



Red Hat Enterprise Linux 6 Gestión del Administrador de volumen lógico

Guía del Gestor de LVM
Edición 1

Landmann

Red Hat Enterprise Linux 6 Gestión del Administrador de volumen lógico

Guía del Gestor de LVM Edición 1

Landmann
rlandmann@redhat.com

Legal Notice

Copyright © 2011 Red Hat, Inc. and others.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, MetaMatrix, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux® is the registered trademark of Linus Torvalds in the United States and other countries.

Java® is a registered trademark of Oracle and/or its affiliates.

XFS® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack® Word Mark and OpenStack Logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

Este libro describe el Administrador de volúmenes lógicos LVM e incluye información sobre cómo ejecutar LVM en entornos de cluster.

Table of Contents

Introduction	6
1. About This Guide	6
2. Audience	6
3. Software Versions	6
4. Related Documentation	6
5. We Need Feedback!	7
6. Convenciones del Documento	7
6.1. Convenciones tipográficas	7
6.2. Convenciones del documento	9
6.3. Notas y Advertencias	9
Capítulo 1. El administrador de volúmenes lógicos LVM	11
1.1. Funcionalidades nuevas y cambiadas	11
1.1.1. Funcionalidades nuevas y cambiadas para Red Hat Enterprise Linux 6.0	11
1.1.2. Funcionalidades nuevas y cambiadas para Red Hat Enterprise Linux 6.1	12
1.2. Volúmenes lógicos	12
1.3. Sinopsis de la arquitectura LVM	13
1.4. Administrador de volúmenes lógicos en cluster (CLVM)	14
1.5. Sinopsis del documento	16
Capítulo 2. Componentes de LVM	18
2.1. Volúmenes físicos	18
2.1.1. Distribución de un volumen físico LVM	18
2.1.2. Múltiples particiones en un disco	19
2.2. Grupos de volúmenes	19
2.3. Volúmenes lógicos LVM	20
2.3.1. Volúmenes lineales	20
2.3.2. Volúmenes lógicos entrelazados	21
2.3.3. Volúmenes lógicos en espejo	22
2.3.4. Volúmenes de instantáneas	23
Capítulo 3. Visión General de la Gestión de LVM	26
3.1. Creación de volúmenes LVM en un cluster	26
3.2. Sinopsis de la creación de volúmenes lógicos	26
3.3. Cómo aumentar el tamaño de un sistema de archivos en un volumen lógico	27
3.4. Copia de seguridad del volumen lógico	27
3.5. Registro	28
Capítulo 4. Gestión de LVM mediante comandos de CLI	29
4.1. Uso de los comandos	29
4.2. Administración de volúmenes físicos	30
4.2.1. Creación de volúmenes físicos	31
4.2.1.1. Cómo establecer el tipo de partición	31
4.2.1.2. Cómo inicializar los volúmenes físicos	31
4.2.1.3. Cómo buscar Dispositivos de Bloque	31
4.2.2. Cómo ver los volúmenes físicos	32
4.2.3. Cómo prevenir la asignación en un volumen físico	33
4.2.4. Cómo redimensionar un volumen físico	33
4.2.5. Cómo retirar volúmenes físicos	33
4.3. Administración del grupo de volúmenes	34
4.3.1. Cómo crear grupos de volúmenes	34
4.3.2. Cómo crear grupos de volúmenes en un cluster	35
4.3.3. Cómo agregar volúmenes físicos al grupo de volúmenes	35

4.3.4. Cómo mostrar los grupos de volúmenes	36
4.3.5. Cómo explorar discos en búsqueda de grupos de volúmenes para construir el archivo de caché	36
4.3.6. Cómo retirar un volumen físico de un grupo de volúmenes	37
4.3.7. Cómo cambiar los parámetros de un grupo de volúmenes	38
4.3.8. Cómo activar y desactivar grupos de volúmenes	38
4.3.9. Cómo retirar grupos de volúmenes	38
4.3.10. Separación de un grupo de volúmenes	38
4.3.11. Cómo combinar grupos de volúmenes	39
4.3.12. Cómo crear copias de seguridad de los metadatos del grupo de volúmenes	39
4.3.13. Cómo cambiar el nombrar de un grupo de volúmenes	39
4.3.14. Cómo desplazar un grupo de volúmenes a otro sistema	39
4.3.15. Cómo recrear un directorio de grupo de volúmenes	40
4.4. Administración del volumen lógico	40
4.4.1. Cómo crear volúmenes lógicos lineales	40
4.4.2. Cómo crear volúmenes entrelazados	42
4.4.3. Cómo crear volúmenes en espejo	42
4.4.3.1. Política de fallas de volumen lógico en espejo	45
4.4.3.2. División fuera de una imagen redundante de un volumen lógico en espejo	46
4.4.3.3. Cómo reparar volúmenes en espejo	46
4.4.3.4. Cómo cambiar la configuración del volumen en espejo	47
4.4.4. Cómo crear volúmenes de instantáneas	47
4.4.5. Cómo fusionar volúmenes de instantáneas	48
4.4.6. Números de dispositivos persistentes	49
4.4.7. Cómo redimensionar volúmenes lógicos	49
4.4.8. Cómo cambiar los parámetros de un grupo de volúmenes lógico	49
4.4.9. Cómo renombrar volúmenes lógicos	50
4.4.10. Cómo retirar volúmenes lógicos	50
4.4.11. Cómo ver los volúmenes lógicos	50
4.4.12. Cómo aumentar los volúmenes lógicos	51
4.4.12.1. Cómo extender un volumen entrelazado	51
4.4.12.2. Extender un volumen lógico con la política de asignación cling	53
4.4.13. Cómo reducir volúmenes lógicos	54
4.5. Cómo controlar la exploración de dispositivos LVM a través de filtros	55
4.6. Asignación de datos en línea	55
4.7. Cómo activar volúmenes lógicos en nodos individuales en un cluster	56
4.8. Reporte personalizado para LVM	56
4.8.1. Control del formato	57
4.8.2. Selección de objetos	58
El comando pvs	59
El comando vgs	61
El comando lvs	63
4.8.3. Cómo ordenar los reportes de LVM	66
4.8.4. Cómo especificar unidades	67
Capítulo 5. Ejemplos de configuración de LVM	69
5.1. Cómo crear un volumen lógico LVM en tres discos	69
5.1.1. Cómo crear los volúmenes físicos	69
5.1.2. Cómo crear el grupo de volúmenes	69
5.1.3. Cómo crear el volumen lógico	69
5.1.4. Cómo crear un sistema de archivos	69
5.2. Cómo crear un volumen lógico entrelazado	70
5.2.1. Cómo crear los volúmenes físicos	70

5.2.2. Cómo crear el grupo de volúmenes	70
5.2.3. Cómo crear el volumen lógico	71
5.2.4. Cómo crear un sistema de archivos	71
5.3. Separación de un grupo de volúmenes	71
5.3.1. Cómo determinar el espacio libre	72
5.3.2. Cómo desplazar los datos	72
5.3.3. Cómo dividir en grupo de volúmenes	72
5.3.4. Cómo crear un nuevo volumen lógico	73
5.3.5. Cómo crear el sistema de archivos y cómo montar el nuevo volumen lógico	73
5.3.6. Cómo activar y montar el volumen lógico original	73
5.4. Cómo retirar un disco de un volumen lógico	73
5.4.1. Cómo desplazar las extensiones a volúmenes físicos existentes	73
5.4.2. Cómo desplazar extensiones a un nuevo disco	74
5.4.2.1. Cómo crear un nuevo volumen físico	75
5.4.2.2. Cómo añadir el nuevo volumen físico al grupo de volúmenes	75
5.4.2.3. Cómo desplazar los datos	75
5.4.2.4. Cómo retirar el volumen físico antiguo del grupo de volúmenes	75
5.5. Cómo crear un volumen lógico LVM en un cluster	75
Capítulo 6. Solución de errores en LVM	79
6.1. Diagnósticos en la solución de errores	79
6.2. Cómo ver la información en dispositivos fallidos	79
6.3. Cómo recuperarse de una falla de un espejo LVM	80
6.4. Cómo recuperar los metadatos de un volumen físico	83
6.5. Cómo reemplazar un volumen físico perdido	85
6.6. Cómo retirar los volúmenes físicos perdidos desde un grupo de volúmenes	85
6.7. Extensiones libres insuficientes para un volumen lógico	85
Capítulo 7. Gestión de LVM con la interfaz gráfica de LVM	87
Mapeador de Dispositivos	88
A.1. Tabla de Mapas de Dispositivo	88
A.1.1. Destino de mapa lineal	89
A.1.2. Destino de mapas entrelazados	89
A.1.3. Espejo de destino de mapa	91
A.1.4. Destinos de mapa instantánea e instantánea-origen	93
A.1.5. Destino de mapa error	95
A.1.6. Destino de mapas cero	96
A.1.7. Destino de mapas multirutas	96
A.1.8. Destino de mapas crypt	98
A.2. Comando dmsetup	99
A.2.1. Comando info dmsetup	99
A.2.2. El comando dmsetup ls	101
A.2.3. Comando dmsetup status	102
A.2.4. Comando dmsetup deps	102
A.3. Soporte de mapeador de dispositivo para el Gestor de dispositivos udev	103
A.3.1. integración udev con el Mapeador de Dispositivos	103
A.3.2. Comandos e Interfaces que soportan udev	105
Archivos de configuración LVM	107
B.1. Archivos de configuración LVM	107
B.2. Archivo lvm.conf de ejemplo	108
Etiquetas de objetos LVM	121
C.1. Cómo añadir y retirar etiquetas de objetos	121
C.2. Etiquetas de host	121
C.3. Cómo controlar la activación con etiquetas	122

Metadatos del grupo de volúmenes LVM	123
D.1. La etiqueta del volumen físico	123
D.2. Contenido de metadatos	123
D.3. Metadatos de ejemplo	124
Historial de revisiones	127
Índice	128
Símbolos	128
A	128
C	128
D	130
E	130
F	131
G	131
I	132
L	132
M	132
N	132
P	133
R	133
S	133
T	134
U	134
V	134

Introduction

1. About This Guide

This book describes the Logical Volume Manager (LVM), including information on running LVM in a clustered environment.

2. Audience

This book is intended to be used by system administrators managing systems running the Linux operating system. It requires familiarity with Red Hat Enterprise Linux 6 and GFS2 file system administration.

3. Software Versions

Tabla 1. Software Versions

Software	Description
RHEL 6	refers to RHEL 6 and higher
GFS2	refers to GFS2 for RHEL 6 and higher

4. Related Documentation

For more information about using Red Hat Enterprise Linux, refer to the following resources:

- ▶ *Installation Guide* — Documents relevant information regarding the installation of Red Hat Enterprise Linux 6.
- ▶ *Deployment Guide* — Documents relevant information regarding the deployment, configuration and administration of Red Hat Enterprise Linux 6.
- ▶ *Storage Administration Guide* — Provides instructions on how to effectively manage storage devices and file systems on Red Hat Enterprise Linux 6.

For more information about the High Availability Add-On and the Resilient Storage Add-On for Red Hat Enterprise Linux 6, refer to the following resources:

- ▶ *High Availability Add-On Overview* — Provides a high-level overview of the Red Hat High Availability Add-On.
- ▶ *Cluster Administration* — Provides information about installing, configuring and managing the Red Hat High Availability Add-On,
- ▶ *Global File System 2: Configuration and Administration* — Provides information about installing, configuring, and maintaining Red Hat GFS2 (Red Hat Global File System 2), which is included in the Resilient Storage Add-On.
- ▶ *DM Multipath* — Provides information about using the Device-Mapper Multipath feature of Red Hat Enterprise Linux 6.
- ▶ *Load Balancer Administration* — Provides information on configuring high-performance systems and services with the Load Balancer Add-On, a set of integrated software components that provide Linux Virtual Servers (LVS) for balancing IP load across a set of real servers.
- ▶ *Release Notes* — Provides information about the current release of Red Hat products.

High Availability Add-On documentation and other Red Hat documents are available in HTML, PDF, and RPM versions on the Red Hat Enterprise Linux Documentation CD and online at <http://www.redhat.com/docs/>.

5. We Need Feedback!

If you find a typographical error in this manual, or if you have thought of a way to make this manual better, we would love to hear from you! Please submit a report in Bugzilla: <http://bugzilla.redhat.com/> against the product **Red Hat Enterprise Linux 6** and the component **doc-Logical_Volume_Manager**. When submitting a bug report, be sure to mention the manual's identifier: **Logical_Volume_Manager_Administration(EN)-6 (2011-05-19-15:20)**.

If you have a suggestion for improving the documentation, try to be as specific as possible when describing it. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.

6. Convenciones del Documento

Este manual utiliza varias convenciones para resaltar algunas palabras y frases y llamar la atención sobre ciertas partes específicas de información.

En ediciones PDF y de papel, este manual utiliza tipos de letra procedentes de [Liberation Fonts](#). Liberation Fonts también se utilizan en ediciones de HTML si están instalados en su sistema. Si no, se muestran tipografías alternativas pero equivalentes. Nota: Red Hat Enterprise Linux 5 y siguientes incluyen Liberation Fonts predeterminadas.

6.1. Convenciones tipográficas

Se utilizan cuatro convenciones tipográficas para llamar la atención sobre palabras o frases específicas. Dichas convenciones y las circunstancias en que se aplican son las siguientes:

Negrita monoespaciado

Utilizado para resaltar la entrada del sistema, incluyendo los comandos de shell, nombres de archivos y rutas. También sirve para resaltar teclas y combinaciones de teclas. Por ejemplo:

Para ver el contenido del archivo **my_next_bestselling_novel** en su directorio actual de trabajo, escriba el comando **cat my_next_bestselling_novel** en el intérprete de comandos de shell y pulse **Enter** para ejecutar el comando.

El ejemplo anterior incluye un nombre de archivo, un comando de shell y una tecla. Todo se presenta en negrita-monoespaciado y distinguible gracias al contexto.

Las combinaciones de teclas se pueden distinguir de las individuales con el signo más que conecta cada parte de la combinación de tecla. Por ejemplo:

Pulse **Enter** para ejecutar el comando.

Pulse **Ctrl+Alt+F2** para pasar a una terminal virtual.

El primer ejemplo resalta una tecla particular a pulsar. El segundo ejemplo, resalta una combinación de teclas: un set de tres teclas pulsadas simultáneamente.

Si se discute el código fuente, los nombres de las clase, los métodos, las funciones, los nombres de variables y valores de retorno mencionados dentro de un párrafo serán presentados en **Negrita-**

monoespaciado. Por ejemplo:

Las clases de archivo relacionadas incluyen **filename** para sistema de archivos, **file** para archivos y **dir** para directorios. Cada clase tiene su propio conjunto asociado de permisos.

Negrita proporcional

Esta denota palabras o frases encontradas en un sistema, incluyendo nombres de aplicación, texto de cuadro de diálogo, botones etiquetados, etiquetas de cajilla de verificación y botón de radio; títulos de menú y títulos del submenú. Por ejemplo:

Seleccione **Sistema** → **Preferencias** → **Ratón** desde la barra del menú principal para lanzar **Preferencias de ratón**. En la pestaña de **Botones**, seleccione la cajilla de **ratón de mano izquierda** y luego haga clic en **Cerrar** para cambiar el botón principal del ratón de la izquierda a la derecha (adecuando el ratón para la mano izquierda).

Para insertar un carácter especial en un archivo **gedit**, seleccione **Aplicaciones** → **Accesorios** → **Mapa de caracteres** de la barra del menú. Luego, seleccione **Búsqueda** → **Buscar...** de la barra del menú de **Mapa de caracteres**, escriba el nombre del carácter en el campo de **Búsqueda** y haga clic en **Siguiente**. El carácter que buscó será resaltado en la **Tabla de caracteres**. Haga doble clic en ese carácter resaltado para colocarlo en el campo de **Texto a copiar** y luego haga clic en el botón **Copiar**. Ahora regrese al documento y elija **Modificar** → **Pegar** de la barra de menú de **gedit**.

El texto anterior incluye nombres de aplicación; nombres y elementos del menú de todo el sistema; nombres de menú de aplicaciones específicas y botones y texto hallados dentro de una interfaz gráfica de usuario, todos presentados en negrita proporcional y distinguibles por contexto.

Itálicas-negrita monoespaciado o Itálicas-negrita proporcional

Ya sea negrita monoespaciado o negrita proporcional, la adición de itálicas indica texto reemplazable o variable. Las itálicas denotan texto que usted no escribe literalmente o texto mostrado que cambia dependiendo de la circunstancia. Por ejemplo:

Para conectar a una máquina remota utilizando ssh, teclee **ssh *nombre de usuario@dominio.nombre*** en un intérprete de comandos de shell. Si la máquina remota es **example.com** y su nombre de usuario en esa máquina es john, teclee **ssh *john@example.com***.

El comando **mount -o remount *file-system*** remonta el sistema de archivo llamado. Por ejemplo, para volver a montar el sistema de archivo **/home**, el comando es **mount -o remount */home***.

Para ver la versión de un paquete actualmente instalado, utilice el comando **rpm -q *paquete***. Éste entregará el resultado siguiente: ***paquete-versión-lanzamiento***.

Observe que las palabras resaltadas en itálicas — nombre de usuario, dominio.nombre, sistema de archivo, paquete, versión y lanzamiento. Cada palabra es un marcador de posición, ya sea de texto a ingresar cuando se ejecuta un comando o para un texto ejecutado por el sistema.

Aparte del uso estándar para presentar el título de un trabajo, las itálicas denotan el primer uso de un término nuevo e importante. Por ejemplo:

Publican es un sistema de publicación de *DocBook*.

6.2. Convenciones del documento

Los mensajes de salida de la terminal o fragmentos de código fuente se distinguen visualmente del texto circundante.

Los mensajes de salida enviados a una terminal se muestran en **romano monoespaciado** y se presentan así:

```
books      Desktop  documentation  drafts  mss    photos  stuff  svn
books_tests Desktop1  downloads      images  notes  scripts svgs
```

Los listados de código fuente también se muestran en **romano monoespaciado**, pero se presentan y resaltan de la siguiente manera:

```
static int kvm_vm_ioctl_deassign_device(struct kvm *kvm,
                                       struct kvm_assigned_pci_dev *assigned_dev)
{
    int r = 0;
    struct kvm_assigned_dev_kernel *match;

    mutex_lock(&kvm->lock);

    match = kvm_find_assigned_dev(&kvm->arch.assigned_dev_head,
                                  assigned_dev->assigned_dev_id);
    if (!match) {
        printk(KERN_INFO "%s: device hasn't been assigned before, "
                  "so cannot be deassigned\n", __func__);
        r = -EINVAL;
        goto out;
    }

    kvm_deassign_device(kvm, match);

    kvm_free_assigned_device(kvm, match);

out:
    mutex_unlock(&kvm->lock);
    return r;
}
```

6.3. Notas y Advertencias

Finalmente, utilizamos tres estilos visuales para llamar la atención sobre la información que de otro modo se podría pasar por alto.



Nota

Una nota es una sugerencia, atajo o enfoque alternativo para una tarea determinada. Ignorar una nota no debería tener consecuencias negativas, pero podría perderse de algunos trucos que pueden facilitarle las cosas.



Importante

Los cuadros con el título de importante dan detalles de cosas que se pueden pasar por alto fácilmente: cambios de configuración únicamente aplicables a la sesión actual, o servicios que necesitan reiniciarse antes de que se aplique una actualización. Ignorar estos cuadros no ocasionará pérdida de datos, pero puede causar enfado y frustración.



Aviso

Las advertencias no deben ignorarse. Ignorarlas muy probablemente ocasionará pérdida de datos.

Capítulo 1. El administrador de volúmenes lógicos LVM

Este capítulo proporciona un resumen de las funcionalidades del Administrador de volúmenes lógicos LVM que son nuevas en la versión inicial y subsiguientes de Red Hat Enterprise Linux 6. Más adelante, este capítulo proporciona una introducción de alto nivel de los componentes del Administrador de volúmenes lógicos (LVM).

1.1. Funcionalidades nuevas y cambiadas

Esta sección lista las funcionalidades nuevas y cambiadas del Administrador de volumen lógico que se incluyen en el lanzamiento inicial y subsiguientes de Red Hat Enterprise Linux 6.

1.1.1. Funcionalidades nuevas y cambiadas para Red Hat Enterprise Linux 6.0

Red Hat Enterprise Linux 6.0 incluye la siguiente documentación, actualizaciones y cambios.

- ▶ Puede definir cómo se comporta un volumen lógico en espejo en el evento de una falla con los parámetros **mirror_image_fault_policy** y **mirror_log_fault_policy** en la sección **activation** del archivo **lvm.conf**. Cuando este parámetro se establece a **remove**, el sistema intenta retirar el dispositivo fallido y se ejecutará sin él. Cuando el parámetro se establece a **allocate**, el sistema intenta retirar el dispositivo fallido y trata de asignar espacio en un nuevo para ser un nuevo remplazo para el dispositivo fallido, esta política actúa como la política **remove** si no se ha asignado un dispositivo o espacio apropiado para el remplazo. Para obtener información sobre políticas de fallas en el espejo LVM, consulte la [Sección 4.4.3.1, “Política de fallas de volumen lógico en espejo”](#).
- ▶ Para el lanzamiento de Red Hat Enterprise Linux 6, la pila de E/S de Linux ha sido mejorada para procesar información de límite de E/S suministrada por el proveedor. De esta manera permite a las herramientas de administración de almacenaje, incluyendo LVM, para optimizar la colocación de datos y acceso. Este soporte puede ser desactivado si cambia los valores predeterminados de **data_alignment_detection** y **data_alignment_offset_detection** en el archivo **lvm.conf**, aunque la desactivación del soporte no es recomendable.
Para obtener información sobre alineación de datos en LVM como también sobre el cambio de valores predeterminados de **data_alignment_detection** y **data_alignment_offset_detection**, consulte la documentación en línea para el archivo **/etc/lvm/lvm.conf**, el cual también se documenta en el [Apéndice B, Archivos de configuración LVM](#). Para obtener información general de soporte para Pila de E/S y límites de E/S en Red Hat Enterprise Linux 6, consulte la *Guía de administración de almacenaje*.
- ▶ En Red Hat Enterprise Linux 6, el Mapeador de Dispositivos proporciona soporte directo para integración **udev**. Este sincroniza el Mapeador de Dispositivos con todo el procesamiento **udev** relacionado con dispositivos de Mapeador de Dispositivos, incluyendo dispositivos LVM. Para obtener información sobre soporte de Mapeador de Dispositivos para el administrador de dispositivos **udev**, consulte la [Sección A.3, “Soporte de mapeador de dispositivo para el Gestor de dispositivos udev”](#).
- ▶ Para el lanzamiento de Red Hat Enterprise Linux 6, puede usar el comando **lvconvert --repair** para reparar un espejo tras la falla de un disco. Esto trae al espejo a un estado consistente. Para obtener información sobre el comando **lvconvert --repair**, consulte la [Sección 4.4.3.3, “Cómo reparar volúmenes en espejo”](#).
- ▶ A partir del lanzamiento de Red Hat Enterprise Linux 6, usted podrá utilizar la opción **--merge** del comando **lvconvert** para fusionar una instantánea en su volumen de origen. Para obtener información sobre fusión de instantáneas, consulte la [Sección 4.4.5, “Cómo fusionar volúmenes de instantáneas”](#).
- ▶ A partir del lanzamiento de Red Hat Enterprise Linux 6, usted puede utilizar el argumento **--**

splitmirrors del comando **lvconvert** para dividir una imagen redundante de un volumen lógico para formar un nuevo volumen lógico. Para obtener información sobre el uso de esta opción, consulte la [Sección 4.4.3.2, “División fuera de una imagen redundante de un volumen lógico en espejo”](#).

- Puede crear un registro de espejo para un dispositivo lógico en espejo que esté él mismo en espejo, mediante el argumento **--mirrorlog mirrored** del comando **lvcreate** al crear un dispositivo lógico en espejo. Para obtener información sobre el uso de esta opción, consulte la [Sección 4.4.3, “Cómo crear volúmenes en espejo”](#).

1.1.2. Funcionalidades nuevas y cambiadas para Red Hat Enterprise Linux 6.1

Red Hat Enterprise Linux 6.1 incluye la siguiente documentación y actualizaciones de funcionalidades y cambios.

- Red Hat Enterprise Linux 6.1 soporta la creación de volúmenes lógicos de instantáneas. Puede crear una instantánea de un volumen en espejo de la misma manera que usted crearía una instantánea de un volumen lógico en franjas o líneal. Para obtener mayor información sobre cómo crear volúmenes de instantáneas, consulte [Sección 4.4.4, “Cómo crear volúmenes de instantáneas”](#).
- Ahora al extender el volumen LVM, usted puede usar la opción **--alloc cling** del comando **lvextend** para especificar la política de asignación **cling**. Dicha política elegirá un espacio en los mismos volúmenes físicos como el último segmento del volumen lógico existente. Si no hay suficiente espacio en los volúmenes físicos y una lista de las etiquetas está definida en el archivo **lvm.conf**, LVM revisará si alguna de las etiquetas está conectada a los volúmenes físicos y buscará esas etiquetas de volumen físico entre las extensiones existentes y las nuevas. Para obtener información sobre cómo extender volúmenes en espejo LVM con la opción **--alloc cling** del comando **lvextend**, consulte [Sección 4.4.12.2, “Extender un volumen lógico con la política de asignación cling”](#).
- Puede especificar múltiples argumentos **--addtag** y **--deltag** dentro de un simple comando **pvchange**, **vgchange**, o **lvchange**. Para obtener información sobre cómo añadir o retirar etiquetas de objetos, consulte [Sección C.1, “Cómo añadir y retirar etiquetas de objetos”](#).
- Se ha ampliado la lista de caracteres en etiquetas de objetos LVM y las etiquetas pueden contener los caracteres **/**, **=**, **!**, **:"**, **#** y **&**. Para obtener información sobre etiquetas de objetos LVM, consulte [Apéndice C, Etiquetas de objetos LVM](#).
- Puede combinar RAID0 (en franja) y RAID1 (en espejo) en un solo volumen lógico. La creación de un volumen lógico cuando se especifican simultáneamente el número de espejos (**--mirrors X**) y el número de franjas (**--stripes Y**) produce un dispositivo de espejo cuyos dispositivos constituyentes están seccionados. Para obtener información sobre la creación de volúmenes lógicos en espejo, consulte [Sección 4.4.3, “Cómo crear volúmenes en espejo”](#).
- A partir de Red Hat Enterprise Linux 6.1, si necesita crear una copia de seguridad consistente de datos en un volumen lógico en cluster, puede activar el volumen exclusivamente y luego crear una instantánea. Para obtener información sobre la activación de volúmenes lógicos en un nodo, consulte [Sección 4.7, “Cómo activar volúmenes lógicos en nodos individuales en un cluster”](#).

1.2. Volúmenes lógicos

La administración de volumen crea una capa de abstracción en el almacenaje físico, lo que permite crear volúmenes lógicos de almacenaje. Proporciona mucha más flexibilidad en una cantidad de formas que el uso directo de almacenaje físico. Con un volumen lógico no hay restricción física de espacio del disco. Además, la configuración de almacenaje del hardware se oculta del software permitiendo así redimensionar y desplazar sin tener que detener la aplicación o desmontar el sistema de archivos. Esto puede reducir costos operacionales.

Los volúmenes lógicos proporcionan las siguientes ventajas sobre el uso directo de almacenamiento físico:

- ▶ Capacidad flexible

Cuando se utilizan volúmenes lógicos, los sistemas de archivos pueden extenderse a lo largo de varios discos, ya que se pueden agregar discos y particiones en un único volumen lógico.
- ▶ Grupos de almacenaje dimensionables

Puede extender los volúmenes lógicos o reducir los volúmenes lógicos con comandos de software sencillos, sin necesidad de volver a dar formato o crear particiones en los dispositivos de discos subyacentes.
- ▶ Asignación de datos en línea

Para implementar subsistemas de almacenamiento más modernos, más rápidos o resistentes, puede trasladar los datos mientras su sistema está activo. Los datos pueden ser reorganizados en discos mientras los discos están siendo utilizados. Por ejemplo, puede vaciar un disco antes de quitarlo.
- ▶ Nombres de dispositivos convenientes

Los volúmenes de almacenaje lógico pueden ser administrados en grupos definidos por el usuario, los cuales se pueden especificar de acuerdo con las necesidades.
- ▶ Entrelazado de disco

Puede crear un volumen lógico que entrelaza datos a lo largo de dos o más discos. Ésto incrementa dramáticamente la tasa de transferencia.
- ▶ Volúmenes en espejos

Los volúmenes lógicos proporcionan una manera conveniente de configurar copias para sus datos.
- ▶ Instantáneas del volumen

Al usar volúmenes lógicos, usted puede crear instantáneas del dispositivo para obtener copias de seguridad consistentes o para probar el efecto de algunos cambios sin afectar los datos reales.

La implementación de estas funcionalidades en LVM se describe en el resto de este documento.

1.3. Sinopsis de la arquitectura LVM

Para el lanzamiento del sistema operativo RHEL 4, el administrador de volúmenes lógico LVM1 fue reemplazado por LVM2, el cual tiene un marco de trabajo de kernel más genérico que LVM1. LVM2 proporciona las siguientes mejoras en comparación con LVM1:

- ▶ capacidad flexible
- ▶ almacenamiento de metadatos más eficiente
- ▶ mejor formato de recuperación
- ▶ un nuevo formato de metadatos ASCII
- ▶ cambios atómicos a los metadatos
- ▶ copias redundantes de los metadatos

LVM2 es compatible con LVM1, con la excepción de instantáneas y soporte de cluster. Puede convertir un grupo de volúmenes del formato LVM1 a LVM2 con el comando **vgconvert**. Para obtener mayor información en convertir el formato de metadatos LVM, consulte las páginas de manual **vgconvert(8)**.

La unidad de almacenaje físico subyacente de un volumen lógico LVM es un dispositivo de bloque como una partición o un disco. Este dispositivo se inicializa como un LVM *volumen físico* (PV por sus siglas en inglés)

Para crear un volumen lógico LVM, los volumen físicos se combinan en un *grupo de volúmenes* (VG). Esto crea un grupo de espacio de disco desde el cual el volumen lógico LVM puede ser asignado. Este proceso es análogo a la manera como los discos se dividen en particiones. Un volumen lógico se utiliza por el sistema de archivo y las aplicaciones (como las bases de datos).

[Figura 1.1, “Componentes de un volumen lógico LVM”](#) muestra los componentes de un volumen lógico LVM sencillo:

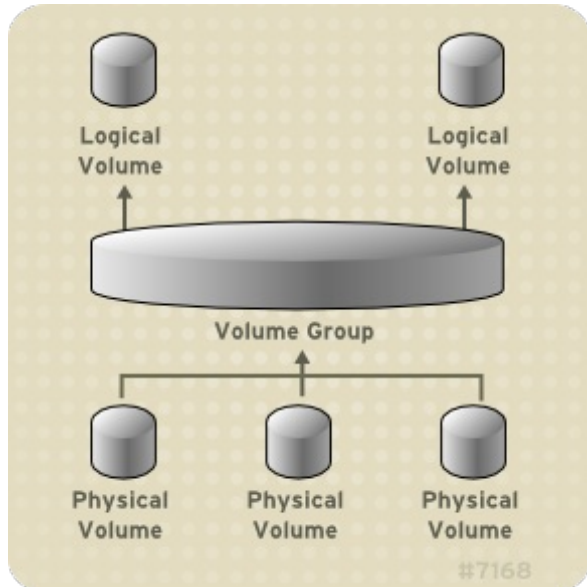


Figura 1.1. Componentes de un volumen lógico LVM

Para obtener mayor información sobre los componentes de un volumen lógico LVM, consulte el [Capítulo 2, Componentes de LVM](#).

1.4. Administrador de volúmenes lógicos en cluster (CLVM)

El administrador de volúmenes lógicos en cluster (CLVM) es un conjunto de extensiones para LVM. Estas extensiones permiten que un cluster de computadores administre el almacenamiento compartido (por ejemplo, en un SAN) mediante LVM. CLVM es parte de la adición de almacenaje resistente.

El uso de CLVM depende de los requisitos de su sistema:

- ▶ Si solo un nodo de su sistema requiere acceder al almacenamiento que está configurando como volúmenes lógicos, entonces puede utilizar LVM sin las extensiones de CLVM y los volúmenes lógicos creados con ese nodo son todos locales al nodo.
- ▶ Si está utilizando un sistema en cluster para recuperación de fallos donde únicamente un solo nodo que accede al almacenamiento está activo a cualquier momento, deberá utilizar los agentes de Alta disponibilidad de administración de volúmenes lógicos (HA-LVM). Para mayor información sobre HA-LVM, consulte *Configuración y administración de un Red Hat Cluster*.
- ▶ Si más de un nodo de su cluster requiere acceder a su almacenamiento, el cual es compartido entre los nodos activos, entonces deberá utilizar CLVM. CLVM permite configurar volúmenes lógicos en almacenaje compartido al bloquear el acceso al almacenaje físico mientras el volumen lógico está siendo configurado. CLVM utiliza servicios de bloqueo en cluster para administrar el almacenaje compartido.

Para utilizar el CLVM, debe estar ejecutándose el software de adición de Alta disponibilidad y almacenaje resistente, incluyendo el demonio `clmvd`. El demonio `clmvd` es la extensión principal de

cluster para LVM. El demonio **clvmd** se ejecuta en cada computador del cluster y distribuye las actualizaciones de metadatos de LVM en un cluster, presentando cada computador de cluster con el mismo panorama de los volúmenes lógicos. Para mayor información sobre la instalación y administración de la adición de Alta disponibilidad, consulte *Configuración y administración de un Red Hat Cluster*.

Para asegurarse de que **clvmd** se inicie en el arranque, puede ejecutar el comando **chkconfig ... on** en el servicio **clvmd**, así:

```
# chkconfig clvmd on
```

Si el demonio **clvmd** no se ha iniciado, puede ejecutar un comando **service ... start** en el servicio **clvmd**, así:

```
# service clvmd start
```

La creación de volúmenes lógicos de LVM en un entorno de cluster es idéntica a la creación de volúmenes lógicos LVM en un solo nodo. No hay diferencia en los comandos LVM, o en la interfaz gráfica LVM de usuario, como se describe en [Capítulo 4, Gestión de LVM mediante comandos de CLI](#) y [Capítulo 7, Gestión de LVM con la interfaz gráfica de LVM](#). Para habilitar los volúmenes LVM que está creando en un cluster, la infraestructura del cluster debe estar ejecutándose y el cluster debe estar en quorum.

Por defecto, los volúmenes lógicos creados con CLVM en almacenamiento compartido están a la vista de todos los computadores que tienen acceso al almacenamiento compartido. Es posible crear grupos de volúmenes lógicos en los cuales todos los dispositivos de almacenamiento están visibles para solo un nodo en el cluster. También es posible cambiar el estatus de un volumen lógico de un volumen lógico a un volumen en cluster. Para mayor información, consulte la [Sección 4.3.2, “Cómo crear grupos de volúmenes en un cluster”](#) y [Sección 4.3.7, “Cómo cambiar los parámetros de un grupo de volúmenes”](#).



Advertencia

Al crear grupos de volúmenes lógicos con CLVM en almacenaje compartido, debe verificar que todos los nodos en el cluster tengan acceso a los volúmenes físicos que constituyen el grupo de volúmenes. Las configuraciones de cluster asimétricas en las cuales algunos nodos tienen acceso al almacenamiento y otros no, no están soportadas.

[Figura 1.2, “Visión general de CLVM”](#) presenta una visión general de CLVM en un cluster.

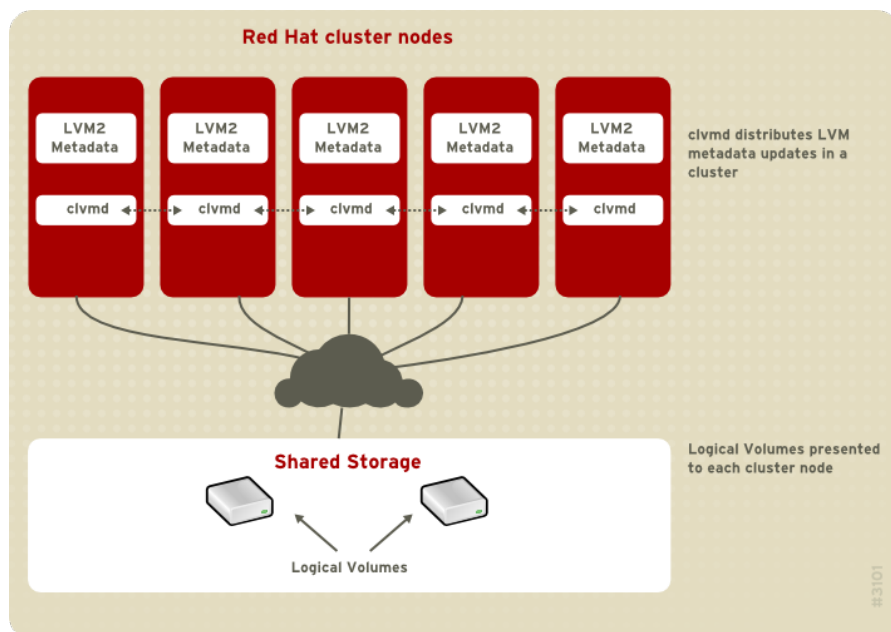


Figura 1.2. Visión general de CLVM



Nota

CLVM requiere cambios al archivo `lvm.conf` para bloqueos a lo largo del clúster. Información sobre la configuración del archivo `lvm.conf` para soportar el bloqueo en clúster se proporciona dentro del mismo archivo `lvm.conf`. Para mayor información sobre el archivo `lvm.conf`, consulte el [Apéndice B, Archivos de configuración LVM](#).

1.5. Sinopsis del documento

Lo que resta de este documento incluye los siguiente capítulos:

- ▶ [Capítulo 2, Componentes de LVM](#) describe los componentes que conforman un volumen lógico LVM.
- ▶ [Capítulo 3, Visión General de la Gestión de LVM](#) proporciona una sinopsis de los pasos básicos necesarios para configurar los volúmenes lógicos LVM, ya sea a través de la línea de comandos o con la interfaz gráfica de usuario.
- ▶ [Capítulo 4, Gestión de LVM mediante comandos de CLI](#) resume las tareas administrativas individuales que se pueden ejecutar a través de la línea de comandos para crear y mantener volúmenes lógicos.
- ▶ [Capítulo 5, Ejemplos de configuración de LVM](#) proporciona varios ejemplos de configuraciones LVM.
- ▶ [Capítulo 6, Solución de errores en LVM](#) proporciona instrucciones sobre solución de errores de conflictos que pueden presentarse con LVM.
- ▶ [Capítulo 7, Gestión de LVM con la interfaz gráfica de LVM](#) resume la operación de la interfaz gráfica de usuario LVM.
- ▶ [Apéndice A, Mapeador de Dispositivos](#) describe el Mapeador de Dispositivos utilizado por LVM para localizar volúmenes lógicos y físicos.
- ▶ [Apéndice B, Archivos de configuración LVM](#) describe los archivos de configuración de LVM.
- ▶ [Apéndice C, Etiquetas de objetos LVM](#) describe las etiquetas host y etiquetas objeto de LVM.
- ▶ [Apéndice D, Metadatos del grupo de volúmenes LVM](#) describe los metadatos del grupo de

volúmenes LVM e incluye un ejemplo de los metadatos para un grupo de volúmenes LVM.

