



# PRÁCTICA 6. INSTALACIÓN DE UN SISTEMA DE ALMACENAMIENTO ISCSI

## Virtualización y Procesamiento Distribuido

El objetivo de esta actividad es crear un área de almacenamiento compartido basado en la tecnología iSCSI

## Tabla de contenido

<b>Introducción .....</b>	<b>2</b>
<b>Requisitos previos .....</b>	<b>3</b>
<b>Desarrollo.....</b>	<b>4</b>
<b>Tarea 1: Creación de la infraestructura básica iSCSI .....</b>	<b>4</b>
Creación de máquinas virtuales mediante clonado .....	4
Configuración del nodo Almacenamiento (target).....	4
Configuración de interfaces de red en el nodo Almacenamiento.....	6
Configuración de nodos initiator (Nodo1 y Nodo2) .....	8
Configuración de red en los nodos initiator .....	10
<b>Tarea 2: Instalación y configuración de servicio iSCSI en el nodo target .....</b>	<b>13</b>
Paso 1: Instalación del software iSCSI en el nodo target.....	13
Paso 2: Inicio del servicio target .....	13
Paso 3: Configuración del arranque automático .....	13
Paso 4: Configuración del cortafuegos .....	14
Paso 5: Configuración del recurso de almacenamiento a exportar .....	14
<b>Tarea 3: Instalación del soporte iSCSI en los nodos initiator .....</b>	<b>18</b>
Paso 1: Instalación del software del cliente iSCSI .....	18
Paso 2: Ejecución del servicio iscsid .....	18
Paso 3: Configuración del nombre del nodo .....	19
Paso 4: Descubrimiento de LUNs exportados .....	20
Paso 5: Conexión de unidades LUNs .....	20
Paso 6: Comprobación de la conexión .....	21
<b>Tarea 4: Creación de sistema de archivos tipo ext4 en la unidad iSCSI importada .....</b>	<b>23</b>
<b>Tarea 5: Exportación del segundo disco iSCSI y creación de un volumen lógico.....</b>	<b>24</b>
1. Exportación del disco /dev/sdb desde el nodo target.....	24
2. Comprobación de la exportación en los nodos initiator .....	26
3. Creación del volumen lógico .....	27
4. Configuración en el segundo nodo initiator .....	30
5. Comprobación de activación en ambos nodos .....	30
<b>Pruebas y Validación.....</b>	<b>31</b>
<b>Comprobación del nodo target (Almacenamiento).....</b>	<b>31</b>
<b>Comprobación del nodo initiator (Nodo1) .....</b>	<b>33</b>
<b>Comprobación del nodo initiator (Nodo2) .....</b>	<b>33</b>
<b>Conclusiones.....</b>	<b>35</b>
<b>Bibliografía.....</b>	<b>36</b>

## Introducción

En esta práctica se configura un entorno de almacenamiento distribuido usando **iSCSI** sobre RHEL 7 (en nuestro caso, empleando Fedora como sistema base con comportamientos equivalentes a RHEL 7). **iSCSI** (*Internet Small Computer System Interface*) es un protocolo que transporta comandos SCSI a través de redes TCP/IP, permitiendo compartir dispositivos de **almacenamiento de bloques** entre servidores a través de una red.

En una arquitectura iSCSI intervienen dos roles principales:

- **Servidor de almacenamiento iSCSI (*Target*)**: el equipo que ofrece el almacenamiento (ej. un disco, volumen lógico o archivo) a la red. Se denomina “objetivo” o target iSCSI.
- **Ciente iSCSI (*Initiator*)**: el equipo que consume o utiliza ese almacenamiento remoto a nivel de bloque, como si fuera un disco más local en su sistema. Se le conoce como “iniciador” iSCSI.

**Nota:** En iSCSI, el servidor **target** presenta su almacenamiento mediante LUNs (*Logical Unit Numbers o Unidades Lógicas*). Cada LUN representa un volumen o disco lógico que los clientes pueden ver. Para distinguir y controlar el acceso, tanto el target como cada iniciador tienen un identificador único llamado **IQN** (*iSCSI Qualified Name*), con formato `iqn.AAAA-MM.dominio:Nombre`. Por ejemplo, un IQN podría ser `iqn.2025-05.com.ejemplo:servidor`. Los IQNs permiten que el target identifique a los clientes autorizados y viceversa.

A continuación, describimos el proceso para implementar un servidor iSCSI (*target*) y un cliente iSCSI (*initiator*), incluyendo la instalación de paquetes, la configuración del almacenamiento a exportar, y la conexión desde el cliente.

## Requisitos previos

Para la realización de esta práctica se requiere:

- Un sistema anfitrión con **KVM** (*Kernel-based Virtual Machine*) correctamente instalado y configurado.
- Conocimientos básicos sobre redes TCP/IP y sistemas de almacenamiento.
- Familiaridad con el sistema operativo Fedora/RHEL y la administración de sistemas Linux.
- Tres máquinas virtuales que funcionarán como:
  - Un servidor de almacenamiento (target iSCSI)
  - Dos nodos clientes (initiators iSCSI)
- Dos redes virtuales configuradas:
  - Red de almacenamiento (10.22.122.0/24): dedicada al tráfico iSCSI
  - Red de cluster (192.168.140.0/24): para comunicación general entre nodos

## Desarrollo

### Tarea 1: Creación de la infraestructura básica iSCSI

En esta tarea se implementará la infraestructura básica necesaria para el servicio iSCSI, que incluye la creación de las máquinas virtuales, la adición de discos y la configuración de las interfaces de red.

### Creación de máquinas virtuales mediante clonado

```
root@lq-d25:~# virt-clone --original mvp1 --name Almacenamiento --
file /var/lib/libvirt/images/Almacenamiento.qcow2 --mac
00:16:3e:76:57:a0
Allocating 'Almacenamiento.qcow2' | 1.6
GB 00:00 ...

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

Explicación del comando:

- virt-clone: Herramienta para clonar máquinas virtuales existentes
- --original mvp1: Especifica la máquina virtual de origen a clonar
- --name Almacenamiento: Asigna el nombre "Almacenamiento" a la nueva máquina virtual
- --file /var/lib/libvirt/images/Almacenamiento.qcow2: Especifica la ruta y nombre del archivo de imagen del disco para la nueva VM
- --mac 00:16:3e:76:57:a0: Asigna una dirección MAC específica a la interfaz de red de la máquina clonada

```
root@lq-d25:~# virt-clone --original mvp1 --name Nodo1 --file
/var/lib/libvirt/images/Nodo1.qcow2 --mac 00:16:3e:2a:bc:c9
Allocating 'Nodo1.qcow2' | 1.6
GB 00:00 ...

El clon 'Nodo1' ha sido creado exitosamente.
```

```
root@lq-d25:~# virt-clone --original mvp1 --name Nodo2 --file
/var/lib/libvirt/images/Nodo2.qcow2 --mac 00:16:3e:35:90:9c
Allocating 'Nodo2.qcow2' | 1.5
GB 00:00 ...

El clon 'Nodo2' ha sido creado exitosamente.
```

### Configuración del nodo Almacenamiento (target)

Primero se elimina el dispositivo CD-ROM y se reconfigura la red:

```
root@lq-d25:~# virsh detach-disk Almacenamiento sda --persistent
El disco ha sido desmontado exitosamente
```

```
root@lq-d25:~# virsh domiflist Almacenamiento
Interfaz  Tipo      Fuente      Modelo      MAC
-----
-          network default      virtio      00:16:3e:76:57:a0

root@lq-d25:~# virsh detach-interface Almacenamiento --type network
--mac 00:16:3e:76:57:a0 --persistent
La interfaz ha sido desmontada exitosamente
```

Explicación de los comandos:

- **virsh detach-disk:** Elimina un dispositivo de disco de una máquina virtual
  - o **--persistent:** Hace que el cambio persista después de reiniciar la VM
- **virsh domiflist:** Muestra las interfaces de red de una máquina virtual
- **virsh detach-interface:** Elimina una interfaz de red de la máquina virtual
  - o **--type network:** Especifica que es una interfaz de tipo red
  - o **--mac:** Identifica la interfaz a través de su dirección MAC

Luego se crean y añaden los volúmenes de almacenamiento adicionales:

```
root@lq-d25:~# virsh vol-create-as default vol1_p6.qcow2 1G --
format qcow2
Se ha creado el volumen vol1_p6.qcow2
```

