



# PRÁCTICA 5. INFRAESTRUCTURA DE RED VIRTUAL

## Virtualización y Procesamiento Distribuido

El objetivo de esta actividad es conocer los diferentes tipos de redes en entornos de virtualización y saber configurarlas

Francisco Javier López-Dufour Morales

27-04-2025

## Tabla de contenido

<b>Introducción .....</b>	<b>2</b>
<b>Desarrollo.....</b>	<b>3</b>
<b>Preparación inicial .....</b>	<b>3</b>
Creación de la Máquina Virtual mvp5 .....	3
Configuración de la Consola Serie .....	5
<b>Tarea 1: Creación de una Red de Tipo NAT .....</b>	<b>8</b>
<b>Tarea 2: Añadir la Primera Interfaz de Red .....</b>	<b>11</b>
<b>Tarea 3: Creación de una Red Aislada .....</b>	<b>15</b>
<b>Tarea 4: Añadir la Segunda Interfaz de Red .....</b>	<b>17</b>
<b>Tarea 5: Creación de una Tercera Interfaz de Red de Tipo Bridge.....</b>	<b>21</b>
<b>Validaciones .....</b>	<b>28</b>
Interfaces de red.....	28
Reglas de firewall.....	29
Redes virtuales .....	31
<b>Conclusiones.....</b>	<b>34</b>
<b>Bibliografía.....</b>	<b>35</b>

## Introducción

El objetivo fundamental de esta práctica es conocer los diferentes tipos de redes en entornos de virtualización y saber configurarlas. Libvirt usa el concepto de *switch virtual*, un componente software que opera en el anfitrión, al que se conectan las máquinas virtuales (MVs). El tráfico de red de las MVs es gobernado por este switch.

En Linux, el sistema anfitrión representa el switch virtual mediante una interfaz de red. Cuando el demonio `libvirtd` está activo, la interfaz de red por defecto que representa el switch virtual es `virbr0`. Esta interfaz se puede visualizar en el anfitrión mediante la orden `ip addr show`.

Por defecto, los switches virtuales operan en modo NAT (*Network Address Translation*). Sin embargo, también se pueden configurar en modo "Red Enrutada" y en modo "Red Aislada". Adicionalmente, es posible configurar una interfaz de red de la máquina virtual para que esté asociada a una interfaz de tipo bridge del anfitrión. En esta práctica se crearán y configurarán diferentes tipos de redes para comprender sus características y aplicaciones.

## Desarrollo

Para abordar esta práctica se debe haber completado la práctica 1 (Instalación de KVM. Creación e instalación de máquinas virtuales), ya que antes de comenzar con la configuración específica de las redes, es necesario preparar una máquina virtual (mvp5) clonada de mvp1 y configurar el acceso por consola serie. Esto es crucial ya que la interfaz de red inicial será eliminada.

## Preparación inicial

### Creación de la Máquina Virtual mvp5

Se clona la máquina virtual mvp1 para crear mvp5. Posteriormente, se elimina la interfaz de red por defecto para reconfigurarla según los requerimientos de esta práctica.

1. **Verificar las máquinas virtuales existentes:** Antes de clonar, se listan las MVs disponibles para confirmar el estado actual.

```
root@lq-d25:~# virsh list --all
Id      Nombre                               Estado
-----
1       mvp5                                 ejecutando
-       clon_copiando_ficheros              apagado
-       clon_virt_clone                     apagado
-       clon_virt_manager                   apagado
-       Creacion_virt_install               apagado
-       mvp1                                apagado
-       mvp3                                apagado
-       mvp4_lqd25                          apagado
```

Explicación del comando:

- `virsh list --all`: Muestra todas las máquinas virtuales definidas en el sistema, tanto las que están en ejecución como las apagadas.

2. **Clonar la máquina virtual mvp1 para crear mvp5:** Se utiliza `virt-clone` especificando la MAC y la ruta del nuevo disco.

```
root@lq-d25:~# virt-clone --original mvp1 --name mvp5 --file
/var/lib/libvirt/images/mvp5.qcow2 --mac=00:16:3e:37:a0:05
Allocating 'mvp5.qcow2'
2.0 GB  00:07 ...

El clon 'mvp5' ha sido creado exitosamente.
```

Explicación del comando:

- `virt-clone`: Herramienta para clonar máquinas virtuales existentes.
- `--original mvp1`: Especifica la máquina virtual de origen.
- `--name mvp5`: Define el nombre de la nueva máquina virtual.

- `--file /var/lib/libvirt/images/mvp5.qcow2`: Especifica la ruta y nombre del archivo de imagen para el nuevo disco virtual.
- `--mac=00:16:3e:37:a0:05`: Establece una dirección MAC específica y única para la interfaz de red del clon. **Es importante asignar una MAC diferente a la original para evitar conflictos en la red.**

### 3. Iniciar la máquina virtual mvp5: Se arranca la MV recién clonada.

```
root@lq-d25:~# virsh start mvp5
Se ha iniciado el dominio mvp5
```

Explicación del comando:

- `virsh start mvp5`: Inicia la máquina virtual especificada (mvp5).

### 4. Verificar la dirección IP y la interfaz de red inicial: Se comprueba la configuración de red asignada por defecto tras el clonado.

```
root@lq-d25:~# virsh domifaddr mvp5
```

Nombre	dirección MAC	Protocol	Address
vnet0	00:16:3e:37:a0:05	ipv4	192.168.122.124/24

Explicación del comando:

- `virsh domifaddr mvp5`: Muestra las direcciones de las interfaces de red asociadas a la máquina virtual mvp5. Se observa la interfaz vnet0 con la MAC especificada y una IP en la red default (192.168.122.0/24).

```
root@lq-d25:~# virsh domiflist mvp5
```

Interfaz	Tipo	Fuente	Modelo	MAC
vnet0	network	default	virtio	00:16:3e:37:a0:05

Explicación del comando:

- `virsh domiflist mvp5`: Lista las interfaces de red virtuales (vnetX) asociadas a la máquina virtual mvp5, indicando su tipo, fuente (red virtual), modelo y dirección MAC.

### 5. Eliminar la interfaz de red por defecto: Se elimina la interfaz de red tanto de la configuración en ejecución como de la configuración persistente.

```
root@lq-d25:~# virsh detach-interface mvp5 network --mac
00:16:3e:37:a0:05
La interfaz ha sido desmontada exitosamente

root@lq-d25:~# virsh detach-interface mvp5 network --mac
00:16:3e:37:a0:05 --config
La interfaz ha sido desmontada exitosamente
```

Explicación del comando:

- `virsh detach-interface mvp5 network --mac <MAC>`: Desconecta una interfaz de red de la máquina virtual mvp5 identificada por su dirección MAC.
- `--config`: Aplica el cambio a la configuración persistente de la máquina virtual, asegurando que la interfaz no se vuelva a crear al reiniciar.

## 6. Verificar la eliminación de la interfaz:

```
root@lq-d25:~# virsh domiflist mvp5
Interfaz  Tipo  Fuente  Modelo  MAC
-----
```

La salida vacía confirma que mvp5 ya no tiene interfaces de red asociadas.

**Nota:** Al eliminar la interfaz de red, la máquina virtual quedará sin conectividad. Para acceder a ella y continuar con la configuración, es imprescindible configurar la consola serie en la siguiente tarea.

## Configuración de la Consola Serie

Para poder acceder a la máquina virtual sin interfaz de red, se configura una consola serie. Este acceso es fundamental para las tareas posteriores de configuración de red.