Capítulo 2. Componentes de LVM

Este capítulo describe los componentes de un volumen lógico LVM.

2.1. Volúmenes físicos

La unidad de almacenaje físico subyacente de un volumen lógico LVM es un dispositivo de bloque como una partición o un disco completo. Para utilizar el dispositivo para un volumen lógico LVM, el dispositivo debe ser inicializado como volumen físico. Al inicializar un dispositivo de bloque como volumen físico, se asigna una etiqueta cerca del inicio del dispositivo.

Por defecto, la etiqueta LVM se ubica en el segundo sector de 512-bytes. Puede sobrescribir este valor predeterminado colocando la etiqueta en cualquiera de los primeros cuatro sectores. Esto permite que los volúmenes lógicos LVM puedan co-existir con otros usuarios de estos sectores de ser necesario.

Una etiqueta LVM proporciona una correcta identificación y ordenamiento del dispositivo para un dispositivo físico, ya que los dispositivos pueden iniciarse en cualquier orden cuando el sistema sea iniciado. Una etiqueta LVM es persistente a lo largo del cluster y entre reinicios.

La etiqueta LVM identifica el dispositivo como un volumen físico LVM. Contiene un número de identificación único (UUID) para el volumen físico. También almacena el tamaño del dispositivo de bloque en bytes y guarda información de dónde los metadatos LVM serán almacenados en el dispositivo.

Los metadatos de LVM contienen los detalles de configuración del grupo de volúmenes LVM en su sistema. Por defecto, una copia idéntica de los metadatos se mantiene en cada área de metadatos en cada volumen físico con el grupo de volúmenes. Los metadatos LVM son pequeños y se almacenan como ASCII.

En la actualidad, LVM permite almacenar 0, 1 o 2 copias idénticas de los metadatos en cada volumen físico. Por defecto se guarda una copia. No se puede cambiar el número de copias de los metadatos en el volumen físico una vez configurado. La primera copia se almacena al principio del dispositivo, poco después de la etiqueta. Si hay una segunda copia, se almacenará al final del dispositivo. Si por accidente sobrescribe la primera parte del dispositivo, la copia de los metadatos al final del dispositivo le permitirá recuperar los datos.

Para obtener mayor información sobre los metadatos de LVM y cómo cambiar sus parámetros, consulte el [Apéndice D, Metadatos del grupo de volúmenes LVM](#).

2.1.1. Distribución de un volumen físico LVM

[Figura 2.1, “Distribución de un volumen físico”](#) muestra la distribución de un volumen físico LVM. La etiqueta LVM está en el segundo sector, seguido del área de metadatos, seguido de espacio a usar en el dispositivo.



Nota

En el kernel de Linux (y a lo largo de esta documentación), los sectores tienen 512 bytes.

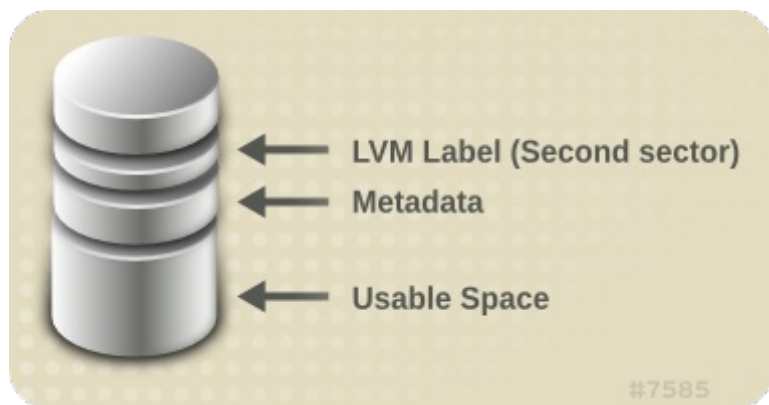


Figura 2.1. Distribución de un volumen físico

2.1.2. Múltiples particiones en un disco

LVM le permite crear volúmenes físicos de particiones de disco. Se recomienda crear una sola partición que cubra todo el disco para que sea etiquetada como volumen físico LVM. Las razones son las siguientes:

► **Conveniencia administrativa**

Es fácil controlar el hardware en un sistema si cada disco real solo aparece una vez. Esto es particularmente cierto cuando el disco falla. Además, varios volúmenes físicos en un solo disco pueden hacer que el kernel advierta que hay particiones desconocidas durante el arranque.

► **Rendimiento de entrelace**

LVM no puede saber si dos volúmenes físicos están en el mismo disco físico. Si crea un volumen lógico entrelazado cuando dos volúmenes físicos están en el mismo disco físico, los enlaces pueden estar en diferentes particiones del mismo disco. Esto llevará a un decremento en el rendimiento.

Aunque no es recomendable, hay situaciones específicas en donde se tendrá que dividir el disco en volúmenes físicos LVM separados. Por ejemplo, en un sistema con pocos discos puede llegar a ser necesario desplazar particiones cuando se migra un sistema existente a un volumen LVM. Por ejemplo, en un sistema con pocos discos puede llegar a ser necesario desplazar los datos entre particiones cuando se migra un sistema existente a volúmenes LVM. Además, si tiene un disco grande y desea tener más de un grupo de volúmenes será necesario particionar el disco. Si tiene un disco con más de una partición y ambas particiones están en el mismo grupo de volúmenes, tenga cuidado en especificar cuáles particiones deben ser incluidas en un volumen lógico cuando cree volúmenes entrelazados.

2.2. Grupos de volúmenes

Los volúmenes físicos se combinan en grupos de volúmenes. Esto crea un grupo de espacio de disco en donde los volúmenes lógicos pueden ser asignados.

En un grupo de volúmenes, el espacio de disco disponible para asignar está dividido en unidades de tamaño fijo llamadas extensiones. Una extensión es la unidad más pequeña de espacio que puede ser asignada. En los volúmenes físicos, las extensiones son conocidas como extensiones físicas.

Un volumen lógico es asignado en extensiones lógicas del mismo tamaño que las extensiones físicas. El tamaño de la extensión es así el mismo para todos los volúmenes lógicos en el grupo de volúmenes. El grupo de volúmenes relaciona las extensiones lógicas con las extensiones físicas.

2.3. Volúmenes lógicos LVM

En LVM, un grupo de volúmenes está dividido en volúmenes lógicos. Hay tres tipos de volúmenes lógicos LVM: volúmenes *lineales*, volúmenes *entrelazados* y volúmenes en *espejo*. Las siguientes secciones describen cada uno de éstos.

2.3.1. Volúmenes lineales

Un volumen lineal añade varios volúmenes físicos en un volumen lógico. Por ejemplo, si tiene dos discos de 60GB, puede crear un volumen lógico de 120GB. El almacenamiento físico es concatenado.

Cuando se crea un volumen lineal se asigna un rango de extensiones físicas a un área de un volumen lógico en orden. Por ejemplo, como se muestra en la [Figura 2.2, “Mapas de extensiones”](#) las extensiones lógicas que van de 1 a 99 pueden asignarse a un volumen físico y las extensiones de 100 a 198 pueden asignarse a un segundo volumen físico. Desde el punto de vista de la aplicación hay un solo dispositivo con 198 extensiones.

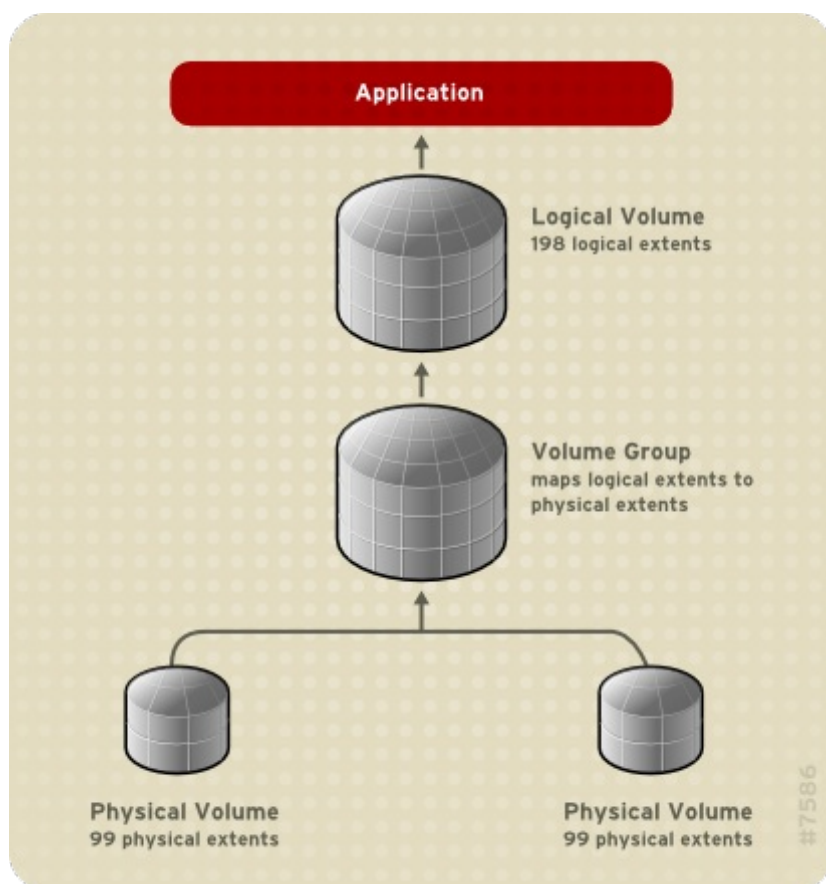


Figura 2.2. Mapas de extensiones

El volumen físico que conforma el volumen lógico puede no tener el mismo tamaño. La [Figura 2.3, “Volumen lineal con volúmenes físicos dispares”](#) muestra el grupo de volúmenes VG1 con un tamaño de extensiones físicas de 4MB. Este grupo de volúmenes incluye dos volúmenes físicos llamados **PV1** y **PV2**. Los volúmenes físicos se dividen en unidades de 4MB, ya que ese es el tamaño de las extensiones. En este ejemplo, **PV1** tiene 100 extensiones (400MB) y **PV2** tiene 200 extensiones (800MB). Puede crear un volumen lineal de cualquier tamaño entre 1 y 300 extensiones (4MB a 1200MB). En este ejemplo, el volumen lineal llamado **LV1** es 300 extensiones.

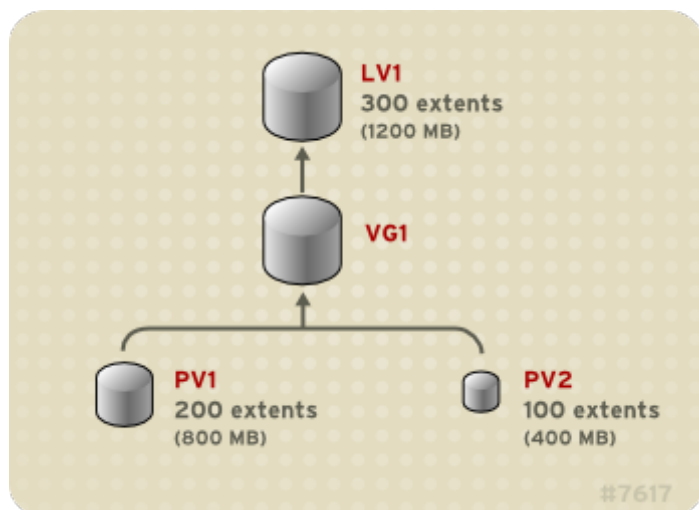


Figura 2.3. Volumen lineal con volúmenes físicos dispares

Puede configurar más de un volumen lógico lineal de cualquier tamaño del grupo de extensiones físicas. La [Figura 2.4, “Volúmenes lógicos múltiples”](#) muestra el mismo grupo de volúmenes que en la [Figura 2.3, “Volumen lineal con volúmenes físicos dispares”](#), pero en este caso se han forjado dos volúmenes lógicos del grupo de volúmenes: **LV1**, el cual tiene 250 extensiones (1000MB) y **LV2** con 50 extensiones (200MB).

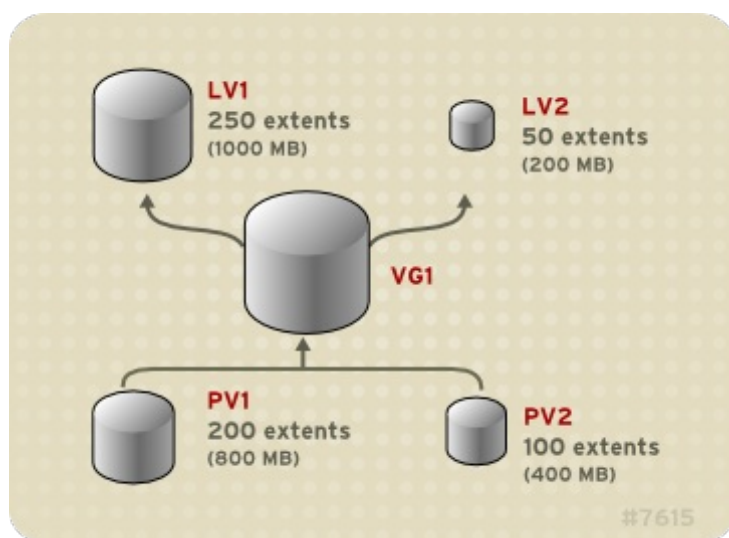


Figura 2.4. Volúmenes lógicos múltiples

2.3.2. Volúmenes lógicos entrelazados

Cuando escribe datos a un volumen lógico LVM, el sistema de archivos pone los datos a lo largo de los volúmenes lógicos subyacentes. Puede controlar el modo en que los datos se escriben en los volúmenes físicos al crear un volumen lógico entrelazado. Para lecturas y escrituras secuenciales de gran número de datos, este tipo de volúmenes mejora la eficiencia de los procesos de E/S de los datos.

El seccionamiento en franjas mejora el rendimiento al escribir los datos a un número predeterminado de volúmenes físicos de una manera circular. La E/S puede realizarse en paralelo. En algunas situaciones, se pueden obtener ganancias de rendimiento lineal por cada volumen físico añadido a la franja.

La siguiente ilustración muestra como los datos son entrelazados a lo largo de tres volúmenes físicos. En esta figura:

- ▶ el primer grupo de datos se escribe a PV1
- ▶ el segundo grupo de datos se escribe a PV2
- ▶ el tercer grupo de datos se escribe a PV3
- ▶ el cuarto grupo de datos se escribe a PV1

En un volumen lógico entrelazado, el tamaño de la banda no puede exceder el tamaño de una extensión.

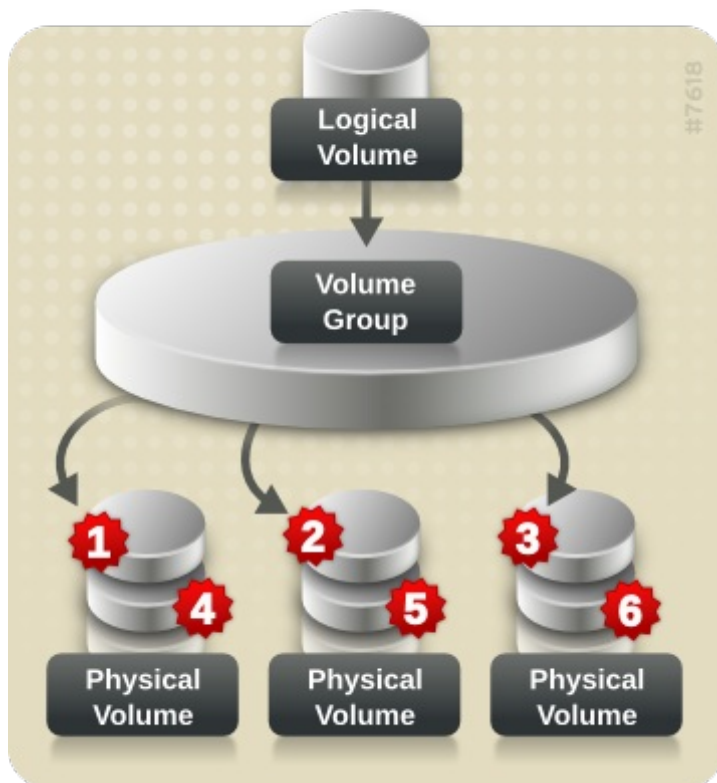


Figura 2.5. Entrelazando datos a lo largo de tres PV

Los volúmenes lógicos entrelazados pueden ser extendidos al concatenar otro grupo de dispositivos al final del primer grupo. Para extender el volumen lógico entrelazado, debe haber suficiente espacio libre en los volúmenes físicos subyacentes que conforman el grupo de volúmenes que soportan el enlace. Por ejemplo, si tiene un enlace de dos rutas que utiliza todo el grupo de volúmenes, no se podrá extender el enlace si solo se añade un volumen físico al grupo de volúmenes. Debe al menos añadir dos volúmenes físicos al grupo de volúmenes. Para mayor información sobre cómo extender un volumen entrelazado, consulte la [Sección 4.4.12.1, “Cómo extender un volumen entrelazado”](#).

2.3.3. Volúmenes lógicos en espejo

Un espejo mantiene una copia idéntica de los datos en los diferentes dispositivos. Cuando los datos se escriben en un dispositivo, éstos se escriben en un segundo dispositivo al mismo tiempo, creando una copia exacta de los datos. Esto proporciona protección para fallos de dispositivos. Cuando un pilar de un espejo falla, el volumen lógico se convierte en un volumen lineal y puede ser aun accedido.

LVM soporta espejos. Cuando se crea un volumen lógico en espejo, LVM se asegura de que los datos escritos en un volumen físico subyacente sean copiados a otro volumen físico separado. Con LVM, usted puede crear volúmenes lógicos en espejo con múltiples espejos.

Un espejo LVM divide los dispositivos a copiar en regiones que son generalmente de 512KB. LVM mantiene un pequeño registro que se utiliza para saber cuáles regiones están en sincronía con los espejos. Este registro puede estar en el disco, el cual asegura la persistencia después de reinicios del sistema. El registro también puede estar en memoria.

[Figura 2.6, “Volumen lógico en espejo.”](#) muestra un volumen lógico en espejo con un espejo. En esta configuración, el registro se guarda en disco.

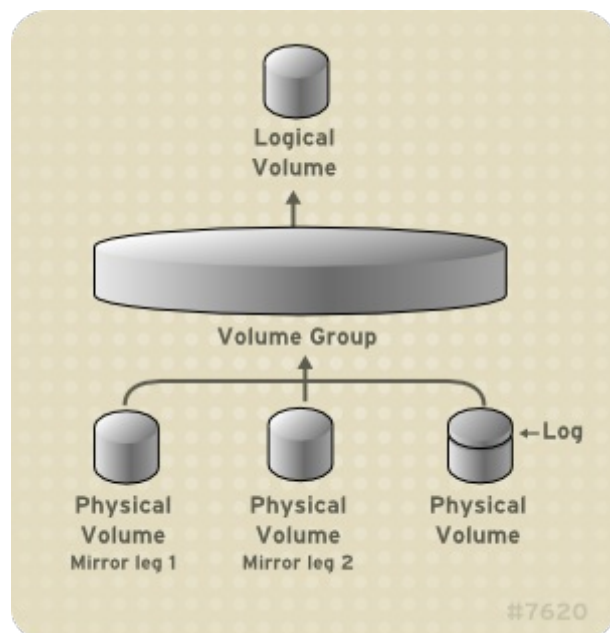


Figura 2.6. Volumen lógico en espejo

Para obtener mayor información sobre cómo crear y modificar espejos, consulte la [Sección 4.4.3, “Cómo crear volúmenes en espejo.”](#)

2.3.4. Volúmenes de instantáneas

La funcionalidad de instantáneas de LVM permite crear imágenes virtuales de un dispositivo en un momento dado sin causar la interrupción de un servicio. Cuando se realiza un cambio en el dispositivo original después de haber tomado la instantánea, ésta tendrá una copia del área de datos que se cambiaron y así permitirá la reconstrucción del estado del dispositivo.



Nota

Las instantáneas de LVM no son soportadas a lo largo de los nodos en un cluster. No se puede crear un volumen de instantánea en un grupo de volúmenes en cluster.



Nota

Las instantáneas de LVM no tienen soporte para volúmenes lógicos LVM en espejo.

Como las instantáneas copian únicamente las áreas de datos que han cambiado desde el momento en que se tomó la instantánea, ésta función requiere una cantidad mínima de almacenaje. Por ejemplo, con

un original que ha cambiado poco, 3-5% de la capacidad del original es suficiente para mantener la instantánea.



Nota

Las copias instantáneas de un sistema de archivos es una copia virtual, no una copia de seguridad del medio para el sistema de archivos. Las instantáneas no son un sustituto de los procedimientos de copias de seguridad.

El tamaño de la instantánea regula la cantidad de espacio reservado para almacenar los cambios en el volumen de origen. Por ejemplo, si usted creaba una instantánea y, a continuación, sobrescribía completamente el origen la instantánea tendría que ser al menos tan grande como el volumen original para mantener los cambios. Es necesario dimensionar una instantánea con el nivel de cambio esperado. Así, por ejemplo la instantánea de poca duración de un volumen de lectura-en su mayoría, `/usr`, necesitaría menos espacio que la instantánea de larga duración de un volumen que ve un mayor número de escrituras, tales como `/home`.

Si la instantánea se ejecuta llena, será invalidada, porque ya no puede rastrear los cambios en el volumen de origen. Por este motivo debe monitorizar el tamaño de las instantáneas con regularidad. Sin embargo, puede redimensionar las instantáneas, si tiene la capacidad de almacenaje, puede agrandarla para evitar que se caiga. Asimismo, si ve que el volumen de la instantánea es muy grande para lo que la necesita, puede reducir el tamaño del volumen para liberar espacio requerido por otros volúmenes lógicos.

Cuando crea un sistema de archivos de instantánea, el acceso de lectura y escritura al original sigue siendo posible. Si un pedazo de la instantánea cambia, ese pedazo es marcado y nunca recibe una copia del volumen original.

Hay varios usos para las instantáneas:

- ▶ Generalmente, una instantánea se toma cuando se necesita ejecutar una copia de seguridad en un volumen lógico sin detener el sistema en vivo que continuamente actualiza los datos.
- ▶ Puede ejecutar el comando `fsck` en un sistema de archivos de instantánea para revisar la integridad del sistema de archivos y determinar si el sistema de archivos original requiere reparación.
- ▶ Porque la instantánea es de lectura y escritura, puede probar aplicaciones contra los datos en producción si toma una instantánea y ejecuta las pruebas en ella, sin tocar así los datos reales.
- ▶ Puede crear volúmenes LVM para usar con virtualización de Red Hat. Las instantáneas de LVM pueden ser utilizadas para crear instantáneas de snapshots can be used to create snapshots of virtual guest images. These snapshots can provide a convenient way to modify existing guests or create new guests with minimal additional storage. For more information on creating LVM snapshots of virtualized guests, see the *Red Hat Enterprise Linux Virtualization Guide*.

Para obtener mayor información sobre cómo crear volúmenes de instantáneas, consulte la [Sección 4.4.4, “Cómo crear volúmenes de instantáneas”](#).

A partir del lanzamiento de Red Hat Enterprise Linux 6, se debe usar la opción `--merge` del comando `lvconvert` para fusionar una instantánea en su volumen original. Una forma de usar esta función es realizar la reversión del sistema si se han perdido datos o archivos o de lo contrario, se necesitará restaurar el sistema a un estado anterior. Después de fusionar el volumen de la instantánea, el volumen lógico resultante tendrá el nombre del volumen original, número menor y UUID, y la instantánea fusionada será eliminada. Para obtener información sobre el uso de esta opción, consulte la

[Sección 4.4.5, "Cómo fusionar volúmenes de instantáneas".](#)

Capítulo 3. Visión General de la Gestión de LVM

Este capítulo proporciona un resumen de los procedimientos administrativos utilizados para configurar volúmenes lógicos LVM. Este capítulo proporciona una visión general de los pasos a seguir. Para obtener ejemplos detallados de los procedimientos de configuración de LVM, consulte el [Capítulo 5, Ejemplos de configuración de LVM](#).

Para obtener una descripción de los comandos CLI a utilizar para administrar LVM, consulte el [Capítulo 4, Gestión de LVM mediante comandos de CLI](#). Alternativamente, puede utilizar la interfaz gráfica descrita en el [Capítulo 7, Gestión de LVM con la interfaz gráfica de LVM](#).

3.1. Creación de volúmenes LVM en un cluster

Para crear volúmenes lógicos en un entorno de cluster, utilice el administrador de volumen lógico en cluster (CLVM), el cual es una serie de extensiones de agrupamiento para LVM. Dichas extensiones permiten a un cluster de computadores administrar el almacenamiento compartido (por ejemplo, en un SAN) mediante LVM. A fin de utilizar CLVM, la adición de alta disponibilidad y software de adición de almacenaje resistente, incluyendo el demonio `clmvd`, debe ser iniciado en el momento de arranque, tal como se describe en la [Sección 1.4, “Administrador de volúmenes lógicos en cluster \(CLVM\)”](#).

La creación de volúmenes lógicos en un entorno de cluster es idéntica a la creación de volúmenes lógicos LVM en un solo nodo. No hay ninguna diferencia en los comandos LVM mismos o en la interfaz gráfica de LVM. Para activar los volúmenes LVM creados en un cluster, la infraestructura del cluster debe estar en ejecución y el cluster debe tener quorum.

CLVM requiere cambios al archivo `lvm.conf` para bloqueos a lo largo del cluster. Información sobre la configuración del archivo `lvm.conf` para soportar el bloqueo en cluster se proporciona dentro del mismo archivo `lvm.conf`. Para mayor información sobre el archivo `lvm.conf`, consulte el [Apéndice B, Archivos de configuración LVM](#).

Por defecto, los volúmenes lógicos creados con CLVM en almacenamiento compartido son visibles a todos los computadores que tienen acceso al almacenamiento compartido. Es posible, no obstante, crear volúmenes lógicos cuando los dispositivos de almacenamiento están visibles a un solo nodo en el cluster. También se puede cambiar el estatus de un grupo de volúmenes de un volumen local a un grupo de volúmenes en cluster. Para obtener información, consulte la [Sección 4.3.2, “Cómo crear grupos de volúmenes en un cluster”](#) y [Sección 4.3.7, “Cómo cambiar los parámetros de un grupo de volúmenes”](#).



Advertencia

Al crear grupos de volúmenes lógicos con CLVM en almacenaje compartido, debe verificar que todos los nodos en el cluster tengan acceso a los volúmenes físicos que constituyen el grupo de volúmenes. Las configuraciones de cluster asimétricas en las cuales algunos nodos tienen acceso al almacenamiento y otros no, no están soportadas.

Para obtener mayor información sobre cómo instalar adiciones de alta disponibilidad y configurar la infraestructura de cluster, consulte *Cluster Administration*.

Para obtener un ejemplo para crear un volumen lógico en espejo en un cluster, consulte la [Sección 5.5, “Cómo crear un volumen lógico LVM en un cluster”](#).

3.2. Sinopsis de la creación de volúmenes lógicos

El siguiente resumen muestra los pasos necesarios para crear un volumen lógico LVM.

1. Inicialice las particiones a usar para el volumen LVM como volúmenes físicos (creando etiquetas en ellos).
2. Cree un grupo de volúmenes.
3. Cree un volumen lógico.

Después de crear el volumen lógico puede crear y montar el sistema de archivos. Los ejemplos en este documento utilizan el sistema de archivos GFS2.



Nota

Aunque un sistema de archivo GFS2 se puede ejecutar en un sistema autónomo o como parte de una configuración de cluster, para el lanzamiento de Red Hat Enterprise 6, Red Hat no soporta el uso de GFS2 como un sistema de archivos de un solo nodo. Red Hat continuará soportando los sistemas de archivos GFS2 de un solo nodo para montar capturas de pantalla de sistemas de archivos de cluster (por ejemplo, para efectos de copias de seguridad).

1. Cree un sistema de archivos GFS2 en el volumen lógico con el comando **mkfs.gfs2**.
2. Cree un nuevo punto de montaje con el comando **mkdir** . En un entorno de cluster, cree el punto de montaje en todos los nodos del cluster.
3. Monte el sistema de archivos. Añada una línea en **fstab** para cada nodo en el sistema.

Además, puede crear y montar el sistema de archivos GFS2 con la interfaz gráfica LVM.

La creación de volúmenes LVM no depende de la máquina ya que el área de almacenamiento para la información de la configuración de LVM está en el volumen físico y no en la máquina donde el volumen fue creado. Los servidores que utilizan el almacenamiento tienen copias locales, pero puede recrear ésta desde la información en el volumen físico. Puede adherir los volúmenes físicos a un servidor diferente si las versiones de LVM son compatibles.

3.3. Cómo aumentar el tamaño de un sistema de archivos en un volumen lógico

Para aumentar el tamaño de un sistema de archivos en un volumen lógico, ejecute los siguientes pasos:

1. Cree un nuevo volumen físico.
2. Extienda el grupo de volúmenes que contiene el volumen lógico con el sistema de archivos que está aumentando para incluir un nuevo volumen físico.
3. Extienda el volumen lógico para incluir el nuevo volumen físico.
4. Aumente el sistema de archivos.

Si tiene suficiente espacio sin asignar en el grupo de volúmenes, puede utilizar ese espacio para extender el volumen lógico en vez de ejecutar los pasos 1 y 2.

3.4. Copia de seguridad del volumen lógico

Los archivos y copias de seguridad de los metadatos son creados automáticamente en cada cambio de la configuración de volúmenes lógicos y grupos de volúmenes a menos que se desactive en el archivo **lvm.conf**. Por defecto, las copias de seguridad de los metadatos se almacenan en el archivo

`/etc/lvm/backup` y los archivos de metadatos se almacenan en `/etc/lvm/archive`. El tiempo de almacenamiento de los metadatos en `/etc/lvm/archive` se determina en parámetros configurables en el archivo `lvm.conf`. Una copia de seguridad del sistema debe incluir el contenido del directorio `/etc/lvm` en la copia de seguridad.

Tenga en cuenta que una copia de seguridad de los metadatos no copia los datos del usuario y del sistema contenidos en los volúmenes lógicos.

Puede manualmente crear una copia de seguridad de los metadatos al archivo `/etc/lvm/backup` con el comando `vgcfgbackup`. Puede restaurar los metadatos con el comando `vgcfgrestore`. Los comandos `vgcfgbackup` y `vgcfgrestore` se describen en la [Sección 4.3.12, “Cómo crear copias de seguridad de los metadatos del grupo de volúmenes”](#).

3.5. Registro

Todos los mensajes pasan a través de un módulo de registro con opciones independientes de niveles de registro para:

- ▶ salida estándar/error
- ▶ syslog
- ▶ archivo de registro
- ▶ función de registro externo

Los niveles de registro se establecen en el archivo `/etc/lvm/lvm.conf`, el cual está descrito en el [Apéndice B, Archivos de configuración LVM](#).

Capítulo 4. Gestión de LVM mediante comandos de CLI

Este capítulo resume las tareas administrativas individuales que se deben ejecutar a través de comandos de la interfaz de la línea de comandos (CLI) de LVM para crear y administrar volúmenes lógicos.



Nota

Si está creando o modificando un volumen LVM para un entorno en cluster, asegúrese de que está ejecutando el demonio **clvmd**. Para mayor información, consulte la [Sección 3.1, “Creación de volúmenes LVM en un cluster”](#).

4.1. Uso de los comandos

Hay varias funciones generales en todos los comandos LVM.

Cuando se requiere especificar el tamaño como parte del argumento de un comando, las unidades pueden ser especificadas explícitamente. Si no especifica una unidad, se asumirá la unidad predefinida -generalmente KB o MB. Los comandos LVM para la línea de comandos no aceptan fracciones.

Tenga en cuenta que las unidades en las opciones de la línea de comando en LVM no reconocen entre mayúsculas y minúsculas (m y M son iguales) y que se debe utilizar potencias de 2 (múltiplos de 1024). Sin embargo, cuando especifique el argumento **--units** en un comando, las minúsculas indican que las unidades son múltiplos de 1024 mientras que las mayúsculas indican que son múltiplos de 1000.

Cuando los comandos reciben el nombre de un grupo de volúmenes o de nombres de un volumen lógico como argumentos, la ruta completa es opcional. Un volumen lógico llamado **lv010** en un grupo de volúmenes llamado **vg0** puede ser especificado como **vg0/lv010**. Cuando una lista de grupos de volúmenes es requerida pero se deja vacía, ésta se reemplaza por una lista de todos los grupos de volúmenes. Cuando se requiere una lista de volúmenes lógicos pero un grupo de volúmenes es dado, la lista de todos los volúmenes lógicos en ese grupo de volúmenes se sustituirá. Por ejemplo, el comando **lvdisplay vg0** mostrará todos los volúmenes lógicos en grupo de volúmenes **vg0**.

Todos los comandos de LVM aceptan el argumento **-v**, el cual puede ser introducido varias veces para aumentar la cantidad de mensajes de salida. El siguiente ejemplo muestra el mensaje de salida por defecto del comando **lvcreate**.

```
# lvcreate -L 50MB new_vg
Rounding up size to full physical extent 52.00 MB
Logical volume "lv010" created
```

El siguiente ejemplo muestra la salida del comando **lvcreate** con el argumento **-v**.


```
# lvcreate -v -L 50MB new_vg
  Finding volume group "new_vg"
Rounding up size to full physical extent 52.00 MB
  Archiving volume group "new_vg" metadata (seqno 4).
  Creating logical volume lvol0
  Creating volume group backup "/etc/lvm/backup/new_vg" (seqno 5).
  Found volume group "new_vg"
  Creating new_vg-lvol0
  Loading new_vg-lvol0 table
  Resuming new_vg-lvol0 (253:2)
  Clearing start of logical volume "lvol0"
  Creating volume group backup "/etc/lvm/backup/new_vg" (seqno 5).
Logical volume "lvol0" created
```

Se puede utilizar **-vv**, **-vvv** o **-vvvv** para incrementar los mensajes de salida y obtener más detalles sobre la ejecución del comando. El argumento **-vvvv** proporciona la mayor cantidad de información. El siguiente ejemplo muestra las primeras líneas del mensaje de salida de **lvcreate** con el el argumento **-vvvv**.

```
# lvcreate -vvvv -L 50MB new_vg
#lvmcmdline.c:913      Processing: lvcreate -vvvv -L 50MB new_vg
#lvmcmdline.c:916      O_DIRECT will be used
#config/config.c:864   Setting global/locking_type to 1
#locking/locking.c:138 File-based locking selected.
#config/config.c:841   Setting global/locking_dir to /var/lock/lvm
#activate/activate.c:358 Getting target version for linear
#ioctl/libdm-iface.c:1569      dm version OF [16384]
#ioctl/libdm-iface.c:1569      dm versions OF [16384]
#activate/activate.c:358      Getting target version for striped
#ioctl/libdm-iface.c:1569      dm versions OF [16384]
#config/config.c:864   Setting activation/mirror_region_size to 512
...
```

Puede ver el mensaje de ayuda de cualquier comando LVM con el argumento **--help**.

```
commandname --help
```

Para ver la página de manual para un comando, ejecute el comando **man**:

```
man commandname
```

El comando **man lv** proporciona información general en línea sobre LVM.

Todos los objetos LVM se relacionan internamente con un UUID, el cual es asignado durante la creación del objeto. Esto puede ser útil cuando se quita un volumen físico llamado **/dev/sdf**, el cual hace parte de un grupo de volúmenes, y cuando se conecta de nuevo, aparece con el nombre **/dev/sdk**. LVM encontrará el volumen físico sin problema porque identifica el volumen físico por el UUID y no por el nombre del dispositivo. Para obtener mayor información sobre cómo especificar el UUID de un volumen físico durante su creación, consulte la [Sección 6.4, "Cómo recuperar los metadatos de un volumen físico"](#).

4.2. Administración de volúmenes físicos

Esta sección describe los comandos que ejecutan varios aspectos de la administración de volúmenes físicos.

4.2.1. Creación de volúmenes físicos

Las siguientes subdivisiones describen los comandos usados para crear volúmenes físicos.

4.2.1.1. Cómo establecer el tipo de partición

Si está usando un dispositivo de disco completo para el volumen físico, el disco no deberá tener una tabla de particiones. Para particiones de disco DOS, el identificador de la partición debe establecerse a 0x8e con el comando **fdisk** o **cdisk**. Para los dispositivos de disco completos, la tabla de particiones debe ser borrada. De esta forma se destruirán todos los datos en el disco. Puede quitar la tabla de particiones con el siguiente comando:

```
dd if=/dev/zero of=PhysicalVolume bs=512 count=1
```

4.2.1.2. Cómo inicializar los volúmenes físicos

Utilice el comando **pvcreate** para inicializar un dispositivo de bloque que será usado como volumen físico. La inicialización es análoga a la creación de formato de un sistema de archivos.

El siguiente comando inicializa **/dev/sdd1**, **/dev/sde1**, y **/dev/sdf1** para ser usados como volúmenes físicos LVM.

```
pvcreate /dev/sdd1 /dev/sde1 /dev/sdf1
```

Para inicializar particiones en vez de discos completos ejecute **pvcreate** en la partición. El siguiente ejemplo inicializa **/dev/hdb1** como un volumen físico LVM para ser usado como parte de un volumen lógico LVM.

```
pvcreate /dev/hdb1
```

4.2.1.3. Cómo buscar Dispositivos de Bloque

Puede buscar dispositivos de bloque que pueden ser usados como parte de volúmenes físicos con el comando **lvmdiskscan**, como se muestra en el siguiente ejemplo.

```

# lvmddiskscan
/dev/ram0          [      16.00 MB]
/dev/sda          [      17.15 GB]
/dev/root         [      13.69 GB]
/dev/ram          [      16.00 MB]
/dev/sda1         [      17.14 GB] LVM physical volume
/dev/VolGroup00/LogVol01 [    512.00 MB]
/dev/ram2         [      16.00 MB]
/dev/new_vg/lvol0 [      52.00 MB]
/dev/ram3         [      16.00 MB]
/dev/pk1_new_vg/sparkie_lv [    7.14 GB]
/dev/ram4         [      16.00 MB]
/dev/ram5         [      16.00 MB]
/dev/ram6         [      16.00 MB]
/dev/ram7         [      16.00 MB]
/dev/ram8         [      16.00 MB]
/dev/ram9         [      16.00 MB]
/dev/ram10        [      16.00 MB]
/dev/ram11        [      16.00 MB]
/dev/ram12        [      16.00 MB]
/dev/ram13        [      16.00 MB]
/dev/ram14        [      16.00 MB]
/dev/ram15        [      16.00 MB]
/dev/sdb          [      17.15 GB]
/dev/sdb1         [      17.14 GB] LVM physical volume
/dev/sdc          [      17.15 GB]
/dev/sdc1         [      17.14 GB] LVM physical volume
/dev/sdd          [      17.15 GB]
/dev/sdd1         [      17.14 GB] LVM physical volume
7 disks
17 partitions
0 LVM physical volume whole disks
4 LVM physical volumes

```

4.2.2. Cómo ver los volúmenes físicos

Hay tres comandos que pueden ser usados para ver las propiedades de los volúmenes físicos LVM: **pvs**, **pvdiskdisplay** y **pvscan**.

El comando **pvs** proporciona información del volumen físico de una manera configurable, mostrando una línea por volumen físico. El comando **pvs** proporciona gran control en el formato y es útil en la creación de scripts. Para obtener información sobre cómo utilizar el comando **pvs** para personalizar su salida, consulte la [Sección 4.8, "Reporte personalizado para LVM"](#).