```
root@lq-d25:~# virsh vol-create-as default vol2_p6.img 1G --format
raw
Se ha creado el volumen vol2_p6.img
```

```
root@lq-d25:~# virsh attach-disk Almacenamiento
/var/lib/libvirt/images/vol1_p6.qcow2 sda --driver qemu --type disk
--subdriver qcow2 --persistent
El disco ha sido asociado exitosamente
```

```
root@lq-d25:~# virsh attach-disk Almacenamiento
/var/lib/libvirt/images/vol2_p6.img sdb --driver qemu --type disk -
-subdriver raw --persistent
El disco ha sido asociado exitosamente
```

Explicación de los comandos:

- **virsh vol-create-as:** Crea un nuevo volumen en el pool de almacenamiento
  - o **default:** Nombre del pool de almacenamiento donde se crea el volumen
  - o **vol1\_p6.qcow2:** Nombre del nuevo volumen
  - o **1G:** Tamaño del volumen (1 Gigabyte)

- `--format qcow2`: Formato del disco, QCOW2 permite características como snapshots y thin provisioning
- `--format raw`: Formato raw es más simple, pero tiene mejor rendimiento
- **virsh attach-disk**: Asocia un dispositivo de almacenamiento a una máquina virtual
  - `--driver qemu`: Especifica el controlador para acceder al disco
  - `--type disk`: Define el dispositivo como un disco completo
  - `--subdriver qcow2/raw`: Especifica el formato del archivo de imagen

Finalmente, se añaden las interfaces de red:

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

```
root@lq-d25:~# virsh attach-interface Almacenamiento --type network
--source Cluster --model virtio --persistent
La interfaz ha sido asociada exitosamente
```

Explicación del comando:

- **virsh attach-interface**: Añade una interfaz de red a una máquina virtual
  - `--type network`: Define el tipo de interfaz como red virtual
  - `--source Almacenamiento/Cluster`: Especifica la red virtual a la que se conectará
  - `--model virtio`: Utiliza el modelo de virtualización paravirtualizado para mejor rendimiento
  - `--persistent`: Garantiza que la configuración persista después de reiniciar

## Configuración de interfaces de red en el nodo Almacenamiento

```
[root@mvp1 ~]# hostnamectl set-hostname almacenamiento.vpd.com
```

```
[root@mvp1 ~]# nmcli con add type ethernet con-name enp8s0 ifname
enp8s0 ipv4.method manual ipv4.addresses 10.22.122.10/24
Conexión «enp8s0» (5dbc051d-b4f5-4990-905f-472a9ef38463) añadida
con éxito.
[root@mvp1 ~]# nmcli con up enp8s0
Conexión activada con éxito (ruta activa D-Bus:
/org/freedesktop/NetworkManager/ActiveConnection/8)
```

Explicación del comando:

- **hostnamectl set-hostname**: Configura el nombre de host del sistema
- **nmcli con add**: Crea una nueva configuración de conexión de red

- type ethernet: Especifica que es una conexión cableada
- con-name enp8s0: Nombre de la conexión
- ifname enp8s0: Interfaz física a la que se aplica la configuración
- ipv4.method manual: Configura la dirección IP de forma manual, sin DHCP
- ipv4.addresses 10.22.122.10/24: Establece la dirección IP y máscara de red
- nmcli con up: Activa la conexión configurada

```
[root@mvp1 ~]# nmcli con add type ethernet con-name enp9s0 ifname enp9s0 ipv4.method auto
Conexión «enp9s0» (d39257b3-3a45-4262-91ff-9401751d0e90) añadida con éxito.
[root@mvp1 ~]# nmcli con up enp9s0
Conexión activada con éxito (ruta activa D-Bus: /org/freedesktop/NetworkManager/ActiveConnection/9)
```

Explicación del comando:

- ipv4.method auto: Configura la interfaz para obtener su dirección IP automáticamente vía DHCP

#### Validaciones:

```
[root@mvp1 ~]# hostnamectl
Static hostname: almacenamiento.vpd.com
Icon name: computer-vm
Chassis: vm
Machine ID: 6d2630bf3d2046a589c37eaa7313994b
Boot ID: 3392fb66d8a846e28e308c314b46b009
Product UUID: 9ac8572e-eda5-44f9-8282-5acbbb68449e
Virtualization: kvm
Operating System: Fedora Linux 41 (Server Edition)
CPE OS Name: cpe:/o:fedoraproject:fedora:41
OS Support End: Mon 2025-12-15
OS Support Remaining: 7month 3w
Kernel: Linux 6.12.11-200.fc41.x86_64
Architecture: x86-64
Hardware Vendor: QEMU
Hardware Model: Standard PC_Q35 + ICH9, 2009_
Firmware Version: 1.16.3-1.fc39
Firmware Date: Tue 2014-04-01
Firmware Age: 11y 3w 3d
```

```
[root@mvp1 ~]# nmcli device status
```

DEVICE	TYPE	STATE	CONNECTION
enp9s0	ethernet	conectado	enp9s0
enp8s0	ethernet	conectado	enp8s0
lo	loopback	connected (externally)	lo

```
[root@mvp1 ~]# 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: enp8s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
fq_codel state UP group default qlen 1000
    link/ether 52:54:00:48:9c:c6 brd ff:ff:ff:ff:ff:ff
    inet 10.22.122.10/24 brd 10.22.122.255 scope global
noprefixroute enp8s0
        valid_lft forever preferred_lft forever
    inet6 fe80::37b9:2f10:35ed:596c/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: enp9s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
fq_codel state UP group default qlen 1000
    link/ether 52:54:00:52:44:62 brd ff:ff:ff:ff:ff:ff
    inet 192.168.140.80/24 brd 192.168.140.255 scope global dynamic
noprefixroute enp9s0
        valid_lft 2218sec preferred_lft 2218sec
    inet6 fe80::3f3d:da7f:2578:6085/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

```

```

[root@mvp1 ~]# ping -c 3 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=114 time=33.8 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=114 time=34.3 ms

--- 8.8.8.8 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 33.839/34.090/34.341/0.251 ms

```

```

[root@mvp1 ~]# nmcli con show --active
NAME      UUID                                  TYPE      DEVICE
enp9s0    d39257b3-3a45-4262-91ff-9401751d0e90 ethernet  enp9s0
enp8s0    5dbc051d-b4f5-4990-905f-472a9ef38463 ethernet  enp8s0
lo        4ccd3034-48b7-4ca9-af3e-6c44e0aa7acb loopback   lo

```

```

[root@mvp1 ~]# ip route
default via 192.168.140.1 dev enp9s0 proto dhcp src 192.168.140.80
metric 103
10.22.122.0/24 dev enp8s0 proto kernel scope link src 10.22.122.10
metric 102
192.168.140.0/24 dev enp9s0 proto kernel scope link src
192.168.140.80 metric 103

```

## Configuración de nodos initiator (Nodo1 y Nodo2)

La configuración de los nodos iniciadores es similar a la del nodo target, comenzando por eliminar el dispositivo CD-ROM y reconfigurar las interfaces de red:

```
root@lq-d25:~# virsh detach-disk Nodo1 sda --persistent
El disco ha sido desmontado exitosamente
```

```
root@lq-d25:~# virsh domiflist Nodo1
Interfaz  Tipo      Fuente      Modelo      MAC
-----
-          network  default     virtio       00:16:3e:2a:bc:c9
```

```
root@lq-d25:~# virsh detach-interface Nodo1 --type network --mac
00:16:3e:2a:bc:c9 --persistent
La interfaz ha sido desmontada exitosamente
```

```
root@lq-d25:~# virsh attach-interface Nodo1 --type network --source
Almacenamiento --model virtio --persistent
La interfaz ha sido asociada exitosamente
```

```
root@lq-d25:~# virsh attach-interface Nodo1 --type network --source
Cluster --model virtio --persistent
La interfaz ha sido asociada exitosamente
```

```
root@lq-d25:~# virsh domiflist Nodo1
Interfaz  Tipo      Fuente      Modelo      MAC
-----
-          network  Almacenamiento  virtio       52:54:00:f9:be:78
-          network  Cluster        virtio       52:54:00:05:fd:ec
```

Se repite el procedimiento para el segundo nodo:

```
root@lq-d25:~# virsh detach-disk Nodo2 sda --persistent
El disco ha sido desmontado exitosamente
```

```
root@lq-d25:~# virsh domiflist Nodo2
Interfaz  Tipo      Fuente      Modelo      MAC
-----
-          network  default     virtio       00:16:3e:35:90:9c
```