1. **Acceder a la máquina virtual mvp5:** Se utiliza `virt-manager` para iniciar sesión en la MV.

```
root@lq-d25:~# virt-manager
```

2. **Editar el archivo de configuración de GRUB:** Dentro de mvp5, se modifica el archivo `/etc/default/grub` para habilitar la consola serie en el arranque.

```
root@mvp5:~# vi /etc/default/grub
```

Se localiza la línea GRUB\_CMDLINE\_LINUX y se añade console=ttyS0. El resultado debería ser similar a:

```
GRUB_CMDLINE_LINUX="console=ttyS0 rd.lvm.lv=fedora/root rhgb quiet"
```

Explicación:

- console=ttyS0: Indica al *kernel* de Linux que utilice el primer puerto serie (ttyS0) como una consola del sistema, permitiendo el envío y recepción de mensajes del kernel y el login a través de él.

3. **Eliminar opciones del *kernel* conflictivas** (opcional pero recomendado): Para asegurar que la configuración de la consola serie tenga prioridad, se eliminan opciones previas del kernel almacenadas por GRUB.

```
root@mvp5:~# grub2-editenv - unset kernelopts
```

Explicación del comando:

- grub2-editenv - unset kernelopts: Elimina la variable kernelopts del entorno de GRUB, la cual podría contener parámetros de arranque previos que interfieran con la nueva configuración.

4. **Regenerar la configuración de GRUB:** Se aplica la configuración modificada generando un nuevo archivo grub.cfg.

```
root@mvp5:~# grub2-mkconfig -o /boot/grub2/grub.cfg
Generating grub configuration file ...
# ...
done
```

Explicación del comando:

- grub2-mkconfig -o /boot/grub2/grub.cfg: Genera el archivo de configuración principal de GRUB (/boot/grub2/grub.cfg) basándose en las plantillas y la configuración de /etc/default/grub.

5. **Reiniciar la máquina virtual:** Se reinicia mvp5 para que arranque con los nuevos parámetros del *kernel*.

```
root@mvp5:~# reboot
```

6. **Probar la conexión a la consola serie desde el anfitrión:** Una vez reiniciada mvp5, se intenta la conexión desde el anfitrión lq-d25.

```
root@lq-d25:~# virsh console mvp5
Connected to domain 'mvp5'
Escape character is ^] (Ctrl + )]
```

```
# ...
mvp5 login: root
Contraseña: <introducir_contraseña>
Last login: Thu Apr  3 19:31:14 on tty1
[root@mvp5 ~]#
```

Explicación del comando:

- `virsh console mvp5`: Establece una conexión con la consola serie de la máquina virtual mvp5.
- El prompt de login (mvp5 login:) confirma que la configuración es exitosa.
- Para salir de la consola serie, se utiliza la secuencia de escape `Ctrl + ]`

Ahora se dispone de acceso a la máquina virtual mvp5 a través de la consola serie, lo que permitirá configurar las interfaces de red en las siguientes tareas.



## Tarea 1: Creación de una Red de Tipo NAT

En esta tarea se crea una red virtual de tipo NAT llamada "Cluster". Esta red permitirá a las máquinas virtuales conectadas a ella comunicarse entre sí, con el anfitrión, y acceder a redes externas (como Internet) a través del anfitrión, que realizará NAT.

**Nota:** Aunque la ficha original de la práctica puede sugerir el uso de virt-manager, en esta documentación se utiliza la línea de comandos (virsh y archivos de definición XML) por su reproducibilidad y facilidad de documentación, siguiendo las buenas prácticas académicas.

1. **Verificar las redes virtuales existentes:** Se comprueba qué redes están ya definidas en el sistema.

```
root@lq-d25:~# virsh net-list --all
Nombre      Estado      Inicio automático  Persistente
-----
default     activo      si                  si
```

Explicación del comando:

- `virsh net-list --all`: Muestra todas las redes virtuales definidas en libvirt, indicando su estado (activa/inactiva), si arrancan automáticamente con el sistema y si su definición es persistente.
2. **Crear un archivo XML para definir la red "Cluster":** Se crea un archivo (cluster-network.xml) que describe la configuración de la nueva red.

```
root@lq-d25:~# cat > cluster-network.xml << EOF
<network>
  <name>Cluster</name>
  <forward mode='nat' />
  <bridge name='virbr1' stp='on' delay='0' />
  <ip address='192.168.140.1' netmask='255.255.255.0'>
    <dhcp>
      <range start='192.168.140.2' end='192.168.140.149' />
    </dhcp>
  </ip>
</network>
EOF
```

Explicación del archivo XML:

- `<network>`: Elemento raíz para la definición de una red virtual.
- `<name>Cluster</name>`: Define el nombre de la red como "Cluster".
- `<forward mode='nat' />`: **Configura el modo de reenvío como NAT.** Esto indica que el tráfico de las MVs hacia redes externas será traducido (NAT) por el anfitrión.

- `<bridge name='virbr1' stp='on' delay='0'/>`: Define un `_switch virtual_` implementado como un bridge de Linux llamado `virbr1`. `stp='on'` habilita el protocolo Spanning Tree para prevenir bucles, y `delay='0'` establece el tiempo de espera del bridge.
- `<ip address='192.168.140.1' netmask='255.255.255.0'/>`: Asigna la dirección IP 192.168.140.1 con máscara 255.255.255.0 (equivalente a /24) a la interfaz del bridge (`virbr1`) en el anfitrión. Esta IP actuará como puerta de enlace para las MVs conectadas.
- `<dhcp>`: Habilita el servicio DHCP integrado de libvirt (`dnsmasq`) para esta red.
- `<range start='192.168.140.2' end='192.168.140.149'/>`: Define el rango de direcciones IP (192.168.140.2 a 192.168.140.149) que el servidor DHCP podrá asignar a las MVs que se conecten a esta red.

3. **Definir la red a partir del archivo XML:** Se carga la definición de la red en libvirt.

```
root@lq-d25:~# virsh net-define cluster-network.xml
La red Cluster se encuentra definida desde cluster-network.xml
```

Explicación del comando:

- `virsh net-define cluster-network.xml`: Lee el archivo XML especificado y crea una definición persistente para la red "Cluster" en la configuración de libvirt. La red aún no está activa.

4. **Iniciar la red:** Se activa la red "Cluster", creando el bridge `virbr1` y iniciando el proceso `dnsmasq` asociado.

```
root@lq-d25:~# virsh net-start Cluster
La red Cluster se ha iniciado
```

Explicación del comando:

- `virsh net-start Cluster`: Inicia la red virtual especificada.

5. **Configurar la red para inicio automático:** Se asegura que la red "Cluster" se inicie automáticamente cuando el servicio `libvirtd` arranque.

```
root@lq-d25:~# virsh net-autostart Cluster
La red Cluster ha sido marcada para iniciarse automáticamente
```

Explicación del comando:

- `virsh net-autostart Cluster`: Marca la red especificada para que se inicie automáticamente con el sistema.

6. **Verificar la creación y estado de la red:** Se confirma que la red "Cluster" aparece en la lista, está activa y configurada para autoarranque.

```

root@lq-d25:~# virsh net-list --all
Nombre      Estado    Inicio automático    Persistente
-----
Cluster     activo   si                    si
default     activo   si                    si

```

7. **Ver los detalles de la red:** Se muestra información específica sobre la red "Cluster".

```

root@lq-d25:~# virsh net-info Cluster
Nombre:      Cluster
UUID:        7ee051d7-e38d-45ab-a26c-232a51e5162e
Activar:     si
Persistente: si
Autoinicio:  si
Puente:      virbr1

```

Explicación del comando:

- `virsh net-info Cluster`: Proporciona detalles clave de la red, como su UUID, estado y el nombre del bridge asociado (virbr1).