El comando **pvdiskdisplay** proporciona un mensaje de salida de varias líneas para cada volumen físico. Muestra las propiedades físicas (tamaño, extensiones, grupo de volúmenes, etc.) en un formato fijo.

El siguiente ejemplo muestra la salida de **pvdiskdisplay** para un solo volumen físico.

```
# pvdisplay
--- Physical volume ---
PV Name           /dev/sdc1
VG Name           new_vg
PV Size           17.14 GB / not usable 3.40 MB
Allocatable       yes
PE Size (KByte)   4096
Total PE          4388
Free PE           4375
Allocated PE      13
PV UUID           Joqlch-yWSj-kuEn-IdwM-01S9-X08M-mcpsVe
```

El comando **pvscan** explora todos los dispositivos de bloque LVM soportados en el sistema por volúmenes físicos.

El siguiente comando muestra todos los dispositivos físicos encontrados:

```
# pvscan
PV /dev/sdb2   VG vg0   lvm2 [964.00 MB / 0   free]
PV /dev/sdc1   VG vg0   lvm2 [964.00 MB / 428.00 MB free]
PV /dev/sdc2           lvm2 [964.84 MB]
Total: 3 [2.83 GB] / in use: 2 [1.88 GB] / in no VG: 1 [964.84 MB]
```

Puede definir un filtro en **lvm.conf** para que este comando no explore un volumen físico específico. Para obtener información en cómo utilizar filtros para controlar cuáles dispositivos deben ser explorados, consulte la [Sección 4.5, “Cómo controlar la exploración de dispositivos LVM a través de filtros”](#).

4.2.3. Cómo prevenir la asignación en un volumen físico

Puede evitar la asignación de extensiones físicas en el espacio libre de uno o más volúmenes físicos con el comando **pvchange**. Esto puede ser necesario si hay errores de disco o si remueve el volumen físico.

El siguiente comando desactiva la asignación de extensiones físicas en **/dev/sdk1**.

```
pvchange -x n /dev/sdk1
```

También se puede utilizar el argumento **-xy** del comando **pvchange** para activar la asignación cuando esta ha sido previamente desactivada.

4.2.4. Cómo redimensionar un volumen físico

Si necesita redimensionar un dispositivo de bloque por cualquier razón, utilice el comando **pvresize** para actualizar LVM con el nuevo tamaño. Puede ejecutar este comando mientras LVM está usando el volumen físico.

4.2.5. Cómo retirar volúmenes físicos

Si LVM ya no necesita un dispositivo, puede retirar la etiqueta LVM con el comando **pvremove**. La ejecución del comando **pvremove** convierte en ceros los metadatos del LVM en un volumen físico vacío.

Si el volumen físico que desea retirar es parte de un grupo de volúmenes, primero se debe retirar el volumen físico del grupo con el comando **vgreduce**, tal y como se describe en la [Sección 4.3.6, “Cómo retirar un volumen físico de un grupo de volúmenes”](#).

```
# pvremove /dev/ram15
Labels on physical volume "/dev/ram15" successfully wiped
```

4.3. Administración del grupo de volúmenes

Esta sección describe los comandos que ejecutan los diferentes aspectos de la administración de grupos de volúmenes.

4.3.1. Cómo crear grupos de volúmenes

Para crear un grupo de volúmenes desde uno o más volúmenes físicos, utilice el comando **vgcreate**. El comando **vgcreate** crea un nuevo grupo de volúmenes por nombre y añade al menos un volumen físico.

El siguiente comando crea un grupo de volúmenes llamado **vg1** que contiene los volúmenes físicos **/dev/sdd1** y **/dev/sde1**.

```
vgcreate vg1 /dev/sdd1 /dev/sde1
```

Cuando se utilizan volúmenes físicos para crear un grupo de volúmenes, su espacio de disco se divide en extensiones de 4MB de forma predeterminada. El valor de la extensión es la cantidad mínima por la cual el volumen lógico puede ser incrementado o reducido. Un número grande de extensiones no tendrá impacto en el rendimiento de E/S del volumen lógico.

Si el valor predeterminado no es el deseado, el tamaño de la extensión puede ser especificado con la opción **-s** del comando **vgcreate**. Puede crear límites en el número de volúmenes físicos o lógicos que un grupo puede tener mediante los argumentos **-p** y **-l** del comando **vgcreate**.

Por defecto, un grupo de volúmenes asigna las extensiones físicas de acuerdo con reglas de sentido común, tales como no ubicar enlaces paralelos en el mismo volumen físico. Esta es la política de asignación **normal**. Puede utilizar el argumento **--alloc** del comando **vgcreate** para especificar una de las siguientes política de asignación: **contiguous**, **anywhere** o **cling**.

La política **contiguous** requiere que las nuevas extensiones sean adyacentes a las extensiones existentes. Si hay suficientes extensiones libres para satisfacer una solicitud de asignación, una política de asignación **normal** no las usará, pero la política de asignación **anywhere** sí lo hará, incluso si el hecho de asignar dos enlaces en el mismo volumen físico reduce el rendimiento. La política **cling** ubica nuevas extensiones en el mismo volumen físico que las extensiones en el mismo enlace del volumen lógico. Estas políticas pueden ser modificadas con el comando **vgchange**.

Para obtener información sobre el uso de la política **cling** junto con las etiquetas LVM para especificar los volúmenes físicos adicionales a usar para extender un volumen LVM, consulte [Sección 4.4.12.2, "Extender un volumen lógico con la política de asignación cling"](#).

En general, las políticas de asignaciones diferentes a **normal** solo se requieren en casos especiales donde se deben especificar asignaciones de extensiones inusuales o no estándar.

Los grupos de volúmenes LVM y los volúmenes lógicos subyacentes están incluidos en el directorio de archivos especiales de dispositivos **/dev** con la siguiente estructura:

```
/dev/vg/lv/
```

Por ejemplo, si crea dos grupos de volúmenes **mivg1** y **mivg2**, cada uno con tres volúmenes lógicos

llamados **lvo1**, **lvo2** y **lvo3**, esto crea seis archivos especiales de dispositivos:

```
/dev/myvg1/lv01
/dev/myvg1/lv02
/dev/myvg1/lv03
/dev/myvg2/lv01
/dev/myvg2/lv02
/dev/myvg2/lv03
```

El tamaño máximo del dispositivo con LVM es 8 EB en CPU de 64 bits

4.3.2. Cómo crear grupos de volúmenes en un cluster

Puede crear grupos de volúmenes en un entorno de cluster con el comando **vgcreate**, de la misma forma que los crea en un único nodo.

Por defecto, los grupos de volumen creados con CLVM en un almacenaje compartido son visibles para todos los computadores que tienen acceso al almacenamiento compartido. Sin embargo, es posible crear grupos de volumen que son locales, visibles únicamente para un nodo en el cluster, mediante **-c n** del comando **vgcreate**.

El siguiente comando, cuando se ejecuta en un entorno de cluster, crea un grupo de volúmenes local para el nodo desde el cual el comando fue ejecutado. El comando crea un volumen lógico llamado **vg1** que contiene los volúmenes físicos **/dev/sdd1** y **/dev/sde1**.

```
vgcreate -c n vg1 /dev/sdd1 /dev/sde1
```

Puede cambiar si un grupo de volúmenes existente es local o en cluster con la opción **-c** del comando **vgchange**, el cual está descrito en la [Sección 4.3.7, “Cómo cambiar los parámetros de un grupo de volúmenes”](#).

Puede comprobar si un grupo de volúmenes existente es un grupo de volúmenes en cluster con el comando **vgs**, el cual visualiza el atributo **c** si el volumen está en cluster. El siguiente comando muestra los atributos de los grupos de volúmenes **VolGroup00** y **testvg1**. En este ejemplo, **VolGroup00** no está en cluster, mientras que **testvg1** sí lo está, tal como lo indica el atributo **c** bajo el encabezado **Attr**.

```
[root@doc-07]# vgs
VG                #PV #LV #SN Attr   VSize  VFree
VolGroup00        1  2  0 wz--n- 19.88G  0
testvg1           1  1  0 wz--nc 46.00G  8.00M
```

Para obtener mayor información sobre el comando **vgs**, consulte la [Sección 4.3.4, “Cómo mostrar los grupos de volúmenes”](#), [Sección 4.8, “Reporte personalizado para LVM”](#) y la página de manual **vgs**.

4.3.3. Cómo agregar volúmenes físicos al grupo de volúmenes

Para agregar volúmenes físicos a un grupo de volúmenes existente, utilice el comando **vgextend**. El comando **vgextend** incrementa la capacidad de un grupo de volúmenes al añadir uno o más volúmenes físicos libres.

El siguiente comando añade el volumen físico **/dev/sdf1** al grupo de volúmenes **vg1**.

```
vgextend vg1 /dev/sdf1
```

4.3.4. Cómo mostrar los grupos de volúmenes

Hay dos comandos que puede utilizar para mostrar las propiedades de los grupos de volúmenes LVM: **vgs** y **vgdisplay**.

El comando **vgscan**, el cual examina todos los discos para grupos de volúmenes y reconstruye el archivo cache, también muestra los grupos de volúmenes. Para obtener mayor información sobre el comando **vgscan**, consulte la [Sección 4.3.5, “Cómo explorar discos en búsqueda de grupos de volúmenes para construir el archivo de caché”](#).

El comando **svg** proporciona información sobre el grupo de volúmenes de una manera configurable, mostrando una línea por grupo de volúmenes. El comando **vgs** proporciona gran control y es útil en scripts. Para obtener mayor información sobre cómo utilizar el comando **vgs** para personalizar el mensaje de salida consulte la [Sección 4.8, “Reporte personalizado para LVM”](#).

El comando **vgdisplay** muestra las propiedades del grupo de volúmenes (como tamaño, extensiones, número de volúmenes físicos, etc) en una forma predeterminada. El siguiente ejemplo muestra la salida de un comando **vgdisplay** para el grupo de volúmenes **new_vg**. Si no especifica un grupo de volúmenes, todos los grupos de volúmenes existentes se visualizarán.

```
# vgdisplay new_vg
--- Volume group ---
VG Name                new_vg
System ID
Format                 lvm2
Metadata Areas         3
Metadata Sequence No  11
VG Access              read/write
VG Status              resizable
MAX LV                 0
Cur LV                1
Open LV               0
Max PV                 0
Cur PV                3
Act PV                3
VG Size                51.42 GB
PE Size                4.00 MB
Total PE              13164
Alloc PE / Size        13 / 52.00 MB
Free PE / Size         13151 / 51.37 GB
VG UUID                jxQJ0a-ZKk0-0pM0-0118-nlw0-wwqd-fD5D32
```

4.3.5. Cómo explorar discos en búsqueda de grupos de volúmenes para construir el archivo de caché

El comando **vgscan** explora todos los dispositivos de disco soportados en el sistema en busca de volúmenes físicos LVM y grupos de volúmenes. Este comando reconstruye el archivo de cache LVM en **/etc/lvm/.cache**, el cual mantiene una lista de dispositivos LVM.

LVM ejecuta automáticamente el comando **vgscan** al inicio del sistema y en otros momentos durante la operación de LVM, como cuando usted ejecuta un comando **vgcreate** o cuando LVM detecta alguna inconsistencia.


Nota

Al cambiar la configuración del hardware, podría necesitar ejecutar el comando **vgscan** de forma manual y añadir o eliminar un dispositivo de un nodo, para que los nuevos dispositivos que no estaban presentes en el tiempo de inicio sean detectados por el sistema. Esto será necesario, por ejemplo, cuando añada nuevos discos al sistema en un SAN o realice una conexión en caliente de un nuevo disco que ha sido etiquetado como volumen físico.

Puede definir un filtro en el archivo **lvm.conf** para restringir la exploración de dispositivos específicos. Para obtener mayor información sobre cómo utilizar filtros para controlar cuáles dispositivos son explorados, consulte la [Sección 4.5, “Cómo controlar la exploración de dispositivos LVM a través de filtros”](#).

Los siguientes ejemplos muestran la salida del comando **vgscan**.

```
# vgscan
Reading all physical volumes. This may take a while...
Found volume group "new_vg" using metadata type lvm2
Found volume group "officevg" using metadata type lvm2
```

4.3.6. Cómo retirar un volumen físico de un grupo de volúmenes

Para quitar volúmenes físicos que no están en uso de un grupo de volúmenes, utilice el comando **vgreduce**. El comando **vgreduce** reduce la capacidad de un grupo de volúmenes al retirar uno o más volúmenes físicos vacíos. Esto libera los volúmenes físicos para que puedan ser utilizados en un grupo de volúmenes diferente o para que sean removidos del sistema.

Antes de quitar un volumen físico de un grupo de volúmenes, puede asegurarse de que el volumen físico no es utilizado por ningún volumen lógico mediante el comando **pvdisplay**.

```
# pvdisplay /dev/hda1

-- Physical volume ---
PV Name           /dev/hda1
VG Name           myvg
PV Size           1.95 GB / NOT usable 4 MB [LVM: 122 KB]
PV#               1
PV Status         available
Allocatable       yes (but full)
Cur LV           1
PE Size (KByte)   4096
Total PE          499
Free PE           0
Allocated PE      499
PV UUID           Sd44tK-9IRw-SrMC-M0kn-76iP-iftz-OVSen7
```

Si el volumen físico está siendo utilizado, debe desplazar los datos a otro volumen físico con el comando **pvmove**. A continuación, remueva el volumen físico con el comando **vgreduce**:

El siguiente comando remueve el volumen físico **/dev/hda1** del grupo de volúmenes **my_volume_group**.

```
# vgreduce my_volume_group /dev/hda1
```


4.3.7. Cómo cambiar los parámetros de un grupo de volúmenes

El comando **vgchange** se utiliza para desactivar y activar grupos de volúmenes, como se describe en [Sección 4.3.8, “Cómo activar y desactivar grupos de volúmenes”](#). También puede usar este comando para cambiar varios parámetros de grupo de volúmenes para un grupo de volúmenes existente.

El siguiente comando cambia el número máximo de volúmenes lógicos del grupo de volúmenes **vg00** a 128.

```
vgchange -l 128 /dev/vg00
```

Para obtener una descripción de los parámetros del grupo de volúmenes que pueden ser modificados con el comando **vgchange**, vea las páginas man de **vgchange(8)**.

4.3.8. Cómo activar y desactivar grupos de volúmenes

Cuando el grupo de volúmenes es creado, se activa en forma predeterminada. Esto significa que el volumen lógico en ese grupo es accesible y está sujeto a cambio.

Hay diferentes motivos por los cuales usted tendría que desactivar un grupo de volúmenes para que sea desconocido por el kernel. Para activar o desactivar un grupo de volúmenes utilice el argumento **-a** (**--available**) del comando **vgchange**.

El siguiente ejemplo desactiva el grupo de volúmenes **my_volume_group**.

```
vgchange -a n my_volume_group
```

Si el bloqueo de cluster está activo, añada 'e' para activar o desactivar un grupo de volúmenes exclusivamente en un nodo o 'l' para activar o desactivar un grupo de volúmenes en el nodo local únicamente. Los volúmenes lógicos con instantáneas de un único host son siempre desactivados exclusivamente porque éstos solo puede ser utilizados en un nodo durante un momento dado.

Puede desactivar volúmenes lógicos individuales con el comando **lvchange**, como se describe en la [Sección 4.4.8, “Cómo cambiar los parámetros de un grupo de volúmenes lógico”](#). Para obtener mayor información sobre cómo activar volúmenes lógicos en nodos individuales en el cluster, consulte la [Sección 4.7, “Cómo activar volúmenes lógicos en nodos individuales en un cluster”](#).

4.3.9. Cómo retirar grupos de volúmenes

Para retirar un grupo de volúmenes que no contiene volúmenes lógicos, utilice el comando **vgremove**.

```
# vgremove officevg
Volume group "officevg" successfully removed
```

4.3.10. Separación de un grupo de volúmenes

Para separar el volumen físico de un grupo de volúmenes y crear un nuevo grupo de volúmenes, utilice el comando **vgsplit**.

Los volúmenes lógicos no se puede separar entre grupos de volúmenes. Cada volumen lógico existente debe estar en el volumen físico formando ya sea el antiguo o el nuevo grupo de volúmenes. En caso de ser necesario, sin embargo, puede utilizar el comando **pvmove** para forzar la separación.

El siguiente ejemplo separa al nuevo grupo de volúmenes **smallvg** del grupo de volúmenes original **bigvg**.

```
# vgsplit bigvg smallvg /dev/ram15
Volume group "smallvg" successfully split from "bigvg"
```

4.3.11. Cómo combinar grupos de volúmenes

Para combinar dos grupos de volúmenes en un solo grupo de volúmenes, utilice el comando **vgmerge**. Puede fusionar un volumen "fuente" inactivo con un volumen de "destino" activo o inactivo si los tamaños de las extensiones físicas del volumen son iguales y los resúmenes del volumen físico y lógico de ambos grupos de volúmenes caben en los límites de los volúmenes de destino.

El siguiente comando fusiona el grupo de volúmenes inactivo **my_vg** en el grupo de volúmenes activo o inactivo **databases** dando información de tiempo de ejecución verbosa.

```
vgmerge -v databases my_vg
```

4.3.12. Cómo crear copias de seguridad de los metadatos del grupo de volúmenes

Los archivos y copias de seguridad de los metadatos se crean automáticamente en cada grupo de volúmenes y volúmenes lógicos si la configuración de éstos cambia. Este comportamiento puede ser desactivado en el archivo **lvm.conf**. Por defecto, las copias de seguridad de los metadatos se almacenan en **/etc/lvm/backup** y los archivos de los metadatos se almacenan en **/etc/lvm/archives**. Puede crear copias de seguridad de los metadatos de forma manual al archivo **/etc/lvm/backup** con el comando **vgcfgbackup**.

El comando **vgcfrestore** restaura los metadatos de un grupo de volúmenes desde el archivo a todos los volúmenes físicos en el grupo de volúmenes.

Para obtener un ejemplo del uso del comando **vgcfrestore** para recuperar metadatos de volumen físico, consulte la [Sección 6.4, "Cómo recuperar los metadatos de un volumen físico"](#).

4.3.13. Cómo cambiar el nombre de un grupo de volúmenes

Utilice el comando **vgrename** para renombrar un grupo de volúmenes existente.

Cualquiera de los siguientes comandos cambia el nombre de un grupo de volúmenes existente de **vg02** a **my_volume_group**

```
vgrename /dev/vg02 /dev/my_volume_group
```

```
vgrename vg02 my_volume_group
```

4.3.14. Cómo desplazar un grupo de volúmenes a otro sistema

Puede desplazar un grupo de volúmenes LVM entero a otro sistema. Se recomienda el uso de los comandos **vgexport** y **vgimport** para ejecutar esta tarea.

El comando **vgexport** desactiva un grupo de volúmenes haciéndolo inaccesible al sistema. Ésto permite la separación del volumen físico. El comando **vgimport** hace que la máquina tenga acceso al grupo de volúmenes que fue desactivado previamente con el comando **vgexport**.

Para desplazar un grupo de volúmenes desde un sistema a otro, ejecute los siguientes pasos:

1. Asegúrese de que ningún usuario esté accediendo a los archivos en los volúmenes activos del

grupo de volúmenes, luego desmonte los volúmenes lógicos.

- Utilice el argumento **-a n** del comando **vgchange** para marcar el grupo de volúmenes como inactivo. Esto previene cualquier actividad en el grupo de volúmenes.
- Utilice el comando **vgexport** para exportar el grupo de volúmenes. Así evita que el grupo de volúmenes sea accedido por el sistema del cual lo está quitando.

Después de exportar el grupo de volúmenes, el volumen físico se verá como parte de un grupo de volúmenes a exportar cuando se ejecute el comando **pvscan**, tal y como se muestra en el siguiente ejemplo.

```
[root@tng3-1]# pvscan
PV /dev/sda1   is in exported VG myvg [17.15 GB / 7.15 GB free]
PV /dev/sdc1   is in exported VG myvg [17.15 GB / 15.15 GB free]
PV /dev/sdd1   is in exported VG myvg [17.15 GB / 15.15 GB free]
...
```

Cuando el sistema sea apagado, puede desconectar los discos que forman el grupo de volúmenes y conectarlos en el nuevo sistema.

- Cuando conecte los discos en el nuevo sistema, utilice el comando **vgimport** para importar el grupo de volúmenes y hacerlo accesible al nuevo sistema.
- Active el grupo de volúmenes con el argumento **-a y** del comando **vgchange**.
- Monte el sistema de archivos para que pueda ser utilizado.

4.3.15. Cómo recrear un directorio de grupo de volúmenes

Para recrear el directorio de un grupo de volúmenes y los archivos especiales de un volumen lógico, utilice el comando **vgmknodes**. Este comando revisa los archivos especiales de LVM2 en el directorio **/dev** que son necesarios para volúmenes lógicos activos. Crea cualquier archivo especial que se necesite y remueve aquellos que no se utilizan.

Puede incorporar el comando **vgmknodes** en el comando **vgscan** con el argumento **--mknodes** del comando **vgscan**.

4.4. Administración del volumen lógico

Esta sección describe los comandos que ejecutan varios aspectos de la administración de volúmenes lógicos.

4.4.1. Cómo crear volúmenes lógicos lineales

Para crear un volumen lógico, use el comando **lvcreate**. Si no especifica un nombre para el volumen lógico, se utiliza el nombre predeterminado **lv_{ol}#** en donde **#** es el número interno del volumen lógico.

Cuando cree un volumen lógico, el volumen lógico se forja de un grupo de volúmenes que utiliza las extensiones libres en los volúmenes físicos que conforman el grupo de volúmenes. Normalmente los volúmenes lógicos utilizan el siguiente espacio disponible en el volumen lógico subyacente. Cuando se modifica el volumen lógico el espacio se libera y se asigna de nuevo en el volumen físico.

El siguiente comando crea un volumen lógico de 10GB en el grupo de volúmenes **vg1**.

```
lvcreate -L 10G vg1
```

El siguiente comando crea un volumen lógico lineal de 1500 megabyte llamado **testlv** en el grupo de volúmenes **testvg**, creando el dispositivo de bloque **/dev/testvg/testlv**.

```
lvcreate -L1500 -n testlv testvg
```

El siguiente comando crea un volumen lógico de 50 GB llamado **gfs1v** con las extensiones libres en el grupo de volúmenes **vg0**.

```
lvcreate -L 50G -n gfs1v vg0
```

Puede usar el argumento **-l** del comando **lvcreate** para especificar el tamaño del volumen lógico en extensiones. Esta opción también puede ser usada para especificar el porcentaje del grupo de volúmenes que se usará en el volumen lógico. El siguiente comando crea un volumen lógico llamado **mylv** que utiliza el 60% del espacio total en el grupo de volúmenes **testvol**.

```
lvcreate -l 60%VG -n mylv testvg
```

Puede utilizar el argumento **-l** de **lvcreate** para especificar el porcentaje del espacio libre restante en un grupo de volúmenes como el tamaño del volumen lógico. El siguiente comando crea un volumen lógico llamado **yourlv** que utiliza todo el espacio sin asignar en el grupo de volúmenes **testvol**.

```
lvcreate -l 100%FREE -n yourlv testvg
```

Puede utilizar el argumento **-l** de **lvcreate** para crear un volumen lógico que utiliza todo el grupo de volúmenes. Otra manera de crear un volumen lógico que utiliza todo el grupo de volúmenes es con el comando **vgdisplay** para encontrar el tamaño "PE Total" y usar estos resultados como datos del comando **lvcreate**.

El siguiente comando crea un volumen lógico llamado **mylv** para llenar el grupo de volúmenes llamado **testvg**.

```
# vgdisplay testvg | grep "Total PE"
Total PE          10230
# lvcreate -l 10230 testvg -n mylv
```

El volumen físico subyacente usado para crear un volumen lógico puede ser importante si el volumen físico necesita ser removido, usted debe considerar esta posibilidad cuando cree el volumen lógico. Para obtener mayor información sobre cómo retirar un volumen físico de un grupo de volúmenes, consulte la [Sección 4.3.6, "Cómo retirar un volumen físico de un grupo de volúmenes"](#).

Para crear un volumen lógico que será asignado desde un volumen físico específico en el grupo de volúmenes, especifique el volumen o volúmenes físicos al final de la línea de comandos **lvcreate**. El siguiente comando crea un volumen lógico llamado **testlv** en el grupo de volúmenes **testvg** asignado desde el volumen físico **/dev/sdg1**.

```
lvcreate -L 1500 -ntestlv testvg /dev/sdg1
```

Puede especificar cuáles extensiones de un volumen físico se utilizarán en un volumen lógico. El siguiente ejemplo crea un volumen lógico lineal con las extensiones del volumen físico que van de 0 a 24 **/dev/sda1** y las extensiones que van de 50 a 124 del volumen físico **/dev/sdb1** en el grupo de volúmenes **testvg**.

```
lvcreate -l 100 -n testlv testvg /dev/sda1:0-24 /dev/sdb1:50-124
```

El siguiente ejemplo crea un volumen lógico lineal con las extensiones que van de 0 a 25 del volumen

físico **/dev/sda1** y luego continúa desde la extensión 100.

```
lvcreate -l 100 -n testlv testvg /dev/sda1:0-25:100-
```

La política predeterminada sobre cómo se aplican las extensiones de un volumen lógico es **inherit**, o sea que se aplican las mismas políticas empleadas en el grupo de volúmenes. Estas políticas pueden ser modificadas con el comando **lvchange**. Para obtener mayor información sobre la asignación de políticas, consulte la [Sección 4.3.1, “Cómo crear grupos de volúmenes”](#).

4.4.2. Cómo crear volúmenes entrelazados

Para grandes procesos secuenciales de lectura y escritura, un volumen lógico entrelazado puede mejorar la eficiencia de la E/S de datos. Para obtener información general sobre volúmenes entrelazados, consulte la [Sección 2.3.2, “Volúmenes lógicos entrelazados”](#).

Cuando cree un volumen lógico entrelazado, especifique el número de enlaces con el argumento **-i** del comando **lvcreate**. Esto determina cuántos volúmenes físicos serán entrelazados por el volumen lógico. El número de enlaces no puede ser mayor al número de volúmenes físicos en el grupo de volúmenes (a menos de que se utilice la opción **--alloc anywhere**).

Si los dispositivos físicos subyacentes que conforman el volumen lógico entrelazado tienen diferentes tamaños, el tamaño máximo del volumen entrelazado se determina de acuerdo con el dispositivo subyacente más pequeño. Por ejemplo, en un enlace basado en dos pilares, el tamaño máximo debe tener el doble del tamaño del dispositivo más pequeño. En un enlace que se base en tres pilares, el tamaño máximo es tres veces el tamaño del dispositivo más pequeño.

El siguiente comando crea un volumen lógico entrelazado con dos volúmenes físicos con una banda de 64KB. El volumen lógico tiene 50 GB, se llama **gfslv**, y se forja del grupo de volúmenes **vg0**.

```
lvcreate -L 50G -i2 -I64 -n gfslv vg0
```

Como con los volúmenes lineales, usted puede especificar las extensiones del volumen físico que está utilizando para la franja. El siguiente comando crea un volumen seccionado con 100 extensiones de tamaño que secciona dos volúmenes físicos, se llama **stripelv** y está en el grupo de volúmenes **testvg**. La franja utilizará los sectores 0-49 de **/dev/sda1** y los sectores 50-99 de **/dev/sdb1**.

```
# lvcreate -l 100 -i2 -nstripelv testvg /dev/sda1:0-49 /dev/sdb1:50-99
Using default stripesize 64.00 KB
Logical volume "stripelv" created
```

Volúmenes lógicos LVM de espejo en un cluster

Para crear un volumen lógico LVM en espejo en un cluster se requieren los mismos comandos y procedimientos que sirven para crear un volumen lógico LVM en espejo en un solo nodo. Sin embargo, para crear un volumen LVM en espejo en un cluster, la infraestructura del cluster y el espejo de cluster deben estar ejecutándose, el cluster debe estar en cuorum, y el tipo de bloqueo en el archivo `lvm.conf` debe establecerse correctamente para permitir el bloqueo de cluster. Para ver un ejemplo de una creación de volumen en espejo en un cluster, consulte [Sección 5.5, “Cómo crear un volumen lógico LVM en un cluster”](#).

Si intenta ejecutar la creación en múltiples LVM en espejo y la conversión de comandos en una rápida sucesión de nodos múltiples en un cluster, podría causar una acumulación de comandos. Esto haría que algunas de las operaciones solicitadas expiraran y como consecuencia fallen. Para evitar este problema, se recomienda que la creación de comandos en espejo de cluster se ejecute desde un nodo del cluster.

Cuando cree un volumen en espejo, se debe especificar el número de copias de los datos a realizar con el argumento `-m` del comando `lvcreate`. Si especifica `-m1` creará un espejo, el cual produce dos copias del sistema de archivos: un volumen lógico lineal y una copia. De igual manera, si se especifica `-m2` se crearán dos espejos, con tres copias del sistema de archivos.

El siguiente comando crea un volumen lógico en espejo con un solo espejo. El volumen tiene 50 GB, se llama `mirrorlv`, y se forja del grupo de volúmenes `vg0`:

```
lvcreate -L 50G -m1 -n mirrorlv vg0
```

Un espejo LVM divide en regiones el dispositivo que está siendo copiado, por defecto en regiones predeterminadas de 512KB. Puede usar el argumento `-R` del comando `lvcreate` para especificar en MB el tamaño de la región. También puede cambiar el tamaño predeterminado de la región al modificar la configuración `mirror_region_size` en el archivo `lvm.conf`.

Nota

Debido a las limitaciones en la infraestructura de cluster, los espejos de cluster mayores de 1.5TB no pueden ser creados con el tamaño de región predeterminado de 512KB. Los usuarios que requieren espejos más grandes deben aumentar el tamaño de la región del predeterminado a algo mayor. No poder aumentar el tamaño de la región hará que la creación de LVM se cuelgue o cuelgue otros comandos LVM.

Como una regla general para especificar el tamaño de la región para espejos de más de 1.5TB, puede tomar su tamaño de espejo en TB y redondear ese número a la siguiente potencia de 2, mediante ese número como el argumento `-R` para el comando `lvcreate`. Por ejemplo, si su espejo tiene .5TB, puede especificar `-R 2`. Si su espejo es de 3TB, puede especificar `-R 4`. Para un espejo de 5TB, puede especificar `-R 8`.

El siguiente comando muestra solo el nombre y el tamaño del volumen físico en espejo con un tamaño de región de 2MB.

```
lvcreate -m1 -L 2T -R 2 -n mirror vol_group
```

LVM mantiene un registro pequeño para determinar cuáles regiones están sincronizadas con el espejo o espejos. Por defecto, este registro se almacena en el disco para mantenerlo persistente a través de

los reinicios y garantiza que el espejo no necesite resincronizarse cada vez que la máquina reinicia o falla. En su lugar, puede especificar que este registro sea guardado en memoria con el argumento **--mirrorlog core**. Así se elimina la necesidad de un dispositivo de registro adicional, pero se requiere que el espejo total sea sincronizado tras cada reinicio.

El siguiente comando crea un volumen lógico en espejo desde el grupo de volúmenes **bigvg**. El volumen lógico se llama **ondiskmirvol** y tiene un solo espejo. El volumen tiene 12MB y mantiene el registro de espejo en memoria.

```
# lvcreate -L 12MB -m1 --mirrorlog core -n ondiskmirvol bigvg
Logical volume "ondiskmirvol" created
```

El registro de espejo es creado en un dispositivo independiente de los dispositivos en los cuales se crean los pilares de espejo. Es posible, no obstante, crear el registro de espejo en el mismo dispositivo como uno de los pilares de espejo mediante el argumento **--alloc anywhere** del comando **vgcreate**. Esto puede degradar el rendimiento, pero le permitirá crear un espejo incluso si tiene solo dos dispositivos subyacentes.

El siguiente comando crea un volumen lógico en espejo con un solo espejo para el cual el registro de espejo está en el mismo dispositivo como uno de los pilares de espejo. En este ejemplo, el grupo de volúmenes **vg0** consta solamente de dos dispositivos. Este comando crea un volumen de 500 MB denominado **mirrorlv** en el grupo de volúmenes **vg0**.

```
lvcreate -L 500M -m1 -n mirrorlv -alloc anywhere vg0
```



Nota

Con espejos en cluster, la administración del registro en espejo es responsabilidad del nodo del cluster con el ID de cluster más bajo actualmente. Por lo tanto, cuando el dispositivo que tiene el registro de espejo de cluster no está disponible, en un subconjunto del cluster, el espejo en cluster puede continuar funcionando sin ningún impacto, siempre y cuando el nodo del cluster con el ID más bajo retenga el acceso al registro de espejo. Puesto que el espejo no se disturba, no se realiza ninguna acción automática correctiva. No obstante, cuando el nodo de cluster con ID más bajo pierde acceso al registro de espejo, la acción automática se activa (independientemente de la accesibilidad al registro desde otros nodos).

Para crear un registro de espejo que se cree en espejo, debe especificar el argumento **--mirrorlog mirrored**. El siguiente comando crea un volumen lógico en espejo del grupo de volúmenes **twologvol**. El volumen lógico se llama **twologvol** y tiene un solo espejo. El volumen es de 12MB y el registro de espejo es copiado en espejo con cada registro que se mantiene en un dispositivo independiente.

```
# lvcreate -L 12MB -m1 --mirrorlog mirrored -n twologvol bigvg
Logical volume "twologvol" created
```

De la misma forma que con el registro en espejo estándar, es posible crear registros de espejo redundantes en el mismo dispositivo como los pilares de espejo. Es posible, no obstante, crear el registro de espejo en el mismo dispositivo como uno de los pilares de espejo mediante el argumento **--alloc anywhere** del comando **vgcreate**. Esto puede degradar el rendimiento, pero le permitirá crear un registro de espejo redundante, incluso si no tiene suficientes dispositivos subyacentes para que cada registro se mantenga en un dispositivo independiente de los pilares de espejo.

Cuando un espejo es creado, las regiones del espejo son sincronizadas. Para componentes de espejo grandes, el proceso de sincronización puede tomar algún tiempo. Cuando esté creando un nuevo espejo que no necesita ser reactivado, puede especificar la opción **nosync** para indicar que una sincronización inicial desde el primer dispositivo no es requerida.

Puede especificar los dispositivos a utilizar para los registros del espejo y cuales extensiones del dispositivo deben usarse. Para forzar el registro en un determinado disco, especifique exactamente una de las extensiones en el disco en el que será colocado. LVM no respeta necesariamente el orden en el cual los dispositivos se listan en la línea de comandos. Si un volumen físico se lista, ese es el único espacio en el cual la asignación puede llevarse a cabo. Cualquier extensión física incluida en la lista que ya ha sido asignada será ignorada.

El siguiente comando crea un volumen lógico en espejo con un solo espejo y un registro único que no está en espejo. El volumen es de 500 MB, se llama **mirrorlv**, y se forja del grupo de volúmenes **vg0**. El primer pilar de espejo se encuentra en el dispositivo **/dev/sda1**, el segundo pilar se encuentra en el dispositivo **/dev/sdb1** y el registro de espejo está en **/dev/sdc1**.

```
lvcreate -L 500M -m1 -n mirrorlv vg0 /dev/sda1 /dev/sdb1 /dev/sdc1
```

El siguiente comando crea un volumen lógico en espejo con un solo espejo. El volumen tiene 500 MB, se llama **mirrorlv**, y se forja del grupo de volúmenes **vg0**. El primer pilar de espejo comprende las extensiones de 0 a 499 del dispositivo **/dev/sda1**, el segundo pilar de espejo abarca las extensiones de 0 a 499 de dispositivo **/dev/sdb1** y el registro del espejo inicia en la extensión 0 de **/dev/sdc1**. Estas son extensiones de 1MB. Si alguna de las extensiones especificadas ha sido asignada, serán ignoradas.

```
lvcreate -L 500M -m1 -n mirrorlv vg0 /dev/sda1:0-499 /dev/sdb1:0-499 /dev/sdc1:0
```



Nota

A partir de Red Hat Enterprise Linux 6.1, usted puede combinar RAID0 (en franja) y RAID1 (en espejo) en un solo volumen lógico. La creación de un volumen lógico cuando se especifican simultáneamente el número de espejos (**--mirrors X**) y el número de franjas (**--stripes Y**) produce un dispositivo de espejo cuyos dispositivos constituyentes son seccionados.

4.4.3.1. Política de fallas de volumen lógico en espejo

Puede definir la forma como se comporta un volumen lógico en el evento de una falla de dispositivo con los parámetros **mirror_image_fault_policy** y **mirror_log_fault_policy** en la sección **activation** del archivo **lvm.conf**. Cuando estos parámetros se establecen a **remove**, el sistema intenta retirar el dispositivo que está fallando y ejecutar sin él. Cuando este parámetro se establece a **allocate**, el sistema intenta retirar el dispositivo que está fallando y trata de asignar espacio en un nuevo dispositivo para que sea el remplazo del dispositivo dañado; esta política actúa como la política **remove** si no se puede asignar ningún dispositivo ni espacio apropiado para el remplazo.

Por defecto, el parámetro **mirror_log_fault_policy** se establece a **allocate**. El uso de esta política para el registro es rápido y mantiene la capacidad de recordar el estado de sincronización a través de caídas y reinicios. Si se establece la política a **remove**, cuando un dispositivo de registro falle el espejo se adaptará para usar un registro en memoria y el espejo no recordará su estatus de sincronización a través de caídas y reinicios y el espejo total se resincronizará.

Por defecto, el parámetro **mirror_image_fault_policy** se establece a **remove**. Con esta política,

si una imagen de espejo falla, el espejo se convertirá en un dispositivo sin espejo si solo queda una copia. Si establece esta política a **allocate** para que el dispositivo de espejo requiera que el espejo resincronice los dispositivos, es un proceso lento, pero preserva la característica de espejo del dispositivo.



Nota

Cuando un espejo LVM sufre una falla de dispositivo, se produce la recuperación en dos etapas. La primera etapa tiene que ver con la remoción de dispositivos fallidos. Esto puede hacer que el espejo se reduzca a un dispositivo lineal. La segunda etapa, si el parámetro **mirror_log_fault_policy** se establece a **allocate**, es para intentar reemplazar cualquiera de los dispositivos fallidos. Observe, sin embargo, que no hay garantía de que la segunda etapa elija dispositivos que hayan sido utilizados por el espejo que no ha sido parte de la falla si otros están disponibles.

Para obtener mayor información sobre recuperación manual de una falla de espejo de LVM, consulte [Sección 6.3, “Cómo recuperarse de una falla de un espejo LVM”](#).

4.4.3.2. División fuera de una imagen redundante de un volumen lógico en espejo

Puede dividir una imagen redundante de un volumen lógico en espejo para formar un nuevo volumen lógico. Para dividir la imagen, use el argumento **--splitmirrors** del comando **lvconvert**, especificando el número de imágenes redundantes a dividir. Debe utilizar el argumento **--name** del comando para especificar un nombre para el volumen lógico recién dividido.

El siguiente comando divide un nuevo volumen lógico llamado **copy** desde el volumen lógico en espejo **vg/lv**. El nuevo volumen lógico contiene dos pilares de espejo. En este ejemplo, LVM selecciona los dispositivos a dividir.

```
lvconvert --splitmirrors 2 --name copy vg/lv
```

Puede especificar los dispositivos a dividir. El siguiente comando separa un nuevo volumen lógico llamado **copy** del volumen lógico en espejo **vg/lv**. El nuevo volumen lógico contiene dos pilares de espejo que constan de los dispositivos **/dev/sdc1** y **/dev/sde1**.

```
lvconvert --splitmirrors 2 --name copy vg/lv /dev/sd[ce]1
```

4.4.3.3. Cómo reparar volúmenes en espejo

Puede utilizar el comando **lvconvert --repair** para reparar un espejo después de una falla de disco. De esta manera devuelve al espejo a un estado consistente. El comando **lvconvert --repair** es un comando interactivo que le pide indicar si desea que el sistema intente reemplazar los dispositivos fallidos.