```
root@lq-d25:~# virsh detach-interface Nodo2 --type network --mac
00:16:3e:35:90:9c --persistent
La interfaz ha sido desmontada exitosamente
```

```
root@lq-d25:~# virsh attach-interface Nodo2 --type network --source
Almacenamiento --model virtio --persistent
La interfaz ha sido asociada exitosamente
```

```
root@lq-d25:~# virsh attach-interface Nodo2 --type network --source
Cluster --model virtio --persistent
La interfaz ha sido asociada exitosamente
```

```
root@lq-d25:~# virsh domiflist Nodo2
Interfaz  Tipo      Fuente      Modelo      MAC
```

```
-----
-          network   Almacenamiento   virtio   52:54:00:a3:82:83
-          network   Cluster           virtio   52:54:00:33:fd:52
```

## Configuración de red en los nodos initiator

Procedemos a configurar las direcciones IP en ambos nodos. Primero para Nodo1:

```
[root@mvp1 ~]# hostnamectl set-hostname nodo1.vpd.com
```

```
[root@mvp1 ~]# nmcli con add type ethernet con-name enp1s0 ifname
enp1s0 ipv4.method manual ipv4.addresses 10.22.122.11/24
Aviso: hay otra conexión con el nombre «enp1s0». Haga referencia a
la conexión por su uuid «a8ab3908-5526-4f91-8171-762e42ac49ea»
Conexión «enp1s0» (a8ab3908-5526-4f91-8171-762e42ac49ea) añadida
con éxito.
```

```
[root@nodo1 ~]# nmcli con add type ethernet con-name enp7s0 ifname
enp7s0 ipv4.method auto
```

```
[root@nodo1 ~]# nmcli con up enp7s0
Conexión activada con éxito (ruta activa D-Bus:
/org/freedesktop/NetworkManager/ActiveConnection/9)
```

Verificación de la configuración en Nodo1:

```
[root@nodo1 ~]# ip a show
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:f9:be:78 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::698a:e0e6:e8b0:d494/64 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 52:54:00:05:fd:ec brd ff:ff:ff:ff:ff:ff
    inet 192.168.140.116/24 brd 192.168.140.255 scope global
    dynamic noprefixroute enp7s0
        valid_lft 3154sec preferred_lft 3154sec
    inet6 fe80::81:d839:b5cd:6ff5/64 scope link noprefixroute
```

```
valid_lft forever preferred_lft forever
```

Validaciones adicionales en Nodo1:

```
[root@nodo1 ~]# ip addr show enp1s0
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
fq_codel state UP group default qlen 1000
    link/ether 52:54:00:f9:be:78 brd ff:ff:ff:ff:ff:ff
    inet 10.22.122.11/24 brd 10.22.122.255 scope global
noprofixroute enp1s0
    valid_lft forever preferred_lft forever
    inet6 fe80::698a:e0e6:e8b0:d494/64 scope link noprofixroute
    valid_lft forever preferred_lft forever
[root@nodo1 ~]# ip addr show enp7s0
3: enp7s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
fq_codel state UP group default qlen 1000
    link/ether 52:54:00:05:fd:ec brd ff:ff:ff:ff:ff:ff
    inet 192.168.140.116/24 brd 192.168.140.255 scope global
dynamic noprofixroute enp7s0
    valid_lft 2937sec preferred_lft 2937sec
    inet6 fe80::81:d839:b5cd:6ff5/64 scope link noprofixroute
    valid_lft forever preferred_lft forever
```

```
[root@nodo1 ~]# 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=114 time=34.0 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=114 time=34.1 ms

--- 8.8.8.8 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 34.041/34.078/34.115/0.037 ms
```

Configuración de red para Nodo2:

```
[root@mvp1 ~]# nmcli con add type ethernet con-name enp1s0 ifname
enp1s0 ipv4.method manual ipv4.addresses 10.22.122.12/24
Conexión «enp1s0» (e695cda7-3af3-4246-b5fb-88d7f0ecfbcd) añadida
con éxito.
```

```
[root@mvp1 ~]# nmcli con add type ethernet con-name enp7s0 ifname
enp7s0 ipv4.method auto
```

```
[root@mvp1 ~]# nmcli con up enp7s0
Conexión activada con éxito (ruta activa D-Bus:
/org/freedesktop/NetworkManager/ActiveConnection/6)
```

```
[root@mvp1 ~]# ip a show
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:a3:82:83 brd ff:ff:ff:ff:ff:ff
    inet 10.22.122.12/24 brd 10.22.122.255 scope global
noprefixroute enp1s0
        valid_lft forever preferred_lft forever
        inet6 fe80::4c9f:ac5d:73af:d24f/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 52:54:00:33:fd:52 brd ff:ff:ff:ff:ff:ff
    inet 192.168.140.120/24 brd 192.168.140.255 scope global
dynamic noprefixroute enp7s0
        valid_lft 3429sec preferred_lft 3429sec
        inet6 fe80::da77:319b:6a4c:659/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

```

Verificación de Nodo2:

```

[root@mvp1 ~]# hostname
nodo2.vpd.com
[root@mvp1 ~]# ip addr show enp1s0
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
fq_codel state UP group default qlen 1000
    link/ether 52:54:00:a3:82:83 brd ff:ff:ff:ff:ff:ff
    inet 10.22.122.12/24 brd 10.22.122.255 scope global
noprefixroute enp1s0
        valid_lft forever preferred_lft forever
        inet6 fe80::4c9f:ac5d:73af:d24f/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[root@mvp1 ~]# ip addr show enp7s0
3: enp7s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
fq_codel state UP group default qlen 1000
    link/ether 52:54:00:33:fd:52 brd ff:ff:ff:ff:ff:ff
    inet 192.168.140.120/24 brd 192.168.140.255 scope global
dynamic noprefixroute enp7s0
        valid_lft 3120sec preferred_lft 3120sec
        inet6 fe80::da77:319b:6a4c:659/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[root@mvp1 ~]# 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=114 time=33.9 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=114 time=34.2 ms

--- 8.8.8.8 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 33.855/34.041/34.228/0.186 ms

```

## Tarea 2: Instalación y configuración de servicio iSCSI en el nodo target

Esta tarea se enfoca en la instalación y configuración del servicio iSCSI en el nodo target, que será el encargado de exportar los dispositivos de almacenamiento a los nodos initiator.

### Paso 1: Instalación del software iSCSI en el nodo target

En el servidor que actuará como target iSCSI, es necesario instalar las herramientas de administración de iSCSI. RHEL 7 utiliza **LIO** (Linux I/O) como subsistema de target iSCSI, configurado a través de la herramienta de consola `targetcli`. Instalaremos `targetcli` y activaremos el servicio del target para que se inicie con el sistema:

```
[root@almacenamiento ~]# dnf install targetcli
Actualizando y cargando repositorios:
...
Installing:
  targetcli                               noarch 2.1.58-3.fc41          fedora
275.7 KiB
...
Installing:
  targetcli                               noarch 2.1.58-3.fc41          fedora
275.7 KiB
```

Explicación del comando:

- **dnf install targetcli**: Instala el paquete `targetcli` que proporciona una interfaz para configurar el target iSCSI en Linux
  - o Este paquete proporciona una shell interactiva para la configuración del subsistema kernel target (LIO)

### Paso 2: Inicio del servicio target

```
[root@almacenamiento ~]# systemctl start target
```

Explicación del comando:

- **systemctl start target**: Inicia el servicio *iSCSI* target en el sistema
  - o El servicio target proporciona la funcionalidad necesaria para exportar almacenamiento block a través de la red

### Paso 3: Configuración del arranque automático

```
[root@almacenamiento ~]# systemctl enable target
Created symlink '/etc/systemd/system/multi-user.target.wants/target.service' →
'/usr/lib/systemd/system/target.service'.
```

Explicación del comando:

- **systemctl enable target**: Configura el servicio para que se inicie automáticamente durante el arranque del sistema
  - o Crea un enlace simbólico en el directorio de servicios del nivel de ejecución multi-usuario

## Paso 4: Configuración del cortafuegos

```
[root@almacenamiento ~]# firewall-cmd --permanent --add-port=3260/tcp
success
```

```
[root@almacenamiento ~]# firewall-cmd --reload
success
```

Explicación del comando:

- **firewall-cmd --permanent --add-port=3260/tcp**: Abre el puerto 3260/TCP en el cortafuegos
  - o **--permanent**: Hace que la regla persista después de reiniciar el servicio firewall
  - o **3260/tcp**: Puerto estándar utilizado por el protocolo iSCSI
- **firewall-cmd --reload**: Recarga la configuración del cortafuegos para aplicar los cambios inmediatamente

**Nota:** En entornos de producción se recomienda ubicar el tráfico iSCSI en una red dedicada o VLAN aislada, ya que iSCSI no cifra los datos en tránsito. Limitar el acceso al puerto 3260 solo a los clientes o subredes confiables es una buena práctica para mejorar la seguridad.

## Paso 5: Configuración del recurso de almacenamiento a exportar

Dentro del shell de **targetcli**, seguiremos estos pasos:

- Crear un **backstore** (almacén de datos) con nuestro archivo o dispositivo.
- Definir un **IQN** único para el target iSCSI.
- Crear un **portal** de red (IP/puerto) donde escuchará el servicio iSCSI.
- Asignar el almacenamiento (LUN) al target.
- Configurar una **ACL** (lista de control de acceso) para permitir a un iniciador específico conectarse.

