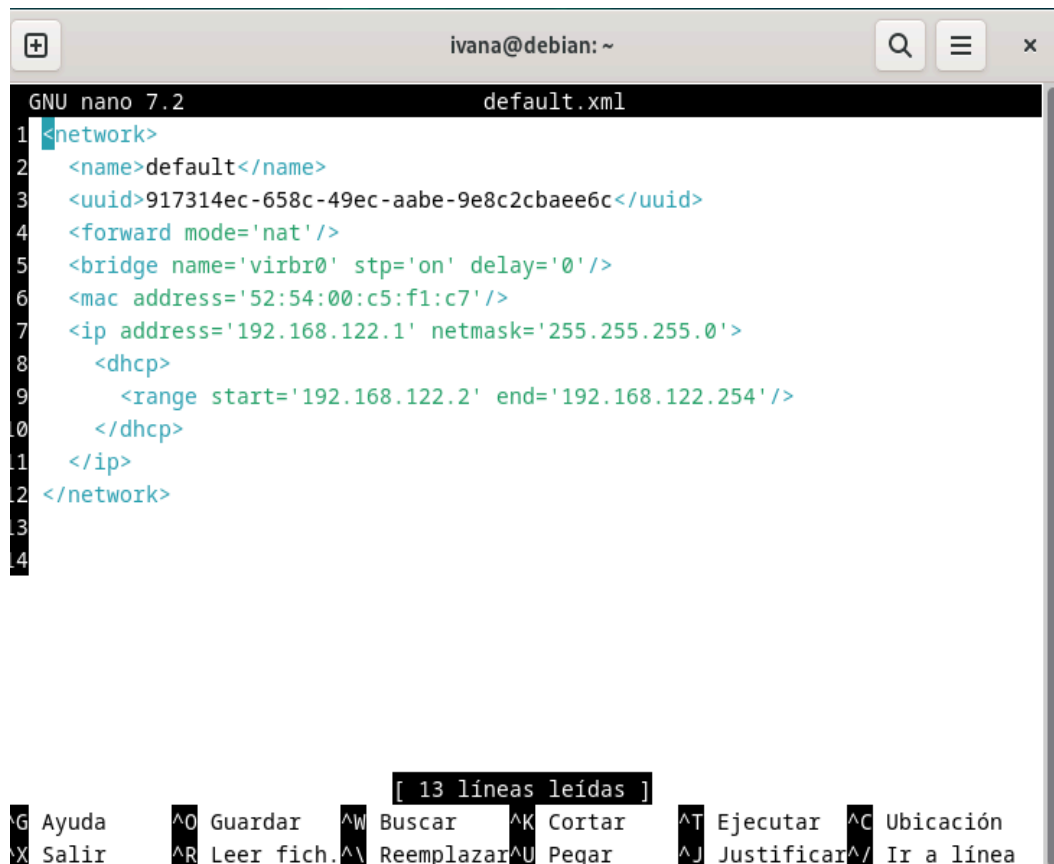


```
ivana@debian: ~  
root@debian:/home/ivana# virsh net-list --all  
Nombre      Estado  Inicio automático  Persistente  
-----  
default     activo  no                  si  
  
root@debian:/home/ivana# virsh net-destroy default  
La red default ha sido destruida  
  
root@debian:/home/ivana#
```

- Modificar la configuración de red. Para ello, lo primero que necesitamos es obtener la configuración actual en formato XML (virsh net-dumpxml default > default.xml. Esto creará un archivo llamado default.xml con la configuración actual de la red.

```
root@debian:/home/ivana# virsh net-dumpxml default > default.xml
root@debian:/home/ivana#
```

- Editar el archivo XML. Abrimos el archivo default.xml en un editor de texto.



```
GNU nano 7.2 default.xml
1 <network>
2   <name>default</name>
3   <uuid>917314ec-658c-49ec-aabe-9e8c2cbaee6c</uuid>
4   <forward mode='nat' />
5   <bridge name='virbr0' stp='on' delay='0' />
6   <mac address='52:54:00:c5:f1:c7' />
7   <ip address='192.168.122.1' netmask='255.255.255.0'>
8     <dhcp>
9       <range start='192.168.122.2' end='192.168.122.254' />
10    </dhcp>
11  </ip>
12 </network>
13
14
```

[13 líneas leídas]

^G Ayuda	^O Guardar	^W Buscar	^K Cortar	^T Ejecutar	^C Ubicación
^X Salir	^R Leer fich.	^E Reemplazar	^U Pegar	^J Justificar	^_ Ir a línea

- Rango de direcciones IP. En el archivo abierto, modificamos el rango de direcciones IP en la etiqueta <ip> en la línea correspondiente. Por ejemplo: <ip address='192.168.133.2' netmask='255.255.255.0'>.
- También se puede modificar la configuración DHCP cambiando las direcciones de inicio y fin en las etiquetas <range>.<dhcp> <range start='192.168.100.2' end='192.168.100.254' /> </dhcp>
- Guardar cambios. Guardamos el archivo (Ctrl+o Enter), y luego salimos (Ctrl+x).
- Definir la red con la nueva configuración con “virsh net-define default.xml”
- Iniciar la red nuevamente con “virsh net-start default”

