

Red Hat Enterprise Linux 5

Administrador de volumen lógico de cluster

LVM Administrator's Guide



Red Hat Enterprise Linux 5 Administrador de volumen lógico de cluster

LVM Administrator's Guide

Edición 3

Copyright © 2009 Red Hat Inc..

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at <http://creativecommons.org/licenses/by-sa/3.0/>. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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.

All other trademarks are the property of their respective owners.

1801 Varsity Drive
Raleigh, NC 27606-2072 USA
Phone: +1 919 754 3700
Phone: 888 733 4281
Fax: +1 919 754 3701

Este libro describe el administrador de volúmenes lógicos LVM, incluyendo información sobre cómo ejecutar LVM en entornos de cluster. El contenido de este documento es específico para el lanzamiento LVM2.

Introducción	vii
1. Sobre este manual	vii
2. Audiencia	vii
3. Versiones de software	vii
4. Documentación relacionada	vii
5. Comentarios	viii
6. Convenciones del Documento	viii
6.1. Convenciones Tipográficas	viii
6.2. Convenciones del documento	x
6.3. Notas y Advertencias	x
1. El administrador de volúmenes lógicos LVM	1
1.1. Volúmenes lógicos	1
1.2. Sinopsis de la arquitectura LVM	2
1.3. Administrador de volúmenes lógicos en cluster (CLVM)	3
1.4. Sinopsis del documento	4
2. Componentes de LVM	7
2.1. Volúmenes físicos	7
2.1.1. LVM Physical Volume Layout	7
2.1.2. Múltiples particiones en un disco	8
2.2. Grupos de volúmenes	8
2.3. Volúmenes lógicos LVM	9
2.3.1. Volúmenes lineales	9
2.3.2. Volúmenes lógicos entrelazados	10
2.3.3. Volúmenes lógicos en espejo	11
2.3.4. Volúmenes de instantáneas	12
3. Sinopsis de la administración de LVM	15
3.1. Creación de volúmenes LVM en un cluster	15
3.2. Sinopsis de la creación de volúmenes lógicos	15
3.3. Cómo aumentar el tamaño de un sistema de archivos en un volumen lógico	16
3.4. Copia de seguridad del volumen lógico	16
3.5. Registro	17
4. Administración de LVM mediante comandos de CLI	19
4.1. Uso de los comandos	19
4.2. Administración de volúmenes físicos	20
4.2.1. Creación de volúmenes físicos	20
4.2.2. Cómo ver los volúmenes físicos	22
4.2.3. Cómo prevenir la asignación en un volumen físico	22
4.2.4. Cómo cambiar el tamaño de un volumen físico	23
4.2.5. Cómo remover volúmenes físicos	23
4.3. Administración del grupo de volúmenes	23
4.3.1. Cómo crear grupos de volúmenes	23
4.3.2. Cómo crear grupos de volúmenes en un cluster	24
4.3.3. Cómo agregar volúmenes físicos al grupo de volúmenes	25
4.3.4. Cómo mostrar los grupos de volúmenes	25
4.3.5. Cómo explorar discos en búsqueda de grupos de volúmenes para construir el archivo de caché	26
4.3.6. Cómo remover un volumen físico de un grupo de volúmenes	26
4.3.7. Cómo cambiar los parámetros de un grupo de volúmenes	27
4.3.8. Cómo activar y desactivar grupos de volúmenes	27
4.3.9. Cómo remover grupos de volúmenes	27
4.3.10. Cómo separar un grupo de volúmenes	28
4.3.11. Cómo combinar grupos de volúmenes	28

4.3.12. Cómo crear copias de seguridad de los metadatos del grupo de volúmenes.....	28
4.3.13. Cómo cambiar el nombrar de un grupo de volúmenes	28
4.3.14. Cómo mover un grupo de volúmenes a otro sistema	29
4.3.15. Cómo recrear un directorio de grupo de volúmenes	29
4.4. Administración del volumen lógico	30
4.4.1. Cómo crear volúmenes lógicos	30
4.4.2. Números de dispositivos persistentes	34
4.4.3. Cómo dimensionar volúmenes lógicos	34
4.4.4. Cómo cambiar los parámetros de un grupo de volúmenes lógico	34
4.4.5. Cómo renombrar volúmenes lógicos	35
4.4.6. Cómo remover volúmenes lógicos	35
4.4.7. Cómo ver los volúmenes lógicos	35
4.4.8. Cómo aumentar los volúmenes lógicos	36
4.4.9. Cómo extender un volumen entrelazado	37
4.4.10. Cómo reducir volúmenes lógicos	38
4.5. Cómo crear volúmenes de instantáneas	38
4.6. Cómo controlar la exploración de dispositivos LVM a través de filtros	40
4.7. Asignación de datos en línea	41
4.8. Cómo activar volúmenes lógicos en nodos individuales en un cluster	41
4.9. Reporte personalizado para LVM	42
4.9.1. Control del formato	42
4.9.2. Selección de objetos	44
4.9.3. Cómo ordenar los reportes de LVM	49
4.9.4. Cómo especificar unidades	50
5. Ejemplos de configuración de LVM	53
5.1. Cómo crear un volumen lógico LVM en tres discos	53
5.1.1. Cómo crear los volúmenes físicos	53
5.1.2. Cómo crear el grupo de volúmenes	53
5.1.3. Cómo crear el volumen lógico	53
5.1.4. Cómo crear un sistema de archivos	53
5.2. Cómo crear un volumen lógico entrelazado	54
5.2.1. Cómo crear los volúmenes físicos	54
5.2.2. Cómo crear el grupo de volúmenes	54
5.2.3. Cómo crear el volumen lógico	55
5.2.4. Cómo crear un sistema de archivos	55
5.3. Separación de un grupo de volúmenes	55
5.3.1. Cómo determinar el espacio libre	56
5.3.2. Cómo desplazar los datos	56
5.3.3. Cómo dividir en grupo de volúmenes	56
5.3.4. Cómo crear un nuevo volumen lógico	57
5.3.5. Cómo crear el sistema de archivos y cómo montar el nuevo volumen lógico	57
5.3.6. Cómo activar y montar el volumen lógico original	57
5.4. Cómo remover un disco de un volumen lógico	58
5.4.1. Cómo mover las extensiones a un volumen físico existente	58
5.4.2. Cómo mover extensiones a un nuevo disco	59
6. Solución de errores en LVM	61
6.1. Diagnósticos en la solución de errores	61
6.2. Cómo ver la información en dispositivos fallidos	61
6.3. Cómo recuperarse de una falla de un espejo LVM	62
6.4. Cómo recuperar los metadatos de un volumen físico	65
6.5. Cómo remplazar un volumen físico perdido	66
6.6. Cómo remover los volúmenes físicos perdidos desde un grupo de volúmenes	67
6.7. Extensiones libres insuficientes para un volumen lógico	67

7. Administración de LVM con la interfaz gráfica de LVM	69
A. Mapeo de Dispositivos	71
A.1. Tabla de Mapas de Dispositivo	71
A.1.1. Destino de mapa lineal	72
A.1.2. Destino de mapas entrelazados	72
A.1.3. Espejo de destino de mapa	74
A.1.4. Destinos de mapa instantánea e instantánea-origen	76
A.1.5. Destino de mapa error	77
A.1.6. Destino de mapas cero	78
A.1.7. Destino de mapas multirutas	78
A.1.8. Destino de mapas crypt	80
A.2. Comando dmsetup	81
A.2.1. Comando info dmsetup	81
A.2.2. El comando dmsetup ls	83
A.2.3. Comando dmsetup status	83
A.2.4. Comando dmsetup deps	83
B. Archivos de configuración LVM	85
B.1. Archivos de configuración LVM	85
B.2. Archivo lvm.conf de ejemplo	85
C. Etiquetas de objetos LVM	93
C.1. Cómo añadir y remover etiquetas de objetos	93
C.2. Etiquetas de host	93
C.3. Cómo controlar la activación con etiquetas	94
D. Metadatos del grupo de volúmenes LVM	95
D.1. La etiqueta del volumen físico	95
D.2. Contenido de metadatos	95
D.3. Metadatos de ejemplo	96
E. Historial de revisión	99
Índice	101

Introducción

1. Sobre este manual

Este libro describe el administrador de volúmenes lógicos LVM, incluyendo información sobre cómo ejecutar LVM en un entorno de cluster. El contenido de este documento es específico para el lanzamiento LVM2.

2. Audiencia

Este libro está dirigido a administradores de sistemas que manejan sistemas con el sistema operativo Linux. Requiere familiaridad con Red Hat Enterprise Linux 5 y la administración de sistemas de archivos GFS.

3. Versiones de software

Tabla 1. Versiones de software

Software	Descripción
RHEL5	referencia a RHEL5 o superior
GFS	referencia a GFS para RHEL5 o superior

4. Documentación relacionada

Para mayor información sobre el uso de Red Hat Enterprise Linux, consulte los siguientes recursos:

- *Manual de instalación de Red Hat Enterprise Linux* — proporciona información sobre la instalación de Red Hat Enterprise Linux 5.
- *Manual de implementación de Red Hat Enterprise Linux* — Proporciona información sobre la implementación, configuración y administración de Red Hat Enterprise Linux 5.

Para obtener mayor información sobre el uso de Red Hat Cluster Suite para Red Hat Enterprise Linux 5, consulte los siguientes recursos:

- *Sinopsis de Red Hat Cluster Suite* — Proporciona un resumen de alto nivel de la suite de cluster de Red Hat.
- *Configuración y administración de Red Hat Cluster* — Proporciona información sobre la instalación, configuración y administración de los componentes de Red Hat Cluster.
- *Sistema de archivos global (GFS) Configuración y administración* — Proporciona información sobre la instalación, configuración y mantenimiento de Red Hat GFS (Red Hat Global File System).
- *Sistema de archivos global 2: Configuración y administración* — Proporciona información sobre la instalación, configuración y mantenimiento de Red Hat GFS (Red Hat Global File System).
- *Uso de multirutas de Mapeo de Dispositivos* — Proporciona información sobre el uso del soporte de multirutas de mapeo de dispositivos de Red Hat Enterprise Linux 5.
- *Uso de GNBD con el sistema de archivos global* — Proporciona un resumen sobre el uso de Dispositivos de Bloque de Red Global (GNBD) con Red Hat GFS.

