

¡Entendido! Disculpa el error en la escritura anterior. Prosigamos correctamente con el último objetivo del Examen 204 de LPIC-2. Después de RAID, la otra tecnología fundamental de administración de almacenamiento a nivel avanzado es LVM. El objetivo 204.3 trata sobre la gestión de Volúmenes Lógicos (LVM).

LVM (Logical Volume Management) proporciona una capa de abstracción sobre los discos físicos, ofreciendo una flexibilidad mucho mayor en la gestión del espacio de almacenamiento. Analizaremos los conceptos de LVM, sus componentes y las herramientas para crearlos y gestionarlos, destacando las diferencias entre las ramas Debian y Red Hat en cuanto a las herramientas complementarias para redimensionar sistemas de archivos.

Comencemos con el objetivo 204.3.

LPIC-2 / Examen 204 - Administración Avanzada de Dispositivos de Almacenamiento

204.3 Gestión de Volúmenes Lógicos (LVM)

Teoría

LVM (Logical Volume Management), o Gestión de Volúmenes Lógicos, es una tecnología que añade una capa de abstracción entre los dispositivos de almacenamiento físicos (discos, particiones, arrays RAID de software o hardware) y los sistemas de archivos que los usuarios utilizan. En lugar de crear sistemas de archivos directamente en particiones físicas, se crean volúmenes lógicos (LVs) dentro de grupos de volúmenes (VGs) que se construyen a partir de volúmenes físicos (PVs).

Beneficios de LVM:

- **Flexibilidad de Redimensionamiento:** Puedes expandir (y a veces reducir) el tamaño de los volúmenes lógicos mientras están montados o con una interrupción mínima.
- **Pool de Almacenamiento:** Puedes agrupar espacio libre de varios discos o particiones en un solo grupo de volúmenes, facilitando la asignación de espacio a diferentes volúmenes lógicos según sea necesario.
- **Snapshots:** Puedes crear "instantáneas" de volúmenes lógicos, proporcionando una copia de solo lectura del estado del volumen en un momento dado (útil para copias de seguridad o pruebas).
- **Migración Sencilla:** Puedes mover volúmenes lógicos de un dispositivo físico a otro dentro del mismo grupo de volúmenes sin interrumpir el acceso.

Conceptos Clave de LVM:

1. **Dispositivo Físico (Physical Device):** Un disco duro completo (ej: `/dev/sdb`), una partición de disco (ej: `/dev/sdb1`), o un dispositivo RAID de software (`/dev/md0`) o hardware. Estos son los bloques de construcción básicos.

2. **Volumen Físico (Physical Volume - PV):** Un dispositivo físico que ha sido inicializado por LVM para ser utilizado como parte de un grupo de volúmenes. LVM escribe metadatos al inicio del dispositivo para marcarlo como un PV.
3. **Extensión Física (Physical Extent - PE):** La unidad más pequeña de asignación en un Volumen Físico. Un PV se divide en PEs de tamaño fijo (configurable, típicamente 4MB).
4. **Grupo de Volúmenes (Volume Group - VG):** Un pool de uno o más Volúmenes Físicos. Representa un pool de espacio de almacenamiento combinado del cual se asignan los volúmenes lógicos. Un VG es equivalente a un "disco virtual grande" hecho de uno o más PVs.
5. **Extensión Lógica (Logical Extent - LE):** La unidad más pequeña de asignación en un Volumen Lógico. Por defecto, el tamaño de un PE es igual al tamaño de un LE.
6. **Volumen Lógico (Logical Volume - LV):** Una "partición virtual" creada a partir del espacio disponible en un Grupo de Volúmenes. El sistema operativo ve un LV como un dispositivo de bloque regular (ej: `/dev/mapper/myvg-mylv`), sobre el cual puedes crear un sistema de archivos (ej: ext4, XFS) y montarlo. Los LVs pueden ser lineales (varios PEs contiguos o no contiguos forman un LE) o "stripeados" (similar a RAID 0, para rendimiento).