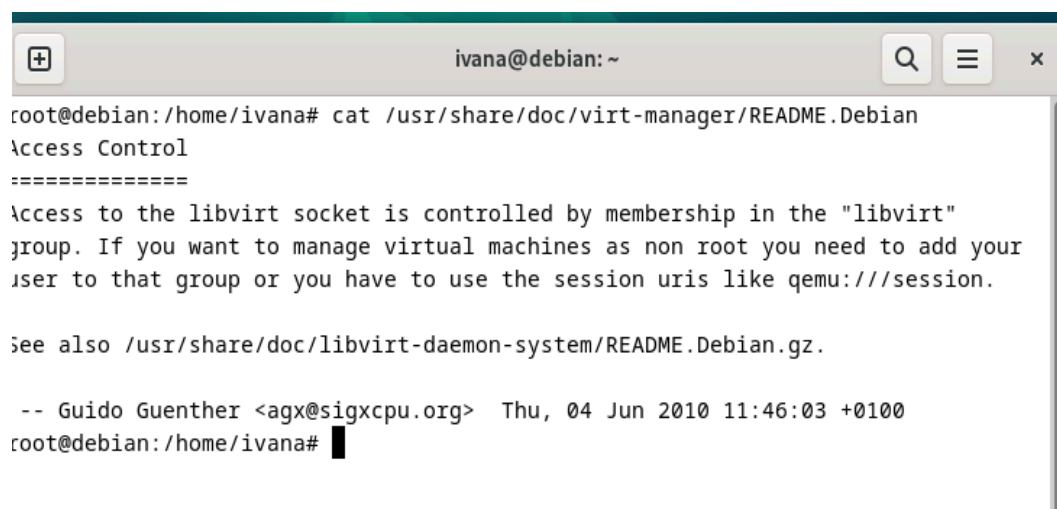
Si queremos que la red se inicie automáticamente cada vez que se inicie libvirt, ejecutamos el comando “virsh net-autostart default”.

Y para comprobar que los cambios se aplicaron correctamente: “virsh net-list –all”.

4. Ejercicio 3

4.1 Leer información virt-manager y realizar los cambios solicitados

El archivo `/usr/share/doc/virt-manager/README.Debian`, es el que nos proporciona información específica para usuarios de Debian sobre cómo se ha empaquetado el programa `virt.manager` en el sistema, así como instrucciones adicionales que pueden no estar cubiertas en la documentación estándar del proyecto. Nos da información de permisos, requisitos previos y dependencias, información sobre problemas conocidos, información de soporte o contacto, etc.



```
ivana@debian: ~  
root@debian:/home/ivana# cat /usr/share/doc/virt-manager/README.Debian  
Access Control  
=====
```

Access to the libvirt socket is controlled by membership in the "libvirt" group. If you want to manage virtual machines as non root you need to add your user to that group or you have to use the session uris like `qemu:///session`.

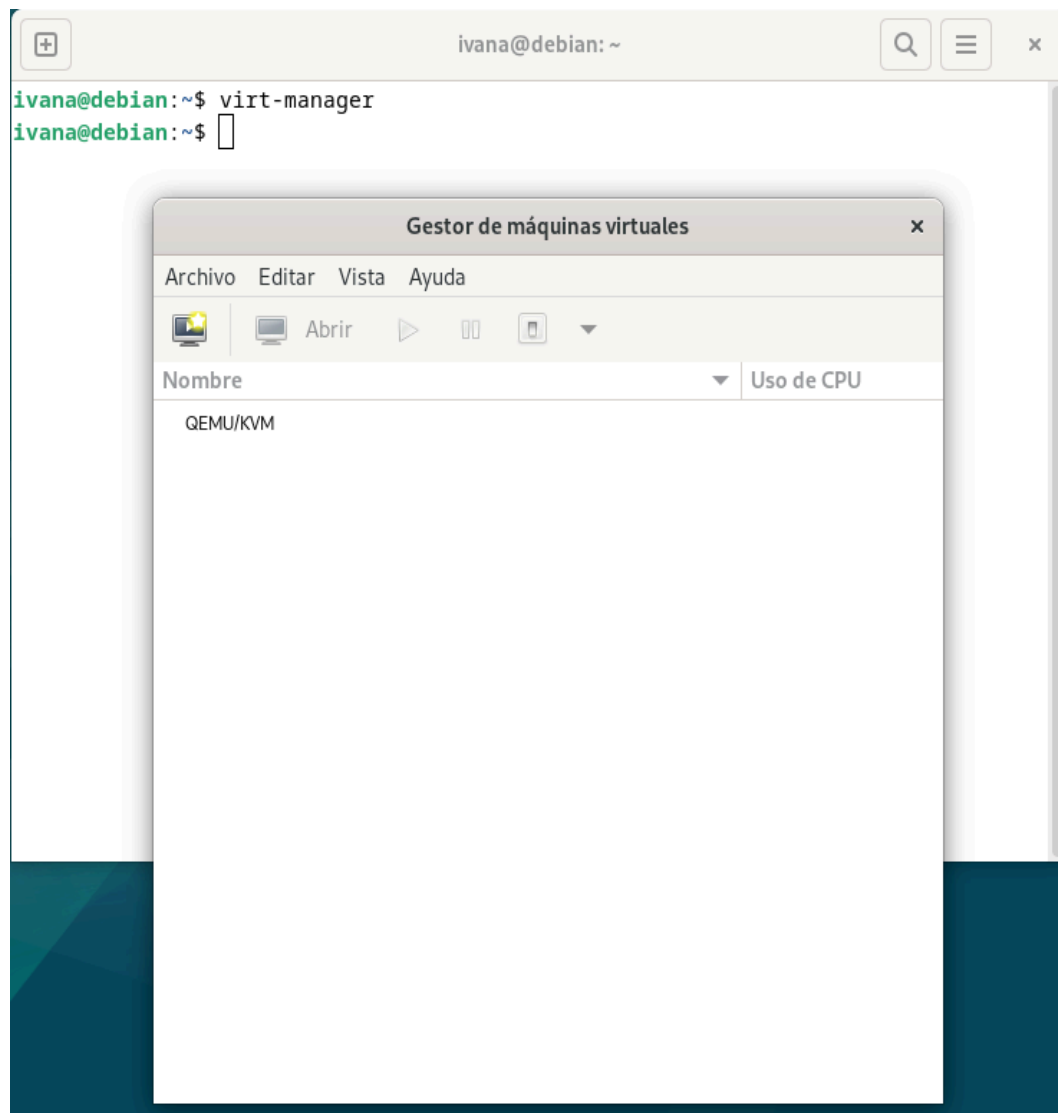
See also `/usr/share/doc/libvirt-daemon-system/README.Debian.gz`.

