

Práctica 1: Instalación de la plataforma de virtualización KVM. Creación y configuración de una máquina virtual

Óscar Alexander Martín Tacoronte

Grupo 01.04 (Cod. 44)

7 de febrero de 2024

1. Índice

1. Índice.....	2
2. Introducción.....	3
3. Objetivos.....	4
3.1 Instalación de la plataforma KVM	4
3.2 Creación y configuración de una máquina virtual.....	4
4. Desarrollo	5
4.1 Instalación de la plataforma KVM	5
4.2 Creación y configuración de una máquina virtual.....	7
5. Pruebas / Validación	8
5.1 Verificación de la habilitación del servicio libvirtd y la carga de los módulos del núcleo kvm y kvm_intel	8
5.2 Verificación de las características básicas de configuración de la máquina virtual	9
5.3 Verificación de la instalación del SO en la MV con nombre MinimalFedoraServer39: método de instalación elegido y funcionamiento	11
6. Fuentes de información.....	13
Referencias	13

2. Introducción

En esta práctica tenemos como objetivo principal hacer la preparación de la plataforma de virtualización KVM. Esta herramienta será la que se empleará en el resto de las prácticas para la creación y gestión de una infraestructura virtual. Además, también crearemos una máquina virtual con una configuración predeterminada y validaremos que dicha máquina virtual tenga conexión al exterior

3. Objetivos

3.1 Instalación de la plataforma KVM

- Primer contacto y configuración del puesto de trabajo
- Completar la encuesta de “Datos PC”
- Instalación de los paquetes de virtualización

3.2 Creación y configuración de una máquina virtual

- Montar la imagen ISO de una máquina virtual
- Comprobar que el entorno de virtualización funcione correctamente

4. Desarrollo

4.1 Instalación de la plataforma KVM

En el puesto de trabajo, se presenta varias particiones. Según las indicaciones [1], se selecciona la partición ANFITRIÓN1 en la que se presenta la instalación de una distribución de Linux: Fedora Server. Se debe iniciar sesión como usuario root, con la contraseña proporcionada, y posteriormente se debe personalizar la contraseña del usuario root, para ello se hará uso de la orden para cambiar la clave:

```
# passwd
```

Una vez configurada las credenciales del usuario root, se completa la encuesta de “Datos del PC” indicada en [1] proporcionando los campos requeridos (Puesto de trabajo, dirección IP y clave del usuario root). Para obtener la dirección IP, se ejecuta la siguiente orden con la que se consulta los parámetros de red y se completa el campo con la dirección IP de la interfaz enp2s0

```
# ifconfig
```

```
[root@pc-1230 ~]# ifconfig
enp2s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.22.147.18 netmask 255.255.255.0 broadcast 10.22.147.255
        inet6 fe80::bb48:b53:29c7:f29a prefixlen 64 scopeid 0x20<link>
        ether e0:d5:5e:0e:a0:1d txqueuelen 1000 (Ethernet)
        RX packets 180039 bytes 206202812 (196.6 MiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 76022 bytes 6273783 (5.9 MiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Figura 1. Verificación de la IP del puesto de trabajo. Nota: esta captura de pantalla se realizó al final de la práctica, y no en el momento exacto de la realización de la encuesta de “Datos del PC”

Antes de instalar los paquetes de virtualización, se debe asegurar de habilitar el status de SELinux, y se podrá comprobar su modo de ejecución. Seguidamente, se muestra por tanto dicha orden:

```
# sestatus
```

Como su estado estaba deshabilitado, se accede a su fichero de configuración principal en la ruta `/etc/selinux/config`, se encontrará unos comentarios para poder asignar una configuración que necesitemos, que en este caso se forzará su ejecución, por tanto, ejecutando el siguiente comando, se cambiará la variable `SELINUX=enforcing`. Luego, se reinicia el sistema para que se efectúen los cambios:

```
# vi /etc/selinux/config
```

Con el objetivo de instalar la plataforma KVM, se procede en este punto a actualizar el gestor de paquetes de Fedora (DNF). Posteriormente, continuamos con la instalación un entorno grafico GNOME, y el navegador Firefox. También se configura en el sistema que se utilice el entorno gráfico en cada arranque. Todo este procedimiento se puede ejecutar de manera secuencial:

```
# dnf upgrade
# dnf group install gnome # No se recuerda la orden exacta
respecto a Gnome
# systemctl set-default graphical.target
# reboot
```

En este punto, el sistema anfitrión ya está configurado, y se continúa con la instalación de los paquetes de virtualización, donde simplemente instalaremos el entorno completo de virtualización

```
# dnf group install virtualization
```

4.2 Creación y configuración de una máquina virtual.

Con los objetivos expuestos en el punto 3.1, ya está el sistema preparado para crear una máquina virtual. Por tanto, montaremos una imagen ISO expuesta en el servidor con dirección 10.22.146.215 en el directorio /**imagenes/fedora/39/isos/x86_64** y lo haremos en un directorio que nombraremos “iso”, donde se irá colocando el resto de las imágenes que se pueda necesitar instalar en sucesivas prácticas. Se ejecuta, por tanto la siguiente orden:

```
# mount 10.22.146.215:/imagenes/fedora/39/isos/x86_64 iso
```

Con la imagen instalada, se hace uso de la herramienta gráfica para crear una máquina virtual siguiendo los pasos de la aplicación y proporcionando los parámetros vistos en la ficha de la práctica. En el punto 5 se podrá observar el cumplimiento de estos parámetros.

5. Pruebas / Validación

5.1 Verificación de la habilitación del servicio libvirtd y la carga de los módulos del núcleo kvm y kvm_intel

Primero verificamos la habilitación del servicio libvirtd, y al estar deshabilitada, posteriormente se habilita para que cada vez que se arranque el sistema se ejecute el servicio libvirtd. El procedimiento de verificación y habilitación del servicio se puede observar en la figura 2, por tanto, según [2] ejecutamos la siguiente orden para:

```
# systemctl status libvirtd
```

Y una vez verificado que no está en ejecución, se habilita para que cada vez que se arranque el sistema con la siguiente orden:

```
# systemctl enable --now libvirtd
```

```
[root@pc-1230 ~]# systemctl status libvirt
Unit libvirt.service could not be found.
[root@pc-1230 ~]# systemctl status libvirtd
● libvirtd.service - Virtualization daemon
  Loaded: loaded (/usr/lib/systemd/system/libvirtd.service; disabled; preset: disabled)
  Drop-In: /usr/lib/systemd/system/service.d
            └─10-timeout-abort.conf
    Active: inactive (dead)
  TriggeredBy: ● libvirtd-ro.socket
                ○ libvirtd-admin.socket
                ○ libvirtd.socket
                ○ libvirtd-tls.socket
                ○ libvirtd-tcp.socket
    Docs: man:libvirtd(8)
          https://libvirt.org
[root@pc-1230 ~]# systemctl enable --now libvirtd
Created symlink /etc/systemd/system/multi-user.target.wants/libvirtd.service → /usr/lib/systemd/system/libvirtd.service.
Created symlink /etc/systemd/system/sockets.target.wants/libvirtd.socket → /usr/lib/systemd/system/libvirtd.socket.
Created symlink /etc/systemd/system/sockets.target.wants/libvirtd-ro.socket → /usr/lib/systemd/system/libvirtd-ro.socket.
[root@pc-1230 ~]# systemctl status libvirtd
● libvirtd.service - Virtualization daemon
  Loaded: loaded (/usr/lib/systemd/system/libvirtd.service; enabled; preset: disabled)
```

Figura 2. Verificación del servicio libvirtd

Además, verificamos también que se cargan los módulos kvm y kvm_intel en el kernel, dicha comprobación es apreciable en la figura 3. Para ello, según algunas de las indicaciones en [3], ejecutamos el siguiente comando para listar los módulos cargados, y se redirige al comando que nos permite filtrar las líneas por los módulos KVM que es de interés:

```
# lsmod | grep kvm
```

```
[root@pc-1230 ~]# lsmod | grep kvm
kvm_intel           454656  2
kvm                 1318912  1 kvm_intel
irqbypass          12288   4 kvm
```

Figura 3. La orden lsmod nos permite consultar los módulos cargados en el kernel, y con la utilidad grep verificamos los módulos que son de interés en la práctica.

5.2 Verificación de las características básicas de configuración de la máquina virtual

Se configura los siguientes parámetros, y se muestra en cada figura dicha verificación:

- Nombre de la máquina virtual: en la figura 5 se puede verificar el nombre de la máquina virtual **MinimalFedoraServer39**, así como otros detalles del hipervisor

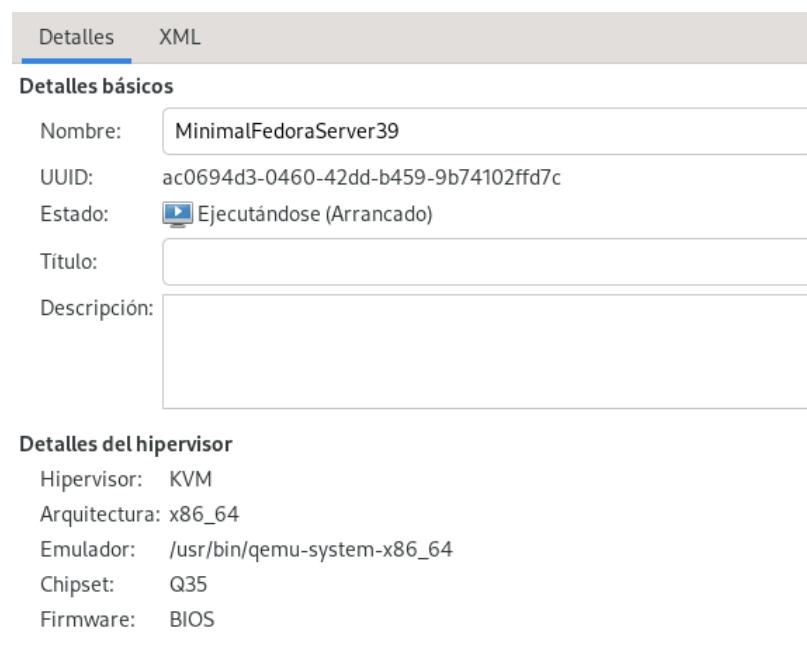


Figura 5. Muestra los detalles sobre la máquina virtual

- El tipo de red sea tipo NAT. En la figura 6 se puede observar los detalles de la interfaz de red, y su configuración.

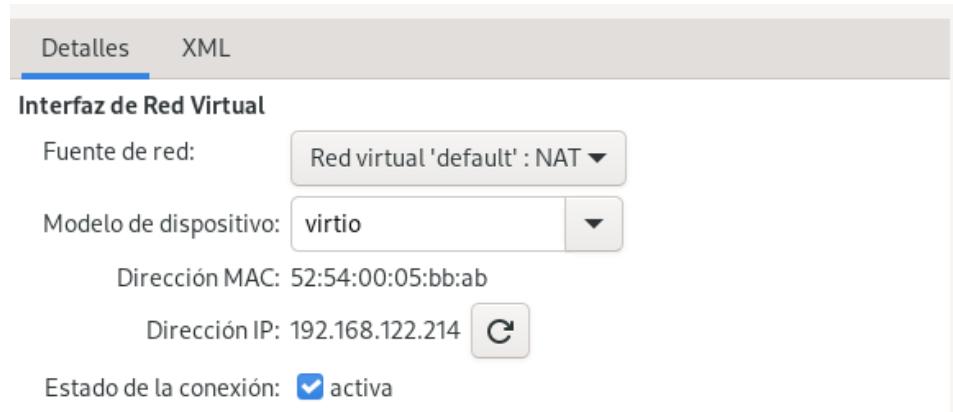


Figura 6. Muestra los detalles sobre la máquina virtual

- Memoria RAM de 2GB y el número de procesadores se selecciona 1. En la figura 7 se puede comprobar las características de memoria así como de la asignación de CPU a la máquina virtual.

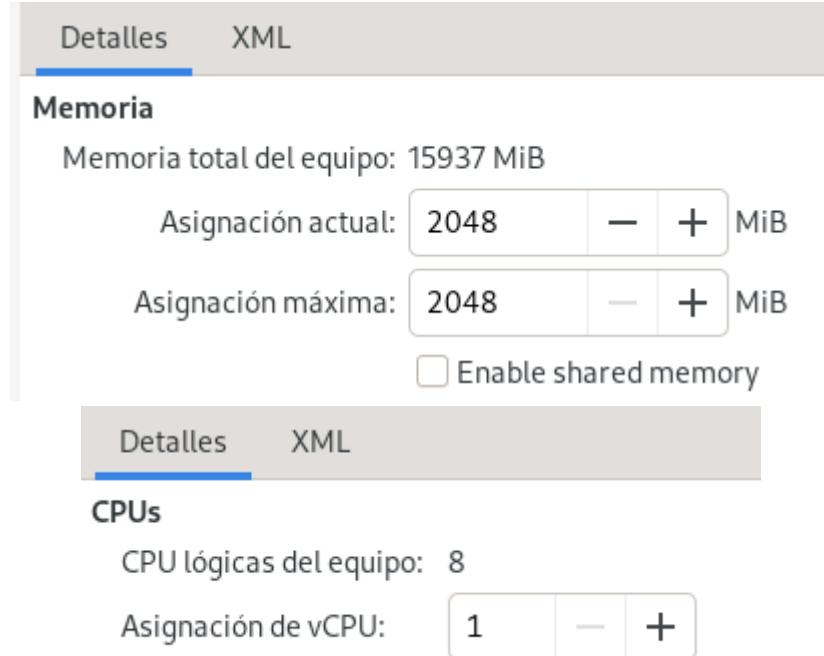


Figura 7. Detalles de la configuración de red de la máquina virtual y asignación de CPU.

- Almacenamiento de disco de la máquina virtual se introduce 10 GB. En la figura 8 se puede visualizar los detalles de disco.

Disco Virtual

Ruta de origen: ... r/lib/libvirt/images/MinimalFedoraServer39.qcow2

Tipo de dispositivo: VirtIO Disk 1

Bus de disco: VirtIO

Tamaño de almacenamiento: 10.00 GiB

▶ Opciones avanzadas

Figura 8. Muestra los detalles sobre la configuración del almacenamiento en disco de la máquina virtual.

5.3 Verificación de la instalación del SO en la MV con nombre MinimalFedoraServer39: método de instalación elegido y funcionamiento

Se selecciona la imagen Fedora-Server-netinst-x86_64-39-1.5.iso de las 3 variantes proporcionadas. También se comprueba el funcionamiento lanzando trazas de ICMP hacia el exterior para comprobar la conectividad de red, así como el funcionamiento de los DNS con el siguiente comando:

```
# ping $ip_address
```

Se puede verificar en la figura 4 la ejecución del comando ping señalado con la flecha en color rojo, la conectividad con la dirección ip 8.8.8.8 y el dominio www.google.com para comprobar los DNS, así como a la máquina anfitriona cuya dirección IP es visible en la figura 1: 10.22.147.18.

```
Last login: Tue Jan 30 14:14:06 on ttys1
[root@localhost ~]# ping 8.8.8.8 [REDACTED]
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=115 time=30.7 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=115 time=31.0 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=115 time=30.8 ms
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 30.662/30.812/31.000/0.140 ms
[root@localhost ~]# ping www.google.com [REDACTED]
PING www.google.com (142.250.200.100) 56(84) bytes of data.
64 bytes from mad41s13-in-f4.1e100.net (142.250.200.100): icmp_seq=1 ttl=115 time=30.4 ms
64 bytes from mad41s13-in-f4.1e100.net (142.250.200.100): icmp_seq=2 ttl=115 time=30.8 ms
64 bytes from mad41s13-in-f4.1e100.net (142.250.200.100): icmp_seq=3 ttl=115 time=30.7 ms
64 bytes from mad41s13-in-f4.1e100.net (142.250.200.100): icmp_seq=4 ttl=115 time=30.8 ms
^C
--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 30.404/30.673/30.823/0.164 ms
[root@localhost ~]# ping 10.22.147.18 [REDACTED]
PING 10.22.147.18 (10.22.147.18) 56(84) bytes of data.
64 bytes from 10.22.147.18: icmp_seq=1 ttl=64 time=0.140 ms
64 bytes from 10.22.147.18: icmp_seq=2 ttl=64 time=0.376 ms
64 bytes from 10.22.147.18: icmp_seq=3 ttl=64 time=0.392 ms
64 bytes from 10.22.147.18: icmp_seq=4 ttl=64 time=0.396 ms
64 bytes from 10.22.147.18: icmp_seq=5 ttl=64 time=0.396 ms
^C
```

Figura 4. Ejecución de tres comandos ping para comprobar la conductividad de red.

6. Fuentes de información

Referencias

- [1] A. Q. R. García, “Práctica 1: Instalación de la plataforma de virtualización KVM. Creación y configuración de una máquina virtual,” Universidad de Las Palmas de Gran Canaria, 2024.
- [2] [Online]. Available: <https://documentation.suse.com/sles/12-SP5/html/SLES-all/cha-libvirt-overview.html#:~:text=Use%20the%20following%20commands%20to,%2Fsystemd%2Fsystem%2Flibvirtd..>
- [3] [Online]. Available: <https://fraterneo.blogspot.com/2015/05/virtualizando-con-kvm-qemu-libvirt-en-fedora-centos.html>.

Práctica 2: Operaciones con máquinas virtuales

Óscar Alexander Martín Tacoronte

Grupo 01.04 (Cod. 44)

28 de febrero de 2024

1. Índice

1.	Índice	2
2.	Introducción	3
3.	Objetivos	4
3.1	Copia de seguridad y creación de una máquina virtual a partir de ficheros ..	4
3.2	Clonación de una máquina virtual	4
3. 3	Creación e instalación de una máquina virtual.....	4
4.	Desarrollo.....	5
4.1	Copia de seguridad y creación de una máquina virtual a partir de ficheros ..	5
4.2	Clonación de una máquina virtual	6
4.3	Clonación de una máquina virtual	7
5.	Pruebas / Validación.....	8
5.1	Comprobación de la creación de las máquinas virtuales.....	8
5.2	Verificación de la conectividad de las máquinas creadas	8
5.3	Comprobación del acceso SSH desde el host anfitrión	9
6.	Fuentes de información	11
	Referencias	11

2. Introducción

En esta práctica se enfocará en la realización de operaciones con máquinas virtuales haciendo uso de herramientas instaladas de virtualización como **virt-manager**, **virsh** y **virt-install**. Así mismo, con esta actividad se pretende familiarizarse con el uso de gestión tanto desde el entorno gráfico como desde la línea de comando.

3. Objetivos

3.1 Copia de seguridad y creación de una máquina virtual a partir de ficheros

- Realizar una copia de seguridad de una máquina virtual a partir de los ficheros
- Crear una máquina virtual a partir de la copia de seguridad a partir de los ficheros
- Utilizar la orden **virsh** en la creación de la máquina virtual

3.2 Clonación de una máquina virtual

- Realizar la de una máquina virtual mediante la herramienta gráfica **virt-manager**
- Realizar la de una máquina virtual mediante la herramienta de comando desde el shell **virt-clone**

3. 3 Creación e instalación de una máquina virtual

- Crear una máquina virtual mediante la herramienta de comando desde el shell **virt-install** a partir de unas especificaciones proporcionadas.

4. Desarrollo

4.1 Copia de seguridad y creación de una máquina virtual a partir de ficheros

En primer lugar, debido a que durante la práctica anterior no se seleccionó la configuración que instala SSH (*SSH - Secure SHell*) al instalar la máquina virtual, se procede a instalar el servicio de SSH siguiendo las instrucciones [1] ya que es necesario para cumplir con uno de los requisitos de la práctica [2]. Se instala, por tanto, servicio SSH y para que se ejecute por defecto se emplea el comando `systemctl` con el objetivo de que en cada arranque de la máquina virtual se inicie el servicio SSH de forma automática con el siguiente conjunto de comandos:

```
# dnf -y install openssh-server openssh-clients  
# systemctl enable sshd.service  
# systemctl start sshd.service
```

A partir de este punto, se procede a realizar una copia de seguridad de una máquina virtual. De esta manera, se navega hasta el directorio donde se pueden localizar los archivos de definición con extensión XML (*XML – Extensible Markup Language*) de las máquinas virtuales en la ruta `/etc/libvirt/qemu`. También se navega a la ruta donde se encuentra los discos de almacenamiento de las máquinas virtuales con extensión `qcow2` en la ruta `/var/lib/libvirt/images` y se realiza una copia con los siguientes comandos:

```
# cp MinimalFedoraServer39.qcow2 Clon_copiando_ficheros.qcow2  
# cp MinimalFedoraServer39.xml Clon_copiando_ficheros.xml
```

Al copiar el fichero de definición, es de necesidad también modificar este fichero ya que al hacer la copia estáticamente, se tiene que modificar el identificador de máquina virtual (*uuid*), así como la dirección MAC (*MAC – Media Access Control*) de la máquina virtual y el campo de dominio. Además, es necesario modificar la variable que define el nombre del archivo de disco de almacenamiento en el archivo de definición para que la máquina virtual seleccione al momento de arrancar la copia del disco en caso de clonar también el disco, que en esta práctica también se realiza. Para proceder con estas tareas, se modifica estas variables con el editor clásico de un sistema UNIX en el archivo de definición en la ruta `/etc/libvirt/qemu`:

```
# vi Clon_copiando_ficheros.xml
```

El identificador de la máquina virtual ha sido generado a partir de una de las utilidades que ofrece Linux según [3] con el siguiente comando:

```
# cat /proc/sys/kernel/random/uuid
```

En cuanto a la generación de la MAC aleatoria ha sido a partir de la herramienta ofrecida durante la práctica, y obtenida con la siguiente orden:

```
# python macgen.py
```

Una vez realizada la copia de seguridad de la máquina, se debe crear la máquina virtual a partir del fichero de definición. De esta manera, se realiza con la orden virsh:

```
# virsh create Clon_copiando_ficheros.xml
```

Por un lado, se ha creado la máquina virtual a partir de una copia de un archivo de definición que se ha modificado, por otro lado, se ha clonado la imagen del disco y dentro del archivo de definición apunta a la clonación de este disco recientemente clonado. En la creación con “**virsh**”, se podido utilizar la opción “**virsh define**” con el objetivo de que se cree la definición en la utilidad gráfica de manera automática, sin embargo, al usar la opción “**virsh create**” no se define de forma instantánea en la utilidad grafica.

Con esto se cumple el objetivo 3.1

4.2 Clonación de una máquina virtual

En este punto, se puede hacer una clonación de forma automática. Por un lado, a partir de la herramienta gráfica, siguiendo los diálogos gráficos que muestra la propia herramienta se completa de forma muy sencilla y de forma automática sin embargo no ofrece una alta personalización. Por otro lado, a partir desde el *shell* se puede realizar la misma tarea con la orden “**virt-clone**”, que realizará una copia de la máquina virtual como en el objetivo 3.1 pero de forma automática sin tener que modificar manualmente valores como la MAC de la máquina virtual, o el identificador de máquina virtual, entre otras opciones. La siguiente orden con la orden “**virt-clone**” hará la clonación de la máquina virtual original de *MinimalFedoraServer39* y dicha copia de nombrará como, y con la opción “--

“**auto-clone**” realizará las modificaciones necesarias como se comentó previamente de forma interna en las librerías de *KVM*:

```
# virt-clone -o MinimalFedoraServer39 -n Clon_virt_manager --  
auto-clone
```

De esta manera, se cumple los objetivos declarados en el punto 3.2.

4.3 Clonación de una máquina virtual

Para finalizar esta práctica, se creará una máquina virtual desde la línea de comando, la cual es una utilidad muy interesante para personalizar en mayor medida la maquina virtual que no abarca en su completitud la utilidad gráfica. De esta manera, en primer lugar, se monta la imagen en el sistema anfitrión con la siguiente orden:

```
# mount 10.22.146.215:/imágenes/fedora/39/isos/x86_64 iso
```

En segundo lugar, se procede a la creación de la máquina virtual con la orden “**virt-install**”. Además, se especifica las características deseadas de la nueva máquina. Sin embargo, las características necesarias como la interfaz de red, pero no especificadas se completará con los valores por defecto según la utilidad “**virt-install**”. En este caso, y según la fuente [4] en el punto 3.2.6, la red por defecto es *NAT* y por tanto no se especificó en la ejecución de la orden.

```
# Virt-install -name Creación_virt_install -memory 2048 -disk  
size=10 -vcpus 1 -network default -cdrom iso/Fedora-Server-  
netinst-x86-39-1.5.iso -os-variant fedora39
```

Al crear esta máquina virtual y posteriormente al proceder a la instalación mínima de *Fedora39*, se cumple el objetivo 3.3.

Figura 1. Rellenar

5. Pruebas / Validación

5.1 Comprobación de la creación de las máquinas virtuales

En la figura 1 se muestra en la utilidad gráfica la clonación y creación de las máquinas virtuales que se han creado durante la práctica con los nombres que identifican el método de su procedimiento de operación.

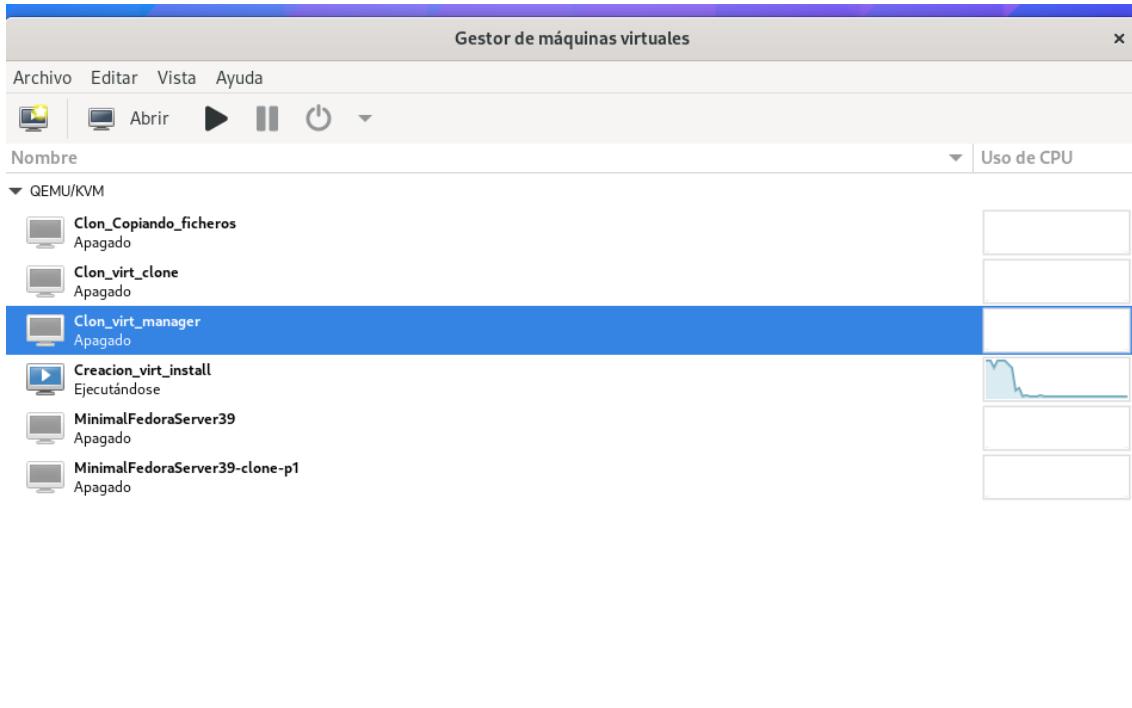


Figura 1. Listado de máquinas virtuales en la utilidad gráfica.

5.2 Verificación de la conectividad de las máquinas creadas

Se procede a enviar trazas ICMP (*ICMP – Internet Control Message Protocol*) con el comando ping para verificar la conectividad con el exterior utilizando el nombre de dominio con el objetivo de verificar también la configuración DNS (*DNS – Domain Name Server*), dicha verificación se puede observar en la imagen 2 donde se muestra la conectividad de una máquina virtual. Se ejecuta por tanto el siguiente comando:

```
# ping Google.com
```

```
[root@fedora ~]# ping google.com
PING google.com (142.250.185.14) 56(84) bytes of data.
64 bytes from mad41s11-in-f14.1e100.net (142.250.185.14): icmp_seq=1 ttl=115 time=30.5 ms
64 bytes from mad41s11-in-f14.1e100.net (142.250.185.14): icmp_seq=2 ttl=115 time=30.8 ms
^C
--- google.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 30.520/30.652/30.784/0.132 ms
[root@fedora ~]#
```

Figura 2. Verificación de la conectividad y configuración DNS.

5.3 Comprobación del acceso SSH desde el host anfitrión

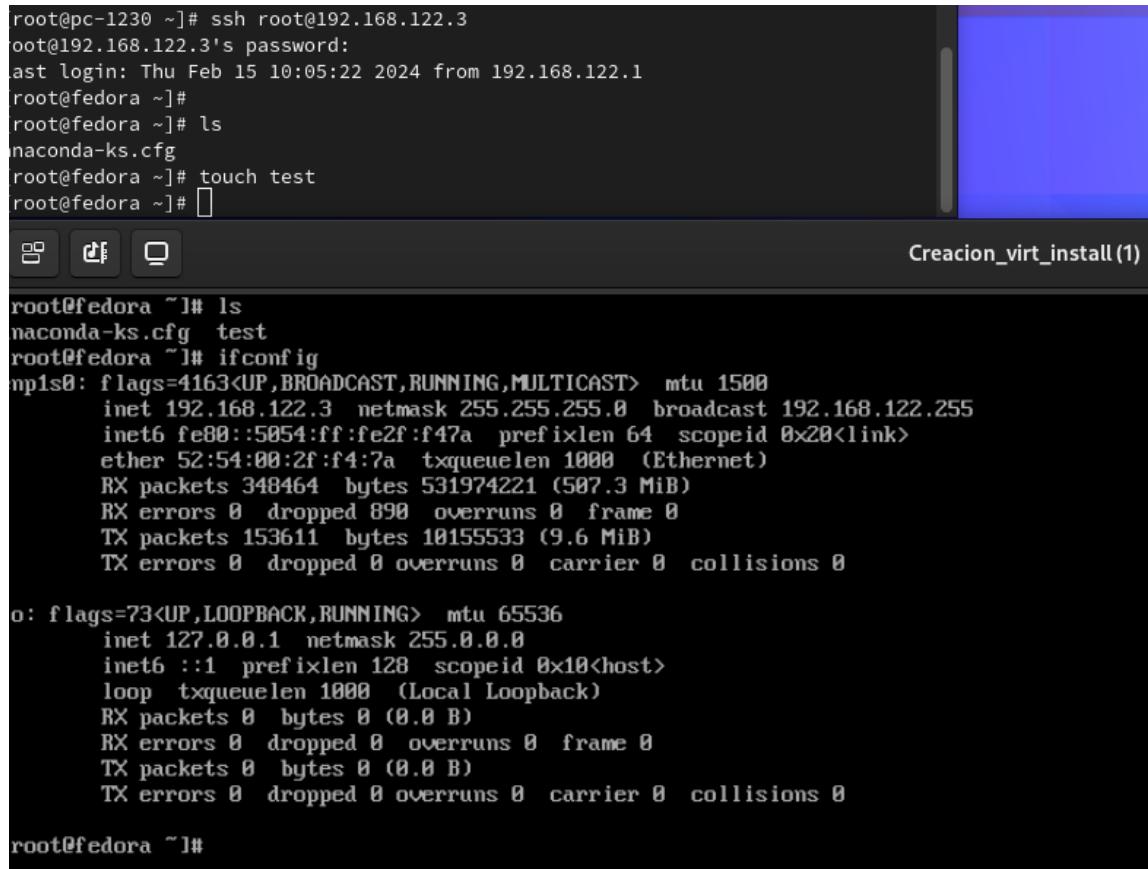
Por último, se verifica el acceso SSH desde el anfitrión hacia la máquina virtual. Para ello, se ejecuta la orden SSH y se crea un fichero de prueba en la ruta /root/ llamado *test* y se verifica en el sistema invitado con la orden “**ifconfig**” su IP para que pueda ser accedido desde el sistema anfitrión, y también obtenemos el listado del directorio /root/para verificar si se ha creado el fichero de prueba desde el sistema anfitrión. Por tanto, en el sistema anfitrión se ejecuta:

```
# ssh root@192.168.122.3
# touch test
```

Mientras que en el sistema invitado se lanza las siguientes ordenes:

```
# ifconfig
# ls
```

La verificación se puede apreciar en la figura 3



The screenshot shows a terminal window titled "Creacion_virt_install (1)". The terminal displays the following SSH session logs:

```
root@pc-1230 ~]# ssh root@192.168.122.3
root@192.168.122.3's password:
Last login: Thu Feb 15 10:05:22 2024 from 192.168.122.1
root@fedora ~]#
root@fedora ~]# ls
naconda-ks.cfg
root@fedora ~]# touch test
root@fedora ~]# [REDACTED]
```

Below this, the terminal shows the output of the "ifconfig" command:

```
root@fedora ~]# ls
naconda-ks.cfg  test
root@fedora ~]# ifconfig
mp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.122.3 netmask 255.255.255.0 broadcast 192.168.122.255
        inet6 fe80::5054:ff:fe2f:f47a prefixlen 64 scopeid 0x20<link>
            ether 52:54:00:2f:f4:7a txqueuelen 1000 (Ethernet)
            RX packets 348464 bytes 531974221 (507.3 MiB)
            RX errors 0 dropped 890 overruns 0 frame 0
            TX packets 153611 bytes 10155533 (9.6 MiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
            RX packets 0 bytes 0 (0.0 B)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 0 bytes 0 (0.0 B)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@fedora ~]#
```

Figura 3. Verificación de acceso SSH

6. Fuentes de información

Referencias

- [1] Angel, “INSTALANDO SSH EN FEDORA, CENTOS O REDHAT,” uGeek, 2019.
 -] [Online]. Available: <https://ugeek.github.io/blog/post/2019-12-11-instalando-ssh-en-fedora-centos-o-redhat.html>.
- [2] A. Q. R. García, “Práctica 2: Operaciones con máquinas virtuales.,” Universidad de Las Palmas de Gran Canaria, 2024.
- [3] “Generate a UUID in Bash,” TransparenTech LLC, 2024. [Online]. Available:
 -] <https://www.uuidgenerator.net/dev-corner/bash>.
- [4] P. S. Y. Z. L. N. D. P. S. R. T. R. Jiri Herrmann, “Virtualization Deployment and Administration Guide,” RED HAT ENTERPRISE LINUX 7, [Online]. Available: https://access.redhat.com/documentation/es-es/red_hat_enterprise_linux/7/html/virtualization_deployment_and_administration_guide/index. [Accessed 15 Febrero 2024].

Práctica 3: Migración de máquinas virtuales

Óscar Alexander Martín Tacoronte

Grupo 01.04 (Cod. 44)

10 de marzo de 2024

1. Índice

1.	Índice	2
2.	Introducción	3
3.	Objetivos	4
3.1	Establecimiento de la conexión	4
3.2	Realizar la migración de una máquina virtual.....	4
4.	Desarrollo.....	5
4.1	Establecimiento de la conexión	5
4.2	Realizar la migración de una máquina virtual.....	7
5.	Pruebas / Validación.....	9
5.1	Verificación de la migración de una MV empleando el servidor de.....	9
	almacenamiento compartido disnas.dis.ulpgc.es	9
6.	Fuentes de información	11
	Referencias	11

2. Introducción

Para esta práctica se deberá migrar máquinas virtuales entre distintos anfitriones. Se utilizará un espacio de almacenamiento compartido mediante NFS en un servidor intermediario entre las dos máquinas anfitriones y se deberá configurar la conexión para la comunicación mediante SSH entre anfitriones.

3. Objetivos

3.1 Establecimiento de la conexión

- Declaración del nombre de dominio del anfitrión emisor y asignación del nombre de dominio del anfitrión receptor
- Configuración de la conexión SSH

3.2 Realizar la migración de una máquina virtual

- Clonación de la máquina virtual en el servidor compartido
- Habilitar los puertos de migración en el receptor de la migración
- Realizar la migración a partir de la utilidad gráfica *virt-manager*

4. Desarrollo

4.1 Establecimiento de la conexión

En primer lugar, se debe asignar el nombre de dominio del anfitrión emisor y receptor. En este caso, se hará uso de la orden `hostnamectl` y se asignará como nombre de anfitrión **pc1230.vpd.com** con el objetivo de que el anfitrión receptor pueda reconocer el anfitrión emisor, mientras que el anfitrión receptor será **pc1231.vpd.com**. Se ejecuta según [1] los siguientes comandos para hacer dicha asignación y comprobar su resultado

```
# hostnamectl set-hostname pc1230.vpd.com  
# hostnamectl status
```

En segundo lugar, se debe configurar el nombre de dominio del anfitrión receptor para que al momento de realizar la migración se introduzca su nombre de dominio **pc1231.vpd.com** en lugar de su dirección IP. Para ello se sigue las instrucciones de [2], por tanto, se modifica el archivo de configuración `/etc/hosts` con el siguiente comando:

```
# vi /etc/hosts
```

Por tanto, se asignará la IP del servidor **10.22.147.19** al nombre de dominio **pc1231.vpd.com** añade la siguiente línea al archivo `/etc/hosts`, y puede ser verificado en el apartado 5:

```
# 10.22.147.19      pc1231.vpd.com
```

Respecto a la configuración de la conexión SSH (*SSH- Secure SHell*), se procede a realizar una autenticación mediante clave pública y privada. Para ello es necesario que cada anfitrión genere dicho par de claves con la orden `ssh-keygen` y se empleará un algoritmo de generación de clave RSA con 2048 bits de longitud siguiendo las instrucciones de [3] ejecutando el siguiente comando:

```
# ssh-keygen -t rsa -b 2048
```

Llegados a este punto, el sistema anfitrión emisor debe compartir su clave pública con el sistema anfitrión receptor para que pueda ser autenticado, y para ello en el sistema receptor debe tener almacenada la clave pública en el archivo `/root/.ssh/authorized_keys` para que pueda reconocer al cliente mientras que en el sistema emisor debe almacenar la clave pública del servidor en el

archivo /root/.ssh/known_hosts para que pueda reconocer al sistema receptor de la migración. Las claves públicas son necesarias para evitar un ataque Hombre en el Medio (“*Man In the Middle*”) y autenticarse entre anfitriones de manera segura.

Para compartirse las claves públicas, se puede proceder de dos maneras. Por un lado, compartir la clave pública de forma manual que se encuentra en /root/.ssh/id_rsa.pub y añadirla al directorio del sistema anfitrión receptor en /root/.ssh/authorized_keys; y obtener la clave pública del sistema receptor y añadirla a /root/.ssh/known_hosts. Durante esta práctica **no** se procedió de esta forma.

Por otro lado, también se puede uso de la herramienta ssh-copy-id, y posteriormente realizar la conexión SSH. De este modo, copiará automáticamente las llaves a dichos ficheros del nombre de dominio **pc1231.vpd.com** para autenticarse en el usuario **root**. Cabe destacar que, sin embargo, que el usuario del sistema anfitrión receptor debe introducir su contraseña manualmente. Durante la práctica **sí** se procedió de esta forma, en la **figura 1** se podrá observar el contenido de los ficheros de configuración donde se podrá comprobar en resaltado en color rojo las partes relevantes la verificación de su correcta configuración. De esta manera, se ejecuta los siguientes comandos en ambos sistemas tanto emisor como receptor cada uno con el nombre de dominio del otro sistema:

```
# ssh-copy-id root@pc1231.vpd.com
# ssh root@pc1231.vpd.com
```

The screenshot shows a terminal window with two sessions. The top session is for user 'root' on host 'pc1230'. It shows the command 'ssh-copy-id' being run to copy the public key of the local machine to the remote host 'pc1231.vpd.com'. The bottom session is for user 'root' on host 'pc1231.vpd.com'. It shows the command 'ssh' being run to log in to the remote host 'pc1230'. Both sessions show the password prompt for the root user.