Proceso Básico de Creación de LVM:

1. **Inicializar Volúmenes Físicos (PVs):** Marcar los dispositivos físicos o particiones como PVs con `pvcreate`.
2. **Crear un Grupo de Volúmenes (VG):** Combinar uno o más PVs en un VG con `vgcreate`.
3. **Crear uno o más Volúmenes Lógicos (LVs):** Asignar espacio del VG para crear LVs con `lvcreate`.
4. **Crear un Sistema de Archivos:** Formatear el nuevo dispositivo de volumen lógico (`/dev/mapper/vgname-lvname`) con un sistema de archivos (ej: `mkfs.ext4`, `mkfs.xfs`).
5. **Montar el Volumen Lógico:** Montar el LV en un punto de montaje (Ej. 203.1).

Comandos de LVM:

La herramienta principal es `lvm` o los comandos individuales (`pv*`, `vg*`, `lv*`).

- **PV Management:**
 - `pvcreate <dispositivo> [<dispositivo>...]`: Inicializa un dispositivo como PV.
 - `pvdisplay [<dispositivo>...]`: Muestra atributos de PVs.
 - `pvscan`: Escanea dispositivos para encontrar PVs.
 - `pvremove <dispositivo>`: Elimina los metadatos de LVM de un dispositivo (¡elimina el PV!).
- **VG Management:**
 - `vgcreate <nombre_vg> <dispositivo_pv> [<dispositivo_pv>...]`: Crea un VG a partir de PVs.

- `vgdisplay [<nombre_vg>...]`: Muestra atributos de VGs.
- `vgscan`: Escanea para encontrar VGs.
- `vgextend <nombre_vg> <dispositivo_pv> [<dispositivo_pv>...]`: Añade PVs a un VG existente (expande el VG).
- `vgreduce <nombre_vg> <dispositivo_pv> [<dispositivo_pv>...]`: Elimina PVs de un VG (requiere que no contengan LVs).
- `vgremove <nombre_vg>`: Elimina un VG vacío.
- **LV Management:**
 - `lvcreate -L <tamaño> -n <nombre_lv> <nombre_vg>`: Crea un LV de un tamaño específico.
 - `lvcreate -l <num_extents> -n <nombre_lv> <nombre_vg>`: Crea un LV de un número específico de extensiones.
 - `lvdisplay [<ruta_lv>...]`: Muestra atributos de LVs. La ruta completa es `/dev/<nombre_vg>/<nombre_lv>`.
 - `lvscan`: Escanea para encontrar LVs.
 - `lvextend -L +<tamaño> <ruta_lv>`: Aumenta el tamaño de un LV en un tamaño específico.
 - `lvextend -l +<num_extents> <ruta_lv>`: Aumenta el tamaño de un LV en un número específico de extensiones.
 - `lvreduce -L -<tamaño> <ruta_lv>`: Reduce el tamaño de un LV en un tamaño específico (¡PELIGROSO!).
 - `lvresize -L <nuevo_tamaño> <ruta_lv>`: Establece el tamaño de un LV a un tamaño específico.
 - `lvremove <ruta_lv>`: Elimina un LV.

Redimensionamiento de LVs y Sistemas de Archivos:

Expandir un LV es sencillo. Después de expandir el LV, debes expandir el sistema de archivos que contiene para que pueda usar el espacio adicional.

- **Expandir LV:** `sudo lvextend -L +<tamaño> /dev/<vg>/<lv>`.
- **Expandir Sistema de Archivos (ext4):** `sudo resize2fs /dev/<vg>/<lv>`. Esto se puede hacer en caliente (mientras el sistema de archivos está montado).
- **Expandir Sistema de Archivos (XFS):** `sudo xfs_growfs /punto_de_montaje` (donde el punto de montaje es el LV montado). Esto se puede hacer en caliente.
- **Reducir LV:** Es más complejo y arriesgado. **Primero debes reducir el sistema de archivos** (lo que a menudo requiere desmontarlo y usar herramientas específicas como `resize2fs`) y luego reducir el LV. No todos los sistemas de archivos soportan reducción online. `xfs_growfs` no soporta reducción.