```
[root@almacenamiento ~]# targetcli
targetcli shell version 2.1.58
Copyright 2011-2013 by Datera, Inc and others.
For help on commands, type 'help'.
/> ls
```

```

o- /
.....
.. [...]
   o- backstores
.....
      | o- block .....
[Storage Objects: 0]
      | o- fileio .....
[Storage Objects: 0]
      | o- pscsi .....
[Storage Objects: 0]
      | o- ramdisk .....
[Storage Objects: 0]
      o- iscsi .....
[Targets: 0]
      o- loopback .....
[Targets: 0]
      o- vhost .....
[Targets: 0]

```

Explicación del comando:

- **targetcli**: Inicia la interfaz de configuración del target iSCSI
  - o **ls**: Muestra la estructura jerárquica de configuración del target, que incluye:
  - o **backstores**: Dispositivos de almacenamiento de respaldo
  - o **iscsi**: Configuración de targets iSCSI
  - o **loopback** y **vhost**: Otros tipos de target disponibles

A continuación, se configura el dispositivo de respaldo:

```

/> cd /backstores/block
/backstores/block>

```

```

/backstores/block> create name=discosda dev=/dev/sdb

```

Explicación del comando:

- **cd /backstores/block**: Navega al directorio de configuración de dispositivos de respaldo de tipo bloque
- **create name=discosda dev=/dev/sdb**: Crea un objeto de almacenamiento block
  - o **name=discosda**: Nombre lógico asignado al dispositivo
  - o **dev=/dev/sdb**: Dispositivo físico que se exportará a través de iSCSI

Luego se crea el target iSCSI:

```

/backstores/block> cd /iscsi
/iscsi> create wwn=iqn.2025-04.com.vpd:discosda
Created target iqn.2025-04.com.vpd:discosda.

```



```

Created TPG 1.
Global pref auto_add_default_portal=true
Created default portal listening on all IPs (0.0.0.0), port 3260.
/iscsi> ls
o- iscsi .....
[Targets: 1]
  o- iqn.2025-04.com.vpd:discosda
    ..... [TPGs: 1]
      o- tpg1 ..... [no-gen-
acls, no-auth]
        o- acls
          ..... [ACLs: 0]
            o- luns
              ..... [LUNs: 0]
                o- portals .....
[Portals: 1]
  o- 0.0.0.0:3260
    ..... [OK]

```

Explicación del comando:

- **cd /iscsi:** Navega al directorio de configuración de targets iSCSI
- **create wwn=iqn.2025-04.com.vpd:discosda:** Crea un nuevo target iSCSI
  - o **wwn=iqn.2025-04.com.vpd:discosda:** Identificador único del target (IQN)
    - **iqn:** *Internet Qualified Name*, formato estándar para nombres iSCSI
    - **2025-04:** Fecha de registro del dominio (año-mes)
    - **com.vpd:** Dominio invertido del propietario
    - **discosda:** Nombre específico del target

**Se configura el portal de red:**

```

/iscsi> cd iqn.2025-04.com.vpd:discosda/tpg1/portals
/iscsi/iqn.20.../tpg1/portals> delete 0.0.0.0 3260
Deleted network portal 0.0.0.0:3260

```

```

/iscsi/iqn.20.../tpg1/portals> create 10.22.122.10
Using default IP port 3260
Created network portal 10.22.122.10:3260.

```

Explicación del comando:

- **cd iqn.2025-04.com.vpd:discosda/tpg1/portals:** Navega a la configuración de portales del target
- **delete 0.0.0.0 3260:** Elimina el portal por defecto que escucha en todas las interfaces
- **create 10.22.122.10:** Crea un nuevo portal que solo escucha en la dirección específica

- Esta configuración aumenta la seguridad al limitar el acceso solo a través de la red de almacenamiento

### Se configura el LUN (*Logical Unit Number*):

El siguiente paso es **mapear** el almacenamiento (backstore) al target como una LUN. Vamos a crear una LUN dentro del TPG usando el backstore discosda creado:

```
/iscsi/iqn.20.../tpg1/portals> cd ../luns
/iscsi/iqn.20...sda/tpg1/luns> create /backstores/block/discosda
Created LUN 0.
```

Explicación del comando:

- `cd ../luns`: Navega a la configuración de LUNs
- `create /backstores/block/discosda`: Asocia el objeto de almacenamiento creado anteriormente con el LUN 0
  - El LUN es un identificador numérico usado para distinguir entre múltiples dispositivos lógicos exportados

Finalmente, se configuran las **ACLs** (*Access Control Lists*):

```
/iscsi/iqn.20...sda/tpg1/luns> cd ../acls
/iscsi/iqn.20...sda/tpg1/acls> create wwn=iqn.2025-04.com.vpd:nodo1
Created Node ACL for iqn.2025-04.com.vpd:nodo1
Created mapped LUN 0.
/iscsi/iqn.20...sda/tpg1/acls> create wwn=iqn.2025-04.com.vpd:nodo2
Created Node ACL for iqn.2025-04.com.vpd:nodo2
Created mapped LUN 0.
```

Explicación del comando:

- `cd ../acls`: Navega a la configuración de listas de control de acceso
- `create wwn=iqn.2025-04.com.vpd:nodo1`: Crea una entrada ACL para el primer nodo initiator
- `create wwn=iqn.2025-04.com.vpd:nodo2`: Crea una entrada ACL para el segundo nodo initiator
  - Las **ACLs** definen qué *initiators* pueden conectarse al target
  - Cada entrada corresponde al nombre IQN de un nodo *client*

Se guarda la configuración y se sale de la interfaz:

```
/iscsi/iqn.20...sda/tpg1/acls> cd /
/> saveconfig
Configuration saved to /etc/target/saveconfig.json
/> exit
Global pref auto_save_on_exit=true
Last 10 configs saved in /etc/target/backup/.
Configuration saved to /etc/target/saveconfig.json
[root@almacenamiento ~]#
```

Explicación del comando:

- `cd /`: Vuelve al directorio raíz de la configuración
- `saveconfig`: Guarda la configuración actual en el archivo `/etc/target/saveconfig.json`
- `exit`: Sale de la interfaz de configuración

**Nota:** La configuración del target iSCSI se guarda automáticamente al salir, pero es una buena práctica guardarla explícitamente antes de salir para asegurarse de que todos los cambios están persistidos.

### Tarea 3: Instalación del soporte iSCSI en los nodos initiator

Esta tarea se centra en la instalación y configuración del software necesario en los nodos initiator para que puedan conectarse al target iSCSI y utilizar el almacenamiento compartido.

#### Paso 1: Instalación del software del cliente iSCSI

```
[root@nodo1 ~]# dnf install iscsi-initiator-utils
Actualizando y cargando repositorios:
Fedora 41 openh264 (From Cisco) - x86_ 100% | 3.1 KiB/s | 989.0
B | 00m00s
Fedora 41 - x86_64 - Updates 100% | 40.7 KiB/s | 23.9
KiB | 00m01s
Fedora 41 - x86_64 100% | 84.9 KiB/s | 26.2
KiB | 00m00s
Fedora 41 - x86_64 - Updates 100% | 3.2 MiB/s | 8.5
MiB | 00m03s
Repositorios cargados.
Package "iscsi-initiator-utils-6.2.1.10-0.gitd0f04ae.fc41.1.x86_64"
is already installed.
```

Explicación del comando:

- **`dnf install iscsi-initiator-utils`**: Instala el paquete con las utilidades necesarias para que el sistema actúe como initiator iSCSI
  - Este paquete incluye herramientas como `iscsiadm` para la administración de sesiones iSCSI y el demonio `iscsid` que gestiona las conexiones

#### Paso 2: Ejecución del servicio iscsid

