PRÁCTICA 4. MIGRACIÓN DE MÁQUINAS VIRTUALES

Virtualización y Procesamiento Distribuido

El objetivo de esta actividad es realizar operaciones de migración de máquinas virtuales

Francisco Javier López-Dufour Morales

13-04-2025

Tabla de contenido

Introducción

El objetivo fundamental de esta práctica es realizar migraciones de máquinas virtuales entre diferentes anfitriones. La migración de máquinas virtuales requiere un almacenamiento compartido del disco, por lo que se utilizará el contenedor **`CONT_VOL_COMP`** creado en la práctica 3, que proporciona un espacio de almacenamiento compartido soportado mediante NFS.

Tanto el almacenamiento compartido como la propia migración requieren comunicación entre los anfitriones, por lo que será necesario configurar el cortafuegos para permitir dichas comunicaciones.

La migración de máquinas virtuales es un proceso que permite trasladar una VM en ejecución de un host físico a otro con un tiempo de inactividad mínimo o nulo. Existen dos tipos principales:

- **Migración en frío**: La máquina virtual se detiene en el host origen antes de transferirse al destino.
- **Migración en vivo**: La VM continúa funcionando durante el proceso de migración, minimizando el tiempo de inactividad percibido por los usuarios.

El almacenamiento compartido es un requisito fundamental para la migración en vivo, ya que ambos hosts deben poder acceder simultáneamente a los archivos de disco de la VM durante la transferencia.

Desarrollo

Para abordar esta práctica se debe haber completado la práctica 3 (Recursos de almacenamiento virtual), teniendo disponible el contenedor `CONT_VOL_COMP` funcionando correctamente. Se verifica el correcto funcionamiento del contenedor:

```
root@lq-d25:~# virsh pool-info CONT_VOL_COMP
Nombre:          CONT_VOL_COMP
UUID:            d131e98a-c98b-4bd5-ae8b-453912a010c8
Estado:          ejecutando
Persistente:     si
Autoinicio:      no
Capacidad:       395,85 GiB
Ubicación:      351,47 GiB
Disponible:      44,37 GiB
```

> Los *pools* de almacenamiento en `libvirt` (como **CONT_VOL_COMP**) son abstracciones que permiten gestionar diferentes tipos de almacenamiento, como directorios locales, LVM, iSCSI, NFS o sistemas de archivos compartidos.

Tarea 1: configurar de forma adecuada los nombres de los anfitriones

Es necesario configurar cada equipo anfitrión con un nombre de host único y completamente cualificado (FQDN) para poder identificarlo en la red y permitir la comunicación entre anfitriones.

```
# Verificar el nombre actual del host
root@lq-d25:~# hostname
lq-d25.vpd.com
```

Explicación del comando:

- `hostname`: Muestra el nombre de host actual.
- `hostname -f`: Muestra el nombre de host completamente cualificado (FQDN).

> Un nombre de dominio completamente cualificado (FQDN) es fundamental en entornos de migración por varias razones:
>
> - **Resolución DNS**: La migración depende de una correcta resolución de nombres entre hosts.
> - **Comunicación *libvirt***: El daemon libvirtd utiliza estos nombres para establecer conexiones seguras entre hosts.
> - **Coherencia del clúster**: En entornos con múltiples hosts, cada uno debe tener un nombre único y resoluble.
>
> La estructura de un FQDN (como lq-d25.vpd.com) incluye el nombre del host (lq-d25), seguido del dominio (vpc.com). Si la resolución de nombres no funciona correctamente, puede ser necesario configurar el archivo /etc/hosts en ambos hosts para asegurar que pueden comunicarse.

Configuración del Cortafuegos

La migración de máquinas virtuales requiere que ciertos puertos estén abiertos en el cortafuegos para permitir la comunicación entre los anfitriones. En Fedora Server, el servicio de cortafuegos por defecto es **`Firewalld`**.