-- Guido Guenther <agx@sigxcpu.org> Thu, 04 Jun 2010 11:46:03 +0100
root@debian:/home/ivana#

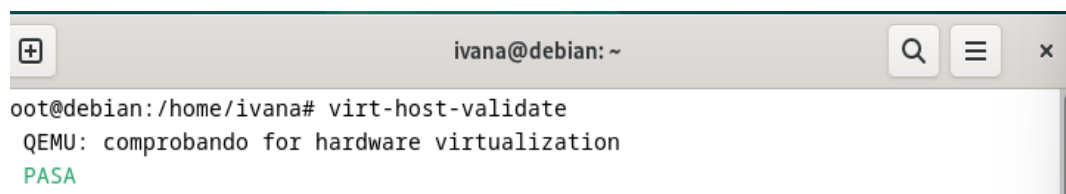
4.2 Arrancar el virt-manager y verificar el entorno KVM

4.2.1 Iniciar virt-manager

Para iniciar `virt-manager`, lo haremos desde el terminal utilizamos el comando “`virt-manager`” sin ser superusuario.



Una vez instalado KVM, validamos ejecutando los siguientes comandos: `sudo virt-host-validate` y `sudo virt-host-validate | grep -i 'FALLA\ADVERTENCIA'`.



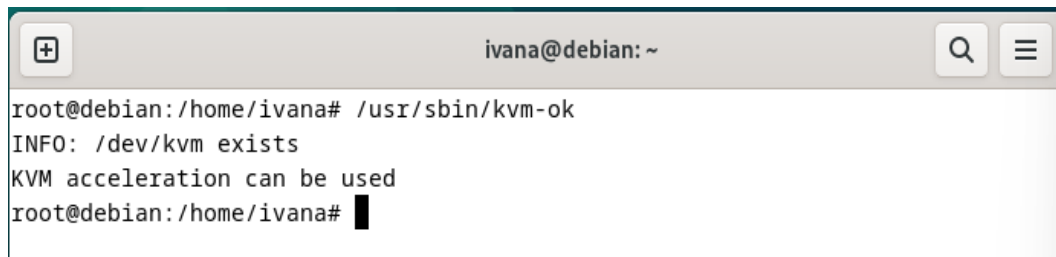
4.2.2 Verificar el entorno KVM

Libvirt proporciona varios mecanismos para conectarse a un hipervisor Qemu/KVM.

```
root@debian:/home/ivana# virsh --connect qemu:///session
Bienvenido a virsh, la terminal de virtualización interactiva.

Ingrese: 'help' para obtener ayuda con los comandos
        'quit' para salir
```

- Verificamos que KVM esté funcionando. En el terminal ejecutando el comando “sudo /usr/sbin/kvm-ok”. (ponemos la ruta).

A screenshot of a terminal window titled 'ivana@debian: ~'. The terminal shows the command 'root@debian:/home/ivana# /usr/sbin/kvm-ok' being executed. The output is 'INFO: /dev/kvm exists' followed by 'KVM acceleration can be used'. The prompt returns to 'root@debian:/home/ivana#'.

```
ivana@debian: ~
root@debian:/home/ivana# /usr/sbin/kvm-ok
INFO: /dev/kvm exists
KVM acceleration can be used
root@debian:/home/ivana#
```

- Verificamos la configuración de redes. En virt-manager, ir a la sección de redes para asegurarnos que la red por defecto esté activa y funcionando.

5. Ejercicio 4

5.1 Crear una máquina virtual Alpine 3.1

- Primero descargaremos en nuestra máquina virtual Debian, la imagen ISO de Alpine-virt-3.17.8-x86_64.iso, para ello entramos en https://uk.alpinelinux.org/alpine/v3.17/releases/x86_64/ y navegamos hacia abajo hasta la versión que buscamos y hacer clic sobre ella.

alpine-virt-3.17.5-x86_64.iso	07-Aug-2023 13:18	49M
alpine-virt-3.17.5-x86_64.iso.asc	07-Aug-2023 15:18	833
alpine-virt-3.17.5-x86_64.iso.sha256	07-Aug-2023 13:18	96
alpine-virt-3.17.5-x86_64.iso.sha512	07-Aug-2023 13:18	160
alpine-virt-3.17.6-x86_64.iso	30-Nov-2023 11:50	49M
alpine-virt-3.17.6-x86_64.iso.asc	30-Nov-2023 12:24	833
alpine-virt-3.17.6-x86_64.iso.sha256	30-Nov-2023 11:50	96
alpine-virt-3.17.6-x86_64.iso.sha512	30-Nov-2023 11:50	160
alpine-virt-3.17.7-x86_64.iso	26-Jan-2024 20:49	49M
alpine-virt-3.17.7-x86_64.iso.asc	26-Jan-2024 22:01	833
alpine-virt-3.17.7-x86_64.iso.sha256	26-Jan-2024 20:49	96
alpine-virt-3.17.7-x86_64.iso.sha512	26-Jan-2024 20:49	160
alpine-virt-3.17.8-x86_64.iso	18-Jun-2024 15:22	49M
alpine-virt-3.17.8-x86_64.iso.sha256	18-Jun-2024 15:22	96
alpine-virt-3.17.8-x86_64.iso.sha512	18-Jun-2024 15:22	160
alpine-virt-3.17.9-x86_64.iso	22-Jul-2024 18:23	49M
alpine-virt-3.17.9-x86_64.iso.asc	22-Jul-2024 18:23	833