Snapshots de LVM:

- `lvcreate -s -L <tamaño_snapshot> -n <nombre_snapshot> <ruta_lv_original>`: Crea un snapshot de un LV existente. El tamaño del snapshot es el espacio reservado para almacenar los cambios *futuros* en el LV original.
- El snapshot representa el estado del LV original en el momento de su creación.
- Si el LV original cambia mucho después de crear el snapshot, el snapshot crecerá y puede quedarse sin espacio, volviéndose inutilizable.

Diferencias Debian vs. Red Hat (LVM):

- El paquete `lvm2` es estándar y se llama `lvm2` en ambas ramas.
- Los comandos `pv*`, `vg*`, `lv*` y su sintaxis son los mismos.
- La forma en que el sistema de inicio (`systemd`) gestiona la activación de los VGs y LVs al arrancar se basa en la detección de metadatos de LVM en los PVs y es similar en ambas ramas.
- Las diferencias principales se relacionan con las herramientas de redimensionamiento del sistema de archivos, que dependen del tipo de sistema de archivos por defecto (`resize2fs` para ext4 en Debian, `xfs_growfs` para XFS en Red Hat).

LPIC-2 / Examen 204 - Administración Avanzada de Dispositivos de Almacenamiento - Ejercicios

*Nota: La creación y gestión de componentes LVM (PVs, VGs, LVs) modifica o destruye datos en los dispositivos subyacentes. **Realiza estos ejercicios SIEMPRE en un entorno de prueba (VM) utilizando discos virtuales adicionales o dispositivos loopback dedicados. NUNCA en discos con datos importantes.** Necesitarás privilegios de superusuario (`sudo`).*

Ejercicio 4.3.1: Verificando la Instalación de LVM y Explorando Comandos

- **Objetivo:** Asegurarse de que LVM está instalado y familiarizarse con sus comandos.
- **Requisitos:** Acceso a la línea de comandos. Instalar `lvm2` si no está (`sudo apt install lvm2` o `sudo dnf install lvm2`).
- **Desarrollo Paso a Paso:**
 1. Abre una terminal.
 2. **Verifica la instalación:** Ejecuta `which vgcreate`. Si se muestra la ruta, LVM está instalado.
 3. **Muestra los comandos principales:** Ejecuta `sudo lvm help`.
 4. **Lee las páginas del manual (recomendado):** Ejecuta `man pvcreate`, `man vgcreate`, `man lvcreate`.
 5. **Verifica si ya existen componentes LVM:** Ejecuta `sudo pvdisplay`, `sudo vgdisplay`, `sudo lvdisplay`. Si tu sistema se instaló usando LVM, verás PVs, VGs y LVs existentes.

Ejercicio 4.3.2: (Conceptual) Preparando Dispositivos para LVM

- **Objetivo:** Entender cómo preparar discos o particiones para usarlos como Volúmenes Físicos (PVs).
- **Requisitos:** Acceso a la línea de comandos. Privilegios de superusuario (sudo). **VM de prueba con discos virtuales adicionales (ej: /dev/sdb, /dev/sdc) o archivos loopback.** Los dispositivos deben estar vacíos o contener datos prescindibles.
- **Desarrollo Paso a Paso:**
 1. Abre una terminal.
 2. **(Si usas discos virtuales adicionales):** Identifica los nombres de los discos (ej: /dev/sdb, /dev/sdc).
 3. **(Si usas archivos loopback):** Crea archivos y asocia dispositivos loop (Ej. 4.1.2). Anota los dispositivos asignados (ej: /dev/loop0, /dev/loop1).
 4. **(Conceptual - Opcional pero recomendado):** Crea particiones en los dispositivos dedicadas a LVM. Usa `fdisk` o `parted`. Crea una partición primaria (n, p, 1, default, default). **Cambia el tipo de partición a "Linux LVM"**. En `fdisk`, usa el comando `t`, código 8e. En `parted`, usa `set 1 lvm on`. Escribe los cambios (w en `fdisk`). Anota los nombres de las particiones (ej: /dev/sdb1, /dev/sdc1 o /dev/loop0p1, /dev/loop1p1).