8. **Verificar la configuración del bridge en el sistema anfitrión:** Se utiliza `ip addr` para confirmar que la interfaz virbr1 ha sido creada en el anfitrión y tiene la IP correcta.

```

root@lq-d25:~# ip addr show virbr1
7: virbr1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc
noqueue state DOWN group default qlen 1000
    link/ether 52:54:00:f5:97:55 brd ff:ff:ff:ff:ff:ff
    inet 192.168.140.1/24 brd 192.168.140.255 scope global
virbr1
        valid_lft forever preferred_lft forever

```

Explicación de la salida:

- Se muestra la interfaz virbr1 con la dirección MAC asignada por libvirt.
- Tiene la dirección IP 192.168.140.1/24 configurada, como se definió en el XML.
- El estado DOWN y NO-CARRIER es normal cuando no hay ninguna MV conectada todavía al bridge. Cambiará a UP cuando se conecte la primera MV.

Se ha creado exitosamente la red NAT "Cluster", que ahora está lista para ser utilizada por la máquina virtual mvp5.

## Tarea 2: Añadir la Primera Interfaz de Red

En esta tarea se añade una interfaz de red a la máquina virtual mvp5 para conectarla a la red NAT "Cluster" creada en la tarea anterior. Se configurará para obtener una dirección IP automáticamente vía DHCP.

1. **Añadir la interfaz de red a mvp5:** Se utiliza `virsh attach-interface` para conectar la MV a la red "Cluster".

```
root@lq-d25:~# virsh attach-interface mvp5 network Cluster --  
model virtio --config  
La interfaz ha sido asociada exitosamente
```

Explicación del comando:

- `virsh attach-interface mvp5`: Comando para añadir una interfaz de red a la MV mvp5.
  - `network Cluster`: Especifica que la interfaz debe conectarse a la red virtual Cluster. El tipo `network` indica conexión a una red gestionada por libvirt.
  - `--model virtio`: **Establece el modelo del controlador de red como virtio.** Este es un controlador paravirtualizado que ofrece un rendimiento significativamente mejor que los emulados al requerir controladores específicos en el sistema operativo invitado.
  - `--config`: Hace que la adición de la interfaz sea persistente en la configuración de la MV.
2. **Verificar la interfaz añadida:** Se lista las interfaces de mvp5 para ver la nueva interfaz (aún sin nombre `vnetX` asignado porque la MV necesita reiniciarse para detectarla activamente).

```
root@lq-d25:~# virsh domiflist mvp5  
Interfaz  Tipo      Fuente      Modelo      MAC  
-----  
-          network  Cluster     virtio       52:54:00:bd:89:a1
```

Nota: Libvirt asigna automáticamente una dirección MAC si no se especifica una. El guión (-) en la columna Interfaz indica que aún no está activa en el host.

3. **Reiniciar la máquina virtual:** Es necesario reiniciar mvp5 para que el sistema operativo invitado detecte y configure la nueva interfaz de red.

```
root@lq-d25:~# virsh reboot mvp5  
El dominio mvp5 está siendo reiniciado
```

4. **Conectarse a la máquina virtual:** Se accede a mvp5 mediante la consola serie configurada previamente.

```
root@lq-d25:~# virsh console mvp5  
Connected to domain 'mvp5'
```

```
Escape character is ^] (Ctrl + )]  
# ... login ...  
[root@mvp5 ~]#
```

5. **Verificar la configuración de red en mvp5 (Comprobación 1):** Dentro de la MV, se utiliza `ip addr` para ver las interfaces y sus direcciones IP.

```
[root@mvp5 ~]# ip addr  
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:bd:89:a1 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.140.17/24 brd 192.168.140.255 scope global  
dynamic noprefixroute enp1s0  
        valid_lft 3590sec preferred_lft 3590sec  
    inet6 fe80::5054:ff:febd:89a1/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever
```

Explicación de la salida:

- La nueva interfaz es detectada por el sistema operativo como **enp1s0** (el nombre puede variar).
  - Tiene la dirección MAC **52:54:00:bd:89:a1** asignada por libvirt.
  - **Ha obtenido automáticamente la dirección IP 192.168.140.17/24** del servidor DHCP de la red "Cluster" (192.168.140.1). La palabra *dynamic* confirma que fue obtenida por DHCP.
  - El estado es UP, indicando que la interfaz está activa y funcional.
  - El tiempo de validez (*valid\_lft*) indica la duración de la concesión DHCP.
6. **Configurar el archivo /etc/hosts en el anfitrión:** Para facilitar el acceso por nombre, se añade una entrada en el archivo `/etc/hosts` del anfitrión (lq-d25) que mapea la IP de la MV a un nombre de host.

```
root@lq-d25:~# echo "192.168.140.17 mvp5i1.vpd.com mvp5i1" >>  
/etc/hosts  
  
root@lq-d25:~# cat /etc/hosts | grep mvp5i1  
192.168.140.17 mvp5i1.vpd.com mvp5i1
```

Explicación:

- Se añade una línea al archivo `/etc/hosts` asociando la IP 192.168.140.17 con el FQDN `mvp5i1.vpd.com` y el alias `mvp5i1`. Esto

permite usar estos nombres en lugar de la IP para referirse a la primera interfaz de mvp5.

7. **Verificar la conectividad desde el anfitrión a la MV (Comprobación 2):** Se utiliza ping desde el anfitrión hacia el nombre configurado para la MV.

```
root@lq-d25:~# ping -c 4 mvp5i1.vpd.com
PING mvp5i1.vpd.com (192.168.140.17) 56(84) bytes of data.
 64 bytes from mvp5i1.vpd.com (192.168.140.17): icmp_seq=1
ttl=64 time=0.280 ms
 64 bytes from mvp5i1.vpd.com (192.168.140.17): icmp_seq=2
ttl=64 time=0.437 ms
 64 bytes from mvp5i1.vpd.com (192.168.140.17): icmp_seq=3
ttl=64 time=0.242 ms
 64 bytes from mvp5i1.vpd.com (192.168.140.17): icmp_seq=4
ttl=64 time=0.226 ms

--- mvp5i1.vpd.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3092ms
rtt min/avg/max/mdev = 0.226/0.296/0.437/0.083 ms
```

Resultado: La MV responde correctamente a los pings (0% *packet loss*), confirmando la conectividad básica entre el anfitrión y la MV a través de la red "Cluster".

8. **Verificar el acceso a Internet desde la MV (Comprobación 3):** Dentro de mvp5, se intenta hacer ping a un destino externo.

```
[root@mvp5 ~]# ping -c 4 google.es
PING google.es (142.250.184.163) 56(84) bytes of data.
 64 bytes from mad07s23-in-f3.1e100.net (142.250.184.163):
icmp_seq=1 ttl=114 time=29.7 ms
# ... (más respuestas) ...

--- google.es ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 29.738/30.110/30.441/0.267 ms
```

Resultado: La MV puede alcanzar google.es, confirmando que la ruta usada es vía 192.168.140.1.

9. **Verificar la configuración de red completa en la MV:** Se revisan la tabla de rutas y la configuración de DNS.

```
[root@mvp5 ~]# ip route
default via 192.168.140.1 dev enp1s0 proto dhcp src
192.168.140.17 metric 100
192.168.140.0/24 dev enp1s0 proto kernel scope link src
192.168.140.17 metric 100
```

Explicación de la tabla de rutas:

- `default via 192.168.140.1 dev enp1s0`: La ruta por defecto (para cualquier destino no local) apunta a 192.168.140.1 (la IP del bridge virbr1 en el anfitrión) a través de la interfaz enp1s0. Fue configurada vía DHCP (proto dhcp).
- `192.168.140.0/24 dev enp1s0`: Ruta para la red local 192.168.140.0/24, accesible directamente a través de enp1s0.
- `metric 100`: La métrica indica la prioridad de la ruta (menor es más preferible).

Estas comprobaciones confirman que la primera interfaz de red se ha configurado correctamente, conectada a la red NAT "Cluster", obteniendo IP por DHCP y con acceso tanto al anfitrión como a Internet.