- *Administración del servidor virtual de Linux* — Proporciona información sobre cómo configurar sistemas de alto rendimiento y servicios con el servidor virtual de Linux (LVS).
- *Notas de lanzamiento de Red Hat Cluster Suite* — Proporciona información sobre el lanzamiento actual de la suite de cluster de Red Hat.

La documentación para Red Hat Cluster Suite y otros documentos de Red Hat están disponibles en HTML, PDF y RPM en el CD de documentación de Red Hat Enterprise Linux y en <http://www.redhat.com/docs/>.

5. Comentarios

Si encuentra algún error o si tiene sugerencias para mejorar este documento, nos gustaría escuchar su opinión. Por favor complete un reporte en Bugzilla (<http://bugzilla.redhat.com/bugzilla/>) usando el componente **rh-cs**.

Be sure to mention the manual's identifier:

```
Bugzilla component: Documentation-cluster
Book identifier: Cluster_Logical_Volume_Manager(EN)-5 (2009-01-05T15:20)
```

By mentioning this manual's identifier, we know exactly which version of the guide you have.

Si tiene una sugerencia para mejorar la documentación trate de ser tan específico como le sea posible. Si encontró algún error, incluya el número de la sección y parte del texto que rodea el error. Esto ayudará a localizar el error más fácilmente.

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](https://fedorahosted.org/liberation-fonts/)¹. 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

Utilizada para resaltar la entrada del sistema, incluyendo comandos de shell, nombres de archivo y rutas. También se utiliza para resaltar teclas claves 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 clave. Todo se presenta en negrita-monoespaciado y distinguible gracias al contexto.

¹ <https://fedorahosted.org/liberation-fonts/>

Las combinaciones de teclas se pueden distinguir de las teclas claves mediante el guión que conecta cada parte de una combinación de tecla. Por ejemplo:

Pulse **Enter** para ejecutar el comando.

Pulse **Control+Alt+F2** para cambiar a la primera terminal virtual. Pulse **Control+Alt+F1** para volver a su sesión de Ventanas-X.

La primera oración resalta la tecla clave determinada que se debe pulsar. La segunda resalta dos conjuntos de tres teclas claves que deben ser presionadas 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 sub-menú. Por ejemplo:

Seleccionar **Sistema** → **Preferencias** → **Ratón** desde la barra del menú principal para lanzar **Preferencias de Ratón**. En la pestaña de **Botones**, haga clic en la cajilla **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 caracter especial en un archivo de **gedit**, seleccione desde la barra del menú principal **Aplicaciones** → **Accessories** → **Mapa de caracteres**. Luego, desde la barra de menús de **mapa de caracteres** elija **Búsqueda** → **Hallar...**, teclee el nombre del caracter en el campo **Búsqueda** y haga clic en **Siguiente**. El caracter buscado se resaltará en la **Tabla de caracteres**. Haga doble clic en este caracter resaltado para colocarlo en el campo de **Texto para copiar** y luego haga clic en el botón de **Copiar**. Ahora regrese a su documento y elija **Editar** → **Pegar** desde 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 nombredeusuario@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 las palabras en itálicas y negrita sobre — nombre de usuario, domain.name, sistema de archivo, paquete, versión y lanzamiento. Cada palabra es un marcador de posición, tanto para el texto que usted escriba al ejecutar un comando como para el texto mostrado 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:

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;

public class ExClient
{
    public static void main(String args[])
        throws Exception
    {
        InitialContext iniCtx = new InitialContext();
        Object          ref    = iniCtx.lookup("EchoBean");
        EchoHome        home   = (EchoHome) ref;
        Echo            echo    = home.create();

        System.out.println("Created Echo");

        System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
    }
}
```

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.



Advertencia

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

El administrador de volúmenes lógicos

LVM

Este capítulo proporciona un resumen de alto nivel de los componentes del administrador de volúmenes lógicos (LVM).

1.1. Volúmenes lógicos

La administración de volúmenes crea una capa de abstracción sobre el almacenaje físico, permitiendo la creación de volúmenes de almacenamiento lógico. Esto ofrece mayor flexibilidad que el uso directo de almacenaje físico.

Un volumen lógico proporciona virtualización de almacenamiento. 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 el dimensionamiento y desplazamiento 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.

1.2. Sinopsis de la arquitectura LVM

Para el lanzamiento del sistema operativo RHEL 4, el administrador de volúmenes lógico LVM1 fue remplazado por LVM2, el cual tiene un marco de trabajo del kernel más genérico. 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, “LVM Logical Volume Components” shows the components of a simple LVM logical volume:

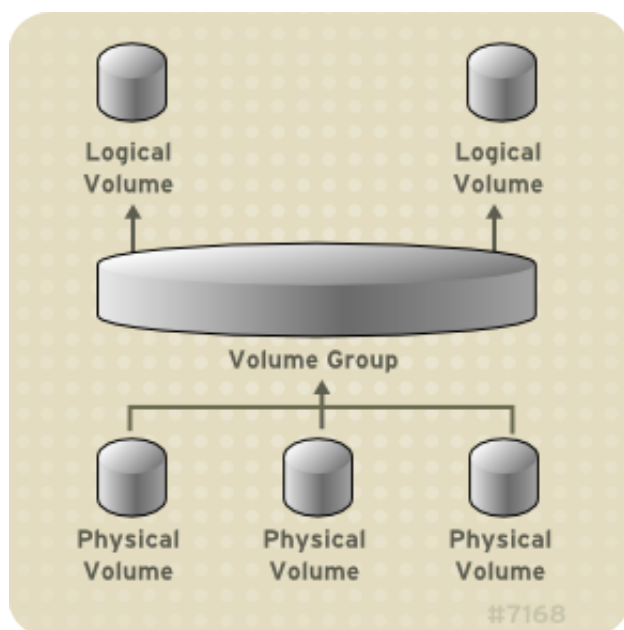


Figura 1.1. LVM Logical Volume Components

For detailed information on the components of an LVM logical volume, see [Capítulo 2, Componentes de LVM](#).

1.3. 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) a través de LVM.

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

- Si sólo 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 Red Hat Cluster Suite, incluyendo el demonio **clmvd**. El demonio **clmvd** es la extensión principal de cluster para LVM. El demonio **clmvd** se ejecuta en cada computador del cluster y distribuye las actualizaciones de metadatos de LVM en un cluster, presentando cada computador del cluster con el mismo panorama de volúmenes lógicos. Para mayor información sobre la instalación y administración de Red Hat Cluster Suite, consulte *Configuración y administración de un Red Hat Cluster*.

Para asegurarse de que **clmvd** se inicie en el arranque, puede ejecutar un 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
```

Creating LVM logical volumes in a cluster environment is identical to creating LVM logical volumes on a single node. There is no difference in the LVM commands themselves, or in the LVM graphical user interface, as described in [Capítulo 4, Administración de LVM mediante comandos de CLI](#) and [Capítulo 7, Administración de LVM con la interfaz gráfica de LVM](#). In order to enable the LVM volumes you are creating in a cluster, the cluster infrastructure must be running and the cluster must be quorate.

By default, logical volumes created with CLVM on shared storage are visible to all computers that have access to the shared storage. It is possible, however, to create logical volumes when the storage devices are visible to only one node in the cluster. It is also possible to change the status of a logical

volume from a local volume to a clustered volume. For information, see [Sección 4.3.2, “Cómo crear grupos de volúmenes en un cluster”](#) and [Sección 4.3.7, “Cómo cambiar los parámetros de un grupo de volúmenes”](#).

Figura 1.2, “Sinopsis de CLVM” shows a CLVM overview in a Red Hat cluster.