Antes de continuar la instalación, y para no tener problemas más adelante, activaremos la red virtual default.

```

ivana@debian: ~
ivana@debian:~$ su
Contraseña:
root@debian:/home/ivana# virt-manager
root@debian:/home/ivana# virsh net-list --all
Nombre      Estado      Inicio automático  Persistente
-----
default     inactivo    no                  si

root@debian:/home/ivana# virsh net-start default
.a red default se ha iniciado

root@debian:/home/ivana# virsh net-autostart default
.a red default ha sido marcada para iniciarse automáticamente

root@debian:/home/ivana# virsh net-list --all
Nombre      Estado      Inicio automático  Persistente
-----
default     activo      si                  si

root@debian:/home/ivana#

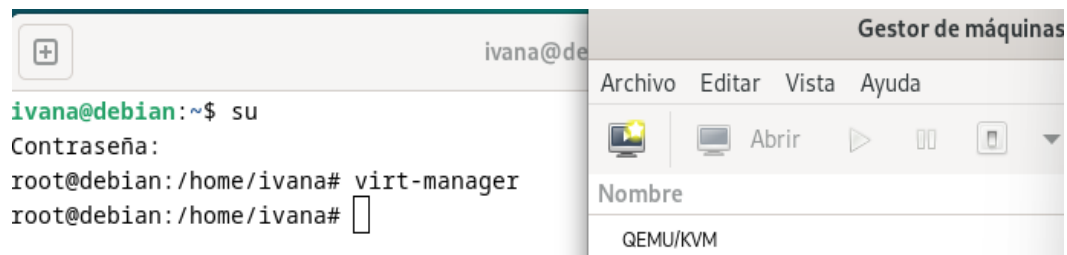
```

Nos aseguramos que KVM está configurado correctamente y que los módulos del kernel para KVM estén cargados. ("lsmod | grep kvm").

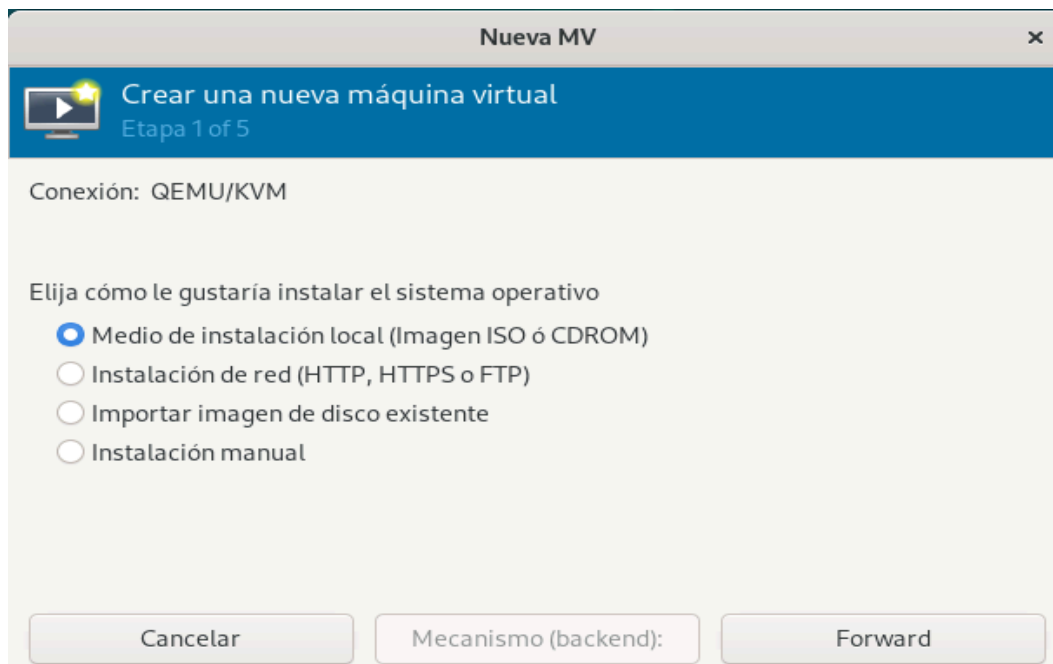
```
ivana@debian: ~  
ivana@debian:~$ su  
Contraseña:  
root@debian:/home/ivana# lsmod | grep kvm  
kvm_intel          380928  0  
kvm                1146880  1 kvm_intel  
irqbypass         16384   1 kvm  
root@debian:/home/ivana#
```

Una vez la ISO descargada, procederemos a abrir el virt-manager para crear la máquina virtual en el terminal.