- ▶ Para ignorar los indicadores y reemplazar todos los dispositivos fallidos, especifique la opción **-y** en la línea de comandos.
- ▶ Para ignorar los indicadores y no reemplazar ninguno de los dispositivos fallidos, especifique la opción **-f** en la línea de comandos.
- ▶ Para ignorar los indicadores e indicar aún políticas diferentes de reemplazo para imagen en espejo y el registro en espejo, especifique el argumento **--use-policies** a usar las política de reemplazo de dispositivo especificadas por los parámetros **mirror_log_fault_policy** y **mirror_device_fault_policy** en el archivo **lvm.conf**.

4.4.3.4. Cómo cambiar la configuración del volumen en espejo

Puede convertir un volumen lógico de volumen en espejo a volumen lineal o viceversa con el comando **lvconvert**. También puede utilizar este comando para reconfigurar otros parámetros de espejo de un volumen lógico existente, tal como **corelog**.

Al convertir un volumen lógico a un volumen con espejos, se crean pilares de espejo para el volumen existente. Esto quiere decir que el grupo de volúmenes debe tener espacio suficiente para los pilares de espejo y su registro.

Si se pierde un pilar de espejo, LVM convierte el volumen a lineal para que aún se pueda tener acceso al volumen, sin redundancia de espejo. Una vez el pilar es remplazado, se puede utilizar el comando **lvconvert** para restaurar el espejo. Este procedimiento se provee en [Sección 6.3, “Cómo recuperarse de una falla de un espejo LVM”](#).

El siguiente comando convierte el volumen lógico lineal **vg00/lvol1** en un volumen lógico en espejo.

```
lvconvert -m1 vg00/lvol1
```

El siguiente comando convierte el volumen lógico en espejo **vg00/lvol1** en un volumen lógico lineal, quitando el pilar de espejo.

```
lvconvert -m0 vg00/lvol1
```

4.4.4. Cómo crear volúmenes de instantáneas

Utilice la opción **-s** del comando **lvcreate** para crear un volumen de instantánea. Un volumen de instantánea tiene permisos de escritura.

Nota

Las instantáneas LVM no están soportadas a través de los nodos en un cluster. No se puede crear un volumen de instantáneas en un grupo de volumen en cluster. A partir de Red Hat Enterprise Linux 6.1, sin embargo, si usted necesita crear una copia de seguridad de datos consistente en un volumen lógico en cluster, puede activar exclusivamente el volumen y luego crear la instantánea. Para obtener mayor información sobre cómo activar exclusivamente volúmenes lógicos en un nodo, consulte [Sección 4.7, “Cómo activar volúmenes lógicos en nodos individuales en un cluster”](#).

Nota

A partir de Red Hat Enterprise Linux 6.1, las instantáneas LVM están soportadas para volúmenes lógicos en espejo.

El siguiente comando crea un volumen lógico de instantáneas que tiene 100 MB y se llama **/dev/vg00/snap**. Esto crea una instantánea del volumen lógico original **/dev/vg00/lvol1**. Si el volumen lógico original contiene un sistema de archivos, puede montar el volumen lógico de la instantánea en un directorio cualquiera para poder tener acceso al sistema de archivos y crear una copia de seguridad mientras el sistema de archivos continúa siendo actualizado.

```
lvcreate --size 100M --snapshot --name snap /dev/vg00/lvol1
```

Después de crear un volumen lógico de instantánea, especifique el volumen original en el comando **lvdisplay** para ver la salida que incluye una lista de todos los volúmenes lógicos de instantáneas y sus estados (activo o inactivo).

El siguiente ejemplo muestra el estado del volumen lógico **/dev/new_vg/lvol10**, para el cual el volumen de instantánea **/dev/new_vg/newvgsnap** ha sido creado.

```
# lvdisplay /dev/new_vg/lvol10
--- Logical volume ---
LV Name                /dev/new_vg/lvol10
VG Name                new_vg
LV UUID                LBy1Tz-sr23-0jsI-LT03-nHLC-y8XW-EhC178
LV Write Access        read/write
LV snapshot status     source of
                       /dev/new_vg/newvgsnap1 [active]

LV Status              available
# open                 0
LV Size                52.00 MB
Current LE             13
Segments               1
Allocation             inherit
Read ahead sectors     0
Block device           253:2
```

El comando **lvs**, por defecto, muestra el volumen original y el porcentaje actual del volumen de instantánea que está siendo usado para cada volumen de instantánea. El siguiente ejemplo muestra la salida predeterminada para el comando **lvs** para un sistema que incluye el volumen lógico **/dev/new_vg/lvol10**, por el cual un volumen de instantánea **/dev/new_vg/newvgsnap** ha sido creado.

```
# lvs
LV          VG      Attr   LSize  Origin Snap%  Move Log Copy%
lvol10     new_vg  owi-a- 52.00M
newvgsnap1 new_vg  swi-a- 8.00M  lvol10  0.20
```



Nota

Como la instantánea aumenta de tamaño mientras el volumen original cambia, es importante monitorizar el porcentaje del volumen de instantánea con el comando **lvs** para asegurarse de que no esté llena. Una instantánea 100% llena se pierde completamente, debido a que la escritura a partes sin cambios del origen no podrá ocurrir sin dañar la instantánea.

4.4.5. Cómo fusionar volúmenes de instantáneas

A partir del lanzamiento de Red Hat Enterprise Linux 6, usted puede usar la opción **--merge** del comando **lvconvert** para fusionar una instantánea en su volumen de origen. Si el origen y el volumen de instantánea no están abiertos, la fusión iniciará inmediatamente. De lo contrario, la fusión iniciará la primera vez que el origen o la instantánea estén activadas y ambas estén cerradas. La fusión de una instantánea en un origen que no se puede cerrar, por ejemplo, un sistema de archivos de root, se difiere hasta la próxima vez que el volumen de origen sea activado. Cuando la fusión inicia, el volumen lógico resultante tendrá el nombre de origen, un número menor y un UUID. Mientras la fusión está en curso,

lee o escribe al origen aparecen como si se dirigieran a la instantánea que está siendo fusionada. Cuando la fusión finaliza, la instantánea de fusión se remueve.

El siguiente comando fusiona el volumen de instantánea **vg00/lvol1_snap** dentro de su origen.

```
lvconvert --merge vg00/lvol1_snap"
```

Puede especificar múltiples instantáneas en la línea de comandos o puede usar las etiquetas de objetos LVM para especificar que múltiples instantáneas sean fusionadas a sus respectivos orígenes. En el siguiente ejemplo, los volúmenes lógicos **vg00/lvol1**, **vg00/lvol2** y **vg00/lvol3** serán nombrados con la etiqueta **@some_tag**. El siguiente comando fusiona en serie los volúmenes lógicos de instantáneas para todos los tres volúmenes: **vg00/lvol1**, luego **vg00/lvol2**, después **vg00/lvol3**. Si la opción **--background** se utilizara, toda la fusión de volúmenes lógicos de instantáneas se iniciaría en paralelo.

```
lvconvert --merge @some_tag"
```

Para información sobre etiquetas de objetos LVM, consulte el [Apéndice C, Etiquetas de objetos LVM](#). Para obtener mayor información sobre el comando **lvconvert --merge**, consulte la página de manual (8) **lvconvert**

4.4.6. Números de dispositivos persistentes

Los números de dispositivo mayor y menor son asignados dinámicamente al cargar el módulo. Algunas aplicaciones funcionan mejor si el dispositivo de bloque es activado siempre con los mismos números de dispositivos (menor y mayor). Puede especificar estos con los comandos **lvcreate** y **lvchange** pasando las siguientes opciones:

```
--persistent y --major major --minor minor
```

Utilice un número menor grande para asegurarse de que no ha sido asignado a otro dispositivo de forma dinámica.

Si está exportando un sistema de archivos con NFS, especifique el parámetro **fsid** en el archivo a exportar para evitar establecer un número de dispositivo persistente dentro de LVM.

4.4.7. Cómo redimensionar volúmenes lógicos

Para reducir el tamaño de un volumen lógico, utilice el comando **lvreduce**. Si el volumen lógico contiene un sistema de archivos, reduzca el sistema de archivos primero (o utilice la interfaz gráfica de LVM) para que el volumen lógico sea tan grande como el sistema de archivos lo espera.

El siguiente comando reduce el tamaño del volumen lógico **lvol1** en el grupo de volúmenes **vg00** por tres extensiones lógicas.

```
lvreduce -l -3 vg00/lvol1
```

4.4.8. Cómo cambiar los parámetros de un grupo de volúmenes lógico

Para cambiar parámetros de un volumen lógico utilice el comando **lvchange**. Para obtener una lista de los parámetros que pueden ser modificados, consulte la página de manual de **lvchange(8)**.

Puede utilizar el comando **lvchange** para activar y desactivar los volúmenes lógicos. Para activar y desactivar todos los volúmenes lógicos en un grupo de volúmenes al mismo tiempo, utilice el comando

vgchange, así como se describe en la [Sección 4.3.7, “Cómo cambiar los parámetros de un grupo de volúmenes”](#).

El siguiente comando cambia los permisos en el volumen **lvo11** en el grupo de volúmenes **vg00** para que sea de solo lectura.

```
lvchange -pr vg00/lvo11
```

4.4.9. Cómo renombrar volúmenes lógicos

Para renombrar un volumen lógico existente, utilice el comando **lvrename**.

Cualquiera de los siguientes comando cambia el nombre del volumen lógico **lvo1d** en el grupo de volúmenes **vg02** a **lvnew**.

```
lvrename /dev/vg02/lvo1d /dev/vg02/lvnew
```

```
lvrename vg02 lvo1d lvnew
```

Para obtener mayor información sobre cómo activar volúmenes lógicos en nodos individuales en un cluster, consulte la [Sección 4.7, “Cómo activar volúmenes lógicos en nodos individuales en un cluster”](#).

4.4.10. Cómo retirar volúmenes lógicos

Para retirar un volumen lógico inactivo, utilice el comando **lvremove**. Si el volumen lógico está actualmente montado, desmóntelo antes de removerlo. Además, si está en un entorno en cluster debe desactivar el volumen lógico antes removerlo.

El siguiente comando remueve el volumen lógico **/dev/testvg/testlv** del grupo de volúmenes **testvg**. Observe que en este caso el volumen lógico no ha sido desactivado.

```
[root@tng3-1 lvm]# lvremove /dev/testvg/testlv
Do you really want to remove active logical volume "testlv"? [y/n]: y
Logical volume "testlv" successfully removed
```

Puede desactivar de forma explícita el volumen lógico antes de removerlo con el comando **lvchange -an**. En este caso no verá el mensaje que pregunta si desea retirar el volumen lógico.

4.4.11. Cómo ver los volúmenes lógicos

Hay tres comandos que pueden utilizarse para ver las propiedades de los volúmenes lógicos LVM: **lvs**, **lvdisplay** y **lvscan**.

El comando **lvs** proporciona información sobre el volumen lógico de una forma configurable, mostrando cada volumen lógico en una línea. El comando **lvs** ofrece gran control del formato y sirve para scripts. Para obtener información sobre el uso del comando **lvs** para personalizar su salida, consulte la [Sección 4.8, “Reporte personalizado para LVM”](#).

El comando **lvdisplay** muestra las propiedades del volumen lógico (como tamaño, distribución y asignación) en un formato fijo.

El siguiente comando muestra los atributos de **lvo12** en **vg00**. Si se han creado volúmenes lógicos de instantánea para este volumen lógico original, este comando muestra una lista de todos los volúmenes lógicos de instantáneas y sus estatus (activo o inactivo).

```
lvdisplay -v /dev/vg00/lvol2
```

El comando **lvscan** busca los volúmenes lógicos en el sistema y los lista, como en el siguiente ejemplo.

```
# lvscan
ACTIVE                               '/dev/vg00/gfslv' [1.46 GB] inherit
```

4.4.12. Cómo aumentar los volúmenes lógicos

Para incrementar el tamaño de los volúmenes lógicos, utilice el comando **lvextend**.

Cuando extienda el volumen lógico, puede especificar cuánto desea añadir al volumen o qué tan grande desea que el volumen sea después de la extensión.

El siguiente comando extiende el volumen lógico **/dev/myvg/homevol** a 12 GB.

```
# lvextend -L12G /dev/myvg/homevol
lvextend -- extending logical volume "/dev/myvg/homevol" to 12 GB
lvextend -- doing automatic backup of volume group "myvg"
lvextend -- logical volume "/dev/myvg/homevol" successfully extended
```

El siguiente comando añade otro gigabyte al volumen lógico **/dev/myvg/homevol**.

```
# lvextend -L+1G /dev/myvg/homevol
lvextend -- extending logical volume "/dev/myvg/homevol" to 13 GB
lvextend -- doing automatic backup of volume group "myvg"
lvextend -- logical volume "/dev/myvg/homevol" successfully extended
```

Del mismo modo que **lvcreate**, puede utilizar la opción **-l** del comando **lvextend** para especificar el número de extensiones con las cuales incrementar el tamaño del volumen lógico. Puede asimismo utilizar el argumento para especificar un porcentaje del grupo de volúmenes o un porcentaje del espacio libre del grupo de volúmenes. El siguiente comando extiende el volumen lógico para llenar todo el espacio sin asignar del grupo de volúmenes **myvg**.

```
[root@tng3-1 ~]# lvextend -l +100%FREE /dev/myvg/testlv
Extending logical volume testlv to 68.59 GB
Logical volume testlv successfully resized
```

Después de extender el volumen lógico, es necesario incrementar el tamaño del sistema de archivos para que coincida.

Por defecto, la mayoría de herramientas para redimensionar los sistemas de archivos incrementarán en sistema de archivos al tamaño del volumen lógico subyacente. Así, no es necesario especificar el mismo tamaño para cada uno de los dos comandos.

4.4.12.1. Cómo extender un volumen entrelazado

Para incrementar el tamaño de un volumen lógico entrelazado, debe haber suficiente espacio libre en el volumen físico subyacente que conforma el grupo de volúmenes donde está en enlace. Por ejemplo, si tiene dos enlaces que utilizan un grupo de volúmenes en su totalidad, un solo volumen físico añadido al grupo de volúmenes no es suficiente para extender el enlace. Es necesario añadir al menos dos volúmenes físicos al grupo de volúmenes.

Por ejemplo, considere un grupo de volúmenes **vg** conformado de dos volúmenes físicos como se

muestra con el siguiente comando **vgs**.

```
# vgs
VG   #PV #LV #SN Attr   VSize   VFree
vg   2   0   0 wz--n- 271.31G 271.31G
```

Puede crear un enlace utilizando todo el espacio en un grupo de volúmenes.

```
# lvcreate -n stripe1 -L 271.31G -i 2 vg
Using default stripesize 64.00 KB
Rounding up size to full physical extent 271.31 GB
Logical volume "stripe1" created
# lvs -a -o +devices
LV      VG   Attr   LSize   Origin Snap%   Move Log Copy%  Devices
stripe1 vg   -wi-a- 271.31G
/dev/sda1(0),/dev/sdb1(0)
```

Tenga en cuenta que el grupo de volúmenes ahora no tiene más espacio libre.

```
# vgs
VG   #PV #LV #SN Attr   VSize   VFree
vg   2   1   0 wz--n- 271.31G   0
```

El siguiente comando añade otro volumen físico al grupo de volúmenes, el cual tiene ahora 135G de espacio adicional

```
# vgextend vg /dev/sdc1
Volume group "vg" successfully extended
# vgs
VG   #PV #LV #SN Attr   VSize   VFree
vg   3   1   0 wz--n- 406.97G 135.66G
```

En este momento no se puede extender el volumen lógico entrelazado al tamaño total del grupo de volúmenes porque se necesitan dos dispositivos subyacentes para enlazar los datos.

```
# lvextend vg/stripe1 -L 406G
Using stripesize of last segment 64.00 KB
Extending logical volume stripe1 to 406.00 GB
Insufficient suitable allocatable extents for logical volume stripe1: 34480
more required
```

Para extender el volumen lógico entrelazado, añada otro volumen físico y luego extienda el volumen lógico. En este ejemplo, después de añadir dos volúmenes físicos al grupo de volúmenes podemos extender el volumen lógico del grupo de volúmenes.

```
# vgextend vg /dev/sdd1
Volume group "vg" successfully extended
# vgs
VG   #PV #LV #SN Attr   VSize   VFree
vg   4   1   0 wz--n- 542.62G 271.31G
# lvextend vg/stripe1 -L 542G
Using stripesize of last segment 64.00 KB
Extending logical volume stripe1 to 542.00 GB
Logical volume stripe1 successfully resized
```

Si no tiene suficientes dispositivos físicos para extender el volumen lógico entrelazado, se puede

extender el volumen de todas maneras si no importa que la extensión no sea entrelazada. Esto puede conllevar a un rendimiento desbalanceado. Cuando se añade espacio al volumen lógico, la operación predeterminada es utilizar los mismos parámetros de enlace usados en el último segmento del volumen lógico existente. Sin embargo, usted puede sobrescribir esos parámetros. El siguiente ejemplo extiende el volumen lógico entrelazado para utilizar el espacio libre restante después de que el comando **lvextend** inicial falla.

```
# lvextend vg/stripel -L 406G
Using stripesize of last segment 64.00 KB
Extending logical volume stripel to 406.00 GB
Insufficient suitable allocatable extents for logical volume stripel: 34480
more required
# lvextend -i1 -l+100%FREE vg/stripel
```

4.4.12.2. Extender un volumen lógico con la política de asignación **cling**

Al extender un volumen LVM, usted puede usar la opción **--alloc cling** del comando **lvextend** para especificar la política de asignación **cling**. Dicha política elige el espacio en los mismos volúmenes físicos como el último segmento del volumen lógico existente. Si hay espacio insuficiente en los volúmenes físicos y una lista de etiquetas se define en el archivo **lvm.conf**, LVM revisará si alguno de los pilares se conecta con los volúmenes físicos y busca las etiquetas de volúmenes físicos coincidentes entre extensiones existentes y nuevas.

Por ejemplo, si usted tiene volúmenes lógicos que están en espejo entre dos sitios dentro de un solo grupo de volumen, puede nombrar los volúmenes físicos según el lugar en donde están localizados etiquetándolos con **@site1** y **@site2** y especificando la siguiente línea en el archivo **lvm.conf**:

```
cling_tag_list = [ "@site1", "@site2" ]
```

Para obtener mayor información sobre cómo etiquetar volúmenes físicos, consulte [Apéndice C, Etiquetas de objetos LVM](#).

En el siguiente ejemplo, el archivo **lvm.conf** ha sido modificado para que contenga la siguiente línea:

```
cling_tag_list = [ "@A", "@B" ]
```

También en este ejemplo, se ha creado el grupo de volumen **taft** el cual consta de los volúmenes físicos **/dev/sdb1**, **/dev/sdc1**, **/dev/sdd1**, **/dev/sde1**, **/dev/sdf1**, **/dev/sgd1**, y **/dev/sdh1**. Estos volúmenes físicos han sido etiquetados como **A**, **B**, y **C**. El ejemplo no usa la etiqueta **C**, pero muestra que LVM usa las etiquetas para seleccionar qué volúmenes físicos usar para pilares de espejo.

```
[root@taft-03 ~]# pvs -a -o +pv_tags /dev/sd[bcdefgh]1
PV          VG      Fmt  Attr PSize  PFree  PV Tags
/dev/sdb1   taft   lvm2 a-    135.66g 135.66g A
/dev/sdc1   taft   lvm2 a-    135.66g 135.66g B
/dev/sdd1   taft   lvm2 a-    135.66g 135.66g B
/dev/sde1   taft   lvm2 a-    135.66g 135.66g C
/dev/sdf1   taft   lvm2 a-    135.66g 135.66g C
/dev/sgd1   taft   lvm2 a-    135.66g 135.66g A
/dev/sdh1   taft   lvm2 a-    135.66g 135.66g A
```

El siguiente comando crea un volumen en espejo de 100G desde el grupo de volumen **taft**.


```
[root@taft-03 ~]# lvcreate -m 1 -n mirror --nosync -L 100G taft
```

El siguiente comando muestra los dispositivos utilizados para los pilares y registro de espejo.

```
[root@taft-03 ~]# lvs -a -o +devices
LV          VG      Attr  LSize   Log           Copy%  Devices
mirror      taft    Mwi-a- 100.00g mirror_mlog 100.00
mirror_mimage_0(0),mirror_mimage_1(0)
[mirror_mimage_0] taft    iwi-ao 100.00g           /dev/sdb1(0)
[mirror_mimage_1] taft    iwi-ao 100.00g           /dev/sdc1(0)
[mirror_mlog]   taft    lwi-ao  4.00m           /dev/sdh1(0)
```

El siguiente comando extiende el volumen en espejo, mediante la política de asignación **cling** para indicar que los pilares de espejo deben extenderse mediante los volúmenes físicos con la misma etiqueta.

```
[root@taft-03 ~]# lvextend --alloc cling -L +100G taft/mirror
Extending 2 mirror images.
Extending logical volume mirror to 200.00 GiB
Logical volume mirror successfully resized
```

El siguiente comando muestra que los pilares de espejo han sido extendidos mediante volúmenes físicos con la misma etiqueta del pilar. Observe que los volúmenes físicos con la etiqueta **C** se omitieron.

```
[root@taft-03 ~]# lvs -a -o +devices
LV          VG      Attr  LSize   Log           Copy%  Devices
mirror      taft    Mwi-a- 200.00g mirror_mlog  50.16
mirror_mimage_0(0),mirror_mimage_1(0)
[mirror_mimage_0] taft    Iwi-ao 200.00g           /dev/sdb1(0)
[mirror_mimage_0] taft    Iwi-ao 200.00g           /dev/sdg1(0)
[mirror_mimage_1] taft    Iwi-ao 200.00g           /dev/sdc1(0)
[mirror_mimage_1] taft    Iwi-ao 200.00g           /dev/sdd1(0)
[mirror_mlog]   taft    lwi-ao  4.00m           /dev/sdh1(0)
```

4.4.13. Cómo reducir volúmenes lógicos

Para reducir el tamaño de un volumen lógico, primero desmote el sistema de archivos. Puede luego utilizar el comando **lvreduce** para reducir el volumen. Después de reducir el volumen, monte de nuevo el sistema de archivos.



Advertencia

Es importante reducir el tamaño del sistema de archivos o lo que esté residiendo en el volumen antes de reducirlo; de lo contrario, los datos podrían perderse.

Al reducir el volumen lógico se libera espacio del grupo de volúmenes para que pueda ser asignado a otro volumen lógico.

El siguiente ejemplo reduce el tamaño del volumen lógico **lv011** en el grupo de volúmenes **vg00** a tres extensiones lógicas.

```
lvreduce -l -3 vg00/lv011
```

4.5. Cómo controlar la exploración de dispositivos LVM a través de filtros

Durante el arranque, el comando **vgscan** se ejecuta para explorar el dispositivo de bloque en el sistema en busca de etiquetas LVM para determinar cuales de estas son volúmenes físicos y para leer los metadatos y construir una lista de grupos de volúmenes. El nombre de los volúmenes físicos son almacenados en el archivo de caché de cada nodo en el sistema, **/etc/lvm/.cache**. Otros comandos podrán leer este archivo para evitar explorar el sistema nuevamente.

Puede controlar que los dispositivos LVM al configurar los filtros en el archivo de configuración **lvm.conf**. Los filtros constan de una serie de expresiones regulares sencillas que se aplican a los nombres de dispositivos en el directorio **/dev** para decidir si aceptan o rechazan cada dispositivo de bloque encontrado.

Los siguientes ejemplos muestran el uso de filtros para controlar qué dispositivos LVM explorar. Observe que algunos de estos ejemplos no representan necesariamente la mejor práctica, ya que las expresiones regulares se combinan libremente con el nombre de la ruta completa. Por ejemplo, **a/loop/** es equivalente a **a/. *loop.* /** y coincidirá con **/dev/solooperation/lvol1**.

El siguiente filtro añade todos los dispositivos descubiertos, el cual es el comportamiento predeterminado cuando no hay filtros en el archivo de configuración:

```
filter = [ "a./.*/" ]
```

El siguiente filtro remueve el dispositivo de cdrom para evitar retrasos si el dispositivo no tiene medios:

```
filter = [ "r|/dev/cdrom|" ]
```

El siguiente filtro añade todos los dispositivos en bucle pero remueve el resto de dispositivos de bloque:

```
filter = [ "a/loop.*/", "r/.*/" ]
```

El siguiente filtro añade todos los dispositivos en bucle e IDE pero remueve el resto de dispositivos de bloque:

```
filter =[ "a|loop.*|", "a|/dev/hd.*|", "r|.*/" ]
```

El siguiente filtro añade la partición 8 en el primer disco IDE pero remueve el resto de dispositivos de bloque:

```
filter = [ "a|^/dev/hda8$|", "r/.*/" ]
```

Para obtener mayor información sobre el archivo de configuración **lvm.conf**, consulte el [Apéndice B, Archivos de configuración LVM](#) y la página de manual (5) **lvm.conf**.

4.6. Asignación de datos en línea

Puede desplazar los datos mientras el sistema está en uso con el comando **pvmove**

El comando **pvmove** divide los datos a desplazar en secciones y crea espejos temporales para mover cada sección. Para obtener mayor información sobre la operación del comando **pvmove**, consulte la

página del manual de **pvmove**(8).

El siguiente comando mueve los espacios asignados del volumen físico **/dev/sdc1** a otros volúmenes físicos en el grupo de volúmenes:

```
pvmove /dev/sdc1
```

El siguiente comando mueve solo las extensiones del volumen lógico **MyLV**.

```
pvmove -n MyLV /dev/sdc1
```

Como el comando **pvmove** puede tardar mucho en ejecutarse, puede ejecutar el comando en el trasfondo para evitar mostrar el progreso en el primer plano. El siguiente comando traslada todas las extensiones asignadas al volumen físico **/dev/sdc1** a **/dev/sdf1** en el trasfondo.

```
pvmove -b /dev/sdc1 /dev/sdf1
```

El siguiente comando reporta el progreso de los movimientos como un porcentaje en intervalos de cinco segundos.

```
pvmove -i5 /dev/sdd1
```

4.7. Cómo activar volúmenes lógicos en nodos individuales en un cluster

Si tiene LVM instalado en un entorno de cluster, podría a veces necesitar activar los volúmenes lógicos exclusivamente en un nodo.

Para activar volúmenes lógicos de forma exclusiva en un nodo, utilice el comando **lvchange -aey**. Alternativamente, puede utilizar el comando **lvchange -aly** para activar el volumen lógico únicamente en el nodo local pero no exclusivamente. Posteriormente puede activarlos en nodos adicionales de forma concurrente.

Puede activar también los volúmenes lógicos en nodos individuales utilizando las etiquetas LVM, descritas en el . Se puede también especificar la activación de nodos en el archivo de configuración, este método se describe en el [Apéndice B, Archivos de configuración LVM](#).

4.8. Reporte personalizado para LVM

Se pueden crear reportes concisos y personalizados de los objetos LVM con los comandos **pvs**, **lvs** y **vgs**. El reporte que estos comandos generan incluye una línea de salida por cada objeto. Cada línea contiene una lista ordenada de campos de propiedades relacionadas con el objeto. Hay cinco formas de seleccionar los objetos que serán reportados: por volumen físico, por grupos de volúmenes, por segmentos de volúmenes físicos y segmentos de volúmenes lógicos.

Las siguientes secciones proporcionan:

- ▶ Un resumen de opciones de comando para controlar el formato del reporte generado.
- ▶ Una lista de los campos que puede seleccionar en cada objeto LVM.
- ▶ Un resumen de las opciones del comando usados para ordenar el reporte generado.
- ▶ Instrucciones para especificar las unidades de la salida del reporte.

4.8.1. Control del formato

El comando **pvs**, **lvs** o **vgs** usado determina el conjunto de campos predeterminados a mostrar y el orden en que éstos serán mostrados. Puede controlar la salida de estos comando con las siguientes opciones:

- Puede hacer que el campo muestre algo diferente al valor predeterminado con el argumento **-o**. Por ejemplo, el siguiente mensaje de salida utiliza los valores predeterminados para el comando **pvs** (el cual muestra información sobre los volúmenes físicos).

```
# pvs
PV          VG      Fmt  Attr PSize  PFree
/dev/sdb1   new_vg  lvm2 a-   17.14G 17.14G
/dev/sdc1   new_vg  lvm2 a-   17.14G 17.09G
/dev/sdd1   new_vg  lvm2 a-   17.14G 17.14G
```

El siguiente comando muestra solo el nombre y el tamaño del volumen físico.

```
# pvs -o pv_name,pv_size
PV          PSize
/dev/sdb1   17.14G
/dev/sdc1   17.14G
/dev/sdd1   17.14G
```

- Puede añadir un campo al mensaje de salida con el signo de adición (+), el cual es usado en combinación con la opción **-o**.

El siguiente ejemplo muestra el UUID del volumen físico además de los campos predeterminados.

```
# pvs -o +pv_uuid
PV          VG      Fmt  Attr PSize  PFree  PV UUID
/dev/sdb1   new_vg  lvm2 a-   17.14G 17.14G onFF2w-1fLC-ughJ-D9eB-M7iv-6XqA-
dqGeXY
/dev/sdc1   new_vg  lvm2 a-   17.14G 17.09G Joq1ch-yWSj-kuEn-IdwM-01S9-X08M-
mcpsVe
/dev/sdd1   new_vg  lvm2 a-   17.14G 17.14G yvfVzK-Cf31-j75k-dECm-0RZ3-0dGW-
UqkCS
```

- La opción **-v** incluye algún campo adicional. Por ejemplo, el comando **pvs -v** mostrará los campos **DevSize** y **PV UUID** además de los campos predeterminados.

```
# pvs -v
Scanning for physical volume names
PV          VG      Fmt  Attr PSize  PFree  DevSize PV UUID
/dev/sdb1   new_vg  lvm2 a-   17.14G 17.14G 17.14G onFF2w-1fLC-ughJ-D9eB-
M7iv-6XqA-dqGeXY
/dev/sdc1   new_vg  lvm2 a-   17.14G 17.09G 17.14G Joq1ch-yWSj-kuEn-IdwM-
01S9-X08M-mcpsVe
/dev/sdd1   new_vg  lvm2 a-   17.14G 17.14G 17.14G yvfVzK-Cf31-j75k-dECm-
0RZ3-0dGW-tUqkCS
```

- El argumento **--noheadings** suprime la línea del encabezado. Esta opción puede ser útil en scripts.

El siguiente ejemplo utiliza el argumento **--noheadings** en combinación con **pv_name**, el cual genera una lista de todos los volúmenes físicos.

```
# pvs --noheadings -o pv_name
/dev/sdb1
/dev/sdc1
/dev/sdd1
```

- El argumento **--separator *separador*** utiliza ***separador*** para separar cada campo.

El siguiente ejemplo separa los campos de salida predeterminada del comando **pvs** con el signo de igualdad (=).

```
# pvs --separator =
PV=VG=Fmt=Attr=PSize=PFree
/dev/sdb1=new_vg=lvm2=a-=17.14G=17.14G
/dev/sdc1=new_vg=lvm2=a-=17.14G=17.09G
/dev/sdd1=new_vg=lvm2=a-=17.14G=17.14G
```

Para mantener los campos alineados cuando se utiliza el argumento **separator** utilice el argumento **separator** junto con el argumento **--aligned**.

```
# pvs --separator = --aligned
PV      =VG      =Fmt =Attr=PSize =PFree
/dev/sdb1 =new_vg=lvm2=a- =17.14G=17.14G
/dev/sdc1 =new_vg=lvm2=a- =17.14G=17.09G
/dev/sdd1 =new_vg=lvm2=a- =17.14G=17.14G
```

Puede utilizar el argumento **-P** del comando **lvs** o **vgs** para mostrar la información de un volumen fallido que de otra manera no aparecería en la salida. Para obtener mayor información sobre este argumento, consulte la [Sección 6.2, “Cómo ver la información en dispositivos fallidos”](#).

Para obtener una lista completa de los argumentos de visualización, consulte las páginas de manual de **pvs(8)**, **vgs(8)** y **lvs(8)**.

Los campos del grupo de volúmenes se pueden mezclar con los campos del volumen físico (y segmentos del volumen físico) o con campos del volumen lógico (y segmentos del volumen físico), pero los campos del volumen físico y del volumen lógico no pueden mezclarse. Por ejemplo, el siguiente comando mostrará una línea de salida para cada volumen físico.

```
# vgs -o +pv_name
VG      #PV #LV #SN Attr   VSize VFree PV
new_vg   3   1   0 wz--n- 51.42G 51.37G /dev/sdc1
new_vg   3   1   0 wz--n- 51.42G 51.37G /dev/sdd1
new_vg   3   1   0 wz--n- 51.42G 51.37G /dev/sdb1
```

4.8.2. Selección de objetos

Esta sección proporciona una serie de tablas que listan la información que se puede mostrar sobre los objetos LVM con los comandos **pvs**, **vgs** y **lvs**.

Por conveniencia, un prefijo de nombre de campo puede ser dejado de lado si coincide con el valor predeterminado del comando. Por ejemplo, con el comando **pvs**, **name** significa **pv_name**, pero con el comando **vgs**, **name** se interpreta como **vg_name**.

La ejecución de los siguientes comandos es similar a la ejecución de **pvs -o pv_free**.

```
# pvs -o +free
PFree
17.14G
17.09G
17.14G
```

El comando pvs

[Tabla 4.1, “Campos de visualización de pvs”](#) lista las opciones de visualización del comando **pvs**, junto con el nombre del campo tal y como aparece en el encabezado y una descripción del campo.

Tabla 4.1. Campos de visualización de pvs

Argumento	Encabezado	Descripción
dev_size	DevSize	El tamaño del dispositivo subyacente en el cual el volumen físico fue creado
pe_start	1st PE	Balance al inicio de la primera extensión física en el dispositivo subyacente
pv_attr	Attr	Estado del volumen físico: asignable (a), exportado (x).
pv_fmt	Fmt	El formato de metadatos del volumen físico (lvm2 o lvm1)
pv_free	PFree	El espacio libre restante en el volumen físico
pv_name	PV	El nombre del volumen físico
pv_pe_alloc_count	Alloc	Número de extensiones físicas usadas
pv_pe_count	PE	Número de extensiones físicas
pvseg_size	SSize	El tamaño de segmento del volumen físico
pvseg_start	Start	La extensión física de inicio del segmento del volumen físico
pv_size	PSize	El tamaño del volumen físico
pv_tags	PV Tags	Etiquetas LVM añadidas al volumen físico
pv_used	Used	La cantidad de espacio actualmente usada en el volumen físico
pv_uuid	PV UUID	El UUID del volumen físico

El comando **pvs** muestra los siguientes campos por defecto: **pv_name**, **vg_name**, **pv_fmt**, **pv_attr**, **pv_size**, **pv_free**. La salida es ordenada según el campo **pv_name**.

```
# pvs
PV          VG      Fmt  Attr PSize  PFree
/dev/sdb1   new_vg  lvm2 a-   17.14G 17.14G
/dev/sdc1   new_vg  lvm2 a-   17.14G 17.09G
/dev/sdd1   new_vg  lvm2 a-   17.14G 17.13G
```

Utilice el argumento **-v** del comando **pvs** para añadir los siguientes campos a la salida predeterminada: **dev_size**, **pv_uuid**.

```
# pvs -v
Scanning for physical volume names
PV          VG          Fmt Attr PSize  PFree  DevSize PV UUID
/dev/sdb1   new_vg      lvm2 a-   17.14G 17.14G  17.14G onFF2w-1fLC-ughJ-D9eB-M7iv-
6XqA-dqGeXY
/dev/sdc1   new_vg      lvm2 a-   17.14G 17.09G  17.14G Joqlch-yWSj-kuEn-IdwM-01S9-
X08M-mcpsVe
/dev/sdd1   new_vg      lvm2 a-   17.14G 17.13G  17.14G yvfvZK-Cf31-j75k-dECm-0RZ3-
0dGW-tUqkCS
```

Puede utilizar el argumento **--segments** del comando **pvs** para mostrar información sobre cada segmento de volumen físico. Un segmento es un grupo de extensiones. Un panorama de un segmento puede ser útil si desea ver si el volumen lógico está fragmentado

El comando **pvs --segments** muestra los siguientes campos de forma predeterminada: **pv_name**, **vg_name**, **pv_fmt**, **pv_attr**, **pv_size**, **pv_free**, **pvseg_start**, **pvseg_size**. La salida está ordenada de acuerdo con el campo **pv_name** y **pvseg_size** dentro del volumen físico.

```
# pvs --segments
PV          VG          Fmt Attr PSize  PFree  Start SSize
/dev/hda2   VolGroup00 lvm2 a-   37.16G 32.00M    0  1172
/dev/hda2   VolGroup00 lvm2 a-   37.16G 32.00M  1172   16
/dev/hda2   VolGroup00 lvm2 a-   37.16G 32.00M  1188    1
/dev/sda1   vg          lvm2 a-   17.14G 16.75G    0   26
/dev/sda1   vg          lvm2 a-   17.14G 16.75G   26   24
/dev/sda1   vg          lvm2 a-   17.14G 16.75G   50   26
/dev/sda1   vg          lvm2 a-   17.14G 16.75G   76   24
/dev/sda1   vg          lvm2 a-   17.14G 16.75G  100   26
/dev/sda1   vg          lvm2 a-   17.14G 16.75G  126   24
/dev/sda1   vg          lvm2 a-   17.14G 16.75G  150   22
/dev/sda1   vg          lvm2 a-   17.14G 16.75G  172  4217
/dev/sdb1   vg          lvm2 a-   17.14G 17.14G    0  4389
/dev/sdc1   vg          lvm2 a-   17.14G 17.14G    0  4389
/dev/sdd1   vg          lvm2 a-   17.14G 17.14G    0  4389
/dev/sde1   vg          lvm2 a-   17.14G 17.14G    0  4389
/dev/sdf1   vg          lvm2 a-   17.14G 17.14G    0  4389
/dev/sdg1   vg          lvm2 a-   17.14G 17.14G    0  4389
```

Puede utilizar el comando **pvs -a** para ver los dispositivos detectados por LVM que no han sido inicializados como volúmenes físicos LVM.

```
# pvs -a
PV                               VG      Fmt  Attr PSize  PFree
/dev/VolGroup00/LogVol01        --      --   0     0
/dev/new_vg/lvol0                --      --   0     0
/dev/ram                         --      --   0     0
/dev/ram0                        --      --   0     0
/dev/ram2                        --      --   0     0
/dev/ram3                        --      --   0     0
/dev/ram4                        --      --   0     0
/dev/ram5                        --      --   0     0
/dev/ram6                        --      --   0     0
/dev/root                        --      --   0     0
/dev/sda                         --      --   0     0
/dev/sdb                         --      --   0     0
/dev/sdb1                       new_vg  lvm2 a-   17.14G 17.14G
/dev/sdc                         --      --   0     0
/dev/sdc1                       new_vg  lvm2 a-   17.14G 17.09G
/dev/sdd                         --      --   0     0
/dev/sdd1                       new_vg  lvm2 a-   17.14G 17.14G
```

El comando vgs

[Tabla 4.2, “Campos de visualización de vgs”](#) lista las opciones de visualización del comando **vgs**, junto con el nombre del campo como aparece en el encabezado y con una descripción del campo.