```
[root@pc1230 ~]# cat /root/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQgDQylJx7nGkaRbyNGwgUyCAI/0HZ3wNLqEeumndgC2l+CDskfbK1EuJg6rlumWllfxZPHKDCB8KxH7GtN/H+Z
0-X0352ILUKV0GZYzUJMn00ZpY3GJDxDnba+858DhFQSjRKmYslmaVsNVA/nEBQzvnhIMhkb0p8Xazddm-TpZ+qQR0ourapvnhr+ONORTciUzYrbB0fLdnjc
ce/tD0wkoMUHg4Vtvze+Ao/Dt94lgf1+r2TWoE7NXrmDQ0iJhYUSBokZ69lsXg4G4bYNXAPU7nf4VZ0alIB/Zp85TAjRZhagtr5M+aC8yU7xEwS
7esqhfMa0zBqzOXARhaRv/DgZli80YAKNzLtfbBndreNrqrz42I62cgeUeEx4Z7hsp9v/+4limY7sPEY94/2hd3wFLPN+w+TT0aYBkUC/9mMNGycCnyCe
0mIjk2isknx52y/EkTC+s970bklyla+PSVxhfn6f+Loro9nk7VzdZueTo/I0tTVE= root@pc1231.vpd.com
[root@pc1230 ~]# cat /root/.ssh/known_hosts
192.168.122.3 ssh-ed25519 AAAAC3NzaC1lZDI1NTESAAAIETrUoH2Q2bThEyRmIab92e5y1kTYUCUP06ckCBSTwVo
192.168.122.3 ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQgC88cRxfdDhNxcCl/03czztg/1Ncf2cYwc0tcEg59WvbgWpuvHQYUpFZnd9u110yjGCh
efGnfO4A+rUVnsmEDptqbF4zmw48N40YqMmCKM8esCQjzkAvn50WjCt/ey4/zEkk44WBRTGrZ+MTyhESrdT/MW+/g4llITKRuKCKl00M8QplfzeFieRRFhvr
DuFxrb14zMyefk6Iur2lzaNWlrzP031vwLhco5975uwOpE1hv+GzA/sss4xuvv/RZ1p9yh6t4g4F6VQJdqZBI9TImso12GF/wd0aa+jixtXyQeqW57vjo9wnT
Pk+ZuqdjWKJ3qGvtfc5dWMyCmCyfadl3v7vQcmAe5Z5iu1QSzXeGmE+R9H4bV7mfDjbJClPvzTRsqua/0yhddR
2o97V5ogv6Mmlt14Kfmng9gfCTJCN4Bfp/TDNLPqWjwNsahFyIrlaLa0UxtTqvnfmFxw60ujTj8=
192.168.122.3 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAYNTYAAAIBmlzdHAYNTYAAABBBM+w+KwcWSHsdTpCjr8Nq6qYlGueaHIGmdqc
bgBQndN5vHj8CY+X5oyDL0XgPaalFOA3+D9lzpWl0Vjz2RhKY=
disnas.dis.ulpgc.es ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQAcuwd+D0zKgaRwxVuCuvfTuwpOCBzW+sYwvMQuVIuduodB+TgyTWKJ7ZG12crDfE
DLR430FPiIldW6ugEaxXvNT1A0RnF1/8Pqj0ksAGQUDvdEGHHjvxXGHoubDzxaw3ZC2BX7C5skjayQsh5nDhRBcnih0pidp+xjXaoSSzySzFz32ykjdjzuov7
Mtxee5xalb3M08HFkomgZHks+EbPROXOXjBhi6hbVaEWhzPkGZ6Qh0LdkfEG0s/1+F4WD44jinQZVs2ISiC/RLENA6qU6b/vf5HQ2CJuGNTVdtfP0zLVQZ/9E8
cmny00FXrVuNggSx7tEEYyGxpyzfCvMwvb
10.22.147.19 ssh-ed25519 AAAAC3NzaC1lZDI1NTESAAAIEBng6K3VVfRl0qcCCykaSbTbrdmjfCKdJbm+Xr1qUsx
10.22.147.19 ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQgC0DmG7ffBj/nCruxnF/2pSha08mMsfcAdwvSGxPcmb4QbdLvv10q1zZYEcgiUiparf+BeP+
JvPtAU5rUfe0SoQh3rN9vRMrY5Z/p3M0QRBVpTZItyH7oURNmLrfMR3svEumYZH8tRZ890vsEWKdbhQvV5glWd4muK0+CrxCNnueIFQdHA8ftu012lmMKgtVB
+Jatwjaja55j6+rqHTqvzTlooniAuqMSH2YqtDFSVhqPsP/070whYc5k4SMY5HApJbV3gfJFrFVvfCunFVQfcQ9v5hegn4NfKxdidi1Kg0qPnzxJCU7cEtj3hL6u
J0Nl3Ey7S55UsbgHxGmcde04ve/cjrfELK+lhf7Uh2G1k3G1M4B/HeV3Y8DV6srkRejVFThn8T1rtGN21ZIXigE+lmURbj6q2WRKRSLuhb1xa887ZPwUjmpj6c
5cL8q/23D0WeNbttogSooEfYsy1ZstUS1sqj8wryRLVGZPuhvutnjF7IawM0lfzs+8n8h9vk43E=
10.22.147.19 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAYNTYAAAIBmlzdHAYNTYAAABBDl0i6AEqlIKM07ffFkGyJcP7Dn26dYDcmZLaT
GKm+1k/FcSd+68Udn5MGa0WhWf1xF5xuQ7prXLIWA3nFYAe+U=
pc1231.vpd.com ssh-ed25519 AAAAC3NzaC1lZDI1NTESAAAIEBng6K3VVfRl0qcCCykaSbTbrdmjfCKdJbm+Xr1qUsx
```

Figura 1. Configuración de los ficheros SSH

Con la conexión configurada, se cumple los objetivos del **punto 3.1**, teniendo en cuenta que la clave privada que se encuentra en `/root/.ssh/id_rsa` debe ser inaccesible.

4.2 Realizar la migración de una máquina virtual

En este apartado se procede a la operación de clonación. En primer lugar, se clona la imagen de la máquina virtual en el servidor compartido NFS proporcionado en [4] y montado en la ruta local `/ISOS/` con los siguientes comandos:

```
# mkdir /ISOS/
# mount -t nfs disnas.dis.ulpgc.es:/disnas-itsi /ISOS/
# cp /var/lib/libvirt/images/MinimalFedorServer39.qcow2
/ISOS/pc1230_Lab21_ANFITRION1_practica_migracion.qcow2
```

Posteriormente, en la utilidad gráfica, en la **figura 2** se procede a la creación de una máquina virtual seleccionando la opción “Importar imagen de disco existente”, y se importa la imagen de disco existente seleccionando la que se encuentra en el directorio `/ISOS/pc1230_Lab21_ANFITRION1_practica_migracion.qcow2`.

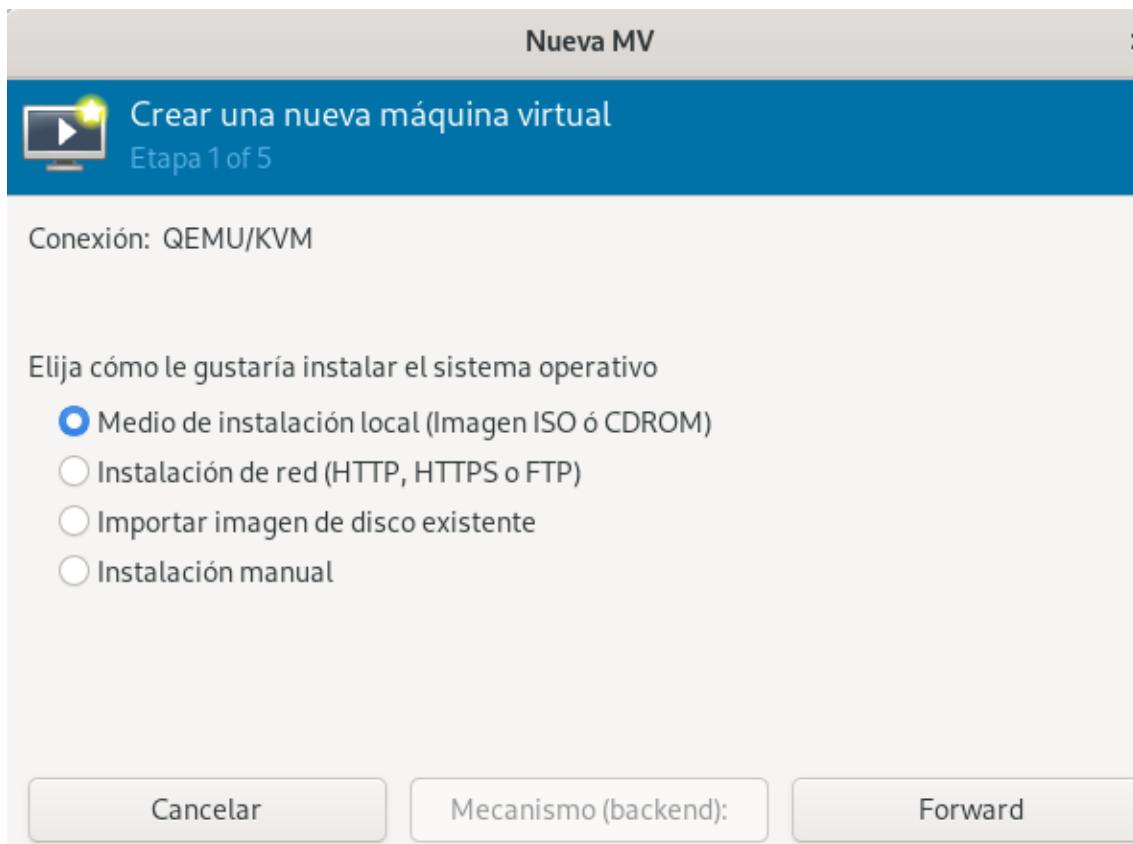


Figura 2. Dialogo gráfico para importar una imagen de disco.

A partir de este punto, es necesario habilitar los puertos de migración 49152 a 49216 en el sistema receptor con el siguiente comando según [5]:

```
# firewall-cmd --permanent --add-port=49152-49216/tcp
```

A continuación, ya es posible realizar la conexión SSH, por lo que se accede al dialogo grafico de la **figura 3**, rellenando el campo Nombre del equipo primero con la dirección IP y segundo con el nombre de dominio de la máquina receptora.

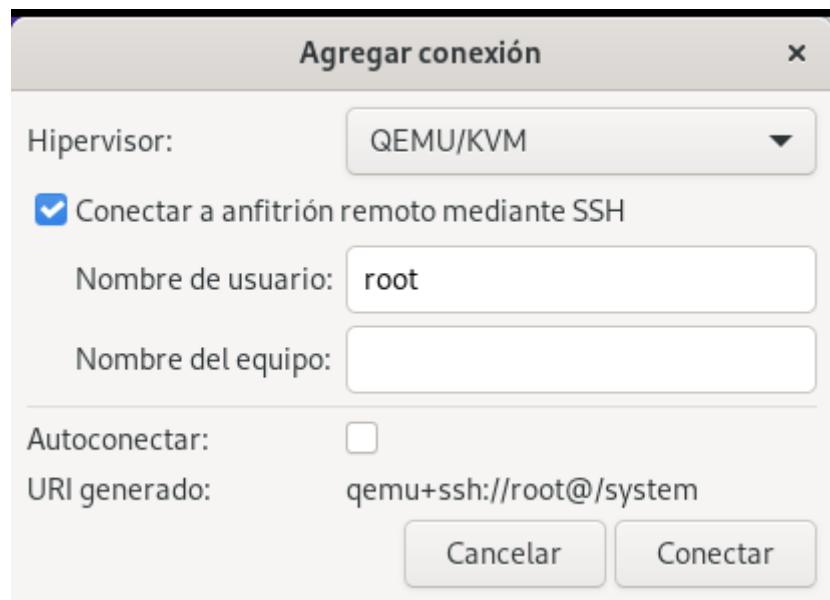


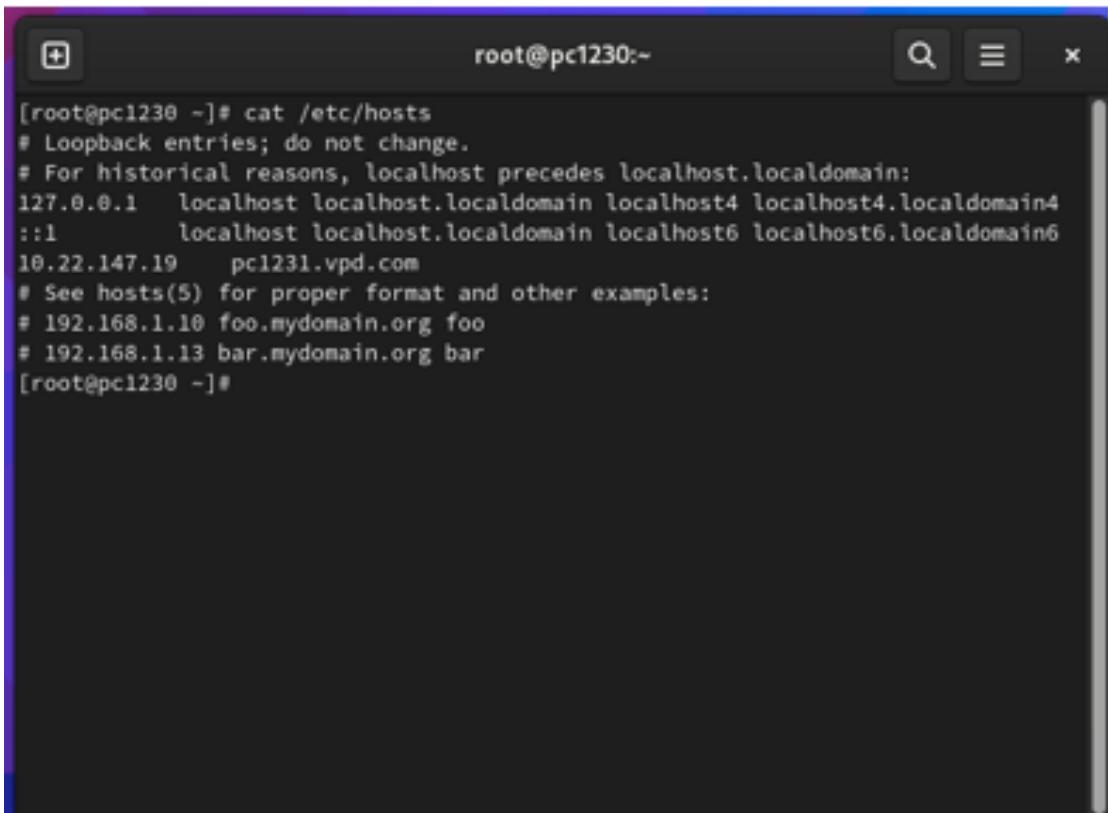
Figura 3. Dialogo gráfico para realizar la conexión entre dos máquinas anfitrionas.

Con la máquina en ejecución, desde la utilidad gráfica se selecciona la opción de Clonar, y se continua con el dialogo gráfico. Una vez clonada, se observa que en el sistema anfitrión emisor deja de ser visible su definición y continúa su ejecución en el sistema anfitrión receptor.

5. Pruebas / Validación

5.1 Verificación de la migración de una MV empleando el servidor de almacenamiento compartido disnas.dis.ulpgc.es

En la **figura 4** se podrá observar la correcta configuración de los *hosts* configurados en una de las máquinas, en la dirección /etc/hosts. Dicho fichero muestra las máquinas reconocidas con su dirección IP y su nombre de dominio.



```
[root@pc1230 ~]# cat /etc/hosts
# Loopback entries; do not change.
# For historical reasons, localhost precedes localhost.localdomain:
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1          localhost localhost.localdomain localhost6 localhost6.localdomain6
10.22.147.19  pc1231.vpd.com
# See hosts(5) for proper format and other examples:
# 192.168.1.10 foo.mydomain.org foo
# 192.168.1.13 bar.mydomain.org bar
[root@pc1230 ~]#
```

Figura 4. Verificación de las conexiones SSH.

Para finalizar, en la **figura 5** se aprecia las dos conexiones realizadas tanto a su propia dirección IP del receptor, como a su propio nombre de dominio. Además, desde la máquina receptoría también se realizó una migración al final de la práctica por lo que la máquina receptoría también fue una máquina emisora y es el sistema invitado pc1230_LAB21_ANFITRION1_practica_migracion.qcow2.

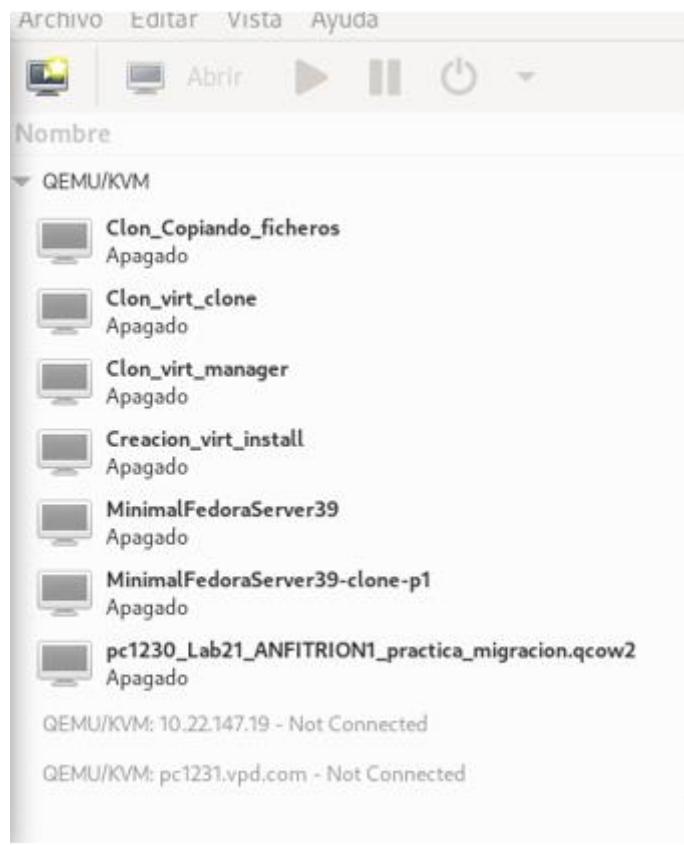


Figura 5. Verificación de las máquinas virtuales y las conexiones realizadas.

6. Fuentes de información

Referencias

- [1] DAVIDOCHOBITS, “Hostnamectl: Configurar hostname en Centos, RHEL y Fedora,” Ochobitshacenunbyte, 21 Enero 2021. [Online]. Available: <https://www.ochobitshacenunbyte.com/2020/01/21/hostnamectl-configurar-hostname-en-centos-rhel-y-fedora/>.
- [2] A. Kili, “How to Setup Local DNS Using /etc/hosts File in Linux,” TecMint, 25 Julio 2017. [Online]. Available: <https://www.tecmint.com/setup-local-dns-using-etc-hosts-file-in-linux/>.
- [3] “How to Use ssh-keygen to Generate a New SSH Key?,” SSH Academy, 2023. [Online]. Available: <https://www.ssh.com/academy/ssh/keygen>.
- [4] A. Q. R. García, “Práctica 3: Migración de máquinas virtuales.” Universidad de Las Palmas de Gran Canaria, 2024.
- [5] “Edición de puertos de cortafuegos,” IBM, 2 Marzo 2021. [Online]. Available: <https://www.ibm.com/docs/es/spp/10.1.6?topic=plus-editing-firewall-ports>.

Práctica 4: Recursos de almacenamiento virtual

Óscar Alexander Martín Tacoronte

Grupo 01.04 (Cod. 44)

2 de abril de 2024

1. Índice

1.	Índice	2
2.	Introducción	3
3.	Objetivos	4
3.1	Almacenamiento local.....	4
3.2	Almacenamiento en red.....	4
4.	Desarrollo.....	5
4.1	Almacenamiento local.....	5
4.1.1	Creación de un volumen y asociarlo a una máquina virtual.	5
4.1.2	Particionar un disco físico y asociarlo a una máquina virtual	7
4.1.3	Creación de un contenedor de almacenamiento para sistemas invitados	10
4.2	Almacenamiento en red.....	13
4.2.1	Contenedor NFS de imágenes ISO	13
4.2.2	Contenedor NFS para volúmenes de máquinas virtuales	16
5.	Pruebas / Validación.....	19
5.1	Almacenamiento de datos desde las máquinas virtuales	19
5.2	Contenedor NFS para volúmenes de máquinas virtuales.....	23
6.	Fuentes de información	25
	Referencias	25

2. Introducción

En esta práctica se abordará un servicio de almacenamiento para los sistemas invitados. Este servicio será, por un lado, desde un recurso físico del sistema anfitrión, y, por otro lado, desde un recurso de almacenamiento remoto al sistema anfitrión.

3. Objetivos

3.1 Almacenamiento local

- Creación de un volumen y asociarlo a una máquina virtual.
- Particionar un disco físico y asociarlo a una máquina virtual.
- Creación de un contenedor de almacenamiento para sistemas invitados.

3.2 Almacenamiento en red

- Contenedor NFS de imágenes ISO
- Contenedor NFS para volúmenes de máquinas virtuales

4. Desarrollo

4.1 Almacenamiento local

4.1.1 Creación de un volumen y asociarlo a una máquina virtual.

En esta primera parte, se creará un nuevo volumen de un 1 GB, por un lado, con nombre `Vol1_p4` y por otro lado con nombre `Vol1_p4_virtmanager`. El primero volumen `Vol1_p4` se tratará por la línea de comandos, y, por otro lado, `Vol1_p4_virtmanager` se configurará desde la utilidad gráfica de `virt-manager`.

En primer lugar, se crea el volumen de 1 GB fácilmente con el siguiente comando y lo almacenará en el contenedor por defecto.

```
# virsh vol-create-as --name Vol1_p4 --capacity 1GB --pool default
```

Una vez creado, es necesario adjuntarlo a una máquina virtual en ejecución. En este caso se le asigna a la máquina `Creación_virt_install` con la opción `--config` se modificará de manera permanente, con la opción `--live` indica que el disco se debe adjuntar en vivo y se especifica el nombre del dispositivo `vdd` como disco dentro de la máquina virtual a la que se va a adjuntar el volumen, sin embargo, se recomienda no asociar que este volumen `Vol1_p4` se asocia a este dispositivo `vdd` durante toda la práctica porque **durante la ejecución de la práctica se hicieron cambios que modifican este tipo de ajustes**. De esta manera, se ejecuta la siguiente orden tal y como provee la fuente [1]:

```
# virsh attach-disk --config Creación_virt_install --source /var/lib/libvirt/images/Vol1_p4 --target vdd --live
```

Finalmente, con la ejecución de la siguiente orden, se verá en la figura 1 una lista de los dispositivos de almacenamiento de la máquina virtual.

```
# virsh domblklist Creacion_virt_install --details
```

```
[root@pc1230 ~]# virsh domblklist Creacion_virt_install --details
  Tipo   Dispositivo   Destino   Fuente
  -----
  file    disk         vda      /var/lib/libvirt/images/Creacion_virt_install.qcow2
  file    disk         vdd      /var/lib/libvirt/images/Vol1_p4_virt_manager.qcow2
  file    cdrom        sda      -
```

Figura 1. Información detallada sobre todos los dispositivos de almacenamiento asociados a una máquina virtual.

En segundo lugar, se procede a la creación del volumen a través del **virt-manager**. Para ello, se accede a las opciones **Editar – Detalles de la conexión – Almacenamiento**, y se abre el diálogo gráfico que se observa en la **figura 2** con las configuraciones visibles en la misma.

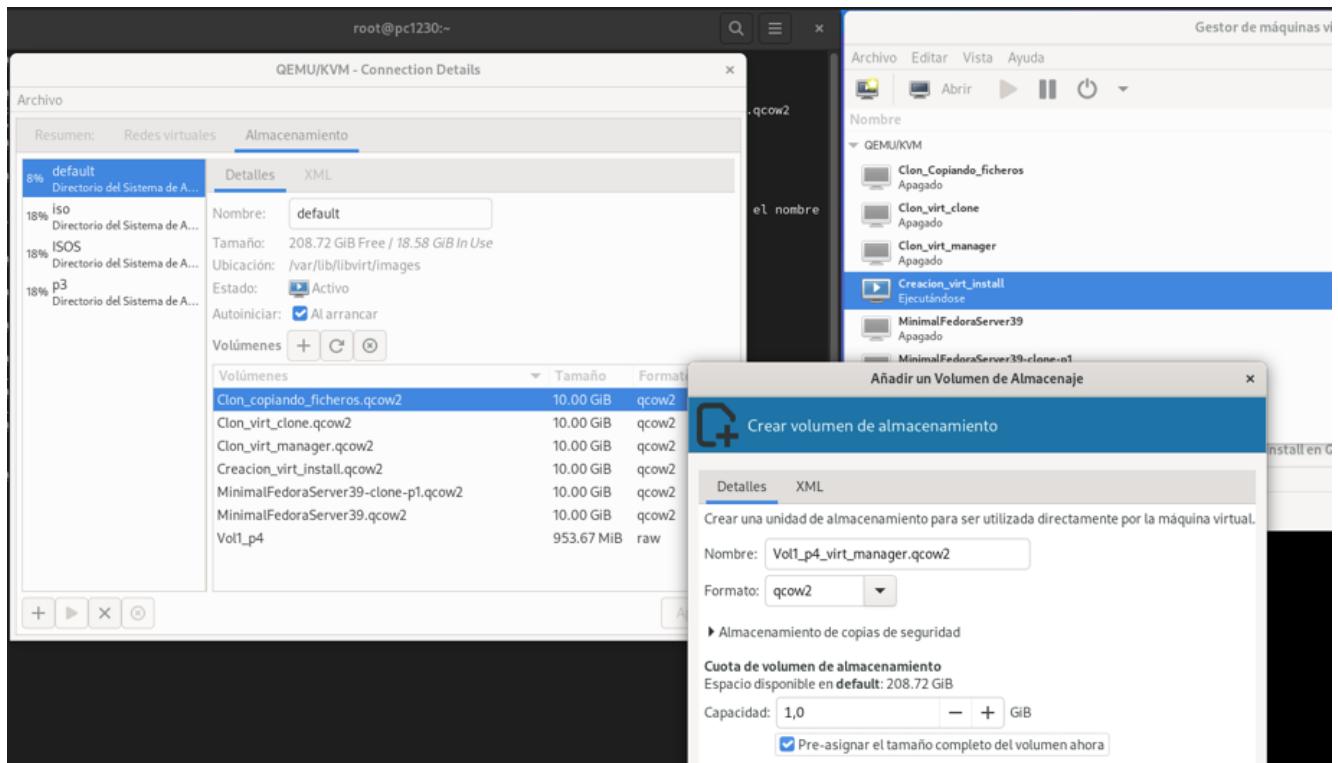


Figura 2. Creación de un volumen por **Virt-manager**.

Para terminar en este apartado, se consultará una lista de todos los volúmenes que estén almacenados en el contenedor de volúmenes por defecto. Se observará en la **figura 3** los dos volúmenes recién creados, y se apreciará que el volumen que se creo desde el **shell**, al **no especificar** el tipo de formato se creó de tipo **raw**, mientras que el volumen creado desde el **virt-manager** es de formato **qcow2**. Se ejecuta el siguiente comando:

```
# virsh vol-list --pool default
```

```

root@pc1230 ~]# virsh vol-list --pool default
  Nombre          Ruta
-----
Clon_copiando_ficheros.qcow2   /var/lib/libvirt/images/Clon_copiando_ficheros.qcow2
Clon_virt_clone.qcow2          /var/lib/libvirt/images/Clon_virt_clone.qcow2
Clon_virt_manager.qcow2        /var/lib/libvirt/images/Clon_virt_manager.qcow2
Creacion_virt_install.qcow2   /var/lib/libvirt/images/Creacion_virt_install.qcow2
MinimalFedoraServer39-clone-p1.qcow2 /var/lib/libvirt/images/MinimalFedoraServer39-clone-p1.qcow2
MinimalFedoraServer39.qcow2    /var/lib/libvirt/images/MinimalFedoraServer39.qcow2
Vol1_p4                      /var/lib/libvirt/images/Vol1_p4
Vol1_p4_virt_manager.qcow2    /var/lib/libvirt/images/Vol1_p4_virt_manager.qcow2

[root@pc1230 ~]#

```

Figura 3. Lista de volúmenes en el contendor por defecto.

4.1.2 Particionar un disco físico y asociarlo a una máquina virtual

En este apartado se particionará un disco de tal manera que se obtendrá una partición lógica nueva de 1GB del sistema anfitrión. El sistema invitado tendrá acceso directamente sobre dicha partición del sistema anfitrión.

Primeramente, se ejecuta `fdisk --list` y se observa que **en la partición secundaria /dev/sdb/ hay una partición extendida**. De esta manera, ejecutando `fdisk /dev/sdb/` se crea una partición lógica `/dev/sdb5` de un 1 GB tal y como se aprecia en la **figura 4**.

```

# fdisk --list
# fdisk /dev/sdb/

```

```

[root@pc1230 ~]# fdisk /dev/sdb

Bienvenido a fdisk (util-linux 2.38.1).
Los cambios solo permanecerán en la memoria, hasta que decida escribirlos.
Tenga cuidado antes de utilizar la orden de escritura.

Este disco está actualmente en uso - no se aconseja volver a crear particiones.
Se recomienda desmontar todos los sistemas de ficheros y deshacer todas las
particiones de intercambio de este disco.

Orden (m para obtener ayuda): n

Se están usando todas las particiones primarias.
Se añade la partición lógica 5
Primer sector (1430806528-1953525167, valor predeterminado 1430806528):
Último sector, +/-sectores o +/-tamaño{K,M,G,T,P} (1430806528-1953525167): + 16B

Crea una nueva partición 5 de tipo 'Linux' y de tamaño 954 MiB.

Orden (m para obtener ayuda): p
Disco /dev/sdb: 931,51 GiB, 1000204886016 bytes, 1953525168 sectores
Modelo de disco: TOSHIBA DT01ACA1
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 4096 bytes
Tamaño de E/S (mínimo/óptimo): 4096 bytes / 4096 bytes
Tipo de etiqueta de disco: dos
Identificador del disco: 0x850b26a5

Disposit. Inicio Comienzo Final Sectores Tamaño Id Tipo
/dev/sdb1      2048 476936191 476934144 227,4G 83 Linux
/dev/sdb2      476936192 953870335 476934144 227,4G 83 Linux
/dev/sdb3      953870336 1430804479 476934144 227,4G 83 Linux
/dev/sdb4      1430804480 1953525167 522720688 249,3G 5 Extendida
/dev/sdb5      1430806528 1432760319   1953792  954M 83 Linux

Orden (m para obtener ayuda): w

```

Figura 4. Creación de una partición lógica.

Como se aprecia en la **Figura 4**, el estado final es una nueva partición `/dev/sdb5` de 1 GB. Una vez creada, se indica la opción “w” para guardar los cambios.

El siguiente paso deberá ser asociar esta partición a una máquina virtual. Para ello se ejecuta el siguiente comando tal como se hizo en el **apartado 4.1.1** pero con la diferencia que ahora el origen es directamente la partición recién creada:

```
# virsh attach-disk -config Creación_virt_install -source /dev/sdb5 -
-target vdc -live
```

Los detalles del disco creado se pueden observar en la **Figura 5**.



Figura 5. Detalles del disco asociado directamente una partición lógica.

4.1.3 Creación de un contenedor de almacenamiento para sistemas invitados

En este apartado, el objetivo será crear un contenedor de almacenamiento, donde sea posible crear volúmenes para los sistemas invitados. Para ello, se creará nuevamente una partición lógica de 2 GB en el sistema anfitrión. Los detalles se pueden observar en la **Figura 6**, siguiendo los mismos pasos que en el **apartado 4.1.2.**

```
[root@pc1230 ~]# fdisk /dev/sdb

Bienvenido a fdisk (util-linux 2.38.1).
Los cambios solo permanecerán en la memoria, hasta que decida escribirlos.
Tenga cuidado antes de utilizar la orden de escritura.

Este disco está actualmente en uso - no se aconseja volver a crear particiones.
Se recomienda desmontar todos los sistemas de ficheros y deshacer todas las
particiones de intercambio de este disco.