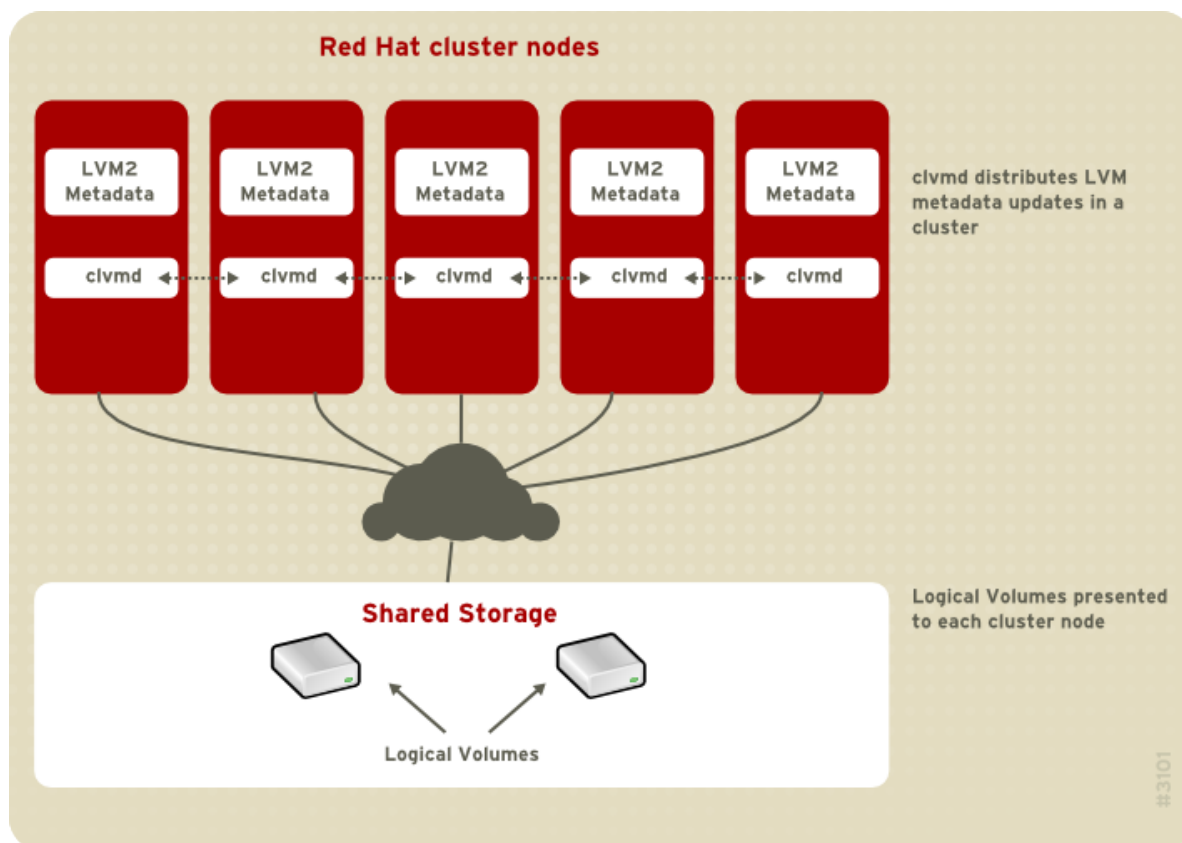


Figura 1.2. Sinopsis de CLVM



Nota

El almacenaje compartido para uso en Red Hat Cluster Suite requiere que usted esté ejecutando el demonio del administrador de volumen lógico de cluster (**clvmd**) o los agentes de administración de alta disponibilidad de volumen lógico (HA-LVM). Si no puede utilizar ni el demonio **clvmd** ni el HA-LVM por razones operacionales o porque no tiene la debida autorización, no debe utilizar una instancia única de LVM en el disco compartido, ya que esto puede ocasionar corrupción de datos. Si tiene alguna duda, por favor contacte al representante de servicio Red Hat.



Nota

CLVM requires changes to the **lvm.conf** file for cluster-wide locking. Information on configuring the **lvm.conf** file to support clustered locking is provided within the **lvm.conf** file itself. For information about the **lvm.conf** file, see [Apéndice B, Archivos de configuración LVM](#).

1.4. Sinopsis del documento

Lo que resta de este documento incluye los siguiente capítulos:

- [Capítulo 2, Componentes de LVM](#) describes the components that make up an LVM logical volume.
- [Capítulo 3, Sinopsis de la administración de LVM](#) provides an overview of the basic steps you perform to configure LVM logical volumes, whether you are using the LVM Command Line Interface (CLI) commands or the LVM Graphical User Interface (GUI).
- [Capítulo 4, Administración de LVM mediante comandos de CLI](#) summarizes the individual administrative tasks you can perform with the LVM CLI commands to create and maintain logical volumes.
- [Capítulo 5, Ejemplos de configuración de LVM](#) provides a variety of LVM configuration examples.
- [Capítulo 6, Solución de errores en LVM](#) provides instructions for troubleshooting a variety of LVM issues.
- [Capítulo 7, Administración de LVM con la interfaz gráfica de LVM](#) summarizes the operating of the LVM GUI.
- [Apéndice A, Mapeo de Dispositivos](#) describes the Device Mapper that LVM uses to map logical and physical volumes.
- [Apéndice B, Archivos de configuración LVM](#) describes the LVM configuration files.
- [Apéndice C, Etiquetas de objetos LVM](#) describes LVM object tags and host tags.
- [Apéndice D, Metadatos del grupo de volúmenes LVM](#) describes LVM volume group metadata, and includes a sample copy of metadata for an LVM volume group.

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, ésta se almacena al final del dispositivo. Si por accidente sobrescribe la primera parte del dispositivo, la copia de los metadatos al final de éste le permitirá recuperar los datos.

For detailed information about the LVM metadata and changing the metadata parameters, see [Apéndice D, Metadatos del grupo de volúmenes LVM](#).

2.1.1. LVM Physical Volume Layout

Figura 2.1, “Distribución de un volumen físico” shows the layout of an LVM physical volume. The LVM label is on the second sector, followed by the metadata area, followed by the usable space on the device.



Nota

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

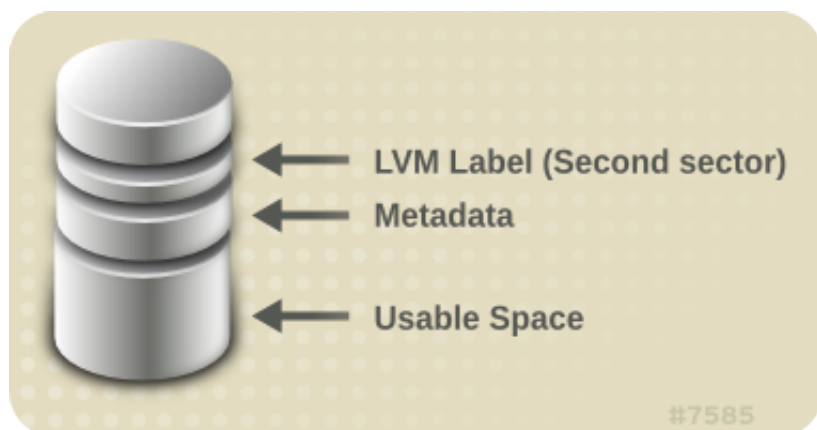


Figura 2.1. Distribución de un volumen físico

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 mover particiones cuando se migra un sistema existente a un volumen LVM. Por ejemplo, en un sistema con pocos discos puede llegar a ser necesario mover 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.

Creating a linear volume assigns a range of physical extents to an area of a logical volume in order. For example, as shown in [Figura 2.2, “Mapas de extensiones”](#) logical extents 1 to 99 could map to one physical volume and logical extents 100 to 198 could map to a second physical volume. From the point of view of the application, there is one device that is 198 extents in size.

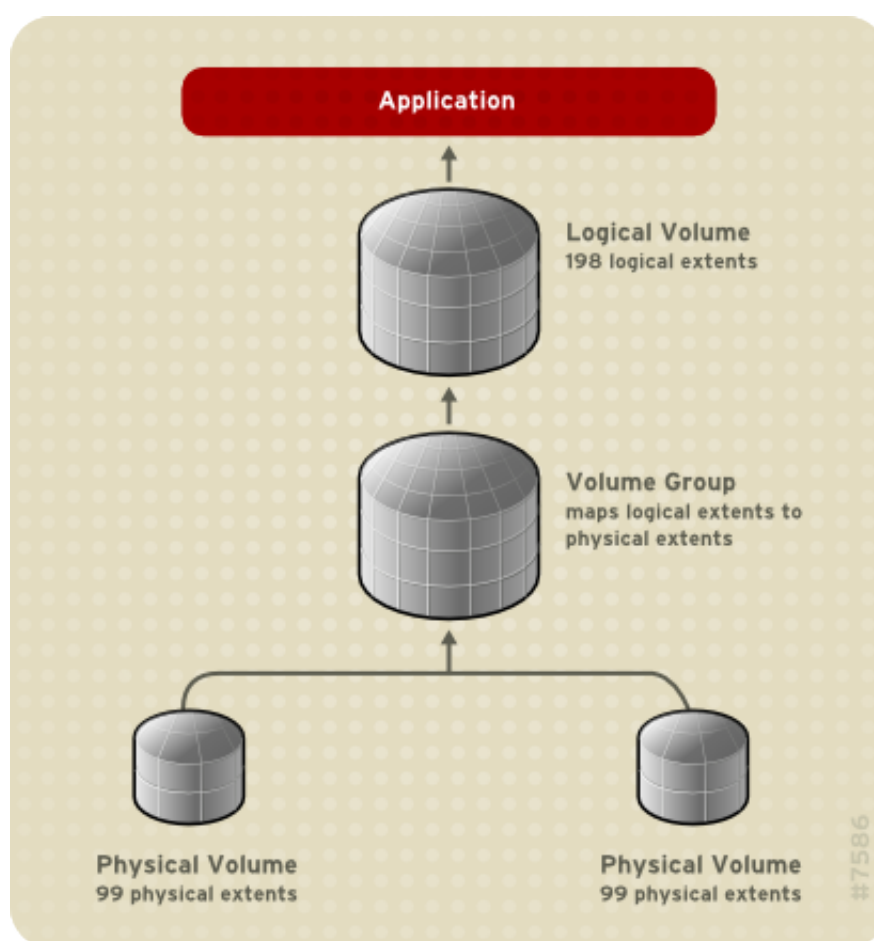


Figura 2.2. Mapas de extensiones

The physical volumes that make up a logical volume do not have to be the same size. [Figura 2.3, “Volumen lineal con volúmenes físicos dispares”](#) shows volume group **VG1** with a physical extent size of 4MB. This volume group includes 2 physical volumes named **PV1** and **PV2**. The physical volumes are divided into 4MB units, since that is the extent size. In this example, **PV1** is 100 extents in size (400MB) and **PV2** is 200 extents in size (800MB). You can create a linear volume any size between 1 and 300 extents (4MB to 1200MB). In this example, the linear volume named **LV1** is 300 extents in size.

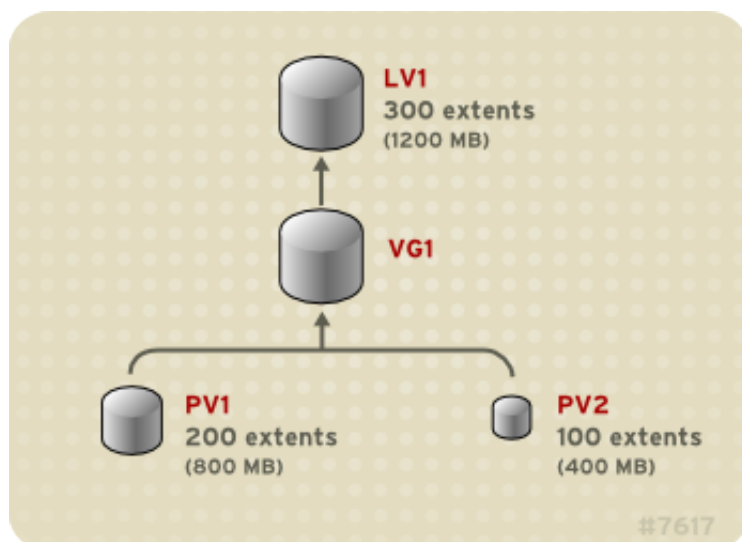


Figura 2.3. Volumen lineal con volúmenes físicos dispares

You can configure more than one linear logical volume of whatever size you desire from the pool of physical extents. *Figura 2.4, “Volúmenes lógicos múltiples”* shows the same volume group as in *Figura 2.3, “Volumen lineal con volúmenes físicos dispares”*, but in this case two logical volumes have been carved out of the volume group: **LV1**, which is 250 extents in size (1000MB) and **LV2** which is 50 extents in size (200MB).