Tabla 4.2. Campos de visualización de vgs

Argumento	Encabezado	Descripción
lv_count	#LV	El número de volúmenes lógicos que el grupo de volúmenes contiene
max_lv	MaxLV	El número máximo de volúmenes lógicos permitidos en el grupo de volúmenes (0 si no hay ningún límite)
max_pv	MaxPV	El número máximo de volúmenes físicos permitidos en el grupo de volúmenes (0 si no hay ningún límite)
pv_count	#PV	El número de volúmenes físicos que define el grupo de volúmenes
snap_count	#SN	El número de instantáneas que el grupo de volúmenes contiene
vg_attr	Attr	Estado del grupo de volúmenes: escribible (w), solo lectura (r), dimensionable (z), exportado (x), parcial (p) y en cluster (c).
vg_extent_count	#Ext	El número de extensiones físicas en el grupo de volúmenes
vg_extent_size	Ext	El tamaño de extensiones físicas en el grupo de volúmenes
vg_fmt	Fmt	El formato de metadatos del grupo de volúmenes (lvm2 o lvm1)
vg_free	VFree	Tamaño de espacio libre restante en el grupo de volúmenes
vg_free_count	Free	Número de extensiones físicas libres en el grupo de volúmenes
vg_name	VG	El nombre del grupo de volúmenes
vg_seqno	Seq	Número que representa la revisión del grupo de volúmenes
vg_size	VSize	El tamaño del grupo de volúmenes
vg_sysid	SYS ID	ID del sistema de LVM1
vg_tags	VG Tags	Etiquetas LVM añadidas al grupo de volúmenes
vg_uuid	VG UUID	El UUID del grupo de volúmenes

El comando **vgs** muestra los siguientes campos de forma predeterminada: **vg_name**, **pv_count**, **lv_count**, **snap_count**, **vg_attr**, **vg_size**, **vg_free**. La salida se ordena de acuerdo con el campo **vg_name**.

```
# vgs
VG      #PV #LV #SN Attr   VSize  VFree
new_vg  3   1   1 wz--n- 51.42G 51.36G
```

Utilice el argumento **-v** del comando **vgs** para añadir los siguientes campos a la salida predeterminada: **vg_extent_size**, **vg_uuid**.

```
# vgs -v
  Finding all volume groups
  Finding volume group "new_vg"
VG      Attr   Ext   #PV #LV #SN VSize  VFree  VG UUID
new_vg  wz--n- 4.00M  3   1   1 51.42G 51.36G jxQJ0a-ZKk0-0pM0-0118-nlw0-wwqd-
fD5D32
```

El comando lvs

[Tabla 4.3, “Campos de visualización de lvs”](#) lista las opciones de visualización del comando **lvs** junto con el nombre del campo como aparece en la salida y una descripción del campo.

Tabla 4.3. Campos de visualización de lvs

Argumento	Encabezado	Descripción
chunksize	Chunk	Tamaño de unidad en el volumen de instantánea
chunk_size		
copy_percent	Copy%	El porcentaje de sincronización de un volumen lógico de espejo; también usado cuando las extensiones físicas son desplazadas con el comando pv_move
devices	Devices	El dispositivo subyacente que conforma el volumen lógico: los volúmenes físicos, los volúmenes lógicos y las extensiones lógicas y extensiones físicas de inicio.
lv_attr	Attr	<p>El estado del volumen lógico. Los bits de atributos del volumen lógico son:</p> <p>Bit 1: tipo de volumen: espejo (m), espejo sin sincronizar (M), original (o), pvmove (p), instantánea (s), instantánea no válida (S), virtual (v)</p> <p>Bit 2: Permisos: escritura (w), solo lectu(r)a</p> <p>Bit 3: Política de asignación: contigua (c), normal (n), cualquier parte (a), heredada (i). Esto se lleva a cabo si el volumen está actualmente bloqueado contra cambios de asignación, por ejemplo cuando se está ejecutando el comando pvmove.</p> <p>Bit 4: espejo corregido (m)</p> <p>Bit 5 Estado: (a)ctivo, (s)uspendido, instantánea (l)nválida, instantánea (S)uspendida inválida, (d)ispositivo asignado presente sin tablas, dispositivo asignado presente con tabla (i)nactiva</p> <p>Bit 6: dispositivo abierto (o)</p>
lv_kernel_major	KMaj	Número mayor del dispositivo del volumen lógico (-1 si está inactivo)
lv_kernel_minor	KMIN	Número menor del dispositivo del volumen lógico (-1 si está inactivo)
lv_major	Maj	El número de dispositivo mayor persistente del volumen lógico (-1 si no está especificado)
lv_minor	Min	El número de dispositivo menor persistente del volumen lógico (-1 si no está especificado)
lv_name	LV	El nombre del volumen lógico
lv_size	LSize	El tamaño del volumen lógico
lv_tags	LV Tags	Etiquetas LVM añadidas en el volumen lógico
lv_uuid	LV UUID	El UUID del volumen lógico.
mirror_log	Log	Dispositivo donde reside el registro del espejo

modules	Modules	Objetivos de mapa de dispositivos del kernel correspondiente necesarios para utilizar este volumen lógico
move_pv	Move	Volumen físico fuente de un volumen lógico temporal creado con el comando pvmove
origin	Origin	El dispositivo original de un volumen de instantánea
regionsize	Region	El tamaño de la unidad de un volumen lógico con espejo
region_size		
seg_count	#Seg	El número de segmentos en el volumen lógico
seg_size	SSize	El tamaño de los segmentos en el volumen lógico
seg_start	Start	Balance del segmento en el volumen lógico
seg_tags	Seg Tags	Etiquetas LVM añadidas a los segmentos del volumen lógico
segtype	Type	El tipo de segmento de un volumen lógico (por ejemplo: espejo, entrelazado, lineal)
snap_percent	Snap%	Porcentaje actual de un volumen de instantánea que está siendo usado
stripes	#Str	Número de instantáneas o espejos en un volumen lógico
stripesize	Stripe	Tamaño de unidades del enlace en un volumen lógico entrelazado
stripe_size		

El comando **lvs** muestra los siguientes campos de forma predeterminada: **lv_name**, **vg_name**, **lv_attr**, **lv_size**, **origin**, **snap_percent**, **move_pv**, **mirror_log**, **copy_percent**. La salida es ordenada según los campos **vg_name** y **lv_name** dentro del grupo de volúmenes.

```
# lvs
LV          VG      Attr   LSize  Origin Snap%  Move Log Copy%
lvol0      new_vg  owi-a- 52.00M
newvgsnap1 new_vg  swi-a-  8.00M lvol0   0.20
```

Al utilizar el argumento **-v** con el comando **lvs** se agregan los siguientes campos a la pantalla predeterminada: **seg_count**, **lv_major**, **lv_minor**, **lv_kernel_major**, **lv_kernel_minor**, **lv_uuid**.

```
# lvs -v
Finding all logical volumes
LV          VG      #Seg Attr   LSize  Maj  Min  KMaj  KMin  Origin Snap%  Move Copy%
Log LV UUID
lvol0      new_vg   1 owi-a- 52.00M  -1  -1  253   3
LBy1Tz-sr23-0jsI-LT03-nHLC-y8XW-EhCl78
newvgsnap1 new_vg   1 swi-a-  8.00M  -1  -1  253   5   lvol0   0.20
1ye10U-1cIu-o79k-20h2-ZGF0-qCJm-CfbsIx
```

Se puede utilizar el argumento **--segments** del comando **lvs** para mostrar las columnas predeterminadas que enfatizan la información del segmento. Al utilizar el argumento **segments**, el prefijo **seg** es opcional. El comando **lvs --segments** muestra los siguientes campos de forma predeterminada: **lv_name**, **vg_name**, **lv_attr**, **stripes**, **segtype**, **seg_size**. La salida

predeterminada se ordena según el campo **vg_name**, **lv_name** dentro del grupo de volúmenes y **seg_start** dentro del volumen lógico. Si el volumen lógico está fragmentado, la salida del comando lo mostrará.

```
# lvs --segments
LV      VG          Attr      #Str Type   SSize
LogVol00 VolGroup00 -wi-ao    1 linear 36.62G
LogVol01 VolGroup00 -wi-ao    1 linear 512.00M
lv      vg          -wi-a-    1 linear 104.00M
lv      vg          -wi-a-    1 linear 104.00M
lv      vg          -wi-a-    1 linear 104.00M
lv      vg          -wi-a-    1 linear 88.00M
```

Al utilizar el argumento **-v** con el comando **lvs --segments** se añaden los siguientes campos a la salida predeterminada: **seg_start**, **stripesize**, **chunksize**.

```
# lvs -v --segments
Finding all logical volumes
LV      VG          Attr      Start SSize #Str Type   Stripe Chunk
lv010   new_vg     owi-a-    0     52.00M  1 linear 0       0
newvgsnap1 new_vg    swi-a-    0     8.00M   1 linear 0       8.00K
```

El siguiente ejemplo muestra la salida predeterminada del comando **lvs** en un sistema con un volumen lógico configurado, seguido de la salida predeterminada del comando **lvs** con el argumento **segments** especificado..

```
# lvs
LV      VG          Attr      LSize  Origin Snap%  Move Log Copy%
lv010   new_vg     -wi-a-    52.00M
# lvs --segments
LV      VG          Attr      #Str Type   SSize
lv010   new_vg     -wi-a-    1 linear 52.00M
```

4.8.3. Cómo ordenar los reportes de LVM

Normalmente toda la salida de los comandos **lvs**, **vgs** o **pvs** tiene que ser generada y almacenada internamente antes de que pueda ser ordenada en columnas de forma correcta. Puede especificar el argumento **--unbuffered** para mostrar la salida sin ordenar tal y como es generada.

Para especificar una lista alternativa de columnas para ser ordenada, utilice el argumento **-O** de cualquiera de los comandos de reporte. No es necesario incluir estos campos dentro de la salida misma.

El siguiente ejemplo muestra la salida de **pvs** con el nombre, el tamaño y el espacio libre del volumen físico.

```
# pvs -o pv_name,pv_size,pv_free
PV      PSize  PFree
/dev/sdb1 17.14G 17.14G
/dev/sdc1 17.14G 17.09G
/dev/sdd1 17.14G 17.14G
```

El siguiente ejemplo muestra la misma salida ordenada según el campo de espacio libre.

```
# pvs -o pv_name,pv_size,pv_free -o pv_free
PV          PSize  PFree
/dev/sdc1   17.14G 17.09G
/dev/sdd1   17.14G 17.14G
/dev/sdb1   17.14G 17.14G
```

El siguiente ejemplo muestra que no se necesita mostrar el campo que está siendo usado para ordenar.

```
# pvs -o pv_name,pv_size -o pv_free
PV          PSize
/dev/sdc1   17.14G
/dev/sdd1   17.14G
/dev/sdb1   17.14G
```

Para mostrar un orden inverso, utilice el argumento **-o** y preceda el campo especificado con el carácter **-**.

```
# pvs -o pv_name,pv_size,pv_free -o -pv_free
PV          PSize  PFree
/dev/sdd1   17.14G 17.14G
/dev/sdb1   17.14G 17.14G
/dev/sdc1   17.14G 17.09G
```

4.8.4. Cómo especificar unidades

Para especificar la unidad para la visualización del reporte LVM, utilice el argumento **--units** del comando de reporte. Puede especificar (b)ytes, (k)ilobytes, (m)egabytes, (g)igabytes, (t)erabytes, (e)xabytes, (p)etabytes y lectura (h)umana. Este último es el valor predeterminado. Puede modificar el valor predeterminado si establece el parámetro **units** en la sección en la sección **global** del archivo **lvm.conf**.

El siguiente ejemplo especifica que la salida del comando **pvs** sea en MB y no en GB.

```
# pvs --units m
PV          VG          Fmt  Attr  PSize      PFree
/dev/sda1   VG          lvm2  --    17555.40M 17555.40M
/dev/sdb1   new_vg      lvm2  a-    17552.00M 17552.00M
/dev/sdc1   new_vg      lvm2  a-    17552.00M 17500.00M
/dev/sdd1   new_vg      lvm2  a-    17552.00M 17552.00M
```

Por defecto, las unidades se muestran como potencias de 2 (múltiplos de 1024). Puede especificar que las unidades se muestren en múltiplos de 1000 utilizando mayúsculas en la unidad especificada (B, K, M, G, T, H).

El siguiente comando muestra la salida con un múltiplo de 1024, la conducta predeterminada.

```
# pvs
PV          VG          Fmt  Attr  PSize  PFree
/dev/sdb1   new_vg      lvm2  a-    17.14G 17.14G
/dev/sdc1   new_vg      lvm2  a-    17.14G 17.09G
/dev/sdd1   new_vg      lvm2  a-    17.14G 17.14G
```

El siguiente comando muestra la salida como un múltiplo de 1000.

```
# pvs --units G
PV          VG          Fmt  Attr  PSize  PFree
/dev/sdb1   new_vg      lvm2 a-    18.40G 18.40G
/dev/sdc1   new_vg      lvm2 a-    18.40G 18.35G
/dev/sdd1   new_vg      lvm2 a-    18.40G 18.40G
```

Puede especificar los sectores (definidos como 512 bytes) o unidades personalizadas.

El siguiente ejemplo muestra la salida del comando **pvs** como un número de sectores.

```
# pvs --units s
PV          VG          Fmt  Attr  PSize      PFree
/dev/sdb1   new_vg      lvm2 a-    35946496S 35946496S
/dev/sdc1   new_vg      lvm2 a-    35946496S 35840000S
/dev/sdd1   new_vg      lvm2 a-    35946496S 35946496S
```

El siguiente ejemplo muestra la salida del comando **pvs** en unidades de 4 MB.

```
# pvs --units 4m
PV          VG          Fmt  Attr  PSize      PFree
/dev/sdb1   new_vg      lvm2 a-    4388.00U 4388.00U
/dev/sdc1   new_vg      lvm2 a-    4388.00U 4375.00U
/dev/sdd1   new_vg      lvm2 a-    4388.00U 4388.00U
```

Capítulo 5. Ejemplos de configuración de LVM

Este capítulo proporciona algunos ejemplos de configuración básica de LVM.

5.1. Cómo crear un volumen lógico LVM en tres discos

Este ejemplo crea un volumen lógico LVM llamado `new_logical_volume` que consta de discos en `/dev/sda1`, `/dev/sdb1`, y `/dev/sdc1`.

5.1.1. Cómo crear los volúmenes físicos

Se deben etiquetar los discos como volúmenes físicos LVM para poder usarlos en un grupo de volúmenes.



Advertencia

Este comando destruye todos los datos en `/dev/sda1`, `/dev/sdb1` y `/dev/sdc1`.

```
[root@tng3-1 ~]# pvcreate /dev/sda1 /dev/sdb1 /dev/sdc1
Physical volume "/dev/sda1" successfully created
Physical volume "/dev/sdb1" successfully created
Physical volume "/dev/sdc1" successfully created
```

5.1.2. Cómo crear el grupo de volúmenes

El siguiente comando crea el grupo de volúmenes `new_vol_group`.

```
[root@tng3-1 ~]# vgcreate new_vol_group /dev/sda1 /dev/sdb1 /dev/sdc1
Volume group "new_vol_group" successfully created
```

Puede utilizar el comando `vgs` para mostrar los atributos del nuevo grupo de volúmenes.

```
[root@tng3-1 ~]# vgs
VG                #PV #LV #SN Attr   VSize  VFree
new_vol_group     3   0   0 wz--n- 51.45G 51.45G
```

5.1.3. Cómo crear el volumen lógico

El siguiente comando crea el volumen lógico `new_logical_volume` desde el grupo de volúmenes `new_vol_group`. Este ejemplo crea un volumen lógico que utiliza 2GB del grupo de volúmenes.

```
[root@tng3-1 ~]# lvcreate -L2G -n new_logical_volume new_vol_group
Logical volume "new_logical_volume" created
```

5.1.4. Cómo crear un sistema de archivos

El siguiente comando crea un sistema de archivos GFS2 en el volumen lógico.


```
[root@tng3-1 ~]# mkfs.gfs2 -plock_nolock -j 1
/dev/new_vol_group/new_logical_volume
This will destroy any data on /dev/new_vol_group/new_logical_volume.

Are you sure you want to proceed? [y/n] y

Device:                /dev/new_vol_group/new_logical_volume
Blocksize:             4096
Filesystem Size:      491460
Journals:              1
Resource Groups:      8
Locking Protocol:     lock_nolock
Lock Table:

Syncing...
All Done
```

Los siguientes comandos montan el volumen lógico y reportan el uso de espacio de disco en el sistema de archivos.

```
[root@tng3-1 ~]# mount /dev/new_vol_group/new_logical_volume /mnt
[root@tng3-1 ~]# df
Filesystem                1K-blocks      Used Available Use% Mounted on
/dev/new_vol_group/new_logical_volume
                          1965840         20   1965820   1% /mnt
```

5.2. Cómo crear un volumen lógico entrelazado

Este ejemplo crea un volumen lógico entrelazado LVM llamado **striped_logical_volume** que entrelaza datos a través de los discos **/dev/sda1**, **/dev/sdb1** y **/dev/sdc1**.

5.2.1. Cómo crear los volúmenes físicos

Etiquete los discos a usar en el grupo de volúmenes como volúmenes físicos LVM.



Advertencia

Este comando destruye todos los datos en **/dev/sda1**, **/dev/sdb1** y **/dev/sdc1**.

```
[root@tng3-1 ~]# pvcreate /dev/sda1 /dev/sdb1 /dev/sdc1
Physical volume "/dev/sda1" successfully created
Physical volume "/dev/sdb1" successfully created
Physical volume "/dev/sdc1" successfully created
```

5.2.2. Cómo crear el grupo de volúmenes

El siguiente comando crea el grupo de volúmenes **volgroup01**.

```
[root@tng3-1 ~]# vgcreate volgroup01 /dev/sda1 /dev/sdb1 /dev/sdc1
Volume group "volgroup01" successfully created
```

Puede utilizar el comando **vgs** para mostrar los atributos del nuevo grupo de volúmenes.

```
[root@tng3-1 ~]# vgs
VG                #PV #LV #SN Attr   VSize  VFree
volgroup01        3   0   0 wz--n- 51.45G 51.45G
```

5.2.3. Cómo crear el volumen lógico

El siguiente comando crea el volumen lógico seccionado **striped_logical_volume** desde el grupo de volúmenes **striped_logical_volume**. Este ejemplo crea un volumen lógico de 2 GB, con tres bandas y un tamaño de banda de 4 KB.

```
[root@tng3-1 ~]# lvcreate -i3 -l4 -L2G -nstriped_logical_volume volgroup01
Rounding size (512 extents) up to stripe boundary size (513 extents)
Logical volume "striped_logical_volume" created
```

5.2.4. Cómo crear un sistema de archivos

El siguiente comando crea un sistema de archivos GFS2 en el volumen lógico.

```
[root@tng3-1 ~]# mkfs.gfs2 -plock_nolock -j 1
/dev/volgroup01/striped_logical_volume
This will destroy any data on /dev/volgroup01/striped_logical_volume.

Are you sure you want to proceed? [y/n] y

Device:                /dev/volgroup01/striped_logical_volume
Blocksize:             4096
Filesystem Size:      492484
Journals:              1
Resource Groups:      8
Locking Protocol:     lock_nolock
Lock Table:

Syncing...
All Done
```

Los siguientes comandos montan el volumen lógico y reportan el uso de espacio de disco en el sistema de archivos.

```
[root@tng3-1 ~]# mount /dev/volgroup01/striped_logical_volume /mnt
[root@tng3-1 ~]# df
Filesystem                1K-blocks      Used Available Use% Mounted on
/dev/mapper/VolGroup00-LogVol00
                          13902624    1656776   11528232   13% /
/dev/hda1                  101086      10787     85080    12% /boot
tmpfs                      127880        0      127880    0% /dev/shm
/dev/volgroup01/striped_logical_volume
                          1969936        20    1969916    1% /mnt
```

5.3. Separación de un grupo de volúmenes

En este ejemplo, hay un grupo de volúmenes que consta de tres volúmenes físicos. Si hay suficiente espacio sin usar en los volúmenes físicos, un nuevo grupo de volúmenes puede ser creado sin tener que añadir nuevos discos.

En la configuración inicial, el volumen lógico **mylv** se crea desde el grupo de volúmenes **myvol**, el cual

consta de tres volúmenes físicos **/dev/sda1**, **/dev/sdb1** y **/dev/sdc1**.

Una vez completado este procedimiento, el grupo de volúmenes **myvg** consistirá de **/dev/sda1** y **/dev/sdb1**. Un segundo grupo de volúmenes, **tuvg** consistirá de **/dev/sdc1**.

5.3.1. Cómo determinar el espacio libre

Puede utilizar el comando **pvscan** para determinar cuánto espacio libre está disponible en el grupo de volúmenes.

```
[root@tng3-1 ~]# pvscan
PV /dev/sda1  VG myvg   lvm2 [17.15 GB / 0    free]
PV /dev/sdb1  VG myvg   lvm2 [17.15 GB / 12.15 GB free]
PV /dev/sdc1  VG myvg   lvm2 [17.15 GB / 15.80 GB free]
Total: 3 [51.45 GB] / in use: 3 [51.45 GB] / in no VG: 0 [0  ]
```

5.3.2. Cómo desplazar los datos

Puede desplazar todas las extensiones usadas en **/dev/sdc1** a **/dev/sdb1** con **pvmove**. **pvmove** puede durar un tiempo prolongado.

```
[root@tng3-1 ~]# pvmove /dev/sdc1 /dev/sdb1
/dev/sdc1: Moved: 14.7%
/dev/sdc1: Moved: 30.3%
/dev/sdc1: Moved: 45.7%
/dev/sdc1: Moved: 61.0%
/dev/sdc1: Moved: 76.6%
/dev/sdc1: Moved: 92.2%
/dev/sdc1: Moved: 100.0%
```

Después de desplazar los datos, puede ver que todos el espacio en **/dev/sdc1** está vacío.

```
[root@tng3-1 ~]# pvscan
PV /dev/sda1  VG myvg   lvm2 [17.15 GB / 0    free]
PV /dev/sdb1  VG myvg   lvm2 [17.15 GB / 10.80 GB free]
PV /dev/sdc1  VG myvg   lvm2 [17.15 GB / 17.15 GB free]
Total: 3 [51.45 GB] / in use: 3 [51.45 GB] / in no VG: 0 [0  ]
```

5.3.3. Cómo dividir en grupo de volúmenes

Para crear el nuevo grupo de volúmenes **yourvg**, utilice el comando **vgsplit** para separar el grupo de volúmenes **myvg**.

Antes de que pueda dividir el grupo de volúmenes, el volumen lógico debe ser desactivado. Si el sistema de archivos está montado, se debe desmontar el sistema de archivos antes de desactivar el volumen lógico.

Puede desactivar el volumen lógico con el comando **lvchange** o el comando **vgchange**. El siguiente comando desactiva el volumen lógico **mylv** y luego divide el grupo de volúmenes **yourvg** del grupo de volúmenes **myvg**, trasladando el volumen físico **/dev/sdc1** al nuevo grupo de volúmenes **yourvg**.

```
[root@tng3-1 ~]# lvchange -a n /dev/myvg/mylv
[root@tng3-1 ~]# vgsplit myvg yourvg /dev/sdc1
Volume group "yourvg" successfully split from "myvg"
```

Puede utilizar el comando **vgs** para ver los atributos de los dos grupos de volúmenes.

```
[root@tng3-1 ~]# vgs
VG      #PV #LV #SN Attr   VSize  VFree
myvg    2   1   0 wz--n- 34.30G 10.80G
yourvg  1   0   0 wz--n- 17.15G 17.15G
```

5.3.4. Cómo crear un nuevo volumen lógico

Después de crear un nuevo grupo de volúmenes, puede crear el nuevo volumen lógico **yourlv**.

```
[root@tng3-1 ~]# lvcreate -L5G -n yourlv yourvg
Logical volume "yourlv" created
```

5.3.5. Cómo crear el sistema de archivos y cómo montar el nuevo volumen lógico

Puede crear un sistema de archivos en el nuevo volumen lógico y montar este sistema de archivos.

```
[root@tng3-1 ~]# mkfs.gfs2 -plock_nolock -j 1 /dev/yourvg/yourlv
This will destroy any data on /dev/yourvg/yourlv.

Are you sure you want to proceed? [y/n] y

Device:                /dev/yourvg/yourlv
Blocksize:              4096
Filesystem Size:       1277816
Journals:               1
Resource Groups:       20
Locking Protocol:      lock_nolock
Lock Table:

Syncing...
All Done

[root@tng3-1 ~]# mount /dev/yourvg/yourlv /mnt
```

5.3.6. Cómo activar y montar el volumen lógico original

Como anteriormente **mylv** tuvo que ser desactivado, deberá reactivarse antes de poder ser montado.

```
root@tng3-1 ~]# lvchange -a y mylv

[root@tng3-1 ~]# mount /dev/myvg/mylv /mnt
[root@tng3-1 ~]# df
Filesystem                1K-blocks      Used Available Use% Mounted on
/dev/yourvg/yourlv         24507776         32 24507744   1% /mnt
/dev/myvg/mylv             24507776         32 24507744   1% /mnt
```

5.4. Cómo retirar un disco de un volumen lógico

Este ejemplo muestra cómo retirar un disco desde un volumen lógico existente, ya sea para reemplazar el disco o para usar el disco como parte de un volumen diferente. Para retirar un disco, primero se deben desplazar las extensiones en el volumen físico LVM a un disco o grupo de discos diferentes.

5.4.1. Cómo desplazar las extensiones a volúmenes físicos existentes

En este ejemplo, el volumen lógico es distribuido a través de cuatro volúmenes físicos en el grupo de

volúmenes **myvg**.

```
[root@tng3-1]# pvs -o+pv_used
PV          VG   Fmt Attr PSize  PFree  Used
/dev/sda1   myvg lvm2 a-   17.15G 12.15G  5.00G
/dev/sdb1   myvg lvm2 a-   17.15G 12.15G  5.00G
/dev/sdc1   myvg lvm2 a-   17.15G 12.15G  5.00G
/dev/sdd1   myvg lvm2 a-   17.15G  2.15G 15.00G
```

Queremos desplazar las extensiones de **/dev/sdb1** para poder quitarlas del grupo de volúmenes.

Si hay suficientes extensiones libres en otro volumen físico en el grupo de volúmenes, puede ejecutar el comando **pvmove** sin ninguna opción en el dispositivo desde el cual desea realizar la remoción y las extensiones serán distribuidas a otros dispositivos.

```
[root@tng3-1 ~]# pvmove /dev/sdb1
/dev/sdb1: Moved: 2.0%
...
/dev/sdb1: Moved: 79.2%
...
/dev/sdb1: Moved: 100.0%
```

Después de la ejecución de **pvmove**, la distribución de extensiones es la siguiente:

```
[root@tng3-1]# pvs -o+pv_used
PV          VG   Fmt Attr PSize  PFree  Used
/dev/sda1   myvg lvm2 a-   17.15G  7.15G 10.00G
/dev/sdb1   myvg lvm2 a-   17.15G 17.15G   0
/dev/sdc1   myvg lvm2 a-   17.15G 12.15G  5.00G
/dev/sdd1   myvg lvm2 a-   17.15G  2.15G 15.00G
```

Utilice el comando **vgreduce** para retirar el volumen físico **/dev/sdb1** del grupo de volúmenes.

```
[root@tng3-1 ~]# vgreduce myvg /dev/sdb1
Removed "/dev/sdb1" from volume group "myvg"
[root@tng3-1 ~]# pvs
PV          VG   Fmt Attr PSize  PFree
/dev/sda1   myvg lvm2 a-   17.15G  7.15G
/dev/sdb1           lvm2 --   17.15G 17.15G
/dev/sdc1   myvg lvm2 a-   17.15G 12.15G
/dev/sdd1   myvg lvm2 a-   17.15G  2.15G
```

El disco puede ahora ser removido físicamente o asignado a otros usuarios.

5.4.2. Cómo desplazar extensiones a un nuevo disco

En este ejemplo, el volumen lógico se distribuye a lo largo de tres volúmenes físicos en el grupo de volúmenes **myvg** así:

```
[root@tng3-1]# pvs -o+pv_used
PV          VG   Fmt Attr PSize  PFree  Used
/dev/sda1   myvg lvm2 a-   17.15G  7.15G 10.00G
/dev/sdb1   myvg lvm2 a-   17.15G 15.15G  2.00G
/dev/sdc1   myvg lvm2 a-   17.15G 15.15G  2.00G
```

Queremos desplazar las extensiones de **/dev/sdb1** a un nuevo dispositivo: **/dev/sdd1**.

5.4.2.1. Cómo crear un nuevo volumen físico

Cree un nuevo volumen físico en **/dev/sdd1**.

```
[root@tng3-1 ~]# pvcreate /dev/sdd1
Physical volume "/dev/sdd1" successfully created
```

5.4.2.2. Cómo añadir el nuevo volumen físico al grupo de volúmenes

Añada **/dev/sdd1** al grupo de volúmenes existente **myvg**.

```
[root@tng3-1 ~]# vgextend myvg /dev/sdd1
Volume group "myvg" successfully extended
[root@tng3-1]# pvs -o+pv_used
PV          VG   Fmt Attr PSize  PFree  Used
/dev/sda1   myvg lvm2 a-   17.15G  7.15G 10.00G
/dev/sdb1   myvg lvm2 a-   17.15G 15.15G  2.00G
/dev/sdc1   myvg lvm2 a-   17.15G 15.15G  2.00G
/dev/sdd1   myvg lvm2 a-   17.15G 17.15G   0
```

5.4.2.3. Cómo desplazar los datos

Utilice **pvmove** para desplazar los datos de **/dev/sdb1** a **/dev/sdd1**.

```
[root@tng3-1 ~]# pvmove /dev/sdb1 /dev/sdd1
/dev/sdb1: Moved: 10.0%
...
/dev/sdb1: Moved: 79.7%
...
/dev/sdb1: Moved: 100.0%

[root@tng3-1]# pvs -o+pv_used
PV          VG   Fmt Attr PSize  PFree  Used
/dev/sda1   myvg lvm2 a-   17.15G  7.15G 10.00G
/dev/sdb1   myvg lvm2 a-   17.15G 17.15G   0
/dev/sdc1   myvg lvm2 a-   17.15G 15.15G  2.00G
/dev/sdd1   myvg lvm2 a-   17.15G 15.15G  2.00G
```

5.4.2.4. Cómo retirar el volumen físico antiguo del grupo de volúmenes

Después de mover los datos de **/dev/sdb1**, puede retirarlo del grupo de volúmenes.

```
[root@tng3-1 ~]# vgreduce myvg /dev/sdb1
Removed "/dev/sdb1" from volume group "myvg"
```

Puede asignar el disco a otro grupo de volúmenes o retirar el disco del sistema.

5.5. Cómo crear un volumen lógico LVM en un cluster

La creación de un volumen lógico en espejo en un cluster requiere los mismos comandos y procedimientos que para crear un volumen lógico en espejo en un solo nodo. Sin embargo, para crear un volumen LVM en espejo en un cluster, el cluster y la infraestructura de espejo de cluster deben estar en ejecución, el cluster debe estar en cuorums, y el tipo de bloqueo en el archivo **lvm.conf** debe ser configurado correctamente para habilitar bloqueo de cluster, ya sea directamente o por medio del comando **lvmconf** como se describe en [Sección 3.1, "Creación de volúmenes LVM en un cluster"](#).

El siguiente procedimiento crea un volumen LVM en espejo en un cluster. Primero, el procedimiento chequea si los servicios de cluster están instalados y en ejecución, luego el procedimiento crea el volumen en espejo.

1. Para crear un volumen lógico en espejo que sea compartido por todos los nodos en un cluster, el tipo de bloqueo debe establecerse correctamente en el archivo `lvm.conf` en cada nodo del cluster. Por defecto, el tipo de bloqueo se establece a `local`. Para cambiarlo, ejecute el siguiente comando en cada nodo del cluster para activar el bloqueo en agrupamiento.

```
# /sbin/lvmconf --enable-cluster
```

2. Para crear un volumen lógico en cluster, la infraestructura de cluster debe estar activa y ejecutándose en cada nodo en el cluster. El siguiente ejemplo verifica si el demonio `clvmd` está ejecutándose en el nodo desde el cual fue generado:

```
[root@doc-07 ~]# ps auxw | grep clvmd
root      17642  0.0  0.1 32164 1072 ?        Ssl  Apr06   0:00 clvmd -T20 -t 90
```

El siguiente comando muestra la vista local del estatus de cluster:

```
[root@example-01 ~]# cman_tool services
fence domain
member count  3
victim count  0
victim now    0
master nodeid 2
wait state    none
members       1 2 3

dml lockspaces
name          clvmd
id            0x4104eefa
flags         0x00000000
change        member 3 joined 1 remove 0 failed 0 seq 1,1
members       1 2 3
```

3. Verifique si el paquete `cmirror` está instalado.
4. Inicie el servicio `cmirror`.

```
[root@hexample-01 ~]# service cmirror start
Starting cmirror: [ OK ]
```

5. Crear el espejo. El primer paso es crea los volúmenes físicos. Los siguientes comandos crean tres volúmenes físicos. Dos de los volúmenes físicos serán utilizados pra los pilares del espejo y el tercer volumen físico contendrá el registro de espejo.

```
[root@doc-07 ~]# pvcreate /dev/xvdb1
Physical volume "/dev/xvdb1" successfully created
[root@doc-07 ~]# pvcreate /dev/xvdb2
Physical volume "/dev/xvdb2" successfully created
[root@doc-07 ~]# pvcreate /dev/xvdc1
Physical volume "/dev/xvdc1" successfully created
```

6. Crear el grupo de volúmenes. Este ejemplo crea un grupo de volúmenes `vg001` que consta de tres volúmenes físicos que fueron creados en el paso anterior.

```
[root@doc-07 ~]# vgcreate vg001 /dev/xvdb1 /dev/xvdb2 /dev/xvdc1
Clustered volume group "vg001" successfully created
```

Observe que la salida del comando **vgcreate** indica que el grupo de volúmenes está en cluster. Puede verificar que un grupo de volúmenes está en cluster con el comando **vgs**, el cual mostrará los atributos de grupo de volúmenes. Si un grupo de volúmenes está en cluster, mostrará un atributo **c**.

```
[root@doc-07 ~]# vgs vg001
VG          #PV #LV #SN Attr   VSize  VFree
vg001      3   0   0 wz--nc 68.97G 68.97G
```

7. El siguiente comando crea el volumen lógico en espejo **mirrorlv** desde el grupo de volúmenes **vg001**. Este ejemplo especifica qué extensiones del volumen físico serán utilizados para el volumen lógico.

```
[root@doc-07 ~]# lvcreate -l 1000 -m1 vg001 -n mirrorlv /dev/xvdb1:1-1000
/dev/xvdb2:1-1000 /dev/xvdc1:0
Logical volume "mirrorlv" created
```

Puede utilizar el comando **lvs** para mostrar el progreso de la creación de espejo. El siguiente ejemplo, muestra que el espejo está 47% synced, luego 91% synced y después 100% synced cuando el espejo está completo.

```
[root@doc-07 log]# lvs vg001/mirrorlv
LV          VG          Attr      LSize Origin Snap%   Move Log              Copy%
Convert
mirrorlv   vg001      mwi-a-   3.91G                vg001_mlog          47.00
[root@doc-07 log]# lvs vg001/mirrorlv
LV          VG          Attr      LSize Origin Snap%   Move Log              Copy%
Convert
mirrorlv   vg001      mwi-a-   3.91G                vg001_mlog          91.00
[root@doc-07 ~]# lvs vg001/mirrorlv
LV          VG          Attr      LSize Origin Snap%   Move Log              Copy%
Convert
mirrorlv   vg001      mwi-a-   3.91G                vg001_mlog         100.00
```

La finalización del espejo se anota en el registro del sistema:

```
May 10 14:52:52 doc-07 [19402]: Monitoring mirror device vg001-mirrorlv for
events
May 10 14:55:00 doc-07 lvm[19402]: vg001-mirrorlv is now in-sync
```

8. Puede utilizar **lvs** con las opciones **-o +devices** para mostrar la configuración del espejo, incluyendo qué dispositivos componen los pilares de espejo. Puede ver que el volumen lógico en este ejemplo está compuesto por dos imágenes lineales y un registro.


```
[root@doc-07 ~]# lvs -a -o +devices
LV          VG          Attr   LSize  Origin Snap%  Move Log
Copy%  Convert Devices
mirrorlv    vg001      mwi-a- 3.91G
mirrorlv_mlog 100.00    mirrorlv_mimage_0(0),mirrorlv_mimage_1(0)
[mirrorlv_mimage_0] vg001      iwi-ao 3.91G
/dev/xvdb1(1)
[mirrorlv_mimage_1] vg001      iwi-ao 3.91G
/dev/xvdb2(1)
[mirrorlv_mlog]    vg001      lwi-ao 4.00M
/dev/xvdc1(0)
```

Puede utilizar la opción **seg_pe_ranges** del **lvs** para mostrar la distribución de datos. Puede usar esta opción para verificar si su diseño es correctamente redundante. La salida de este comando muestra registros de PE en el mismo formato en que los comandos **lvcreate** y **lvresize** toman como entrada.

```
[root@doc-07 ~]# lvs -a -o +seg_pe_ranges --segments
PE Ranges
mirrorlv_mimage_0:0-999 mirrorlv_mimage_1:0-999
/dev/xvdb1:1-1000
/dev/xvdb2:1-1000
/dev/xvdc1:0-0
```



Nota

Para obtener información sobre recuperación de uno de los pilares de un volumen LVM en espejo, consulte la [Sección 6.3, “Cómo recuperarse de una falla de un espejo LVM”](#).

Capítulo 6. Solución de errores en LVM

Este capítulo proporciona instrucciones sobre cómo solucionar diferentes problemas de LVM.

6.1. Diagnósticos en la solución de errores

Si un comando no está funcionando como se espera, puede obtener diagnósticos de las siguientes maneras:

- ▶ Utilice **-v**, **-vv**, **-vvv** o **-vvvv** en cualquier comando para incrementar la cantidad de mensajes de salida.
- ▶ Si el problema está relacionado con la activación de volúmenes lógicos, establezca 'activation = 1' en la sección 'log' del archivo de configuración y ejecute el comando con la opción **-vvvv**. Una vez haya finalizado la revisión de los mensajes de salida no olvide establecer este parámetro a 0, para evitar que la máquina se bloquee durante las situaciones de baja memoria.
- ▶ Ejecute el comando **lvm dump**, el cual proporciona un volcado de información para propósitos de diagnóstico. Para obtener mayor información, consulte la página de manual (8) de **lvm dump**
- ▶ Ejecute los comandos **lvs -v**, **pvs -a** o **dmsetup info -c** para obtener información adicional del sistema.
- ▶ Examine la última copia de seguridad de los metadatos en el archivo **/etc/lvm/backup** y las versiones archivadas en el archivo **/etc/lvm/archive**.
- ▶ Revise la información de configuración actual ejecutando el comando **lvm dumpconfig**.
- ▶ Revise el archivo **.cache** en el directorio **/etc/lvm** para obtener un registro de cuáles dispositivos tienen volúmenes físicos en ellos.