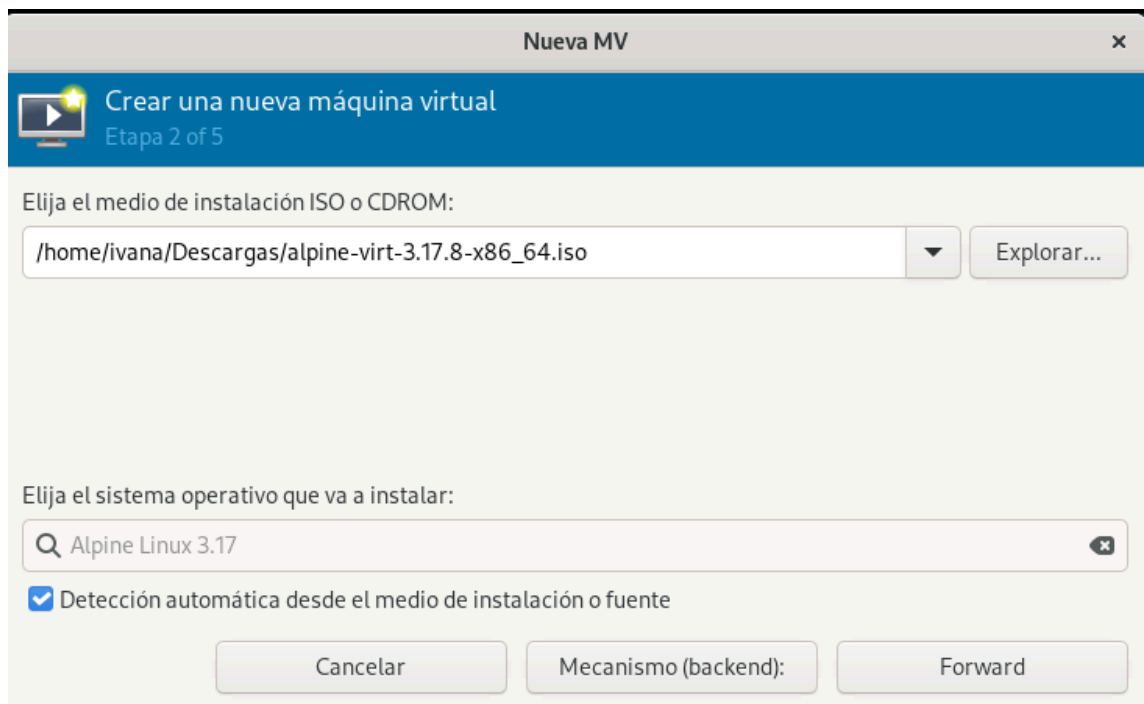
1- Iniciamos virt-manager



2- Creamos la máquina virtual. En la ventana principal de virt-manager, hacemos click en el botón “Crear una nueva máquina virtual” y seleccionamos “Medio de instalación local (imagen ISO ó CDROM)” y hacemos click en “Forward” (siguiente).



3- Pinchamos “Explorar” - “Explore localmente” - “Descargas”(Abrir) para, por último, seleccionar la ISO de Alpine Linux 3.20-3-x86_64.iso (Abrir) que nos hemos descargado anteriormente. Especificamos el tipo de sistema operativo. Para ello desactivamos la detección automática y escribimos linux en el buscador. Saldrá un desplegable y elegiremos “Generic Linux 2022 (linux2022)”. Forward.



4- Completamos los pasos asignando memoria RAM y un tamaño adecuado al disco duro. Finalizamos la configuración y comenzará la instalación de Alpine desde la ISO.

Nueva MV

Crear una nueva máquina virtual
Etapa 3 of 5

Elija la configuración de la memoria y la CPU:

Memoria: 1967 — +
Hasta 1967 MiB disponible en el equipo

CPU: 2 — +
Hasta 2 disponibles

Cancelar Mecanismo (backend): Forward

Nueva MV

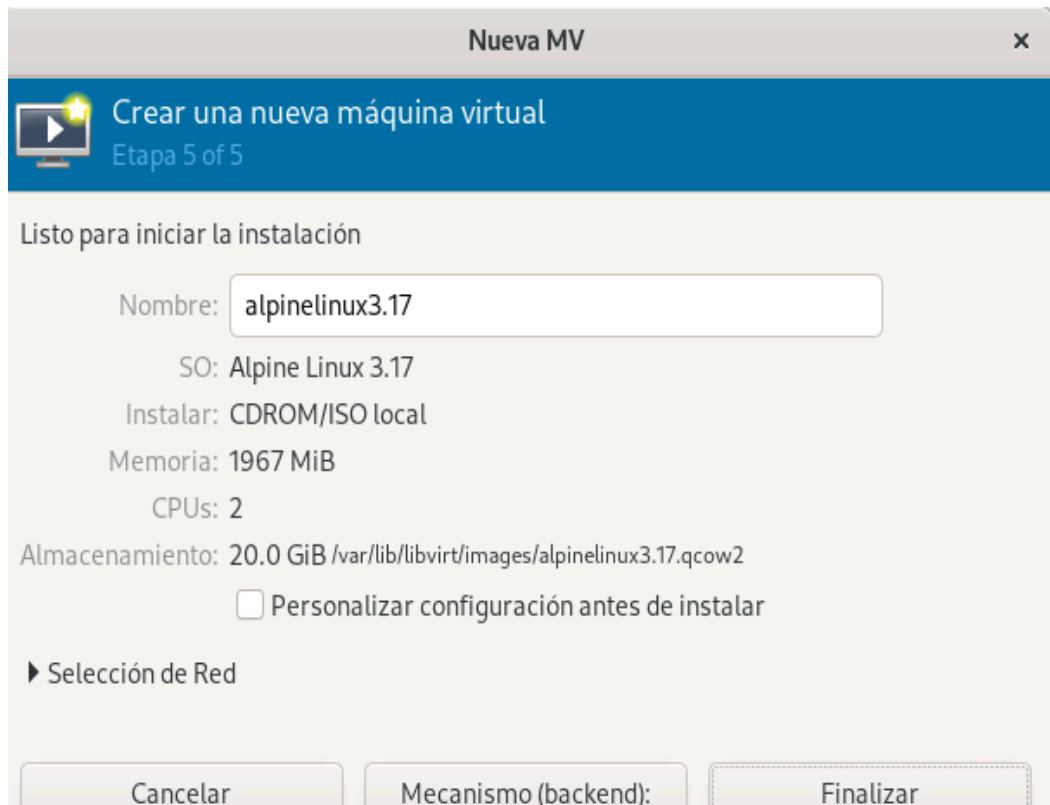
Crear una nueva máquina virtual
Etapa 4 of 5

☒ Habilitar almacenamiento para esta máquina virtual

☒ Crear una imagen de disco para la máquina virtual
20,0 — + GiB
11.4 GiB disponible en la ubicación por defecto

☐ Seleccionar o crear almacenaje personalizado
Administrar...

Cancelar Mecanismo (backend): Forward



Terminado el asistente, nos encontraremos con el mensaje "Welcome to Alpine Linux 3.17", ya estamos dentro del terminal del sistema.