Orden (m para obtener ayuda): n

Se están usando todas las particiones primarias.
Se añade la partición lógica 6
Primer sector (1432762368-1953525167, valor predeterminado 1432762368):
Último sector, +/-sectores o +/-tamaño{K,M,G,T,P} (1432762368-1953525167, valor predeterminado 1953525167): +2 GB
Último sector, +/-sectores o +/-tamaño{K,M,G,T,P} (1432762368-1953525167, valor predeterminado 1953525167): +2GB

Crea una nueva partición 6 de tipo 'Linux' y de tamaño 1,9 GiB.

Orden (m para obtener ayuda):
```

Figura 6. Detalles de la partición lógica para el nuevo contenedor.

El siguiente paso será crear el contenedor en la partición lógica. Para ello se recurre a la interfaz de comando, y se crea el nuevo contenedor apreciable en la **Figura 7**, con cuatro parámetros fundamentales: el nombre del contenedor **Contenedor_Particion**, la partición lógica **/dev/sdb6**, el punto de montaje bajo el directorio que recomienda KVM para la seguridad del módulo SELinux, e indicando que el contenedor de almacenamiento utilizará un sistema de archivos **fylesystem**. Los detalles serán observables en la **Figura 8**.

```
# virsh pool-define-as Contenedor_Particion fs -- /dev/sdb6 --
"/var/lib/libvirt/images/Contenedor_Particion"
```

```

root@pc1230:~#
error: Error XML: no se encuentra el nombre del dispositivo del grupo de almacenamiento origen
[root@pc1230 ~]# virsh pool-define-as Contenedor_Particion fs /dev/sdb6 /var/lib/libvirt/images/Contenedor_Particion
error: Falló al definir el grupo Contenedor_Particion
error: Error XML: no se encuentra el nombre del dispositivo del grupo de almacenamiento origen

[root@pc1230 ~]# virsh pool-define-as --name Contenedor_Particion --source /dev/sdb6 --target /var/lib/libvirt/images/Contenedor_Particion
error: El comando 'pool-define-as' no tiene soporte para la opción --source
[root@pc1230 ~]# virsh pool-define-as Contenedor_Particion fs - - /dev/sdb6 - /var/lib/libvirt/images/Contenedor_Particion
El grupo Contenedor_Particion ha sido definido

[root@pc1230 ~]# virsh pool-list --all
  Nombre          Estado      Inicio automático
-----
Contenedor_Particion  inactivo  no
default              activo    si
iso                  activo    si
ISOS                activo    si
p3                  activo    si

[root@pc1230 ~]# virsh pool-build Contenedor_Particion
El pool Contenedor_Particion ha sido compilado

[root@pc1230 ~]# virsh pool-autostart Contenedor_Particion
El grupo Contenedor_Particion ha sido marcado como iniciable automáticamente

[root@pc1230 ~]#

```

Figura 7. Creación del nuevo contenedor.

A continuación, se debe compilar el contenedor creado, iniciar lo para que esté en estado operativo o activo, y que se inicie de forma automática con los siguientes comandos:

```

# virsh pool-build Contenedor_Particion
# virsh pool-start Contenedor_Particion
# virsh pool-autostart Contenedor_Particion

```

```

root@pc1230:~#
El grupo Contenedor_Particion ha sido marcado como iniciable automáticamente

[root@pc1230 ~]# virsh pool-info Contenedor_Particion
Nombre:        Contenedor_Particion
UUID:         b082ae0c-0b49-4115-8925-156666aa46e2
Estado:        inactivo
Persistente:   si
Autoinicio:   si

[root@pc1230 ~]# virsh pool-start Contenedor_Particion
Se ha iniciado el grupo Contenedor_Particion

[root@pc1230 ~]# virsh pool-info Contenedor_Particion
Nombre:        Contenedor_Particion
UUID:         b082ae0c-0b49-4115-8925-156666aa46e2
Estado:        ejecutando
Persistente:   si
Autoinicio:   si
Capacidad:    1,80 GiB
Ubicación:    24,00 KiB
Disponible:   1,80 GiB

[root@pc1230 ~]# mount | grep /var/lib/libvirt/images/Contenedor_Particion/
[root@pc1230 ~]# ls -la /var/lib/libvirt/images/Contenedor_Particion/
total 24
drwxr-xr-x. 3 root root  4096 mar 12 13:18 .
drwx--x--x. 4 root root  4096 mar 12 13:50 ..
drwx-----. 2 root root 16384 mar 12 13:18 lost+found
[root@pc1230 ~]#

```

Figura 8. Configuración de la partición lógica para el nuevo contenedor.

En este punto, el siguiente paso será crear un nuevo volumen llamado Vol2_p4 de 1 GB. Desde el virt-manager se crea dicho volumen con las características mencionadas de tipo qcow2 visible en la **Figura 9** y en la **Figura 10** se verifica que dicho volumen pertenece al recién creado contenedor.

Por último, se debe asociar a través de la interfaz gráfica: selecciona la máquina virtual – Editar – Detalles de la conexión – Añadir Nuevo Hardware Virtual – Selecciona Vol2_p4.

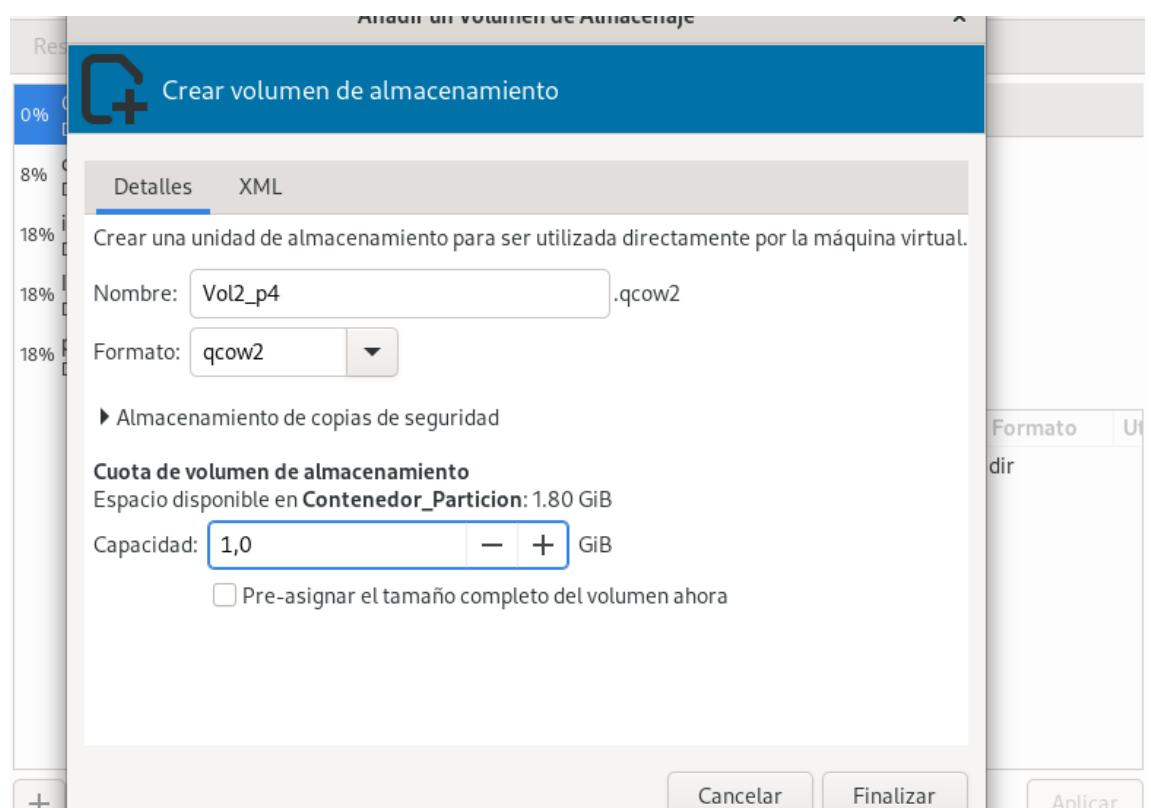


Figura 9. Creación del nuevo volumen.

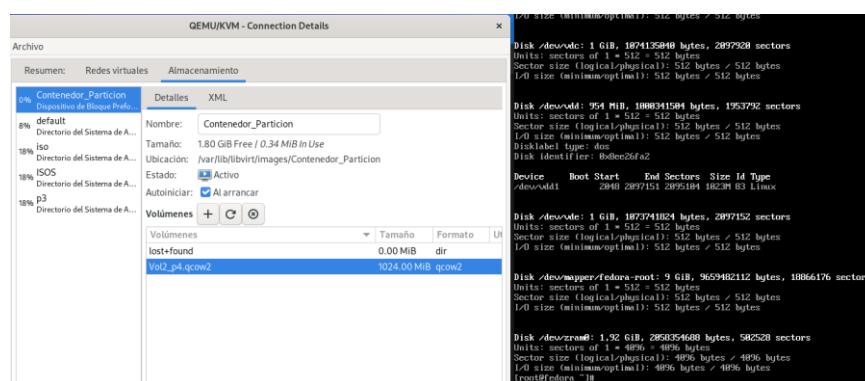


Figura 10. Detalles del nuevo volumen en el contenedor recién creado.

4.2 Almacenamiento en red

4.2.1 Contenedor NFS de imágenes ISO

En este apartado se debe proporcionar un servicio NFS como contenedor en el host anfitrión. Para ello, o desde la utilidad gráfica se creará un nuevo contenedor nfsiso, seleccionando el tipo NETFS, y rellenando los datos de configuración que proporciona [2]. Todo este proceso se puede consultar en la **Figura 11**.

Posteriormente, se puede consultar en la **Figura 12** las imágenes de disco proporcionadas por el contenedor NFS.

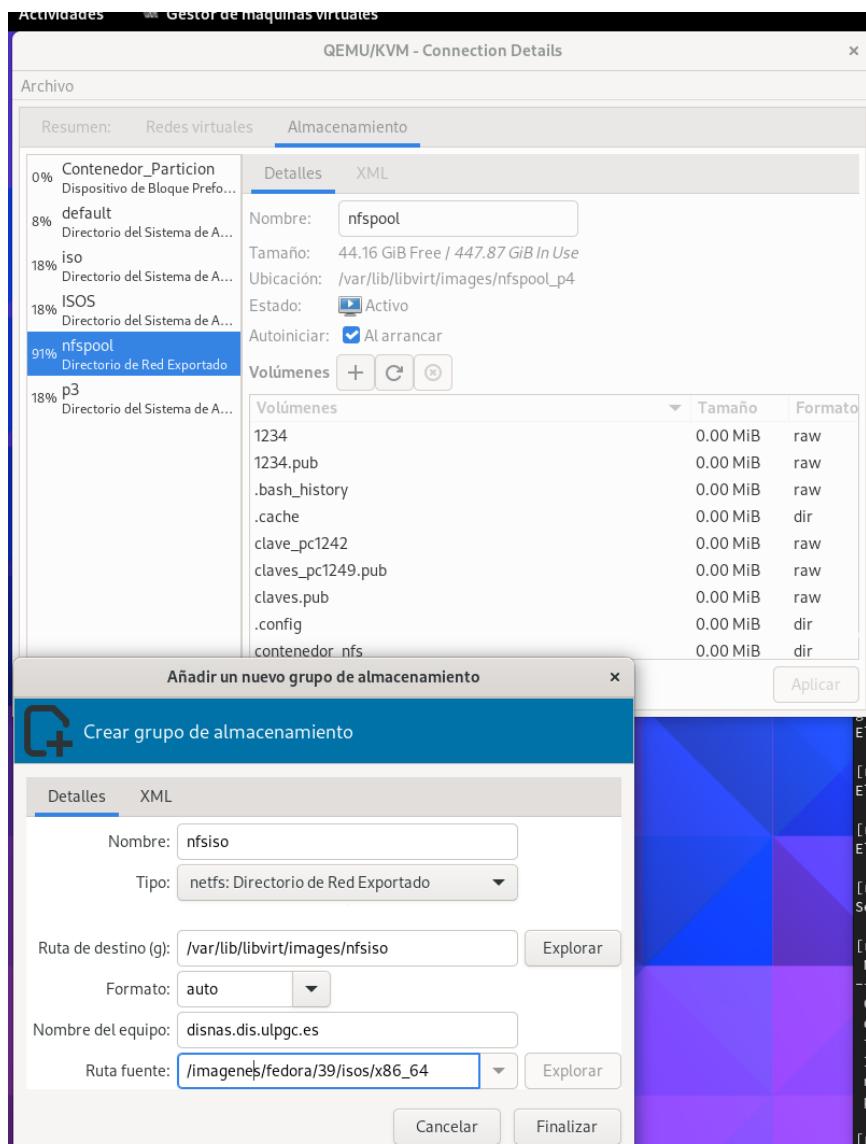


Figura 11. Procedimiento de creación del contenedor para el servicio de imágenes ISO.

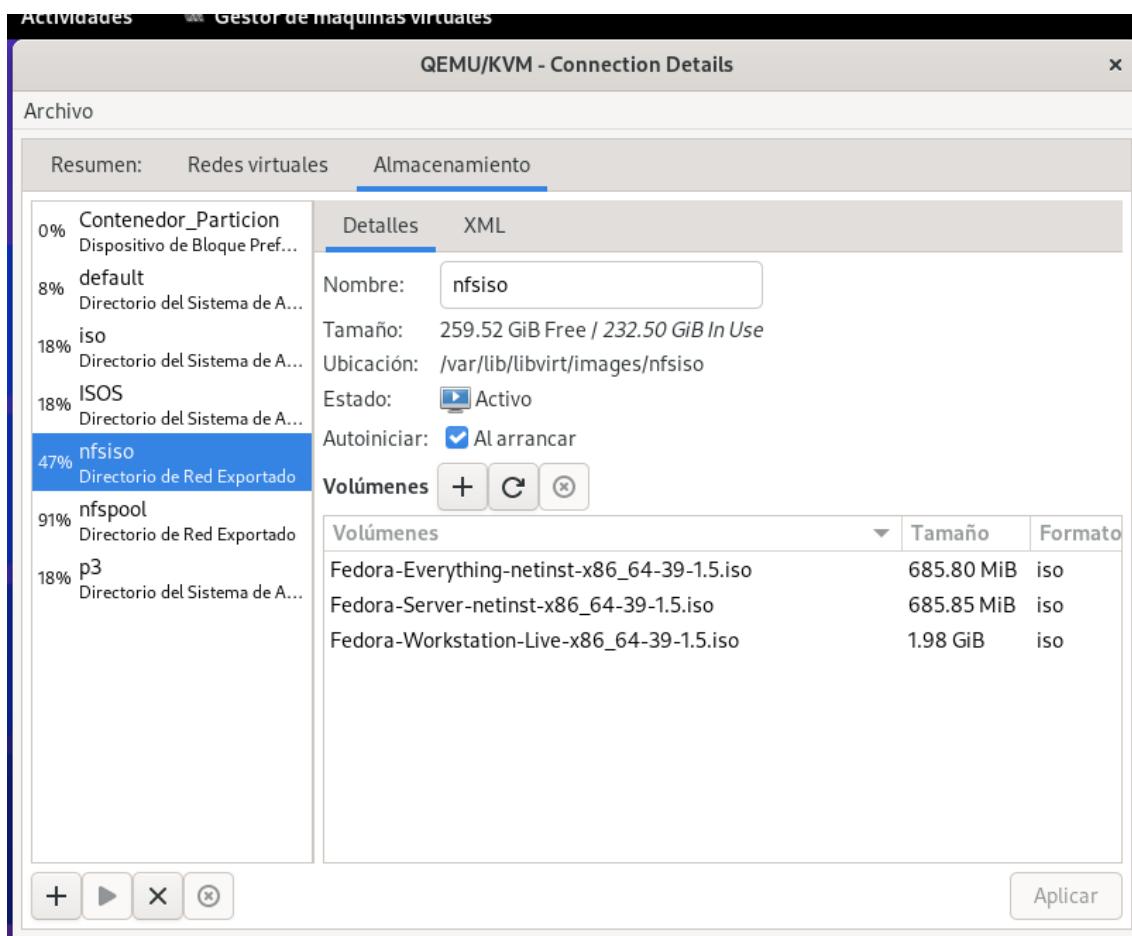
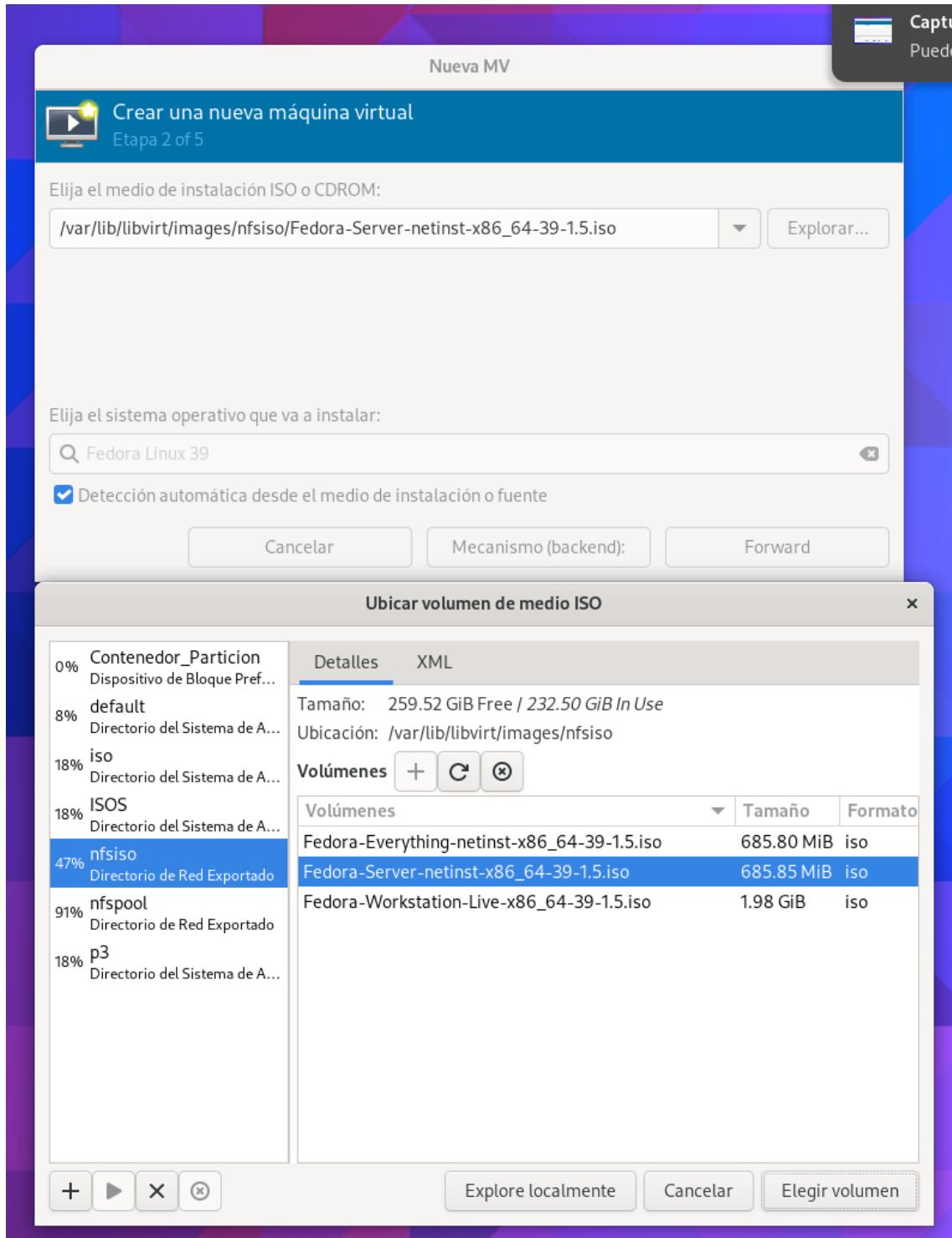
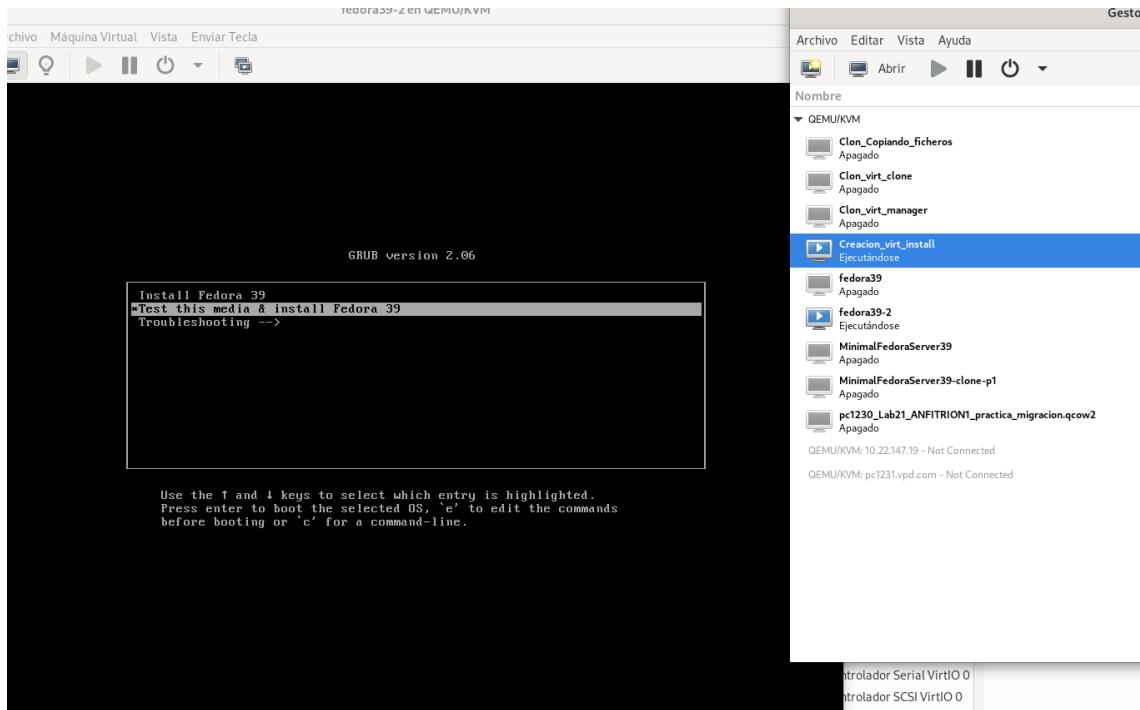


Figura 12. Detalles del nuevo contenedor NFS para imágenes ISO.





4.2.2 Contenedor NFS para volúmenes de máquinas virtuales

Para finalizar la práctica, se deberá añadir un contenedor de tipo NFS al sistema anfitrión dado las configuraciones proporcionadas [2].

De esta manera, se crea el nuevo contenedor con el siguiente comando y se sigue el procedimiento similar de compilación y activación del contenedor como en el **apartado 4.1.3**, tal y como se aprecia en la **Figura 13**:

```
# virsh pool-define-as nfspool netfs --source-host
disnas.dis.ulpgc.es --source-path /disnas-itsi --target
var/lib/libvirt/images/nfspool_p4
```

```
[root@pc1230 ~]# virsh pool-define-as nfspool netfs --source-host disnas.dis.ulpgc.es --source-path /disnas-itsi --tar  
get /var/lib/libvirt/images/nfspool_p4  
El grupo nfspool ha sido definido  
  
[root@pc1230 ~]# virsh pool-build nfspool  
El pool nfspool ha sido compilado  
  
[root@pc1230 ~]# virsh pool-autostart nfspool  
El grupo nfspool ha sido marcado como iniciable automáticamente  
  
[root@pc1230 ~]# virsh pool-start nfspool  
Se ha iniciado el grupo nfspool  
  
[root@pc1230 ~]#
```

```
[root@pc1230 ~]# virsh pool-define-as nfspool netfs --source-host disnas.dis.ulpgc.es --source-path /disnas-itsi --tar  
get /var/lib/libvirt/images/nfspool_p4  
El grupo nfspool ha sido definido  
  
[root@pc1230 ~]# virsh pool-build nfspool  
El pool nfspool ha sido compilado  
  
[root@pc1230 ~]# virsh pool-autostart nfspool  
El grupo nfspool ha sido marcado como iniciable automáticamente  
  
[root@pc1230 ~]# virsh pool-start nfspool  
Se ha iniciado el grupo nfspool  
  
[root@pc1230 ~]# virsh pool-list --all  
Nombre Estado Inicio automático  
-----  
Contenedor_Particion activo sí  
default activo sí  
iso activo sí  
ISOS activo sí  
nfspool activo sí  
p3 activo sí  
  
[root@pc1230 ~]#
```

Figura 13. Creación y detalles del nuevo contenedor NFS.

En la **Figura 15** se puede observar el contenedor recién creado apuntando al sistema fuente del contenedor, y la creación del nuevo volumen creado que define a partir de la utilidad gráfica en la **Figura 14**.

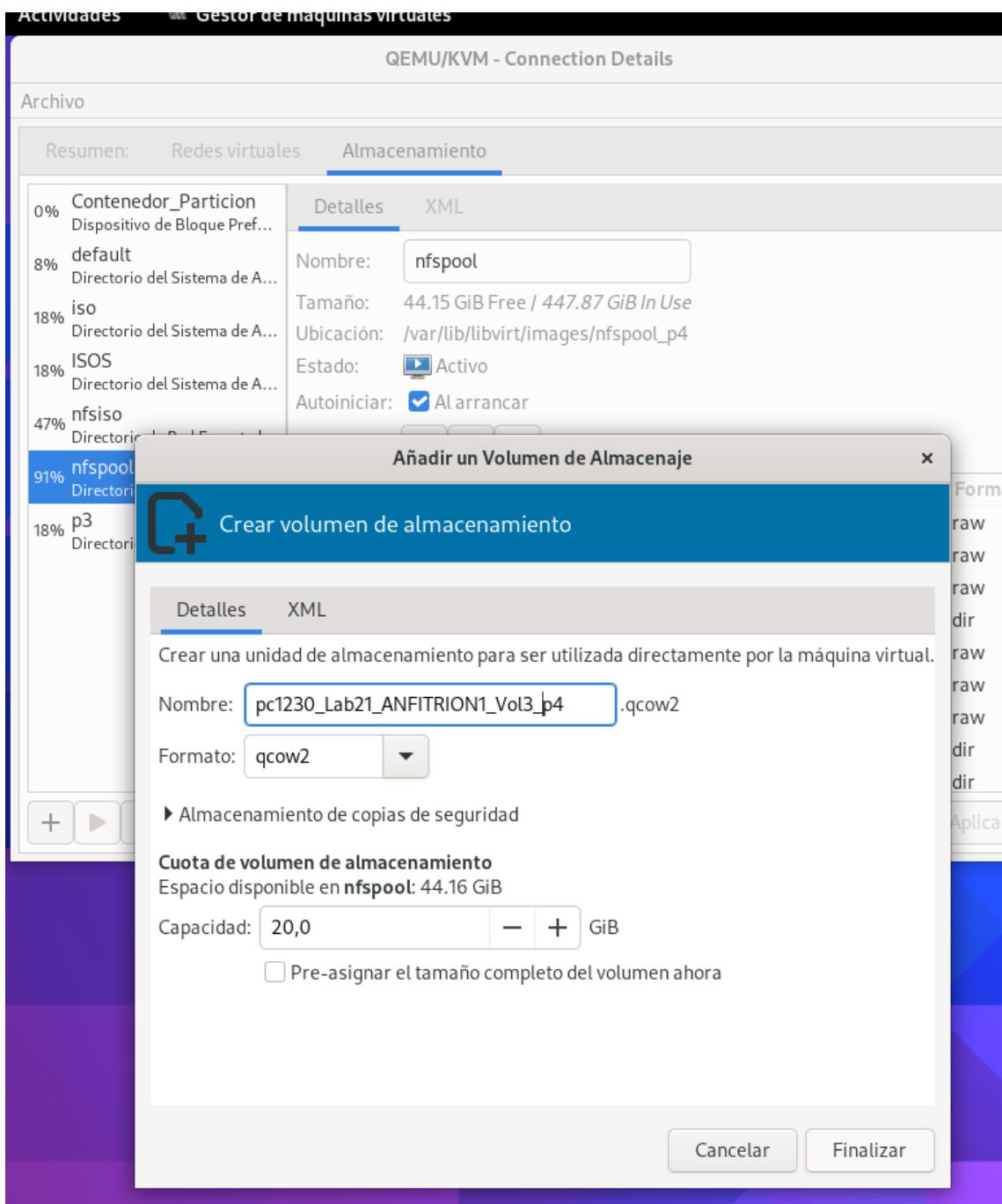


Figura 14. Creación del nuevo volumen en el contenedor NFS.

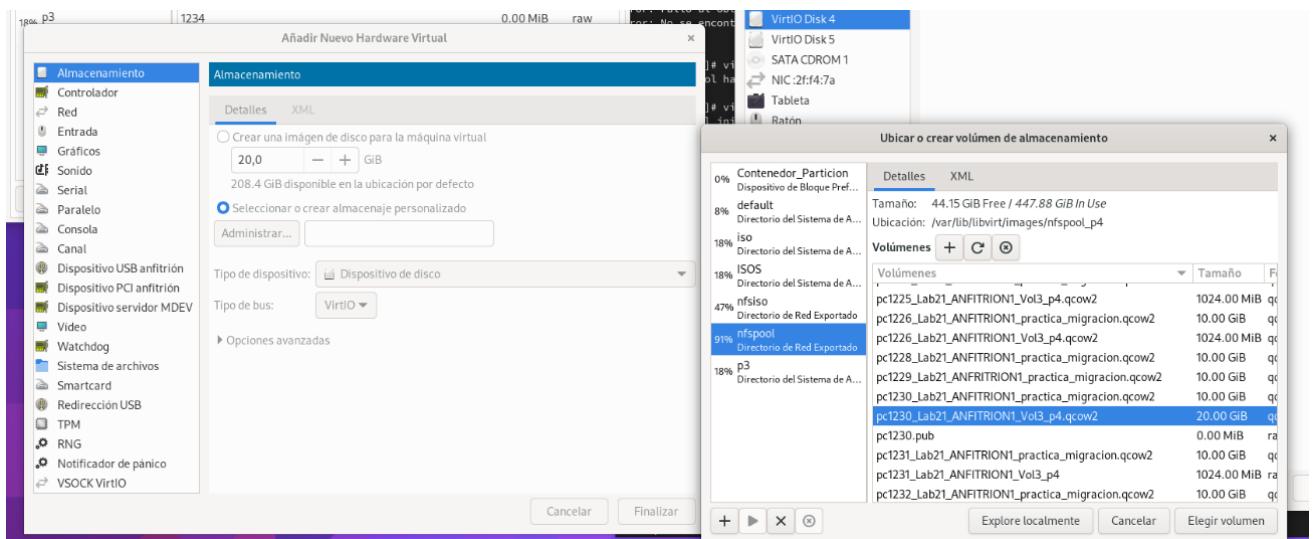


Figura 15. Detalles del contenido del contenedor NFS.

5. Pruebas / Validación

5.1 Almacenamiento de datos desde las máquinas virtuales

Se deberá crear algún fichero de prueba en cada nuevo volumen para verificar su uso. El procedimiento es el mismo para cada volumen, en la **Figura 16** se muestra el procedimiento que se llevó a cabo para el volumen agregado en el **apartado 4.2**:

Una vez adjuntado el volumen a la máquina virtual se crea una partición primaria sobre ese volumen:

```
# fdisk /dev/$volumen/
```

Después se formatea la partición como sistema de fichero ext4 [3]:

```
# mkfs.ext4 /dev/$particion
```

Posteriormente, se monta sobre un directorio temporal para crear el fichero de prueba [4]:

```
# mount -t ext4 /dev/$particion /$dir_tmp
```

```
Welcome to fdisk (util-linux 2.39.3).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS (MBR) disklabel with disk identifier 0x91cdd497.

Command (m for help): n
Partition type
  p   primary (0 primary, 0 extended, 4 free)
      e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-41943039, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-41943039, default 41943039):

Created a new partition 1 of type 'Linux' and of size 20 GiB.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

[root@fedora ~]# mkfs.ext4 /dev/vdf1
mke2fs 1.47.0 (5-Feb-2023)
Descartando los bloques del dispositivo: hecho
Se está creando un sistema de ficheros con 5242624 bloques de 4k y 1318720 nodos-i
UUID del sistema de ficheros: baa15226-4dab-4c61-a142-7d5ae599a28a
Respaldos del superbloque guardados en los bloques:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000

Reservando las tablas de grupo: hecho
Escribiendo las tablas de nodos-i: hecho
Creando el fichero de transacciones (32768 bloques): hecho
Escribiendo superbloques y la información contable del sistema de ficheros: hecho

[root@fedora ~]# mount -t ext4 /dev/vdf1 mnt/
[root@fedora ~]# ls mnt/
lost+found
[root@fedora ~]# touch mnt/nfs_pool
[root@fedora ~]# ls mnt/
lost+found_nfs_pool
```

Figura 16. Procedimiento común para preparar el almacenamiento en cada volumen del sistema invitado.

```
[root@fedora ~]# mount -t ext4 /dev/vdc1 mnt/
[root@fedora ~]# touch mnt/vdc_vol
[root@fedora ~]# mount -t ext4 /dev/vdb1 mnt/
[root@fedora ~]# touch mnt/vdb_vol
[root@fedora ~]# ls mnt/
lost+found vdb_vol
[root@fedora ~]# mount -t ext4 /dev/vdc1 mnt/
[root@fedora ~]# ls mnt/
lost+found vdc_vol
[root@fedora ~]# mount -t ext4 /dev/vdb1 mnt/
[root@fedora ~]# ls mnt/
lost+found vdb_vol
[root@fedora ~]# mount -t ext4 /dev/vde1 mnt/
[root@fedora ~]# ls mnt/
lost+found something
[root@fedora ~]# mount -t ext4 /dev/vdd1 mnt/
[root@fedora ~]# ls mnt/
anything lost+found
[root@fedora ~]#
```

Figura 17. Operaciones que verifican el uso para cada volumen creado.

En la **Figura 17** se puede observar dichas operaciones. Se monta cada partición sobre un directorio temporal `mnt/` y para cada volumen se aprecia un archivo con un nombre distinto, se verificando así correctamente su uso..

Sin embargo, en cuanto a este punto de verificación se hace necesario la modificación del fichero modificar el archivo `/etc/fstab` para que los volúmenes asociados a la máquina virtual se adjunten a los directorios que se especifiquen dicho archivo en cada arranque del sistema. Y, además, evidentemente cada montaje debe ser en un directorio diferente y **no** el mismo directorio como en el que se procedió.

De esta manera, en la **Figura 18** se puede apreciar el estado final de los puntos de montaje de cada partición:

```
/etc/fstab
Created by anaconda on Tue Jan 30 14:03:52 2024

Accessible filesystems, by reference, are maintained under '/dev/disk/'.
See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.

After editing this file, run 'systemctl daemon-reload' to update systemd
units generated from this file.