## Tarea 3: Creación de una Red Aislada

En esta tarea se crea una segunda red virtual llamada "Almacenamiento". A diferencia de la red "Cluster", esta será una **red aislada**. Las máquinas virtuales conectadas a ella podrán comunicarse entre sí y con el anfitrión, pero **no tendrán conectividad con redes externas** (como Internet) a través de esta red. Además, **no se configurará un servidor DHCP** para ella; las IPs deberán asignarse manualmente.

1. **Crear un archivo XML para definir la red "Almacenamiento":** Se crea el archivo `almacenamiento-network.xml`.

```
root@lq-d25:~# cat > almacenamiento-network.xml << EOF
<network>
  <name>Almacenamiento</name>
  <bridge name='virbr2' stp='on' delay='0' />
  <ip address='10.22.122.1' netmask='255.255.255.0'>
  </ip>
</network>
EOF
```

Explicación del archivo XML:

- `<name>Almacenamiento</name>`: Define el nombre de la red.
- `<bridge name='virbr2' stp='on' delay='0' />`: Define un bridge `virbr2` para esta red.
- `<ip address='10.22.122.1' netmask='255.255.255.0'>`: Asigna la IP `10.22.122.1/24` al bridge `virbr2` en el anfitrión.
- **Ausencia de `<forward mode='...' />`**: La omisión de la etiqueta `<forward>` es lo que define esta red como **aislada**. Libvirt no configurará reenvío ni NAT para el tráfico de esta red hacia el exterior.
- **Ausencia de `<dhcp>`**: La omisión de la sección `<dhcp>` indica que no se iniciará un servidor DHCP para esta red.

2. **Definir la red a partir del archivo XML:**

```
root@lq-d25:~# virsh net-define almacenamiento-network.xml
La red Almacenamiento se encuentra definida desde
almacenamiento-network.xml
```

3. **Iniciar la red:**

```
root@lq-d25:~# virsh net-start Almacenamiento
La red Almacenamiento ha sido iniciada
```

4. **Configurar la red para inicio automático:**

```
root@lq-d25:~# virsh net-autostart Almacenamiento
La red Almacenamiento ha sido marcada para autoarranque
```



##### 5. Verificar la creación y estado de la red:

```
root@lq-d25:~# virsh net-list --all
```

Nombre	Estado	Inicio automático	Persistente
Almacenamiento	activo	si	si
Cluster	activo	si	si
default	activo	si	si

La red "Almacenamiento" aparece activa y configurada correctamente.

##### 6. Ver los detalles de la red:

```
root@lq-d25:~# virsh net-info Almacenamiento
```

Nombre:	Almacenamiento
UUID:	5c2735be-366d-4c47-907c-8fbfcf600175
Activar:	si
Persistente:	si
Autoinicio:	si
Puente:	<b>virbr2</b>

Confirma que el bridge asociado es **virbr2**.

##### 7. Verificar la configuración del bridge en el sistema anfitrión:

```
root@lq-d25:~# ip addr show virbr2
```

19: <b>virbr2</b> :	<NO-CARRIER,BROADCAST,MULTICAST,UP>	mtu 1500 qdisc noqueue state DOWN group default qlen 1000
link/ether	52:54:00:fc:4f:fa	brd ff:ff:ff:ff:ff:ff
inet	10.22.122.1/24	brd 10.22.122.255 scope global virbr2
valid_lft	forever	preferred_lft forever

Explicación de la salida:

- La interfaz virbr2 se ha creado con la IP 10.22.122.1/24.
- Al igual que virbr1, su estado inicial es DOWN / NO-CARRIER hasta que se conecte una MV.

Se ha creado correctamente la red aislada "Almacenamiento", lista para conectar la segunda interfaz de mvp5.

## Tarea 4: Añadir la Segunda Interfaz de Red

Se añade una segunda interfaz de red a mvp5, conectándola a la red aislada "Almacenamiento". Dado que esta red no tiene DHCP, la dirección IP deberá configurarse manualmente dentro de la máquina virtual.

1. **Añadir la interfaz de red a mvp5:** Se conecta la MV a la red "Almacenamiento".

```
root@lq-d25:~# virsh attach-interface mvp5 network
Almacenamiento --model virtio --config
La interfaz ha sido asociada exitosamente
```

Explicación del comando:

- Similar a la Tarea 2, pero conectando a network Almacenamiento.

2. **Verificar las interfaces de mvp5:**

```
root@lq-d25:~# virsh domiflist mvp5
Interfaz  Tipo      Fuente      Modelo  MAC
-----
vnet0     network  Cluster     virtio
52:54:00:bd:89:a1
-         network  Almacenamiento virtio
52:54:00:e4:bf:90
```

Ahora mvp5 tiene dos interfaces definidas: una conectada a "Cluster" (ya activa como vnet0) y la nueva conectada a "Almacenamiento" (aún inactiva).

3. **Reiniciar la máquina virtual:** Necesario para que mvp5 detecte la nueva interfaz.

```
root@lq-d25:~# virsh reboot mvp5
El dominio mvp5 está siendo reiniciado
```

4. **Conectarse a la máquina virtual:** Vía consola serie.

```
root@lq-d25:~# virsh console mvp5
# ... login ...
[root@mvp5 ~]#
```

5. **Verificar la detección de la nueva interfaz (Comprobación 1):** Dentro de mvp5, se usa `ip addr`.

```
[root@mvp1 ~]# ip addr
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:bd:89:a1 brd ff:ff:ff:ff:ff:ff
    inet 192.168.140.17/24 brd 192.168.140.255 scope global dynamic
noprefixroute enp1s0
        valid_lft 3474sec preferred_lft 3474sec
        inet6 fe80::5054:ff:febd:89a1/64 scope link noprefixroute
        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:e4:bf:90 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::b6f4:4b5d:b647:cfbf/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

```

Explicación de la salida:

- Se detecta una nueva interfaz, nombrada enp7s0 (el nombre puede variar).
- Tiene la MAC 52:54:00:e4:bf:90 asignada por libvirt.
- Está UP.
- No tiene dirección IPv4 asignada, como era de esperar, ya que la red "Almacenamiento" no tiene DHCP. Solo tiene la dirección IPv6 link-local (fe80::...).

6. **Configurar la dirección IP estática:** Se utiliza **nmcli** (*NetworkManager Command Line Interface*) para configurar la dirección IP estática 10.22.122.2/24 para la interfaz enp7s0.

```

[root@mvp5 ~]# nmcli connection add type ethernet con-name
Almacenamiento ifname enp7s0 ipv4.method manual ipv4.addresses
10.22.122.2/24
Connection 'Almacenamiento' (uuid) successfully added.

```

Explicación del comando:

- **nmcli connection add:** Añade una nueva configuración de conexión de NetworkManager.
- **type ethernet:** Especifica el tipo de conexión.
- **con-name Almacenamiento:** Asigna un nombre descriptivo ("Almacenamiento") a esta configuración de conexión.
- **ifname enp7s0:** Asocia esta configuración a la interfaz física enp7s0.
- **ipv4.method manual:** Establece el método de configuración IPv4 como manual (estático).
- **ipv4.addresses 10.22.122.2/24:** Asigna la dirección IP 10.22.122.2 con la máscara de red /24.

```

[root@mvp5 ~]# nmcli connection up Almacenamiento
Conexión activada con éxito (ruta activa D-Bus:
/org/freedesktop/NetworkManager/ActiveConnection/...)

```

Explicación del comando:

- nmcli connection up Almacenamiento: Activa la conexión de NetworkManager llamada "Almacenamiento", aplicando la configuración IP a la interfaz enp7s0.

## 7. Verificar la configuración de la interfaz enp7s0:

```
[root@mvp5 ~]# 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:e4:bf:90 brd ff:ff:ff:ff:ff:ff
    inet 10.22.122.2/24 brd 10.22.122.255 scope global
noprofixroute enp7s0
    valid_lft forever preferred_lft forever
    inet6 fe80::8046:bb5d:3e14:c0dc/64 scope link noprofixroute
    valid_lft forever preferred_lft forever
```

Resultado: La interfaz enp7s0 ahora tiene la dirección IPv4 estática 10.22.122.2/24 configurada correctamente.