Lo primero que nos pedirá será el localhost, y como aún no se ha configurado un usuario personalizado, usaremos como Usuario "root" y la contraseña la dejamos vacía.

Nos volverá a dar la bienvenida y lanzaremos el comando "setup-alpine" para comenzar su configuración.


```
alpinelinux3.17 en QEMU/KVM
Archivo  Máquina Virtual  Vista  Enviar Tecla
[Icons]
SeaBIOS (version 1.16.2-debian-1.16.2-1)
machine UUID d4b2778b-f71e-4233-be32-ceb99aafb953

IPXE (http://ipxe.org) 01:00:0 CA00 PCI2.10 PnP PMM+2EFCE8E0+2EF0E8E0 CA00

Booting from DVD/CD...

SYSLINUX 6.04 6.04-pre1 ETCD Copyright (C) 1994-2015 H. Peter Anvin et al
boot:

Welcome to Alpine Linux 3.17
Kernel 5.15.161-0-virt on an x86_64 (/dev/tty1)

localhost login: root
Welcome to Alpine!

The Alpine Wiki contains a large amount of how-to guides and general
information about administrating Alpine systems.
See <https://wiki.alpinelinux.org/>.

You can setup the system with the command: setup-alpine

You may change this message by editing /etc/motd.

localhost:~#
```

El teclado, por defecto, está en inglés, por lo que pondremos “es” y continuaremos.

```
You can setup the system with the command: setup-alpine
You may change this message by editing /etc/motd.

localhost:~# setup-alpine

Available keyboard layouts:
af  az  br  cm  dz  fo  gr  il  it  kz  lv  nl  nl  pt  si  tn  uz
al  ba  brai  cn  ee  fr  hr  in  jp  la  na  nm  no  ro  sk  tr  vn
am  bd  by  cz  epo  gb  hu  iq  ke  latam  nd  nt  ph  rs  sy  tw
ara  be  ca  de  es  ge  id  ir  kg  lk  me  my  pk  ru  th  ua
at  bg  ch  dk  fi  gh  ie  is  kr  lt  mk  ng  pl  se  tj  us
Select keyboard layout: [none] Enter system hostname (fully qualified form, e.g. 'foo.example.org') [localhost]
```

El script de instalación pasará por diferentes pasos de configuración:

- Iniciar interfaz: eth0
- Configuración IP: dhcp
- ¿Realizar alguna configuración de red manual?: no.
- Establecer el password del root:
- Zona horaria: Europa y luego Madrid.
- Proxy: none
- Elección del mirror: f (el más rápido).

- Instalación del servidor SSH: openssh (opción por defecto)
- Cliente NTP: chrony (opción por defecto)

```
Available interfaces are: eth0.
Enter '?' for help on bridges, bonding and vlans.
Which one do you want to initialize? (or '?' or 'done') [eth0]
Ip address for eth0? (or 'dhcp', 'none', '?') [dhcp]
Do you want to do any manual network configuration? (y/n) [n] n
udhcpd: started, v1.35.0
udhcpd: broadcasting discover
udhcpd: broadcasting discover
udhcpd: broadcasting select for 192.168.122.198, server 192.168.122.1
udhcpd: lease of 192.168.122.198 obtained from 192.168.122.1, lease time 3600
Changing password for root.
```

```
Changing password for root
New password:
Bad password: too weak
Retype password:
passwd: password for root changed by root
Which timezone are you in? ('?' for list) [UTC] ?
Africa/      Chile/      GB-Eire      Israel       Nava jo      US/
America/     Cuba        GMT           Jamaica      PRC          UTC
Antarctica/  EET         GMT+0        Japan        PST8PDT      Universal
Arctic/      EST         GMT-0        Kwa jalein  Pacific/     W-SU
Asia/        EST5EDT     GMT0         Libya        Poland       WET
Atlantic/    Egypt      Greenwich    MET          Portugal     Zulu
Australia/   Eire        HST          MST          ROC          leap-seconds.list
Brazil/      Etc/        Hongkong     MST7MDT      ROK          posixrules
CET          Europe/     Iceland      Mexico/      Singapore    right/
CST6CDT      Factory     Indian/      NZ           Turkey
Canada/      GB          Iran          NZ-CHAT      UCT
Which timezone are you in? ('?' for list) [UTC] Europe
What sub-timezone of 'Europe' are you in? ('?' for list) ?
Amsterdam    Brussels    Guernsey     Kyiv         Minsk        Riga         Sofia         Vatican
Andorra      Bucharest   Helsinki     Lisbon       Monaco       Rome         Stockholm    Vienna
Astrakhan    Budapest    Isle_of_Man  Ljubljana    Moscow       Samara       Tallinn      Vilnius
Athens       Busingen    Istanbul     London       Nicosia      San_Marino   Tirane       Volgograd
Belfast      Chisinau    Jersey       Luxembourg    Oslo         Sara jevo    Tiraspol     Warsaw
Belgrade     Copenhagen  Kaliningrad  Madrid       Paris        Saratov      Ulyanovsk    Zagreb
Berlin       Dublin      Kiev         Malta         Podgorica    Simferopol   Uzhgorod     Zaporozhye
Bratislava   Gibraltar   Kirov        Mariehamn    Prague       Skopje       Vaduz        Zurich
What sub-timezone of 'Europe' are you in? ('?' for list) Madrid
* Seeding random number generator ... * Saving 256 bits of creditable seed for next boot
[ ok ]
* Starting busybox acpid ... [ ok ]
* Starting busybox cron ... [ ok ]
```

```

76) alpine.koyanet.lv
77) mirrors.hosterion.ro
78) mirror.tux.si
79) mirrors.lax.silicloud.com
80) mirrors.tyo.silicloud.com
81) mirrors.neterra.net
82) mirror.twds.com.tw
83) mirrors.jcut.edu.cn
84) mirror.raiolanetworks.com
85) elmirror.cl
86) alpinelinux.mirrors.ovh.net
87) debian.ethz.ch
88) mirror.leitecastro.com
89) pkg.adfinis-on-exoscale.ch
90) mirrors.hust.edu.cn

r) Add random from the above list
f) Detect and add fastest mirror from above list
e) Edit /etc/apk/repositories with text editor