Primero, verificamos el estado del cortafuegos:

```
root@lq-d25:~# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
     Loaded: loaded (/usr/lib/systemd/system/firewalld.service;
enabled; preset: enabled)
    Drop-In: /usr/lib/systemd/system/service.d
             └─10-timeout-abort.conf
     Active: active (running) since Fri 2025-03-28 18:20:54 WET;
26min ago
       Docs: man:firewalld(1)
   Main PID: 1046 (firewalld)
      Tasks: 2 (limit: 76221)
     Memory: 50.5M
        CPU: 322ms
     CGroup: /system.slice/firewalld.service
             └─1046 /usr/bin/python3 -sP /usr/sbin/firewalld --
nofork --nopid



mar 28 18:20:53 lq-d25.vpc.com systemd[1]: Starting
firewalld.service - firewalld - dynamic firewall daemon...
mar 28 18:20:54 lq-d25.vpc.com systemd[1]: Started
firewalld.service - firewalld - dynamic firewall daemon.
```

Explicación del comando:

- `systemctl status firewalld`: Muestra el estado del servicio Firewalld, incluyendo si está activo y en ejecución.

A continuación, identificamos la **zona activa** para la interfaz de red que utilizaremos:

```
root@lq-d25:~# firewall-cmd --get-active-zones
FedoraServer (default)
  interfaces: enp6s0
libvirt
  interfaces: virbr0
```

Explicación del comando:

- `firewall-cmd --get-active-zones`: Muestra las zonas activas y las interfaces asociadas a cada una.

Firewalld utiliza el concepto de "**zonas**" para definir niveles de confianza para las diferentes interfaces de red y sus conexiones. Cada zona puede tener diferentes reglas de filtrado:

- La zona FedoraServer está asignada a la interfaz física del sistema (enp6s0)
- La zona libvirt está configurada automáticamente para la interfaz virtual (virbr0) que utilizan las máquinas virtuales

Este modelo de zonas permite aplicar políticas de seguridad diferenciadas según el origen/destino del tráfico, lo que es especialmente útil en servidores que alojan servicios de virtualización.

Ahora, configuramos el cortafuegos para permitir la migración de máquinas virtuales. Para ello, habilitaremos los puertos necesarios:

```
# Añadir el servicio libvirt a la zona activa
sudo firewall-cmd --add-service=libvirt --permanent

# Habilitar el rango de puertos para la migración (49152-49216)
sudo firewall-cmd --add-port=49152-49216/tcp --permanent

# Habilitar puerto para tráfico SSH
sudo firewall-cmd --add-service=ssh --permanent

# Recargar la configuración del cortafuegos
sudo firewall-cmd --reload

# Verificar la configuración
sudo firewall-cmd --list-all
```

```
root@lq-d25:~# firewall-cmd --add-service=libvirt --permanent
success
root@lq-d25:~# firewall-cmd --add-port=49152-49216/tcp --permanent
success
root@lq-d25:~# firewall-cmd --add-service=ssh --permanent
Warning: ALREADY_ENABLED: ssh
success
root@lq-d25:~# firewall-cmd --reload
success
root@lq-d25:~# firewall-cmd --list-all
FedoraServer (default, active)
  target: default
  ingress-priority: 0
  egress-priority: 0
  icmp-block-inversion: no
  interfaces: 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 los comandos:

- `firewall-cmd --add-service=libvirt --permanent`: Añade el servicio predefinido libvirt a la configuración permanente del cortafuegos.
- `firewall-cmd --add-port=49152-49216/tcp --permanent`: Habilita el rango de puertos (49152-49216) que utiliza libvirt para la migración.
- `firewall-cmd --add-service=ssh --permanent`: Habilita el servicio SSH en el cortafuegos.
- `firewall-cmd --reload`: Recarga la configuración del cortafuegos, aplicando los cambios permanentes.
- `firewall-cmd --list-all`: Muestra la configuración actual del cortafuegos para verificar que los cambios se han aplicado correctamente.

> La configuración del cortafuegos es crítica para la migración por estas razones:
>
> - El rango de puertos 49152-49216 es utilizado por libvirt para establecer conexiones de migración en vivo. Estos puertos permiten:
>
> - Transferencia de la memoria RAM de la VM entre hosts
> - Sincronización del estado de CPU
> - Comunicación de control durante la migración
>
> - El servicio SSH es necesario para la autenticación segura entre hosts
> - El servicio predefinido "libvirt" incluye los puertos necesarios para la comunicación básica entre demonios libvirtd
>
> En entornos de producción, es recomendable limitar estas reglas para que solo acepten tráfico desde las IPs específicas de los otros hosts del clúster, reduciendo la superficie de exposición.

> **Nota**: Es importante habilitar estos puertos en ambos anfitriones (origen y destino) para que la migración pueda realizarse correctamente.

Tarea 2: Uso de un contenedor de almacenamiento compartido

Tal y como ya se ha indicado, el espacio compartido de almacenamiento requerido para realizar esta práctica lo proporciona el contenedor **`CONT_VOL_COMP`** creado en la práctica 3.

Deberá crear una nueva máquina virtual llamada mvp4_etiqueta_de_equipo, que en nuestro caso será mvp4_lqd25. Esta nueva máquina virtual deberá ser el resultado de clonar la máquina virtual mvp1 creada en la práctica 1 pero con una diferencia

importante: la imagen de disco de la nueva máquina mvp4_lqd25 debe almacenarse en el espacio de almacenamiento compartido proporcionado por el contenedor **CONT_VOL_COMP**

1. Verificar que el contenedor CONT_VOL_COMP está activo:

```
root@lq-d25:~# virsh pool-list --all
Nombre                 Estado   Inicio automático
----------------------------------------------------
CONT_ISOS_COMP         activo   no
CONT_VOL_COMP          activo   no
Contenedor_Particion   activo   si
default                activo   si
ISO                    activo   si
```

2. Verificar los volúmenes existentes en el contenedor CONT_VOL_COMP:

```
root@lq-d25:~# virsh vol-list CONT_VOL_COMP
 Nombre                                  Ruta
------------------------------------------------------------------------
--------------------------------------
 pc25_LQD_ANFITRION1_Vol3_p3
/var/lib/libvirt/images/COMPARTIDO/pc25_LQD_ANFITRION1_Vol3_p3
```

3. Clonar directamente la máquina virtual mvp1 a mvp4_lqd25 utilizando la sintaxis URI correcta para el volumen compartido:

```
root@lq-d25:~# virt-clone --original mvp1 --name mvp4_lqd25 --file
/var/lib/libvirt/images/COMPARTIDO/pc25_LQD_ANFITRION1_p4.qcow2 --
mac=00:16:3e:31:13:b3
Allocating 'pc25_LQD_ANFITRION1_p4.qcow2'
| 1.4 GB  00:00:00 ...
```

El clon 'mvp4_lqd25' ha sido creado exitosamente.

Explicación del comando:

- `virt-clone`: Herramienta para clonar máquinas virtuales existentes
- `--original mvp1`: Especifica la máquina virtual de origen
- `--name mvp4_lqd25`: Define el nombre de la nueva máquina virtual
- `--file /var/lib/libvirt/images/COMPARTIDO/pc25_LQD_ANFITRION1_p4.qcow2`: Especifica la ruta completa al archivo de disco en el almacenamiento compartido

- `--mac=00:16:3e:31:13:b3`: Establece una dirección MAC diferente para la interfaz de red

> **Nota**: Al utilizar la ruta completa del archivo, nos aseguramos de que virt-clone pueda crear correctamente el volumen en el espacio de almacenamiento compartido sin problemas de interpretación de sintaxis. Es importante verificar previamente que la ruta al directorio compartido es correcta.

> La clonación de máquinas virtuales y la especificación de una MAC única son aspectos críticos en entornos virtualizados:
>
> Al clonar una máquina, todos sus identificadores únicos deben cambiarse para evitar conflictos, incluida la dirección MAC.
>
> El formato de disco qcow2 utilizado (visible en la extensión del archivo) es el formato Copy-On-Write de QEMU que permite:
> 1. Instantáneas eficientes
> 2. Compresión y cifrado de datos
> 3. Optimización del espacio utilizado al escribir sólo los bloques modificados
> 4. Mejor rendimiento en operaciones de clonación

4. Verificar que el volumen se ha creado correctamente:

```
root@lq-d25:~# virsh vol-list CONT_VOL_COMP | grep pc25_LQD
pc25_LQD_ANFITRION1_p4.qcow2
/var/lib/libvirt/images/COMPARTIDO/pc25_LQD_ANFITRION1_p4.qcow2
pc25_LQD_ANFITRION1_Vol3_p3
/var/lib/libvirt/images/COMPARTIDO/pc25_LQD_ANFITRION1_Vol3_p3
pc25_LQD_ANFITRION1_Vol3_p3.qcow2
/var/lib/libvirt/images/COMPARTIDO/pc25_LQD_ANFITRION1_Vol3_p3.qcow
2
```

5. Iniciar la máquina virtual clonada:

```
root@lq-d25:~# virsh start mvp4_lqd25
Se ha iniciado el dominio mvp4_lqd25
```

6. Verificar que la máquina está operativa:

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

7. Obtener la dirección IP de la máquina virtual:

```
root@lq-d25:~# virsh domifaddr mvp4_lqd25
Nombre     dirección MAC       Protocol     Address
-------------------------------------------------------------------------------
vnet0      00:16:3e:31:13:b3    ipv4         192.168.122.124/24
```

8. Conectarse a la máquina virtual para verificar que está completamente operativa:

```
root@lq-d25:~# ssh root@192.168.122.124
Web console: https://mvp1.vpd.com:9090/ or
https://192.168.122.124:9090/


Last login: Thu Mar 20 19:28:58 2025
root@mvp1:~#
```

> **Nota**: Es normal que el arranque de la máquina demore más tiempo al tener su disco en un espacio de almacenamiento externo, a diferencia de cuando la máquina huésped tiene el disco en el host anfitrión.

Tarea 3: Migración

3.1. Generación y Configuración de Claves SSH

Para realizar la migración sin necesidad de introducir contraseñas, se utilizará la autenticación mediante clave pública/privada SSH.

```
# Generar par de claves SSH para el usuario root
sudo ssh-keygen -t rsa -b 4096
```

Explicación del comando:

- `ssh-keygen -t rsa -b 4096`: Genera un par de claves RSA de 4096 bits.
- `-t rsa`: Especifica el tipo de clave (RSA).
- `-b 4096`: Establece el tamaño de la clave (4096 bits, recomendado para mayor seguridad).

Durante el proceso de generación, se puede aceptar la ubicación predeterminada (`/root/.ssh/id_rsa`) y opcionalmente establecer una frase de contraseña (aunque para automatizar el proceso se suele dejar en blanco).

> La autenticación basada en claves SSH ofrece dos ventajas fundamentales en el contexto de la migración:

- **Automatización**: Permite ejecutar la migración sin intervención manual para introducir contraseñas
- Seguridad mejorada: Es más segura que la autenticación por contraseña, ya que:
    - La longitud de 4096 bits hace prácticamente imposible descifrar la clave privada
    - No hay contraseñas que puedan ser interceptadas durante la transmisión
    - Cada conexión utiliza un desafío criptográfico único

Es importante entender que estas claves se generan para el usuario **root**, lo que otorga privilegios completos. En entornos de producción, se recomienda crear un usuario específico con privilegios limitados sólo para operaciones de migración.

A continuación, se debe compartir la clave pública con el anfitrión de destino:

```
root@lq-d25:~# ssh-copy-id root@lq-d26.vpd.com
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new
key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if
you are prompted now it is to install the new keys
root@lq-d26.vpd.com's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'root@lq-
d26.vpd.com'"
and check to make sure that only the key(s) you wanted were added.
```

Explicación del comando:

- `ssh-copy-id root@lq-d26.vpd.com`: Copia la clave pública al archivo authorized_keys del usuario root en el anfitrión de destino.
- `root@lq-d26.vpd.com`: Especifica el usuario y el anfitrión de destino (cambiar por el FQDN del anfitrión de destino).

Se solicitará la contraseña de root en el anfitrión de destino una única vez para realizar la operación.

Para verificar que la autenticación con clave funciona correctamente:

```
root@lq-d25:~# ssh root@lq-d26.vpd.com
Last login: Fri Mar 28 20:01:11 2025 from 10.140.92.125
```

Si todo está configurado correctamente, se establecerá la conexión sin solicitar contraseña.

3.2. Migración con virsh

La migración también puede realizarse mediante la línea de comandos usando virsh:

```
# Listar las máquinas virtuales disponibles
sudo virsh list --all
```

```
# Realizar la migración en vivo
root@lq-d25:~# virsh migrate --live mvp4_lqd25 qemu+ssh://lq-
d26.vpd.com/system --verbose
root@lq-d26.vpd.com's password:
Migración: [100,00 %]
```

Explicación del comando:

- `virsh migrate`: Comando para iniciar la migración de una máquina virtual.
- `--live`: Realiza una migración en vivo, donde la máquina continúa funcionando durante el proceso.
- `mvp4_lqd25`: Nombre de la máquina virtual a migrar.
- `qemu+ssh://lq-d26.vpd.com/system`: URI del destino, especificando el protocolo (qemu+ssh), el anfitrión de destino y el tipo de conexión (/system).
- `--verbose`: Muestra información detallada durante el proceso de migración.

> El proceso de migración en vivo es una operación compleja que sigue estos pasos técnicos:
>
> 1. Fase de pre-copia: Mientras la VM sigue ejecutándose en el origen, se transfiere la memoria inicial al destino.
> 2. Transferencia iterativa: Se detectan y transfieren las páginas de memoria que han cambiado desde la última iteración.
> 3. Fase de convergencia: El proceso se repite hasta que la tasa de cambio de memoria es lo suficientemente baja.
> 4. Pausa mínima: La VM se pausa brevemente (milisegundos) en el origen, se transfiere el estado final de CPU y memoria.
> 5. Activación en el destino: La VM continúa su ejecución en el host destino.
>
> El tiempo total de migración depende principalmente del tamaño de la memoria asignada a la VM y del ancho de banda disponible entre los hosts.
>
> Según la documentación de Red Hat: "Durante una migración en vivo, libvirt establece una conexión directa entre el host de origen y el de destino. La memoria de la VM, el estado del dispositivo y el almacenamiento en disco se transfieren del origen al destino mientras la VM continúa ejecutándose en el host de origen. Cuando casi todos los datos se han transferido al destino, la VM es detenida brevemente, y el resto del estado en memoria se transfiere al destino, donde la VM se reactiva."

Para una migración más avanzada con opciones adicionales:

```
sudo virsh migrate --live --persistent --undefinesource mvp4_lqd25
qemu+ssh://lq-d26.vpd.com/system
```

Explicación de parámetros adicionales:

- `--persistent`: Hace que la máquina virtual persista en el host de destino después de reiniciarlo.
- `--undefinesource`: Elimina la definición de la máquina virtual en el host de origen después de la migración.

> Las opciones de migración avanzadas permiten diferentes estrategias según los requisitos:
>
> `--persistent`: Garantiza que la VM se reinicie automáticamente en el host destino tras un reinicio del sistema
> `--undefinesource`: Útil para migraciones permanentes, no solo temporales
>
> Otras opciones importantes no utilizadas aquí:
>
> `--unsafe`: Omite comprobaciones de seguridad para migraciones problemáticas
> `--timeout`: Establece un límite de tiempo para completar la migración
> `--compressed`: Comprime los datos transferidos para ahorrar ancho de banda
>
> La elección entre estas opciones depende del equilibrio entre minimizar el tiempo de inactividad, maximizar la fiabilidad y optimizar el rendimiento de la red.

3.3. Revocación de Accesos Temporales

Una vez completada la migración, es importante revocar los accesos temporales otorgados durante el proceso para mantener la seguridad del sistema:

```
# Eliminar la clave pública del anfitrión remoto
sudo ssh root@lq-d26.vpd.com "sed -i '/$(cat ~/.ssh/id_rsa.pub |
cut -d' ' -f2)/d' ~/.ssh/authorized_keys"
```

Explicación del comando:

- Este comando se conecta al anfitrión remoto y elimina la entrada correspondiente a nuestra clave pública del archivo authorized_keys.

Alternativamente, se puede acceder al anfitrión remoto y editar manualmente el archivo:

```
# Acceder al anfitrión remoto
sudo ssh root@lq-d26.vpd.com

# Editar el archivo authorized_keys
nano ~/.ssh/authorized_keys
```

Y eliminar la línea correspondiente a la clave pública del anfitrión de origen.

La revocación de accesos temporales tras completar la migración es una práctica de seguridad esencial que sigue el principio de privilegio mínimo:

- Reduce la superficie de ataque al eliminar credenciales que ya no son necesarias
- Evita el acceso no autorizado entre hosts en caso de compromiso de uno de ellos
- Facilita la auditoría de seguridad al mantener solo las configuraciones necesarias

Este paso es especialmente importante en entornos donde las migraciones son operaciones ocasionales, no constantes. En clústeres de alta disponibilidad donde las migraciones son frecuentes, puede ser preferible mantener una configuración permanente, pero con políticas de seguridad más estrictas.

Pruebas y Validación

Para verificar que la migración se ha realizado correctamente, se deben realizar las siguientes pruebas:

```
# En el anfitrión de destino, verificar que la máquina virtual está
en ejecución
root@lq-d26:~# virsh list --all
Id   Nombre                    Estado
-----------------------------------------
4    mvp4_lqd25                ejecutando
```

Resultado esperado:

- La máquina virtual mvp4_lqd25 debe aparecer en la lista y estar en estado "ejecutando".