## 8. Configurar el archivo /etc/hosts en el anfitrión: Se añade una entrada para la segunda interfaz de mvp5.

```
root@lq-d25:~# echo "10.22.122.2 mvp5i2.vpd.com mvp5i2" >>
/etc/hosts

root@lq-d25:~# cat /etc/hosts | grep mvp5
192.168.140.17 mvp5i1.vpd.com mvp5i1
10.22.122.2 mvp5i2.vpd.com mvp5i2
```

Ahora se pueden usar mvp5i2.vpd.com o mvp5i2 para referirse a la IP 10.22.122.2.

## 9. Verificar la conectividad desde el anfitrión a la MV (Comprobación 2): Se hace ping desde el anfitrión a la IP de la segunda interfaz usando el nombre definido.

```
root@lq-d25:~# ping -c 4 mvp5i2.vpd.com
PING mvp5i2.vpd.com (10.22.122.2) 56(84) bytes of data.
 64 bytes from mvp5i2.vpd.com (10.22.122.2): icmp_seq=1 ttl=64
time=0.259 ms
# ... (más respuestas) ...

--- mvp5i2.vpd.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3089ms
rtt min/avg/max/mdev = 0.259/0.310/0.324/0.010 ms
```

Resultado: El ping es exitoso, confirmando la conectividad entre el anfitrión y la segunda interfaz de mvp5 a través de la red "Almacenamiento".

10. **Verificar la conectividad desde la MV al anfitrión (Comprobación 3):** Se hace ping desde mvp5 a la IP del bridge virbr2 en el anfitrión.

```
[root@mvp5 ~]# ping -c 4 10.22.122.1
PING 10.22.122.1 (10.22.122.1) 56(84) bytes of data.
64 bytes from 10.22.122.1: icmp_seq=1 ttl=64 time=0.152 ms
# ... (más respuestas) ...

--- 10.22.122.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 0.152/0.243/0.343/0.073 ms
```

Resultado: El ping es exitoso, confirmando la comunicación bidireccional dentro de la red aislada "Almacenamiento".

11. **Intentar acceder a Internet desde la MV (Comprobación 4):** Se verifica que esta interfaz no tiene acceso a redes externas. Se comprueba qué ruta se usaría para alcanzar una IP externa.

```
# Dentro de mvp5
[root@mvp5 ~]# ip route get 8.8.8.8
8.8.8.8 via 192.168.140.1 dev enp1s0 proto dhcp src
192.168.140.17 metric 102 # Ruta vía Cluster
```

Explicación:

- El comando `ip route get 8.8.8.8` consulta la tabla de enrutamiento para determinar cómo se alcanzaría la dirección 8.8.8.8 (Google DNS).
- La salida indica que se utilizaría la interfaz **enp1s0 (conectada a la red NAT "Cluster")** a través de la puerta de enlace 192.168.140.1.
- Esto confirma que la interfaz `enp7s0` (conectada a la red aislada "Almacenamiento") **no se utiliza para el tráfico hacia Internet**, como se esperaba. La red aislada funciona correctamente.

Se ha configurado exitosamente la segunda interfaz de mvp5, conectada a la red aislada "Almacenamiento" con una IP estática, permitiendo la comunicación interna pero no el acceso externo a través de ella.

## Tarea 5: Creación de una Tercera Interfaz de Red de Tipo Bridge

En esta tarea se crea un bridge en el sistema anfitrión y se añade una tercera interfaz de red a la máquina virtual mvp5 conectada a este bridge. Esta configuración permitirá que la máquina virtual esté en la misma red física que el anfitrión, accediendo directamente a la red del laboratorio.

1. **Crear un bridge en el sistema anfitrión:** Se utiliza `nmcli` para crear un bridge llamado `bridge0`.

```
root@lq-d25:~# nmcli con add type bridge con-name bridge0
ifname bridge0
Conexión «bridge0» (a41999da-60ce-46e6-8d81-2dc245c7aeb1)
añadida con éxito.
```

Explicación del comando:

- `nmcli con add type bridge`: Crea una nueva conexión de tipo bridge (puente).
- `con-name bridge0`: Asigna el nombre `bridge0` a la conexión.
- `ifname bridge0`: Especifica que la interfaz física se llamará también `bridge0`.

2. **Verificar el estado de los dispositivos de red:** Se comprueba que el bridge se ha creado correctamente.

```
root@lq-d25:~# nmcli device status
```

DEVICE	TYPE	STATE	CONNECTION
bridge0	bridge	conectado	bridge0
lo	loopback	connected (externally)	lo
virbr0	bridge	connected (externally)	virbr0
virbr1	bridge	connected (externally)	virbr1
virbr2	bridge	connected (externally)	virbr2
vnet18	tun	connected (externally)	vnet18
vnet19	tun	connected (externally)	vnet19
vnet20	tun	connected (externally)	vnet20
enp6s0	ethernet	conectado	enp6s0

Explicación de la salida:

- El dispositivo `bridge0` aparece como un dispositivo de tipo bridge y está en estado conectado.
  - También se muestran los bridges creados anteriormente (`virbr0`, `virbr1` y `virbr2`).
  - La interfaz física `enp6s0` es la interfaz de red del anfitrión que se añadirá al bridge.
3. **Añadir la interfaz física al bridge:** Se conecta la interfaz física `enp6s0` al bridge `bridge0`.

```
root@lq-d25:~# nmcli con mod enp6s0 master bridge0
```

Explicación del comando:

- nmcli con mod enp6s0: Modifica la configuración de la conexión asociada a la interfaz enp6s0.
- master bridge0: Establece que la interfaz será esclava del bridge bridge0.

4. **Verificar la configuración del bridge:** Se confirma que la interfaz física se ha añadido correctamente al bridge.

```
root@lq-d25:~# nmcli con show enp6s0
# ...
connection.master:                bridge0
connection.slave-type:            bridge
# ...
```

Explicación de la salida:

- Se confirma que la conexión enp6s0 ahora tiene como maestro (master) al *bridge* bridge0.
- El tipo de esclavo (slave-type) está configurado como bridge.

5. **Comprobar el estado actualizado de los dispositivos:** Se verifica que la configuración se ha aplicado correctamente.

```
root@lq-d25:~# nmcli device status
```

DEVICE	TYPE	STATE	CONNECTION
bridge0	bridge	conectado	bridge0
lo	loopback	connected (externally)	lo
virbr0	bridge	connected (externally)	virbr0
virbr1	bridge	connected (externally)	virbr1
virbr2	bridge	connected (externally)	virbr2
enp6s0	ethernet	conectado	enp6s0

Resultado: La interfaz enp6s0 y el bridge bridge0 aparecen conectados, lo que indica que el bridge está funcionando correctamente.

6. **Verificar la configuración IP del bridge:** Se comprueba la dirección IP asignada al bridge.

```
root@lq-d25:~# ip addr show bridge0
3: bridge0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
noqueue state UP group default qlen 1000
    link/ether 08:bf:b8:ee:b1:69 brd ff:ff:ff:ff:ff:ff
    inet 10.140.92.125/24 brd 10.140.92.255 scope global
dynamic noprefixroute bridge0
        valid_lft 8501sec preferred_lft 8501sec
    inet6 fe80::e6e7:c28d:331c:1d14/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Explicación de la salida:

- El bridge bridge0 ha heredado la configuración IP de la interfaz física enp6s0.
- Ha obtenido la dirección IP 10.140.92.125/24 a través de DHCP, como indica dynamic.
- El estado del bridge es UP, lo que confirma que está operativo.

7. **Generar una dirección MAC para la nueva interfaz:** Se utiliza un script para generar una dirección MAC válida que se utilizará en la nueva interfaz de la máquina virtual.