Ejercicio 4.3.3: (Conceptual) Creando Volúmenes Físicos (PVs), Grupos de Volúmenes (VGs) y Volúmenes Lógicos (LVs)

- **Objetivo:** Entender el proceso y los comandos para crear la estructura LVM.
- **Requisitos:** Dispositivos preparados (Ej. 4.3.2). Privilegios de superusuario (sudo). **VM de prueba.**
- **Desarrollo Paso a Paso:**
 1. Abre una terminal.
 2. **Comprende el comando pvcreate:**
 Bash


```
# Inicializa los dispositivos como PVs
# sudo pvcreate /dev/sdb1 /dev/sdc1
# o con dispositivos loopback:
# sudo pvcreate /dev/loop0p1 /dev/loop1p1
```

 - Esto escribe metadatos de LVM al inicio de los dispositivos.
 3. **Comprende el comando vgcreate:**
 Bash


```
# Crea un Grupo de Volúmenes llamado myvg a partir de los PVs
# sudo vgcreate myvg /dev/sdb1 /dev/sdc1
# o con dispositivos loopback:
# sudo vgcreate myvg /dev/loop0p1 /dev/loop1p1
```

 - Esto crea el VG myvg y añade los PVs a él.
 4. **Comprende el comando lvcreate:**
 Bash

```
# Crea un Volumen Lógico llamado mylv de 1GB en el VG myvg
# sudo lvcreate -L 1G -n mylv myvg

# Crea un Volumen Lógico llamado otro_lv usando el 50% del espacio
libre en myvg
# sudo lvcreate -l 50%FREE -n otro_lv myvg
```

- **-L <tamaño> o -l <porcentaje>:** Especifica el tamaño.
- **-n <nombre_lv>:** Especifica el nombre del LV.
- **<nombre_vg>:** El VG del que se asigna espacio.
- Esto crea el dispositivo de bloque `/dev/mapper/myvg-mylv`.

5. **(Conceptual):** Después de crear el LV, se puede formatear con un sistema de archivos (ej: `sudo mkfs.ext4 /dev/mapper/myvg-mylv`) y luego montar (Ej. 3.1.3).

Ejercicio 4.3.4: Visualizando Componentes LVM

- **Objetivo:** Usar los comandos `pvdisk`, `vgdisk` y `lvdisk` para ver los detalles de la estructura LVM.
- **Requisitos:** LVM instalado. Idealmente, tener algún componente LVM (del sistema operativo o de ejercicios de prueba).
- **Desarrollo Paso a Paso:**
 1. Abre una terminal.
 2. **Muestra los PVs:** Ejecuta `sudo pvdisk`. Verás detalles de cada PV (nombre, tamaño, UUID, VG al que pertenece, PEs libres/usados).
 3. **Muestra los VGs:** Ejecuta `sudo vgdisk`. Verás detalles de cada VG (nombre, tamaño, PVs miembros, LVs contenidos, PEs libres/usados en el VG).
 4. **Muestra los LVs:** Ejecuta `sudo lvdisk`. Verás detalles de cada LV (nombre, VG al que pertenece, tamaño, UUID, ruta del dispositivo - `/dev/mapper/...`).
 5. **Muestra un resumen rápido:** Ejecuta `sudo pvs`, `sudo vgs`, `sudo lvs`. Son versiones más concisas de los comandos `*disk`.
 6. **Escanea para encontrar componentes LVM (útil después de añadir discos):** Ejecuta `sudo pvscan`, `sudo vgscan`, `sudo lvscan`.

Ejercicio 4.3.5: (Conceptual) Expandiendo un Volumen Lógico y su Sistema de Archivos

- **Objetivo:** Entender el proceso para aumentar el tamaño de un LV y el sistema de archivos que contiene.
- **Requisitos:** Un VG con espacio libre. Un LV existente con un sistema de archivos. Privilegios de superusuario (sudo). **VM de prueba.**
- **Desarrollo Paso a Paso:**
 1. Abre una terminal.
 2. **Verifica el espacio libre en el VG:** Ejecuta `sudo vgdisk <nombre_vg>`. Observa la línea `Free PE / Size`.