```

```

Enter mirror number (1-90) or URL to add (or r/f/e/done) [11 f]
Finding fastest mirror...
0.11 http://dl-cdn.alpinelinux.org/alpine/
0.79 http://uk.alpinelinux.org/alpine/
0.21 http://mirror.yandex.ru/mirrors/alpine/
0.31 http://mirrors.gigenet.com/alpinelinux/
0.13 http://mirror1.hs-esslingen.de/pub/Mirrors/alpine/
0.15 http://mirror.leaseweb.com/alpine/
0.22 http://mirror.fel.cvut.cz/alpine/
0.19 http://alpine.mirror.far.fi/
0.14 http://alpine.mirror.wearetriple.com/
0.33 http://mirror.clarkson.edu/alpine/
0.7 http://mirror.aarnet.edu.au/pub/alpine
0.17 http://mirrors.dotsrc.org/alpine/
0.15 http://ftp.halifax.rwth-aachen.de/alpine/
wget: server returned error: HTTP/1.1 403 Forbidden
wget: server returned error: HTTP/1.1 403 Forbidden
1.61 http://mirrors.nju.edu.cn/alpine/
2.12 http://mirror.lzu.edu.cn/alpine/
0.22 http://mirror.accum.se/mirror/alpinelinux.org/
0.58 http://mirror.xtom.com.hk/alpine/
0.38 http://mirror.csclub.uwaterloo.ca/alpine/
0.15 http://pkg.adfinis.com/alpine/
0.3 http://mirror.ps.kz/alpine/
0.11 http://mirrors.ircam.fr/pub/alpine/
1.29 http://mirror.math.princeton.edu/pub/alpinelinux/
wget: download timed out

```

```

Setup a user? (enter a lower-case loginname, or 'no') [no]
Which ssh server? ('openssh', 'dropbear' or 'none') [openssh]
Allow root ssh login? ('?' for help) [prohibit-password]
Enter ssh key or URL for root (or 'none') [none]
* service sshd added to runlevel default
* Caching service dependencies ... [ ok ]

ssh-keygen: generating new host keys: RSA ECDSA ED25519
* Starting sshd ... [ ok ]
Available disks are:
uda (4.3 GB 0x1af4)
Which disk(s) would you like to use? (or '?' for help or 'none') [none] Enter where to store configs ('floppy', 'usb' or 'none')
[none]
Enter apk cache directory (or '?' or 'none') [/var/cache/apk]
fetch http://mirror.raiolanetworks.com/alpine/v3.17/main/x86_64/openssh-keygen-9.1_p1-r6.apk
fetch http://mirror.raiolanetworks.com/alpine/v3.17/main/x86_64/openssh-client-common-9.1_p1-r6.apk
fetch http://mirror.raiolanetworks.com/alpine/v3.17/main/x86_64/openssh-client-default-9.1_p1-r6.apk
fetch http://mirror.raiolanetworks.com/alpine/v3.17/main/x86_64/openssh-sftp-server-9.1_p1-r6.apk
fetch http://mirror.raiolanetworks.com/alpine/v3.17/main/x86_64/openssh-server-common-9.1_p1-r6.apk
fetch http://mirror.raiolanetworks.com/alpine/v3.17/main/x86_64/openssh-server-9.1_p1-r6.apk
es: #

```

- ```

Available disks are:
 vda (4.3 GB 0x1af4)
Which disk(s) would you like to use? (or '?' for help or 'none') [none] ?

The disk you select can be used for a traditional disk install or for a
data-only install.

The disk will be erased.

Enter 'none' if you want to run diskless.

Available disks are:
 vda (4.3 GB 0x1af4)
Which disk(s) would you like to use? (or '?' for help or 'none') [none] vda
The following disk is selected:
 vda (4.3 GB 0x1af4)
How would you like to use it? ('sys', 'data', 'crypt', 'lvm' or '?' for help) [?] sys
WARNING: The following disk(s) will be erased:
 vda (4.3 GB 0x1af4)
WARNING: Erase the above disk(s) and continue? (y/n) [n] y
Creating file systems...
Installing system on /dev/vda3:
/mnt/boot is device /dev/uda1
100%
=> initramfs: creating /boot/initramfs-virt for 6.6.53-0-virt
/boot is device /dev/uda1

Installation is complete. Please reboot.
root:~#

```

Para "lsblk" → apk add util-linux → información de los discos

Para "lspci" → apk add pciutils → dispositivos PCI

Para "lscpu" → apk add coreutils → detalles del CPU

Una vez instalados, procederemos a ejecutarlos para encontrar los datos que necesitamos.

## Información de la CPU

```
root:~# lscpu
Architecture: x86_64
CPU op-mode(s): 32-bit, 64-bit
Address sizes: 40 bits physical, 48 bits virtual
Byte Order: Little Endian
CPU(s): 2
On-line CPU(s) list: 0,1
Vendor ID: AuthenticAMD
BIOS Vendor ID: QEMU
Model name: QEMU Virtual CPU version 2.5+
 BIOS Model name: pc-q35-7.2 CPU @ 2.0GHz
 BIOS CPU family: 1
 CPU family: 15
 Model: 107
 Thread(s) per core: 1
 Core(s) per socket: 1
 Socket(s): 2
 Stepping: 1
 BogomIPS: 4223.68
 Flags: fpu de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2
 nx lm rep_good nopl cpuid extd_apicid pni cx16 hypervisor lahf_lm sm 3dnowprefetch ummcall
Virtualization features:
 Virtualization: AMD-V
Caches (sum of all):
 L1d: 128 KiB (2 instances)
 L1i: 128 KiB (2 instances)
 L2: 1 MiB (2 instances)
 L3: 32 MiB (2 instances)
Vulnerabilities:
 Gather data sampling: Not affected
 Itlb multihit: Not affected
 L1tf: Not affected
 Mds: Not affected
 Meltdown: Not affected
 Mmio stale data: Not affected
 Reg file data sampling: Not affected
 Retbleed: Not affected
 Spec rstack overflow: Not affected
 Spec store bypass: Not affected
 Spectre v1: Mitigation: usercopy/swapgs barriers and __user pointer sanitization
 Spectre v2: Mitigation: Retpolines; STIBP disabled; RSB filling; PBRSB-eIBRS Not affected; BHI Not affected
 Srbds: Not affected
 Tsx async abort: Not affected
```