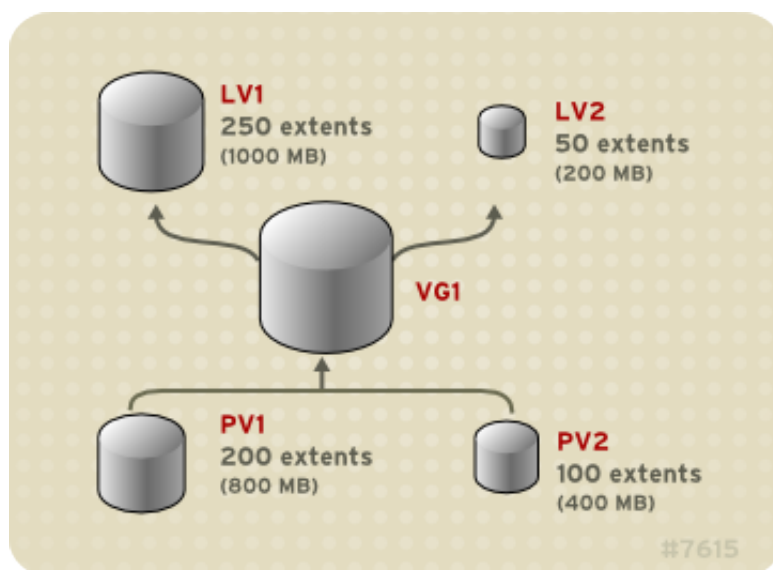


Figura 2.4. Volúmenes lógicos múltiples

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.

Este tipo de volúmenes mejoran 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, esto puede resultar en ganancias de rendimiento lineal por cada volumen físico añadido al enlace.

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 del enlace no puede exceder el tamaño de una extensión.

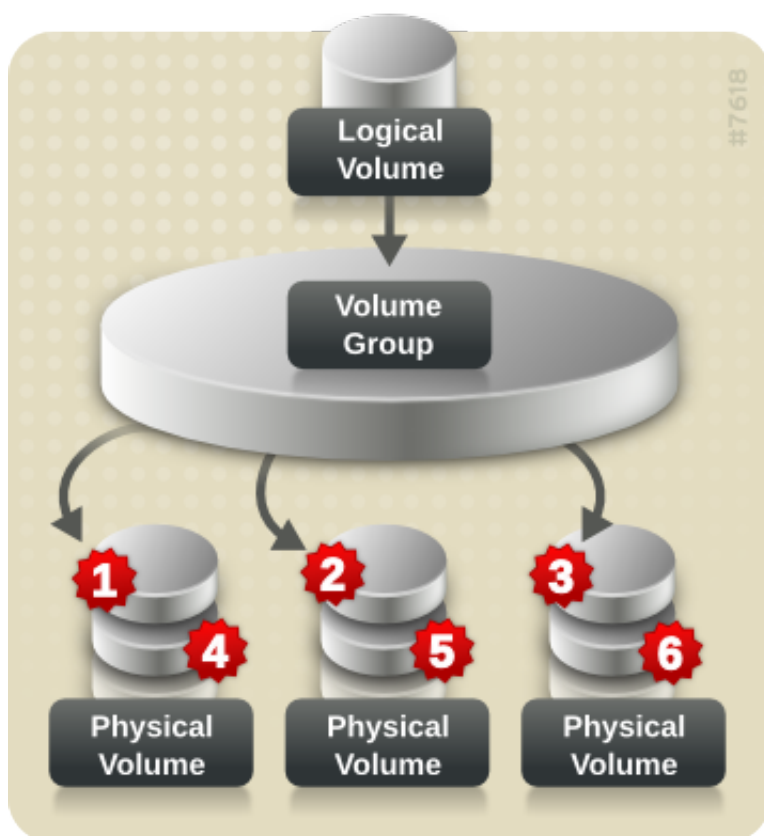


Figura 2.5. Entrelazando datos a lo largo de tres PV

Striped logical volumes can be extended by concatenating another set of devices onto the end of the first set. In order extend a striped logical volume, however, there must be enough free space on the underlying physical volumes that make up the volume group to support the stripe. For example, if you have a two-way stripe that uses up an entire volume group, adding a single physical volume to the volume group will not enable you to extend the stripe. Instead, you must add at least two physical volumes to the volume group. For more information on extending a striped volume, see [Sección 4.4.9, "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 cuales 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, “Mirrored Logical Volume” shows a mirrored logical volume with one mirror. In this configuration, the log is maintained on disk.

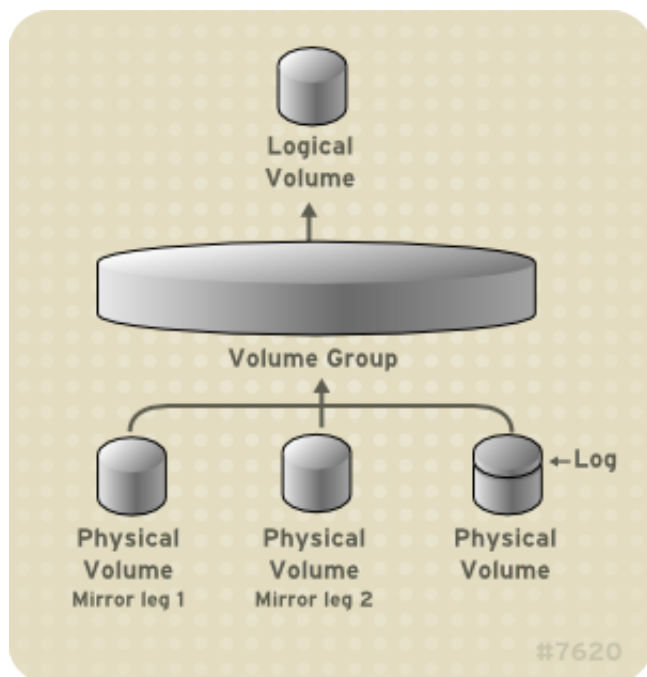


Figura 2.6. Mirrored Logical Volume



Nota

A partir del lanzamiento de RHEL 5.3, los volúmenes lógicos en espejo son soportados en un cluster.

For information on creating and modifying mirrors, see [Sección 4.4.1.3, “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.

Because a snapshot copies only the data areas that change after the snapshot is created, the snapshot feature requires a minimal amount of storage. For example, with a rarely updated origin, 3-5 % of the origin's capacity is sufficient to maintain the snapshot.

**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.

Si la instantánea se llena, ésta es abandonada. Esto es para asegurar que hay suficiente espacio en el original. Por este motivo se debe monitorizar el tamaño de las instantáneas con regularidad. Las instantáneas, sin embargo, pueden ser incrementadas; así, si se tiene la capacidad de almacenaje, se puede incrementar el tamaño de la instantánea para que esta no sea abandonada. Asimismo, si ve que el volumen de la instantánea es mayor de lo que se necesita, puede reducir el tamaño del volumen para liberar espacio que es necesario 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 para utilizar con el monitor de máquinas virtuales Xen. Puede utilizar la funcionalidad de instantáneas para crear una imagen de disco, crear la instantánea y modificar la instantánea para una instancia particular de domU. Puede crear otra instantánea y modificarla para otra instancia de domU. Como el único almacenamiento usado son trozos que cambiaron en el original o la instantánea, la mayoría del volumen es compartido.

Sinopsis de la administración de LVM

This chapter provides an overview of the administrative procedures you use to configure LVM logical volumes. This chapter is intended to provide a general understanding of the steps involved. For specific step-by-step examples of common LVM configuration procedures, see [Capítulo 5, Ejemplos de configuración de LVM](#).

For descriptions of the CLI commands you can use to perform LVM administration, see [Capítulo 4, Administración de LVM mediante comandos de CLI](#). Alternately, you can use the LVM GUI, which is described in [Capítulo 7, Administración de LVM con la interfaz gráfica de LVM](#).

3.1. Creación de volúmenes LVM en un cluster

To create logical volumes in a cluster environment, you use the Clustered Logical Volume Manager (CLVM), which is a set of clustering extensions to LVM. These extensions allow a cluster of computers to manage shared storage (for example, on a SAN) using LVM. In order to use CLVM, the Red Hat Cluster Suite software, including the **clvm** daemon, must be started at boot time, as described in [Sección 1.3, “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 quórum

CLVM requires changes to the **lvm.conf** file for cluster-wide locking. Information on configuring the **lvm.conf** file to support clustered locking is provided within the **lvm.conf** file itself. For information about the **lvm.conf** file, see [Apéndice B, Archivos de configuración LVM](#).

By default, logical volumes created with CLVM on shared storage are visible to all computers that have access to the shared storage. It is possible, however, to create logical volumes when the storage devices are visible to only one node in the cluster. It is also possible to change the status of a logical volume from a local volume to a clustered volume. For information, see [Sección 4.3.2, “Cómo crear grupos de volúmenes en un cluster”](#) and [Sección 4.3.7, “Cómo cambiar los parámetros de un grupo de volúmenes”](#)



Nota

El almacenamiento compartido para uso en Red Hat Cluster Suite requiere que se esté ejecutando el demonio de administrador de volumen lógico (**clvm**) o los agentes de administración de volúmenes lógicos de alta disponibilidad (HA-LVM). Si no puede utilizar ni el demonio **clvm** ni HA-LVM por razones operacionales o porque no tiene la debida autorización, no debe utilizar una instancia única de LVM en un disco compartido ya que esto puede conllevar a la corrupción de datos. Si tiene alguna duda, por favor contacte al representante de servicio de Red Hat.