3. **Expande el Volumen Lógico (ej: añade 500MB):** Ejecuta `sudo lvextend -L +500M /dev/<vg>/<lv>`. Puedes usar `-L +<tamaño> o -l +<número_extents>`.
4. **Verifica que el LV creció:** Ejecuta `sudo lvdisplay /dev/<vg>/<lv>`. El tamaño debería ser mayor.
5. **Expande el Sistema de Archivos (ext4):** Si el LV tiene un sistema de archivos ext4 y está montado, ejecuta `sudo resize2fs /dev/<vg>/<lv>`. Se expandirá para llenar el LV. Si no está montado, quita la opción `-P`.
6. **Expande el Sistema de Archivos (XFS):** Si el LV tiene un sistema de archivos XFS y está montado, ejecuta `sudo xfs_growfs /punto_de_montaje` (donde `/punto_de_montaje` es donde está montado el LV).
7. **Verifica que el sistema de archivos creció:** Ejecuta `df -h /punto_de_montaje`. El tamaño reportado debería ser mayor.
8. **(Concepto - Reducción):** Reducir implica **primero** reducir el sistema de archivos (desmontado, con herramientas específicas) y **luego** reducir el LV (`lvreduce`). **MUY RIESGOSO** y no todos los sistemas de archivos lo soportan.

Ejercicio 4.3.6: (Conceptual) Creando un Snapshot de LVM

- **Objetivo:** Entender cómo crear una instantánea de un LV.
- **Requisitos:** Un VG con suficiente espacio libre. Un LV existente con datos. Privilegios de superusuario (sudo). **VM de prueba.**
- **Desarrollo Paso a Paso:**
 1. Abre una terminal.
 2. **Comprende el comando `lvcreate -s`:**
 Bash


```
# Crea un snapshot llamado mylv_snap de 1GB del LV mylv en el VG myvg
# sudo lvcreate -s -L 1G -n mylv_snap /dev/myvg/mylv
```

 - `-s`: Indica que es un snapshot.
 - `-L <tamaño>`: El espacio reservado para los cambios.
 - `-n <nombre_snapshot>`: Nombre del snapshot.
 - `<ruta_lv_original>`: El LV del que se toma el snapshot.
 3. **(Conceptual):** Después de crearlo, el snapshot aparece como un nuevo dispositivo de bloque (`/dev/mapper/myvg-mylv_snap`). Puedes montar este dispositivo en modo solo lectura (`sudo mount -o ro /dev/mapper/myvg-mylv_snap /mnt/snapshot`) para acceder a los datos como estaban en el momento de la creación.
 4. **(Concepto):** El snapshot crecerá a medida que cambien los datos en el LV original. Si el espacio reservado para el snapshot (`-L 1G` en el ejemplo) se llena, el snapshot se vuelve inutilizable.

Ejercicio 4.3.7: Limpieza (Componentes LVM de Prueba)

- **Objetivo:** Eliminar los componentes LVM de prueba.
- **Requisitos:** Los componentes LVM de prueba creados (PVs, VGs, LVs). Privilegios de superusuario (sudo). **VM de prueba.**
- **Desarrollo Paso a Paso:**
 1. Abre una terminal.
 2. **Desmonta cualquier sistema de archivos montado en los LVs de prueba:** `sudo umount /mnt/my_lvm_fs` (o el punto de montaje correcto).
 3. **Elimina los LVs de prueba:** `sudo lvremove /dev/<vg>/<lv>` (ej: `sudo lvremove /dev/myvg/mylv`). Confirma la eliminación.
 4. **Elimina los VGs de prueba:** `sudo vgremove <nombre_vg>` (ej: `sudo vgremove myvg`). Confirma la eliminación. (El VG debe estar vacío).
 5. **Elimina los PVs de prueba:** `sudo pvremove <dispositivo>` (ej: `sudo pvremove /dev/sdb1 /dev/sdc1` o los dispositivos loopback).
 6. **(Si usaste dispositivos loopback):** Desasócialos (`sudo losetup -d <dispositivo_loop>`) y elimina los archivos de imagen (`rm ~/disk1.img ~/disk2.img`).