UUID=401e4b07-8628-48e7-be5f-d5a9c9b24c0b /           xfs    defaults      0 0
UUID=8f134cc2-df49-4468-9f8a-e7205e4b94c3 /boot       xfs    defaults      0 0
dev/vdb1 /root/1 ext4 defaults 0 0
dev/vdc1 /root/2 ext4 defaults 0 0
dev/vdd1 /root/3 ext4 defaults 0 0
dev/vde1 /root/4 ext4 defaults 0 0
dev/vdf1 /root/5 ext4 defaults 0 0
```

Figura 18. Contenido del archivo de configuración de montaje de volúmenes y particiones.

Y en la **Figura 19** se observa la ruta de origen que pertenece a disco virtual asociado a la máquina virtual:

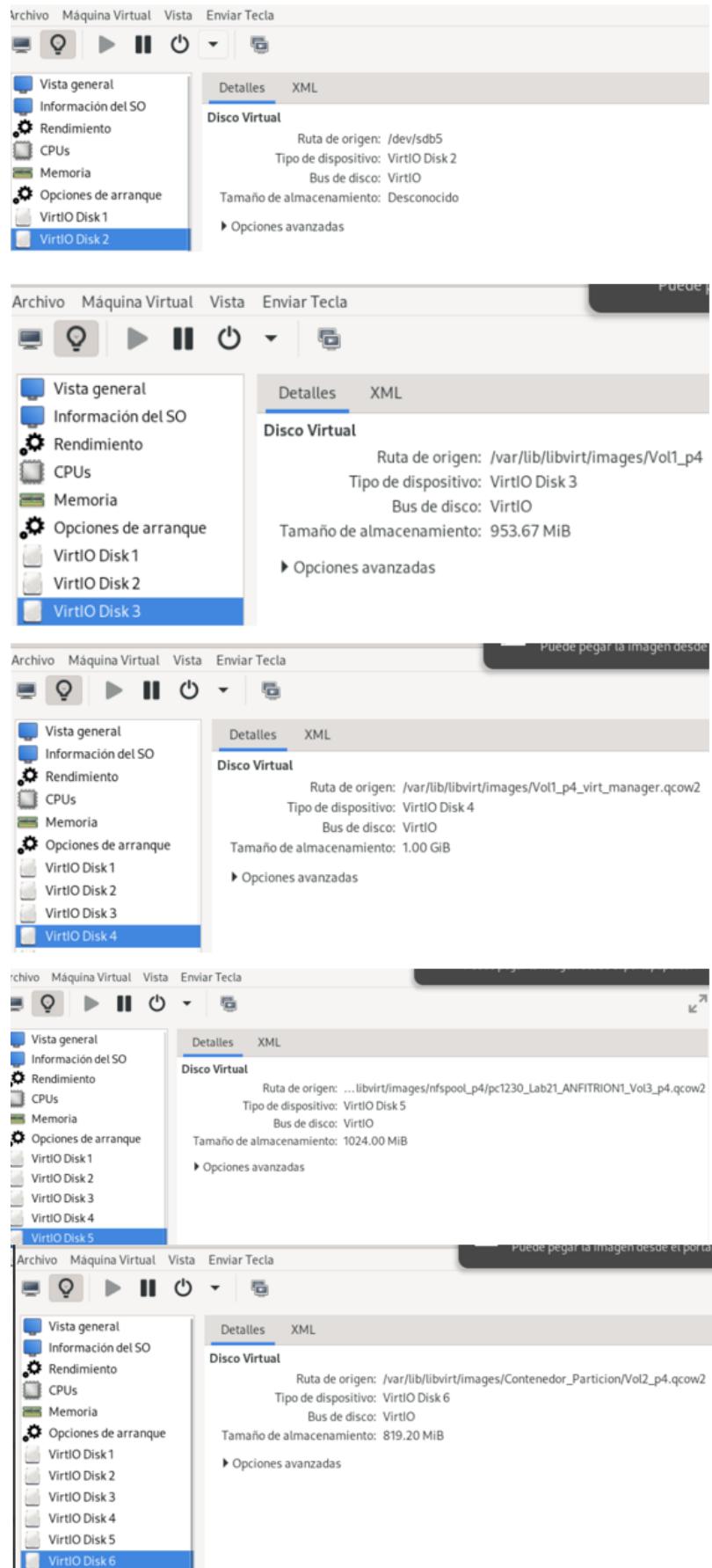


Figura 19. Configuración de discos virtuales de la máquina virtual.

5.2 Contenedor NFS para volúmenes de máquinas virtuales

Para comprobar que el servicio NFS esté disponible, se procederá a un intento de instalación de una máquina virtual utilizando una imagen ISO de dicho servicio.

De esta manera, al crear una máquina virtual en la interfaz gráfica, en primer lugar, como se aprecia en la **Figura 20**, se selecciona el medio de instalación que será a través de una imagen ISO, y, en segundo lugar, como se aprecia en la **Figura 21**, se selecciona del contenedor `nfsiso` creado, una imagen ISO de Fedora 39 de instalación mínima.

Tal y como se observa en la **Figura 22**, se puede instalar el sistema operativo.

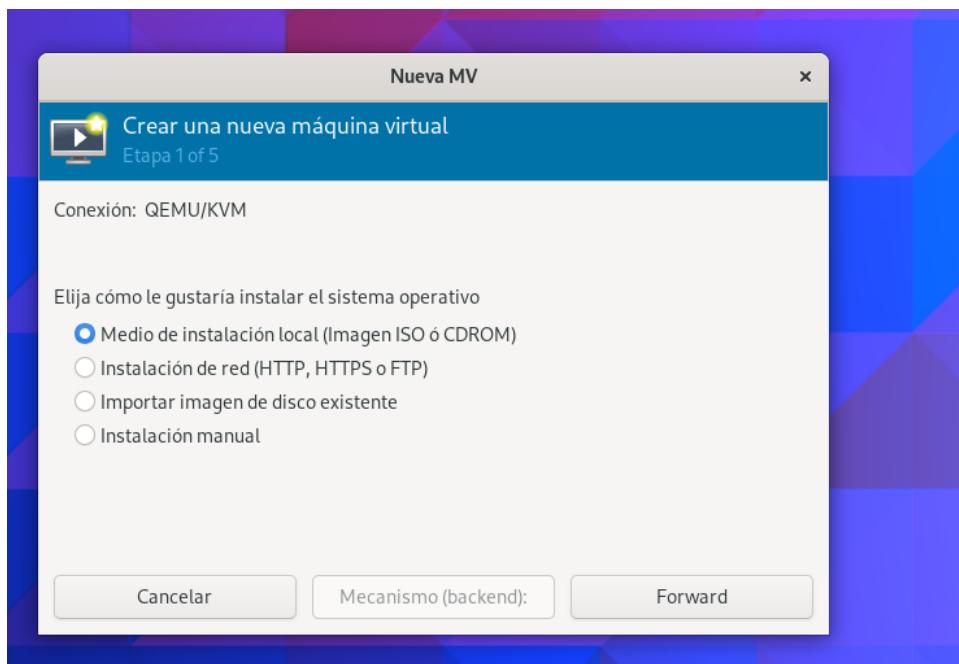


Figura 20. Método de instalación de la máquina virtual.

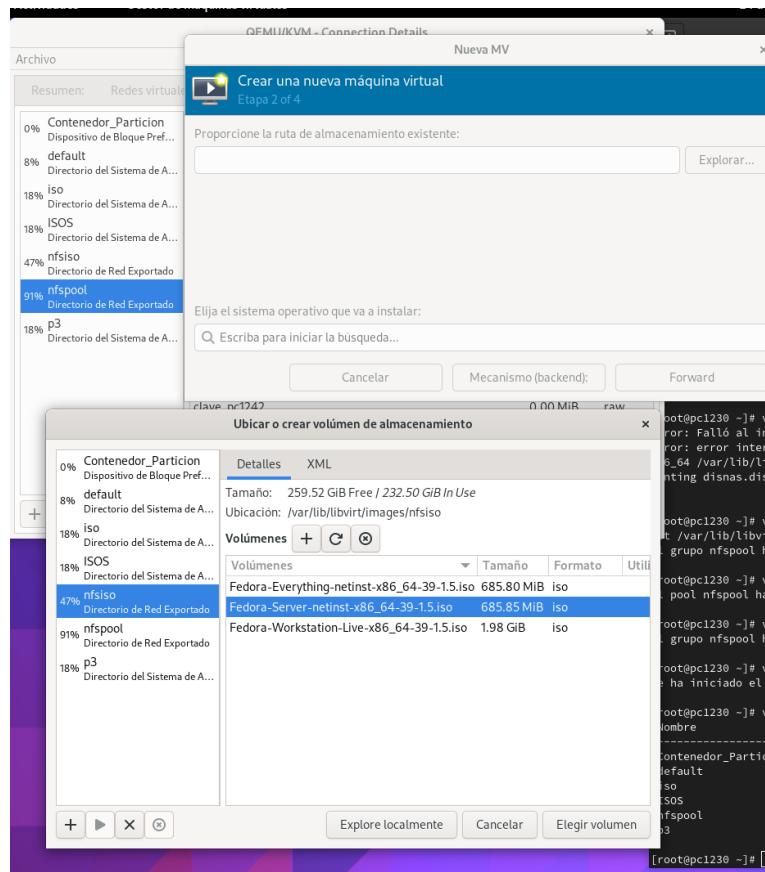


Figura 21. Detalles del contendor ISO al momento de seleccionar una imagen ISO en la creación de una máquina virtual.

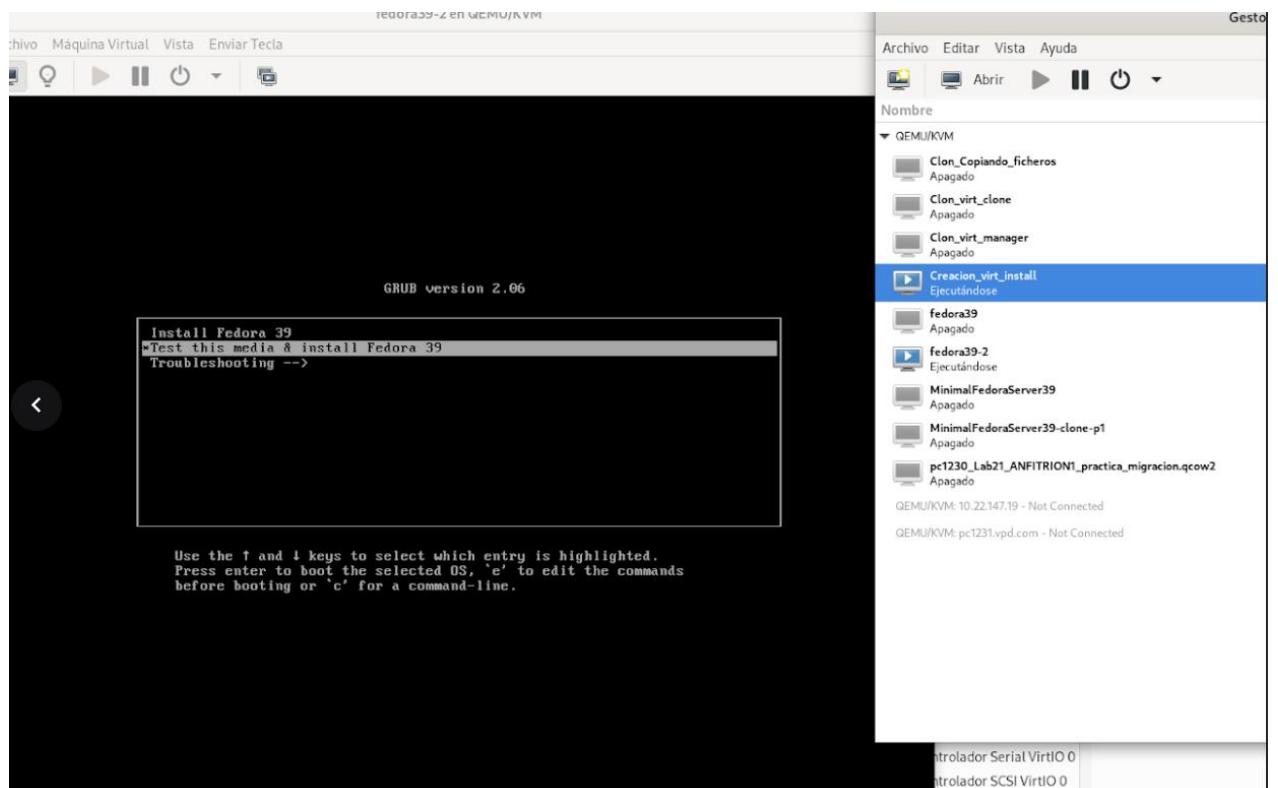


Figura 22. Instalación de la máquina virtual.

6. Fuentes de información

Referencias

- [1] Gabriel, “Aregar dispositivos de almacenamiento a Guests,” DevTechs, [Online]. Available:
<https://devtechs.readthedocs.io/en/latest/topics/virtualization/storage-virtualization/add-storage-guests.html>.
- [2] A. Q. R. García, “Práctica 4: Recursos de almacenamiento virtual,” Universidad de Las Palmas de Gran Canaria, 2024.
- [3] A. Cervera, “Cómo Formatear Linux y Borrar Disco Linux usando comandos?,” Wondershare, 15 Diciembre 2023 . [Online]. Available:
<https://recoverit.wondershare.es/harddrive-tips/format-and-wipe-linux-disk.html>.
- [4] D. Córdoba, “Mount: montando sistemas de archivos,” Junco TIC, 2 julio 2016. [Online]. Available: <https://juncotic.com/mount-montando-sistemas-de-archivos/>.

Práctica 5: Recursos de almacenamiento virtual

Óscar Alexander Martín Tacoronte

Grupo 01.04 (Cod. 44)

15 de abril de 2024

1. Índice

1.	Índice	2
2.	Introducción	3
3.	Objetivos	4
3.1	Configuración de red	4
4.	Desarrollo.....	5
4.1	Configuración de red	5
5.	Pruebas / Validación.....	12
5.1	Verificación de conectividad de la red NAT, la red aislada, y conexión a más de una red	12
5.2	Verificación del puente de red	15
6.	Fuentes de información	17
	Referencias	17

2. Introducción

En esta práctica se configurará distintos tipos de red en un entorno virtualizado. Para ello se hará uso del recurso de *libvirt* con el que se configurará distintos modos del switch virtual. Se operará en modo “NAT”, en “Red Aislada”, y en modo “Red Bridge”, de tal manera que se asociará el switch virtual a una interfaz física del sistema anfitrión.

3. Objetivos

3.1 Configuración de red

- Creación y asignación a una máquina virtual de una red tipo NAT.
- Creación y asignación a una máquina virtual de una red tipo aislada.
- Creación y asignación a una máquina virtual de un puente de red.
- Conexión de una máquina virtual a más de una red.

4. Desarrollo

4.1 Configuración de red

4.1.1 Creación y asignación a una máquina virtual de una red tipo NAT.

En primer lugar, se creará esta red **NAT** mediante **virt-manager**. Para ello, en detalles de conexión – Redes Virtuales, se accede al dialogo gráfico para crear la red, tal como se observa en la **Figura 1**, donde además se puede consultar la configuración: IP de red, rango de IP disponibles por DHCP, modo NAT, nombre de red, entre otros parámetros.

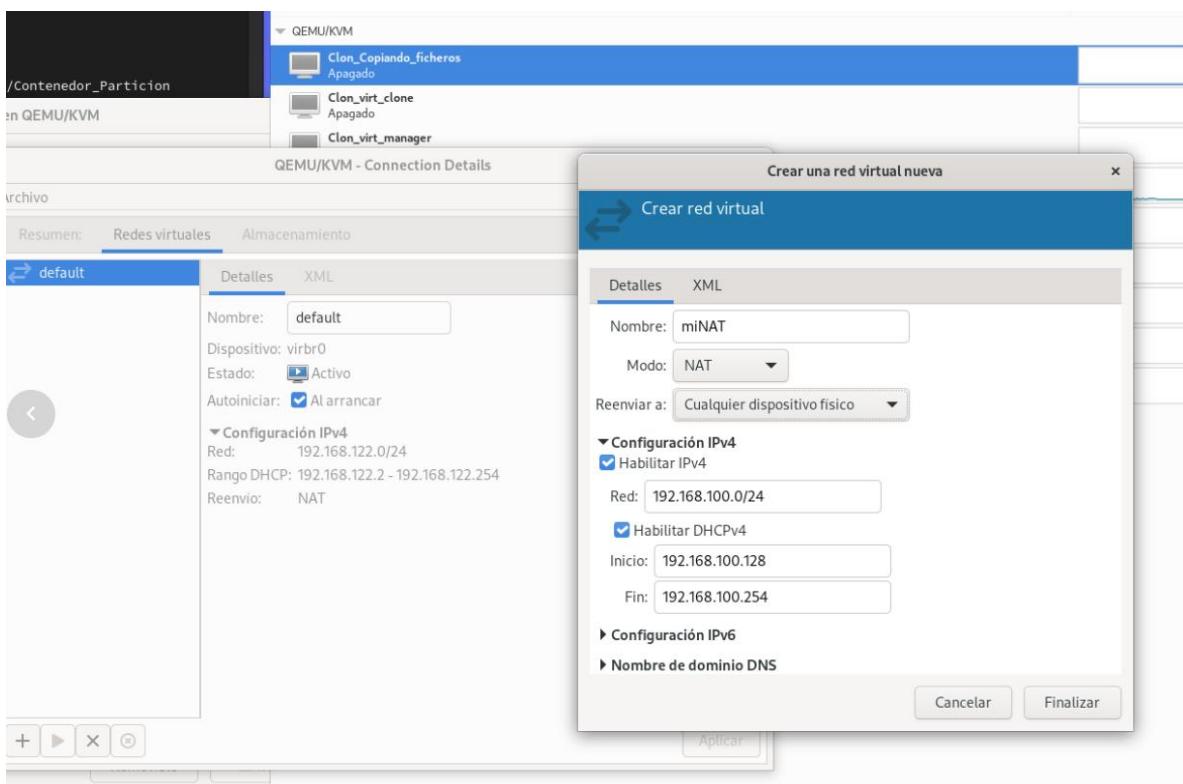


Figura 1. Información detallada sobre la red virtual de tipo NAT creada.

Para la asignación de la red virtual a la maquina virtual utilizando la interfaz gráfica, se abre los detalles de la máquina virtual, y o bien se cambia la fuente de red actual por defecto a la nueva red virtual (**color amarillo** de la **Figura 2**), o bien se accede a “Añadir Hardware” (**color negro** de la **Figura 2**), y se selecciona “Red”, y se añade la fuente de red deseada (**Figura 3**). En cualquier caso, posteriormente se debe aplicar los cambios en tiempo real (**color verde** de la **Figura 2**).

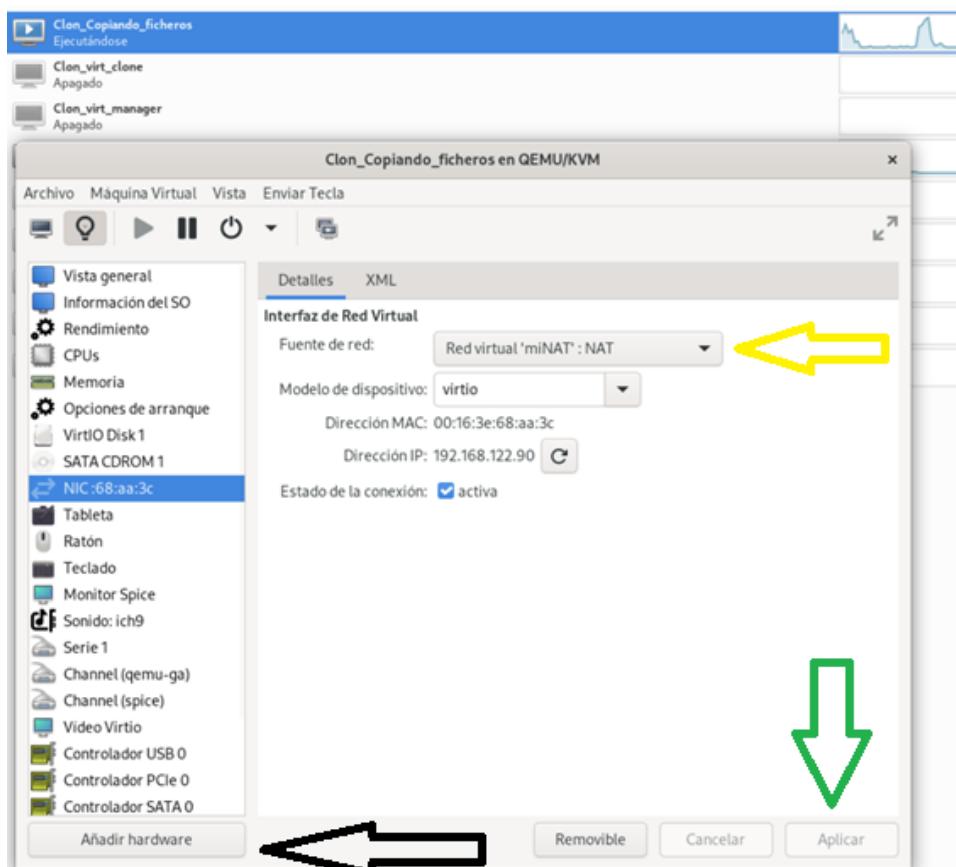


Figura 2. Configuración de la red a una máquina virtual.

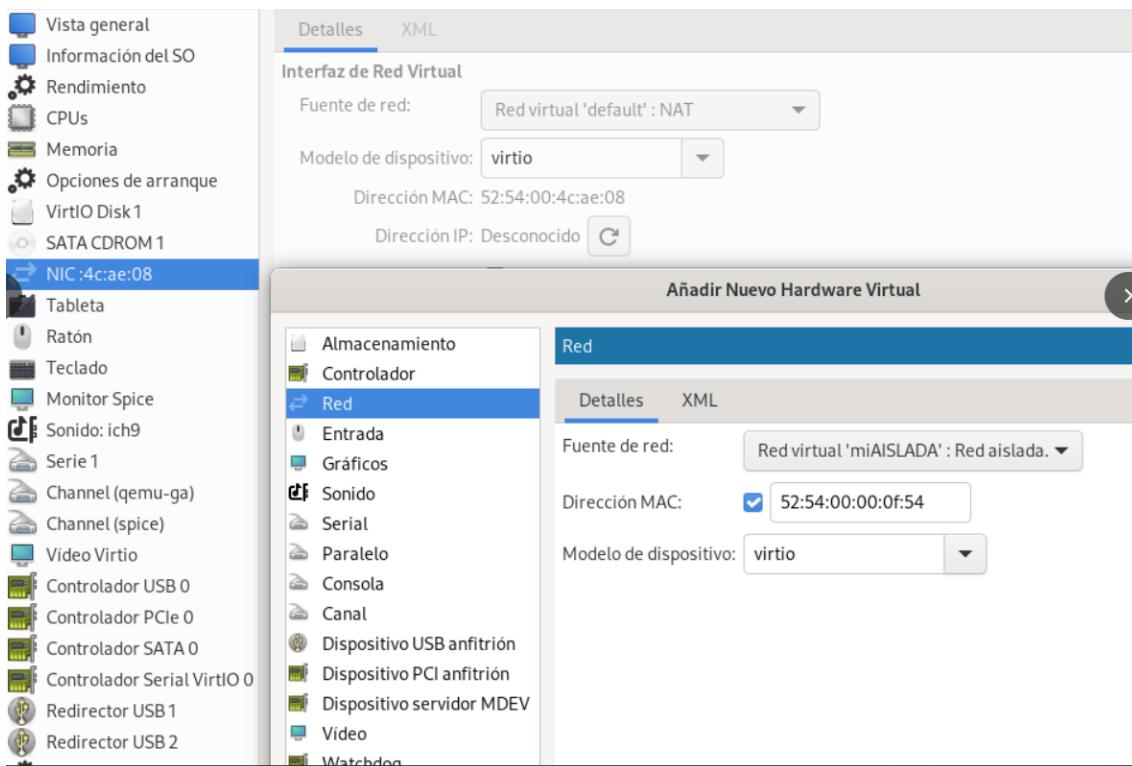


Figura 3. Agregación de una red virtual a una máquina virtual.

También se debe activar la opción de Estado de la conexión en la **Figura 2** para que se active en el sistema anfitrión. Asimismo, también se desactivará en apartados posteriores la red por defecto para asegurarse de la correcta configuración.

Queda por tanto finalizada la creación de la red NAT para las máquinas virtuales, dicha red será 192.168.100.0/24, cuyo servicio DHCP proporcionará direcciones en el rango 192.168.100.128 - 192.168.100.254

4.1.2 Creación y asignación a una máquina virtual de una red tipo aislada.

En segundo lugar, se creará esta red **aislada** mediante **virt-manager**. Para ello, se seguirá el procedimiento seguido en el apartado anterior. En **Detalles de conexión - Redes Virtuales**, se accede al dialogo gráfico para crear la red, tal como se observa en la **Figura 4** donde además se puede consultar la configuración: IP de red, rango de IP disponibles por DHCP, modo NAT, nombre de red, entre otros parámetros.

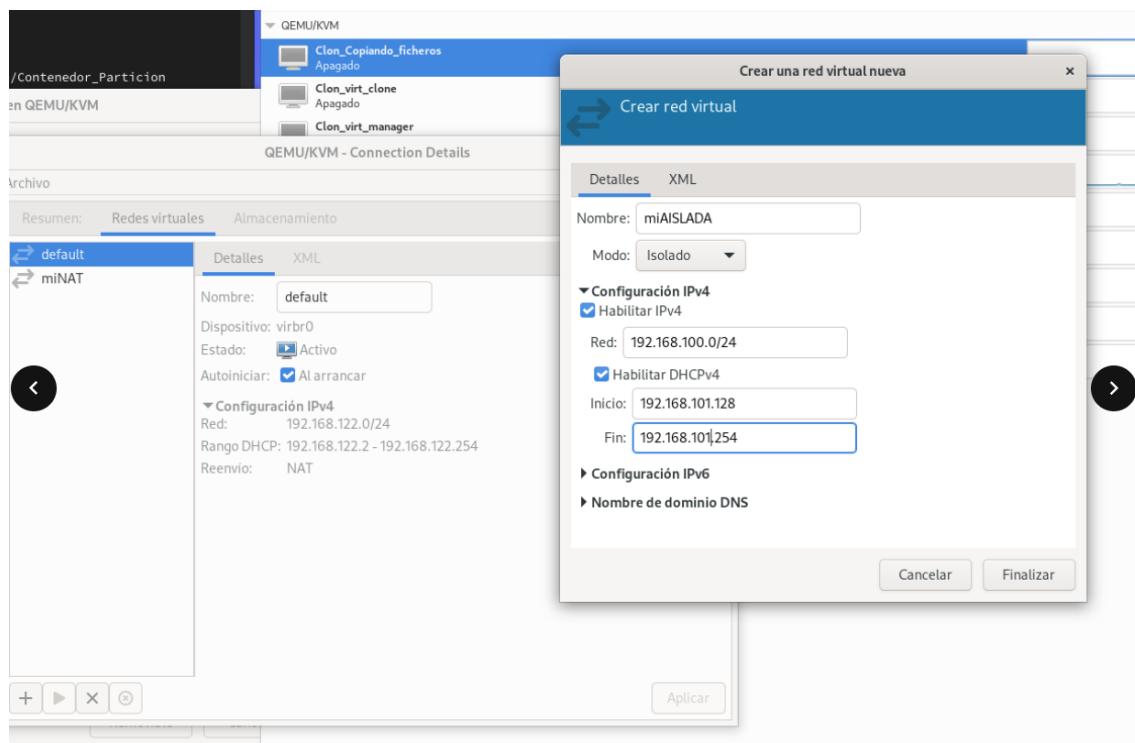


Figura 4. Información detallada sobre la red virtual de tipo aislada creada.

Se sigue el mismo procedimiento de asignación de red virtual al sistema invitado como fue explicado en el apartado anterior.

De esta manera, para las máquinas virtuales que acceda a esta red virtual será la red con IP 192.168.101.0/24, cuyo servicio DHCP proporcionará direcciones en el rango 192.168.101.128 - 192.168.101.254 observable en la **Figura 4**.

4.1.3 Creación y asignación a una máquina virtual de un puente de red.

Para la creación del puente, previamente se decide tomar una instantánea (ver **Figura 5**) de la máquina virtual de prueban para garantizar que la conexión de realice siempre desde la misma MAC, cumpliendo con lo expuesto en [1]. Por lo que si ocurriera un fallo critico por algún motivo inesperado, se pudiera volver al estado inicial de la máquina, conservando la MAC.

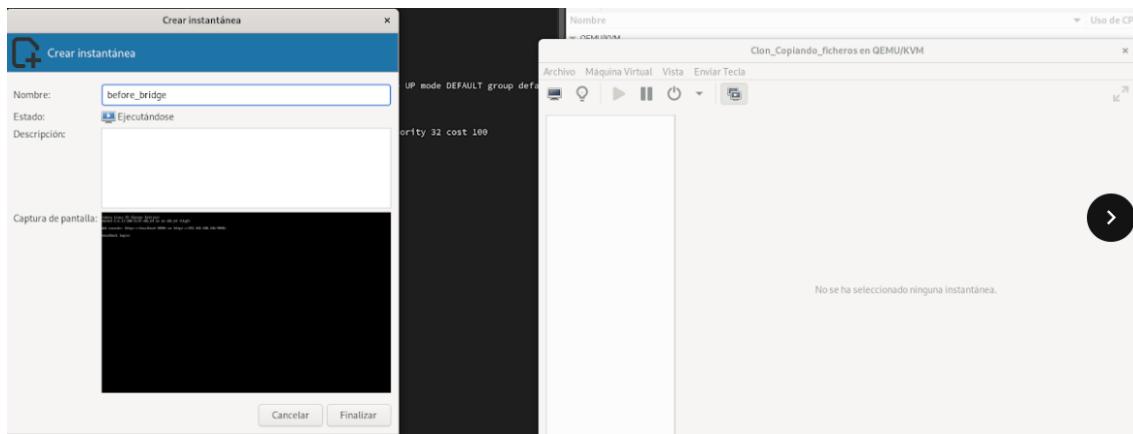


Figura 4. Instantánea de la máquina virtual.

Preparado la instantánea del sistema invitado, desde el sistema anfitrión se configurará el puente haciendo uso de la herramienta nmcli.

De esta manera, en primer lugar, se deberá crear un tipo de conexión de tipo puente con nombre bridge0 siguiendo las instrucciones de [2]:

```
# nmcli connection add type bridge con-name bridge0 ifname bridge0
```

Posteriormente, se debe asignar una interfaz al Puente de red:

```
# nmcli connection add type ethernet slave-type con-name bridge0 ifname  
enp2s0 master bridge0
```

Y, por último, activar la conexión del puente de red:

```
# nmcli connection up bridge0
```

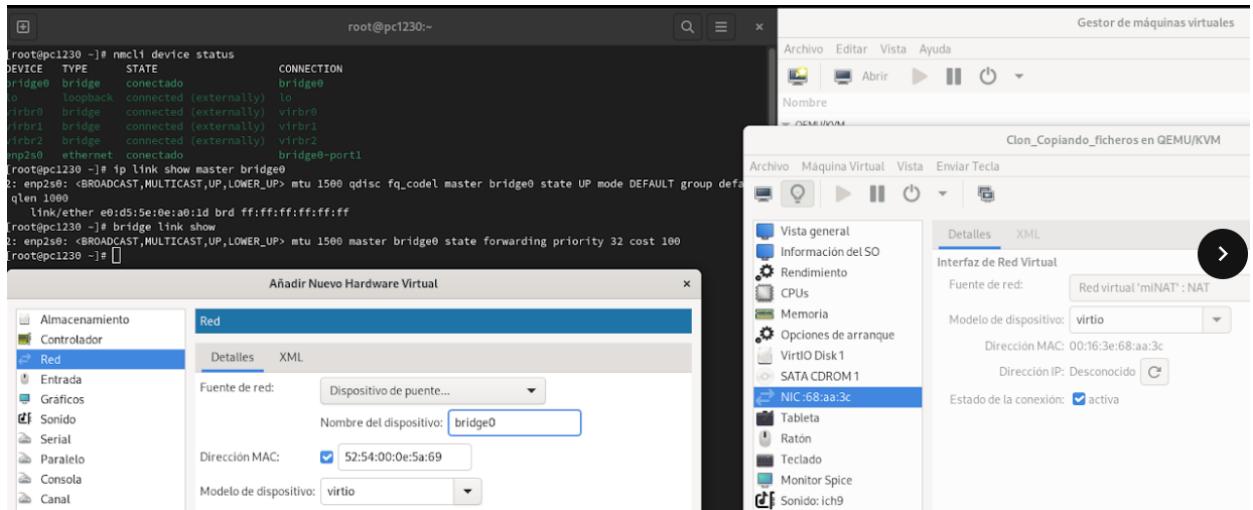


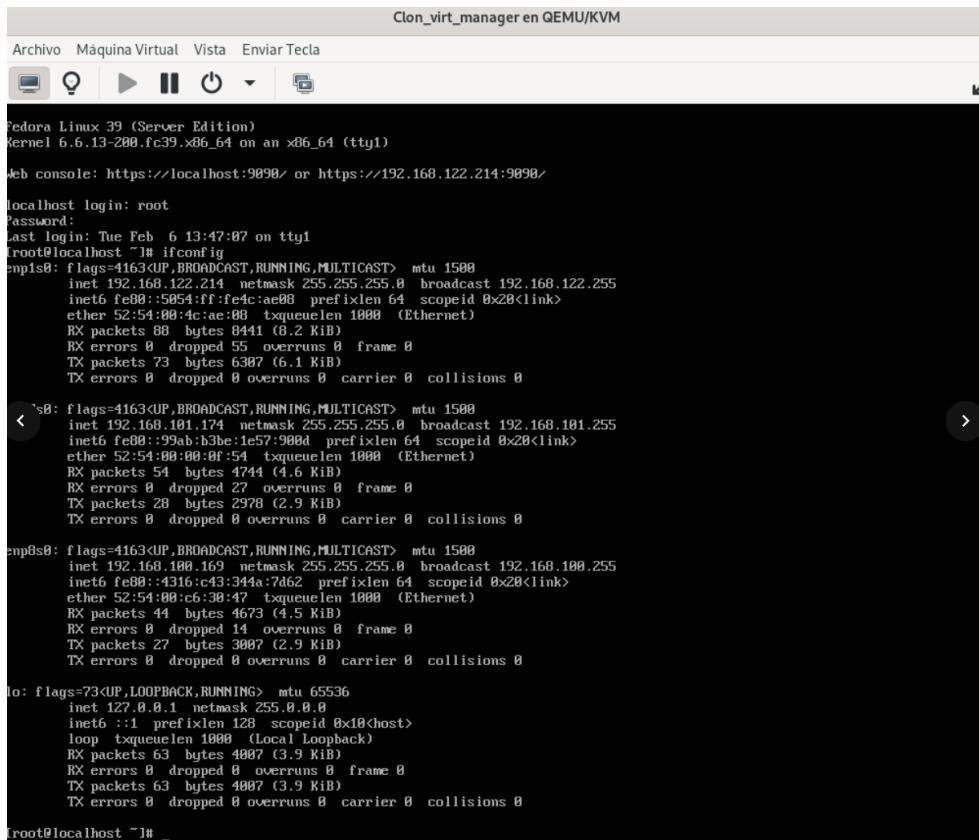
Figura 4. Asignación del puente de red a la máquina virtual.

4.1.4 Conexión de una máquina virtual a más de una red.

Se puede tener, evidentemente, varias interfaces de red al mismo tiempo para una máquina virtual. En este apartado se realizará la configuración de conectividad de cada red virtual creada hasta este momento. La red por defecto de KVM será desactivada. De esta manera, los siguientes sistemas invitados tendrán las siguientes redes, que podrán ser consultadas dicha configuración para cada sistema invitado en la **Figura 5**, **Figura 6** y **Figura 7**.

La verificación de este apartado, y del **apartado 4.1.1** y el **apartado 4.1.2** será explicado en el **apartado 5.1**.

- En la máquina **Clon_virt_manager**, se conectará a la red **miNAT** (**192.168.100.0/24**) y **miAISLADA** (**192.168.101.0/24**).



The screenshot shows a terminal window titled "Clon_virt_manager en QEMU/KVM". The window has tabs for "Archivo", "Máquina Virtual", "Vista", and "Enviar Tecla". Below the tabs are icons for monitor, search, play/pause, power, and close. The terminal displays the following output:

```
Fedora Linux 39 (Server Edition)
Kernel 6.6.13-200.fc39.x86_64 on an x86_64 (tty1)

Web console: https://localhost:9090/ or https://192.168.122.214:9090/

localhost login: root
Password:
Last login: Tue Feb  6 13:47:07 on tty1
root@localhost ~]# ifconfig
enp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.122.214  netmask 255.255.255.0  broadcast 192.168.122.255
                inet6 fe80::5b85:ff:fe4c:ae88  prefixlen 64  scopeid 0x20<link>
                  ether 52:54:00:4c:ae:88  txqueuelen 1000  (Ethernet)
                    RX packets 88 bytes 8441 (8.2 Kib)
                    RX errors 0 dropped 55 overruns 0 frame 0
                    TX packets 73 bytes 6307 (6.1 Kib)
                    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

      ens0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.101.174  netmask 255.255.255.0  broadcast 192.168.101.255
                inet6 fe80::99ab:b3be:1e57:900d  prefixlen 64  scopeid 0x20<link>
                  ether 52:54:00:00:0f:54  txqueuelen 1000  (Ethernet)
                    RX packets 54 bytes 4744 (4.6 Kib)
                    RX errors 0 dropped 27 overruns 0 frame 0
                    TX packets 28 bytes 2978 (2.9 Kib)
                    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

      enp0s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.100.169  netmask 255.255.255.0  broadcast 192.168.100.255
                inet6 fe80::4316:c43:344a:7d62  prefixlen 64  scopeid 0x20<link>
                  ether 52:54:00:c6:38:47  txqueuelen 1000  (Ethernet)
                    RX packets 44 bytes 4673 (4.5 Kib)
                    RX errors 0 dropped 14 overruns 0 frame 0
                    TX packets 27 bytes 3007 (2.9 Kib)
                    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

      lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
                inet6 ::1  prefixlen 128  scopeid 0x10<host>
                  loop  txqueuelen 1000  (Local Loopback)
                    RX packets 63 bytes 4007 (3.9 Kib)
                    RX errors 0 dropped 0 overruns 0 frame 0
                    TX packets 63 bytes 4007 (3.9 Kib)
                    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@localhost ~]#
```

Figura 5. Información detallada sobre la red virtual para la máquina **Clon_virt_manager**.

- En la máquina Clon_virt_clone, se conectará a la red miAISLADA (192.168.101.0/24).