```
[root@nodo1 ~]# systemctl restart iscsid
```

Explicación del comando:

- **systemctl restart iscsid**: Reinicia el servicio iscsid, que es el demonio que gestiona las conexiones iSCSI
  - o Este servicio es fundamental para mantener las sesiones iSCSI y gestionar la reconexión automática

Como resultado de la ejecución del servicio se creará el archivo de configuración `/etc/iscsi/initiatorname.iscsi` que contiene el nombre IQN del initiator:

```
[root@nodo1 ~]# cat /etc/iscsi/initiatorname.iscsi
InitiatorName=iqn.1994-05.com.redhat:265ed95a5ff3
```

```
[root@nodo2 ~]# cat /etc/iscsi/initiatorname.iscsi
InitiatorName=iqn.1994-05.com.redhat:a66dcccc74dc
```

El archivo `initiatorname.iscsi` contiene el IQN con el que este cliente se identificará ante los targets. Podemos editar este archivo si queremos asignarle un nombre más acorde a nuestra organización o entorno, pero debe coincidir con el IQN que configuramos en la ACL del target.

**Nota:** Es importante que cada cliente tenga un IQN único. Dos máquinas diferentes no deberían usar el mismo IQN simultáneamente, ya que el target las vería como si fueran el mismo iniciador, pudiendo causar conflictos en la conexión.

### Paso 3: Configuración del nombre del nodo

Para ser coherente con la configuración de ACLs en el target, es necesario modificar el nombre del initiator:

Nodo 1:

```
[root@nodo1 ~]# cat /etc/iscsi/initiatorname.iscsi
InitiatorName=iqn.2025-04.com.vpd:nodo1
```

Nodo 2:

```
[root@nodo2 ~]# cat /etc/iscsi/initiatorname.iscsi
InitiatorName=iqn.2025-04.com.vpd:nodo2
```

Explicación:

- El archivo `/etc/iscsi/initiatorname.iscsi` se ha editado para establecer nombres IQN personalizados
- El formato utilizado es: `iqn.año-mes.dominio.invertido:nombre`
- Estos nombres deben coincidir exactamente con los configurados en las ACLs del target

## Paso 4: Descubrimiento de LUNs exportados

```
[root@nodo1 ~]# iscsiadm --mode discovery --type sendtargets --portal 10.22.122.10 --discover
10.22.122.10:3260,1 iqn.2025-04.com.vpd:discosda
```

```
[root@nodo2 ~]# iscsiadm --mode discovery --type sendtargets --portal 10.22.122.10 --discover
10.22.122.10:3260,1 iqn.2025-04.com.vpd:discosda
```

Explicación del comando:

- **iscsiadm --mode discovery --type sendtargets --portal 10.22.122.10 --discover**: Realiza un descubrimiento de targets iSCSI disponibles
  - o **--mode discovery**: Opera en modo descubrimiento para buscar targets
  - o **--type sendtargets**: Utiliza el método SendTargets, el más común para descubrimiento iSCSI
  - o **--portal 10.22.122.10**: Dirección IP del target iSCSI a consultar
  - o **--discover**: Indica que se debe realizar una operación de descubrimiento

## Paso 5: Conexión de unidades LUNs

```
[root@nodo1 ~]# sudo iscsiadm --mode node --targetname iqn.2025-04.com.vpd:discosda --portal 10.22.122.10 --login
Logging in to [iface: default, target: iqn.2025-04.com.vpd:discosda, portal: 10.22.122.10,3260]
Login to [iface: default, target: iqn.2025-04.com.vpd:discosda, portal: 10.22.122.10,3260] successful.
```

```
[root@nodo2 ~]# sudo iscsiadm --mode node --targetname iqn.2025-04.com.vpd:discosda --portal 10.22.122.10 --login
Logging in to [iface: default, target: iqn.2025-04.com.vpd:discosda, portal: 10.22.122.10,3260]
Login to [iface: default, target: iqn.2025-04.com.vpd:discosda, portal: 10.22.122.10,3260] successful.
```

Explicación del comando:

- **iscsiadm --mode node --targetname iqn.2025-04.com.vpd:discosda --portal 10.22.122.10 --login**: Inicia una sesión iSCSI con el target
  - o **--mode node**: Opera en modo nodo, para gestionar sesiones iSCSI
  - o **--targetname iqn.2025-04.com.vpd:discosda**: Especifica el nombre IQN del target al que conectarse
  - o **--portal 10.22.122.10**: Dirección IP y puerto del portal iSCSI
  - o **--login**: Indica que se debe iniciar sesión en el target

## Paso 6: Comprobación de la conexión

Verificar en los logs del sistema:

```
[root@nodo1 ~]# sudo journalctl --since today | grep -i iscsi
abr 25 19:15:55 nodo1.vpd.com sudo[1323]:    root : TTY=ttyS0 ;
PWD=/root ; USER=root ; COMMAND=/usr/sbin/iscsiadm --mode node --
targetname iqn.2025-04.com.vpd:discosda --portal 10.22.122.10 --
login
abr 25 19:15:55 nodo1.vpd.com kernel: scsi host6: iSCSI Initiator
over TCP/IP
abr 25 19:15:55 nodo1.vpd.com iscsid[1286]: iscsid: Connection1:0
to [target: iqn.2025-04.com.vpd:discosda, portal:
10.22.122.10,3260] through [iface: default] is operational now
```

```
[root@nodo2 ~]# sudo journalctl --since today | grep -i iscsi
abr 25 19:16:00 nodo2.vpd.com sudo[1243]:    root : TTY=ttyS0 ;
PWD=/root ; USER=root ; COMMAND=/usr/sbin/iscsiadm --mode node --
targetname iqn.2025-04.com.vpd:discosda --portal 10.22.122.10 --
login
abr 25 19:16:00 nodo2.vpd.com kernel: scsi host6: iSCSI Initiator
over TCP/IP
abr 25 19:16:00 nodo2.vpd.com iscsid[1234]: iscsid: Connection1:0
to [target: iqn.2025-04.com.vpd:discosda, portal:
10.22.122.10,3260] through [iface: default] is operational now
```

Explicación del comando:

- **journalctl --since today | grep -i iscsi:** Filtra los logs del sistema para mostrar las entradas relacionadas con iSCSI
  - o La salida muestra la creación del host SCSI virtual para la conexión iSCSI
  - o También confirma que la conexión está operativa a través de la interfaz predeterminada

Verificación de sesiones activas:

```
[root@nodo1 ~]# iscsiadm -m session
tcp: [1] 10.22.122.10:3260,1 iqn.2025-04.com.vpd:discosda (non-
flash)
```

```
[root@nodo2 ~]# iscsiadm -m session
tcp: [1] 10.22.122.10:3260,1 iqn.2025-04.com.vpd:discosda (non-
flash)
```

Explicación del comando:

- **iscsiadm -m session:** Muestra información sobre las sesiones iSCSI activas
  - o tcp: [1]: Indica el número de sesión

- 10.22.122.10:3260,1: Muestra la IP, puerto y grupo del portal
- iqn.2025-04.com.vpd:discosda: Nombre del target conectado
- (non-flash): Indica que no se trata de un dispositivo flash optimizado

Verificación de dispositivos disponibles con lsblk:

```
[root@nodo1 ~]# lsblk -o NAME,TRAN,SIZE,TYPE,MOUNTPOINT
NAME          TRAN      SIZE TYPE MOUNTPOINT
sda           iscsi      1G disk
zram0         1,9G disk [SWAP]
vda          virtio    10G disk
├─vda1        virtio      1M part
├─vda2        virtio      1G part /boot
└─vda3        virtio      9G part
   └─fedora-root 9G lvm  /
```

```
[root@nodo2 ~]# lsblk -o NAME,TRAN,SIZE,TYPE,MOUNTPOINT
NAME          TRAN      SIZE TYPE MOUNTPOINT
sda           iscsi      1G disk
zram0         1,9G disk [SWAP]
vda          virtio    10G disk
├─vda1        virtio      1M part
├─vda2        virtio      1G part /boot
└─vda3        virtio      9G part
   └─fedora-root 9G lvm  /
```

Explicación del comando:

- **lsblk -o NAME,TRAN,SIZE,TYPE,MOUNTPOINT**: Muestra información detallada sobre los dispositivos de bloques
  - NAME: Nombre del dispositivo
  - TRAN: Tipo de transporte (iscsi para dispositivos iSCSI)
  - SIZE: Tamaño del dispositivo
  - TYPE: Tipo de dispositivo (disk, part, lvm)
  - MOUNTPOINT: Punto de montaje, si está montado

**Nota:** En la salida se puede ver que el dispositivo sda está disponible con transporte iscsi y un tamaño de 1G, que corresponde al disco exportado desde el target.

## Tarea 4: Creación de sistema de archivos tipo ext4 en la unidad iSCSI importada

En esta tarea se formateará la unidad iSCSI importada con un sistema de archivos ext4 y se montará para su uso.