Para obtener mayor información sobre cómo instalar un Red Hat Cluster Suite y configurar la infraestructura de cluster, consulte [Configuración y administración de un Red Hat 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 GFS.

1. Crear un sistema de archivos GFS en el volumen lógico con el comando **gfs_mkfs**.
2. Crear 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 GFS 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 volumen a menos que se desactive la opción 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**. La duración en que los metadatos se almacenan en **/etc/lvm/archive** y la cantidad de éstos se determina en parámetros configurables del archivo de configuración **lvm.conf**. Una copia de seguridad del sistema diaria debería incluir el contenido del directorio **/etc/lvm**.

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.

You can manually back up the metadata to the `/etc/lvm/backup` file with the **vgcfgbackup** command. You can restore metadata with the **vgcfgrestore** command. The **vgcfgbackup** and **vgcfgrestore** commands are described in [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

The logging levels are set in the `/etc/lvm/lvm.conf` file, which is described in [Apéndice B, Archivos de configuración LVM](#).

Administración 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

If you are creating or modifying an LVM volume for a clustered environment, you must ensure that you are running the **clvmd** daemon. For information, see [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 un 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 la opción **--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 un volumen lógico, la ruta completa es opcional. Un volumen lógico llamado **lvol10** en un grupo de volúmenes llamado **vg0** puede ser especificado como **vg0/lvol10**. Cuando una lista de grupos de volúmenes requerida se deja vacía, ésta se reemplaza con una lista de todos los grupos de volúmenes. Cuando se da un grupo de volúmenes en una opción que requiere una lista de volúmenes lógicos, todos los volúmenes lógicos del grupo son añadidos a la lista. Por ejemplo, el comando **lvdisplay vg0** mostrará todos los volúmenes lógicos del 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 "lvol0" 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 lv010
Creating volume group backup "/etc/lvm/backup/new_vg" (seqno 5).
Found volume group "new_vg"
Creating new_vg-lv010
Loading new_vg-lv010 table
Resuming new_vg-lv010 (253:2)
Clearing start of logical volume "lv010"
Creating volume group backup "/etc/lvm/backup/new_vg" (seqno 5).
Logical volume "lv010" 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 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.

All LVM objects are referenced internally by a UUID, which is assigned when you create the object. This can be useful in a situation where you remove a physical volume called **/dev/sdf** which is part of a volume group and, when you plug it back in, you find that it is now **/dev/sdk**. LVM will still find the physical volume because it identifies the physical volume by its UUID and not its device name. For information on specifying the UUID of a physical volume when creating a physical volume, see [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, éste no debe tener una tabla de particiones. Para particiones de disco DOS, el identificador de la partición debe establecerse a 0x8e con el comando **fdisk** o **cfdisk**. 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.

Los siguientes comandos inicializan **/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.

```
# lvmdiskscan
/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**, **pvdisplay** y **pvscan**.

The **pvs** command provides physical volume information in a configurable form, displaying one line per physical volume. The **pvs** command provides a great deal of format control, and is useful for scripting. For information on using the **pvs** command to customize your output, see [Sección 4.9, “Reporte personalizado para LVM”](#).

El comando **pvdisplay** 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 **pvdisplay** 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]
```

You can define a filter in the **lvm.conf** so that this command will avoid scanning specific physical volumes. For information on using filters to control which devices are scanned, see [Sección 4.6, “Cómo controlar la exploración de dispositivos LVM a través de filtros”](#).

4.2.3. Cómo prevenir la asignación en un volumen físico

Puede prevenir la asignación de las 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 éste será removido del 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 cambiar el tamaño de un volumen físico

Si necesita cambiar el tamaño de 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 remover volúmenes físicos

Si un dispositivo no es requerido en el LVM, éste se puede remover con el comando **pvremove**. La ejecución de **pvremove** convierte en ceros los metadatos del LVM en un volumen físico vacío.

If the physical volume you want to remove is currently part of a volume group, you must remove it from the volume group with the **vgreduce** command, as described in [Sección 4.3.6, “Cómo remover 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

To create a volume group from one or more physical volumes, use the **vgcreate** command. The **vgcreate** command creates a new volume group by name and adds at least one physical volume to it.

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 volumen, el 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 con las opciones **-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 la opción **--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 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**.

En general, las políticas de asignaciones diferentes a **normal** sólo se requieren en casos especiales donde se usted debe 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 **myvg1** y **myvg2**, cada uno con tres volúmenes lógicos llamados **lv01**, **lv02** y **lv03**, 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 Exabytes en CPUs de 64-bit

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 volumen 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
```

You can change whether an existing volume group is local or clustered with the **-c** option of the **vgchange** command, which is described in [Sección 4.3.7, “Cómo cambiar los parámetros de un grupo de volúmenes”](#).

Puede comprobar si un grupo de volumen existente es un grupo de volumen 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 volumen **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
```

For more information on the **vgs** command, see [Sección 4.3.4, “Cómo mostrar los grupos de volúmenes”](#)[Sección 4.9, “Reporte personalizado para LVM”](#), and the **vgs** man page.

4.3.3. Cómo agregar volúmenes físicos al grupo de volúmenes

To add additional physical volumes to an existing volume group, use the **vgextend** command. The **vgextend** command increases a volume group's capacity by adding one or more free physical volumes.

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**.

The **vgscan** command will also display the volume groups, although its primary purpose is to scan all the disks for volume groups and rebuild the LVM cache file. For information on the **vgscan** command, see [Sección 4.3.5, “Cómo explorar discos en búsqueda de grupos de volúmenes para construir el archivo de caché”](#).

The **vgs** command provides volume group information in a configurable form, displaying one line per volume group. The **vgs** command provides a great deal of format control, and is useful for scripting. For information on using the **vgs** command to customize your output, see [Sección 4.9, “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 serán mostrados.

```
# 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. Éste construye el archivo de caché LVM en **/etc/lvm/.cache**, el cual mantiene una lista de dispositivos LVM.

LVM ejecuta el comando **vgscan** automáticamente al iniciar el sistema y en otros momentos durante la operación de LVM, tales como cuando se ejecuta el comando **vgcreate** o cuando LVM detecta una inconsistencia. Podría tener que ejecutar **vgscan** de forma manual cuando cambia la configuración del hardware, para que los nuevos dispositivos sean detectados por el sistema. Por ejemplo, cuando añada un disco al sistema en un SAN o realice una conexión en caliente de un nuevo disco que ha sido etiquetado como volumen físico.

You can define a filter in the **lvm.conf** file to restrict the scan to avoid specific devices. For information on using filters to control which devices are scanned, see [Sección 4.6, “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 remover un volumen físico de un grupo de volúmenes

To remove unused physical volumes from a volume group, use the **vgreduce** command. The **vgreduce** command shrinks a volume group's capacity by removing one or more empty physical volumes. This frees those physical volumes to be used in different volume groups or to be removed from the system.

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-0VSen7

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 **mi_grupo_vol**.

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

4.3.7. Cómo cambiar los parámetros de un grupo de volúmenes

There are several volume group parameters that you can change for an existing volume group with the **vgchange** command. Primarily, however, this command is used to deactivate and activate volume groups, as described in [Sección 4.3.8, “Cómo activar y desactivar grupos de volúmenes”](#),

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 crea un grupo de volúmenes, éste se activa de 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 la opción **-a** (**--available**) del comando **vgchange**.

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

```
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.

You can deactivate individual logical volumes with the **lvchange** command, as described in [Sección 4.4.4, “Cómo cambiar los parámetros de un grupo de volúmenes lógico”](#), For information on activating logical volumes on individual nodes in a cluster, see [Sección 4.8, “Cómo activar volúmenes lógicos en nodos individuales en un cluster”](#).

4.3.9. Cómo remover grupos de volúmenes

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

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

4.3.10. Cómo separar 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

Two combine two volume groups into a single volume group, use the **vgmerge** command. You can merge an inactive "source" volume with an active or an inactive "destination" volume if the physical extent sizes of the volume are equal and the physical and logical volume summaries of both volume groups fit into the destination volume groups limits.

El siguiente comando fusiona el grupo de volumen inactivo **my_vg** en el grupo de volumen activo o inactivo **databases** dando información 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.

For an example of using the **vgcfrestore** command to recover physical volume metadata, see [Sección 6.4, "Cómo recuperar los metadatos de un volumen físico"](#).

4.3.13. Cómo cambiar el nombrar 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 mover un grupo de volúmenes a otro sistema

Puede mover 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 mover 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.
2. Utilice la opción **-a n** del comando **vgchange** para marcar el grupo de volúmenes como inactivo. Ésto previene cualquier actividad en el grupo de volúmenes.
3. Utilice el comando **vgexport** para exportar el grupo de volúmenes. Esto previene que éste sea accedido por el sistema desde el 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.

4. Cuando los discos sean conectados en el nuevo sistema, utilice el comando **vgimport** para importar el grupo de volúmenes y que el sistema pueda tener acceso a éste.
5. Active el grupo de volúmenes con la opción **-a y** del comando **vgchange**.
6. Monte el sistema de archivos para que éste pueda ser usado.

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 el volumen lógico activo. Crea cualquier archivo especial que se necesite y remueve aquellos que no se utilizan.

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

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

Para crear un volumen lógico, utilice el comando **lvcreate**. Puede crear volúmenes lineales, volúmenes entrelazados y volúmenes en espejo, tal y como se describe en las siguientes subdivisiones.

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.

Las siguientes secciones proporcionan ejemplos de creación de volúmenes lógicos para tres tipos de volúmenes lógicos que se pueden crear con LVM.

4.4.1.1. Cómo crear volúmenes lineales

Cuando cree un volumen lógico, el volumen lógico es creado desde 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 volumen **vg0**.