```

Clon_virt_clone en QEMU/KVM
Archivo Máquina Virtual Vista Enviar Tecla
Fedora Linux 39 (Server Edition)
Kernel 6.6.13-200.fc39.x86_64 on an x86_64 (tty1)

Web console: https://localhost:9090/

localhost login: root
Password:
[ root@localhost ~ ]# ifconfig
ens3f0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.101.253 netmask 255.255.255.0 broadcast 192.168.101.255
        inet6 fe80::fe00:101%ens3f0 prefixlen 64 scopeid 0x20<link>
            ether 52:54:00:10:01:00 txqueuelen 1000 (Ethernet)
            RX packets 54 bytes 5393 (5.2 Kib)
            RX errors 0 dropped 19 overruns 0 frame 0
            TX packets 28 bytes 3103 (3.0 Kib)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
            RX packets 62 bytes 3914 (3.8 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 62 bytes 3914 (3.8 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@localhost ~]# ping 8.8.8.8
ping: connect: La red es inaccesible
[root@localhost ~]#

```

Figura 6. Información detallada sobre la red virtual para la máquina Clon_virt_clone.

- En la máquina Clon_Copiando_ficheros, se conectará a la red miNAT (192.168.100.0/24).

```

Clon_Copiando_ficheros en QEMU/KVM
Archivo Máquina Virtual Vista Enviar Tecla
Fedora Linux 39 (Server Edition)
Kernel 6.6.13-200.fc39.x86_64 on an x86_64 (tty1)

Web console: https://localhost:9090/ or https://192.168.100.136:9090/

localhost login: root
Password:
Last login: Tue Mar 19 13:20:09 on tty1
[root@localhost ~]# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=113 time=31.3 ms
^C
--- 8.8.8.8 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 31.342/31.342/31.342/0.000 ms
[root@localhost ~]# ifconfig
ens3f0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.100.136 netmask 255.255.255.0 broadcast 192.168.100.255
        inet6 fe80::fe00:100%ens3f0 prefixlen 64 scopeid 0x20<link>
            ether 00:16:3e:68:aa:3c txqueuelen 1000 (Ethernet)
            RX packets 24 bytes 3531 (3.4 KiB)
            RX errors 0 dropped 9 overruns 0 frame 0
            TX packets 40 bytes 4227 (4.1 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
            RX packets 0 bytes 0 (0.0 B)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 0 bytes 0 (0.0 B)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@localhost ~]#

```

Figura 7. Información detallada sobre la red virtual para la máquina Clon_Copiando_ficheros.

5. Pruebas / Validación

5.1 Verificación de conectividad de la red NAT, la red aislada, y conexión a más de una red

Una vez configurada la infraestructura de red, desde la red **miNAT** debe ser accesible cada equipo entre sí, y, además, deben estar conectados al exterior. En este caso, el equipo **Clon_virt_manager** y **Clon_Copiando_ficheros** deben tener conexión.

Desde la red **miAISLADA** deben estar conectados entre sí, y sin conectividad al exterior. En este caso, el equipo **Clon_virt_manager** y **Clon_virt_clone** deben tener conexión.

De esta manera, en la **Figura 8** se muestra ambas conectividades desde cada red **miNAT** (máquinas **Clon_virt_manager** y **Clon_Copiando_ficheros**) en la red **192.168.100.0/24** cuando envía y recibe las trazas ICMP a la dirección **192.168.100.136** y **miAISLADA** (máquinas **Clon_virt_manager** y **Clon_virt_clone**) en la red **192.168.101.0/24** cuando envía y recibe las trazas ICMP a la dirección **192.168.101.253**.

Clon_virt_manager en QEMU/KVM

Archivo Máquina Virtual Vista Enviar Tecla

```

TX packets 28 bytes 2978 (2.9 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp8s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.100.169 netmask 255.255.255.0 broadcast 192.168.100.255
inet6 fe80::4316:c43:344a:7d62 prefixlen 64 scopeid 0x20<link>
ether 52:54:00:c6:30:47 txqueuelen 1000 (Ethernet)
RX packets 44 bytes 4673 (4.5 KiB)
RX errors 0 dropped 14 overruns 0 frame 0
TX packets 27 bytes 3007 (2.9 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 63 bytes 4007 (3.9 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 63 bytes 4007 (3.9 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@localhost ~]# ping 192.168.1.254
PING 192.168.1.254 (192.168.1.254) 56(84) bytes of data.
^C
--- 192.168.1.254 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2077ms

[root@localhost ~]# ping 192.168.1.253
PING 192.168.1.253 (192.168.1.253) 56(84) bytes of data.
^C
--- 192.168.1.253 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 5112ms

[root@localhost ~]# ping 192.168.101.253
PING 192.168.101.253 (192.168.101.253) 56(84) bytes of data.
64 bytes from 192.168.101.253: icmp_seq=1 ttl=64 time=0.754 ms
64 bytes from 192.168.101.253: icmp_seq=2 ttl=64 time=0.683 ms
^C
--- 192.168.101.253 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.683/0.718/0.754/0.035 ms
[root@localhost ~]# ping 192.168.100.136
PING 192.168.100.136 (192.168.100.136) 56(84) bytes of data.
64 bytes from 192.168.100.136: icmp_seq=1 ttl=64 time=0.885 ms
64 bytes from 192.168.100.136: icmp_seq=2 ttl=64 time=0.895 ms
^C
--- 192.168.100.136 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.885/0.890/0.895/0.045 ms

```

Figura 8. Verificación de conectividad de miNAT:

(máquinas Clon_virt_manager y Clon_Copiando_ficheros) en la red 192.168.100.0/24.

Y verificación de la conectividad de miAISLADA :

(máquinas Clon_virt_manager y Clon_virt_clone) en la red 192.168.101.0/24

Además, la conectividad con el exterior también se verifica en cada red. En la **Figura 9** se muestra que la red miAISLADA no tiene conectividad con el exterior mientras que en la **Figura 10**, la red miNAT sí puede acceder al exterior.

```
File Máquina Virtual Vista Enviar Tecla  
[ ]| [ ]| [ ]| [ ]| [ ]| [ ]|  
Clon_Linux_39 (Server Edition)  
Kernel 6.6.13-200.fc39.x86_64 on an x86_64 (tty1)  
  
console: https://localhost:9090/ or https://192.168.100.136:9090/  
  
localhost login: root  
password:  
t login: Tue Mar 19 13:20:09 on tty1  
root@localhost ~]# ping 8.8.8.8  
64 bytes from 8.8.8.8 (8.8.8.8) 56(84) bytes of data.  
bytes from 8.8.8.8: icmp_seq=1 ttl=113 time=31.3 ms  
  
8.8.8.8 ping statistics ---  
ackets transmitted, 1 received, 0% packet loss, time 0ms  
min/avg/max/mdev = 31.342/31.342/31.342/0.000 ms  
root@localhost ~]# ifconfig  
ens0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
inet 192.168.100.136 netmask 255.255.255.0 broadcast 192.168.100.255  
inet6 fe80::216:3eff:fe68:aa3c prefixlen 64 scopeid 0x20<link>  
ether 00:16:3e:68:aa:3c txqueuelen 1000 (Ethernet)  
RX packets 24 bytes 3531 (3.4 KiB)  
RX errors 0 dropped 9 overruns 0 frame 0  
TX packets 40 bytes 4227 (4.1 KiB)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Figura 9. Clon_Copiando_ficheros intenta salir con éxito desde la red miNAT.

```
Clon_virt_clone en QEMU/KVM  
File Máquina Virtual Vista Enviar Tecla  
[ ]| [ ]| [ ]| [ ]| [ ]| [ ]|  
Clon_virt_clone (Server Edition)  
Kernel 6.6.13-200.fc39.x86_64 on an x86_64 (tty1)  
  
web console: https://localhost:9090/  
  
localhost login: root  
password:  
t login: Tue Feb 6 14:28:47 on tty1  
root@localhost ~]# ifconfig  
ens1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
inet 192.168.101.253 netmask 255.255.255.0 broadcast 192.168.101.255  
inet6 fe80::5054:ff:fe3b:6045 prefixlen 64 scopeid 0x20<link>  
ether 52:54:00:3b:60:45 txqueuelen 1000 (Ethernet)  
RX packets 54 bytes 5393 (5.2 KiB)  
RX errors 0 dropped 19 overruns 0 frame 0  
TX packets 28 bytes 3103 (3.0 KiB)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
inet 127.0.0.1 netmask 255.0.0.0  
inet6 ::1 prefixlen 128 scopeid 0x10<host>  
loop txqueuelen 1000 (Local Loopback)  
RX packets 62 bytes 3914 (3.8 KiB)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 62 bytes 3914 (3.8 KiB)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
root@localhost ~]# ping 8.8.8.8  
ping: connect: La red es inaccesible
```

Figura 10. Clon_virt_clone intenta salir sin éxito desde la red miAISLADA.

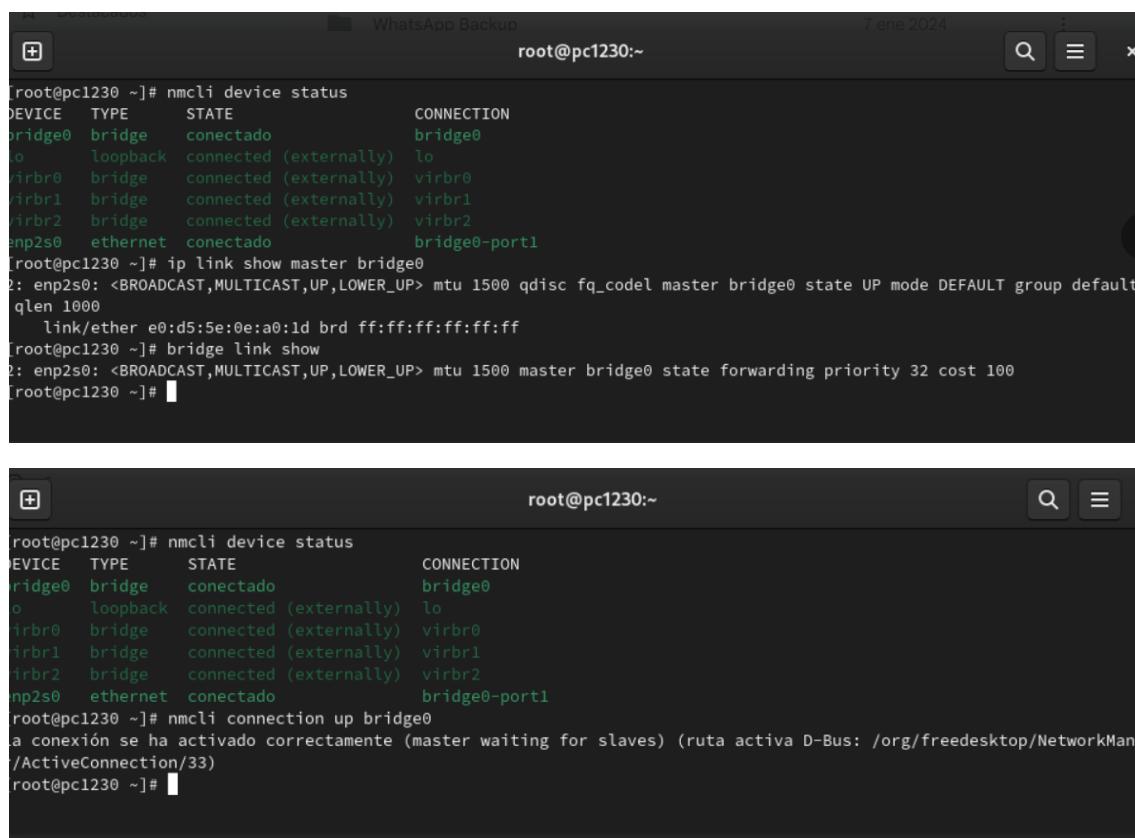
5.2 Verificación del puente de red

Para comprobar que el puente de red está en activo, se puede usar la herramienta de `ip`, o la herramienta de `nmcli`. En cualquier caso, se ejecuta ambas según [2], y se observa en la **Figura 11**, el uso del puente de red. También se observa en la **Figura 12** la asignación de una IP del laboratorio, en concreto la IP “10.22.147.8”, así como la configuración del sistema anfitrión en el puente red “`bridge0`” y su interfaz de red “`enp2s0`” a través de la herramienta `ifconfig`, como del sistema invitado con su IP recibida y su tabla de enrutamiento al exterior.

También se ejecuta trazas de ICMP al exterior y se verifica su conectividad.

Algunos comandos de verificación:

```
# nmcli device status  
# ip link show master bridge0  
# bridge link show  
# ping 8.8.8.8
```



The figure consists of two screenshots of a terminal window. The top screenshot shows the root shell at `root@pc1230:~`. It displays the output of four commands: `nmcli device status`, `ip link show master bridge0`, `bridge link show`, and `ping 8.8.8.8`. The bottom screenshot also shows the root shell at `root@pc1230:~` and displays the output of `nmcli device status` followed by the command `nmcli connection up bridge0`.

Figura 11. Verificación del configuración y activación del puente de red `bridge0`.

```

AME MAJ:MIN RM SIZE RO TYPE MOUNT
da 8:0 0 447,1G 0 disk
-sdal 8:1 0 50M 0 part
-sda2 8:2 0 272,9G 0 part
-sda3 8:3 0 500M 0 part
-sda4 8:4 0 1K 0 part
-sda5 8:5 0 48,8G 0 part
-sda6 8:6 0 7,8G 0 part [SWA]
-sda7 8:7 0 51G 0 part /
-sda8 8:8 0 51G 0 part
db 8:16 0 931,5G 0 disk
-sdb1 8:17 0 227,4G 0 part
-sdb2 8:18 0 227,4G 0 part /var
-sdb3 8:19 0 227,4G 0 part
-sdb4 8:20 0 1K 0 part
-sdb5 8:21 0 954M 0 part
-sdb6 8:22 0 1,9G 0 part
-sdb7 8:23 0 954M 0 part /var
ram0 252:0 0 8G 0 disk [SWA]
root@pc1230 ~]# virsh pool-list
Nombre Estado Inic
-----<
edor_Particion activo si
\ .t activo si
iso activo si
ISOS activo si
nfsiso activo si
nfspool activo si
p3 activo si

root@pc1230 ~]# ifconfig
bridge0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.22.147.18 netmask 255.255.255.0 broadcast 10.22.147.255
        ether e0:d5:5e:0e:a0:1d txqueuelen 1000 (Ethernet)
        RX packets 711452 bytes 7557
        RX errors 0 dropped 0 overr
        TX packets 470435 bytes 9222
        TX errors 0 dropped 0 overru

np2s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.22.147.19 netmask 255.255.255.0 broadcast 10.22.147.255
        ether e0:d5:5e:0e:a0:1e txqueuelen 1000 (Ethernet)
        RX packets 730728 bytes 7679
        RX errors 0 dropped 233 ove
        TX packets 470730 bytes 9223
        TX errors 0 dropped 0 overru

enp7s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.22.147.8 netmask 255.255.255.0 broadcast 10.22.147.255
        ether 00:16:3e:68:aa:3c txqueuelen 1000 (Ethernet)
        RX packets 270 bytes 108895 (10.3 KiB)
        RX errors 0 dropped 204 overruns 0 frame 0
        TX packets 76 bytes 6471 (6.3 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        loop txqueuelen 1000 (Local Loopback)
        RX packets 60 bytes 3784 (3.6 KiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 60 bytes 3784 (3.6 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@localhost ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref  Use Iface
0.0.0.0         10.22.147.1   0.0.0.0         UG   100   0      0 enp7s0
10.22.147.0     0.0.0.0       255.255.255.0  U     100   0      0 enp7s0
[root@localhost ~]#

```

Figura 12. Verificación en el sistema anfitrión y sistema invitado del puente de red bridge0.

6. Fuentes de información

Referencias

- [1] A. Q. R. García, “Práctica 5: Infraestructura de Red Virtual,” Universidad de Las Palmas de Gran Canaria, 2024.
- [2] “Configuring a network bridge [Chapter 6],” Red Hat Enterprise Linux 8, [Online]. Available: https://access.redhat.com/documentation/es-es/red_hat_enterprise_linux/8/html/configuring_and_managing_networking/configuring-a-network-bridge_configuring-and-managing-networking.

Práctica 6: Instalación de un servicio de almacenamiento iSCSI

Óscar Alexander Martín Tacoronte

Grupo 01.04 (Cod. 44)

30 de abril de 2024

1. Índice

1.	Índice	2
2.	Introducción	3
3.	Objetivos	4
3.1	Creación de la infraestructura básica iSCS.....	4
3.2	Exportación del segundo disco SCSI	4
4.	Desarrollo.....	5
4.1	Creación de la infraestructura básica iSCS.....	5
4.2	Creación de la infraestructura básica iSCS.....	15
4.2.1	Exportar el disco sdb vía iSCSI del nodo target	15
4.2.2	Creación de un volumen lógico con sistema de archivo XFS	18
5.	Pruebas / Validación.....	21
5.1	Verificación de exportación de los discos sda.....	21
5.2	Verificación de exportación de los discos sdb.....	22
5.3	Verificación de las redes virtuales configuradas.....	23
6.	Fuentes de información	24
	Referencias	24

2. Introducción

En esta práctica se trata de establecer un espacio de almacenamiento compartido utilizando la tecnología *iSCSI*. Esto implica instalar y configurar un servicio de almacenamiento *iSCSI* y luego, utilizando el servicio de gestión de volúmenes de [1], gestionar un grupo de volumen lógico dentro de ese espacio de almacenamiento.

Se creará, por tanto, una infraestructura virtual donde múltiples nodos pueden acceder y utilizar el mismo almacenamiento de manera compartida.

3. Objetivos

Dado que esta actividad práctica es muy guiada, el informe de la práctica sólo deberá contemplar los siguientes objetivos tal y como se describe en [2]:

3.1 Creación de la infraestructura básica iSCS

- Creación de los recursos de redes y almacenamiento virtuales.
- Conexión de los recursos virtuales a las máquinas virtuales.

3.2 Exportación del segundo disco SCSI

- Exportar el disco sdb vía iSCSI del nodo target.
- Creación de un volumen lógico con sistema de archivo XFS.

4. Desarrollo

4.1 Creación de la infraestructura básica iSCS

4.1.1 Creación de los recursos de redes y almacenamiento virtuales

La creación de redes virtuales se lleva a cabo a través de la interfaz gráfica. Para ello, en **detalles de conexión – Redes Virtuales**, se accede al diálogo gráfico para crear la red, tal como se observa en la **Figura 1**, donde además se puede completar la configuración: IP de red, rango de IP, servicio DHCP, modo de red, entre otras opciones.

Al Habilitar IPv4 en la **Figura 1**, se habilita o deshabilita según se necesite el servicio DHCP, y se introduce el rango de direcciones IP en caso de habilitar DHCP.

Se trata de una tarea similar a las anteriores prácticas, y trivial al tener únicamente que llenar la configuración de red en el diálogo gráfico.

En primer lugar, se creará esta red nombrada como **Almacenamiento** mediante **virt-manager**. Las características de esta red son:

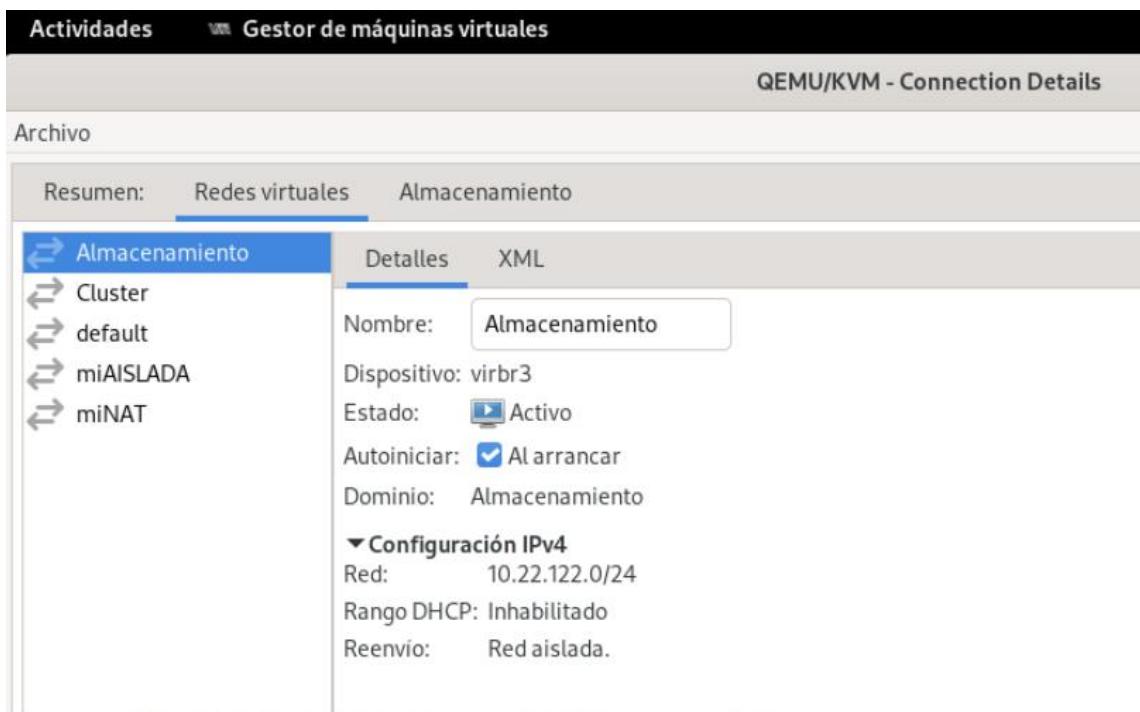
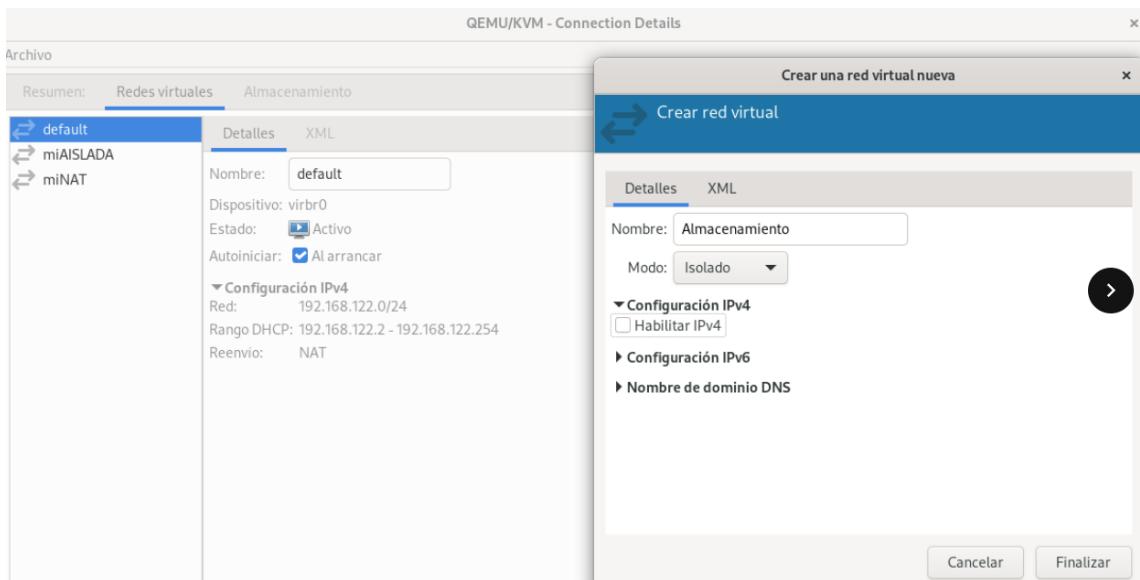
- Tipo aislada.
- Direcciones de la red: **10.22.122.0/24**.
- Servicio DHCP desactivado.

En segundo lugar, se creará esta red nombrada como **Cluster** mediante **virt-manager**. Las características de esta red son:

- Tipo NAT.
- Direcciones de la red: **192.168.140.0/24**.
- Servicio DHCP activado, cuyo rango de direcciones IP son:
192.168.140.2 – 192.168.140.149.

En la red **Almacenamiento** se hace necesario la configuración mientras que en la red **Cluster** no será necesario al tener el servicio DHCP activo.

La configuración final también se puede observar en la **Figura 1**.



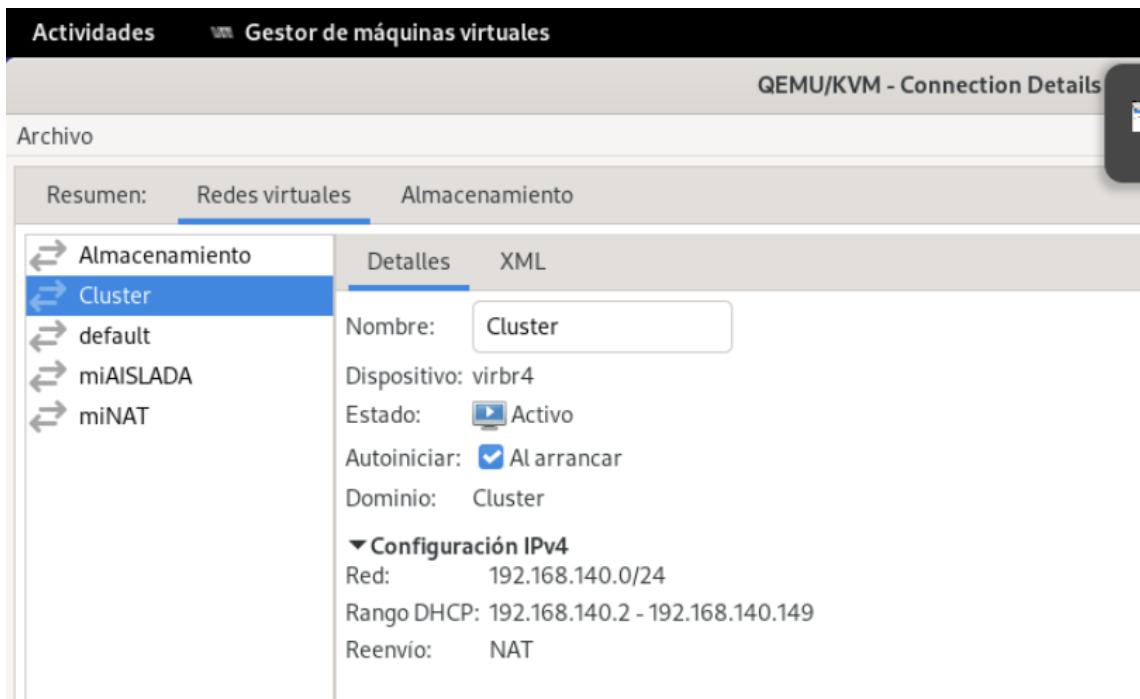


Figura 1. Diálogo en la creación de red virtual Almacenamiento y Cluster creada, y su configuración final.

En cuanto al almacenamiento virtual, se crea a través de `virt-manager` y se debe especificar el **tipo de bus SCSI** como se observa en la **Figura 2**. En cuanto a la configuración del almacenamiento virtual, se debe proceder mediante órdenes `virsh` para poder mapear un volumen de almacenamiento como `sda` de 1 GB y otro volumen de almacenamiento como `sdb` de 1 GB, ambos alojados en el contenedor de almacenamiento por defecto denominado como `default`. Estos volúmenes deben adjuntarse a la máquina virtual `Almacenamiento`, que es el nodo target y exportará cada disco como servicio de almacenamiento virtual de tipo SCSI.

De esta manera, se ejecuta los siguientes comandos para crear los volúmenes:

```
# virsh vol-create-as --pool default --name iscsi_vol_1 --format raw 1 G
# virsh vol-create-as --pool default --name iscsi_vol_1 --format raw 1 G
```

Y posteriormente, se ejecuta los siguientes comandos para adjuntar los volúmenes creados y mapear cada volumen con un identificador de dispositivo específico (`iscsi_vol_1` como `sda` y `iscsi_vol_2` como `sdb`):

```
# virsh attach-disk --config Almacenamiento --source
/var/lib/libvirt/images/iscsi_vol_1 --target sda --live

# virsh attach-disk --config Almacenamiento --source
/var/lib/libvirt/images/iscsi_vol_2 --target sdb --live
```

Ejecutando el siguiente comando se puede consultar los detalles de almacenamiento para una máquina virtual:

```
# virsh domblklist Almacenamiento --details
```

En la **Figura 2**, y en la **Figura 3** se puede observar la configuración de los volúmenes.

The screenshot shows two windows. On the left is the virt-manager interface with a sidebar containing options like Controlador, Red, Entrada, Gráficos, Sonido, Serial, Paralelo, Consola, Canal, Dispositivo USB anfitrión, Dispositivo PCI anfitrión, Dispositivo servidor MDEV, Vídeo, Watchdog, Sistema de archivos, Smartcard, Redirección USB, TPM, RNG, Notificador de pánico, and VSOCK VirtIO. The main window is titled 'Almacenamiento' and shows a 'Detalles' tab selected. It has two radio button options: 'Crear una imagen de disco para la máquina virtual' (disabled) and 'Seleccionar o crear almacenaje personalizado' (selected). A '20,0' value with +/- buttons is shown next to 'GiB'. Below it says '188.0 GiB disponible en la ubicación por defecto'. A 'Administrador...' button is present. A dropdown menu shows 'Tipo de dispositivo: Dispositivo de disco' and 'Tipo de bus: SCSI'. There is also a 'Opciones avanzadas' link. At the bottom are 'Cancelar' and 'Finalizar' buttons. On the right is a terminal window showing command outputs:

```
root@pc1230 ~]# virsh domblklist Almacenamiento --details
  Tipo  Dispositivo  Destino  Fuente
  file  disk        vda      /var/lib/libvirt/images/Almacenamiento.qcow2
-----
[root@pc1230 ~]# virsh attach-disk --config Almacenamiento --source /var/lib/libvirt/images/iscsi_vol_1 --target sda
  El disco ha sido asociado exitosamente
[root@pc1230 ~]# virsh attach-disk --config Almacenamiento --source /var/lib/libvirt/images/iscsi_vol_2 --target sdb
  El disco ha sido asociado exitosamente
```

Figura 2. Por un lado, creación del volumen iSCSI mediante **virt-manager**. Por otro lado, volúmenes adjuntos al nodo target **Almacenamiento**.

Nota: Posteriormente **se hace necesario** adjuntarlo **nuevamente** con la opción **--live**.

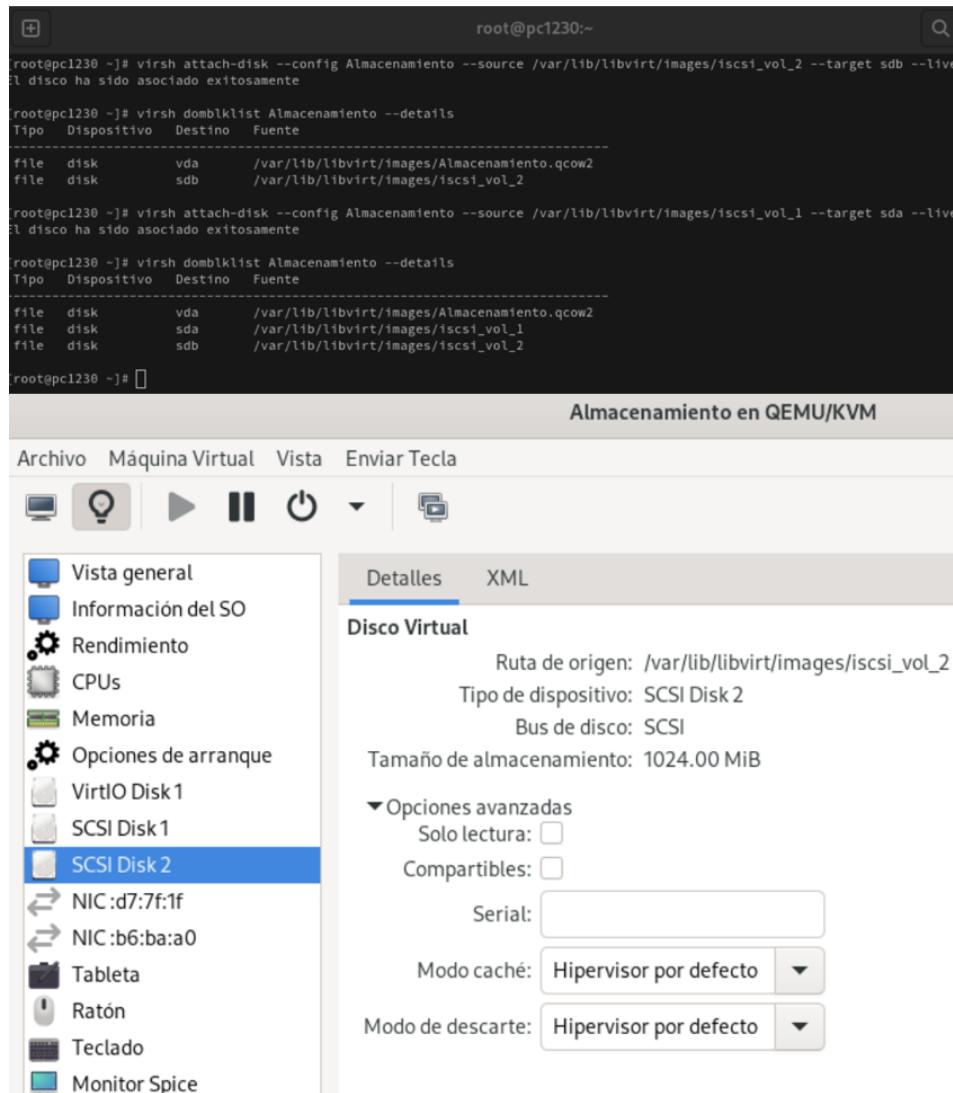


Figura 3. Información detallada sobre los volúmenes creados y su configuración al adjuntarlo al nodo target.

Para terminar con el procedimiento de creación y configuración del almacenamiento virtual, en la **Figura 4** y en la **Figura 5** se puede observar el estado final y el mapeo de cada volumen en la máquina virtual **Almacenamiento** que será el nodo target.

Aunque en la **Figura 4** se observa los volúmenes montados sobre un directorio, en la práctica se desmontó y en el fichero `/etc/fstab` se configuró con la opción `noauto` para que no se montara en cada arranque.