6.2. Cómo ver la información en dispositivos fallidos

Puede utilizar el argumento **-P** de **lvs** o **vgs** para visualizar la información sobre un volumen fallido que de otra forma no aparecería en los mensajes de salida. Este argumento permite algunas operaciones aunque los metadatos no sean completamente consistentes internamente. Por ejemplo, si uno de los dispositivos que conforman el grupo de volúmenes **vg** falla, el comando **vgs** mostraría la siguiente salida.

```
[root@link-07 tmp]# vgs -o +devices
Volume group "vg" not found
```

Si especifica la opción **-P** del comando **vgs**, el grupo de volúmenes no se puede usar pero puede ver más información sobre el dispositivo fallido.

```
[root@link-07 tmp]# vgs -P -o +devices
Partial mode. Incomplete volume groups will be activated read-only.
VG   #PV #LV #SN Attr   VSize VFree Devices
vg   9  2  0  rz-pn- 2.11T 2.07T unknown device(0)
vg   9  2  0  rz-pn- 2.11T 2.07T unknown device(5120),/dev/sda1(0)
```

En este ejemplo, el dispositivo fallido hace que tanto un volumen lógico lineal como uno entrelazado en el grupo de volúmenes falle. El comando **lvs** sin la opción **-P** muestra el siguiente mensaje de salida.

```
[root@link-07 tmp]# lvs -a -o +devices
Volume group "vg" not found
```

Al usar la opción **-P** se muestra el volumen lógico que ha fallado.

```
[root@link-07 tmp]# lvs -P -a -o +devices
Partial mode. Incomplete volume groups will be activated read-only.
LV      VG      Attr   LSize  Origin Snap%  Move Log Copy%  Devices
linear  vg      -wi-a- 20.00G                unknown device(0)
stripe  vg      -wi-a- 20.00G                unknown
device(5120),/dev/sda1(0)
```

Los siguientes ejemplos muestran la salida de los comandos **pvs** y **lvs** con la opción **-P** cuando uno de los espejos de un volumen lógico ha fallado.

```
root@link-08 ~]# vgs -a -o +devices -P
Partial mode. Incomplete volume groups will be activated read-only.
VG      #PV #LV #SN Attr   VSize VFree Devices
corey   4   4   0 rz-pnc 1.58T 1.34T
my_mirror_mimage_0(0),my_mirror_mimage_1(0)
corey   4   4   0 rz-pnc 1.58T 1.34T /dev/sdd1(0)
corey   4   4   0 rz-pnc 1.58T 1.34T unknown device(0)
corey   4   4   0 rz-pnc 1.58T 1.34T /dev/sdb1(0)
```

```
[root@link-08 ~]# lvs -a -o +devices -P
Partial mode. Incomplete volume groups will be activated read-only.
LV      VG      Attr   LSize  Origin Snap%  Move Log Copy%  Devices
my_mirror      corey mwi-a- 120.00G                my_mirror_mlog
1.95 my_mirror_mimage_0(0),my_mirror_mimage_1(0)
[my_mirror_mimage_0] corey iwi-ao 120.00G
unknown device(0)
[my_mirror_mimage_1] corey iwi-ao 120.00G
/dev/sdb1(0)
[my_mirror_mlog]   corey lwi-ao   4.00M
/dev/sdd1(0)
```

6.3. Cómo recuperarse de una falla de un espejo LVM

Esta sección proporciona un ejemplo sobre cómo realizar una recuperación cuando falla un pilar de un volumen LVM en espejo, debido a que el dispositivo subyacente para el volumen físico se cae y el parámetro **mirror_log_fault_policy** se establece a **remove**, lo que requiere que usted reconstruya de forma manual el espejo. Para obtener información sobre cómo configurar el parámetro **mirror_log_fault_policy**, consulte [Sección 6.3, “Cómo recuperarse de una falla de un espejo LVM”](#).

Cuando un espejo falla, LVM convierte el volumen en espejo en un volumen lineal, el cual continúa operando como antes, pero sin la redundancia en espejo. A ese punto, puede añadir un nuevo dispositivo de disco al sistema para utilizar como reemplazo al dispositivo físico y reconstruir el espejo.

El siguiente comando crea el volumen físico el cual será usado para el espejo.

```
[root@link-08 ~]# pvcreate /dev/sd[abcdefgh][12]
Physical volume "/dev/sda1" successfully created
Physical volume "/dev/sda2" successfully created
Physical volume "/dev/sdb1" successfully created
Physical volume "/dev/sdb2" successfully created
Physical volume "/dev/sdc1" successfully created
Physical volume "/dev/sdc2" successfully created
Physical volume "/dev/sdd1" successfully created
Physical volume "/dev/sdd2" successfully created
Physical volume "/dev/sde1" successfully created
Physical volume "/dev/sde2" successfully created
Physical volume "/dev/sdf1" successfully created
Physical volume "/dev/sdf2" successfully created
Physical volume "/dev/sg1" successfully created
Physical volume "/dev/sg2" successfully created
Physical volume "/dev/sdh1" successfully created
Physical volume "/dev/sdh2" successfully created
```

El siguiente comando crea el grupo de volúmenes **vg** y el volumen en espejo **groupfs**.

```
[root@link-08 ~]# vgcreate vg /dev/sd[abcdefgh][12]
Volume group "vg" successfully created
[root@link-08 ~]# lvcreate -L 750M -n groupfs -m 1 vg /dev/sda1 /dev/sdb1
/dev/sdc1
Rounding up size to full physical extent 752.00 MB
Logical volume "groupfs" created
```

Puede utilizar el comando **lvs** para verificar la distribución de un volumen en espejo y los dispositivos subyacentes para el espejo y el registro del espejo. Note que en el primer ejemplo el espejo no está completamente sincronizado; debe esperar hasta que el campo **Copy%** muestre 100.00 antes de continuar.

```
[root@link-08 ~]# lvs -a -o +devices
LV          VG      Attr   LSize   Origin Snap%   Move Log           Copy%
Devices
groupfs     vg      mwi-a- 752.00M                groupfs_mlog 21.28
groupfs_mimage_0(0),groupfs_mimage_1(0)
[groupfs_mimage_0] vg      iwi-ao 752.00M
/dev/sda1(0)
[groupfs_mimage_1] vg      iwi-ao 752.00M
/dev/sdb1(0)
[groupfs_mlog]   vg      lwi-ao  4.00M
/dev/sdc1(0)

[root@link-08 ~]# lvs -a -o +devices
LV          VG      Attr   LSize   Origin Snap%   Move Log           Copy%
Devices
groupfs     vg      mwi-a- 752.00M                groupfs_mlog 100.00
groupfs_mimage_0(0),groupfs_mimage_1(0)
[groupfs_mimage_0] vg      iwi-ao 752.00M
/dev/sda1(0)
[groupfs_mimage_1] vg      iwi-ao 752.00M
/dev/sdb1(0)
[groupfs_mlog]   vg      lwi-ao  4.00M      i
/dev/sdc1(0)
```

En este ejemplo, el espejo primario **/dev/sda1** falla. Cualquier actividad en el volumen en espejo hace

que LVM detecte el fallo. Cuando esto ocurre, LVM convierte el espejo en un volumen lineal sencillo. En este caso, para activar la conversión, ejecute un comando **dd**.

```
[root@link-08 ~]# dd if=/dev/zero of=/dev/vg/groupfs count=10
10+0 records in
10+0 records out
```

Puede utilizar el comando **lvs** para verificar que el dispositivo es un dispositivo lineal. Debido a que el disco falla, se produce un error de E/S.

```
[root@link-08 ~]# lvs -a -o +devices
/dev/sda1: read failed after 0 of 2048 at 0: Input/output error
/dev/sda2: read failed after 0 of 2048 at 0: Input/output error
LV      VG   Attr   LSize   Origin Snap%   Move Log Copy%  Devices
groupfs vg   -wi-a- 752.00M                               /dev/sdb1(0)
```

En este momento puede seguir utilizando el volumen lógico, pero no habrá duplicación de los datos en el espejo.

Para reconstruir el volumen en espejo puede reemplazar el disco dañado y recrear el volumen físico. Si utiliza el mismo disco en vez de reemplazarlo por uno nuevo, verá la advertencia "inconsistent" al ejecutar el comando **pvcreate**. Puede evitar esta advertencia, si ejecuta el comando **vgreduce --removemissing**.

```
[root@link-08 ~]# pvcreate /dev/sdi[12]
Physical volume "/dev/sdi1" successfully created
Physical volume "/dev/sdi2" successfully created

[root@link-08 ~]# pvscan
PV /dev/sdb1   VG vg   lvm2 [67.83 GB / 67.10 GB free]
PV /dev/sdb2   VG vg   lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdc1   VG vg   lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdc2   VG vg   lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdd1   VG vg   lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdd2   VG vg   lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sde1   VG vg   lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sde2   VG vg   lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdf1   VG vg   lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdf2   VG vg   lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdg1   VG vg   lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdg2   VG vg   lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdh1   VG vg   lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdh2   VG vg   lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdi1   VG vg   lvm2 [603.94 GB]
PV /dev/sdi2   VG vg   lvm2 [603.94 GB]
Total: 16 [2.11 TB] / in use: 14 [949.65 GB] / in no VG: 2 [1.18 TB]
```

A continuación extienda el grupo de volúmenes original con el nuevo volumen físico.

```
[root@link-08 ~]# vgextend vg /dev/sdi[12]
Volume group "vg" successfully extended

[root@link-08 ~]# pvscan
PV /dev/sdb1   VG vg   lvm2 [67.83 GB / 67.10 GB free]
PV /dev/sdb2   VG vg   lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdc1   VG vg   lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdc2   VG vg   lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdd1   VG vg   lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdd2   VG vg   lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sde1   VG vg   lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sde2   VG vg   lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdf1   VG vg   lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdf2   VG vg   lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdg1   VG vg   lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdg2   VG vg   lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdh1   VG vg   lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdh2   VG vg   lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdi1   VG vg   lvm2 [603.93 GB / 603.93 GB free]
PV /dev/sdi2   VG vg   lvm2 [603.93 GB / 603.93 GB free]
Total: 16 [2.11 TB] / in use: 16 [2.11 TB] / in no VG: 0 [0 ]
```

Convierta el volumen lineal de regreso a su estado original con el espejo.

```
[root@link-08 ~]# lvconvert -m 1 /dev/vg/groupfs /dev/sdi1 /dev/sdb1 /dev/sdc1
Logical volume mirror converted.
```

Puede utilizar el comando **lvs** para verificar que los espejos sean restaurados.

```
[root@link-08 ~]# lvs -a -o +devices
LV          VG   Attr   LSize   Origin Snap%  Move Log           Copy%
Devices
groupfs     vg   mwi-a- 752.00M                groupfs_mlog 68.62
groupfs_mimage_0(0),groupfs_mimage_1(0)
[groupfs_mimage_0] vg   iwi-ao 752.00M
/dev/sdb1(0)
[groupfs_mimage_1] vg   iwi-ao 752.00M
/dev/sdi1(0)
[groupfs_mlog]   vg   lwi-ao   4.00M
/dev/sdc1(0)
```

6.4. Cómo recuperar los metadatos de un volumen físico

Si el área de metadatos de un volumen físico es accidentalmente destruida o sobrescrita, se recibirá un mensaje de error informando que el área de metadatos es incorrecta o que el sistema fue incapaz de encontrar un volumen físico con un UUID particular. Puede recuperar los datos del volumen físico al escribir una nueva área de metadatos en el volumen físico especificando el mismo UUID de los metadatos perdidos.



Advertencia

No intente este procedimiento en un volumen lógico LVM en funcionamiento. Perderá sus datos si especifica el UUID incorrecto.

El siguiente ejemplo muestra el tipo de mensajes de salida que se pueden ver si el área de metadatos se pierde.

```
[root@link-07 backup]# lvs -a -o +devices
  Couldn't find device with uuid 'FmGRh3-zhok-iVI8-7qTD-S5BI-MAEN-NYM5Sk'.
  Couldn't find all physical volumes for volume group VG.
  Couldn't find device with uuid 'FmGRh3-zhok-iVI8-7qTD-S5BI-MAEN-NYM5Sk'.
  Couldn't find all physical volumes for volume group VG.
  ...
```

Puede encontrar el UUID para el volumen físico que fue sobrescrito al mirar en el directorio `/etc/lvm/archive`. Revise el archivo `VolumeGroupName_xxxx.vg` para ver el último archivo válido conocido de los metadatos LVM para ese grupo de volúmenes.

Alternativamente, si desactiva el volumen y usa la opción **partial** (**-P**) podrá encontrar el UUID del volumen físico corrupto.

```
[root@link-07 backup]# vgchange -an --partial
  Partial mode. Incomplete volume groups will be activated read-only.
  Couldn't find device with uuid 'FmGRh3-zhok-iVI8-7qTD-S5BI-MAEN-NYM5Sk'.
  Couldn't find device with uuid 'FmGRh3-zhok-iVI8-7qTD-S5BI-MAEN-NYM5Sk'.
  ...
```

Utilice los argumentos `--uuid` y `--restorefile` del comando **pvcreate** para restaurar el volumen físico. El siguiente ejemplo etiqueta el dispositivo `/dev/sdh1` como un volumen físico con el UUID indicado arriba, `FmGRh3-zhok-iVI8-7qTD-S5BI-MAEN-NYM5Sk`. Este comando restaura la etiqueta del volumen físico con la información de los metadatos contenidos en `VG_00050.vg`, el archivo más reciente de metadatos para el grupo de volúmenes. El argumento `restorefile` ordena al comando **pvcreate** para que el nuevo volumen físico sea compatible con el anterior en el grupo de volúmenes, garantizando que los nuevos metadatos no sean ubicados en donde el antiguo volumen físico contiene los datos (lo cual puede pasar, por ejemplo, si el comando original **pvcreate** ha utilizado las opciones de la línea de comandos que controlan la ubicación de los metadatos o si el volumen físico fue originalmente creado utilizando una diferente versión del software que utiliza diferentes valores predeterminados). El comando **pvcreate** sobrescribe solo las áreas de metadatos LVM y no afecta las áreas de datos existentes.

```
[root@link-07 backup]# pvcreate --uuid "FmGRh3-zhok-iVI8-7qTD-S5BI-MAEN-NYM5Sk" --restorefile /etc/lvm/archive/VG_00050.vg /dev/sdh1
  Physical volume "/dev/sdh1" successfully created
```

Puede utilizar el comando **vgcfgrestore** para restaurar los metadatos del grupo de volúmenes.

```
[root@link-07 backup]# vgcfgrestore VG
  Restored volume group VG
```

Ahora puede ver los volúmenes lógicos.

```
[root@link-07 backup]# lvs -a -o +devices
  LV      VG      Attr  LSize   Origin Snap%   Move Log Copy%  Devices
  stripe VG      -wi--- 300.00G                               /dev/sdh1
  (0),/dev/sda1(0)
  stripe VG      -wi--- 300.00G                               /dev/sdh1
  (34728),/dev/sdb1(0)
```

Los siguientes comandos activan los volúmenes y muestran los volúmenes activos.

```
[root@link-07 backup]# lvchange -ay /dev/VG/stripe
[root@link-07 backup]# lvs -a -o +devices
  LV      VG      Attr  LSize   Origin Snap%  Move Log Copy%  Devices
  stripe VG      -wi-a- 300.00G                /dev/sdh1
(0),/dev/sda1(0)
  stripe VG      -wi-a- 300.00G                /dev/sdh1
(34728),/dev/sdb1(0)
```

Si los metadatos LVM en disco ocupan por lo menos tanto espacio como los datos sobrescritos, este comando puede recuperar el volumen físico. Si lo que sobrescribió los metadatos pasa el área de metadatos, los datos en el volumen se pueden afectar. Puede utilizar el comando **fsck** para recuperar los datos.

6.5. Cómo reemplazar un volumen físico perdido

Si un volumen físico falla o necesita ser reemplazado, puede etiquetar un nuevo volumen físico para reemplazar aquel fallido en el grupo de volúmenes. Los pasos a seguir son los mismos que aquellos usados para recuperar los metadatos de un grupo de volúmenes, descritos en [Sección 6.4, "Cómo recuperar los metadatos de un volumen físico"](#). Puede utilizar los argumentos **--partial** y **--verbose** del comando **vgdisplay** para ver los UUID y tamaños de cualquier volumen físico que no esté presente. Si desea sustituir otro volumen físico del mismo tamaño, puede utilizar el comando **pvcreeate** con los argumentos **--restorefile** y **--uuid** para inicializar un nuevo dispositivo con el mismo UUID del volumen físico faltante. Luego puede utilizar el comando **vgcfcfgrestore** para restaurar los metadatos del grupo de volúmenes.

6.6. Cómo retirar los volúmenes físicos perdidos desde un grupo de volúmenes

Si pierde un volumen físico, puede activar los volúmenes físicos restantes en el grupo de volúmenes con la opción **vgchange**. Puede retirar todos los volúmenes lógicos que utilizan el volumen físico desde el grupo de volúmenes con el argumento **--removemissing** del comando **vgreduce**.

Se recomienda ejecutar el comando **vgreduce** con la opción **--test** para verificar lo que será destruido.

Como con la mayoría de operaciones en LVM, el comando **vgreduce** es reversible si utiliza el comando **vgcfcfgrestore** de forma inmediata para restaurar los metadatos del grupo de volúmenes a su estado previo. Por ejemplo, si utiliza la opción **--removemissing** del comando **vgreduce** sin la opción **--test** y se da cuenta que ha removido volúmenes lógicos que desea conservar, puede reemplazar el volumen físico y utilizar el comando **vgcfcfgrestore** para retornar el grupo de volúmenes a su estado previo.

6.7. Extensiones libres insuficientes para un volumen lógico

Puede ver el mensaje "Insufficient free extents" al crear un volumen lógico cuando piensa que tiene suficientes extensiones basado en la salida de los comandos **vgdisplay** o **vgs**. Esto es porque estos comandos redondean las figuras para proporcionar mensajes fáciles de leer. Para especificar el tamaño exacto, utilice el conteo de extensiones físicas libres en vez de múltiplos de bytes para determinar el tamaño del volumen lógico.

El comando **vgdisplay**, por defecto, incluye esta línea en el mensaje de salida para indicar las extensiones físicas libres.

```
# vgdisplay
--- Volume group ---
...
Free PE / Size          8780 / 34.30 GB
```

Alternativamente, puede utilizar las opciones **vg_free_count** y **vg_extent_count** del comando **vgs** para ver las extensiones libres y el número total de extensiones.

```
[root@tng3-1 ~]# vgs -o +vg_free_count,vg_extent_count
VG      #PV #LV #SN Attr   VSize  VFree  Free #Ext
testvg  2   0   0 wz--n- 34.30G 34.30G 8780 8780
```

Con 8780 extensiones físicas libres, puede ejecutar el siguiente comando con la opción **l** (en minúscula) para usar las extensiones en vez de bytes:

```
# lvcreate -l8780 -n testlv testvg
```

Esto utiliza todas las extensiones libres en el grupo de volúmenes.

```
# vgs -o +vg_free_count,vg_extent_count
VG      #PV #LV #SN Attr   VSize  VFree  Free #Ext
testvg  2   1   0 wz--n- 34.30G    0     0 8780
```

Alternativamente, puede extender el volumen lógico para usar un porcentaje del espacio libre restante en el grupo de volúmenes con el argumento **-l** del comando **lvcreate**. Para obtener mayor información, consulte la [Sección 4.4.1, “Cómo crear volúmenes lógicos lineales”](#).

Capítulo 7. Gestión de LVM con la interfaz gráfica de LVM

Además de la interfaz para la línea de comandos (CLI), LVM proporciona una interfaz gráfica de usuario (GUI) que puede ser utilizada para configurar volúmenes lógicos LVM. Para acceder a la utilidad escriba **system-config-lvm**. El capítulo de LVM de la *Guía de administración de almacenaje* proporciona instrucciones detalladas para configurar volúmenes lógicos LVM a través de esta herramienta.

Mapeador de Dispositivos

El Mapeador de Dispositivos es un controlador del kernel que proporciona un marco de trabajo para la administración de volúmenes. Ofrece un medio genérico para crear dispositivos asignados que puedan usarse como volúmenes lógicos. No conoce específicamente formatos de metadatos o grupos de volúmenes.

El Mapeador de Dispositivos proporciona la base para varias tecnologías de alto nivel. Además del LVM, el multirutas del Mapeador de Dispositivos y el comando **dmraid** usan el Mapeador de Dispositivos. La interfaz de aplicación para el Mapeador de Dispositivos es la llamada de sistema **ioctl**. La interfaz de usuario es el comando **dmsetup**.

Los volúmenes lógicos de LVM se activan con el Mapeador de Dispositivos. Cada volumen lógico se traduce en un dispositivo asignado. Cada segmento se traduce en una línea en la tabla de mapas que describe el dispositivo. El Mapeador de Dispositivos soporta una variedad de destinos de mapas, incluyendo mapas lineales, mapas entrelazados y mapas de error. Por lo tanto, por ejemplo, dos discos pueden concatenarse en un volumen lógico con un par de mapas lineales, uno para cada disco. Cuando LVM2 crea un volumen, crea un dispositivo de mapas de dispositivo subyacente que se puede consultar con el comando **dmsetup**. Para información acerca del formato de dispositivos en una tabla de mapas, consulte la [Sección A.1, “Tabla de Mapas de Dispositivo”](#). Para mayor información acerca del uso del comando **dmsetup** para solicitar un dispositivo, consulte la [Sección A.2, “Comando dmsetup”](#).

A.1. Tabla de Mapas de Dispositivo

Un dispositivo asignado está definido por una tabla que especifica cómo asignar cada rango de sectores lógicos del dispositivo mediante la Tabla de Mapas de Dispositivos. La tabla para un dispositivo asignado está constituido por una lista de líneas de la forma:

```
start length mapping [mapping_parameters...]
```

En la primera línea la Tabla de Mapeador de Dispositivo, el parámetro **start** debe ser igual a 0. Los parámetros **start + length** en una línea deben ser iguales a **start** en la línea siguiente. Los parámetros especificados en una línea de la tabla de mapas depende del tipo de **mapping** especificado en la línea.

Los tamaños en el Mapeador de Dispositivos siempre se especifican en sectores (512 bytes).

Cuando un dispositivo se especifica como un parámetro de mapas en el Mapeador de Dispositivos, puede ser llamado por el nombre de dispositivo en el sistema de archivos (por ejemplo, **/dev/hda**) o por el número mayor o menor en el formato **major:minor**. Se prefiere el formato mayor:menor porque evita bloqueos de nombre de rutas.

A continuación se visualiza una muestra de tabla de mapas para un dispositivo. En esta tabla hay cuatro destinos lineales:

```
0 35258368 linear 8:48 65920
35258368 35258368 linear 8:32 65920
70516736 17694720 linear 8:16 17694976
88211456 17694720 linear 8:16 256
```

Los primeros 2 parámetros de cada línea son el segmento de bloque de inicio y la longitud del segmento. La siguiente palabra clave es el destino de mapa, la cual en todos los casos de este ejemplo es **linear**. Las líneas restantes constan de los parámetros para un destino **linear**.

Las siguientes subdivisiones describen el formato de los siguientes mapas:

- lineal
- entrelazado
- espejo
- instantánea e instantánea-origen
- error
- cero
- multirutas
- crypt

A.1.1. Destino de mapa lineal

Un destino de mapa lineal asigna un rango continuo de bloques en otro dispositivo de bloque. El formato de un destino lineal es el siguiente:

```
start length linear device offset
```

start

iniciando bloque en dispositivo virtual

length

longitud de este segmento

device

dispositivo de bloque, relacionado por el nombre de dispositivo en el sistema de archivos o por los números mayor y menor en el formato *major:minor*

offset

iniciando desplazamiento de mapas en el dispositivo

El siguiente ejemplo muestra un destino lineal con un bloque de inicio en el dispositivo virtual de 0, una longitud de segmento de 1638400, un número par mayor:menor de 8:2 e inicio de desplazamiento para el dispositivo de 41146992.

```
0 16384000 linear 8:2 41156992
```

El siguiente ejemplo muestra un destino lineal con el parámetro de dispositivo especificado como el dispositivo **/dev/hda**.

```
0 20971520 linear /dev/hda 384
```

A.1.2. Destino de mapas entrelazados

El destino de mapas entrelazados soporta franjas a través de dispositivos físicos. Recibe como argumento el número de franjas y el tamaño de la unidad seguido por una lista de pares del nombre de dispositivo y sector. El formato de un destino entrelazado es el siguiente:

```
start length striped #stripes chunk_size device1 offset1 ... deviceN offsetN
```

Hay una serie de parámetros **device** y **offset** para cada franja.

start

iniciando bloque en dispositivo virtual

length

longitud de este segmento

#stripes

número de franjas para el dispositivo virtual

cd chunk_size

número de sectores escritos para cada franja antes de cambiar a la siguiente; debe ser potencia de 2 al menos del tamaño de la página de kernel

device

dispositivo de bloque, relacionado por el nombre de dispositivo en el sistema de archivos o por los números mayor y menor en el formato **major:minor**.

offset

iniciando desplazamiento de mapas en el dispositivo

El siguiente ejemplo muestra un destino entrelazado con tres franjas y un tamaño de unidad de 128:

```
0 73728 striped 3 128 8:9 384 8:8 384 8:7 9789824
```

0

iniciando bloque en dispositivo virtual

73728

longitud de este segmento

entrelazado 3 128

franja a través de tres dispositivos con un tamaño de unidad de 128 bloques

8:9

números mayor:menor del primer dispositivo

384

iniciando desplazamiento del mapa en el primer dispositivo

8:8

números mayor:menor de segundo dispositivo

384

iniciando desplazamiento de mapas del segundo dispositivo

8:7

números mayor:menor del tercer dispositivo

9789824

iniciando desplazamiento de mapas en el tercer dispositivo

El ejemplo a continuación muestra un destino entrelazado para 2 franjas con unidades de 256 KiB, con los parámetros de dispositivo especificados por los nombres de dispositivo en el sistema de archivos y no por los números mayor y menor.

```
0 65536 striped 2 512 /dev/hda 0 /dev/hdb 0
```

A.1.3. Espejo de destino de mapa

El espejo de destino de mapa soporta el mapa de un dispositivo lógico en espejo. El formato de un destino en espejo es el siguiente:

```
start length mirror log_type #logargs logarg1 ... logargN #devs device1 offset1  
... deviceN offsetN
```

start

iniciando bloque en dispositivo virtual

length

longitud de este segmento

log_type

Los tipos posibles de registro y sus argumentos son los siguientes:

core

El espejo es local y el registro de espejo se mantiene en el núcleo de la memoria. Este tipo de registro recibe 1 - 3 argumentos:

```
regionsize [[no]sync] [block_on_error]
```

disk

El espejo es local y el registro de espejo se mantiene en disco. Este tipo de registro recibe 2 - 4 argumentos:

```
logdevice regionsize [[no]sync] [block_on_error]
```

clustered_core

El espejo es puesto en cluster y el registro de espejo se mantiene en el núcleo de memoria. Este tipo de registro recibe 2 - 4 argumentos:

regionsize UUID [[no]sync] [block_on_error]

clustered_disk

El espejo es puesto en cluster y el registro de espejo se guarda en el disco. Este tipo de registro recibe 3 - 5 argumentos:

logdevice regionsize UUID [[no]sync] [block_on_error]

LVM mantiene un registro pequeño que utiliza para mantener el rastro de las regiones que están sincronizadas con el espejo o espejos. El argumento **regionsize** especifica el tamaño de estas regiones.

En un entorno en cluster, el argumento **UUID** es un identificador único asociado con el dispositivo de registro de espejo para que el estado de registro se pueda mantener a través del cluster.

El argumento opcional **[no]sync** sirve para especificar el espejo como "in-sync" o "out-of-sync". El argumento **block_on_error** se utiliza para pedir al espejo que responda a errores en lugar de ignorarlos.

#log_args

número de argumentos de registro que serán especificados en el mapa

logargs

los argumentos de registro para el espejo; el número de registro de argumentos de registro provisto es especificado por el parámetro **#log-args** y los argumentos de registro válidos son determinados por el parámetro **log_type**.

#devs

el número de pilares en el espejo; el dispositivo y el desplazamiento se especifican para cada pilar.

device

dispositivo de bloque para cada pilar de espejo, relacionado por el nombre de dispositivo en el sistema de archivos o por los números mayor o menor en el formato **major:minor**. Un dispositivo de bloque y desplazamiento es especificado para cada pilar de espejo, como es indicado por el parámetro **#devs**.

offset

iniciando desplazamiento de mapas en el dispositivo. Un dispositivo de bloque y desplazamiento es especificado por cada pilar de espejo, como es indicado por el parámetro **#devs**.

El siguiente ejemplo muestra un espejo de destino de mapa para un espejo en cluster con un registro de espejo guardado en disco.

```
0 52428800 mirror clustered_disk 4 253:2 1024 UUID block_on_error 3 253:3 0 253:4
0 253:5 0
```

0

iniciando bloque en dispositivo virtual

52428800

longitud de este segmento

mirror clustered_disk

destino espejo con un tipo de registro especificando que el espejo está en cluster y el registro de espejo está guardado en disco

4

4 argumentos de registro de espejo seguirán

253:2

números mayor:menor del dispositivo de registro

1024

tamaño de región que el registro de espejo utiliza para guardar rastro de lo que está en sincronización

UUID

UUID de dispositivo de registro de espejo para mantener información de registro a través de un cluster

block_on_error

espejo debe responder a errores

3

número de pilares en espejo

253:3 0 253:4 0 253:5 0

números mayor:menor y desplazamiento para dispositivos que conforman cada pilar de espejo

A.1.4. Destinos de mapa instantánea e instantánea-origen

Para crear la primera instantánea LVM de un volumen, se utilizan cuatro Mapeadores de Dispositivos:

1. Un dispositivo con un mapa **linear** conformado por la tabla de mapas del volumen de destino.

2. Un dispositivo con un mapa **linear** utilizado como dispositivo de copia-escrita (COW) para el volumen de destino; para cada escritura, los datos originales se guardan en el dispositivo COW de cada instantánea para mantener el contenido visible sin cambios (hasta que el dispositivo COW se llene).
3. Un dispositivo con un mapa de **snapshot** combinando #1 y #2, el cual es el volumen de instantánea visible
4. El volumen "original" (que emplea el número de dispositivo utilizado por el volumen de fuente original), cuya tabla es remplazada por un mapa "snapshot-origin" desde el dispositivo #1.

Un esquema de nombre fijo sirve para crear estos dispositivos. Por ejemplo, podría utilizar los siguientes comandos para crear un volumen LVM llamado **base** y un volumen de instantánea llamado **snap** basado en ese volumen.

```
# lvcreate -L 1G -n base volumeGroup
# lvcreate -L 100M --snapshot -n snap volumeGroup/base
```

Se generan cuatro dispositivos, los cuales se pueden ver con los siguientes comandos:

```
# dmsetup table|grep volumeGroup
volumeGroup-base-real: 0 2097152 linear 8:19 384
volumeGroup-snap-cow: 0 204800 linear 8:19 2097536
volumeGroup-snap: 0 2097152 snapshot 254:11 254:12 P 16
volumeGroup-base: 0 2097152 snapshot-origin 254:11

# ls -lL /dev/mapper/volumeGroup-*
brw----- 1 root root 254, 11 29 ago 18:15 /dev/mapper/volumeGroup-base-real
brw----- 1 root root 254, 12 29 ago 18:15 /dev/mapper/volumeGroup-snap-cow
brw----- 1 root root 254, 13 29 ago 18:15 /dev/mapper/volumeGroup-snap
brw----- 1 root root 254, 10 29 ago 18:14 /dev/mapper/volumeGroup-base
```

El formato para el destino **snapshot-origin** es el siguiente:

```
start length snapshot-origin origin
```

start

iniciando bloque en dispositivo virtual

length

longitud de este segmento

origin

volumen de base de instantánea

El **snapshot-origin** normalmente tendrá una o más instantáneas de base. Las lecturas serán asignadas directamente al dispositivo de respaldo. Para cada escritura, los datos originales serán guardados en el dispositivo COW de cada instantánea para mantener su contenido visible sin cambios hasta que se llene el dispositivo COW.

El formato para el destino **snapshot** es el siguiente:

```
start length snapshot origin COW-device P|N chunksize
```

start

iniciando bloque en dispositivo virtual

length

longitud de este segmento

origin

volumen de base de instantánea

COW-device

Dispositivo en el cual las unidades cambiadas de datos son almacenadas

P|N

P (Persistente) o N (No persistente); indica si la instantánea sobrevivirá después del reinicio. Para instantáneas transitorias (N) se deben guardar menos metadatos en disco; estos pueden ser guardados en memoria por el kernel.

chunksize

Tamaño en sectores de unidades de datos cambiadas que serán almacenadas en el dispositivo COW.

El siguiente ejemplo muestra un destino **snapshot-origin** con un dispositivo de origen de 254:11.

```
0 2097152 snapshot-origin 254:11
```

El siguiente ejemplo muestra un destino de **snapshot** con un dispositivo de origen de 254:11 y un dispositivo COW de 254:12. Este dispositivo de instantánea persiste a través de reinicios y el tamaño de unidad para los datos almacenados en el dispositivo COW es de 16 sectores.

```
0 2097152 snapshot 254:11 254:12 P 16
```

A.1.5. Destino de mapa error

Con un destino de mapa error, cualquier operación de E/S para el sector mapeado falla.

Un destino de mapa error sirve para pruebas. Para probar cómo se comporta un dispositivo fallido, puede crear un mapa de dispositivo con un sector incorrecto en el medio de un dispositivo, o puede cambiar el pilar de un espejo y reemplazarlo por un destino de error.

Un destino de error puede ser utilizado en lugar de un dispositivo fallido, como una forma de evitar tiempos de espera y reintentos en el dispositivo real. Puede servir como un destino intermedio mientras reorganiza los metadatos LVM durante las fallas.

El destino de mapas **error** solo recibe los parámetros **start** y **length**.

El siguiente ejemplo muestra un destino de **error**.

```
0 65536 error
```

A.1.6. Destino de mapas cero

El destino de mapas **zero** es un dispositivo de bloque equivalente de **/dev/zero**. Una operación de lectura para este mapa retorna bloques de ceros. Los datos escritos a este mapa son descartados, pero la escritura tiene éxito. El destino de mapas **zero** solo recibe los parámetros **start** y **length**.

El siguiente ejemplo muestra un destino **zero** para un dispositivo de 16TB.

```
0 65536 zero
```

A.1.7. Destino de mapas multirutas

El destino de mapas multirutas soporta el mapa de un dispositivo en multirutas. El formato para el destino **multipath** es el siguiente:

```
start length multipath #features [feature1 ... featureN] #handlerargs
[handlerarg1 ... handlerargN] #pathgroups pathgroup pathgroupargs1 ...
pathgroupargsN
```

Hay una serie de parámetros **pathgroupargs** para cada grupo de rutas.

start

iniciando bloque en dispositivo virtual

length

longitud de este segmento

#features

El número de funcionalidades de multirutas, acompañado por esas funcionalidades. Si este parámetro es cero, entonces no hay parámetro **feature** y el siguiente parámetro de mapa de dispositivos será **#handlerargs**. Actualmente, la funcionalidad multirutas soportada es **queue_if_no_path**. Esto indica que este dispositivo en multirutas se establece a operaciones de E/S de cola si no hay ninguna ruta disponible.

Por ejemplo, si la opción **no_path_retry** en el archivo **multipath.conf** se estableció a operaciones de E/S de cola solo hasta que todas las rutas hayan sido marcadas como fallidas después intentar el número de veces establecido para utilizar las rutas, el mapa aparecerá como sigue hasta que todos los controladores de ruta hayan fallado el número de controles especificado.

```
0 71014400 multipath 1 queue_if_no_path 0 2 1 round-robin 0 2 1 66:128 \
1000 65:64 1000 round-robin 0 2 1 8:0 1000 67:192 1000
```

Después de que todos los controladores de ruta hayan fallado el número de controles especificado, el mapa aparecería así:

```
0 71014400 multipath 0 0 2 1 round-robin 0 2 1 66:128 1000 65:64 1000 \
round-robin 0 2 1 8:0 1000 67:192 1000
```

#handlerargs

El número de argumentos del manejador de hardware, seguido por esos argumentos. Un manejador de hardware especifica un módulo que será utilizado para realizar acciones específicas de hardware al cambiar grupos de rutas o al manejar errores de E/S. Si se establece a 0, entonces el siguiente parámetro será **#pathgroups**.

#pathgroups

El número de grupos de ruta. Un grupo de ruta es una serie de rutas sobre las cuales un dispositivo en multirutas cargará equilibrio. Hay una serie de parámetros **pathgroupargs** para cada grupo de rutas.

pathgroup

El siguiente grupo de ruta para probar.

pathgroupsargs

Cada grupo de ruta consta de los siguientes argumentos:

```
pathselector #selectorargs #paths #pathargs device1 ioreqs1 ... deviceN ioreqsN
```

Hay una serie de argumentos de ruta para cada ruta en el grupo de rutas.

pathselector

Especifica el algoritmo en uso para determinar qué ruta utilizar en este grupo de ruta para la siguiente operación de E/S.

#selectorargs

El número de argumentos de selector de ruta que sigue este argumento en el mapa de multirutas. Actualmente, el valor de este argumento es siempre 0.

#paths

El número de rutas en este grupo de rutas.

#pathargs

El número de argumentos de ruta especificado para cada ruta en este grupo. Actualmente este número es siempre 1, el argumento **ioreqs**.

device

El número de dispositivo de bloque de la ruta, relacionada por los números mayor y menor en el formato **major:minor**

ioreqs

El número de peticiones de E/S para dirigirse a esta ruta antes de cambiar a la próxima ruta en el grupo actual.

Figura A.1, “Destino de mapas multirutas” shows the format of a multipath target with two path groups.

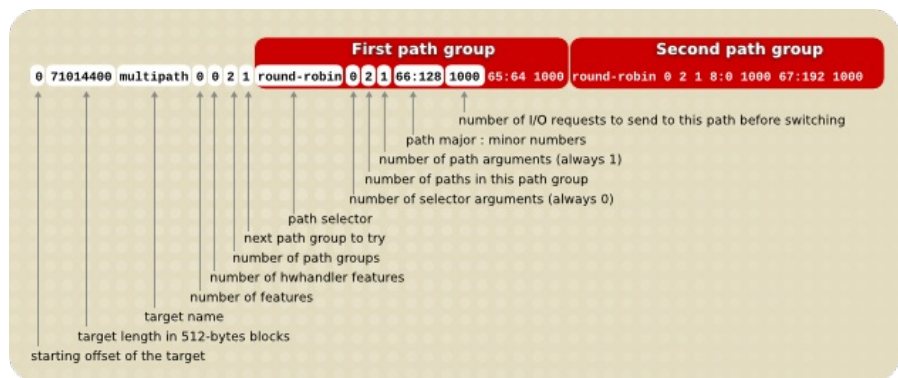


Figura A.1. Destino de mapas multirutas

El siguiente ejemplo muestra una definición de destino de recuperación de fallos para el mismo dispositivo multirutas. En este destino hay tres grupos de cuatro grupos de ruta, con una sola ruta abierta por grupo de ruta para que el dispositivo en multirutas utilice solamente una ruta a la vez.

```
0 71014400 multipath 0 0 4 1 round-robin 0 1 1 66:112 1000 \
round-robin 0 1 1 67:176 1000 round-robin 0 1 1 68:240 1000 \
round-robin 0 1 1 65:48 1000
```

El siguiente ejemplo muestra una definición de destino de difusión total (multibus) para el mismo dispositivo en multirutas. En este destino hay únicamente un grupo de ruta, el cual incluye todas las demás rutas. En esta configuración, multirutas difunde la carga equitativamente a todas las rutas.

```
0 71014400 multipath 0 0 1 1 round-robin 0 4 1 66:112 1000 \
67:176 1000 68:240 1000 65:48 1000
```

Para mayor información sobre multirutas, consulte el documento *Uso de multirutas de Mapeador de Dispositivos*.