## Información de los discos

```
root:~# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
sr0 11:0 1 1024M 0 rom
vda 253:0 0 4G 0 disk
├─vda1 253:1 0 300M 0 part /boot
├─vda2 253:2 0 1G 0 part [SWAP]
└─vda3 253:3 0 2.7G 0 part /
```

## Información dispositivos PCI

```
root:~# lspci
00:00.0 Host bridge: Intel Corporation 82G33/G31/P35/P31 Express DRAM Controller
00:01.0 VGA compatible controller: Red Hat, Inc. QXL paravirtual graphic card (rev 05)
00:02.0 PCI bridge: Red Hat, Inc. QEMU PCIe Root port
00:02.1 PCI bridge: Red Hat, Inc. QEMU PCIe Root port
00:02.2 PCI bridge: Red Hat, Inc. QEMU PCIe Root port
00:02.3 PCI bridge: Red Hat, Inc. QEMU PCIe Root port
00:02.4 PCI bridge: Red Hat, Inc. QEMU PCIe Root port
00:02.5 PCI bridge: Red Hat, Inc. QEMU PCIe Root port
00:02.6 PCI bridge: Red Hat, Inc. QEMU PCIe Root port
00:02.7 PCI bridge: Red Hat, Inc. QEMU PCIe Root port
00:03.0 PCI bridge: Red Hat, Inc. QEMU PCIe Root port
00:03.1 PCI bridge: Red Hat, Inc. QEMU PCIe Root port
00:03.2 PCI bridge: Red Hat, Inc. QEMU PCIe Root port
00:03.3 PCI bridge: Red Hat, Inc. QEMU PCIe Root port
00:03.4 PCI bridge: Red Hat, Inc. QEMU PCIe Root port
00:03.5 PCI bridge: Red Hat, Inc. QEMU PCIe Root port
00:1b.0 Audio device: Intel Corporation 828011 (ICH9 Family) HD Audio Controller (rev 03)
00:1f.0 ISA bridge: Intel Corporation 828011B (ICH9) LPC Interface Controller (rev 02)
00:1f.2 SATA controller: Intel Corporation 828011R/10/1H (ICH9R/DO/DH) 6 port SATA Controller [AHCI mode] (rev 02)
00:1f.3 SMBus: Intel Corporation 828011 (ICH9 Family) SMBus Controller (rev 02)
01:00.0 Ethernet controller: Red Hat, Inc. Virtio 1.0 network device (rev 01)
02:00.0 USB controller: Red Hat, Inc. QEMU XHCI Host Controller (rev 01)
03:00.0 Communication controller: Red Hat, Inc. Virtio 1.0 console (rev 01)
04:00.0 SCSI storage controller: Red Hat, Inc. Virtio 1.0 block device (rev 01)
05:00.0 Unclassified device [00ff]: Red Hat, Inc. Virtio 1.0 memory balloon (rev 01)
06:00.0 Unclassified device [00ff]: Red Hat, Inc. Virtio 1.0 RNG (rev 01)
root:~#
```

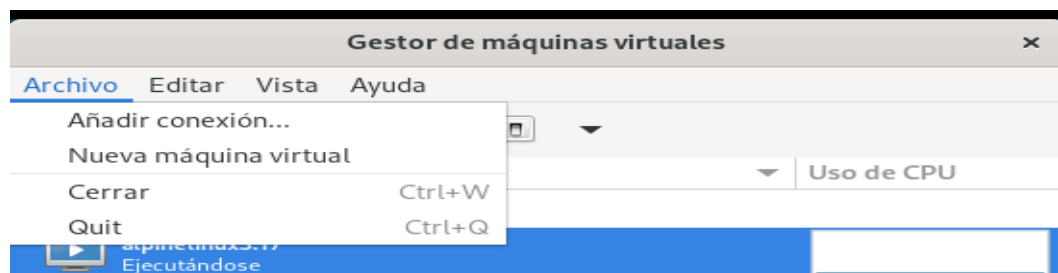
Información de la memoria RAM, con el comando “free”

```
root:~# free
 total used free shared buff/cache available
Mem: 738456 66248 608484 120 63724 583496
Swap: 1048572 0 1048572
root:~#
root:~#
```

Dirección IP con “ip a”

```
root:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1000
 link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
 inet 127.0.0.1/8 scope host lo
 valid_lft forever preferred_lft forever
 inet6 ::1/128 scope host
 valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
 link/ether 52:54:00:c4:42:94 brd ff:ff:ff:ff:ff:ff
 inet 192.168.122.198/24 scope global eth0
 valid_lft forever preferred_lft forever
 inet6 fe80::5054:ff:fec4:4294/64 scope link
 valid_lft forever preferred_lft forever
root:~#
```

Otra forma, es desde la interfaz de Alpine. Ponemos en el terminal el comando “virt-manager” y nos ponemos encima de nuestra máquina virtual, para luego pinchar en la pestaña “Editar” y dentro en “Detalles de la máquina virtual”.



Esto nos dará todos los datos solicitados