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

Puede usar la opción **-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 la opción **-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
```

You can use **-l** argument of the **lvcreate** command to create a logical volume that uses the entire volume group. Another way to create a logical volume that uses the entire volume group is to use the **vgdisplay** command to find the "Total PE" size and to use those results as input to the **lvcreate** command.

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
```

The underlying physical volumes used to create a logical volume can be important if the physical volume needs to be removed, so you may need to consider this possibility when you create the logical volume. For information on removing a physical volume from a volume group, see [Sección 4.3.6, “Cómo remover 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 **/dev/sda1** que van desde 0 hasta 25 y las extensiones que van desde 50 hasta 125 del volumen físico **/dev/sdb1** en el grupo de volúmenes **testvg**.

```
lvcreate -l 100 -n testlv testvg /dev/sda1:0-25 /dev/sdb1:50-125
```

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-
```

The default policy for how the extents of a logical volume are allocated is **inherit**, which applies the same policy as for the volume group. These policies can be changed using the **lvchange** command. For information on allocation policies, see [Sección 4.3.1, “Cómo crear grupos de volúmenes”](#).

4.4.1.2. Cómo crear volúmenes entrelazados

For large sequential reads and writes, creating a striped logical volume can improve the efficiency of the data I/O. For general information about striped volumes, see [Sección 2.3.2, “Volúmenes lógicos entrelazados”](#).

Cuando cree un volumen lógico entrelazado, especifique el número de enlaces con la opción **-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 basado en tres pilares, el tamaño máximo es tres veces el tamaño del dispositivo más pequeño.

El siguiente comando crea un volumen lógico con dos volúmenes físicos con un enlazado de 64KB. El volumen lógico tiene 50 GB, se llama **gfslv** y está construido sobre el grupo **vg0**:

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

Como con los volúmenes lineales, usted puede especificar las extensiones del volumen físico que está siendo utilizado para el enlace. El siguiente comando crea un volumen entrelazado con 100 extensiones en tamaño que cubre dos volúmenes físicos, se llama **stripelv** y está en el grupo de volúmenes **testvg**. El enlace utilizará los sectores 0-50 de **/dev/sda1** y los sectores 50-100 de **/dev/sdb1**.

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

4.4.1.3. Cómo crear volúmenes en espejo

Cuando cree un volumen en espejo, se debe especificar el número de copias de los datos a realizar con la opción **-m** del comando **lvcreate**. Si se especifica **-m1** se creará un espejo, creando dos copias del sistema de archivos: un volumen lógico lineal y una copia. De igual manera, si se especifica **-m2** se crean 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' gigabytes, se llama **mirrorlv**, y está construido sobre el grupo de volúmenes **vg0**:

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

Un espejo LVM divide el dispositivo a copiar en regiones que, por defecto, tienen 512KB. Puede utilizar la opción **-R** para especificar el tamaño de la región en MB. 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 mantener la persistencia a través de los reinicios. Puede especificar que este registro sea guardado en memoria con el argumento **--corelog**; eliminando así la necesidad de un dispositivo de registro adicional, pero requiere que el espejo sea sincronizado tras cada reinicio.

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

```
# lvcreate -L 12MB -m1 --corelog -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 sólo 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 volumen **vg0** consta solamente de dos dispositivos. El volumen en espejo creado por este comando es de 500 MB, se denomina **mirrorlv**, y se construye del grupo de volumen **vg0**.

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

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 disco particular, especifique exactamente una de las extensiones en el disco en el cual será ubicado. 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. El volumen es de 500 MB, se llama **mirrorlv**, y se construye del grupo de volúmenes **vg0**. El primer pilar del espejo está en el dispositivo **/dev/sda1**, el segundo está en **/dev/sdb1** y el registro del espejo 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 megabytes en tamaño, se llama **mirrorlv** y está construido del grupo de volúmenes **vg0**. El primer pilar del espejo comprende las extensiones desde 0 hasta 499 del dispositivo **/dev/sda1**, el segundo pilar del espejo abarca las extensiones desde 0 a 499 de **/dev/sdb1** y el registro del espejo inicia en la extensión 0 de **/dev/sdc1**. Éstas son extensiones de 1MB. Si alguna de las extensiones especificadas ha sido asignadas éstas serán ignoradas.

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



Nota

A partir del lanzamiento de RHEL 5.3, los volúmenes lógicos en espejo son soportados en un cluster.

4.4.1.4. 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 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.

If you lose a leg of a mirror, LVM converts the volume to a linear volume so that you still have access to the volume, without the mirror redundancy. After you replace the leg, you can use the **lvconvert** command to restore the mirror. This procedure is provided in [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 con espejo.

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

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

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

4.4.2. 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
```

Use a large minor number to be sure that it hasn't already been allocated to another device dynamically.

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.

4.4.3. Cómo dimensionar volúmenes lógicos

Para cambiar 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
```

4.4.4. 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).

You can use the **lvchange** command to activate and deactivate logical volumes. To activate and deactivate all the logical volumes in a volume group at the same time, use the **vgchange** command, as described in [Sección 4.3.7, “Cómo cambiar los parámetros de un grupo de volúmenes”](#).

El siguiente comando cambia los permisos en el volumen **lv01** en el grupo de volúmenes **vg00** para que sea de solo lectura.

```
lvchange -pr vg00/lv01
```

4.4.5. 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 **lvold** en el grupo de volúmenes **vg02** a **lvnew**.

```
lvrename /dev/vg02/lvold /dev/vg02/lvnew
```

```
lvrename vg02 lvold lvnew
```

For more information on activating logical volumes on individual nodes in a cluster, see [Sección 4.8, “Cómo activar volúmenes lógicos en nodos individuales en un cluster”](#).

4.4.6. Cómo remover volúmenes lógicos

Para remover un volumen lógico inactivo, utilice el comando **lvremove**. Desmonte el volumen lógico con el comando **umount** antes de removerlo. Además, en un entorno de cluster se debe desactivar el volumen lógico antes de removerlo.

Si el volumen lógico está montado, desmonte el volumen antes de removerlo.

El siguiente comando remueve el volumen lógico **/dev/testvg/testlv** del grupo de volúmenes **testvg**. Note que en este caso el volumen lógico 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 preguntando si desea remover el volumen lógico.

4.4.7. Cómo ver los volúmenes lógicos

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

The **lvs** command provides logical volume information in a configurable form, displaying one line per logical volume. The **lvs** command provides a great deal of format control, and is useful for scripting. For information on using the **lvs** command to customize your output, see [Sección 4.9, “Reporte personalizado para LVM”](#).

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

El siguiente comando muestra los atributos de **lvol12** 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 estados (activo o inactivo).

```
lvdisplay -v /dev/vg00/lvol12
```

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

```
# lvscan
ACTIVE                               '/dev/vg0/gfslv' [1.46 GB] inherit
```

4.4.8. Cómo aumentar los volúmenes lógicos

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

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

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 **testlv** 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 dimensionar 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.

4.4.9. 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 volumen 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 ahora el volumen lógico 5A al tamaño completo 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/strip1 -L 542G
Using stripesize of last segment 64.00 KB
Extending logical volume strip1 to 542.00 GB
Logical volume strip1 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/strip1 -L 406G
Using stripesize of last segment 64.00 KB
Extending logical volume strip1 to 406.00 GB
Insufficient suitable allocatable extents for logical volume strip1: 34480
more required
# lvextend -i1 -l+100%FREE vg/strip1
```

4.4.10. 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 **lv01** en el grupo de volúmenes **vg00** a tres extensiones lógicas.

```
lvreduce -l -3 vg00/lv01
```

4.5. 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 de LVM no están soportadas a través de los nodos de un cluster.

Since LVM snapshots are not cluster-aware, they require exclusive access to a volume. For information on activating logical volumes on individual nodes in a cluster, see [Sección 4.8, “Cómo activar volúmenes lógicos en nodos individuales en un cluster”](#).

El siguiente comando crea un volumen lógico de instantáneas que tiene 100 megabytes en tamaño 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/lvol0**, para el cual el volumen de instantánea **/dev/new_vg/newvgsnap** ha sido creado.

```
# lvdisplay /dev/new_vg/lvol0
--- Logical volume ---
LV Name                /dev/new_vg/lvol0
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/lvol0**, por el cual un volumen de instantánea **/dev/new_vg/newvgsnap** ha sido creado.

```
# 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
```



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.

4.6. 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 los dispositivos LVM a explorar utilizando filtros en el archivo de configuración **lvm.conf**. Los filtros consisten en una serie de expresiones regulares simples que se aplican a los nombres de dispositivos en el directorio **/dev** para decidir cuales dispositivos de bloque encontrados deben ser aceptados o rechazados.

Los siguientes ejemplos muestran el uso de filtros para controlar cuáles dispositivos LVM deben ser explorados. Note que algunos de estos ejemplos no representan necesariamente la mejor práctica, ya que las expresiones regulares se combinan libremente en cambio del 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/.*/" ]
```

For more information on the **lvm.conf** file, see [Apéndice B, Archivos de configuración LVM](#) and the **lvm.conf(5)** man page.

4.7. Asignación de datos en línea

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

El comando **pvmove** divide los datos a mover 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)**.

Because the **pvmove** command uses mirroring, it is not cluster-aware and needs exclusive access to a volume. For information on activating logical volumes on individual nodes in a cluster, see [Sección 4.8, “Cómo activar volúmenes lógicos en nodos individuales en un cluster”](#).

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 sólo 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
```

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

Si tiene LVM instalado en un entorno de cluster, podrá necesitar activar el volumen lógico exclusivamente en un nodo. Por ejemplo, el comando **pvmove** no reconoce los clusters y necesita acceso exclusivo al volumen. Las instantáneas de LVM también requieren acceso al volumen.

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.

You can also activate logical volumes on individual nodes by using LVM tags, which are described in [Apéndice C, Etiquetas de objetos LVM](#). You can also specify activation of nodes in the configuration file, which is described in [Apéndice B, Archivos de configuración LVM](#).

4.9. 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.