A.1.8. Destino de mapas crypt

El destino **crypt** encripta los datos que pasan por el dispositivo especificado. Utiliza el Crypto API de kernel.

El formato para el destino **crypt** es el siguiente:

```
start length crypt cipher key IV-offset device offset
```

start
 iniciando bloque en dispositivo virtual

length
 longitud de este segmento

cipher

Cipher consta de ***cipher[-chainmode]-ivmode[:iv options]***.

cipher

Los cipher disponibles se listan en `/proc/crypto` (por ejemplo, **aes**).

chainmode

Siempre usan **cbc**. No utilice **ebc**; no utiliza un vector inicial (IV).

ivmode[:iv options]

IV es un vector inicial usado para variar la codificación. El modo IV es **plain** o **essiv:hash**. Un **ivmode** de **-plain** usa el número de sector (más desplazamiento IV) como el IV. Un **ivmode** de **-essiv** es una mejora para evitar la debilidad de la marca de agua digital.

key

Clave codificada, suministrada en hex

IV-offset

Desplazamiento de Vector inicial (IV)

device

dispositivo de bloque, relacionado por el nombre de dispositivo en el sistema de archivos o por los números mayor y menor en el formato **major:minor**

offset

iniciando desplazamiento de mapas en el dispositivo

El siguiente es un ejemplo de un destino **crypt**.

```
0 2097152 crypt aes-plain 0123456789abcdef0123456789abcdef 0 /dev/hda 0
```

A.2. Comando **dmsetup**

El comando **dmsetup** es una envoltura de la línea de comando para la comunicación con el Mapeador de Dispositivos. Para información general del sistema sobre dispositivos LVM, puede hallar de utilidad las opciones **info**, **ls**, **status**, y **deps** del comando **dmsetup** como se describe en las siguientes subdivisiones.

Para obtener mayor información sobre las capacidades y opciones de **dmsetup**, consulte la página de manual (8) de **dmsetup**.

A.2.1. Comando **info dmsetup**

El comando **dmsetup info device** proporciona información resumida sobre los dispositivos del Mapeador de Dispositivos. Si no especifica un nombre de dispositivo, la salida será información sobre

todos los dispositivos del Mapeador de Dispositivos configurados actualmente. Si especifica el dispositivo, entonces este comando generará información sobre ese dispositivo únicamente.

El comando **dmsetup info** proporciona información en la siguientes categorías:

Name

El nombre del dispositivo. Un dispositivo LVM se expresa como el grupo de volúmenes y el nombre de volumen lógico separado por un guión. Un guión en el nombre original se traduce a dos guiones.

State

Los posibles estados de dispositivo son **SUSPENDED**, **ACTIVE**, y **READ-ONLY**. El comando **dmsetup suspend** establece un dispositivo a un estado de dispositivo **SUSPENDED**. Cuando un dispositivo es suspendido, todas las operaciones de E/S para ese dispositivo se detienen. El comando **dmsetup resume** restaura un estado de dispositivo a **ACTIVE**.

Read Ahead

El número de bloques de datos leído por el sistema antes que cualquier archivo en el cual estén en curso operaciones de lectura. Por defecto, el kernel elige automáticamente un valor adecuado. Puede cambiar este valor con la opción **--readahead** del comando **dmsetup**.

Tables present

Los posibles estados para esta categoría son **LIVE** e **INACTIVE**. Un estado **INACTIVE** indica que ha cargado una tabla que se va a intercambiar cuando el comando **dmsetup resume** restaure el estado de dispositivo a **ACTIVE**, en cuyo punto el estado de la tabla se convierte en **LIVE**. Para obtener mayor información, consulte la página de manual **dmsetup**.

Open count

La cuenta abierta de referencia indica las veces que se abre el dispositivo. El comando **mount** abre un dispositivo.

Event number

El número actual de eventos recibidos. Emitir un comando **dmsetup wait n** permite al usuario esperar que el evento cambie, bloqueando la llamada hasta que ésta sea recibida.

Major, minor

Número de dispositivo mayor y menor

Number of targets

El número de partes que componen un dispositivo. Por ejemplo, un dispositivo lineal que genere 3 discos tendría 3 destinos. Un dispositivo lineal compuesto por el comienzo y el final de un disco, pero no por la mitad tendría 2 destinos.

UUID

UUID del dispositivo.

El siguiente ejemplo muestra salida parcial para el comando **dmsetup info**.

```
[root@ask-07 ~]# dmsetup info
Name:                testgfsvg-testgfs1v1
State:               ACTIVE
Read Ahead:         256
Tables present:     LIVE
Open count:         0
Event number:       0
Major, minor:       253, 2
Number of targets:  2
UUID: LVM-K528WUGQgPadNXycFrrf9LnPlUMswgkCkpgPIgYzSvigM7SfewCypddNSwtNzc2N
...
Name:                VolGroup00-LogVol100
State:               ACTIVE
Read Ahead:         256
Tables present:     LIVE
Open count:         1
Event number:       0
Major, minor:       253, 0
Number of targets:  1
UUID: LVM-t0cS1kqFV9drb0X1Vr8sxeYP0tqcrpdegyqj5lZxe45JMG1mvtqLmbLpBcenh2L3
```

A.2.2. El comando **dmsetup ls**

Sirve para listar nombres de dispositivos de dispositivos mapeados con el comando **dmsetup ls**. Puede listar dispositivos que tengan al menos un destino de un tipo especificado con el comando **dmsetup ls --target *target_type***. Para otras opciones del comando **dmsetup ls**, consulte la página de manual **dmsetup**.

El siguiente ejemplo muestra el comando para listar nombres de dispositivos de dispositivos mapeados configurados actualmente.

```
[root@ask-07 ~]# dmsetup ls
testgfsvg-testgfs1v3    (253, 4)
testgfsvg-testgfs1v2    (253, 3)
testgfsvg-testgfs1v1    (253, 2)
VolGroup00-LogVol101    (253, 1)
VolGroup00-LogVol100    (253, 0)
```

El siguiente ejemplo muestra el comando para listar nombres de dispositivos de mapas de espejos configurados actualmente.

```
[root@grant-01 ~]# dmsetup ls --target mirror
lock_stress-grant--02.1722    (253, 34)
lock_stress-grant--01.1720    (253, 18)
lock_stress-grant--03.1718    (253, 52)
lock_stress-grant--02.1716    (253, 40)
lock_stress-grant--03.1713    (253, 47)
lock_stress-grant--02.1709    (253, 23)
lock_stress-grant--01.1707    (253, 8)
lock_stress-grant--01.1724    (253, 14)
lock_stress-grant--03.1711    (253, 27)
```

Las configuraciones LVM que están apiladas en dispositivos multirutas o en otro mapeador de dispositivo puede ser complejas de clasificar. El comando **dmsetup ls** proporciona una opción --

tree que muestra dependencias entre dispositivos como un árbol, como en el siguiente ejemplo:

```
# dmsetup ls --tree
vgtest-lvmir (253:13)
├─vgtest-lvmir_mimage_1 (253:12)
│   └─mpathep1 (253:8)
│       └─mpathe (253:5)
│           ├── (8:112)
│           └─ (8:64)
├─vgtest-lvmir_mimage_0 (253:11)
│   └─mpathcp1 (253:3)
│       └─mpathc (253:2)
│           ├── (8:32)
│           └─ (8:16)
└─vgtest-lvmir_mlog (253:4)
    └─mpathfp1 (253:10)
        └─mpathf (253:6)
            ├── (8:128)
            └─ (8:80)
```

A.2.3. Comando **dmsetup status**

El comando **dmsetup status device** proporciona información de estatus para cada destino en un dispositivo determinado. Si no especifica el nombre de dispositivo, la salida será información sobre todos los dispositivos de Mapeador de Dispositivos actualmente configurados. Puede listar el estatus únicamente de dispositivos que tengan al menos un destino de un tipo especificado con el comando **dmsetup status --target target_type**.

El siguiente ejemplo muestra el comando para listar el estatus de los destinos en todos los dispositivos mapeados configurados actualmente.

```
[root@ask-07 ~]# dmsetup status
testgfsvg-testgfs1v3: 0 312352768 linear
testgfsvg-testgfs1v2: 0 312352768 linear
testgfsvg-testgfs1v1: 0 312352768 linear
testgfsvg-testgfs1v1: 312352768 50331648 linear
VolGroup00-LogVol01: 0 4063232 linear
VolGroup00-LogVol00: 0 151912448 linear
```

A.2.4. Comando **dmsetup deps**

El comando **dmsetup deps device** proporciona un listado de pares (mayor, menor) para dispositivos relacionados por tabla de mapas para el dispositivo especificado. Si no se especifica el nombre de dispositivo, la salida será información sobre todos los dispositivos del Mapeador de Dispositivos.

El siguiente ejemplo muestra el comando para listar las dependencias de todos los dispositivos mapeados configurados actualmente.

```
[root@ask-07 ~]# dmsetup deps
testgfsvg-testgfs1v3: 1 dependencies : (8, 16)
testgfsvg-testgfs1v2: 1 dependencies : (8, 16)
testgfsvg-testgfs1v1: 1 dependencies : (8, 16)
VolGroup00-LogVol01: 1 dependencies : (8, 2)
VolGroup00-LogVol00: 1 dependencies : (8, 2)
```

El siguiente ejemplo muestra el comando para listar las dependencias únicamente del dispositivo

lock_stress-grant--02.1722:

```
[root@grant-01 ~]# dmsetup deps lock_stress-grant--02.1722
3 dependencies : (253, 33) (253, 32) (253, 31)
```

A.3. Soporte de mapeador de dispositivo para el Gestor de dispositivos udev

El papel principal del gestor de dispositivos **udev** es el de proveer una forma dinámica para configurar los nodos en el directorio **/dev**. La creación de estos nodos está dirigida por la aplicación de las reglas **udev** en el espacio de usuario. Estas reglas se procesan en eventos **udev** enviados directamente desde el kernel como resultado de la adición, remoción o cambio de dispositivos particulares. Así se proporciona un mecanismo central y conveniente para soporte de conexión en caliente.

Además de crear nodos reales, el gestor de dispositivo **udev** puede crear enlaces simbólicos también con sus propios nombres, proporcionando a los usuarios la libertad de escoger su propio nombre personalizado y la estructura de directorio en el directorio **/dev**, en caso de necesidad.

Cada evento **udev** contiene información básica sobre el dispositivo que está siendo procesado, tal como su nombre, el subsistema al que pertenece, el tipo de dispositivo, su número mayor e inferior utilizado y el tipo del evento. De acuerdo con esto, y teniendo la posibilidad de acceder a toda la información hallada en el directorio **/sys** que también es accesible dentro de las reglas **udev**, los usuarios pueden utilizar filtros sencillos basados en esta información y ejecutar las reglas condicionalmente con base en esta información.

El gestor de dispositivo **udev** también proporciona una forma centralizada de establecer los permisos de nodos. Un usuario puede fácilmente agregar un conjunto de reglas personalizado para definir los permisos para cualquier dispositivo especificado por cualquier bit de información que está disponible mientras procesa el evento.

También es posible agregar directamente ganchos de programa en reglas **udev**. El gestor de dispositivo **udev** puede llamar estos programas para proveer más procesamiento necesario para manejar el evento. También, el programa puede exportar variables de entorno como resultado de este procesamiento. Los resultados proporcionados pueden utilizarse más adelante en las reglas como fuente complementaria de información.

Cualquier software que utilice la biblioteca **udev** puede recibir y procesar eventos **udev** con toda la información que está disponible, por lo tanto el procesamiento no está ligado al demonio **udev** únicamente.

A.3.1. Integración udev con el Mapeador de Dispositivos

En RHEL 6, el Mapeador de Dispositivo proporciona soporte directo para integración **udev**. De esta manera sincroniza el Mapeador de Dispositivo con todo el procesamiento **udev** relacionado con dispositivos de Mapeador de Dispositivo, incluyendo dispositivos LVM. La sincronización es necesaria porque la aplicación de reglas en el demonio **udev** es una forma de procesamiento paralelo con el programa que es la fuente de los cambios de dispositivos (tales como **dmsetup** y LVM). Sin este soporte, era un problema común para el usuario intentar retirar un dispositivo que estaba aún abierto y procesado por reglas **udev** como resultado de un cambio de evento previo; esto era en particular común cuando había un tiempo muy corto entre cambios para ese dispositivo.

El lanzamiento de RHEL 6 proporciona oficialmente reglas **udev** soportadas para dispositivos de Mapeador de Dispositivo en general y para LVM. [Tabla A.1, “Reglas udev para Dispositivos de](#)

[Mapeador de Dispositivo](#) resume dichas reglas, las cuales se instalan en `/lib/udev/rules.d`.

Tabla A.1. Reglas udev para Dispositivos de Mapeador de Dispositivo

Nombre de archivo	Descripción
10-dm.rules	<p>Contiene reglas básicas y generales de Mapeador de Dispositivo y crea los symlinks en <code>/dev/mapper</code> con un destino <code>/dev/dm-N</code> donde N es un número asignado dinámicamente por el kernel a un dispositivo (<code>/dev/dm-N</code> es un nodo)</p> <p>NOTA: nodos <code>/dev/dm-N</code> nunca deben utilizarse en scripts para acceder el dispositivo puesto que el número N es asignado dinámicamente y cambia con la secuencia de cómo se activan los dispositivos. Por lo tanto, deben utilizarse los nombres verdaderos en el directorio <code>/dev/mapper</code>. Este diseño es para soportar requerimientos udev de cómo deben ser creados nodos y symlinks.</p>
11-dm-lvm.rules	<p>Contiene reglas aplicadas para dispositivos LVM y crea los symlinks para los volúmenes lógicos de grupo. Los symlinks son creados en el directorio <code>/dev/vgname</code> con un destino <code>/dev/dm-N</code>.</p> <p>NOTA: Para ser consistente con las normas de nomenclatura de reglas futuras para subsistemas de Mapeador de Dispositivo, las reglas udev deben seguir el formato 11-dm-subsystem_name.rules. Cualquier usuario libdevmapper que proporcione reglas udev también debe seguir este estándar.</p>
13-dm-disk.rules	<p>Contiene reglas para aplicar a dispositivos de Mapeador de Dispositivo en general y crea symlinks en los directorios <code>/dev/disk/by-id</code>, <code>/dev/disk/by-uuid</code> y <code>/dev/disk/by-uuid</code>.</p>
95-dm-notify.rules	<p>Contiene la regla para notificar el proceso de espera mediante libdevmapper (justo como LVM y dmsetup). La notificación se hace después de que se aplican todas las reglas anteriores para asegurarse de que todo el procesamiento udev esté completo. Luego, el proceso notificado se reanuda.</p>

Puede agregar reglas de permisos personalizadas por medio del archivo **12-dm-permissions.rules**. Este archivo *no* está instalado en el directorio `/lib/udev/rules`; se encuentra en el directorio `/usr/share/doc/device-mapper-version`. El archivo **12-dm-permissions.rules** es una plantilla que contiene algunos consejos de cómo establecer los permisos, basados en algunas reglas coincidentes dadas como ejemplo; el archivo contiene ejemplos para situaciones más comunes. Puede editar este archivo y colocarlo manualmente en el directorio `/etc/udev/rules.d` donde sobrevivirá actualizaciones, por lo que la configuración permanecerá.

Estas reglas establecen todas las variables básicas que podrían ser utilizadas por otras reglas durante el procesamiento de eventos.

Las siguientes variables están establecidas en `10-dm.rules`:

- ▶ **DM_NAME**: nombre de dispositivo de Mapeador de Dispositivo
- ▶ **DM_UUID**: UUID de dispositivo de Mapeador de Dispositivo
- ▶ **DM_SUSPENDED**: el estado suspendido del dispositivo del Mapeador de Dispositivos
- ▶ **DM_UDEV_RULES_VSN**: versión de reglas **udev** (principalmente para todas las otras reglas para verificar que variables mencionadas anteriormente sean establecidas por reglas Mapeador de Dispositivo oficiales)

Las siguientes variables se establecen en `11-dm-lvm.rules`:

- ▶ **DM_LV_NAME**: nombre de volumen lógico
- ▶ **DM_VG_NAME**: nombre de grupo de volúmenes
- ▶ **DM_LV_LAYER**: nombre de capa de LVM

Todas las variables pueden ser utilizadas en el archivo `12-dm-permissions.rules` para definir un permiso para dispositivos de Mapeador de Dispositivos, como se documentó en el archivo `12-dm-permissions.rules` file.

A.3.2. Comandos e Interfaces que soportan udev

[Tabla A.2, “Comandos dmsetup para soportar udev”](#) resume los comandos **dmsetup** que soportan integración **udev**.

Tabla A.2. Comandos dmsetup para soportar udev

Comando	Descripción
dmsetup udevcomplete	Sirve para informar que udev ha completado el procesamiento de reglas y desbloquea proceso de espera (llamado desde dentro de las reglas udev en <code>95-dm-notify.rules</code>)
dmsetup udevcomplete_all	Utilizado en propósitos de depuración para desbloquear manualmente todos los procesos en espera
dmsetup udevcookies	Utilizado en propósitos de depuración, para mostrar todas las cookies existentes (system-wide semaphores)
dmsetup udevcreatecookie	Utilizado para crear una cookie (semaphore) manualmente. Sirve para ejecutar más procesos bajo un recurso de sincronización.
dmsetup udevreleasecookie	Utilizado para esperar todo el procesamiento de udev relacionado con todos los procesos bajo esa cookie de sincronización.

Las opciones **dmsetup** que soportan integración **udev** son las siguientes.

--udevcookie

Necesita definirse para todos los procesos **dmsetup** que queramos añadir a una transacción **udev**. Se utiliza junto con **udevcreatecookie** y **udevreleasecookie**:

```
COOKIE=$(dmsetup udevcreatecookie)
dmsetup command --udevcookie $COOKIE ...
dmsetup command --udevcookie $COOKIE ...
...
dmsetup command --udevcookie $COOKIE ...
dmsetup udevreleasecookie --udevcookie $COOKIE
```

Además de usar la opción **--udevcookie**, puede exportar la variable a un entorno del proceso:

```
export DM_UDEV_COOKIE=$(dmsetup udevcreatecookie)
dmsetup command ...
dmsetup command ...
...
dmsetup command ...
```

--noudevrules

Desactiva reglas udev. Los nodos y symlinks serán creados por el mismo **libdevmapper** (la forma anterior). Esta opción es para propósitos de depuración, si **udev** no funciona correctamente.

--noudevsync

Desactiva sincronización **udev**. También es para propósitos de depuración.

Para obtener mayor información sobre **dmsetup** y sus opciones, consulte la página de manual **dmsetup(8)**.

Los comandos LVM soportan las siguientes opciones que soportan integración **udev**:

- ▶ **--noudevrules**: como para el comando **dmsetup**, desactiva reglas **udev**.
- ▶ **--noudevsync**: como para el comando **dmsetup**, desactiva sincronización **udev**.

El archivo **lvm.conf** incluye las siguientes opciones que soportan integración **udev**:

- ▶ **udev_rules**: activa/desactiva **udev_rules** para todos los comandos LVM2 globalmente
- ▶ **udev_sync**: activa/desactiva sincronización **udev** para todos los comandos LVM globalmente

Para obtener mayor información sobre opciones del archivo **lvm.conf**, consulte los comentarios en línea en el archivo **lvm.conf**.

Archivos de configuración LVM

LVM soporta varios archivos de configuración. Durante el inicio del sistema, el archivo de configuración **lvm.conf** se carga desde el directorio especificado por la variable de entorno **LVM_SYSTEM_DIR**, la cual está establecida de forma predeterminada a **/etc/lvm**.

El archivo **lvm.conf** puede especificar archivos de configuración adicional. Los parámetros en archivos de configuración posteriores tienen prioridad sobre los parámetros en archivos anteriores. Para ver los parámetros en uso después de cargar todos los archivos de configuración, ejecute el comando **lvm dumpconfig**.

Para obtener mayor información sobre carga de archivos de configuración adicionales, consulte la [Sección C.2, “Etiquetas de host”](#).

B.1. Archivos de configuración LVM

Los siguientes archivos se utilizan en la configuración de LVM:

/etc/lvm/lvm.conf

Archivos de configuración central leídos por las herramientas.

etc/lvm/lvm_etiqueta_de_host.conf

Para cada etiqueta de host, se lee un archivo de configuración si este existe: **lvm_etiqueta_de_host.conf**. Si este archivo define etiquetas nuevas, entonces otros archivos de configuración se añadirán a la lista para leer. Para obtener mayor información sobre etiquetas de host, consulte la [Sección C.2, “Etiquetas de host”](#).

Además de los archivos de configuración de LVM, los sistemas que ejecutan LVM incluyen los siguientes archivos que afectan la configuración del sistema LVM:

/etc/lvm/cache

Archivo de caché del filtro de nombres de dispositivos (configurable)

/etc/lvm/backup/

Directorio para las copias de seguridad automáticas de los metadatos del grupo de volúmenes (configurable).

/etc/lvm/archive/

Directorio para los archivos de metadatos del grupo de volúmenes (configurables con respecto a la profundidad de ruta del directorio y el historial).

/var/lock/lvm/

En una configuración de un solo host, bloquea los archivos para prevenir la corrupción de los metadatos debido a la ejecución paralela de varias herramientas; en un cluster, se utiliza DLM a lo largo del cluster.

B.2. Archivo `lvm.conf` de ejemplo

La siguiente es una muestra del archivo de configuración `lvm.conf`. Su archivo puede diferir un poco de este.

```
# Este es un ejemplo de archivo de configuración para el sistema LVM2.
# Contiene los parámetros predeterminados que se usarían si no hubiera
# /etc/lvm/lvm.conf file.
#
# Consulte 'man lvm.conf' para obtener mayor información que incluya la
distribución del archivo .
#
# Para colocar este archivo en un directorio diferente y sobrescribir /etc/lvm
establezca
# la variable de entorno LVM_SYSTEM_DIR antes de ejecutar las herramientas.

# Esta sección le permite configurar los dispositivos de bloque que deben ser
utilizados por
# el sistema de LVM.
dispositivos {

    # ¿Dónde desea que los grupos de volumen aparezcan ?
    dir = "/dev"

    # Un grupo de directorios que contienen los nodos de dispositivos que usted
desea
    # usar con LVM2.
    scan = [ "/dev" ]

    # Si varias entradas en los directorios examinados corresponden al mismo
dispositivo de bloque
    # y las herramientas necesitan mostrar un nombre por dispositivo,
    # todos los nombres de ruta coinciden con cada elemento en la siguiente lista
    # de expresión regulares a su vez y se utiliza la primera concordancia.
    # nombres_preferidos = [ ]

    # Trate de evitar el uso de nombres no descriptivos /dev/dm-N names, si está
presente.
    nombres_preferidos = [ "^/dev/mpath/", "^/dev/mapper/mpath", "^/dev/[hs]d" ]

    # Un filtro que dice a LVM2 que únicamente use un conjunto restringido de
dispositivos.
    # El filtro consta de un conjunto de expresiones regulares. Estas
    # expresiones pueden estar delimitadas por un caracter de su elección y
    # por un prefijo con una 'a' (para aceptar) o 'r' (para rechazar).
    # La primera expresión que concuerde con el nombre de dispositivo determina
si
    # el dispositivo será aceptado o rechazado (ignorado). Los dispositivos que
    # no concuerden con ningún patrón son aceptados.

    # Tenga cuidado si no hay enlaces simbólicos o entradas de sistema de archivos
múltiples para el mismo dispositivo,
    # ya que cada nombre es comparado independientemente
    # con la lista de patrones. El efecto es que si algún nombre coincide con
algún patrón 'a'
    # el dispositivo es aceptado; y si algún nombre coincide con el patrón 'r'
    # es rechazado; de lo contrario es aceptado.

    # No tenga más de una línea de filtro activa a la vez: solamente se utiliza
una.

    # Ejecute vgscan después de cambiar este parámetro para asegurarse de que el
archivo cache sea regenerado (ver abajo).
    # Si no hace lo esperado, revise la salida de 'vgscan -vvvv'.
```



```
# De forma predeterminada, aceptamos cada dispositivo de bloque:
filtro = [ "a/*/" ]

# Excluir la unidad de cdrom
# filtro = [ "r|/dev/cdrom|" ]

# Al probar me gustaría trabajar con dispositivos de bucle de retorno:
# filtro = [ "a/loop/", "r/*/" ]

# O quizás todos los bucles y unidades ide a excepción de hdc:
# filtro =[ "a|loop|", "r|/dev/hdc|", "a|/dev/ide|", "r|.*|" ]

# Use anclas si desea que sean verdaderamente específicos
# filtro = [ "a|^/dev/hda8$|", "r/*/" ]

# Los resultados del filtraje se guardan en un disco para evitar
# reescanear dispositivos dud (lo cual puede tardarse mucho tiempo).
# De forma predeterminada esta cache se almacena en el directorio
/etc/lvm/cache
# en un archivo llamado '.cache'.
# Se puede borrar el contenido: las herramientas lo regeneran.
# (El antiguo parámetro 'cache' aún se respeta si ninguno de los nuevos está
# presente.)
cache_dir = "/etc/lvm/cache"
cache_file_prefix = ""

# Puede desactivar la escritura de este archivo cache estableciéndolo a 0.
write_cache_state = 1

# Parámetros avanzados.

# Lista de pares de tipos de dispositivos de bloque adicionales aceptables
encontrados
# en /proc/devices con un máximo número de particiones (no-cero).
# tipos = [ "fd", 16 ]

# Si se monta sysfs (2.6 kernels) restringe el escaneo de dispositivos a
# los dispositivos de bloque que cree que son válidos.
# 1 habilita; 0 inhabilita.
sysfs_scan = 1

# De forma predeterminada, LVM2 ignorará los dispositivos usados como
componente de
# dispositivos RAID (md) de software al buscar superbloques md.
# 1 habilita; 0 inhabilita.
md_component_detection = 1

# De forma predeterminada, si un volumen físico es colocado directamente
después de un dispositivo md, LVM2
# alineará sus bloques de datos con el dispositivo de ancho de banda md.
# 1 habilita; 0 inhabilita.
md_chunk_alignment = 1

# La alineación predeterminada del inicio del área de datos en MB. Si se
establece a 0,
# se usará un valor de 64KB. Establezca a 1 para 1MiB, 2 para 2MiB, etc.
# default_data_alignment = 1
```

```

# De forma predeterminada, el inicio de área de datos de volumen físico será
un múltiplo de
# 'minimum_io_size' u 'optimal_io_size' expuesto en sysfs.
# - minimum_io_size - la solicitud mínima que el dispositivo puede realizar
# sin incurrir en una sanción de read-modify-write (por ejemplo, tamaño de
trozo de MD)
# - optimal_io_size - la unidad de dispositivo preferida para recibir E/S
# (e.g. MD's stripe width)
# minimum_io_size se utiliza si optimal_io_size es indefinido (0).
# Si md_chunk_alignment está activado, detecta el optimal_io_size.
# Este parámetro tiene prioridad sobre md_chunk_alignment.
# 1 habilita; 0 inhabilita.
data_alignment_detection = 1

# Alineación (en KB) de inicio de área de datos al crear un nuevo volumen
físico.
# md_chunk_alignment y data_alignment_detection se desactivan.
# Establecer a 0 para la alineación predeterminada (ver:
data_alignment_default)
# o el tamaño de página, si es más grande.
data_alignment = 0

# De forma predeterminada, el inicio de datos alineados de volumen físico se
cambiará por
# el 'alignment_offset' expuesto en sysfs. Esta compensación suele ser 0,
pero
# puede ser no-cero; por ejemplo.: algunas unidades de sectores de 4KB que
compensan el
# particionamiento de windows tendrán un alignment_offset de 3584 bytes
# (sector 7 es el bloque lógico alineado más bajo, los sectores de 4KB inician
# en LBA -1, y como consecuencia el sector 63 se alinea en un límite de 4KB).
# Sin embargo, observe que pvcreate --dataalignmentoffset omitirá esta
detección.
# 1 habilita; 0 inhabilita.
data_alignment_offset_detection = 1

# Si, al escanear el sistema para volúmenes físicos, LVM2 encuentra un mapeador
de dispositivo
# que tiene suspendida su E/S, espera a que sea accesible.
# Establézcalo a 1 para omitir estos dispositivos. Esto solamente será
necesario en situaciones de
# recuperación.
ignore_suspended_devices = 0

# Durante cada operación LVM se cuentan los errores recibidos de cada
dispositivo.
# Si el contador de un dispositivo particular excede el límite establecido
aquí, no
# se enviará más E/S al dispositivo para lo que resta de la operación
respectiva.
# Al establecer el parámetro a inhabilita todos los contadores.
disable_after_error_count = 0

# Permitir el uso de pvcreate --uuid sin requerir --restorefile.
require_restorefile_with_uuid = 1
}

# Esta sección le permite configurar la forma en que LVM selecciona
# espacio libre para sus volúmenes lógicos.
#asignación {

```

```
# Durante la búsqueda de espacio libre para extender un volumen lógico, la
# política de asignación "cling"
# eligirá un espacio en los mismos volúmenes físicos como el último segmento
# del
# volumen lógico existente. Si no hay espacio suficiente y se define aquí una
# lista de
# etiquetas, revisará si alguna de ellas está conectada al volumen lógico
# concernientes
# y luego busca concordar esas etiquetas para que coincidan con las etiquetas
# de volúmenes físicos
# entre las extensiones existentes y las nuevas extensiones.
# Use la etiqueta especial "@"*"" como comodín para que coincida con cualquier
# etiqueta de volumen físico.
#
# Ejemplo: Los volúmenes lógicos se copian en espejo entre dos sitios dentro de
# un único Grupo de Volumen.
# Los volúmenes físicos se etiquetan con @site1 o @site2 para indicar dónde
# están situados.
#
# cling_tag_list = [ "@site1", "@site2" ]
# cling_tag_list = [ "@"*"" ]
#}

# Esta sección le permite configurar la naturaleza de la información que reporta
# LVM2.
log {

    # Controla los mensajes enviados a stdout o stderr.
    # Hay tres niveles de verbosidad, 3 siendo el más verboso.
    verbose = 0

    # ¿Debe enviar mensajes de registro a través de syslog?
    # 1 si sí; 0 si no.
    syslog = 1

    # ¿Debemos registrar mensajes de error o depuración a un archivo?
    # No hay archivo de registro predeterminado.
    #file = "/var/log/lvm2.log"

    # ¿Debemos sobrescribir el archivo de registro cada vez que se ejecute el
# programa?
    # Lo añadimos de forma predeterminada.
    overwrite = 0

    # ¿Qué nivel de mensajes de registro debemos enviar al archivo de registro y
# o al syslog?
    # Hay 6 niveles syslog-like actualmente en uso - 2 a 7 inclusive.
    # 7 es el más verboso (LOG_DEBUG).
    level = 0

    # Formato de mensajes de salida
    # Si o no (1 o 0) los mensajes para marcar los mensajes de acuerdo a su
# severidad
    indent = 1

    # Si o no (1 o 0) para mostrar el nombre de comando en cada salida de línea
# command_names = 0

    # Un prefijo para usar antes del texto del mensaje (pero después del nombre de
# comandos,
```

```

# si se ha seleccionado). El predeterminado es dos espacios, para que usted
puede ver/grep la severidad
# de cada mensaje.
prefix = "  "

# Para que los mensajes se vean similares a las herramientas LVM originales
use:
#   indent = 0
#   command_names = 1
#   prefix = " -- "

# Establézcalo si desea mensajes de registro durante la activación.
# No lo use en situaciones de poca memoria (puede interbloquear).
# activation = 0
}

# Configuración de respaldos de metadatos y archivación. En LVM2 cuando
# hablamos de 'backup' o respaldo significa hacer una copia de los metadatos para
el sistema
# *actual* system. El 'archive' contiene configuraciones de metadatos antiguas.
# Los respaldos se almacenan en formato de texto de lectura humana.
backup {

    # ¿Debemos mantener un respaldo de la configuración de metadatos actual ?
    # Use 1 para Sí; 0 para No.
    # Piense mucho antes de desactivarlo
    backup = 1

    # ¿Dónde debemos guardarlo ?
    # Recuerde hacer copia de este directorio con regularidad
    backup_dir = "/etc/lvm/backup"

    # ¿Debemos mantener un archivo de configuraciones de metadatos antiguo?.
    # Use 1 para Sí; 0 para No.
    # On predeterminado. Piense mucho antes de desactivarlo.
    archive = 1

    # ¿A dónde deben ir los ficheros archivados ?
    # Recuerde hacer copia de este directorio regularmente
    archive_dir = "/etc/lvm/archive"

    #¿Cuál es el número mínimo de ficheros de archivos que desea mantener ?
    retain_min = 10

    # ¿Cuál es el tiempo mínimo que usted desea mantener un fichero de archivos ?
    retain_days = 30
}

# Parámetros para ejecutar LVM2 en modo shell (readline).
shell {

    # Número de líneas de historia a almacenar en ~/.lvm_history
    history_size = 100
}

# Varias configuraciones de LVM2 global
global {

    # La máscara de creación de archivo para los archivo y directorios creados.

```

```
# Se interpreta como octal si el primer dígito es cero.
umask = 077

# Permitir a otros usuario la lectura de archivos
#umask = 022

# Habilitar el modo de prueba significa que no se harán cambios de metadatos
en disco
# Equivalente a tener la opción -t en cada
# comando. Se predetermina a apagado.
test = 0

# Valor predeterminado para argumento --units
units = "h"

# Puesto que la versión 2.02.54, las herramientas distinguen entre poderes de
# 1024 bytes (e.g. KiB, MiB, GiB) y poderes de 1000 bytes (e.g.
# KB, MB, GB).
# Si tiene scripts que dependen de la antigua conducta, establézcala a 0
# temporalmente hasta que las actualice.
si_unit_consistency = 1

# Si se comunica o no con el mapeador de dispositivos de kernel.
# Establezca a 0 si desea usar las herramientas para manipular metadatos de
LVM
# sin activar ningún volumen lógico.
# Si el controlador de kernel mapeador de dispositivo no está presente en su
configuración de
# kernel establecerlo a 0 suprimiría los mensajes de error.
activation = 1

# Si no podemos comunicarnos con el mapeador de dispositivos, ¿deberíamos
tratar de ejecutar
# las herramientas de LVM1?
# Esta opción solamente se aplica a kernels 2.4 y se proporciona para ayudarle
# a cambiar entre kernels de mapeador de dispositivo y kernels LVM1.
# Las herramientas LVM1 necesitan ser instaladas con .lvm1
# e.g. vgscan.lvm1 y dejarán de funcionar después de comenzar a usar
# el nuevo formato de metadatos en disco de lvm2.
# El valor predeterminado se establece cuando las herramientas se construyan.
# fallback_to_lvm1 = 0

# El formato de metadatos predeterminado que los comandos deben usar - "lvm1"
o "lvm2".
# La línea de comandos sobrescrita es -M1 o -M2.
# Se predetermina a "lvm2".
# formato = "lvm2"

# Ubicación del sistema de archivos proc
proc = "/proc"

# Tipo de cierre a usar. Se predetermina a cierre basado en archivo local (1).
# Desactívelo estableciéndolo a 0 (peligroso: riesgo de dañar metadatos
# si los comandos de LVM2 se ejecutan al mismo tiempo).
# Tipo 2 usa la biblioteca externa compartida locking_library.
# Tipo 3 usa cierre de cluster incorporado.
# Tipo 4 usa cierre de solo lectura, el cual prohíbe operaciones que podrían
# cambiar metadatos.
locking_type = 1
```

```
# Establezca a 0 para falla cuando una solicitud fail when a lock request
cannot be satisfied immediately.
wait_for_locks = 1

# Si el uso de cierre externo falla (tipo 2) y la inicialización falla,
# estableciendo a 1 se hara un intento de usar cierre de agrupamiento
# incorporado.
# Si está usando un locking_library personalizado, establézcalo a 0.
fallback_to_clustered_locking = 1

# Si se intenta inicializar tipo 2 o tipo 3 el cierre falla, quizás debido
# a componentes de cluster tales como clvmd que no se están ejecutando,
establecido a
# 1 se hará un intento para usar cierre basado en archivo local (tipo 1).
# Si esto resulta, solamente los grupos de volumen local proseguirán.
# Los Grupos de Volumen marcados como en cluster serán ignorados.
fallback_to_local_locking = 1

# El directorio local non-LV que guarda los cierres basados en archivos
mientras los comandos están en progreso
# Un directorio como /tmp puede ser borrado en el reinicio sin problema
locking_dir = "/var/lock/lvm"

# Cuando hay solicitudes de solo lectura y solo escritura compitiendo por
# metadatos de un grupo de volumen, en lugar de siempre otorgar solo lectura
# inmediatamente, los demora para permitir que las solicitudes de lectura-
escritura sean
# servidas. Sin esta configuración, el acceso a escritura puede detenerse por
un alto
# volumen de solicitudes de solo lectura.
# NB. Esta opción solo afecta cierre basado en locking_type = 1 viz. local
file-based
# locking.
prioritise_write_locks = 1

# Otras entradas puede ir aquí para permitirel cargar las bibliotecas
compartidas
# e.g. if support for LVM1 metadata was compiled as a shared library use
#   format_libraries = "liblvm2format1.so"
# Nombres de ruta completos pueden darse.

# Buscar este directorio primero para bibliotecas compartidas.
#   library_dir = "/lib"

# La biblioteca de cierre externo a cargar si locking_type se establece a 2.
#   locking_library = "liblvm2clusterlock.so"

# Tratar los errores internos como errores fatales, abortando el proceso que
# encontró el error interno. Por favor solamente habilite para depuración.
abort_on_internal_errors = 0

# Si establece a 1, ninguna operación que cambie en los metadatos de disco
será permitida.
# Además, los comandos de solo lectura que encuentren metadatos que necesitan
reparación
# aún se les permitirá proceder exactamente como si la reparación se hubiera
realizado
# (excepto para el vg_seqno sin modificar).
# El uso inapropiado puede dañar su sistema, por lo tanto busque ayuda primero
metadata_read_only = 0
```

```

}

activation {
    # Establezca a 0 to desactivar la sincronización udev (si está compilado en
    los binarios).
    # Los procesos no esperarán notificación de udev.
    # Continuarán independientemente de cualquier procesamiento udev posible
    # en el fondo. Debe usarlo solamente si udev no se está ejecutando
    # o tiene reglas que ignoran los dispositivos que crea LVM2.
    # El argumento de línea de comandos --nodevsvnc tiene prioridad sobre esta
    configuración.
    # Si establece a 1 cuando udev no está en ejecución y hay procesos LVM2
    # esperando por udev, ejecute en forma manual 'dmsetup udevcomplete_all' para
    despertarlos.
    udev_sync = 1

    # Establezca a 0 para desactivar las reglas udev instaladas por LVM2 (si se
    construye con
    # --enable-udev_rules). LVM2 administrará los /dev nodes y symlinks
    # para volúmenes lógicos activos directamente.
    # N.B. La intervención manual puede requerirse si la configuración se cambia
    # cuando alguno de los volúmenes lógicos está activo.
    udev_rules = 1

    # ¿Cómo llenar las franjas faltantes si se activa un volumen incompleto?
    # El uso de "error" hará que partes inaccesibles del dispositivo retornen
    # errores de E/S en acceso. Puede en su lugar usar una ruta de dispositivo, en
    cuyo caso
    # ese dispositivo se utilizará en lugar de las franjas faltantes.
    # Pero observar que el usar algo que no sea "error" con los volúmenes en
    espejo
    # o en instantáneas es probable que devuelva datos corruptos.
    missing_stripe_filler = "error"

    # ¿Cuánta cantidad (en KB) reservar para usar mientras los dispositivos están
    suspendidos?
    reserved_stack = 256

    # ¿Cuánta memoria (en KB) reservar para usar mientras los dispositivos están
    suspendidos?
    reserved_memory = 8192

    # Valor Nice utilizado mientras los dispositivos están suspendidos
    process_priority = -18

    # Si volume_list está definido, cada LV se activa únicamente si coincide con
    alguno en la
    # lista.
    # "vgname" y "vgname/lvname" concuerdan exactamente.
    # "@tag" coincide con cualquier etiqueta establecida en LV o VG.
    # "@*" coincide si cualquier etiqueta definida en el host también se
    establece en el volumen lógico oo en el Grupo de volumen
    #
    # volume_list = [ "vg1", "vg2/lvol1", "@tag1", "@*" ]

    # Tamaño (en KB) de cada operación de copia al copiar en espejo
    mirror_region_size = 512

    # Configuración a usar cuando no hay un valor de lectura previa almacenado en
    los metadatos.