```
[root@nodo1 ~]# mkfs.ext4 /dev/sda
mke2fs 1.47.1 (20-May-2024)
Se está creando un sistema de ficheros con 262144 bloques de 4k y
65536 nodos-i
UUID del sistema de ficheros: 733dd613-51be-4533-9245-f71c2afe9bb2
Respaldos del superbloque guardados en los bloques:
    32768, 98304, 163840, 229376

Reservando las tablas de grupo: hecho
Escribiendo las tablas de nodos-i: hecho
Creando el fichero de transacciones (8192 bloques): hecho
Escribiendo superbloques y la información contable del sistema de
ficheros: hecho
```

Explicación del comando:

- **mkfs.ext4 /dev/sda**: Crea un sistema de archivos ext4 en el dispositivo iSCSI
  - Este comando formatea el dispositivo completo, destruyendo cualquier dato existente
  - Los parámetros se establecen automáticamente basados en el tamaño del dispositivo
  - El sistema generará un UUID único para el sistema de archivos
  - Los respaldos del superbloque permiten recuperar el sistema de archivos en caso de daño

Después, montamos el nuevo sistema de archivos para probar su funcionamiento:

```
[root@nodo1 ~]# mount /dev/sda /mnt
```

Explicación del comando:

- **mount /dev/sda /mnt**: Monta el sistema de archivos recién creado en el directorio /mnt
  - Esta operación hace que el sistema de archivos sea accesible a través del punto de montaje /mnt
  - No se especifican opciones adicionales, por lo que se utilizan las opciones por defecto

**Nota:** El sistema de archivos ext4 es una buena elección para almacenamiento compartido básico debido a su amplia compatibilidad y soporte para recuperación después de fallos.



## Tarea 5: Exportación del segundo disco iSCSI y creación de un volumen lógico

Esta tarea demuestra una configuración más avanzada, donde se exporta un segundo disco iSCSI y se crea un volumen lógico (LVM) que puede ser compartido entre múltiples nodos.

### 1. Exportación del disco /dev/sdb desde el nodo target

```
[root@almacenamiento ~]# targetcli
targetcli shell version 2.1.58
Copyright 2011-2013 by Datera, Inc and others.
For help on commands, type 'help'.

/> cd backstores/block/
/backstores/block>

/backstores/block> create name=discosdb dev=/dev/sdb
Created block storage object discosdb using /dev/sdb.

/backstores/block> cd /iscsi

/iscsi> create wwn=iqn.2025-04.com.vpd:servidorapache
Created target iqn.2025-04.com.vpd:servidorapache.
Created TPG 1.
Default portal not created, TPGs within a target cannot share
ip:port.
```

Explicación del comando:

- **create name=discosdb dev=/dev/sdb**: Crea un objeto de almacenamiento con nombre "discosdb" respaldado por el dispositivo físico /dev/sdb
- **create wwn=iqn.2025-04.com.vpd:servidorapache**: Crea un segundo target iSCSI con un identificador diferente para el segundo disco
  - El mensaje "Default portal not created..." indica que no se creó un portal por defecto porque no se pueden compartir las mismas combinaciones IP:puerto entre diferentes targets

```
/iscsi> cd iqn.2025-04.com.vpd:servidorapache/tpg1/portals

/iscsi/iqn.20.../tpg1/portals> delete 0.0.0.0 3260
No such NetworkPortal in configs:
/sys/kernel/config/target/iscsi/iqn.2025-
04.com.vpd:servidorapache/tpgt_1/np/0.0.0.0:3260

/iscsi/iqn.20.../tpg1/portals> create 10.22.122.10
Using default IP port 3260
Created network portal 10.22.122.10:3260.
```

Explicación del comando:

- La eliminación del portal por defecto falla porque no se creó automáticamente
- **create 10.22.122.10:** Crea un portal específico en la dirección IP de la red de almacenamiento
  - o Utiliza el puerto predeterminado 3260, que es el estándar para iSCSI

```
/iscsi/iqn.20.../tpg1/portals> cd ..  
/iscsi/iqn.20...orapache/tpg1> cd luns  
/iscsi/iqn.20...che/tpg1/luns> create /backstores/block/discosdb  
Created LUN 0.
```

Explicación del comando:

- **create /backstores/block/discosdb:** Asocia el objeto de almacenamiento al LUN 0 del target
  - o La numeración de LUNs comienza desde 0 y es visible para los initiators

```
/iscsi/iqn.20...che/tpg1/luns> cd ..  
/iscsi/iqn.20...orapache/tpg1> cd acls  
/iscsi/iqn.20...che/tpg1/acls> create wwn=iqn.2025-04.com.vpd:nodo1  
Created Node ACL for iqn.2025-04.com.vpd:nodo1  
Created mapped LUN 0.  
/iscsi/iqn.20...che/tpg1/acls> create wwn=iqn.2025-04.com.vpd:nodo2  
Created Node ACL for iqn.2025-04.com.vpd:nodo2  
Created mapped LUN 0.
```

Explicación del comando:

- **create wwn=iqn.2025-04.com.vpd:nodo1:** Crea una ACL para el primer nodo initiator
- **create wwn=iqn.2025-04.com.vpd:nodo2:** Crea una ACL para el segundo nodo initiator
  - o Las ACLs definen qué initiators pueden acceder al target
  - o Para cada ACL se crea automáticamente un mapeo al LUN 0

```
/iscsi/iqn.20...che/tpg1/acls> cd /  
/> saveconfig  
Configuration saved to /etc/target/saveconfig.json  
/> exit  
Global pref auto_save_on_exit=true  
Last 10 configs saved in /etc/target/backup/.  
Configuration saved to /etc/target/saveconfig.json
```

Explicación del comando:

- **saveconfig:** Guarda la configuración de forma explícita antes de salir
- Al salir, se muestra que la configuración se guarda automáticamente y se mantienen las últimas 10 versiones

## 2. Comprobación de la exportación en los nodos initiator

```
[root@nodo1 ~]# iscsiadm --mode discovery --type sendtargets --portal 10.22.122.10 --discover
10.22.122.10:3260,1 iqn.2025-04.com.vpd:discosda
10.22.122.10:3260,1 iqn.2025-04.com.vpd:servidorapache
```

```
[root@nodo2 ~]# iscsiadm --mode discovery --type sendtargets --portal 10.22.122.10 --discover
10.22.122.10:3260,1 iqn.2025-04.com.vpd:discosda
10.22.122.10:3260,1 iqn.2025-04.com.vpd:servidorapache
```

Explicación del comando:

- Ahora el descubrimiento muestra dos targets disponibles desde el mismo portal:
  - El target original iqn.2025-04.com.vpd:discosda
  - El nuevo target iqn.2025-04.com.vpd:servidorapache

```
[root@nodo1 ~]# iscsiadm --mode node \
    --targetname iqn.2025-04.com.vpd:servidorapache \
    --portal 10.22.122.10 --login
Logging in to [iface: default, target: iqn.2025-04.com.vpd:servidorapache, portal: 10.22.122.10,3260]
Login to [iface: default, target: iqn.2025-04.com.vpd:servidorapache, portal: 10.22.122.10,3260] successful.
```

```
[root@nodo2 ~]# iscsiadm --mode node \
    --targetname iqn.2025-04.com.vpd:servidorapache \
    --portal 10.22.122.10 --login
Logging in to [iface: default, target: iqn.2025-04.com.vpd:servidorapache, portal: 10.22.122.10,3260]
Login to [iface: default, target: iqn.2025-04.com.vpd:servidorapache, portal: 10.22.122.10,3260] successful.
```

Explicación del comando:

- Con estos comandos, ambos nodos inician sesión con el nuevo target iSCSI
- La opción --targetname especifica exactamente con qué target se desea conectar

Verificación:

```
[root@nodo1 ~]# lsblk -o NAME,TRAN,SIZE,TYPE
NAME        TRAN      SIZE TYPE
sda         iscsi     1G  disk
sdb         iscsi     1G  disk
zram0              1,9G disk
vda         virtio    10G disk
├─vda1       virtio     1M part
├─vda2       virtio     1G part
└─vda3       virtio     9G part
    └─fedora-root          9G lvm
```

```
[root@nodo2 ~]# lsblk -o NAME,TRAN,SIZE,TYPE
NAME          TRAN      SIZE TYPE
sda            iscsi      1G disk
sdb            iscsi      1G disk
zram0          1,9G disk
vda            virtio     10G disk
├─vda1         virtio      1M part
├─vda2         virtio      1G part
└─vda3         virtio      9G part
   └─fedora-root 9G lvm
```

Explicación del comando:

- Después de conectarse, aparece un segundo dispositivo iSCSI (sdb) en ambos nodos
- La columna TRAN muestra que ambos dispositivos son de tipo iscsi

### 3. Creación del volumen lógico

#### I. Creación del volumen físico

```
[root@nodo1 ~]# pvcreate /dev/sdb
Physical volume "/dev/sdb" successfully created.
```

Explicación del comando:

- **pvcreate /dev/sdb**: Inicializa el dispositivo iSCSI como un volumen físico de LVM
  - o Escribe una cabecera PV al inicio del dispositivo
  - o Esta es la base para la creación posterior de volúmenes lógicos

#### II. Creación del grupo de volúmenes ApacheVG

```
[root@nodo1 ~]# vgcreate --setautoactivation n --locktype none
ApacheVG /dev/sdb
Volume group "ApacheVG" successfully created
```

Explicación del comando:

- **vgcreate**: Crea un grupo de volúmenes donde se pueden crear volúmenes lógicos
  - o **--setautoactivation n**: Desactiva la activación automática del VG en el arranque
    - Esto es esencial para entornos compartidos para evitar activaciones conflictivas
  - o **--locktype none**: No utiliza bloqueo, permitiendo que el VG sea activado desde cualquier host

- Esta es una configuración simplificada; en producción, normalmente se usaría un sistema de bloqueo distribuido
- ApacheVG: Nombre del grupo de volúmenes
- /dev/sdb: Volumen físico a incluir en el grupo

### III. Creación del volumen lógico ApacheLV

```
[root@nodo1 ~]# lvcreate -n ApacheLV -L 900M ApacheVG
```

Explicación del comando:

- lvcreate: Crea un volumen lógico dentro del grupo de volúmenes
  - -n ApacheLV: Especifica el nombre del volumen lógico
  - -L 900M: Define el tamaño del volumen (900 MiB)
  - ApacheVG: Nombre del grupo de volúmenes donde se creará el LV

```
[root@nodo1 ~]# lvs
  LV          VG          Attr          LSize   Pool Origin Data%  Meta%
Move Log Cpy%Sync Convert
ApacheLV ApacheVG -wi-a----- 900,00m
root        fedora    -wi-ao----- <9,00g
```

Explicación del comando:

- lvs: Muestra información sobre los volúmenes lógicos
  - Se muestra el volumen ApacheLV recién creado con un tamaño de 900 MB
  - También se muestra el volumen lógico root del sistema, que pertenece a otro grupo de volúmenes (fedora)

### IV. Creación del sistema de archivos XFS

```
[root@nodo1 ~]# 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=1
       =                               reflink=1    bigtime=1
inobtcount=1 nnext64=1
data    =                               bsize=4096   blocks=230400,
imaxpct=25
       =                               sunit=0      swidth=0 blks
naming  =version 2                     bsize=4096   ascii-ci=0, ftype=1
log      =internal log                 bsize=4096   blocks=16384,
version=2
       =                               sectsz=512    sunit=0 blks, lazy-
count=1
realtime =none                         extsz=4096   blocks=0, rtextents=0
```

Explicación del comando:

- **mkfs.xfs**: Crea un sistema de archivos XFS en el volumen lógico
  - o XFS es especialmente adecuado para entornos compartidos y volúmenes grandes
  - o La salida muestra los diversos parámetros del sistema de archivos:
    - **isize=512**: Tamaño de los inodos
    - **agcount=4**: Número de grupos de asignación
    - **crc=1**: Se utilizan checksums CRC para validar datos
    - **reflink=1**: Soporte para copias eficientes (reflink)

#### V. Prueba de montaje y validación

```
[root@nodo1 ~]# mount /dev/AdobeVG/AdobeLV /mnt
```

Explicación del comando:

- Monta el volumen lógico en el directorio /mnt para acceder al sistema de archivos

```
[root@nodo1 ~]# pvs
PV          VG          Fmt  Attr  PSize   PFree
/dev/sdb    AdobeVG    lvm2  a--   992,00m 92,00m
/dev/vda3   fedora     lvm2  a--    <9,00g   0
[root@nodo1 ~]# vgs
VG          #PV #LV #SN Attr   VSize   VFree
AdobeVG     1   1   0 wz--n- 992,00m 92,00m
fedora       1   1   0 wz--n-  <9,00g   0
```

```
[root@nodo1 ~]# lvs
LV          VG          Attr              LSize   Pool Origin Data%  Meta%
Move Log Cpy%Sync Convert
AdobeLV    AdobeVG    -wi-ao----   900,00m
root        fedora     -wi-ao----    <9,00g
```

Explicación de los comandos:

- **pvs**: Muestra información sobre volúmenes físicos
  - o /dev/sdb es parte del grupo AdobeVG con espacio libre de 92 MB
- **vgs**: Muestra información sobre grupos de volúmenes
  - o AdobeVG tiene 1 PV, 1 LV, y atributo "wz--n-" (write, resizable, sin autoactivación)
- **lvs**: Muestra información detallada sobre volúmenes lógicos
  - o El atributo "-wi-ao----" indica "writable, active, open"

```
[root@nodo1 ~]# mount | grep AdobeVG
/dev/mapper/AdobeVG-AdobeLV on /mnt type xfs
(rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquota)
```

Explicación del comando:

- Verifica que el volumen está correctamente montado
- Muestra las opciones de montaje del sistema de archivos XFS

#### 4. Configuración en el segundo nodo initiator

```
[root@nodo2 ~]# lvmdevices --adddev /dev/sdb
```

Explicación del comando:

- **lvmdevices --adddev /dev/sdb**: Añade el dispositivo iSCSI al inventario de dispositivos LVM en el segundo nodo
  - o Esto es necesario para que LVM reconozca y pueda gestionar este dispositivo

#### 5. Comprobación de activación en ambos nodos

```
[root@nodo1 ~]# vgchange -ay ApacheVG
1 logical volume(s) in volume group "ApacheVG" now active
[root@nodo1 ~]# mount /dev/ApacheVG/ApacheLV /mnt
[root@nodo1 ~]# df -h /mnt
S.ficheros                                Tamaño Usados  Disp Uso% Montado en
/dev/mapper/ApacheVG-ApacheLV             836M      49M   788M   6% /mnt
```

```
[root@nodo2 ~]# vgchange -ay ApacheVG
1 logical volume(s) in volume group "ApacheVG" now active
[root@nodo2 ~]# mount /dev/ApacheVG/ApacheLV /mnt
[root@nodo2 ~]# df -h /mnt
S.ficheros                                Tamaño Usados  Disp Uso% Montado en
/dev/mapper/ApacheVG-ApacheLV             836M      49M   788M   6% /mnt
```

Explicación del comando:

- **vgchange -ay ApacheVG**: Activa el grupo de volúmenes en el nodo
  - o -a: Cambia el estado de activación
  - o y: Activo (yes)
- **mount /dev/ApacheVG/ApacheLV /mnt**: Monta el volumen lógico
- **df -h /mnt**: Muestra información sobre el espacio utilizado
  - o El sistema de archivos XFS tiene un tamaño total de 836 MB, con 49 MB utilizados

**Nota:** Es importante resaltar que, en entornos de producción, cuando múltiples nodos acceden al mismo volumen lógico, se necesita un sistema de archivos en clúster o un mecanismo de bloqueo para prevenir la corrupción de datos. En este ejemplo simplificado, los nodos deberían coordinarse manualmente para evitar acceder al mismo tiempo al volumen.

## Pruebas y Validación

Esta sección demuestra que la infraestructura iSCSI se ha configurado correctamente y que los volúmenes están accesibles y funcionales desde ambos nodos.

### Comprobación del nodo target (Almacenamiento)

Verificar las interfaces de red:

```
[root@almacenamiento ~]# ip addr show
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: enp8s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
fq_codel state UP group default qlen 1000
    link/ether 52:54:00:48:9c:c6 brd ff:ff:ff:ff:ff:ff
    inet 10.22.122.10/24 brd 10.22.122.255 scope global
    noprefixroute enp8s0
        valid_lft forever preferred_lft forever
    inet6 fe80::37b9:2f10:35ed:596c/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: enp9s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
fq_codel state UP group default qlen 1000
    link/ether 52:54:00:52:44:62 brd ff:ff:ff:ff:ff:ff
    inet 192.168.140.80/24 brd 192.168.140.255 scope global dynamic
    noprefixroute enp9s0
        valid_lft 3588sec preferred_lft 3588sec
    inet6 fe80::3f3d:da7f:2578:6085/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

```
[root@almacenamiento ~]# nmcli device status
DEVICE  TYPE      STATE      CONNECTION
enp9s0  ethernet  conectado  enp9s0
enp8s0  ethernet  conectado  enp8s0
lo      loopback  connected (externally)  lo
```

Discos presentes:

```
[root@almacenamiento ~]# lsblk -o NAME,TRAN,SIZE,TYPE
NAME        TRAN      SIZE TYPE
sda         spi       1G  disk
sdb         spi       1G  disk
zram0              1,9G disk
vda         virtio    10G disk
├─vda1       virtio    1M  part
├─vda2       virtio    1G  part
└─vda3       virtio    9G  part
```



## Configuración iSCSI exportada:

```

[root@almacenamiento ~]# targetcli /iscsi ls
o- iscsi .....
[Targets: 2]
  o- iqn.2025-04.com.vpd:discosda
    ..... [TPGs: 1]
      | o- tpg1 ..... [no-gen-
acls, no-auth]
      |   o- acls
    ..... [ACLs: 2]
      |   | o- iqn.2025-04.com.vpd:nodo1 .....
[Mapped LUNs: 1]
      |   | | o- mapped_lun0 ..... [lun0
block/discosda (rw)]
      |   | o- iqn.2025-04.com.vpd:nodo2 .....
[Mapped LUNs: 1]
      |   |   o- mapped_lun0 ..... [lun0
block/discosda (rw)]
      |   |   o- luns
    ..... [LUNs: 1]
      |   | o- lun0 ..... [block/discosda (/dev/sda)
(default_tg_pt_gp)]
      |   o- portals .....
[Portals: 1]
      |     o- 10.22.122.10:3260
    ..... [OK]
  o- iqn.2025-04.com.vpd:servidorapache
    ..... [TPGs: 1]
      o- tpg1 ..... [no-gen-
acls, no-auth]
      o- acls
    ..... [ACLs: 2]
      | o- iqn.2025-04.com.vpd:nodo1 .....
[Mapped LUNs: 1]
      | | o- mapped_lun0 ..... [lun0
block/discosdb (rw)]
      | o- iqn.2025-04.com.vpd:nodo2 .....
[Mapped LUNs: 1]
      |   o- mapped_lun0 ..... [lun0
block/discosdb (rw)]
      |   o- luns
    ..... [LUNs: 1]
      | o- lun0 ..... [block/discosdb (/dev/sdb)
(default_tg_pt_gp)]
      o- portals .....
[Portals: 1]
      o- 10.22.122.10:3260
    ..... [OK]

```

## Comprobación del nodo initiator (Nodo1)

Interfaces de red:

```
[root@nodo1 ~]# ip addr show
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:f9:be:78 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::698a:e0e6:e8b0:d494/64 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 52:54:00:05:fd:ec brd ff:ff:ff:ff:ff:ff
    inet 192.168.140.116/24 brd 192.168.140.255 scope global
    dynamic noprefixroute enp7s0
        valid_lft 3154sec preferred_lft 3154sec
    inet6 fe80::81:d839:b5cd:6ff5/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Probar escritura en el sistema de archivos ext4:

```
[root@nodo1 ~]# mount /dev/sda /mnt
[root@nodo1 ~]# echo "VPD1" > /mnt/test1
[root@nodo1 ~]# umount /mnt
```

Probar escritura en el volumen lógico:

```
[root@nodo1 ~]# vgchange -ay ApacheVG
  1 logical volume(s) in volume group "ApacheVG" now active
[root@nodo1 ~]# mount /dev/ApacheVG/ApacheLV /mnt
[root@nodo1 ~]# echo "VPD2" > /mnt/test2
[root@nodo1 ~]# umount /mnt
```

## Comprobación del nodo initiator (Nodo2)

Verificar que podemos acceder a los datos escritos desde Nodo1:

```
[root@nodo2 ~]# mount /dev/sda /mnt
[root@nodo2 ~]# cat /mnt/test1
VPD1
[root@nodo2 ~]# umount /mnt
```

```
[root@nodo2 ~]# vgchange -ay ApacheVG
  1 logical volume(s) in volume group "ApacheVG" now active
[root@nodo2 ~]# mount /dev/ApacheVG/ApacheLV /mnt
[root@nodo2 ~]# cat /mnt/test2
VPD2
```

Estas pruebas confirman que:

1. Los dispositivos iSCSI se han exportado correctamente desde el nodo target
2. Ambos nodos initiator pueden conectarse a los dispositivos iSCSI
3. Se puede crear y utilizar un sistema de archivos ext4 directamente sobre un dispositivo iSCSI
4. Se puede crear un volumen lógico sobre un dispositivo iSCSI y utilizarlo desde múltiples nodos
5. Los datos escritos por un nodo son visibles por otro nodo

## Conclusiones

Las siguientes conclusiones sintetizan los aspectos más relevantes y las buenas prácticas extraídas de la puesta en marcha del servicio de almacenamiento iSCSI siguiendo la documentación oficial de RHEL 7.

### 1. Entorno virtualizado reproducible

Se ha desplegado satisfactoriamente una topología completa (target + dos initiators) utilizando KVM y redes aisladas, lo que permite repetir la práctica de forma fiable y segura.

### 2. iSCSI como solución SAN flexible

La implementación de LIO mediante `targetcli` demuestra que, con software incluido de serie en RHEL 7, es posible construir una SAN IP de bajo coste y alto rendimiento, prescindiendo de hardware *Fibre Channel* dedicado.

### 3. Control de acceso robusto

La combinación de IQN únicos, ACL por iniciador y el uso recomendado de autenticación CHAP (pendiente de implementación en la práctica) ofrece una capa sólida de seguridad para entornos multi-cliente.

### 4. Almacenamiento de bloques exportado y gestionado

Se han exportado discos iSCSI tanto como dispositivos individuales como a través de LVM, evidenciando la versatilidad del esquema backend de LIO (*fileio*, *block*, etc.) descrito en la guía de administración de almacenamiento de Red Hat.

### 6. Sistemas de archivos adecuados al caso de uso

EXT4 proporciona simplicidad y compatibilidad; XFS otorga escalabilidad y rendimiento. Elegir el sistema de archivos apropiado según la carga de trabajo y si habrá acceso concurrente es fundamental.

### 7. LVM sobre iSCSI para mayor flexibilidad

El uso de `pvcreate` → `vgcreate` → `lvcreate` sobre los LUN remotos permite ampliar o reducir capacidad sin interrumpir el servicio, conforme a la filosofía de gestión dinámica de volúmenes de RHEL 7.

### 8. Buenas prácticas operativas

- Dedicación de VLAN o red física separada para tráfico iSCSI.
- Habilitación de multipath-IO para alta disponibilidad.
- Monitorización continua de sesiones con `iscsiadm -m session` y del estado de LIO con `targetcli /iscsi ls`.
- Respaldos periódicos del archivo `/etc/target/saveconfig.json`.

Con esta práctica se ha aprendido a diseñar, desplegar y validar una solución de almacenamiento iSCSI completa, asentando las bases para evolucionar hacia arquitecturas de alta disponibilidad basadas en RHEL 7.

## Bibliografía

- [1] R. HAT, «Storage Administration Guide. RED HAT ENTERPRISE LINUX 7. Deploying and configuring single-node storage in RHEL,» [En línea]. Available: [https://docs.redhat.com/en/documentation/red\\_hat\\_enterprise\\_linux/7/html/storage\\_administration\\_guide/index](https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/7/html/storage_administration_guide/index). [Último acceso: 01 05 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].
- [6] 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].