4.9.1. 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 la opción **-o**. Por ejemplo, el siguiente mensaje de salida utiliza los valores predeterminados de **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 sólo 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 Joqlch-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 Joqlch-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
```

- La opción **--noheadings** suprime el encabezado. Esta opción puede ser útil en scripts.

El siguiente ejemplo utiliza la opción **--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
```

- La opción **--separator *separador*** utiliza *separador* para separar cada campo. Esto puede ser útil en un script si está ejecutando un comando **grep** sobre la salida.

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 la opción **separator** utilice la opción **separator** junto con la opción **--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
```

You can use the **-P** argument of the **lvs** or **vgs** command to display information about a failed volume that would otherwise not appear in the output. For information on the output this argument yields, see [Sección 6.2, “Cómo ver la información en dispositivos fallidos”](#).

Para obtener una lista completa de las opciones 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
```

4.9.2. 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 psv” lists the display arguments of the **pvs** command, along with the field name as it appears in the header display and a description of the field.

Tabla 4.1. Campos de visualización de psv

Opción	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 la opción **-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 la opción **--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
/dev/new_vg/lvol0
/dev/ram
/dev/ram0
/dev/ram2
```

/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” lists the display arguments of the **vgs** command, along with the field name as it appears in the header display and a description of the field.

Tabla 4.2. Campos de visualización de vgs

Opción	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 la opción **-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” lists the display arguments of the **lvs** command, along with the field name as it appears in the header display and a description of the field.

Tabla 4.3. Campos de visualización de lvs

Opción	Encabezado	Descripción
chunksize chunk_size	Chunk	Tamaño de unidad en el volumen de instantánea
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	El estado del volumen lógico. Los bits de atributos del volumen lógico son: Bit 1: tipo de volumen: espejo (m), espejo sin sincronizar (M), original (o), pvmove (p), instantánea (s), instantánea no válida (S), virtual (v) Bit2: Permisos: escritura (w), solo lectura (r) 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 . Bit 4: espejo corregido (m) Bit 5 Estado: activo (a), suspendido (s), instantánea no válida (l), instantánea suspendida no válida (S), dispositivo en mapa presente sin tablas (d), dispositivo en mapa presente sin tabla inactivo (i) Bit 6: dispositivo abierto (o)
lv_kernel_major	Maj	Número mayor del dispositivo del volumen lógico (-1 si está inactivo)
lv_kernel_minor	Min	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

Opción	Encabezado	Descripción
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_size	Region	El tamaño de la unidad de un volumen lógico con espejo
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_size	Stripe	Tamaño de unidades del enlace en un volumen lógico entrelazado

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-OjsI-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 la opción **--segments** del comando **lvs** para mostrar las columnas predeterminadas que enfatizan la información del segmento. Al utilizar la opción **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
newvg1  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 la opción **segments** especificada.

```
# 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
```

4.9.3. 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 la opción **-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 -0 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 -0 pv_free
PV          PSize
/dev/sdc1   17.14G
/dev/sdd1   17.14G
/dev/sdb1   17.14G
```

Para mostrar un orden inverso, utilice la opción **-0** y preceda el campo especificado con el carácter **-**.

```
# pvs -o pv_name,pv_size,pv_free -0 -pv_free
PV          PSize  PFree
/dev/sdd1   17.14G 17.14G
/dev/sdb1   17.14G 17.14G
/dev/sdc1   17.14G 17.09G
```

4.9.4. Cómo especificar unidades

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

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

```
# 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 múltiplos de 1024. Éste es el comportamiento predeterminado.

```
# 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 múltiplos 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 megabytes.

```
# 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
```

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 **nuevo_volumen_lógico** que consta de los 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 **nuevo_grupo_vol**.

```
[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 **nuevo_volumen_logico** desde el grupo de volúmenes **nuevo_grupo_vol**. 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 GFS en el volumen lógico.

```
[root@tng3-1 ~]# gfs_mkfs -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 **volumen_logico_entrelazado** 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 **grupo_vol_entrelazado**.

```
[root@tng3-1 ~]# vgcreate striped_vol_group /dev/sda1 /dev/sdb1 /dev/sdc1
Volume group "striped_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
striped_vol_group  3   0   0 wz--n- 51.45G 51.45G
```

5.2.3. Cómo crear el volumen lógico

El siguiente comando crea el volumen lógico entrelazado **volumen_logico_entrelazado** desde el grupo de volúmenes **grupo_vol_entrelazado**. Este ejemplo crea un volumen lógico de 2 GB en tamaño, con enlaces y un tamaño de enlace de 4 KB.

```
[root@tng3-1 ~]# lvcreate -i3 -l4 -L2G -nstriped_logical_volume striped_vol_group
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 GFS en el volumen lógico.

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

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

Device:                /dev/striped_vol_group/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/striped_vol_group/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/striped_vol_group/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 **milv** se crea desde el grupo de volúmenes **mivol1**, el cual consta de tres volúmenes físicos **/dev/sda1**, **/dev/sdb1** y **/dev/sdc1**.

Una vez completado este procedimiento, el grupo de volúmenes **mivg** 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 mover 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 **tuvg**, utilice el comando **vgsplit** para separar el grupo de volúmenes **mivg**.

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 **miv1** y luego divide el grupo de volúmenes **tuvg** del grupo de volúmenes **mivg**, moviendo el volumen físico **/dev/sdc1** al nuevo grupo de volúmenes **tuvg**.

```
[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 **tu1v**.

```
[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 ~]# gfs_mkfs -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 **mi1v** tuvo que ser desactivado anteriormente, éste debe ser activado de nuevo 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 remover un disco de un volumen lógico

Este ejemplo muestra cómo remover 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 remover un disco, primero se deben mover las extensiones en el volumen físico LVM a un disco o grupo de discos diferentes.

5.4.1. Cómo mover las extensiones a un volumen físico existente

En este ejemplo, el volumen lógico es distribuido a través de cuatro volúmenes físicos en el grupo de volumen **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 mover 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 remover 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   myvg 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 mover 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**:

```
[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 mover 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 mover 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 remover el volumen físico antiguo del grupo de volúmenes

Después de mover los datos de **/dev/sdb1**, puede removerlo 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 remover el disco del sistema.

Solución de errores en LVM

Este capítulo proporciona instrucciones sobre cómo solucionar diferentes problemas que puedan encontrarse en 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.
- If the problem is related to the logical volume activation, set 'activation = 1' in the 'log' section of the configuration file and run the command with the **-vvvv** argument. After you have finished examining this output be sure to reset this parameter to 0, to avoid possible problems with the machine locking during low memory situations.
- 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 la opción **-P** del comando **lvs** o **vgs** para ver la información sobre un volumen fallido que de otra forma no aparecería en los mensajes de salida. Esta opción 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 el siguiente mensaje de 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 causa 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 un espejo de un volumen LVS falla porque el dispositivo subyacente para un volumen físico falla. Cuando un espejo falla, LVM convierte el volumen en espejo en un volumen lineal, el cual continua operacional como antes, pero sin la redundancia del 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/sdg1" successfully created
Physical volume "/dev/sdg2" 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 causa 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
```

Capítulo 6. Solución de errores en LVM

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.

To rebuild the mirrored volume, you replace the broken drive and recreate the physical volume. If you use the same disk rather than replacing it with a new one, you will see "inconsistent" warnings when you run the **pvcreeate** command.

```
[root@link-08 ~]# pvcreeate /dev/sda[12]
Physical volume "/dev/sda1" successfully created
Physical volume "/dev/sda2" 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/sda1    VG vg      lvm2 [603.94 GB]
PV /dev/sda2    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/sda[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/sda1    VG vg      lvm2 [603.93 GB / 603.93 GB free]
PV /dev/sda2    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/sda1 /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/sda1(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 la opción **--uuid** y **--restorefile** del comando **pvccreate** para restaurar el volumen físico. El siguiente ejemplo etiqueta el dispositivo **/dev/sdh1** como un volumen físico con el UUID

indicado anteriormente, **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 volumen. La opción **restorefile** ordena al comando **pvcreeate** para que el nuevo volumen físico sea compatible con el anterior en el grupo de volúmenes, asegurando 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 **pvcreeate** 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 **pvcreeate** sobrescribe sólo las áreas de metadatos LVM y no afecta las áreas de datos existentes.