```

```

#
# "none" - Desactivar lectura previa.
# "auto" - Usar valor predeterminado elegido por kernel.
readahead = "auto"

# 'mirror_image_fault_policy' y 'mirror_log_fault_policy' definen
# ¿Cómo se maneja una falla de un dispositivo que afecta un espejo?
# Un espejo se compone de imágenes de espejo (copias) y de un registro.
# Un registro de disco garantiza que un espejo no necesite ser resincronizado
# (todas las copias se crean iguales) cada vez que una máquina reinicia o se
cae.
#
# En el evento de una falla, la política especificada se utilizará para
determinar lo que
# sucede. Esto se aplica a reparaciones automáticas (cuando el espejo está
# monitorizado por dmeventd) y para lvconvert --repair de forma manual cuando
se
# proporciona --use-policies.
#
# "remove" - Simplemente retirar el dispositivo que no sirve y ejecutar sin
él. Si
# el dispositivo de registro falla, el espejo se convertiría para
usar
# un registro en memoria. Esto significa que el espejo no
recordará
# su estatus de sync a través de caídas y reinicios y
# todo el espejo será resincronizado. Si una imagen
# de espejo falla, el espejo se convertirá a
# un dispositivo no- mirrored si solamente queda una copia
# buena.
#
# "allocate" - Retirar el dispositivo que no sirve y tratar de asignar espacio
en
# un nuevo dispositivo para que sea un remplazo para el dispositivo
dañado.
# El uso de esta política para el registro es rápido y mantiene la
# habilidad de recordar el estado sync a través de daños y
reinicios.
# El uso de esta política para un dispositivo de espejo es lento,
como lo
# requiere el espejo para resincronizar los dispositivos, pero
# preservará la característica de espejo del dispositivo.
# Esta política actúa como "remove" si no es un dispositivo
apropiado y se
# puede asignar espacio para el remplazo.
#
# "allocate_anywhere" - Aún no implementado. Útil para colocar el dispositivo
de registro
# temporalmente en el mismo volumen físico como una de las
imágenes de
# espejo. Esta política no se recomienda para dispositivos de
espejo
# puesto que rompería la naturaleza redundante del espejo. Esta
# política actúa como "remove" si ningún dispositivo y espacio
adecuado puede ser
# asignado para el remplazo.

mirror_log_fault_policy = "allocate"
mirror_image_fault_policy = "remove"

```



```
# 'snapshot_autoextend_threshold' y 'snapshot_autoextend_percent' definen
# cómo manejar la extensión de instantánea automática. La anterior define cuándo
se
# debe extender la instantánea: cuando su espacio exceda este
# porcentaje. Lo último define cuánto espacio adicional debe asignarse para
# la instantánea, en porcentaje de su tamaño actual.
#
# Por ejemplo, si establece snapshot_autoextend_threshold a 70 y
# snapshot_autoextend_percent a 20, cuando una instantánea excede el uso de
70%,
# se extenderá a otro 20%. Para una instantánea de 1G , que usa hasta 700M
provocará
# un cambio a 1.2G. Cuando el uso excede 840M, la instantánea se
# extenderá a 1.44G, y así sucesivamente.
#
# Al configurar snapshot_autoextend_threshold a 100 inhabilita las extensiones
# automáticas. El valor mínimo es 50 (Una configuración por debajo de 50 será
tratada
# como de 50).

snapshot_autoextend_threshold = 100
snapshot_autoextend_percent = 20

# Al activar dispositivos, la E/S a dispositivos que están siendo
(re)configurados
# se suspende y para prevenir interbloqueos, LVM2 necesita mantener
# la memoria que está usando para que no sea reenviada a otra memoria. Los
grupos de páginas que
# se sabe no que son accedidos durante la activación no necesitan mantenerse
# en la memoria. Cada cadena listada en la configuración se compara con
# cada línea en /proc/self/maps, y las páginas que correspondan a cualquiera
de las líneas
# coincidentes no serán mantenidas. En algunos sistemas el archivo de
indicador regional se
# halló que constituía más del 80% de la memoria utilizada por el proceso.
# mlock_filter = [ "locale/locale-archive", "gconv/gconv-modules.cache" ]

# Establezca a 1 para revertir a la conducta predeterminada anterior a la
versión 2.02.62
# que usaba mlockall() para mantener toda la memoria del proceso durante la
activación de
# dispositivos.
use_mlockall = 0

# La monitorización se activa de forma predeterminada al activar los volúmenes
lógico.
# Establezca a 0 para desactivar la monitorización o use la opción --
ignoremonitoring.
monitorización = 1

# Cuando pvmove o lvconvert deben esperar a que el kernel termine
# de sincronizar o fusionar datos, ellos revisan y reportan el progreso
# en intervalos de este número de segundos. El predeterminado es 15 segundos.
# Si se establece a 0 y solamente hay que esperar, no hay
# reportes de progreso, pero el proceso se despierta inmediatamente cuando
# se completa la operación.
polling_interval = 15
}
```

```

#####
# Sección avanzada #
#####

# Parámetros de metadatos
#
# metadatos {
# Número predeterminado de copias de metadatos en cada volumen físico, 0, 1
o 2.
# Podría desear sobrescribirlo desde la línea de comandos con 0
# al ejecutar pvcreate en nuevos volúmenes físicos que se añaden a grandes
grupos de volúmenes.

# pvmetadatasize = 1

# El número predeterminado de copias de metadatos a mantener para cada Grupo
de Volumen.
# Si se establece a valor no-cero, LVM automáticamente elige las áreas
# disponibles a usar para cumplir con el número de copias de metadatos de GV
solicitado
# Si establece un valor mayor que
# el número total de áreas de metadatos disponibles, los metadatos se
# almacenarán todos en ellos.
# El valor predeterminado de 0 ("no administrado") desactiva esta
administración
# automática y le permite controlar las áreas de metadatos
# usadas a nivel de volumen físico individual mediante 'pvchange
# --metadatasize y/n'.

# vgmetadatasize = 0

# Tamaño predeterminado aproximado de áreas de metadatos en disco en sectores.
# Debe aumentarlo si tiene grandes grupos de volúmenes o
# si desea retener en disco una gran cantidad de historia de sus cambios de
metadatos.

# pvmetadatasize = 255

# Lista de directorios que contienen copias vivas de metadatos de formato de
texto.
# Estos directorios no deber ser volúmenes lógicos.
# Es posible usar LVM2 con un par de directorios aquí,
# preferiblemente en diferentes sistemas de archivos (non-LV), y con ningún
otro
# on-disk metadata (pvmetadatasize = 0). 0 puede ser una adición de áreas de
# metadatos en disco.
# La funcionalidad se añadía originalmente para simplificar la prueba y no
está
# soportada bajo situaciones de poca memoria - la máquina se podría bloquear.
#
# Nunca modifique a mano archivos en estos directorios a menos que
# usted esté absolutamente seguro de que sabe lo que está haciendo. Use
# la herramienta provista para hacer los cambios (por ejemplo, vgcfgrestore).

# dirs = [ "/etc/lvm/metadata", "/mnt/disk2/lvm/metadata2" ]
#}

# Demonio de eventos
#
dmeventd {

```

```
# mirror_library es la biblioteca utilizada al monitorizar un dispositivo de
espejo.
#
# "libdevmapper-event-lvm2mirror.so" intenta recuperarse de
# las fallas. Remueve los dispositivos que fallan desde un grupo de volumen y
# reconfigura un espejo cuando es necesario. Si no se proporciona ninguna
biblioteca de espejo,
#los espejos no serán monitorizados a través de dmeventd.

mirror_library = "libdevmapper-event-lvm2mirror.so"

# snapshot_library es la biblioteca utilizada para monitorizar un dispositivo
de instantánea.
#
# "libdevmapper-event-lvm2snapshot.so" monitoriza el filtraje de
# instantáneas y emite una advertencia a través de syslog cuando el uso de la
# instantánea excede 80%. La advertencia se repite cuando el 85%, 90% y
# 95% de la instantánea se llena.

snapshot_library = "libdevmapper-event-lvm2snapshot.so"

# Ruta completa del binario dmeventd.
#
# ejecutable = "/sbin/dmeventd"
}
```

Etiquetas de objetos LVM

Una etiqueta LVM es una palabra que puede ser usada para agrupar objetos LVM2 del mismo tipo. Las etiquetas pueden ser añadidas a objetos como volúmenes físicos, grupos de volúmenes y volúmenes lógicos. Las etiquetas pueden ser añadidas a los hosts en una configuración de cluster. Las instantáneas no pueden tener etiquetas.

Las etiquetas pueden ser dadas en la línea de comandos en vez de argumentos PV, VG o LV. Las etiquetas pueden llevar el prefijo @ para evitar la ambigüedad. Cada etiqueta es expandida reemplazándola con todos los objetos que tienen la etiqueta y que son del tipo esperado por su posición en la línea de comandos.

A partir del lanzamiento de Red Hat Enterprise Linux 6.1, las etiquetas de LVM son cadenas de hasta 1.024 caracteres (para lanzamientos anteriores el límite de longitud era de 128 caracteres). Las etiquetas LVM no pueden comenzar por un guión.

Una etiqueta válida puede constar de un rango limitado de caracteres únicamente. Para el lanzamiento de Red Hat Enterprise Linux 6.0, los caracteres permitidos son [A-Za-z0-9_+.-]. A partir del lanzamiento de Red Hat Enterprise Linux 6.1, la lista de caracteres permitidos se ha extendido y las etiquetas pueden contener los caracteres "/", "=", "!", ":", "#" y "&".

Solo los objetos en un grupo de volúmenes pueden tener etiquetas. Los volúmenes físicos pierden sus etiquetas si éstos se remueven del grupo de volúmenes; esto es porque las etiquetas son almacenadas como parte de los metadatos del grupo de volúmenes y que se borran cuando el volumen físico es removido. Las instantáneas no pueden tener etiquetas.

El siguiente comando lista todos los volúmenes lógicos con la etiqueta **database**.

```
lvs @database
```

C.1. Cómo añadir y retirar etiquetas de objetos

Para añadir o borrar etiquetas desde un volumen físico utilice la opción **--addtag** o **--deltag** del comando **pvchange**

Para añadir o borrar las etiquetas desde los grupos de volúmenes, utilice la opción **--addtag** o **--deltag** de los comandos **vgchange** o **vgcreate**

Para añadir o borrar etiquetas desde los volúmenes lógicos, utilice la opción **--addtag** o **--deltag** de los comandos **lvchange** o **lvcreate**.

A partir de Red Hat Enterprise Linux 6.1, usted puede especificar varios argumentos **--addtag** y **--deltag** dentro de un solo comando **pvchange**, **vgchange**, o **lvchange**. Por ejemplo, el siguiente comando borra las etiquetas **T9** y **T10** y añade las etiquetas **T13** y **T14** al grupo de volumen **grant**.

```
vgchange --deltag T9 --deltag T10 --addtag T13 --addtag T14 grant
```

C.2. Etiquetas de host

En una configuración de cluster, se pueden definir etiquetas de host en los archivos de configuración. Si establece **hosttags = 1** en la sección **tags**, una etiqueta de host es inmediatamente definida usando el nombre de host de la máquina. Esto permite utilizar un archivo de configuración común que puede ser duplicado en todas las máquinas. La etiqueta, sin embargo, será diferente en cada máquina

dependiendo del nombre de host.

Para obtener mayor información sobre los archivos de configuración, consulte el [Apéndice B, Archivos de configuración LVM](#).

Para cada etiqueta de host, se lee un archivo de configuración adicional si este existe:

`lvm_etiqueta_de_host.conf`. Si este archivo define nuevas etiquetas, más archivos de configuración serán añadidos a la lista de archivos a leer.

Por ejemplo, la siguiente entrada en el archivo de configuración siempre define **tag1** y define **tag2** si el nombre de host es **host1**.

```
tags { tag1 { } tag2 { host_list = ["host1"] } }
```

C.3. Cómo controlar la activación con etiquetas

Puede especificar en el archivo de configuración que solo ciertos volúmenes lógicos deben ser activados en el host. Por ejemplo, la siguiente entrada actúa como un filtro para la activación de solicitudes (como **vgchange -ay**) y solo activa **vg1/lvol0** y cualquier volumen lógico o grupo de volúmenes con la etiqueta **database** en los metadatos de ese host.

```
activation { volume_list = ["vg1/lvol0", "@database" ] }
```

Hay una coincidencia especial "@" que produce una coincidencia solo si la etiqueta de los metadatos coincide con alguna etiqueta de host en esa máquina.

Considere una situación en la cual cada máquina en el cluster tiene la siguiente entrada en el archivo de configuración:

```
tags { hosttags = 1 }
```

Si desea activar **vg1/lvol12** únicamente en el host **db2**, haga lo siguiente:

1. Ejecute **lvchange --addtag @db2 vg1/lvol12** desde cualquier host en el cluster.
2. Ejecute **lvchange -ay vg1/lvol12**.

En esta solución los nombres de host se almacenan en los metadatos del grupo de volúmenes.

Metadatos del grupo de volúmenes LVM

La información de la configuración de un grupo de volúmenes se conoce como metadatos. Por defecto, una copia idéntica de los metadatos se mantiene en cada área de metadatos en cada volumen físico entre el grupo de volúmenes. Los metadatos de LVM son pequeños y son almacenados en ASCII.

Si un grupo de volúmenes contiene varios volúmenes físicos, es ineficiente tener tantas copias de los metadatos. Es posible crear un volumen físico sin copias de los metadatos si se utiliza la opción `--metadaticopies 0` del comando `pvcreate`. El número de copias de los metadatos que el volumen físico contendrá no puede ser modificada una vez hecha la selección inicial. Si se selecciona no tener ninguna copia, la actualización de la configuración será más rápida. Note, sin embargo, que cada grupo de volúmenes debe contener al menos un volumen físico con los metadatos (a menos que esté usando los parámetros de configuración avanzada que permiten almacenar los metadatos del grupo de volúmenes en un sistema de archivos). Si desea dividir el grupo de volúmenes en el futuro, cada grupo necesita al menos una copia de los metadatos.

Los metadatos del núcleo se almacenan en ASCII. Un área de metadatos es un buffer circular. Los nuevos metadatos son añadidos a los datos anteriores y luego el puntero al inicio de los datos se actualiza.

Puede especificar el tamaño de los metadatos con la opción `--metadatasize` del comando `pvcreate`. El tamaño por defecto es muy pequeño para grupos de volúmenes con numerosos volúmenes lógicos o volúmenes físicos.

D.1. La etiqueta del volumen físico

Por defecto, el comando `pvcreate` pone la etiqueta del volumen físico en el segundo sector de 512-byte. Esta etiqueta puede ser ubicada en cualquiera de los primero cuatro sectores, ya que las herramientas de LVM que exploran esta etiqueta revisan los primeros cuatro sectores. La etiqueta de volúmenes físicos inicia con la cadena **LABELONE**.

La etiqueta de volúmenes físicos contiene:

- ▶ UUID del volumen físico
- ▶ Tamaño del dispositivo del bloque en bytes
- ▶ Lista de ubicaciones de áreas de datos terminadas en NULL
- ▶ Listas de ubicaciones de área de metadatos terminadas en NULL

La ubicación de los metadatos se almacena como registros y tamaños (en bytes). En la etiqueta hay espacio para 15 ubicaciones, pero las herramientas LVM utilizan solo 3: un área de datos única y dos áreas de metadatos.

D.2. Contenido de metadatos

Los metadatos del grupo de volúmenes contienen:

- ▶ Información sobre cómo y cuándo fue creado
- ▶ Información sobre el grupo de volúmenes:

La información del grupo de volúmenes contiene:

- ▶ Nombre y ID único
- ▶ Un número de versión que se incrementa cada vez que los metadatos son actualizados

- ▶ Cualquier propiedad: ¿Lectura/escritura? ¿Dimensionable?
- ▶ Cualquier límite administrativo sobre el número de volúmenes físicos y volúmenes lógicos que puede contener
- ▶ El tamaño de las extensiones (en unidades de sectores que tienen 512 bytes)
- ▶ Una lista sin orden de los volúmenes físicos que conforman el grupo de volúmenes, cada cual con:
 - Su UUID, usado para determinar el dispositivo de bloque que lo contiene
 - Cualquier propiedad, por ejemplo si el volumen físico es asignable
 - El lugar del comienzo de la primera extensión del volumen físico (en sectores)
 - El número de extensiones
- ▶ Una lista sin orden de los volúmenes lógicos. Cada uno con
 - Una lista ordenada de los segmentos del volumen lógico. Para cada segmento los metadatos incluyen una relación aplicada a un alista ordenada de los segmentos del volumen físico o los segmentos del volumen lógico.

D.3. Metadatos de ejemplo

El siguiente muestra un ejemplo de los metadatos de un grupo de volúmenes LVM para un grupo llamado **myvg**.

```
# Generated by LVM2: Tue Jan 30 16:28:15 2007

contents = "Text Format Volume Group"
version = 1

description = "Created *before* executing 'lvextend -L+5G /dev/myvg/mylv
/dev/sdc'"

creation_host = "tng3-1"           # Linux tng3-1 2.6.18-8.el5 #1 SMP Fri Jan 26
14:15:21 EST 2007 i686
creation_time = 1170196095        # Tue Jan 30 16:28:15 2007

myvg {
  id = "0zd3UT-wbYT-lDHq-lMPs-EjoE-0o18-wL28X4"
  seqno = 3
  status = ["RESIZEABLE", "READ", "WRITE"]
  extent_size = 8192              # 4 Megabytes
  max_lv = 0
  max_pv = 0

  physical_volumes {

    pv0 {
      id = "ZBW5qW-dXF2-0bGw-ZCad-2RlV-phwu-1c1RFt"
      device = "/dev/sda"         # Hint only

      status = ["ALLOCATABLE"]
      dev_size = 35964301        # 17.1491 Gigabytes
      pe_start = 384
      pe_count = 4390 # 17.1484 Gigabytes
    }

    pv1 {
      id = "ZHEZJW-MR64-D3QM-Rv7V-Hxsa-zU24-wztY19"
      device = "/dev/sdb"         # Hint only

      status = ["ALLOCATABLE"]
      dev_size = 35964301        # 17.1491 Gigabytes
      pe_start = 384
      pe_count = 4390 # 17.1484 Gigabytes
    }

    pv2 {
      id = "wCoG4p-55Ui-9tbp-VTEA-j06s-RAVx-UREW0G"
      device = "/dev/sdc"         # Hint only

      status = ["ALLOCATABLE"]
      dev_size = 35964301        # 17.1491 Gigabytes
      pe_start = 384
      pe_count = 4390 # 17.1484 Gigabytes
    }

    pv3 {
      id = "hGlUwi-zsBg-39FF-do88-pHxY-8XA2-9WKIiA"
      device = "/dev/sdd"         # Hint only

      status = ["ALLOCATABLE"]
      dev_size = 35964301        # 17.1491 Gigabytes
      pe_start = 384
      pe_count = 4390 # 17.1484 Gigabytes
    }
  }
}
```



```
    }
  }
  logical_volumes {
    mylv {
      id = "GhUYSF-qVM3-rzQo-a6D2-o0aV-LQet-Ur90F9"
      status = ["READ", "WRITE", "VISIBLE"]
      segment_count = 2

      segment1 {
        start_extent = 0
        extent_count = 1280      # 5 Gigabytes

        type = "striped"
        stripe_count = 1        # linear

        stripes = [
          "pv0", 0
        ]
      }
      segment2 {
        start_extent = 1280
        extent_count = 1280      # 5 Gigabytes

        type = "striped"
        stripe_count = 1        # linear

        stripes = [
          "pv1", 0
        ]
      }
    }
  }
}
```

Historial de revisiones

Revisión 1-6.400	2013-10-31	Rüdiger Landmann
Rebuild with publican 4.0.0		
Revisión 1-6	2012-07-18	Anthony Towns
Rebuild for Publican 3.0		
Revisión 2.0-1	Thu May 19 2011	Steven Levine
Lanzamiento inicial de Red Hat Enterprise Linux 6.1		
Resuelve: #694619 Documenta nueva política de asignación cling cuando se extiende un volumen lógico.		
Resuelve: #682649 Añade una advertencia sobre la ejecución de comandos de creación múltiple en espejo en sucesión en los volúmenes de cluster.		
Resuelve: #674100 Añade ejemplo de salida para el comando dmsetup ls --tree .		
Resuelve: #694607 Documenta soporte para incluir múltiples argumentos --addtag and --deltag en una sola línea de comandos.		
Resuelve: #694604 Documenta soporte para la lista de caracteres extendidos en etiquetas.		
Resuelve: #694611 Documenta soporte para espejos en franjas.		
Resuelve: #694616 Documenta soporte para instantáneas de volúmenes en espejo.		
Resuelve: #694618 Documenta soporte para instantáneas de volúmenes de cluster exclusivamente activados.		
Resuelve: #682648 Documenta que cuando un pilar de espejo se reasigna, el registro de espejo también puede ser desplazado.		
Resuelve: #661530 Actualiza ejemplo cluster.conf para uno que documente funcionalidades actuales.		
Resuelve: #642400 Añade nota sobre el manejo de registro de cluster que es mantenido por el nodo de cluster con el ID de cluster más bajo.		
Resuelve: #663462 Remueve datos desactualizados para monitor de máquina virtual Xen.		

Índice

Símbolos

`/lib/udev/rules.d` directory, [integración udev con el Mapeador de Dispositivos](#)

A

activación de grupo de volúmenes, [Cómo activar y desactivar grupos de volúmenes](#)

- nodos individuales, [Cómo activar y desactivar grupos de volúmenes](#)

activación de grupos de volúmenes

- únicamente nodo local, [Cómo activar y desactivar grupos de volúmenes](#)

activación de volúmenes lógicos

- nodos individuales, [Cómo activar volúmenes lógicos en nodos individuales en un cluster](#)

archivo `archive`, [Cómo crear copias de seguridad de los metadatos del grupo de volúmenes](#)

archivo `backup`, [Cómo crear copias de seguridad de los metadatos del grupo de volúmenes](#)

archivo `cache`

- crear, [Cómo explorar discos en búsqueda de grupos de volúmenes para construir el archivo de caché](#)

archivo de archivador, [Copia de seguridad del volumen lógico](#)

asignación

- política, [Cómo crear grupos de volúmenes](#)

- prevención, [Cómo prevenir la asignación en un volumen físico](#)

asignación de datos, en línea, [Asignación de datos en línea](#)

C

CLVM

- definición, [Administrador de volúmenes lógicos en cluster \(CLVM\)](#)

comando `lvchange`, [Cómo cambiar los parámetros de un grupo de volúmenes lógico](#)

comando `lvconvert`, [Cómo cambiar la configuración del volumen en espejo](#)

comando `lvcreate`, [Cómo crear volúmenes lógicos lineales](#)

comando `lvdisplay`, [Cómo ver los volúmenes lógicos](#)

comando `lvextend`, [Cómo aumentar los volúmenes lógicos](#)

comando lvmdiskscan, [Cómo buscar Dispositivos de Bloque](#)

comando lvreduce, [Cómo redimensionar volúmenes lógicos](#), [Cómo reducir volúmenes lógicos](#)

comando lvremove, [Cómo retirar volúmenes lógicos](#)

comando lvrename, [Cómo renombrar volúmenes lógicos](#)

comando lvs, [Reporte personalizado para LVM](#), [El comando lvs](#)

- argumentos de visualización, [El comando lvs](#)

comando lvscan, [Cómo ver los volúmenes lógicos](#)

comando pvdisplay, [Cómo ver los volúmenes físicos](#)

comando pvmove, [Asignación de datos en línea](#)

comando pvremove, [Cómo retirar volúmenes físicos](#)

comando pvresize, [Cómo redimensionar un volumen físico](#)

comando pvs, [Reporte personalizado para LVM](#)

- argumentos de visualización, [El comando pvs](#)

comando pvscan, [Cómo ver los volúmenes físicos](#)

comando vgcfsbackup, [Cómo crear copias de seguridad de los metadatos del grupo de volúmenes](#)

comando vgcfsrestore, [Cómo crear copias de seguridad de los metadatos del grupo de volúmenes](#)

comando vgchange, [Cómo cambiar los parámetros de un grupo de volúmenes](#)

comando vgcreate, [Cómo crear grupos de volúmenes](#), [Cómo crear grupos de volúmenes en un cluster](#)

comando vgdisplay, [Cómo mostrar los grupos de volúmenes](#)

comando vgexport, [Cómo desplazar un grupo de volúmenes a otro sistema](#)

comando vgextend, [Cómo agregar volúmenes físicos al grupo de volúmenes](#)

comando vgimport, [Cómo desplazar un grupo de volúmenes a otro sistema](#)

comando vgmerge, [Cómo combinar grupos de volúmenes](#)

comando vgmknodes, [Cómo recrear un directorio de grupo de volúmenes](#)

comando vgreduce, [Cómo retirar un volumen físico de un grupo de volúmenes](#)

comando vgrename, [Cómo cambiar el nombrar de un grupo de volúmenes](#)

comando vgs, [Reporte personalizado para LVM](#)

- muestra argumentos, [El comando vgs](#)

comando vgscan, [Cómo explorar discos en búsqueda de grupos de volúmenes para construir el archivo de caché](#)

comando vgsplit, [Separación de un grupo de volúmenes](#)

copia de seguridad

- archivo, [Copia de seguridad del volumen lógico](#)

- metadatos, [Copia de seguridad del volumen lógico](#), [Cómo crear copias de seguridad de](#)

[los metadatos del grupo de volúmenes](#)

creación

- ejemplo, volumen lógico, [Cómo crear un volumen lógico LVM en tres discos](#)
- grupo de volúmenes, en cluster, [Cómo crear grupos de volúmenes en un cluster](#)
- grupos de volúmenes, [Cómo crear grupos de volúmenes](#)
- volumen lógico seccionado, ejemplo, [Cómo crear un volumen lógico entrelazado](#)
- volúmenes lógicos, [Cómo crear volúmenes lógicos lineales](#)

crear

- volúmenes físicos, [Creación de volúmenes físicos](#)
- volúmenes LVM en un cluster, [Creación de volúmenes LVM en un cluster](#)

crear volúmenes

- visión general, [Sinopsis de la creación de volúmenes lógicos](#)

creciendo en un volumen lógico

- volumen lógico, [Cómo aumentar el tamaño de un sistema de archivos en un volumen lógico](#)

D

demonio clvmd, [Administrador de volúmenes lógicos en cluster \(CLVM\)](#)

desactivación de grupo de volúmenes, [Cómo activar y desactivar grupos de volúmenes](#)

desactivación de grupos de volúmenes

- exclusivo en un nodo, [Cómo activar y desactivar grupos de volúmenes](#)
- únicamente en nodo local, [Cómo activar y desactivar grupos de volúmenes](#)

directorio de archivo especial de dispositivo, [Cómo crear grupos de volúmenes](#)

dispositivo de bloque

- escanear, [Cómo buscar Dispositivos de Bloque](#)

dispositivos fallidos

- visualización, [Cómo ver la información en dispositivos fallidos](#)

E

ejemplos de configuración, [Ejemplos de configuración de LVM](#)

entorno de cluster, [Administrador de volúmenes lógicos en cluster \(CLVM\)](#), [Creación de volúmenes LVM en un cluster](#)

escanear

- dispositivos de bloque, [Cómo buscar Dispositivos de Bloque](#)

escanear dispositivos, filtros, [Cómo controlar la exploración de dispositivos LVM a través de filtros](#)

extensión

- asignación, [Cómo crear grupos de volúmenes](#)
- definición, [Grupos de volúmenes](#), [Cómo crear grupos de volúmenes](#)

extensión física

- prevenir asignación, [Cómo prevenir la asignación en un volumen físico](#)

F

feedback

- contact information for this manual, [We Need Feedback!](#)

filtros, [Cómo controlar la exploración de dispositivos LVM a través de filtros](#)

filtros de escanear dispositivos, [Cómo controlar la exploración de dispositivos LVM a través de filtros](#)

formato de informe, dispositivos LVM, [Reporte personalizado para LVM](#)

funcionalidades, nuevas y cambiadas, [Funcionalidades nuevas y cambiadas](#)

G

gestor de dispositivos udev, [Soporte de mapeador de dispositivo para el Gestor de dispositivos udev](#)

grupo de volúmenes

- activación, [Cómo activar y desactivar grupos de volúmenes](#)
- administración, general, [Administración del grupo de volúmenes](#)
- cambio de parámetros, [Cómo cambiar los parámetros de un grupo de volúmenes](#)
- combinación, [Cómo combinar grupos de volúmenes](#)
- creación, [Cómo crear grupos de volúmenes](#)
- creación en un cluster, [Cómo crear grupos de volúmenes en un cluster](#)
- crecimiento, [Cómo agregar volúmenes físicos al grupo de volúmenes](#)
- definición, [Grupos de volúmenes](#)
- desactivación, [Cómo activar y desactivar grupos de volúmenes](#)
- desplazamiento entre sistemas, [Cómo desplazar un grupo de volúmenes a otro sistema](#)
- dividiendo
 - procedimiento de ejemplo, [Separación de un grupo de volúmenes](#)
- división, [Separación de un grupo de volúmenes](#)
- encogimiento, [Cómo retirar un volumen físico de un grupo de volúmenes](#)
- extensión, [Cómo agregar volúmenes físicos al grupo de volúmenes](#)
- fusión, [Cómo combinar grupos de volúmenes](#)
- reducción, [Cómo retirar un volumen físico de un grupo de volúmenes](#)
- remoción, [Cómo retirar grupos de volúmenes](#)
- renombrar, [Cómo cambiar el nombrar de un grupo de volúmenes](#)
- vgs argumentos de visualización, [El comando vgs](#)
- visualización, [Cómo mostrar los grupos de volúmenes](#), [Reporte personalizado para LVM](#), [El comando vgs](#)

I

inicializar

- particiones, [Cómo inicializar los volúmenes físicos](#)
- volúmenes físicos, [Cómo inicializar los volúmenes físicos](#)

L

LVM

- administración de volumen físico, [Administración de volúmenes físicos](#)
- administración de volumen lógico, [Administración del volumen lógico](#)
- ayuda, [Uso de los comandos](#)
- componentes, [Sinopsis de la arquitectura LVM](#), [Componentes de LVM](#)
- en cluster, [Administrador de volúmenes lógicos en cluster \(CLVM\)](#)
- estructura de directorio, [Cómo crear grupos de volúmenes](#)
- etiqueta, [Volúmenes físicos](#)
- grupo de volúmenes, definición, [Grupos de volúmenes](#)
- historia, [Sinopsis de la arquitectura LVM](#)
- personalizar formato de informe, [Reporte personalizado para LVM](#)
- registro, [Registro](#)
- visión general de arquitectura, [Sinopsis de la arquitectura LVM](#)
- volumen físico, definición, [Volúmenes físicos](#)

LVM1, [Sinopsis de la arquitectura LVM](#)

LVM2, [Sinopsis de la arquitectura LVM](#)

M

Mensaje de extensiones libres insuficientes, [Extensiones libres insuficientes para un volumen lógico](#)

metadatos

- copia de seguridad, [Copia de seguridad del volumen lógico](#), [Cómo crear copias de seguridad de los metadatos del grupo de volúmenes](#)
- recuperación, [Cómo recuperar los metadatos de un volumen físico](#)

mostrar

- volúmenes físicos, [Cómo ver los volúmenes físicos](#), [El comando pvs](#)
- volúmenes lógicos, [Cómo ver los volúmenes lógicos](#), [El comando lvs](#)

N

nombres de ruta, [Uso de los comandos](#)

nombres de ruta de dispositivo, [Uso de los comandos](#)

números de dispositivo persistentes, [Números de dispositivos persistentes](#)

números de dispositivos

- mayores, [Números de dispositivos persistentes](#)

- menores, [Números de dispositivos persistentes](#)
- persistentes, [Números de dispositivos persistentes](#)

P

pantalla de ayuda, [Uso de los comandos](#)

pantalla de página man, [Uso de los comandos](#)

parámetro de configuración `mirror_image_fault_policy`, [Política de fallas de volumen lógico en espejo](#)

parámetro de configuración `mirror_log_fault_policy`, [Política de fallas de volumen lógico en espejo](#)

particiones

- múltiples, [Múltiples particiones en un disco](#)

procedimientos administrativos, [Visión General de la Gestión de LVM](#)

R

reasignación de datos en línea, [Asignación de datos en línea](#)

redimensionar

- volumen físico, [Cómo redimensionar un volumen físico](#)
- volumen lógico, [Cómo redimensionar volúmenes lógicos](#)

registro, [Registro](#)

remoción

- volumen lógico, [Cómo retirar volúmenes lógicos](#)
- volúmenes físicos, [Cómo retirar volúmenes físicos](#)

renombrar

- grupo de volúmenes, [Cómo cambiar el nombrar de un grupo de volúmenes](#)
- volumen lógico, [Cómo renombrar volúmenes lógicos](#)

Retirar

- disco desde el volumen lógico, [Cómo retirar un disco de un volumen lógico](#)

rules.d directory, [integración udev con el Mapeador de Dispositivos](#)

S

salida verbosa, [Uso de los comandos](#)

sistema de archivos

- creciendo en un volumen lógico, [Cómo aumentar el tamaño de un sistema de archivos en un volumen lógico](#)

solución de problemas, [Solución de errores en LVM](#)

T

- tamaño de dispositivo, máximo, [Cómo crear grupos de volúmenes](#)
- tipo de partición, configuración, [Cómo establecer el tipo de partición](#)

U

- udev rules, [integración udev con el Mapeador de Dispositivos](#)
- unidades de línea de comandos, [Uso de los comandos](#)
- unidades, línea de comandos, [Uso de los comandos](#)

V

visión general

- funcionalidades, nuevas y cambiadas, [Funcionalidades nuevas y cambiadas](#)

visualización

- clasificación de salida, [Cómo ordenar los reportes de LVM](#)
- grupos de volúmenes, [Cómo mostrar los grupos de volúmenes](#), [El comando vgs](#)

volumen de instantánea

- definición, [Volúmenes de instantáneas](#)

volumen físico

- adición de un grupo de volúmenes, [Cómo agregar volúmenes físicos al grupo de volúmenes](#)
- administración, general, [Administración de volúmenes físicos](#)
- crear, [Creación de volúmenes físicos](#)
- definición, [Volúmenes físicos](#)
- distribución, [Distribución de un volumen físico LVM](#)
- ilustración, [Distribución de un volumen físico LVM](#)
- inicialización, [Cómo inicializar los volúmenes físicos](#)
- mostrar, [Cómo ver los volúmenes físicos](#), [Reporte personalizado para LVM](#)
- pvsargumentos de visualización, [El comando pvs](#)
- recuperación, [Cómo reemplazar un volumen físico perdido](#)
- redimensionar, [Cómo redimensionar un volumen físico](#)
- remoción, [Cómo retirar volúmenes físicos](#)
- remoción de volumen perdido, [Cómo retirar los volúmenes físicos perdidos desde un grupo de volúmenes](#)
- remoción del grupo de volúmenes, [Cómo retirar un volumen físico de un grupo de volúmenes](#)
- visualización, [El comando pvs](#)

volumen lógica

- definición, [Volúmenes lógicos](#), [Volúmenes lógicos LVM](#)

volumen lógica entrelazado

- definición, [Volúmenes lógicos entrelazados](#)

volumen lógico

- acceso exclusivo, [Cómo activar volúmenes lógicos en nodos individuales en un cluster](#)
- acceso local, [Cómo activar volúmenes lógicos en nodos individuales en un cluster](#)
- administración, general, [Administración del volumen lógico](#)
- cambio de parámetros, [Cómo cambiar los parámetros de un grupo de volúmenes lógico](#)
- creación, [Cómo crear volúmenes lógicos lineales](#)
- crecimiento, [Cómo aumentar los volúmenes lógicos](#)
- ejemplo de creación, [Cómo crear un volumen lógico LVM en tres discos](#)
- en espejo, [Cómo crear volúmenes en espejo](#)
- encogimiento, [Cómo reducir volúmenes lógicos](#)
- entrelazado, [Cómo crear volúmenes entrelazados](#)
- extensión, [Cómo aumentar los volúmenes lógicos](#)
- instantánea, [Cómo crear volúmenes de instantáneas](#)
- lineal, [Cómo crear volúmenes lógicos lineales](#)
- lvs argumentos de visualización, [El comando lvs](#)
- mostrar, [Cómo ver los volúmenes lógicos](#), [Reporte personalizado para LVM](#), [El comando lvs](#)
- redimensionar, [Cómo redimensionar volúmenes lógicos](#)
- reducción, [Cómo reducir volúmenes lógicos](#)
- remoción, [Cómo retirar volúmenes lógicos](#)
- renombre, [Cómo renombrar volúmenes lógicos](#)

volumen lógico de instantánea

- creación, [Cómo crear volúmenes de instantáneas](#)

volumen lógico en espejo

- agrupado, [Cómo crear un volumen lógico LVM en un cluster](#)
- conversión a lineal, [Cómo cambiar la configuración del volumen en espejo](#)
- creación, [Cómo crear volúmenes en espejo](#)
- definición, [Volúmenes lógicos en espejo](#)
- política de fallas, [Política de fallas de volumen lógico en espejo](#)
- reconfiguración, [Cómo cambiar la configuración del volumen en espejo](#)
- recuperación de fallas, [Cómo recuperarse de una falla de un espejo LVM](#)

volumen lógico entrelazado

- creación, [Cómo crear volúmenes entrelazados](#)

volumen lógico lineal

- conversión a espejo, [Cómo cambiar la configuración del volumen en espejo](#)
- creación, [Cómo crear volúmenes lógicos lineales](#)
- definición, [Volúmenes lineales](#)

volumen lógico seccionado

- crecimiento, [Cómo extender un volumen entrelazado](#)
- ejemplo de creación, [Cómo crear un volumen lógico entrelazado](#)
- extensión, [Cómo extender un volumen entrelazado](#)