```
# Verificar la información detallada de la máquina virtual
root@lq-d26:~# virsh dominfo mvp4_lqd25
Id:             4
Nombre:         mvp4_lqd25
UUID:           4b35bb33-6531-4fdf-8eaf-6b749ae432f1
Tipo de sistema operatuvo: hvm
Estado:         ejecutando
CPU(s):         1
Hora de la CPU: 36,6s
Memoria máxima: 2097152 KiB
Memoria utilizada: 2097152 KiB
Persistente:    si
Autoinicio:     desactivar
Guardar administrado: no
Modelo de seguridad: selinux
DOI de seguridad: 0
Etiqueta de seguridad: system_u:system_r:svirt_t:s0:c661,c935
(enforcing)
Messages:       tainted: potentially unsafe use of host CPU
passthrough
```

Explicación del comando:

- `virsh dominfo`: Muestra información detallada sobre la máquina virtual, incluyendo su estado, memoria asignada, y número de CPU virtuales.

> La validación post-migración debería incluir varias dimensiones de comprobación:
>
> 1. Estado: Verificar que la máquina está en ejecución y con los recursos correctamente asignados
> 2. Contextualización de seguridad: Notar que el modelo de seguridad SELinux está activo (enforcing)

3. Rendimiento: Comparar métricas de rendimiento pre y post migración (tiempo de CPU, uso de memoria)

4. Mensajes de advertencia: El mensaje "tainted" indica que se está utilizando la característica de CPU *passthrough*, que proporciona mejor rendimiento pero puede complicar futuras migraciones si los hosts tienen CPUs diferentes

En entornos de producción críticos, se recomendaría también realizar pruebas funcionales de las aplicaciones que se ejecutan dentro de la máquina virtual.

Red Hat advierte que: "Para migraciones exitosas, los procesadores del host de origen y destino deben tener la misma arquitectura. Además, los conjuntos de características de CPU deberían ser compatibles. En entornos de producción, se recomienda utilizar la función `'cpu-model'` en la definición XML de la VM para garantizar la compatibilidad de CPU entre hosts."

Verificar que se puede acceder a la máquina virtual migrada

```
root@lq-d26:~# virsh console mvp4_lqd25
Connected to domain 'mvp4_lqd25'
Escape character is ^] (Ctrl + ])
```

Explicación del comando:

- `virsh console`: Conecta a la consola de la máquina virtual, permitiendo interactuar con ella.
- Para salir de la consola, se usa la combinación de teclas Ctrl+].

Para monitorizar el rendimiento tras la migración, además de las verificaciones básicas, se pueden utilizar herramientas como:

- `sar`: Para examinar el rendimiento del sistema a lo largo del tiempo
- `iostat`: Analizar el rendimiento de E/S, que puede revelar problemas con el almacenamiento compartido
- `top/htop`: Verificar que los procesos dentro de la VM funcionan correctamente
- `libvirt-top`: Herramienta específica para monitorizar múltiples máquinas virtuales

El rendimiento de disco puede ser ligeramente diferente tras la migración debido a la latencia de red en el acceso al almacenamiento compartido desde diferentes hosts físicos.

La documentación de Red Hat señala: "Después de una migración, es recomendable verificar no solo que la VM está en ejecución, sino también sus métricas de rendimiento. Un indicador importante de problemas es un aumento significativo en la latencia de E/S o una disminución en el throughput comparado con la ejecución en el host original."

Conclusiones

La migración de máquinas virtuales entre anfitriones es una funcionalidad fundamental en entornos virtualizados, ya que permite:

- **Alta disponibilidad**: La capacidad de mover máquinas virtuales entre anfitriones sin interrupciones significativas mejora la disponibilidad del servicio.
- **Mantenimiento sin tiempo de inactividad**: Permite realizar mantenimiento en los anfitriones físicos sin necesidad de apagar las máquinas virtuales.
- **Balanceo de carga**: Facilita la distribución de la carga entre distintos anfitriones físicos.

Durante el desarrollo de esta práctica, se han aplicado varios conceptos importantes:

- Configuración adecuada de nombres de host completamente cualificados (**FQDN**).
- Gestión del cortafuegos mediante **`Firewalld`** para permitir la comunicación necesaria entre anfitriones.
- Configuración de autenticación SSH mediante claves para automatizar procesos sin comprometer la seguridad.
- Uso de almacenamiento compartido (**`NFS`**) como requisito para la migración en vivo.

La implementación de migraciones en vivo representa una habilidad esencial para administradores de sistemas en entornos virtualizados, especialmente en infraestructuras que requieren alta disponibilidad.

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. [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.

[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/. [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. [Último acceso: 30 03 2025].