También cada volumen fue formateado como sistema de fichero `ext4` con la orden `mkfs.ext4 $volume` aunque **fue una operación innecesaria** ya que cada nodo initiator lo formateará según sus necesidades, y dicho nodo tiene la responsabilidad de operar con el volumen según su utilidad.

```
[root@localhost ~]# lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda        8:0    0   1G  0 disk /root/iscsi_vol1
sdb        8:16   0   1G  0 disk /root/iscsi_vol2
zram0     251:0   0 1,9G  0 disk [SWAP]
vda       252:0   0 10G  0 disk
└─vda1    252:1   0   1M  0 part
└─vda2    252:2   0   1G  0 part /boot
└─vda3    252:3   0   9G  0 part
  └─fedora-root 253:0   0   9G  0 lvm /
[root@localhost ~]# _
```

Figura 4. Configuración y mapeo de los volúmenes en el nodo target.

```
/etc/fstab
Created by anaconda on Tue Jan 30 14:03:52 2024

Accessible filesystems, by reference, are maintained under '/dev/disk/'.
See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.

After editing this file, run 'systemctl daemon-reload' to update systemd
units generated from this file.

UUID=401e4b87-8628-48e7-be5f-d5a9c9b24c0b /           xfs      defaults      0 0
UUID=8f134cc2-df49-4468-9f8a-e7205e4b94c3 /boot       xfs      defaults      0 0
/dev/sda /root/iscsi_vol1 ext4 noauto 0 0
/dev/sdb /root/iscsi_vol2 ext4 noauto 0 0
```

Figura 5. Fichero /etc/fstab del nodo target.

Para la configuración estática de tipo aislada identificada como *Almacenamiento*, en la red 10.22.122.0/24, se utiliza la herramienta *nmcli* para modificación estática.

En primer lugar, se procede a la asignación de IP a la interfaz de red asociada a la red virtual de *Almacenamiento* y a su puerta de enlace *ipv4.gateway* se especifica como *none* al introducir dobles comillas como parámetro en la segunda instrucción para asegurarse de que no se añade la puerta de enlace a la tabla de enrutamiento. Por último, se debe configurar como asignación estática para que Network Manager no solicite una nueva dirección IP, que, aunque la interfaz de red tenga DHCP desactivado y KVM no proporcionaría una dirección IP nueva, debe especificarse con *nmcli* para que Network Manager no use DHCP [3]:

(Los comandos se realiza en cada nodo, y se debe detectar cada interfaz de red que corresponda a la red virtual aislada para realizar dicha configuración):

```
# ncmlcli connection modify enp1s0 ipv4.addresses 10.22.122.11/24
# ncmlcli connection modify enp1s0 ipv4.gateway ""
# ncmlcli connection modify enp1s0 ipv4.method manual
# ncmlcli connection enp1s0 up
```

En la **Figura 6**, se muestra el procedimiento con una de las máquinas virtuales del clúster por simplicidad. En la **Figura 7**, se muestra más detalles del procedimiento y en la **Figura 8** se observa la conexión de red de las máquinas correctamente configuradas en el clúster.

```
Nodo1 en QEMU/KVM
Archivo Máquina Virtual Vista Enviar Tecla
Fedora Linux 39 (Server Edition)
Kernel 6.6.13-200.fc39.x86_64 on an x86_64 (tty1)
Web console: https://localhost:9090/
localhost login: root
Password:
Last login: Tue Apr  9 14:28:35 on tty1
[root@localhost ~]# ifconfig
enp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.22.122.11 netmask 255.255.255.0 broadcast 10.22.122.255
        inet6 fe80::feee8:b666 prefixlen 64 scopeid 0x20<link>
            ether 52:54:80:e8:b6:66 txqueuelen 1000 (Ethernet)
            RX packets 62 bytes 6844 (6.6 KiB)
            RX errors 0 dropped 17 overruns 0 frame 0
            TX packets 47 bytes 5036 (4.9 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp7s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.140.33 netmask 255.255.255.0 broadcast 192.168.140.255
        inet6 fe80::b17e:75bc:536:b9b9 prefixlen 64 scopeid 0x20<link>
            ether 52:54:80:38:68:b1 txqueuelen 1000 (Ethernet)
            RX packets 26 bytes 1999 (1.9 KiB)
            RX errors 0 dropped 16 overruns 0 frame 0
            TX packets 23 bytes 1958 (1.9 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
            RX packets 0 bytes 0 (0.0 B)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 0 bytes 0 (0.0 B)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@localhost ~]# nmcli connection modify enp1s0 ipv4 10.22.122.11/24 gw4 10.22.122.1_
```

```
[root@localhost ~]# nmcli connection up enp1s0 _
```

Figura 6. Procedimiento común a cada nodo del clúster al configurarlo mediante `nmcli`. Se observa cómo **erróneamente** se introdujo una **puerta de enlace**, posteriormente **fue eliminado**.

```

# virsh attach-disk --config Almacenamiento --source /var/lib/libvirt/images/iscsi_vol_2 --target sdb --live
do asociado exitosamente

Almacenamiento en QEMU/KVM
Nodo0 en QEMU/KVM
Archivo Máquina Virtual Vista Enviar Tecla
Almacenamiento en QEMU/KVM
Nodo2 en QEMU/KVM
Archivo Máquina Virtual Vista Enviar Tecla

[...]

```

Left Terminal (Node0):

```

[root@localhost ~]# route add -net 18.22.122.0 netmask 255.255.255.0 gw 18.22.122.1
SIOCADDRT: La red es inaccesible
[root@localhost ~]# route add -net 18.22.122.0 netmask 255.255.255.0 gw 18.22.122.1
SIOCADDRT: La red es inaccesible
[root@localhost ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref    Use Iface
0.0.0.0          192.168.148.1  0.0.0.0        UG   181      0      0 enp7s0
192.168.148.0   0.0.0.0        255.255.255.0  U     181      0      0 enp7s0
[root@localhost ~]# nmcli connection modify enp1s0 ipv4.gateway 18.22.122.12/24 gw4 18.22.122.1/24
Error: Failed to modify ipv4.gateway: dirección de puerta de enlace <18.22.122.1/24> no válida.
[root@localhost ~]# nmcli connection modify
Conexión: cableada[1] filename           id           path
enp1s0           help             lo           --temporary
[root@localhost ~]# nmcli connection modify enp1s0 ipv4.gateway 18.22.122.1
[root@localhost ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref    Use Iface
0.0.0.0          192.168.148.1  0.0.0.0        UG   181      0      0 enp7s0
0.0.0.0          18.22.122.1    0.0.0.0        UG   182      0      0 enp1s0
18.22.122.0     0.0.0.0        255.255.255.0  U     182      0      0 enp1s0

```

Right Terminal (Node2):

```

RX packets 62 bytes 6844 (6.6 KIB)
RX errors 0 dropped 17 overruns 0 frame 0
TX packets 47 bytes 5836 (4.9 KIB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp7s0: flags=4163UP,BROADCAST,RUNNING,MULTICAST  mtu 1500
inet 192.168.148.33 brd 255.255.255.0 broadcast 192.168.148.255
inet6 fe80::5f54:ff:fe07:deaf%enp7s0 brd fe80::ff:fe07:deaf link
      ether 52:54:00:2f:2c:ed txqueuelen 1000  (Ethernet)
RX packets 44 bytes 4212 (4.1 KIB)
RX errors 0 dropped 16 overruns 0 frame 0
TX packets 26 bytes 1999 (1.9 KIB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
TX packets 23 bytes 1598 (1.5 KIB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73UP,LOOPBACK,RUNNING  mtu 65536
inet 127.0.0.1 brd 127.255.255.255  broadcast 127.0.0.1
      inet6 ::1 brd ::1 link
      ether 00:00:00:00:00:01 txqueuelen 1000  (Local Loopback)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

Iroot@localhost ~]# iw dev connection modify enp1s0
Display all 304 possibilities? (y or n)
Iroot@localhost ~]# iw dev connection modify enp1s0 ipv4.gateway 18.22.122.1
Iroot@localhost ~]# iw dev connection modify enp1s0
^Cerror: nmcli finalizado por la señal Interrupción (2)
Iroot@localhost ~]# ping 18.122.22.11 (18.122.22.11) 56(84) bytes of data.
PING 18.122.22.11 (18.122.22.11) 56(84) bytes of data.
12 packets transmitted, 0 received, 100% packet loss, time 11206ms
Iroot@localhost ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref    Use Iface
0.0.0.0          192.168.148.1  0.0.0.0        UG   181      0      0 enp7s0
0.0.0.0          18.22.122.1    0.0.0.0        UG   182      0      0 enp1s0
18.22.122.0     0.0.0.0        255.255.255.0  U     182      0      0 enp1s0
192.168.148.0   0.0.0.0        255.255.255.0  U     181      0      0 enp7s0
Iroot@localhost ~]# ping 18.22.122.11 (18.22.122.11) 56(84) bytes of data.
64 bytes from 18.22.122.11: icmp_seq=1 ttl=64 time=0.849 ms
64 bytes from 18.22.122.11: icmp_seq=2 ttl=64 time=0.892 ms
^C
--- 18.22.122.11 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1819ms
rtt min/avg/max/mdev = 0.849/0.878/0.892/0.821 ms
Iroot@localhost ~]#

```

Figura 7. Procedimiento más detallado para configurar la puerta de enlace al configurar los nodos mediante nmcli.

```

Almacenamiento en QEMU/KVM
Nodo0 en QEMU/KVM
Archivo Máquina Virtual Vista Enviar Tecla
Almacenamiento en QEMU/KVM
Nodo1 en QEMU/KVM
Archivo Máquina Virtual Vista Enviar Tecla
Almacenamiento en QEMU/KVM
Nodo2 en QEMU/KVM
Archivo Máquina Virtual Vista Enviar Tecla

```

Left Terminal (Node0):

```

[root@localhost ~]# ping 18.22.122.11
PING 18.22.122.11 (18.22.122.11) 56(84) bytes of data.
4 bytes from 18.22.122.11: icmp_seq=1 ttl=64 time=1.16 ms
4 bytes from 18.22.122.11: icmp_seq=2 ttl=64 time=0.749 ms
^C
--- 18.22.122.11 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.749/0.953/1.158/0.284 ms
root@localhost ~]# ping 18.22.122.11
PING 18.22.122.11 (18.22.122.11) 56(84) bytes of data.
4 bytes from 18.22.122.11: icmp_seq=1 ttl=64 time=0.462 ms
4 bytes from 18.22.122.11: icmp_seq=2 ttl=64 time=0.788 ms
^C
--- 18.22.122.11 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1034ms
rtt min/avg/max/mdev = 0.462/0.585/0.788/0.123 ms
root@localhost ~]#

```

Middle Terminal (Node1):

```

Fedoralinux39 (Server Edition)
Kernel 6.6.13-200.fc39.x86_64 on an x86_64 (tty1)
Web console: https://localhost:9090/
localhost login: root
Password:
Last login: Tue Apr 16 12:35:22 on ttym1
Iroot@localhost ~]# ping 18.22.122.11
PING 18.22.122.11 (18.22.122.11) 56(84) bytes of data.
64 bytes from 18.22.122.11: icmp_seq=1 ttl=64 time=0.115 ms
64 bytes from 18.22.122.11: icmp_seq=2 ttl=64 time=0.842 ms
^C
--- 18.22.122.11 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.842/0.996/1.158/0.154 ms
Iroot@localhost ~]#

```

Right Terminal (Node2):

```

Fedoralinux39 (Server Edition)
Kernel 6.6.13-200.fc39.x86_64 on an x86_64 (tty1)
Web console: https://localhost:9090/
localhost login: root
Password:
Last login: Tue Apr 16 12:35:37 on ttym1
Iroot@localhost ~]# ping 18.22.122.11
PING 18.22.122.11 (18.22.122.11) 56(84) bytes of data.
64 bytes from 18.22.122.11: icmp_seq=1 ttl=64 time=0.387 ms
64 bytes from 18.22.122.11: icmp_seq=2 ttl=64 time=0.886 ms
^C
--- 18.22.122.11 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1005ms
rtt min/avg/max/mdev = 0.387/0.596/0.886/0.289 ms
Iroot@localhost ~]#

```

Figura 8. Conexión de los nodos a través de la red aislada Almacenamiento.

Y en la **Figura 9** se puede observar que desde la red NAT nombrada como Cluster hay conexión al exterior al lanzar trazas ICMP a la dirección 8.8.8.8.

```

pipe 3
[root@localhost ~]# ip addr show enp
enp1s0  enp7s0
[root@localhost ~]# ip addr show enp7s0
3: enp7s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:b6:ba:a0 brd ff:ff:ff:ff:ff:ff
    inet 192.168.140.124/24 brd 192.168.140.255 scope global dynamic nopref ixroute enp7s0
        valid_lft 3370sec preferred_lft 3370sec
    inet6 fe80::8e56:bfc:8156:f311/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[root@localhost ~]# ip route add default via 192.168.140.1 dev enp7s0
[root@localhost ~]# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=113 time=31.3 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=113 time=31.7 ms
^C
--- 8.8.8.8 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 31.313/31.489/31.666/0.176 ms
[root@localhost ~]#

```

Figura 9. Conexión al exterior de un nodo a través de la red NAT Cluster.

También se configuró para que fuese de forma permanente a través del directorio /etc/sysconfig/network-scripts/ **para asegurar la persistencia de la configuración** ya que con nmcli no se logró este hecho a menos que se usara la opción --permanent, pero fue útil para hacer modificaciones de forma dinámica hasta lograr su completa configuración.

Dentro de este directorio, por tanto, se creará un fichero para cada interfaz que se desee con el nombre “ifcfg-\$interfaz_de_red” para cada máquina. En la **Figura 10** se aprecia cómo se configuró en el caso de la maquina Almacenamiento con las mismas características que se siguió con *nmcli*. Con la opción de BOOTPROTO=none se desactivaría el DHCP, aunque al momento se crear la red virtual esta opción ya estaba desactivada por el gestor de KVM. Al momento se hacer la captura de la **Figura 10**, en ese momento no se había añadido la opción de “GATEWAY0=10.22.122.1”, pero se añadió posteriormente.

Almacenamiento en QEMU/KVM

Archivo Máquina Virtual Vista Enviar Tecla

```
[root@localhost ~]# ifconfig
enp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.22.122.10 netmask 255.255.255.0 broadcast 10.22.122.255
                inet6 fe80::5e8:ff:fed7:7f1f prefixlen 64 scopeid 0x20<link>
                    ether 52:54:00:d7:7f:1f txqueuelen 1000 (Ethernet)
                    RX packets 0 bytes 0 (0.0 B)
                    RX errors 0 dropped 0 overruns 0 frame 0
                    TX packets 19 bytes 2362 (2.3 KiB)
                    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp7s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.140.124 netmask 255.255.255.0 broadcast 192.168.140.255
                inet6 fe80::8e56:bfc:8156:f311 prefixlen 64 scopeid 0x20<link>
                    ether 52:54:00:b6:ba:a0 txqueuelen 1000 (Ethernet)
                    RX packets 18 bytes 2324 (2.2 KiB)
                    RX errors 0 dropped 11 overruns 0 frame 0
                    TX packets 50 bytes 4407 (4.3 KiB)
                    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
                inet6 ::1 prefixlen 128 scopeid 0x10<host>
                    loop txqueuelen 1000 (Local Loopback)
                    RX packets 0 bytes 0 (0.0 B)
                    RX errors 0 dropped 0 overruns 0 frame 0
                    TX packets 0 bytes 0 (0.0 B)
                    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@localhost ~]# cat /etc/sysconfig/network-scripts/ifcfg-enp1s0
DEVICE="enp1s0"
ONBOOT="yes"
BOOTPROTO="none"
NM_CONTROLLED="no"
IPADDR0="10.22.122.10"
PREFIX0="255.255.255.0"
[root@localhost ~]#
```

Figura 10. Fichero de configuración de red para el nodo de Almacenamiento. Para el resto de los nodos, se procede de forma similar, verificando que la red virtual se asocia a dicha interfaz de red con el nombre “ifcfg-\$interfaz_de_red”.

Tanto para la herramienta nmcli y con el fichero de configuración de interfaz de red, se siguió la fuente [4].

Por último, cada nombre de dominio se debe especificar para cada máquina: *nodo1.vpd.com* en *Nodo1* y *nodo2.vpd.com* en *Nodo2*; y *almacenamiento.vpd.com* en *Almacenamiento*. Según la máquina, se ejecuta una de las siguientes órdenes según le corresponda:

```
# hostnamectl set-hostname almacenamiento.vpd.com
# hostnamectl set-hostname nodo1.vpd.com
# hostnamectl set-hostname nodo2.vpd.com
```

4.2 Creación de la infraestructura básica iSCS

4.2.1 Exportar el disco sdb vía iSCSI del nodo target

Se debe iniciar la utilidad targetcli

A continuación, se creará la entidad de almacenamiento del disco sdb (/dev/sdb) que es el que será exportado.

```
# targetcli  
# cd backstores/block/  
# create name=discosdb dev=/dev/sdb
```

Una vez creado, se crea el objeto iSCSI al que será asociado a la entidad de almacenamiento recién creada.

```
# cd /iscsi/  
# /iscsi> create wwn= iqn.2024-04.com.vpd:ServidorApache
```

En este punto, el siguiente paso consistirá en asociar la dirección IP y portal al objeto iSCSI recién creado. El portal o **puerto** se usará el por **defecto 3260**, y se usará de dirección IP, la propia dirección del nodo de almacenamiento 10.22.122.10. Primero hay que ubicarse en la configuración del objeto iSCSI y luego crear la propia configuración del portal.

```
# /iscsi > cd iqn.2024-04.com.vpd:servidorapache/tpg1/portals  
# /iscsi/.../tpg1/portals> create 10.22.122.10
```

De esta manera, ya se ha especificado la dirección IP y puerto al objeto iSCSI, por lo que se deberá configurar la LUN del objeto iSCSI asociada a la entidad de almacenamiento. Para ello, se ubica en la configuración de LUN y se crea el identificador para acceder al volumen.

```
# cd ../luns  
# /iscsi/.../luns > create /backstores/block/dicosdb
```

Para finalizar, para poder asociar cada nodo al objeto iSCSI, se debe configurar la lista de control de acceso (ACL) que controla la autorización del objeto. En este paso, hay que introducir el nombre de cara al servicio iSCSI de los nodos initiator

que se configura en el fichero de configuración `/etc/iscsi/initiatorname.iscsi`. El contenido del archivo es:

- `InitiatorName= iqn.2024-04.com.vpd:nodo1`; para Nodo1.
- `InitiatorName= iqn.2024-04.com.vpd:nodo2`; para Nodo2.

(Durante el procedimiento de la **Figura 11** se introdujo otro nombre, y se hizo necesario eliminar el erróneo para luego introducir el correcto). Por tanto, el procedimiento correcto debe ser:

```
# cd ../acls
# /iscsi/.../acls >create wwn=iqn.2024-04.com.vpd:nodo1
# /iscsi/.../acls >create wwn=iqn.2024-04.com.vpd:nodo2
```

El resultado final de la configuración se puede consultar en la **Figura 12**, mientras que el procedimiento se puede encontrar en la **Figura 11**.

```
/backstores/block> create name=discosdb dev=/dev/sdb
Created block storage object discosdb using /dev/sdb.
/backstores/block> cd /iscsi/
/iscsi> create wwn=iqn.2024-04.com.vpd:ServidorApache
Created target iqn.2024-04.com.vpd:servidorapache.
Created TPG 1.
└─< ! portal not created, TPGs within a target cannot share ip:port.
/iscsi> ls
o- iscsi
  o- iqn.2024-04.com.vpd:discosda
    o- tpg1
      o- acls
        o- iqn.2024-04.com.vpd:nodo1
          o- mapped_lun0
        o- iqn.2024-04.com.vpd:nodo2
          o- mapped_lun0
      o- luns
        o- lun0
      o- portals
        o- 10.22.122.10:3260
    o- iqn.2024-04.com.vpd:servidorapache
      o- tpg1
        o- acls
        o- luns
        o- portals
/iscsi> cd iqn.2024-04.com.vpd:servidorapache/tpg1/portals
/iscsi/10.22.122.10:3260> ls
o- portals
/iscsi/10.22.122.10:3260> create 10.22.122.10
Using default IP port 3260.
Created network portal 10.22.122.10:3260.
/iscsi/10.22.122.10:3260> ls
o- portals
  o- 10.22.122.10:3260
/iscsi/10.22.122.10:3260> cd ../luns
/iscsi/10.22.122.10:3260> create
/backstores/block/discosda /backstores/block/discosdb anaconda-ks.cfg
iscsi_voll/                                         storage_object=
iscsi_voll/   add_mapped_luns= lun=
/iscsi/10.22.122.10:3260> create /backstores/block/discosdb
Created LUN 0.
/iscsi/10.22.122.10:3260> cd ../acls
/iscsi/10.22.122.10:3260> create wwn=iqn.2024-04.com.vpd:ServidorApache:nodo1
Created Node ACL for iqn.2024-04.com.vpd:servidorapache:nodo1
Created mapped LUN 0.
/iscsi/10.22.122.10:3260> create wwn=iqn.2024-04.com.vpd:ServidorApache:nodo2
Created Node ACL for iqn.2024-04.com.vpd:servidorapache:nodo2
Created mapped LUN 0.
/iscsi/10.22.122.10:3260> create wwn=iqn.2024-04.com.vpd:nodo2
Created Node ACL for iqn.2024-04.com.vpd:nodo2
Created mapped LUN 0.
/iscsi/10.22.122.10:3260> delete wwn=iqn.2024-04.com.vpd:ServidorApache:nodo1
Deleted Node ACL iqn.2024-04.com.vpd:ServidorApache:nodo1.
/iscsi/10.22.122.10:3260> delete wwn=iqn.2024-04.com.vpd:ServidorApache:nodo2
Deleted Node ACL iqn.2024-04.com.vpd:ServidorApache:nodo2.
```

Figura 11. Procedimiento de configuración de exportación del disco sdb a través de iSCSI.

```

/iscsi/iqn.20...che/tpg1/acls> ls /iscsi/
o- iscsi ..... [Targets: 2] [TPGs: 1]
| o- iqn.2024-04.com.vpd:discosda ..... [no-gen-acls, no-auth] [ACLs: 2]
| | o- tpg1 ..... [Mapped LUNs: 1]
| | | o- acls ..... [ACLs: 2]
| | | | o- iqn.2024-04.com.vpd:nodo1 ..... [lun0 block/discosda (rw)]
| | | | o- mapped_lun0 ..... [Mapped LUNs: 1]
| | | | o- iqn.2024-04.com.vpd:nodo2 ..... [lun0 block/discosda (rw)]
| | | | o- mapped_lun0 ..... [Mapped LUNs: 1]
| | | o- luns ..... [LUNs: 1]
| | | | o- lun0 ..... [block/discosda (/dev/sda) (default_tg_pt_gpp)] [Portals: 1] [OK]
| | | o- portals ..... [Portals: 1]
| | | | o- 10.22.122.10:3260 ..... [OK]
| | o- iqn.2024-04.com.vpd:servidorapache ..... [TPGs: 1]
| | | o- tpg1 ..... [no-gen-acls, no-auth] [ACLs: 2]
| | | | o- acls ..... [ACLs: 2]
| | | | | o- iqn.2024-04.com.vpd:nodo1 ..... [Mapped LUNs: 1]
| | | | | o- mapped_lun0 ..... [Mapped LUNs: 1]
| | | | | o- iqn.2024-04.com.vpd:nodo2 ..... [Mapped LUNs: 1]
| | | | | o- mapped_lun0 ..... [Mapped LUNs: 1]
| | | | o- luns ..... [LUNs: 1]
| | | | | o- lun0 ..... [block/discosdb (/dev/sdb) (default_tg_pt_gpp)] [Portals: 1] [OK]
| | | | o- portals ..... [Portals: 1]
| | | | | o- 10.22.122.10:3260 ..... [OK]
/iscsi/iqn.20...che/tpg1/acls>

```

Figura 11. Configuración final de exportación del disco sdb a través de iSCSI.

En la **Figura 12** se observa cómo el nodo initiator Nodo2 lanza una petición de *discovery* cuya respuesta del nodo target responde con los dos discos configurados durante la práctica.

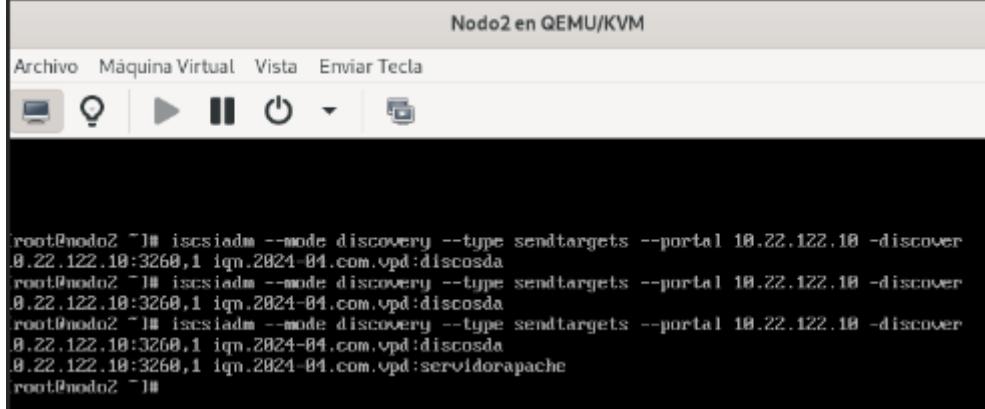


Figura 12. Verificación de los LUNs exportados por el nodo target.

Por tanto, ya el disco está preparado para ser conectado a un nodo initiator. De esta manera, se conecta a la unidad LUN exportada y posteriormente se procede concluye este apartado en la **Figura 13** donde se verifica que el disco sdb aparece en el nodo initiator.

```
# iscsiadm --mode node --targetname iqn.2024-04.com.vpd:servidorapache --portal 10.22.122.10 --login
```

```
[root@nodo2 ~]# iscsiadm -m session
tcp: [1] 10.22.122.10:3260,1 iqn.2024-04.com.vpd:discosda (non-flash)
tcp: [4] 10.22.122.10:3260,1 iqn.2024-04.com.vpd:servidorapache (non-flash)
[root@nodo2 ~]# iscsiadm -m node -T iqn.2024-04.com.vpd:discosda -p 10.22.122.10 -o update -n node.startup -v manual
[root@nodo2 ~]# iscsiadm -m session
tcp: [1] 10.22.122.10:3260,1 iqn.2024-04.com.vpd:discosda (non-flash)
tcp: [4] 10.22.122.10:3260,1 iqn.2024-04.com.vpd:servidorapache (non-flash)
[root@nodo2 ~]# iscsiadm -m node -T iqn.2024-04.com.vpd:discosda -p 10.22.122.10 -u
Logging out of session [sid: 1, target: iqn.2024-04.com.vpd:discosda, portal: 10.22.122.10,3260]
Logout of [sid: 1, target: iqn.2024-04.com.vpd:discosda, portal: 10.22.122.10,3260] successful.
[root@nodo2 ~]# iscsiadm -m session
tcp: [4] 10.22.122.10:3260,1 iqn.2024-04.com.vpd:servidorapache (non-flash)
[root@nodo2 ~]# lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sdb      8:16   0   1G  0 disk
sr0     11:0    1 1024M 0 rom
zram0   251:0   0  1,9G 0 disk [SWAP]
vda     252:0   0   10G 0 disk
└─vda1  252:1   0   1M  0 part
└─vda2  252:2   0   1G  0 part /boot
└─vda3  252:3   0   9G  0 part
  └─fedora-root 253:0   0   9G  0 lvm /
[root@nodo2 ~]#
```

Figura 13. Disco sdb correctamente exportado a un nodo initiator.

4.2.2 Creación de un volumen lógico con sistema de archivo XFS

Para este apartado, se ha seguido las instrucciones tanto de [1] como de [5]. Aunque no se exponga todos los comandos por no alargar el documento tales como `pvscan`, `vgdisplay`, `lvs`, `pvs`, entre otros comandos; fue de gran utilidad ambas fuentes para poder consultar la lista de volúmenes físicos y lógicos, y verificación de la configuración realizada o corrección de errores.

En primer lugar, se crea el volumen físico, de tamaño 900 MB en el disco sdb. Se puede verificar en la **Figura 14**.

```
# pvcreate /dev/sdb -setphysicalvolumesize 900MB
```

```
[root@nodo2 ~]# lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sdb      8:16   0   1G  0 disk
sr0     11:0    1 1024M 0 rom
zram0   251:0   0  1,9G 0 disk [SWAP]
vda     252:0   0   10G 0 disk
└─vda1  252:1   0   1M  0 part
└─vda2  252:2   0   1G  0 part /boot
└─vda3  252:3   0   9G  0 part
  └─fedora-root 253:0   0   9G  0 lvm /
[root@nodo2 ~]# pvs
  PV          VG      Fmt Attr PSize  PFree
  /dev/vda3  fedora  lvm2 a-- <9,00g    0
[root@nodo2 ~]# pvcreate /dev/sdb --setphysicalvolumesize 900MB
WARNING: ext4 signature detected on /dev/sdb at offset 1080. Wipe it? [y/n]: y
Wiping ext4 signature on /dev/sdb.
Physical volume "/dev/sdb" successfully created.
[root@nodo2 ~]# pvs
  PV          VG      Fmt Attr PSize  PFree
  /dev/sdb        lvm2 ---  900,00m 900,00m
  /dev/vda3  fedora  lvm2 a-- <9,00g    0
[root@nodo2 ~]#
```

Figura 14. Creación y verificación de un disco físico en el dispositivo sdb

A continuación, se crea el grupo de volumen lógico en el volumen físico creado:

```
# vgcreate ApacheVG /dev/sdb
```

Posteriormente, se desactiva la activación automática en el arranque de los sistemas en el grupo de volumen lógico y se deshabilita el bloqueo del grupo de volúmenes para que otros sistemas pueda tener acceso al grupo de volúmenes y pueda crearlo automáticamente. Se puede verificar en la **Figura 15**.

```
# vgchange ApacheVG --setautoactivation n
```

```
[root@nodo2 ~]# vgchange --help | grep auto
      --setautoactivation y\ln )
[ -A|--autobackup y\ln ]
[ -A|--autobackup y\ln ]
[ -A|--autobackup y\ln ]
[ -A|--autobackup y\ln ]
[     --autoactivation String ]
[ -A|--autobackup y\ln ]
[root@nodo2 ~]# vgchange ApacheVG --autoactivation no
  No command with matching syntax recognised. Run 'vgchange --help' for more information.
[root@nodo2 ~]# vgchange ApacheVG --autoactivation "no"
  No command with matching syntax recognised. Run 'vgchange --help' for more information.
[root@nodo2 ~]# vgchange ApacheVG --autoactivation "n"
  No command with matching syntax recognised. Run 'vgchange --help' for more information.
[root@nodo2 ~]# vgchange ApacheVG --setautoactivation n
  Volume group "ApacheVG" successfully changed.
```

Figura 15. Creación y verificación de un disco físico en el dispositivo sdb

En este punto, se creará un volumen lógico llamado ApacheLV de tamaño 900 Mbytes en el grupo de volumen recién creado.

```
# lvcreate -L 900MB -n ApacheLV ApacheVG
```

Ejecutando lvs se puede consultar la configuración de los volúmenes así como al grupo de volumen al que pertenece. Se puede observar en la **Figura 16**.

```
[root@nodo2 ~]# lvcreate -L 900MB -n ApacheLV ApacheVG
  Logical volume "ApacheLV" created.
[root@nodo2 ~]# lvs
  LV        UG      Attr       LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  ApacheLV  ApacheVG -wi-a----  900,00m
  root      fedora  -wi-ao---  <9,00g
```

Figura 16. Listado de volúmenes lógicos.

Por último, se procede a formatear el volumen lógico como sistema de archivo XFS, se puede observar el procedimiento en la **Figura 17**.

```
# mkfs.xfs /dev/ApacheVG/ApacheLV
```

```
[root@nodo2 ~]# mkfs.xfs /
afs/    bin/   boot/  dev/   etc/   home/  lib/   lib64/  media/  mnt/   opt/   proc/  ro
[root@nodo2 ~]# mkfs.xfs /dev/ApacheVG/ApacheLV
meta-data=/dev/ApacheVG/ApacheLV isize=512    agcount=4, agsize=57600 blks
          =                      sectsz=512  attr=2, projid32bit=1
          =                      crc=1     finobt=1, sparse=1, rmapbt=0
          =                      reflink=1  bigtime=1 inobtcount=1 nrext64=0
data     =                      bsize=4096   blocks=230400, imaxpct=25
          =                      sunit=8    swidth=0 blks
naming   =version 2           bsize=4096   blocks=16384, version=2
log      =internal log        bsize=4096   sunit=0 blks, lazy-count=1
          =                      sectsz=512  extsz=4096  blocks=0, rtextents=0
realtime =none                sectsz=512  extsz=4096  blocks=0, rtextents=0
[root@nodo2 ~]#
```

Figura 17. Creación de sistema de archivos XFS en el volumen lógico ApacheLV.

Esta configuración fue realizada en el nodo Nodo2. Ahora se debe añadir dicho volumen lógico al Nodo1. Para ello, ambos nodos están conectados al nodo target:

```
# iscsiadm --mode node --targetname iqn.2024-
04.com.vpd:servidorapache --portal
```

Nodo2 debe añadir el disco sdb al grupo de voluemens del sistema. Para ello ejecuta lsblk para comprobar que correctamente el disco conectado está mapeado como /dev/sdb, y una vez verificado, se ejecuta:

```
# lvmdevices -adddev /dev/sdb
```

Todo este procedimiento se puede verificar en la **Figura 18.**

```
[root@nodo1 ~]# iscsiadm --mode discovery --type sendtargets --portal 10.22.122.10 --discover
10.22.122.10:3260,1 iqn.2024-04.com.vpd:discosda
10.22.122.10:3260,1 iqn.2024-04.com.vpd:servidorapache
[root@nodo1 ~]# iscsiadm -m session
tcp: [1] 10.22.122.10:3260,1 iqn.2024-04.com.vpd:discosda (non-flash)
[root@nodo1 ~]# iscsiadm --mode node --targetname iqn.2024-04.com.vpd:servidorapache --portal 10.22.122.10 --login
Logging in to [iface: default, target: iqn.2024-04.com.vpd:servidorapache, portal: 10.22.122.10,3260]
Login to [iface: default, target: iqn.2024-04.com.vpd:servidorapache, portal: 10.22.122.10,3260] successful.
[root@nodo1 ~]# lvmdevices -adddev /dev/sd
sda  sdb
[root@nodo1 ~]# lvmdevices -adddev /dev/sdb
lvmdevices: opción inválida -- 'a'
  Error during parsing of command line.
[root@nodo1 ~]# lvmdevices --adddev /dev/sdb
[root@nodo1 ~]# lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda      8:0    0   1G  0 disk /share_sda
sdb      8:16   0   1G  0 disk
sr0     11:0    1 1024M 0 rom
zram0   251:0   0  1.9G 0 disk [SWAP]
vda     252:0   0   18G 0 disk
└─vda1   252:1   0   1M  0 part
└─vda2   252:2   0   1G  0 part /boot
└─vda3   252:3   0   9G  0 part
      └─fedora-root 253:0   0   9G  0 lvm /
```

Figura 18. Nodo1 añade con éxito al disco sdb del nodo target.

Se comprueba que ha sido añadido sdb tiene el volumen lógico correctamente configurado, y luego con vgchange se activa el volumen lógico manualmente. Finalmente se verifica con lsblk su configuración:

```
# pvscan  
# vgchange -a y ApacheVG
```

```
[root@nodo1 ~]# pvscan  
  PV /dev/sdb      VG ApacheVG    lvm2 [868,00 MiB / 68,00 MiB free]  
  PV /dev/vda3     VG fedora      lvm2 [<9,00 GiB / 0     free]  
    Total: 2 [9,84 GiB] / in use: 2 [9,84 GiB] / in no VG: 0 [0     ]  
[root@nodo1 ~]# vgchange -a y ApacheVG  
  1 logical volume(s) in volume group "ApacheVG" now active  
[root@nodo1 ~]# lsblk  
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS  
sda        8:0    0   1G  0 disk /share_sda  
sdb        8:16   0   1G  0 disk  
└─ApacheVG-ApacheLV 253:1   0 800M  0 lvm  
sr0       11:0    1 1024M  0 rom  
zram0     251:0   0  1,9G  0 disk [SWAP]  
vda       252:0   0   10G  0 disk  
├─vda1     252:1   0    1M  0 part  
├─vda2     252:2   0    1G  0 part /boot  
└─vda3     252:3   0    9G  0 part  
  └─fedora-root 253:0   0    9G  0 lvm  /  
[root@nodo1 ~]# _
```

Figura 18. Nodo1 configura con éxito el disco sdb y su volumen lógico configurado en el Nodo2.