```
root@lq-d25:~# python ./macgen.py
00:16:3e:6b:8b:d9
```

Explicación del comando: Esta dirección MAC se utilizará para la tercera interfaz de red de mvp5 para garantizar que sea única en la red.

8. **Añadir la interfaz de red a la máquina virtual:** Se conecta una nueva interfaz a la máquina virtual mvp5 y se configura para usar el bridge.

```
root@lq-d25:~# virsh attach-interface mvp5 bridge bridge0 --
model virtio --mac 00:16:3e:6b:8b:d9 --config
La interfaz ha sido asociada exitosamente
```

Explicación del comando:

- virsh attach-interface mvp5: Añade una interfaz de red a la máquina virtual mvp5.
- bridge bridge0: Especifica que la interfaz debe conectarse al bridge bridge0 del anfitrión.
- --model virtio: Utiliza el controlador de red paravirtualizado virtio para un mejor rendimiento.
- --mac 00:16:3e:6b:8b:d9: Asigna la dirección MAC generada anteriormente a la interfaz.
- --config: Hace que la adición de la interfaz sea persistente en la configuración de la MV.

9. **Verificar las interfaces de la máquina virtual:** Se comprueba que la nueva interfaz se ha añadido correctamente.

```
root@lq-d25:~# virsh domiflist mvp5
Interfaz  Tipo      Fuente      Modelo  MAC
-----
vnet18    network  Cluster    virtio
52:54:00:bd:89:a1
```



vnet19	bridge	bridge0	virtio
00:16:3e:6b:8b:d9			
vnet20	network	Almacenamiento	virtio
52:54:00:e4:bf:90			

Explicación de la salida:

- La máquina virtual mvp5 ahora tiene tres interfaces de red:
  - o vnet18: Conectada a la red NAT "Cluster"
  - o vnet19: Conectada al bridge bridge0 (nueva interfaz)
  - o vnet20: Conectada a la red aislada "Almacenamiento"
- La nueva interfaz tiene la dirección MAC especificada (00:16:3e:6b:8b:d9).

10. **Reiniciar la máquina virtual:** Para aplicar los cambios y que el sistema operativo invitado detecte la nueva interfaz.

```
root@lq-d25:~# virsh reboot mvp5
El dominio mvp5 está siendo reiniciado
```

11. **Conectarse a la máquina virtual:** Mediante la consola serie.

```
root@lq-d25:~# virsh console mvp5
Connected to domain 'mvp5'
Escape character is ^] (Ctrl + )

Fedora Linux 39 (Server Edition)
Kernel 6.5.13-300.fc39.x86_64 on an x86_64

mvp5 login: root
Password:
Last login: Mon Mar 25 14:28:15 on ttyS0
[root@mvp5 ~]#
```

12. **Verificar la configuración de red en la máquina virtual (Comprobación 1):** Se comprueba que la nueva interfaz se ha detectado y configurado correctamente.

```
[root@mvp5 ~]# ip addr
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:bd:89:a1 brd ff:ff:ff:ff:ff:ff
    inet 192.168.140.17/24 brd 192.168.140.255 scope global dynamic noprefixroute enp1s0
```

```

        valid_lft 3174sec preferred_lft 3174sec
        inet6 fe80::5054:ff:febd:89a1/64 scope link noprefixroute
        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:e4:bf:90 brd ff:ff:ff:ff:ff:ff
        inet 10.22.122.2/24 brd 10.22.122.255 scope global
noprefixroute enp7s0
        valid_lft forever preferred_lft forever
        inet6 fe80::8046:bb5d:3e14:c0dc/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
    4: enp8s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
fq_codel state UP group default qlen 1000
        link/ether 00:16:3e:6b:8b:d9 brd ff:ff:ff:ff:ff:ff
        inet 10.140.92.178/24 brd 10.140.92.255 scope global
dynamic noprefixroute enp8s0
        valid_lft 10392sec preferred_lft 10392sec
        inet6 fe80::f0dc:8e70:5631:d16d/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

```

Explicación del resultado:

- La tercera interfaz (enp8s0) ha sido configurada correctamente con la dirección MAC 00:16:3e:6b:8b:d9.
- Ha obtenido la dirección IP 10.140.92.178/24 automáticamente mediante DHCP de la red del laboratorio, como indica dynamic.
- El estado de la interfaz es UP, lo que indica que está funcionando correctamente.
- Al estar en la misma red que el anfitrión (10.140.92.0/24), puede comunicarse directamente con los equipos de la red física.

13. **Verificar los dispositivos de red en el anfitrión:** Se comprueba cómo aparece la interfaz virtual en el sistema anfitrión.

```

root@lq-d25:~# nmcli device status
DEVICE    TYPE        STATE                CONNECTION
bridge0   bridge      conectado            bridge0
lo         loopback    connected (externally) lo
virbr0    bridge      connected (externally) virbr0
virbr1    bridge      connected (externally) virbr1
virbr2    bridge      connected (externally) virbr2
vnet18    tun         connected (externally) vnet18
vnet19    tun         connected (externally) vnet19
vnet20    tun         connected (externally) vnet20
enp6s0    ethernet    conectado            enp6s0

```

14. **Configurar el archivo hosts en el anfitrión:** Se añade una entrada para facilitar el acceso a la tercera interfaz de mvp5.

```

root@lq-d25:~# echo "10.140.92.178 mvp5i3.vpd.com mvp5i3" >>
/etc/hosts

```

```
root@lq-d25:~# cat /etc/hosts | grep mvp5
192.168.140.17 mvp5i1.vpd.com mvp5i1
10.22.122.2 mvp5i2.vpd.com mvp5i2
10.140.92.178 mvp5i3.vpd.com mvp5i3
```

Explicación: se añade una entrada al archivo /etc/hosts para asociar la dirección IP 10.140.92.178 con el nombre mvp5i3.vpd.com y el alias mvp5i3.

- 15. Verificar la conectividad desde el anfitrión a la MV (Comprobación 2):** Se prueba la comunicación con la tercera interfaz.

```
root@lq-d25:~# ping -c 4 mvp5i3.vpd.com
PING mvp5i3.vpd.com (10.140.92.178) 56(84) bytes of data.
 64 bytes from mvp5i3.vpd.com (10.140.92.178): icmp_seq=1 ttl=64
time=0.058 ms
 64 bytes from mvp5i3.vpd.com (10.140.92.178): icmp_seq=2 ttl=64
time=0.068 ms
 64 bytes from mvp5i3.vpd.com (10.140.92.178): icmp_seq=3 ttl=64
time=0.049 ms
 64 bytes from mvp5i3.vpd.com (10.140.92.178): icmp_seq=4 ttl=64
time=0.059 ms

--- mvp5i3.vpd.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3085ms
rtt min/avg/max/mdev = 0.049/0.058/0.068/0.006 ms
```

Resultado: La máquina virtual responde correctamente a los paquetes ICMP enviados desde el anfitrión, lo que confirma la conectividad entre ambos sistemas a través del bridge.

- 16. Verificar el acceso a Internet desde la MV (Comprobación 3):** Se comprueba que la interfaz bridge tiene conectividad externa.

```
[root@mvp5 ~]# ping -c 4 google.es
PING google.es (142.250.184.163) 56(84) bytes of data.
 64 bytes from mad07s23-in-f3.1e100.net (142.250.184.163):
icmp_seq=1 ttl=114 time=30.3 ms
 64 bytes from mad07s23-in-f3.1e100.net (142.250.184.163):
icmp_seq=2 ttl=114 time=30.3 ms
 64 bytes from mad07s23-in-f3.1e100.net (142.250.184.163):
icmp_seq=3 ttl=114 time=29.8 ms
 64 bytes from mad07s23-in-f3.1e100.net (142.250.184.163):
icmp_seq=4 ttl=114 time=29.9 ms

--- google.es ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 29.762/30.058/30.297/0.238 ms
```

Resultado: La máquina virtual puede acceder a sitios de Internet a través de la interfaz bridge. A diferencia de la configuración NAT, en este caso la máquina virtual tiene acceso directo a la red externa a través del bridge, sin necesidad de NAT o reenvío de puertos.

17. **Comprobar la tabla de enrutamiento en la máquina virtual:** Se revisa cómo se encamina el tráfico en mvp5.