5. Pruebas / Validación

El correcto mapeo de los dispositivos vda, sda y sdb; y la configuración de almacenamiento de red ha sido verificado implícitamente durante la práctica en el **apartado 4**.

En este apartado se verificará la exportación y uso de los discos en los nodos initiator, y la correcta configuración de red.

5.1 Verificación de exportación de los discos sda.

Teniendo en cuenta que es un sistema de archivos ext4, no se puede montar al mismo tiempo en ambos nodos initiator.

Se creará un archivo de prueba fromnode2 desde el nodo Nodo2. Para ello, se montará /dev/sda en /mnt, y posteriormente se desmontará el volumen en dicho nodo. Luego, se montará /dev/sda en el nodo Nodo1 y se verificará que se haya creado dicho archivo. Se puede verificar en la **Figura 19**.

```

[root@nodo1 ~]# umount /mnt
[root@nodo1 ~]# lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda      8:0    0   1G  0 disk
sdb      8:16   0   1G  0 disk
└─ApacheVG-ApacheLV 253:1   0 880M 0 lvm
sr0     11:0    1 1024M 0 rom
zram0   251:0   0  19G 0 disk [SWAP]
vda    252:0   0  10G 0 disk
├─vda1  252:1   0   1M 0 part
└─vda2  252:2   0   1G 0 part /boot
└─vda3  252:3   0   9G 0 part
└─fedora-root 253:0   0   9G 0 lvm /
[root@nodo1 ~]# mount /dev/sda /mnt
[root@nodo1 ~]# ls /mnt/
fromnode2 lost+found test1
[root@nodo1 ~]# umount /mnt
[root@nodo1 ~]#

```

```

vda      252:0   0   10G 0 disk
└─vda1  252:1   0   1M 0 part
└─vda2  252:2   0   1G 0 part /boot
└─vda3  252:3   0   9G 0 part
└─fedora-root 253:0   0   9G 0 lvm /
[root@nodo2 ~]# ls /mnt/
node1 test1 test2
[root@nodo2 ~]# cat /mnt/test1
VPD1
[root@nodo2 ~]# umount /mnt
[root@nodo2 ~]# mount /dev/ApacheVG/ApacheLV /mnt
[root@nodo2 ~]# ls /mnt/
media/ mnt/
[root@nodo2 ~]# ls /mnt/
another node1 test1 test2
[root@nodo2 ~]# lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda      8:0    0   1G  0 disk
sdb      8:16   0   1G  0 disk
└─ApacheVG-ApacheLV 253:1   0 880M 0 lvm /mnt
sr0     11:0    1 1024M 0 rom
zram0   251:0   0  19G 0 disk [SWAP]
vda    252:0   0  10G 0 disk
├─vda1  252:1   0   1M 0 part
└─vda2  252:2   0   1G 0 part /boot
└─vda3  252:3   0   9G 0 part
└─fedora-root 253:0   0   9G 0 lvm /
[root@nodo2 ~]# umount /mnt
[root@nodo2 ~]# mount /dev/sda /mnt
[root@nodo2 ~]# touch /mnt/fromnode2
[root@nodo2 ~]# umount /mnt
[root@nodo2 ~]#

```

Figura 19. Verificación de uso del disco sda.

5.2 Verificación de exportación de los discos sdb.

Parecido a la verificación anterior, se montará manualmente el sistema de archivos contenido en `/dev/sda` en el directorio `/mnt` y se creará el archivo de prueba `/mnt/test1` conteniendo el siguiente texto “VPD1” en el nodo Nodo1.

Se ejecuta los siguientes comandos en Nodo1:

```

# mount /dev/ApacheVG/ApacheLV /mnt
# touch /mnt/test1; echo "VPD1" > /mnt/test1
# umount /mnt

```

De forma análoga a lo que se hizo en el nodo Nodo1, se ejecuta los siguientes comandos en Nodo2:

```

# mount /dev/ApacheVG/ApacheLV /mnt
# cat /mnt/test1
# touch /mnt/test2; echo "VPD2" > /mnt/test2
# umount /mnt

```

Finalmente, en la **Figura 20**, se observa el correcto empleo del disco sdb

```

└─fedora-root 253:0   0   9G 0 lvm /
[root@nodo1 ~]# mount /dev/ApacheVG/ApacheLV /mnt
[root@nodo1 ~]# ls /mnt
[root@nodo1 ~]# touch /mnt/node1
[root@nodo1 ~]# umount /mnt
[root@nodo1 ~]# mount /dev/ApacheVG/ApacheLV /mnt
[root@nodo1 ~]# cat /mnt/
node1 test1
[root@nodo1 ~]# cat /mnt/test2
VPD2
[root@nodo1 ~]# touch /mnt/test1; echo "VPD1" > /mnt/test1
[root@nodo1 ~]# umount /mnt
[root@nodo1 ~]#

```

```

└─vda3  252:3   0   9G 0 part
└─fedora-root 253:0   0   9G 0 lvm /
[root@nodo2 ~]# mount /dev/ApacheVG/ApacheLV /mnt
[root@nodo2 ~]# ls /mnt/
[root@nodo2 ~]# touch /mnt/test1
[root@nodo2 ~]# cat /mnt/
node1 test1 test2
[root@nodo2 ~]# cat /mnt/test1
VPD1
[root@nodo2 ~]#

```

Figura 20. Verificación de uso del disco sdb en el volumen lógico ApacheLV perteneciente al grupo de volumen ApacheVG.

5.3 Verificación de las redes virtuales configuradas.

Se procede a consultar tanto la tabla de enrutamiento como las direcciones IP de cada nodo para observar su correcta configuración en la **Figura 21**. La verificación de conectividad entre redes se demuestra implícitamente durante la práctica en el **apartado 4**, siendo más concretamente en la **Figura 8** y en la **Figura 9**.

The image shows two terminal windows side-by-side. Both windows have a title bar at the top with icons for file, virtual machine, view, and send key. The left window is titled "Almacenamiento en QEMU/KVM" and the right window is titled "Nodo1 en QEMU/KVM". Both windows show a command-line interface with the "ip" command being run. The output of the command shows network interfaces (lo, eth0, eth1, eth2) with their respective MTU, queueing discipline (qdisc), link layer information (MAC address), and IP configurations (inet). The output is identical for both nodes, indicating they have been configured identically.

```
[root@almacenamiento ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default 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 noprefixroute
            valid_lft forever preferred_lft forever
2: empis0: <POINTOPOINT,BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:d7:7f:1f brd ff:ff:ff:ff:ff:ff
        inet 10.22.122.10/24 brd 10.22.122.255 scope global noprefixroute enp1s0
            valid_lft forever preferred_lft forever
        inet6 fe80::5b54:ff:fed7:7f1f/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
3: emp7s0: <POINTOPOINT,BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:b6:ba:a0 brd ff:ff:ff:ff:ff:ff
        inet 192.168.140.124/24 brd 192.168.140.255 scope global dynamic noprefixroute enp7s0
            valid_lft 2854sec preferred_lft 2854sec
        inet6 fe80::8e56:bfc:8156:f311/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
[root@almacenamiento ~]# ip r
default via 192.168.140.1 dev enp7s0 proto dhcp src 192.168.140.124 metric 101
10.22.122.0/24 dev emp1s0 proto kernel scope link src 10.22.122.10 metric 102
192.168.140.0/24 dev emp7s0 proto kernel scope link src 192.168.140.124 metric 101
[root@almacenamiento ~]#
[root@nodo1 ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default 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 noprefixroute
            valid_lft forever preferred_lft forever
2: emp1s0: <POINTOPOINT,BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:e8:b6:66 brd ff:ff:ff:ff:ff:ff
        inet 10.22.122.11/24 brd 10.22.122.255 scope global noprefixroute enp1s0
            valid_lft forever preferred_lft forever
        inet6 fe80::5b54:ff:fee8:b666/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
3: emp7s0: <POINTOPOINT,BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:38:60:b1 brd ff:ff:ff:f1:ff:ff
        inet 192.168.140.33/24 brd 192.168.140.255 scope global dynamic noprefixroute enp7s0
            valid_lft 2908sec preferred_lft 2908sec
        inet6 fe80::b17e:75bc:536:b9b9/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
[root@nodo1 ~]# ip r
default via 192.168.140.1 dev enp7s0 proto dhcp src 192.168.140.33 metric 101
10.22.122.0/24 dev emp1s0 proto kernel scope link src 10.22.122.11 metric 102
192.168.140.0/24 dev emp7s0 proto kernel scope link src 192.168.140.33 metric 101
```

Nodo2 en QEMU/KVM

Archivo Máquina Virtual Vista Enviar Tecla

[root@nodo2 ~]# ip a

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default 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 noprefixroute
 valid_lft forever preferred_lft forever

2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
link/ether 52:54:00:5f:4e:d7 brd ff:ff:ff:ff:ff:ff
inet 10.22.122.12/24 brd 10.22.122.255 scope global nopref ixroute enp1s0
 valid_lft forever preferred_lft forever
inet6 fe80::5054:ff:fe5f:4ed7/64 scope link nopref ixroute
 valid_lft forever preferred_lft forever

3: enp7s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
link/ether 52:54:00:2f:a1:29 brd ff:ff:ff:ff:ff:ff
inet 192.168.140.127/24 brd 192.168.140.255 scope global dynamic nopref ixroute enp7s0
 valid_lft 3056sec preferred_lft 3056sec
inet6 fe80::d8df:9d59:3ff2:2ced/64 scope link nopref ixroute
 valid_lft forever preferred_lft forever

[root@nodo2 ~]# ip r

default via 192.168.140.1 dev enp7s0 proto dhcp src 192.168.140.127 metric 101
10.22.122.0/24 dev enp1s0 proto kernel scope link src 10.22.122.12 metric 102
192.168.140.0/24 dev enp7s0 proto kernel scope link src 192.168.140.127 metric 101

[root@nodo2 ~]#

Figura 19. Configuración de redes virtuales.

6. Fuentes de información

Referencias

- [1] “Configuring and managing logical volumes,” RED HAT ENTERPRISE LINUX 8, [Online]. Available: https://access.redhat.com/documentation/id-id/red_hat_enterprise_linux/8/html/configuring_and_managing_logical_volumes/index.
 - [2] A. Q. R. García, “Práctica 6: Instalación de un servicio de almacenamiento iSCSI.,” Universidad de Las Palmas de Gran Canaria, 2024.
 - [3] R. Saive, “How to Configure Network Connection Using ‘nmcli’ Tool,” TecMint, 11 MAnyo 2020. [Online]. Available: <https://www.tecmint.com/nmcli-configure-network-connection/>.
 - [4] “<https://h4ckseed.wordpress.com/2023/09/08/configurar-direccion-ip-estatica-en-gnu-linux/>,” H4ckseed, 8 Septiembre 2023. [Online]. Available: <https://h4ckseed.wordpress.com/2023/09/08/configurar-direccion-ip-estatica-en-gnu-linux/>.

- [5] “Logical volume activation [Chapter 13],” Red Hat Enterprise Linux 8, [Online]. Available: https://access.redhat.com/documentation/es-es/red_hat_enterprise_linux/8/html/configuring_and_managing_networking/configuring-a-network-bridge_configuring-and-managing-networking.

Práctica 7: Diseño y despliegue de un clúster básico

Óscar Alexander Martín Tacoronte

Grupo 01.04 (Cod. 44)

22 de mayo de 2024

1. Índice

1.	Índice	2
2.	Introducción	3
3.	Objetivos	4
3.1	Creación de la infraestructura básica del clúster	4
3.2	Instalación de un clúster básico y proporcionar un servidor web Apache en alta disponibilidad.....	4
➤	Creación de la infraestructura básica del clúster.....	4
➤	Instalación de un clúster básico y proporcionar un servidor web Apache en alta disponibilidad	4
4.	Desarrollo.....	5
4.2.1	Configuración final del clúster resultante.....	13
5.	Pruebas / Validación.....	17
5.1	Creación y configuración de las tres interfaces de red.....	17
5.2	Instalación y configuración del servicio httpd. ¡Error! Marcador no definido.	
5.3	Configuración final del clúster resultante y validación del servicio en alta disponibilidad.	21
6.	Fuentes de información	23
	Referencias	23

2. Introducción

En esta práctica se trata de diseñar la infraestructura básica un clúster que posteriormente será desplegado de forma que ofrezca un servicio en alta disponibilidad de un servidor web.

Atendiendo a las instrucciones, y a las fichas correspondientes seguidas en este informe [1] y [2], se detallará aquellos aspectos que hayan sido indicados, y que se podrá consultar en el apartado 3 donde se describe los objetivos del informe (por tanto, no estará todos los pasos).

3. Objetivos

3.1 Creación de la infraestructura básica del clúster

- Creación de los nuevos recursos de redes y almacenamiento virtuales necesarios a partir de los recursos ya preparados de la práctica anterior número 6.
- Instalación y configuración del servidor Apache.

3.2 Instalación de un clúster básico y proporcionar un servidor web Apache en alta disponibilidad

- Instalación y configuración del servicio de administración para alta disponibilidad, conocido por el término “*software High Availability Add-On*” de Red Hat.
- Creación del clúster a partir del servicio *Pacemaker/Corosync*, herramienta de *software High Availability Add-On*.
- Instalación, configuración y establecimiento del mecanismo de aislamiento de nodos conocido en inglés por el término “*fencing configuration*”
- Configuración del servicio de Apache, o “*httpd*”, para la ejecución en un clúster de alta disponibilidad.
- Configuración del almacenamiento para la ejecución controlada descentralizada del servicio de administración del clúster.
- Creación de los recursos y grupos de recursos del clúster.
- Despliegue del servicio en alta disponibilidad.

Consideraciones sobre los objetivos

A continuación, se deberá contemplar en este informe los siguientes aspectos según las indicaciones proporcionadas de cada parte:

1. Creación de la infraestructura básica del clúster:

- Creación y configuración de las tres interfaces de red, durante el desarrollo del **apartado 4**.
- Instalación y configuración del servicio *httpd*, durante el desarrollo del **apartado 4**.

2. Instalación de un clúster básico y proporcionar un servidor web Apache en alta disponibilidad:

- Configuración final del clúster resultante, durante el desarrollo del **apartado 4**.
- Validación del servicio desplegado del clúster en alta disponibilidad, en el **apartado 5**.

4. Desarrollo

4.1 Creación de la infraestructura básica del clúster

4.1.1 Creación y configuración de las tres interfaces de red.

A partir de la infraestructura ya configurada de la anterior práctica, se deberá crear una nueva red aislada de nombre Control sin el servicio DHCP activo. En esta red se utilizará para la coordinación del mecanismo de aislamiento. La red deberá ser de dirección 10.22.132.0/24, siendo el nodo1 con dirección estática 10.22.132.11 y nodo2 con 10.22.132.12.

En primer lugar, se crea la red a partir de la interfaz gráfica del virt-manager. Para ello, en **detalles de conexión – Redes Virtuales**, se accede al dialogo gráfico para crear la red y finalmente en la **Figura 1** se puede consultar los detalles de la configuración final.

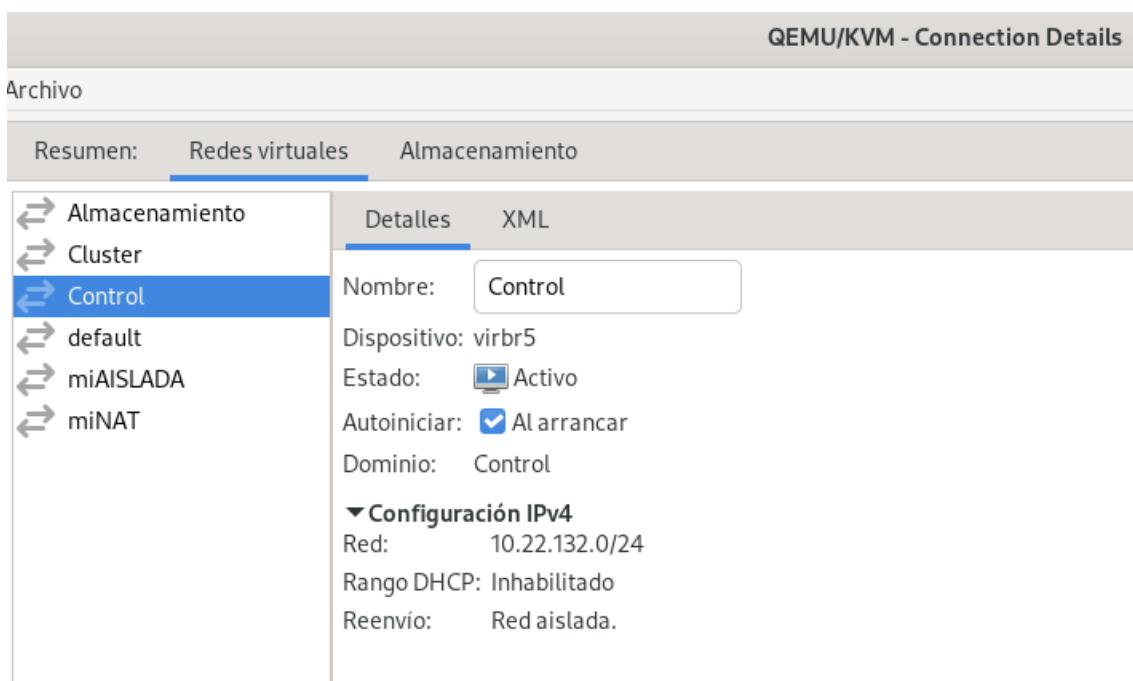


Figura 1. Creación de red virtual Control y su configuración final.

Una vez creada, el siguiente paso será la configuración en cada nodo que utilizará dicha red, es decir, nodo1 y nodo2. Para ello, se utilizará la herramienta **nmcli** [3] en cada nodo. El procedimiento se puede apreciar en la **Figura 2**, o bien con los siguientes comandos se resumen su configuración en cada nodo con su interfaz de red asociada a la red Control y con su dirección IP previamente comentada (**la conectividad de red y sus configuraciones será validada en el apartado 5**):

```
# nmcli connection add type ethernet ifname $interfaz_red
# nmcli connection modify $interfaz_red ipv4.addresses ifname
$direccion_ip
# nmcli connection modify $interfaz_red ipv4.method manual
# nmcli connection up $interfaz_red
# nmcli con show
```

```

[root@node01 ~]# nmcli connection add type ethernet ifname enp0s0
Conexión «ethernet-enp0s0» (a9891f86-13d2-4a0c-a89c-72a9c3b61568) añadida con éxito.
[root@node01 ~]# nmcli connection modify enp0s0 ipv4.addresses 10.22.132.11
Error: conexión «enp0s0» desconocida.
[root@node01 ~]# nmcli connection modify enp1s0 ipv4.addresses 10.22.132.11
enp1s0
[root@node01 ~]# nmcli connection add ifname enp0s8
Error: se necesita el argumento «type».
[root@node01 ~]# nmcli connection show
NAME                UUID                                  TYPE      DEVICE
ethernet-enp0s0      a9891f86-13d2-4a0c-a89c-72a9c3b61568  ethernet  enp0s8
Conexión cableada 1  4279b89f-fba3-3dd5-81a9-bb0a06310f964  ethernet  enp7s0
enp1s0               1a97b93e-c280-37a1-bc08-41fafebc1b06  ethernet  enp1s0
lo                  fe33be6b-0269-4404-a174-f1fb1b95be20  loopback lo
Conexión cableada 2  6ac903d7-56c8-3318-b1f1-818b2c6da169  ethernet  --
[root@node01 ~]# nmcli connection modify ethernet-enp0s0 ipv4.addresses 10.22.132.11
[root@node01 ~]# nmcli connection modify ethernet-enp0s0 ipv4.method manual
[root@node01 ~]# nmcli connection up
Conexión cableada 1 enp1s0           filename          id          path
Conexión cableada 2 ethernet-enp0s0 help            lo          uuid
[root@node01 ~]# nmcli connection up ethernet-enp0s0
Conexión activada con éxito (ruta activa D-Bus: /org/freedesktop/NetworkManager/ActiveConnection/13)
[root@node01 ~]# _


[root@node02 ~]# nmcli connection add vgo0 connection.id vgo0
Conexión «vgo0» añadida.
[root@node02 ~]# nmcli connection modify enp1s0 enp0s2 lo
Conexión «ethernet-enp0s0» (560d0c7d-5a24-49a0-a639-97212900545c) añadida con éxito.
[root@node02 ~]# nmcli connection modify
Conexión cableada 1 enp1s0           filename          id          path
Conexión cableada 2 ethernet-enp0s0 help            lo          uuid
[root@node02 ~]# nmcli connection modify e
enp1s0           ethernet-enp0s0
[root@node02 ~]# nmcli connection modify ethernet-enp0s0 ipv4.addresses 10.22.132.12 ipv4.dns-search
1.ipv4.addresses          ipv4.dns-search          ipv6.addr-gen-mode
2.ipv4.auto-route-ext-gw  ipv4.gateway          ipv6.auto-route-ext-gw
3.ipv4.dad-timeout       ipv4.ignore-auto-dns  ipv6.dhcp-duid
4.ipv4.dhcp-client-id    ipv4.ignore-auto-routes  ipv6.dhcp-hostname
5.ipv4.dhcp-fqdn         ipv4.link-local        ipv6.dhcp-hostname-flags
6.ipv4.dhcp-hostname     ipv4.may-fail          ipv6.dhcp-iaid
7.ipv4.dhcp-hostname-flags  ipv4.method          ipv6.dhcp-pd-hint
8.ipv4.dhcp-iaid         ipv4.never-default   ipv6.replace-local-rule
9.ipv4.dhcp-reject-servers  ipv4.replace-local-rule  ipv6.dhcp-timeout
10.ipv4.dhcp-send-hostname  ipv4.required-timeout  ipv6.dns
11.ipv4.dhcp-timeout     ipv4.route-metric      ipv6.dns-options
12.ipv4.dhcp-vendor-class-identifier  ipv4.routes          ipv6.dns-priority
13.ipv4.dns               ipv4.route-table      ipv6.dns-search
14.ipv4.dns-options      ipv4.routing-rules    ipv6.gateway
15.ipv4.dns-priority     ipv6.addresses       ipv6.ignore-auto-dns
[root@node02 ~]# nmcli connection modify ethernet-enp0s0 ipv4.addresses 10.22.132.12 ipv4.method manual
[root@node02 ~]# _


[root@node02 ~]# nmcli connection show
NAME                UUID                                  TYPE      DEVICE
enp7s0 proto dhcp src 192.168.148.33 metric 101
lo kernel scope link src 10.22.122.11 metric 102

```

Figura 2. Procedimiento de configuración de interfaz de red para la red virtual Control.

4.1.2 Instalación y configuración del servicio httpd.

La instalación de Apache se deberá realizar en ambos nodos. Para ello, se ejecuta simultánea en ambos nodos para que el servicio **no** se inicie automáticamente. Ejecutando los siguientes comandos se puede verificar su instalación y configuración inicial (posteriormente se deberá seguir ajustando sus parámetros de configuración), y su desarrollo se encuentra en la **Figura 3:**

```
# dnf install httpd
# systemctl disable httpd
# systemctl start httpd
# curl localhost:80 | head -n 10
# systemctl status httpd
```

```
[root@nodo2 ~]# systemctl disable httpd
[root@nodo2 ~]# systemctl start httpd
[root@nodo2 ~]# curl localhost:80 | head -n 10
  % Total    % Received % Xferd  Average Speed   Time     Time      Current
          Dload  Upload   Total   Spent    Left  Speed
100  8474  100  8474    0      0  897k      0 --:--:-- --:--:-- 919k
<!DOCTYPE html>
<html>
  <head>
    <meta charset='utf-8'>
    <meta name='viewport' content='width=device-width, initial-scale=1'>
    <title>Test Page for the HTTP Server on Fedora</title>
    <style type="text/css">
      /*<![CDATA[*/
      html {
```

[root@nodo2 ~]# systemctl status httpd

- httpd.service - The Apache HTTP Server
 - Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
 - Drop-In: /usr/lib/systemd/system/service.d
 - └─10-timeout-abort.conf
 - Active: active (running) since Tue 2024-04-30 14:41:21 WEST; 45s ago
 - Docs: man:httpd.service(8)
 - Main PID: 1324 (httpd)
 - Status: "Total requests: 1; Idle/Busy workers 100/0;Requests/sec: 0.0256; Bytes served/sec: 225 B/sec"
 - Tasks: 177 (limit: 2305)
 - Memory: 14.7M
 - CPU: 89ms
 - CGroup: /system.slice/httpd.service
 - └─1324 /usr/sbin/httpd -DFOREGROUND
 ├─1325 /usr/sbin/httpd -DFOREGROUND
 ├─1326 /usr/sbin/httpd -DFOREGROUND
 ├─1327 /usr/sbin/httpd -DFOREGROUND
 ├─1328 /usr/sbin/httpd -DFOREGROUND

```
abr 30 14:41:21 nodo2.vpd.com systemd[1]: Starting httpd.service - The Apache HTTP Server...
abr 30 14:41:21 nodo2.vpd.com (httpd)[1324]: httpd.service: Referenced but unset environment variable evalua
abr 30 14:41:21 nodo2.vpd.com httpd[1324]: Server configured, listening on: port 80
abr 30 14:41:21 nodo2.vpd.com systemd[1]: Started httpd.service - The Apache HTTP Server.
```

Figura 3. Instalación y verificación de instalación de Apache.

También se deberá seguir configurando para que Apache se encuentre en el volumen lógico exportado por el nodo Almacenamiento apacheLV, y que es accesible por el nodo1 y el nodo2. Para ello se sigue las siguientes pautas, observable en la **Figura 4**:

- Se activará el volumen lógico con la herramienta **vgchange**, se montará el volumen lógico compartido apacheLV sobre la ruta del servicio *httpd* (es decir, la siguiente dirección: */var/www/*).

```
# vgchange -a y ApacheVG
# lsblk
# mount /dev/ApacheVG/apacheLV /var/www/
```

- Se formará la estructura básica del servicio: script de cgi, la página inicial de tipo html y error.

```
# mkdir /var/www/html /var/www/cgi-bin /var/www/error
# echo "<html><body>configuración correcta</body></html>" >
/var/www/index.html
```

- En cuanto a la configuración de la seguridad, se tiene en cuenta los dos siguientes aspectos:
 - Establecimiento de atributos de contexto SELinux, para que el servicio httpd pueda acceder al volumen lógico sin errores de seguridad.
También se establecieron permisos con seamange manualmente para contextos como httpd_sys_content para el mismo directorio, pero se entendió que ya había sido establecido (ver **Figura 5**).

```
# restorecon -Rv /var/www
# systemctl restart httpd
```

- La configuración del cortafuegos, para que habilite en la zona public el puerto para el servicio http por el puerto 80.

```
# firewall-cmd --zone=public --add-port=80/tcp --permanent
# firewall-cmd --reload
```

```

[root@nodo1 ~]# lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda      8:0    0   1G  0 disk
└─ApacheUG-ApacheLU 253:1   0 800M  0 lvm
sdb      8:16   0   1G  0 disk
sr0     11:0    1 1024M 0 rom
zram0   251:0   0  1,9G 0 disk [SWAP]
vda     252:0   0   10G 0 disk
├─vda1   252:1   0   1M  0 part
└─vda2   252:2   0   1G  0 part /boot
└─vda3   252:3   0   9G  0 part
└─fedora-root 253:0   0   9G  0 lvm /
[root@nodo1 ~]# mount /dev/ApacheUG/ApacheLU /var/www/
[root@nodo1 ~]# mkdir /var/www/html /var/www/cgi-bin /var/www/error
[root@nodo1 ~]# echo "<html><body>Configuración correcta</body></html>" > /var/www/html/index.html
[root@nodo1 ~]# res
reset      resizecons   resizepart   resolvect1   restorecon_xattr
resize2fs   resize.f2fs    resolvconf   restorecon
[root@nodo1 ~]# restorecon --help
restorecon: invalid option -- '-'
usage: restorecon [-iIDFmmpRv0xT] [-e excludedir] pathname...
usage: restorecon [-iIDFmmpRv0xT] [-e excludedir] -f filename
[root@nodo1 ~]# restorecon -Rv /var/www
Relabeled '/var/www' from system_u:object_r:unlabeled_t:s0 to system_u:object_r:httpd_sys_content_t:s0
Relabeled '/var/www/nodel' from unconfined_u:object_r:unlabeled_t:s0 to unconfined_u:object_r:httpd_sys_content_t:s0
Relabeled '/var/www/test1' from unconfined_u:object_r:unlabeled_t:s0 to unconfined_u:object_r:httpd_sys_content_t:s0
Relabeled '/var/www/another' from unconfined_u:object_r:unlabeled_t:s0 to unconfined_u:object_r:httpd_sys_content_t:s0
Relabeled '/var/www/fromnode2_now' from unconfined_u:object_r:unlabeled_t:s0 to unconfined_u:object_r:httpd_sys_content_t:s0
Relabeled '/var/www/fromnode1_now' from unconfined_u:object_r:unlabeled_t:s0 to unconfined_u:object_r:httpd_sys_content_t:s0
Relabeled '/var/www/test2' from unconfined_u:object_r:unlabeled_t:s0 to unconfined_u:object_r:httpd_sys_content_t:s0
Relabeled '/var/www/html' from unconfined_u:object_r:unlabeled_t:s0 to unconfined_u:object_r:httpd_sys_content_t:s0
Relabeled '/var/www/html/index.html' from unconfined_u:object_r:unlabeled_t:s0 to unconfined_u:object_r:httpd_sys_content_t:s0
Relabeled '/var/www/cgi-bin' from unconfined_u:object_r:unlabeled_t:s0 to unconfined_u:object_r:httpd_sys_script_exec_t:s0
Relabeled '/var/www/error' from unconfined_u:object_r:unlabeled_t:s0 to unconfined_u:object_r:httpd_sys_content_t:s0
[root@nodo1 ~]# systemctl start httpd
[root@nodo1 ~]# curl localhost:80
<html><body>Configuración correcta</body></html>
[root@nodo1 ~]# firewall-cmd --zone=public --add-port=80/tcp --permanent
usage: 'firewall-cmd --help' for usage information or see firewall-cmd(1) man page
firewall-cmd: error: unrecognized arguments: --zone=public
[root@nodo1 ~]# firewall-cmd --zone=public --add-port=80/tcp --permanent
success
[root@nodo1 ~]# firewall-cmd --reload
success
[root@nodo1 ~]# firewall-cmd --list-ports
[root@nodo1 ~]# firewall-cmd --zone=public --add-port=80/tcp --permanent
Warning: ALREADY_ENABLED: 80/tcp
success
[root@nodo1 ~]#

```

Figura 4. Configuración **inicial** de Apache.

```

[root@nodo1 ~]# semanage fcontext -a -d -e -f -h -l -m -n -o
[root@nodo1 ~]# semanage fcontext -a -t httpd_sys_content_t "/var/www(/.*)?"
ValueError: El contexto de archivo para /var/www(/.*)? ya está definido
[root@nodo1 ~]# restorecon -Rv /var/www
[root@nodo1 ~]# matchpathcon -U /var/www/
/var/www verified.
[root@nodo1 ~]# _

pcsa@actice:~$ history | grep chcon
root@nodo1 ~]# history | grep chcon
600 chcon -R -t httpd_sys_content_t /var/www/
601 chcon -R -t httpd_config_t /var/www/
602 chcon -R -t httpd_sys_script_exec_t /var/www/
611 chcon -R -t httpd_sys_content_t /var/www/
692 chcon -R -t httpd_sys_content_t /var/www/
693 chcon -R -t httpd_sys_content_t /var/run/httpd/
745 history | grep chcon
749 history | grep chcon
751 history | grep chcon
root@nodo1 ~]# history | grep semanage
394 semanage fcontext -a -t httpd_sys_content_t "/var/www(/.*)?"
547 semanage fcontext -a -t httpd_sys_content_t "/var/www(/.*)?"
556 semanage fcontext -a -t httpd_sys_script_exec_t "/var/www(/.*)?"
557 semanage fcontext -a -t httpd_sys_script_exec_t "/var/www(/.*)?"
558 semanage fcontext -a -t httpd_sys_config_t "/var/www(/.*)?"
563 semanage fcontext -a -t httpd_sys_config_t "/var/www/"
564 semanage fcontext -a -t httpd_sys_config_t "/var/www"
565 semanage fcontext -a -t httpd_sys_config_t "/var/www(/.*)?"
569 semanage fcontext -a -t httpd_config_t "/var/www/"
570 semanage fcontext -a -t httpd_config_t "/var/www"
571 semanage fcontext -a -t httpd_config_t "/var/www(/.*)?"
572 semanage fcontext -a -t httpd_sys_config_t "/var/www(/.*)?"
573 semanage fcontext -a -t httpd_sys_config_t "/var/www/html"
578 semanage fcontext -a -t httpd_sys_config_t "/var/www(/.*)?"
579 semanage fcontext -a -t httpd_sys_config_t "/var/www"
580 semanage fcontext -a -t httpd_config_t "/var/www"
581 semanage fcontext -a -t httpd_config_t "/var/www(/.*)?"
582 semanage fcontext -a -t httpd_sys_script_exec_t "/var/www/cgi-bin(/.*)?"
583 semanage fcontext -a -t httpd_sys_script_exec_t "/var/www/cgi-bin"
584 semanage fcontext -a -t httpd_config_t "/var/www"
585 semanage fcontext -a -t httpd_config_t "/var/www"
586 semanage fcontext -a -t httpd_config_t "/var/www(/.*)?"
587 semanage fcontext -a -t httpd_config_t "/var/www/html(/.*)?"
588 semanage fcontext -a -t httpd_config_t "/var/www/html"
589 semanage fcontext -a -t httpd_config_t "/var/www"
595 semanage fcontext -a -t httpd_config_t "/var/www/html(/.*)?"
606 semanage fcontext -a -t httpd_sys_script_exec_t "/var/www(/.*)?"
607 semanage fcontext -a -t httpd_sys_content_t "/var/www(/.*)?"
608 semanage fcontext -a -t httpd_config_t "/var/www(/.*)?"
752 history | grep semanage
root@nodo1 ~]# _

[root@nodo1 ~]# ls -Zs /var/www/
total 0
0 unconfined_u:object_r:httpd_sys_script_exec_t:s0 cgi-bin 0 unconfined_u:object_r:httpd_sys_script_exec_t:s0 html
0 unconfined_u:object_r:httpd_sys_script_exec_t:s0 error
[root@nodo1 ~]# _

```

Figura 5. Establecimiento de contextos para Apache en SELinux.

Finalmente, desde el nodo Almacenamiento se conecta al servicio de Apache del nodo1 y se verifica su correcta configuración en la **Figura 6**.

```

curl: (7) Failed to connect to 10.22.122.11 port 80 after 0 ms: Couldn't connect to server
root@almacenamiento ~]# curl http://10.22.122.11:80
curl: (7) Failed to connect to 10.22.122.11 port 80 after 1 ms: Couldn't connect to server
root@almacenamiento ~]# curl http://10.22.122.11:80
<html><body>Configuración correcta</body></html>
root@almacenamiento ~]# curl http://nodo1.vpd.com
<html><body>Configuración correcta</body></html>
root@almacenamiento ~]# _
```

```

root@almacenamiento ~]# curl http://10.22.122.11:80
<html><body>Configuración correcta</body></html>
root@almacenamiento ~]# curl http://nodo1.vpd.com
<html><body>Configuración correcta</body></html>
root@almacenamiento ~]# curl http://nodo2.vpd.com
curl: (7) Failed to connect to nodo2.vpd.com port 80 after 1 ms: Couldn't connect to server
root@almacenamiento ~]# curl http://nodo2.vpd.com
<html><body>Configuración correcta</body></html>
root@almacenamiento ~]# _
```

The browser window shows the URL 10.22.122.12/. The page content is "Configuración correcta". Below the browser window, the status bar shows "Most Visited" and "Fedora Docs, Fedora Magazine".

Figura 6. Verificación de conectividad de Apache desde el nodo Almacenamiento, tanto desplegado Apache tanto por nodo1 como por nodo2.

Llegados a este punto, ahora se ha de configurar para que el servicio httpd se ejecute en un clúster de alta disponibilidad.