```
root@lq-d25:~# ip route
default via 10.140.92.1 dev bridge0 proto dhcp src 10.140.92.125
metric 425
10.22.122.0/24 dev virbr2 proto kernel scope link src 10.22.122.1
10.140.92.0/24 dev bridge0 proto kernel scope link src
10.140.92.125 metric 425
192.168.122.0/24 dev virbr0 proto kernel scope link src
192.168.122.1 linkdown
192.168.140.0/24 dev virbr1 proto kernel scope link src
192.168.140.1
```

Explicación de la tabla de rutas:

- default via 10.140.92.1 dev enp8s0: La ruta por defecto (hacia Internet) ahora es a través de la puerta de enlace de la red física (10.140.92.1) utilizando la interfaz **bridge (enp8s0)**. Esta configuración fue obtenida por DHCP.
- Las rutas para cada segmento de red están asignadas a la interfaz correspondiente:
  - Red 10.22.122.0/24 a través de enp7s0 (red aislada "Almacenamiento")
  - Red 10.140.92.0/24 a través de enp8s0 (interfaz bridge)
  - Red 192.168.140.0/24 a través de enp1s0 (red NAT "Cluster")

En resumen, se ha configurado con éxito una tercera interfaz de red en la máquina virtual mvp5 de tipo bridge. Esta configuración permite que la máquina virtual esté directamente conectada a la red física del laboratorio, obteniendo una dirección IP del mismo rango que los equipos físicos. A diferencia de las interfaces NAT y aislada configuradas anteriormente, esta interfaz bridge proporciona acceso directo a la red externa y permite que otros equipos de la red física se comuniquen directamente con la máquina virtual sin necesidad de configuraciones adicionales en el anfitrión.

## Validaciones

En esta sección se realizan diversas comprobaciones para verificar el correcto funcionamiento de todas las configuraciones de red implementadas en las tareas anteriores. Estas validaciones permiten confirmar que los tres tipos de redes están operativos y cumplen sus respectivos propósitos.

## Interfaces de red

Para tener una visión general de todas las interfaces de red configuradas en el sistema anfitrión, se utiliza el comando `ip -br addr show`:

```
root@lq-d25:~# ip -br addr show
lo                UNKNOWN      127.0.0.1/8  ::1/128
enp6s0            UP
bridge0           UP            10.140.92.125/24
fe80::e6e7:c28d:331c:1d14/64
virbr1            UP            192.168.140.1/24
virbr0            DOWN          192.168.122.1/24
virbr2            UP            10.22.122.1/24
vnet18            UNKNOWN      fe80::fc54:ff:febd:89a1/64
vnet19            UNKNOWN      fe80::fc16:3eff:fe6b:8bd9/64
vnet20            UNKNOWN      fe80::fc54:ff:fee4:bf90/64
```

Explicación de la salida:

- lo: La interfaz de loopback con la IP estándar 127.0.0.1/8.
- enp6s0: La interfaz física del anfitrión, ahora sin IP asignada directamente ya que está conectada al bridge.
- bridge0: El bridge para la conexión directa a la red física, con IP 10.140.92.125/24.
- virbr1: El bridge para la red NAT "Cluster", con IP 192.168.140.1/24.
- virbr0: El bridge para la red NAT por defecto, actualmente inactivo (DOWN).
- virbr2: El bridge para la red aislada "Almacenamiento", con IP 10.22.122.1/24.
- vnet18, vnet19, vnet20: Las interfaces virtuales que conectan las interfaces de la máquina virtual mvp5 con los respectivos bridges del anfitrión.

Se verifica también la tabla de enrutamiento del anfitrión:

```
root@lq-d25:~# ip route
default via 10.140.92.1 dev bridge0 proto dhcp src 10.140.92.125
metric 425
10.22.122.0/24 dev virbr2 proto kernel scope link src 10.22.122.1
10.140.92.0/24 dev bridge0 proto kernel scope link src
10.140.92.125 metric 425
192.168.122.0/24 dev virbr0 proto kernel scope link src
192.168.122.1 linkdown
```

```
192.168.140.0/24 dev virbr1 proto kernel scope link src
192.168.140.1
```

Explicación de la tabla de rutas:

- La ruta por defecto es a través de bridge0 hacia 10.140.92.1, que es el router de la red física.
- Existen rutas específicas para cada una de las redes virtuales:
  - o 10.22.122.0/24 a través de virbr2 (red aislada "Almacenamiento")
  - o 10.140.92.0/24 a través de bridge0 (red física)
  - o 192.168.122.0/24 a través de virbr0 (red NAT por defecto, inactiva)
  - o 192.168.140.0/24 a través de virbr1 (red NAT "Cluster")

## Reglas de firewall

Es importante verificar la configuración del firewall para asegurar que el tráfico de red fluya correctamente:

```
root@lq-d25:~# firewall-cmd --list-all
FedoraServer (default, active)
  target: default
  ingress-priority: 0
  egress-priority: 0
  icmp-block-inversion: no
  interfaces: bridge0 enp6s0
  sources:
  services: cockpit dhcpv6-client libvirt ssh
  ports: 49152-49216/tcp
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Explicación de la salida:

- El firewall tiene la zona FedoraServer como predeterminada y activa.
- Las interfaces bridge0 y enp6s0 están asignadas a esta zona.
- Los servicios permitidos incluyen cockpit, dhcpv6-client, libvirt y ssh.
- El reenvío de paquetes está habilitado (forward: yes).
- No se está realizando enmascaramiento (masquerade: no), lo que es coherente con la configuración de tipo bridge.

Se verifican también las reglas de NAT utilizando iptables:

```

root@lq-d25:~# iptables -t nat -L -v
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target      prot opt in      out     source
  destination

Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target      prot opt in      out     source
  destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target      prot opt in      out     source
  destination

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target      prot opt in      out     source
  destination
  3897 363K LIBVIRT_PRT all -- any     any     anywhere
  anywhere

Chain LIBVIRT_PRT (1 references)
  pkts bytes target      prot opt in      out     source
  destination
  1    40 RETURN    all -- any     any     192.168.122.0/24
  base-address.mcast.net/24
  0    0 RETURN    all -- any     any     192.168.122.0/24
  255.255.255.255
  0    0 MASQUERADE tcp -- any     any     192.168.122.0/24
  !192.168.122.0/24 masq ports: 1024-65535
  0    0 MASQUERADE udp -- any     any     192.168.122.0/24
  !192.168.122.0/24 masq ports: 1024-65535
  0    0 MASQUERADE all -- any     any     192.168.122.0/24
  !192.168.122.0/24
  4    160 RETURN    all -- any     any     192.168.140.0/24
  base-address.mcast.net/24
  0    0 RETURN    all -- any     any     192.168.140.0/24
  255.255.255.255
  0    0 MASQUERADE tcp -- any     any     192.168.140.0/24
  !192.168.140.0/24 masq ports: 1024-65535
  663 50388 MASQUERADE udp -- any     any     192.168.140.0/24
  !192.168.140.0/24 masq ports: 1024-65535
  0    0 MASQUERADE all -- any     any     192.168.140.0/24
  !192.168.140.0/24

```

Explicación de las reglas NAT:

- La cadena **LIBVIRT\_PRT** contiene las reglas de NAT creadas por libvirt para las redes virtuales.
- Las reglas MASQUERADE para la red 192.168.140.0/24 (red "Cluster") permiten que las máquinas virtuales en esta red accedan a redes externas a través del anfitrión.
- Se observa tráfico UDP significativo (663 50388) desde la red "Cluster" hacia redes externas, lo que confirma que el NAT está funcionando.

- No hay reglas de NAT para la red 10.22.122.0/24 (red "Almacenamiento"), lo que confirma que es una red aislada.

## Redes virtuales

Para entender en detalle la configuración de las redes virtuales, se examina su definición XML:

```
root@lq-d25:~# virsh net-dumpxml Cluster
<network connections='1'>
  <name>Cluster</name>
  <uuid>eca302af-e62c-4b0f-9823-6f06fa5e9282</uuid>
  <forward mode='nat'>
    <nat>
      <port start='1024' end='65535' />
    </nat>
  </forward>
  <bridge name='virbr1' stp='on' delay='0' />
  <mac address='52:54:00:f5:97:55' />
  <ip address='192.168.140.1' netmask='255.255.255.0'>
    <dhcp>
      <range start='192.168.140.2' end='192.168.140.149' />
    </dhcp>
  </ip>
</network>
```

Explicación de la red "Cluster":

- La red tiene el modo de reenvío configurado como nat.
- El rango de puertos para NAT es 1024-65535.
- Utiliza el bridge virbr1 con el protocolo STP activado.
- Tiene configurado un servidor DHCP para asignar IPs en el rango 192.168.140.2 a 192.168.140.149.

```
root@lq-d25:~# virsh net-dumpxml Almacenamiento
<network connections='1'>
  <name>Almacenamiento</name>
  <uuid>1f78a08c-c862-4419-98ed-bde618c88f6e</uuid>
  <bridge name='virbr2' stp='on' delay='0' />
  <mac address='52:54:00:fc:4f:fa' />
  <domain name='Almacenamiento' />
  <ip address='10.22.122.1' netmask='255.255.255.0'>
    </ip>
</network>
```

Explicación de la red "Almacenamiento":



- No tiene configurada la etiqueta <forward>, lo que confirma que es una red aislada.
- Utiliza el bridge virbr2.
- Tiene asignada la dirección IP 10.22.122.1/24 para el bridge.
- No contiene la sección <dhcp>, por lo que no proporciona asignación automática de IPs.

**Se examina el archivo de estado de las asignaciones DHCP para la red "Cluster":**

```
root@lq-d25:~# cat /var/lib/libvirt/dnsmasq/virbr1.status
[
  {
    "ip-address": "192.168.140.17",
    "mac-address": "52:54:00:bd:89:a1",
    "hostname": "mvp1",
    "client-id": "01:52:54:00:bd:89:a1",
    "expiry-time": 1744403512
  }
]
```

Explicación del archivo:

- Muestra la asignación DHCP activa para la red "Cluster" (bridge virbr1).
- La máquina virtual con la MAC 52:54:00:bd:89:a1 (primera interfaz de mvp5) tiene asignada la IP 192.168.140.17.
- La asignación tiene un tiempo de expiración definido (formato de fecha en segundos desde la época UNIX).

**Se verifican también las concesiones DHCP activas utilizando el comando virsh:**

```
root@lq-d25:~# virsh net-dhcp-leases Cluster
Expiry Time      dirección MAC      Protocol   IP address
Hostname    Client ID or DUID
-----
2025-04-11 21:31:52  52:54:00:bd:89:a1  ipv4
192.168.140.17/24  mvp1              01:52:54:00:bd:89:a1
```

Explicación de la salida:

- Muestra la concesión DHCP activa para la red "Cluster".
- La dirección MAC 52:54:00:bd:89:a1 corresponde a la primera interfaz de mvp5.
- La concesión tiene una fecha de expiración futura (2025-04-11).
- El cliente se identifica como mvp1, que es el nombre de host original de la máquina virtual antes de ser clonada para crear mvp5.

Estas validaciones confirman que todas las redes virtuales configuradas están funcionando correctamente según su diseño:

- La red NAT "Cluster" proporciona conectividad a Internet a través del anfitrión
- La red aislada "Almacenamiento" permite comunicación solo entre el anfitrión y la máquina virtual
- La interfaz bridge permite que la máquina virtual esté directamente conectada a la red física

## Conclusiones

Esta práctica ha permitido explorar y configurar los tipos fundamentales de redes virtuales disponibles en entornos KVM/libvirt: NAT, Aislada y Bridge. Cada tipo ofrece diferentes niveles de conectividad y aislamiento, adecuados para distintos escenarios de uso.

- La **Red NAT** (ej. "Cluster") es útil para proporcionar acceso a Internet a las máquinas virtuales de forma sencilla, sin requerir IPs adicionales de la red física y ofreciendo un nivel básico de aislamiento.
- La **Red Aislada** (ej. "Almacenamiento") es ideal para crear segmentos de red privados entre máquinas virtuales y el anfitrión, por ejemplo, para tráfico de almacenamiento o gestión, sin exponerlos a redes externas. La configuración manual de IP es necesaria al no disponer de DHCP.
- La **Interfaz Bridge** conecta la máquina virtual directamente a la red física del anfitrión, comportándose como un equipo más en la LAN. Esto simplifica el acceso a la MV desde otros equipos de la red física, pero requiere una gestión cuidadosa de las direcciones IP y MAC para evitar conflictos.

Se ha trabajado con herramientas de línea de comandos como `virsh` para la gestión de redes y máquinas virtuales, y `nmcli` para la configuración de red tanto en el anfitrión como en el invitado. La definición de redes mediante archivos XML ha demostrado ser un método eficaz y reproducible para la configuración de infraestructuras virtuales.

La configuración de la consola serie se reveló como un paso preparatorio esencial, permitiendo el acceso a la máquina virtual incluso después de eliminar su conectividad de red inicial.

Comprender las diferencias entre estos tipos de redes y saber cómo configurarlas es fundamental para diseñar e implementar infraestructuras virtualizadas seguras, eficientes y adaptadas a las necesidades específicas de cada aplicación o servicio.

## Bibliografía

- [1] P. S. Y. Z. L. N. D. P. S. R. T. R. Jiri Herrmann, «Virtualization Deployment and Administration Guide, Installing, configuring, and managing virtual machines on a RHEL physical machine,» Red Hat Enterprise , [En línea]. Available: [https://docs.redhat.com/es/documentation/red\\_hat\\_enterprise\\_linux/7/html-single/virtualization\\_deployment\\_and\\_administration\\_guide/index#chap-KVM\\_live\\_migration](https://docs.redhat.com/es/documentation/red_hat_enterprise_linux/7/html-single/virtualization_deployment_and_administration_guide/index#chap-KVM_live_migration). [Último acceso: 13 04 2025].
- [2] Z. Y. P. D. N. L. E. J. R. S. Herrmann J, «Documentación oficial de KVM,» 2025. [En línea]. Available: [https://access.redhat.com/documentation/enus/red\\_hat\\_enterprise\\_linux/7/html/virtualization\\_getting\\_started\\_guide/index](https://access.redhat.com/documentation/enus/red_hat_enterprise_linux/7/html/virtualization_getting_started_guide/index).
- [3] J. K. P. W. S. Peter Boy, «Fedora Server Documentation,» 2025. [En línea]. Available: <https://docs.fedoraproject.org/en-US/fedora-server>.
- [4] Z. Y. P. D. N. L. E. J. R. S. Herrmann J, «Red Hat Enterprise Linux 7. Virtualization Deployment and Administration Guide. Installing, configuring, and managing virtual machines on a RHEL physical machine,» [En línea]. Available: [https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/7/](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/). [Último acceso: 09 03 2025].
- [5] Red Hat Documentation, «Storage Management Guide,» [En línea]. Available: [https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/7/html/storage\\_administration\\_guide/index](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/storage_administration_guide/index). [Último acceso: 30 03 2025].