Paso 1. Configurar los nombres de dominio, tal y como se observa en la **Figura 7**, necesario para los servicios de administración del clúster. Habría que recalcar la importancia en la que se conectan los nodos: el nodo almacenamiento de comunicación con el nodo1 y nodo2 por la red de Almacenamiento para proporcionar el volumen por iSCSI, mientras que el nodo1 y nodo2 se comunicarán por la red Control para la ejecución del servicio de aislamiento “*fencing o stonith*”, cuya configuración del cortafuegos y de instalación del *fencing* no se contempla en el informe por las indicaciones proporcionadas y descriptas en el **apartado 3**.

```

root@almacenamiento ~]# cat /etc/hosts
Loopback entries: do not change.
For historical reasons, localhost precedes localhost.localdomain:
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
See hosts(5) for proper format and other examples:
192.168.1.10 foo.mydomain.org foo
192.168.1.13 bar.mydomain.org bar
10.22.122.18 almacenamiento.vpd.com
10.22.122.11 nodo1.vpd.com
10.22.122.12 nodo2.vpd.com
root@almacenamiento ~]# _
```

```

root@nodo1 ~]# cat /etc/hosts
# Loopback entries: do not change.
# For historical reasons, localhost.localdomain precedes localhost:
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
# See hosts(5) for proper format and other examples:
# 192.168.1.10 foo.mydomain.org foo
# 192.168.1.13 bar.mydomain.org bar
10.22.122.18 almacenamiento.vpd.com
10.22.132.11 nodo1.vpd.com
10.22.132.12 nodo2.vpd.com
[root@nodo1 ~]# _
```

The virtual machine window title is "Nodo2 en QEMU/KVM". The terminal window shows the contents of /etc/hosts for node2, which is identical to the one on node1. The prompt is "nodo2 login: root".

Figura 7. Configuración de nombres de dominio en /etc/hosts para el nodo Almacenamiento, nodo1 y nodo2.

Paso 2. Añadir las siguientes líneas en /etc/httpd/conf.d/status.conf, para que el administrador del clúster pueda leer la información de estado del servicio httpd, tal y como se explica en [2] y se puede observar en la **Figura 8**:

```
<Location /server-status>
    SetHandler server-status
    Require local
</Location>
```

```
[root@nodo1 ~]# cat /etc/httpd/conf.d/status.conf
<Location /server-status>
    SetHandler server-status
    Require local
</Location>
```

Figura 8. Configuración de /etc/httpd/conf.d/status.conf.

Paso 3. Configurar cada uno de estos nodos para que el control de este servicio lo realice el agente del clúster, tal y como se explica en [2] y se puede observar en la **Figura 9**. Para ello, en el archivo de configuración /etc/logrotate.d/httpd se debe eliminar la siguiente línea:

```
/bin/systemctl reload httpd.service > /dev/null 2>/dev/null ||
true
```

Y se reemplaza por las siguientes líneas:

```
/usr/bin/test -f /var/run/httpd-Website.pid >/dev/null 2>/dev/null &&
&&

/usr/bin/ps -q $(/usr/bin/cat /var/run/httpd-Website.pid) >/dev/null
2>/dev/null &&

/usr/sbin/httpd -f /etc/httpd/conf/httpd.conf -c "PidFile
/var/run/httpd-Website.pid" -k graceful > /dev/null 2>/dev/null ||
true
```

```
[root@nodo1 ~]# cat /etc/logrotate.d/httpd
# Note that logs are not compressed unless "compress" is configured,
# which can be done either here or globally in /etc/logrotate.conf.
/var/log/httpd/*log {
    missingok
    notifempty
    sharedscripts
    delaycompress
    postrotate
        /usr/bin/test -f /var/run/httpd-Website.pid >/dev/null 2>/dev/null &&
        /usr/bin/ps -q $(/usr/bin/cat /var/run/httpd-Website.pid) >/dev/null 2>/dev/null &&
        /usr/sbin/httpd -f /etc/httpd/conf/httpd.conf -c "PidFile /var/run/httpd-Website.pid" -k graceful > /dev/null 2>
    endscript
}
[root@nodo1 ~]# _
```

Figura 9. Configuración de /etc/httpd/conf.d/status.conf.

Paso 4. Para finalizar, se muestra en la **Figura 10**, la configuración del recurso de Apache que se utiliza por el administrador del clúster para proporcionar el servicio de Apache en alta disponibilidad.

Se observa una serie de atributos que corresponde a los ficheros de configuración previamente comentados, por un lado, para leer el estado del servicio y su configuración “configfile=/etc/httpd/conf/httpd.conf”, y por otro lado, para reportar su estado “statusurl=http://127.0.0.1/server-status”.

```
[root@nodo1 ~]# pcs resource config Apache_Script
Resource: Apache_Script (class=ocf provider=heartbeat type=apache)
  Attributes: Apache_Script-instance_attributes
    configfile=/etc/httpd/conf/httpd.conf
    statusurl=http://127.0.0.1/server-status
  Operations:
    monitor: Apache_Script-monitor-interval-10s
      interval=10s timeout=20s
    start: Apache_Script-start-interval-0s
      interval=0s timeout=40s
    stop: Apache_Script-stop-interval-0s
      interval=0s timeout=60s
```

Figura 10. Configuración del recurso de Apache en el administrador del clúster.

4.2 Instalación de un clúster básico y proporcionar un servidor web Apache en alta disponibilidad

4.2.1 Configuración final del clúster resultante.

Paso 1. La configuración de los volúmenes debe ser controlada por el administrador del clúster. Para ello, en cada nodo del clúster se debe desactivar la activación local del volumen lógico compartido por el nodo Almacenamiento. En la **Figura 11** se observa cómo se excluye el volumen compartido del grupo de volúmenes gestionado por el sistema local propio.

```

[root@nodo1 ~]# cat /etc/lvm/lvm.conf | grep volume_list
# Configuration option activation/volume_list.
#      or VG. See tags/hosttags. If any host tags exist but volume_list
volume_list = [ "fedora" ]
# Configuration option activation/auto_activation_volume_list.
# The volume_list setting and the "activation skip" property
#      or VG. See tags/hosttags. If any host tags exist but volume_list
# auto_activation_volume_list = [ "vg1", "vg2/lvol1", "@tag1", "@*" ]
# Configuration option activation/read_only_volume_list.
#      or VG. See tags/hosttags. If any host tags exist but volume_list
# read_only_volume_list = [ "vg1", "vg2/lvol1", "@tag1", "@*" ]
# The rules are the same as those for volume_list.
# The rules are the same as those for auto_activation_volume_list.

[root@nodo1 ~]# vgs --noheadings -o vg_name
ApacheVG
fedora
[root@nodo1 ~]# -

```

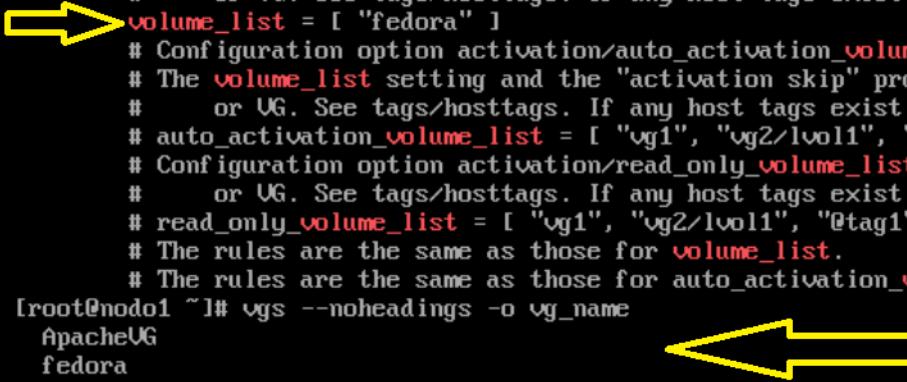


Figura 11. Configuración final de la gestión de volúmenes del clúster .

Paso 2. La configuración del módulo *fencing*, seleccionando manualmente en la máquina anfitriona la interfaz de red donde se conecta a la red de Control del servicio, que será declarada en los recursos de creación del clúster en la red 10.22.132.00/24, observable en la **Figura 12**.

```

he libvirt backend module is designed for single desktops or
servers. Do not use in environments where virtual machines
may be migrated between hosts.

libvirt URI [qemu:///system]: 

configuration complete.

== Begin Configuration ==
ence_virtd {
    listener = "multicast";
    backend = "libvirt";
    module_path = "/usr/lib64/fence-virt/";

listeners {
    multicast {
        key_file = "/etc/cluster/fence_xvm.key";
        address = "225.0.0.12";
        interface = "virbr5";
        family = "ipv4";
        port = "1229";
    }
}

ackends {
    libvirt {
        uri = "qemu:///system";
    }
}

== End Configuration ==
eplace /etc/fence_virt.conf with the above [y/N]? y

```

Figura 12. Configuración final del módulo de aislamiento del clúster .

Paso 3. Configuración final de la infraestructura del clúster, para ello se accede al estado del clúster mediante la herramienta `pcs` ejecutando `pcs status` y se puede observar en la **Figura 13** el resultado del estado de clúster

```
root@nodo1 ~]# pcs status
Cluster name: Apache
Cluster Summary:
  * Stack: corosync (Pacemaker is running)
  * Current DC: nodo1.vpd.com (version 2.1.7-5.fc39-8ee39f2) - partition with quorum
  * Last updated: Thu May 16 09:07:33 2024 on nodo1.vpd.com
  * Last change: Thu May 16 08:56:59 2024 by root via root on nodo2.vpd.com
  * 2 nodes configured
  * 5 resource instances configured

Node List:
  * Online: [ nodo1.vpd.com nodo2.vpd.com ]

Full List of Resources:
  * Resource Group: apachegroup:
    * Apache_LVM          (ocf:heartbeat:LVM):      Started nodo1.vpd.com
    * Apache_FS           (ocf:heartbeat:Filesystem): Started nodo1.vpd.com
    * Apache_IP            (ocf:heartbeat:IPAddr2):   Started nodo1.vpd.com
    * Apache_Script        (ocf:heartbeat:apache):   Started nodo1.vpd.com
    * xvmfence             (stonith:fence_xvm):     Started nodo1.vpd.com

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
```

Figura 13. Configuración final del clúster.

Se aprecia dos nodos configurados y en línea: nodo1 y nodo2. Además, se observa la lista de recursos que contiene:

- Un grupo de recursos *Apachegroup* con el objetivo de que se ejecute todo en el mismo nodo, y se ejecute de manera ordenada.
 - o *Apache_LVM*: activa y usa el grupo del volumen de almacenamiento compartido iSCSI: *ApacheVG*.
 - o *Apache_FS*: usa el sistema de archivos XFS del volumen lógico *ApacheLV* que pertenece al grupo de volumen *ApacheVG* configurado en la anterior práctica, y lo monta en el directorio del servicio de Apache: */var/www/*.
 - o *Apache_IP*: establece la dirección IP flotante 192.168.140.253/24 para el grupo de recursos de forma estática en la red. Mediante esta dirección se desplegará el servicio y estará vinculada al nodo que esté ofreciendo el servicio.
 - o *Apache_Script*: ejecuta el servicio de apache, para ello verifica su estado a través del fichero de configuración */etc/httpd/conf/httpd.conf* y controlará el funcionamiento del servicio.
- El recurso de *xvmfence* con el propósito de garantizar la alta disponibilidad, y proporcionar un mecanismo de control de aislamiento para apagar o reiniciar la actividad del nodo en el clúster.

- Los tres demonios principales de administración del clúster:
 - o Pacemaker: gestor de recursos de alta disponibilidad
 - o Corosync: proporciona la comunicación entre los nodos
 - o pcsd: que no se utiliza, se trata de la interfaz web para interactuar con la configuración del clúster).

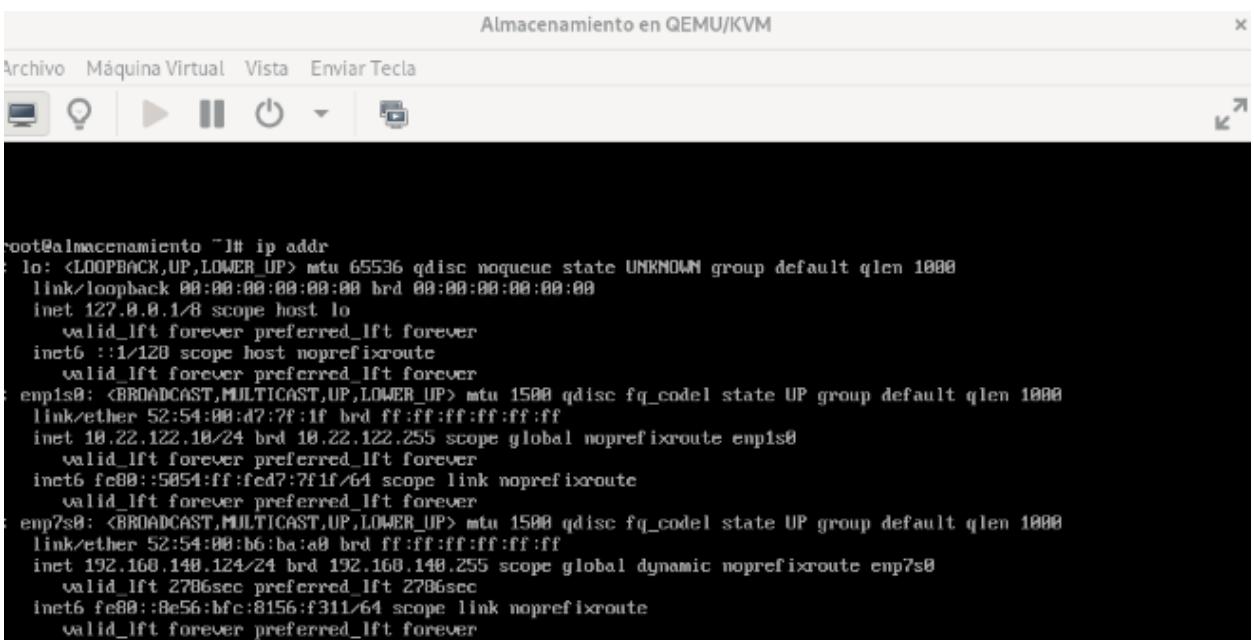
5. Pruebas / Validación

En este apartado se verificará

- La configuración de red en el **apartado 5.1**.
- Validación del servicio y configuración Apache desplegado y del clúster en alta disponibilidad en el **apartado 5.2**.
- El resto de los aspectos fueron verificados implícitamente durante el **apartado 4**.

5.1 Creación y configuración de las interfaces de red.

La verificación de las interfaces se verificará con envíos de trazas ICMP hacia las distintas máquinas virtuales a través del comando **ping**, aunque tras la validación del clúster en el **apartado 4.2.2** implícitamente se entiende que la infraestructura de red debería estar correctamente configurada. De esta manera, en la **Figura 18** se muestra el resultado de conectividad entre los nodos, verificando también sus nombres de dominio para nodo1, nodo2 y Almacenamiento, y sus direcciones IP a través del comando “**ip a**” así como sus tablas de enrutamiento “**ip r**” para el nodo1 y nodo2 se pueden consultar según el nodo en la **Figura 14**, **Figura 15**, **Figura 16**, **Figura 17** y **Figura 18**.



```
Almacenamiento en QEMU/KVM
Archivo Máquina Virtual Vista Enviar Tecla
root@almacenamiento ~# ip addr
: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default 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 noprefixroute
      valid_lft forever preferred_lft forever
: emp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
  link/ether 52:54:00:47:7f:1f brd ff:ff:ff:ff:ff:ff
    inet 10.22.122.10/24 brd 10.22.122.255 scope global noprefixroute emp1s0
      valid_lft forever preferred_lft forever
    inet6 fe80::5054:ff:fed7:7f1f/64 scope link noprefixroute
      valid_lft forever preferred_lft forever
: emp7s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
  link/ether 52:54:00:b6:a0 brd ff:ff:ff:ff:ff:ff
    inet 192.168.148.124/24 brd 192.168.148.255 scope global dynamic noprefixroute emp7s0
      valid_lft 2706sec preferred_lft 2706sec
    inet6 fe80::8e56:bfc:8156:f311/64 scope link noprefixroute
      valid_lft forever preferred_lft forever
```

Figura 14. Configuración de red del nodo de Almacenamiento.

```

conexión activada con éxito (ruta activa b=bus: /org/freedesktop/networkmanager/activeconnection/1)
[root@nodo2 ~]# ip r
default via 192.168.140.1 dev emp7s0 proto dhcp src 192.168.140.127 metric 101
10.22.122.0/24 dev emp1s0 proto kernel scope link src 10.22.122.12 metric 102
10.22.132.0/24 dev emp8s0 proto kernel scope link src 10.22.132.12 metric 103
192.168.140.0/24 dev emp7s0 proto kernel scope link src 192.168.140.127 metric 101
[root@nodo2 ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default 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 noprefixroute
        valid_lft forever preferred_lft forever
2: emp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:5f:4e:d7 brd ff:ff:ff:ff:ff:ff
    inet 10.22.122.12/24 brd 10.22.122.255 scope global noprefixroute emp1s0
        valid_lft forever preferred_lft forever
    inet6 fe80::5054:ff:fe5f:4ed7%254 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: emp7s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:2f:a1:29 brd ff:ff:ff:ff:ff:ff
    inet 192.168.140.127/24 brd 192.168.140.255 scope global dynamic noprefixroute emp7s0
        valid_lft 3297sec preferred_lft 3297sec
    inet6 fe80::d8df:9d59:3ff2:2ced%254 scope link noprefixroute
        valid_lft forever preferred_lft forever
4: emp8s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:e6:2f:8e brd ff:ff:ff:ff:ff:ff
    inet 10.22.132.12/24 brd 10.22.132.255 scope global noprefixroute emp8s0
        valid_lft forever preferred_lft forever
    inet6 fe80::959b:1d49:7887:6464%254 scope link noprefixroute
        valid_lft forever preferred_lft forever
[...]
[root@nodo1 ~]# ip r
default via 192.168.140.1 dev emp7s0 proto dhcp src 192.168.140.33 metric 101
10.22.122.0/24 dev emp1s0 proto kernel scope link src 10.22.122.11 metric 102
10.22.132.0/24 dev emp8s0 proto kernel scope link src 10.22.132.11 metric 103
192.168.140.0/24 dev emp7s0 proto kernel scope link src 192.168.140.33 metric 101
[root@nodo1 ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default 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 noprefixroute
        valid_lft forever preferred_lft forever
2: emp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:e8:b6:66 brd ff:ff:ff:ff:ff:ff
    inet 10.22.122.11/24 brd 10.22.122.255 scope global noprefixroute emp1s0
        valid_lft forever preferred_lft forever
    inet6 fe80::5054:ff:fee8:b666%254 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: emp7s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:38:60:b1 brd ff:ff:ff:ff:ff:ff
    inet 192.168.140.33/24 brd 192.168.140.255 scope global dynamic noprefixroute emp7s0
        valid_lft 3021sec preferred_lft 3021sec
    inet6 fe80::b17e:75bc:536:b9b9%254 scope link noprefixroute
        valid_lft forever preferred_lft forever
4: emp8s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:bc:78:3e brd ff:ff:ff:ff:ff:ff
    inet 10.22.132.11/24 brd 10.22.132.255 scope global noprefixroute emp8s0
        valid_lft forever preferred_lft forever
    inet6 fe80::e912:3788:59a4:da6f%254 scope link noprefixroute
        valid_lft forever preferred_lft forever
[root@nodo1 ~]# ping 10.22.132.12
PING 10.22.132.12 (10.22.132.12) 56(84) bytes of data.
64 bytes from 10.22.132.12: icmp_seq=1 ttl=64 time=0.798 ms
64 bytes from 10.22.132.12: icmp_seq=2 ttl=64 time=0.303 ms
^C

```

Figura 15. Configuración de red del nodo de nodo1 y nodo2.

```

root@almacenamiento ~]# ping 10.22.122.11; ping nodo1.vpd.com
PING 10.22.122.11 (10.22.122.11) 56(84) bytes of data.
4 bytes from 10.22.122.11: icmp_seq=1 ttl=64 time=0.388 ms
4 bytes from 10.22.122.11: icmp_seq=2 ttl=64 time=0.856 ms
C
-- 10.22.122.11 ping statistics ---
packets transmitted, 2 received, 0% packet loss, time 1011ms
rtt min/avg/max/mdev = 0.388/0.622/0.856/0.234 ms
PING nodo1.vpd.com (10.22.122.11) 56(84) bytes of data.
4 bytes from nodo1.vpd.com (10.22.122.11): icmp_seq=1 ttl=64 time=1.15 ms
4 bytes from nodo1.vpd.com (10.22.122.11): icmp_seq=2 ttl=64 time=0.667 ms
C
-- nodo1.vpd.com ping statistics ---
packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.667/0.908/1.149/0.241 ms
root@almacenamiento ~]# ping 10.22.122.12; ping nodo2.vpd.com
PING 10.22.122.12 (10.22.122.12) 56(84) bytes of data.
4 bytes from 10.22.122.12: icmp_seq=1 ttl=64 time=0.403 ms
4 bytes from 10.22.122.12: icmp_seq=2 ttl=64 time=0.808 ms
C
-- 10.22.122.12 ping statistics ---
packets transmitted, 2 received, 0% packet loss, time 1043ms
rtt min/avg/max/mdev = 0.403/0.605/0.808/0.202 ms
PING nodo2.vpd.com (10.22.122.12) 56(84) bytes of data.
4 bytes from nodo2.vpd.com (10.22.122.12): icmp_seq=1 ttl=64 time=0.314 ms
4 bytes from nodo2.vpd.com (10.22.122.12): icmp_seq=2 ttl=64 time=0.809 ms
C
-- nodo2.vpd.com ping statistics ---
packets transmitted, 2 received, 0% packet loss, time 1009ms
rtt min/avg/max/mdev = 0.314/0.561/0.809/0.247 ms
root@almacenamiento ~]# cat /etc/hosts
Loopback entries: do not change.
For historical reasons, localhost precedes localhost.localdomain:
27.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
:1          localhost localhost.localdomain localhost6 localhost6.localdomain6
See hosts(5) for proper format and other examples:
192.168.1.10 foo.mydomain.org foo
192.168.1.13 bar.mydomain.org bar
0.22.122.10 almacenamiento.vpd.com
0.22.122.11 nodo1.vpd.com
0.22.122.12 nodo2.vpd.com
root@almacenamiento ~]# _

```

Figura 16. Verificación de conectividad del nodo Almacenamiento y configuración de del fichero /etc/hosts.

```

[root@nodo1 ~]# ping 10.22.122.10; ping almacenamiento.vpd.com
PING 10.22.122.10 (10.22.122.10) 56(84) bytes of data.
64 bytes from 10.22.122.10: icmp_seq=1 ttl=64 time=0.755 ms

--- 10.22.122.10 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.755/0.755/0.755/0.000 ms
^C
^CPING almacenamiento.vpd.com (10.22.122.10) 56(84) bytes of data.
64 bytes from almacenamiento.vpd.com (10.22.122.10): icmp_seq=1 ttl=64 time=1.48 ms
64 bytes from almacenamiento.vpd.com (10.22.122.10): icmp_seq=2 ttl=64 time=0.423 ms
^C
--- almacenamiento.vpd.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.423/0.952/1.482/0.529 ms
[root@nodo1 ~]# ping 10.22.122.12; ping 10.22.132.12; ping nodo2.vpd.com
PING 10.22.122.12 (10.22.122.12) 56(84) bytes of data.
64 bytes from 10.22.122.12: icmp_seq=1 ttl=64 time=1.79 ms
^C
--- 10.22.122.12 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.789/1.789/1.789/0.000 ms
PING 10.22.132.12 (10.22.132.12) 56(84) bytes of data.
64 bytes from 10.22.132.12: icmp_seq=1 ttl=64 time=0.335 ms
64 bytes from 10.22.132.12: icmp_seq=2 ttl=64 time=0.625 ms
^C
--- 10.22.132.12 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1029ms
rtt min/avg/max/mdev = 0.335/0.480/0.625/0.145 ms
PING nodo2.vpd.com (10.22.132.12) 56(84) bytes of data.
64 bytes from nodo2.vpd.com (10.22.132.12): icmp_seq=1 ttl=64 time=1.38 ms
64 bytes from nodo2.vpd.com (10.22.132.12): icmp_seq=2 ttl=64 time=0.650 ms

--- nodo2.vpd.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.650/1.014/1.378/0.364 ms
^C
[root@nodo1 ~]# cat /etc/hosts
# Loopback entries; do not change.
# For historical reasons, localhost precedes localhost.localdomain:
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
# See hosts(5) for proper format and other examples:
# 192.168.1.10 foo.mydomain.org foo
# 192.168.1.13 bar.mydomain.org bar
10.22.122.10 almacenamiento.vpd.com
10.22.132.11 nodo1.vpd.com
10.22.132.12 nodo2.vpd.com
[root@nodo1 ~]#

```

Figura 17. Verificación de conectividad del nodo nodo1(se omite la del nodo2 por simplicidad, ya que es similar) y configuración de del fichero /etc/hosts .

```

[root@nodo1 ~]# cat /etc/hosts
# Loopback entries; do not change.
# For historical reasons, localhost precedes localhost.localdomain:
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
# See hosts(5) for proper format and other examples:
# 192.168.1.10 foo.mydomain.org foo
# 192.168.1.13 bar.mydomain.org bar
10.22.122.10 almacenamiento.vpd.com
10.22.132.11 nodo1.vpd.com
10.22.132.12 nodo2.vpd.com
[root@nodo1 ~]# ping nodo2.vpd.com
PING nodo2.vpd.com (10.22.122.12) 56(84) bytes of data.
64 bytes from nodo1.vpd.com (10.22.122.11): icmp_seq=1 ttl=64 time=0.695 ms
64 bytes from nodo1.vpd.com (10.22.122.11): icmp_seq=2 ttl=64 time=0.622 ms
^C
--- nodo2.vpd.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1061ms
rtt min/avg/max/mdev = 0.488/0.635/0.798/0.155 ms
[root@almacenamiento ~]#

```

Figura 18. Verificación de conectividad entre los tres nodos: nodo1, nodo2 y Almacenamiento.

5.2 Instalación, configuración final del clúster resultante y validación del servicio en alta disponibilidad.

Para verificar el servicio, debe ser posible acceder al servidor web desde el navegador del host anfitrión accediendo a la dirección pública del servicio configurada en el recurso del clúster 192.168.122.253.

Se verificará qué nodo está dando el servicio en cada momento mediante la orden **pcs status**, la configuración de red en el nodo servidor que deberá tener una dirección IP flotante (192.168.122.253) a través de un comando **ip addr**, y será empleará el mecanismo de *fencing* para inactivar o activar el nodo a través de **pcs node unstandby \$node.vpd.com** y **pcs node standby \$node.vpd.com**.

En la **Figura 19** se aprecia cómo primeramente los recursos se ejecutaban en el nodo2. Posteriormente se desactiva el nodo2 ejecutando **node unstandby node2.vpd.com**, y se activa y se ejecuta los recursos en el nodo1. De esta manera se valida la **alta disponibilidad**.

```
* Current DC: nodo1.vpd.com (version 2.1.7-5.fc39-0ee39f2) - partition with quorum
* Last updated: Thu May 16 09:21:50 2024 on nodo1.vpd.com
* Last change: Thu May 16 09:21:29 2024 by root via root on nodo1.vpd.com
* 2 nodes configured
* 5 resource instances configured

Node List:
  * Node nodo1.vpd.com: standby
  * Online: [ nodo2.vpd.com ]

Full List of Resources:
  * Resource Group: apachegroup:
    * Apache_LUM (ocf:heartbeat:LUM):      Started nodo2.vpd.com
    * Apache_FS (ocf:heartbeat:Filesystem): Started nodo2.vpd.com
    * Apache_IP (ocf:heartbeat:IPaddr2):   Started nodo2.vpd.com
    * Apache_Script (ocf:heartbeat:apache): Started nodo2.vpd.com
    * xvmfence (stonith:fence_xvm):       Started nodo2.vpd.com

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
[root@nodo1 ~]# pcs node standby nodo2.vpd.com; pcs node unstandby nodo1.vpd.com
[root@nodo1 ~]# pcs status
Cluster name: Apache
Cluster Summary:
  * Stack: corosync (Pacemaker is running)
  * Current DC: nodo1.vpd.com (version 2.1.7-5.fc39-0ee39f2) - partition with quorum
  * Last updated: Thu May 16 09:22:28 2024 on nodo1.vpd.com
  * Last change: Thu May 16 09:22:08 2024 by root via root on nodo1.vpd.com
  * 2 nodes configured
  * 5 resource instances configured

Node List:
  * Node nodo2.vpd.com: standby
  * Online: [ nodo1.vpd.com ]

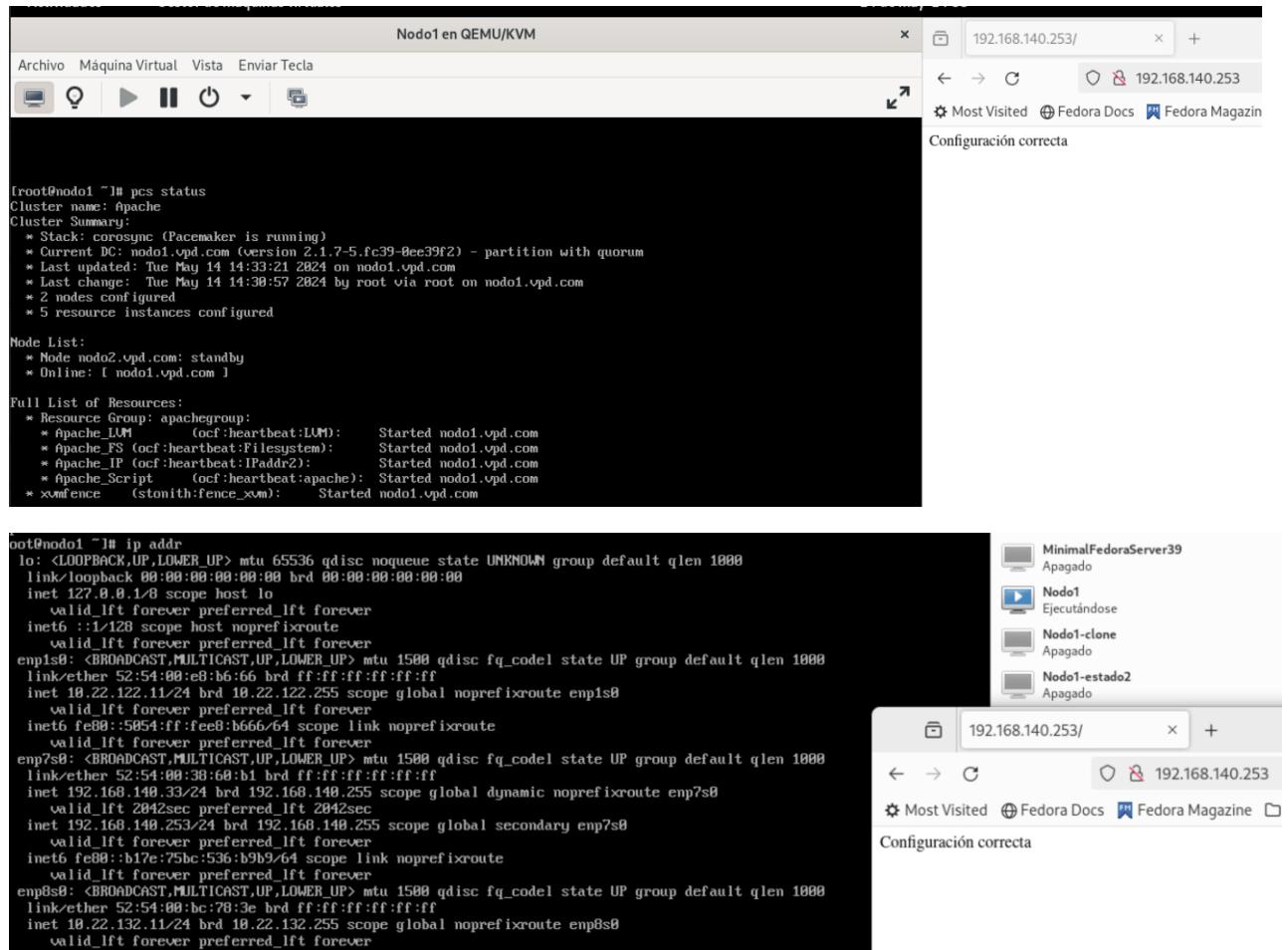
Full List of Resources:
  * Resource Group: apachegroup:
    * Apache_LUM (ocf:heartbeat:LUM):      Started nodo1.vpd.com
    * Apache_FS (ocf:heartbeat:Filesystem): Started nodo1.vpd.com
    * Apache_IP (ocf:heartbeat:IPaddr2):   Started nodo1.vpd.com
    * Apache_Script (ocf:heartbeat:apache): Started nodo1.vpd.com
    * xvmfence (stonith:fence_xvm):       Started nodo1.vpd.com

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
```

Figura 19. Verificación del mecanismo de aislamiento.

Llegados a este punto, se procede en la **Figura 20**, a verificar el correcto funcionamiento del servicio (debe mostrar el script), y, además, que se despliega y se sirve a través de la dirección IP flotante 192.168.122.253.

Para ello, se hace una petición desde el sistema anfitrión a dicha dirección a través del navegador, y se muestra el nodo1 que proporciona el servicio en la interfaz **enp7s0** contiene dos direcciones, 192.168.140.33/24 que es la propia IP de la máquina virtual y 192.168.140.253/24 que es la que proporciona el servicio. También se muestra el estado del clúster para observar qué nodo ejecuta los recursos del clúster, y el navegador web del nodo anfitrión con el resultado del script realizado en la instalación del servidor de Apache y la solicitud a la IP 192.168.140.253/24.



The screenshot shows a KVM session titled "Nodo1 en QEMU/KVM". On the left, a terminal window displays the output of several commands:

```

root@nodo1 ~]# pcs status
Cluster name: Apache
Cluster Summary:
  * Stack: corosync (Pacemaker is running)
  * Current DC: nodo1.vpd.com (version 2.1.7-5.fc39-0ee39f2) - partition with quorum
  * Last updated: Tue May 14 14:33:21 2024 on nodo1.vpd.com
  * Last change: Tue May 14 14:30:57 2024 by root via root on nodo1.vpd.com
  * 2 nodes configured
  * 5 resource instances configured

Node List:
  * Node nodo2.vpd.com: standby
  * Online: [ nodo1.vpd.com ]

Full List of Resources:
  * Resource Group: apachegroup:
    * Apache_LM (ocf:heartbeat:LJM): Started nodo1.vpd.com
    * Apache_FS (ocf:heartbeat:Filesystem): Started nodo1.vpd.com
    * Apache_IP (ocf:heartbeat:IPaddr2): Started nodo1.vpd.com
    * Apache_Script (ocf:heartbeat:apache): Started nodo1.vpd.com
    * xvmmfence (stonith:fence_xvm): Started nodo1.vpd.com

root@nodo1 ~]# ip addr
  
```

On the right, a browser window shows the Apache welcome page at 192.168.140.253/. The address bar also shows the URL. Below the browser are two terminal windows showing the output of the 'ip addr' command. A sidebar on the right lists nodes: MinimalFedoraServer39 (Apagado), Nodo1 (Ejecutándose), Nodo1-clone (Apagado), and Nodo1-estad02 (Apagado).

Figura 20. Verificación de la configuración del clúster y del servicio proporcionado .

De esta manera, se verifica el **correcto funcionamiento del clúster y del servicio proporcionado y su configuración**.

6. Fuentes de información

Referencias

- [1] A. Q. R. García, “Práctica 7: Instalación de un clúster básico para proporcionar un servidor web Apache en alta disponibilidad,” Universidad de Las Palmas de Gran Canaria, 2024.
- [2] A. Q. R. García, “Práctica 7: Diseño y despliegue de la infraestructura de un clúster básico para proporcionar un servicio en alta disponibilidad,” Universidad de Las Palmas de Gran Canaria, 2024.
- [3] R. Saive, “How to Configure Network Connection Using ‘nmcli’ Tool,” TecMint, 11 Mayo 2020. [Online]. Available: <https://www.tecmint.com/nmcli-configure-network-connection/>.