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

You can then use the **vgcfgrestore** command to restore the volume group's metadata.

```
[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

If a physical volume fails or otherwise needs to be replaced, you can label a new physical volume to replace the one that has been lost in the existing volume group by following the same procedure as you would for recovering physical volume metadata, described in [Sección 6.4, “Cómo recuperar los metadatos de un volumen físico”](#). You can use the **--partial** and **--verbose** arguments of the **vgdisplay** command to display the UUIDs and sizes of any physical volumes that are no longer present. If you wish to substitute another physical volume of the same size, you can use the **pvcreeate** command with the **--restorefile** and **--uuid** arguments to initialize a new device with the same UUID as the missing physical volume. You can then use the **vgcfgrestore** command to restore the volume group's metadata.

6.6. Cómo remover 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 remover todos los volúmenes lógicos que utilizan el volumen físico desde el grupo de volúmenes con la opción **--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 **vgcfgrestore** 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 **vgcfgrestore** para retornar el grupo de volúmenes a su estado previo.

6.7. Extensiones libres insuficientes para un volumen lógico

You may get the error message "Insufficient free extents" when creating a logical volume when you think you have enough extents based on the output of the **vgdisplay** or **vgs** commands. This is because these commands round figures to 2 decimal places to provide human-readable output. To specify exact size, use free physical extent count instead of some multiple of bytes to determine the size of the logical volume.

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
--------	---	---	---	--------	--------	---	---	------

Alternately, you can extend the logical volume to use a percentage of the remaining free space in the volume group by using the **-1** argument of the **lvcreate** command. For information, see [Sección 4.4.1.1, “Cómo crear volúmenes lineales”](#).

Administración de LVM con la interfaz gráfica de LVM

Además de la interfaz para la línea de comandos, LVM proporciona una interfaz gráfica de usuario que puede ser utilizada para configurar volúmenes lógicos LVM. Para acceder a la utilidad escriba **system-config-lvm**. El capítulo de LVM del *Manual de implementación de Red Hat Enterprise Linux* proporciona instrucciones detalladas para configurar volúmenes lógicos LVM a través de esta utilidad.

Además, la interfaz gráfica de LVM está disponible como parte de la interfaz de administración Conga. Para obtener mayor información sobre cómo utilizar la interfaz gráfica LVM con Conga, consulte la ayuda en línea para Conga.

Apéndice A. Mapeo de Dispositivos

El Mapeo 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 mapeados que puedan usarse como volúmenes lógicos. No sabe sobre formatos de metadatos o grupos de volúmenes específicos.

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

LVM logical volumes are activated using the Device Mapper. Each logical volume is translated into a mapped device. Each segment translates into a line in the mapping table that describes the device. The Device Mapper supports a variety of mapping targets, including linear mapping, striped mapping, and error mapping. So, for example, two disks may be concatenated into one logical volume with a pair of linear mappings, one for each disk. When LVM2 creates a volume, it creates an underlying device-mapper device that can be queried with the **dmsetup** command. For information about the format of devices in a mapping table, see [Sección A.1, “Tabla de Mapas de Dispositivo”](#). For information about using the **dmsetup** command to query a device, see [Sección A.2, “Comando dmsetup”](#).

A.1. Tabla de Mapas de Dispositivo

Un dispositivo mapeado 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 mapeado está constituida por una lista de líneas de la forma:

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

En la primera línea la Tabla de Mapas 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 Mapeo de Dispositivos siempre se especifican en sectores (512 bytes).

Cuando un dispositivo se especifica como un parámetro de mapas en el Mapeo 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 muestra 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

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.

The optional **[no]sync** argument can be used to specify the mirror as "in-sync" or "out-of-sync". The **block_on_error** argument is used to tell the mirror to respond to errors rather than ignoring them.

#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; se especifica un dispositivo y un desplazamiento 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 Mapeo 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. The "original" volume (which uses the device number used by the original source volume), whose table is replaced by a "snapshot-origin" mapping from device #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 vuelve a ensayar en el dispositivo real. Puede servir como un destino intermedio mientras se reorganizan los metadatos LVM metadata durante las fallas.

El destino de mapas **error** sólo 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** sólo recibe los parámetros *start* y *length*.

El siguiente ejemplo muestra un destino **zero** para un dispositivo 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 sólo 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.

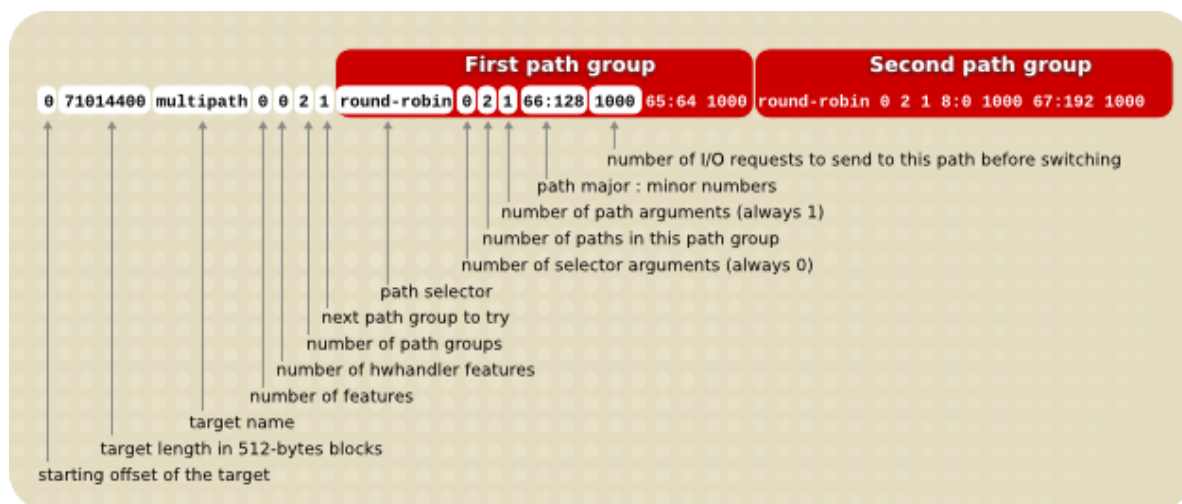


Figura A.1. Destino de mapas multirutas

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 Mapeo 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 Mapeo 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 Mapeo de Dispositivos. Si no especifica un nombre de dispositivo, la salida será información sobre todos los dispositivos del Mapeo 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 volumen 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

Possible states for this category are **LIVE** and **INACTIVE**. An **INACTIVE** state indicates that a table has been loaded which will be swapped in when a **dmsetup resume** command restores a device state to **ACTIVE**, at which point the table's state becomes **LIVE**. For information, see the **dmsetup** man page.

Open count

La cuenta abierta de referencia indica las veces que se abre el dispositivo. El comando **mount** abre un dispositivo.

Event number

The current number of events received. Issuing a **dmsetup wait n** command allows the user to wait for the n'th event, blocking the call until it is received.

Major, minor

Número de dispositivo mayor y menor

Number of targets

El numero 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-LogVol00
State:               ACTIVE
Read Ahead:         256
Tables present:     LIVE
Open count:         1
Event number:       0
Major, minor:       253, 0
Number of targets:  1
UUID: LVM-t0cS1kqFV9drb0X1Vr8sxeYP0tqcrpdegyqj5lZxe45JMGlmvtqLmbLpBcenh2L3
```

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-LogVol01     (253, 1)
VolGroup00-LogVol00     (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)
```

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 Mapeo de Dispositivo 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

Apéndice A. Mapeo de Dispositivos

el nombre de dispositivo, la salida será información sobre todos los dispositivos del Mapeo 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-testgfslv3: 1 dependencies : (8, 16)
testgfsvg-testgfslv2: 1 dependencies : (8, 16)
testgfsvg-testgfslv1: 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)
```

Apéndice B. 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**.

For information on loading additional configuration files, see [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

For each host tag, an extra configuration file is read if it exists: **lvm_hosttag.conf**. If that file defines new tags, then further configuration files will be appended to the list of files to read in. For information on host tags, see [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 volumen (configurable).

/etc/lvm/archive/

Directorio para los archivos de metadatos del grupo de volumen (la ruta del directorio y el tamaño del historial son configurables)

/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**. Este archivo de configuración es el archivo predeterminado para la versión de RHEL 5.3. Si su sistema está ejecutando una versión diferente a la de RHEL5, algunas configuraciones pueden diferir.

```
[root@tng3-1 lvm]# cat /etc/lvm/lvm.conf
# This is an example configuration file for the LVM2 system.
```

```
# It contains the default settings that would be used if there was no
# /etc/lvm/lvm.conf file.
#
# Refer to 'man lvm.conf' for further information including the file layout.
#
# To put this file in a different directory and override /etc/lvm set
# the environment variable LVM_SYSTEM_DIR before running the tools.

# This section allows you to configure which block devices should
# be used by the LVM system.
devices {

    # Where do you want your volume groups to appear ?
    dir = "/dev"

    # An array of directories that contain the device nodes you wish
    # to use with LVM2.
    scan = [ "/dev" ]

    # If several entries in the scanned directories correspond to the
    # same block device and the tools need to display a name for device,
    # all the pathnames are matched against each item in the following
    # list of regular expressions in turn and the first match is used.
    preferred_names = [ ]

    # Try to avoid using un-descriptive /dev/dm-N names, if present.
    # preferred_names = [ "^/dev/mpath/", "^/dev/mapper/mpath", "^/dev/[hs]d" ]

    # A filter that tells LVM2 to only use a restricted set of devices.
    # The filter consists of an array of regular expressions. These
    # expressions can be delimited by a character of your choice, and
    # prefixed with either an 'a' (for accept) or 'r' (for reject).
    # The first expression found to match a device name determines if
    # the device will be accepted or rejected (ignored). Devices that
    # don't match any patterns are accepted.

    # Be careful if there are symbolic links or multiple filesystem
    # entries for the same device as each name is checked separately against
    # the list of patterns. The effect is that if any name matches any 'a'
    # pattern, the device is accepted; otherwise if any name matches any 'r'
    # pattern it is rejected; otherwise it is accepted.

    # Don't have more than one filter line active at once: only one gets used.

    # Run vgscan after you change this parameter to ensure that
    # the cache file gets regenerated (see below).
    # If it doesn't do what you expect, check the output of 'vgscan -vvvv'.

    # By default we accept every block device:
    filter = [ "a/*/" ]

    # Exclude the cdrom drive
    # filter = [ "r|/dev/cdrom|" ]

    # When testing I like to work with just loopback devices:
    # filter = [ "a|loop/", "r/*/" ]

    # Or maybe all loops and ide drives except hdc:
    # filter = [ "a|loop|", "r|/dev/hdc|", "a|/dev/ide|", "r|.*|" ]

    # Use anchors if you want to be really specific
    # filter = [ "a|^/dev/hda8$", "r/*/" ]

    # The results of the filtering are cached on disk to avoid
    # rescanning dud devices (which can take a very long time).
```

```
# By default this cache is stored in the /etc/lvm/cache directory
# in a file called '.cache'.
# It is safe to delete the contents: the tools regenerate it.
# (The old setting 'cache' is still respected if neither of
# these new ones is present.)
cache_dir = "/etc/lvm/cache"
cache_file_prefix = ""

# You can turn off writing this cache file by setting this to 0.
write_cache_state = 1

# Advanced settings.

# List of pairs of additional acceptable block device types found
# in /proc/devices with maximum (non-zero) number of partitions.
# types = [ "fd", 16 ]

# If sysfs is mounted (2.6 kernels) restrict device scanning to
# the block devices it believes are valid.
# 1 enables; 0 disables.
sysfs_scan = 1

# By default, LVM2 will ignore devices used as components of
# software RAID (md) devices by looking for md superblocks.
# 1 enables; 0 disables.
md_component_detection = 1

# By default, if a PV is placed directly upon an md device, LVM2
# will align its data blocks with the the chunk_size exposed in sysfs.
# 1 enables; 0 disables.
md_chunk_alignment = 1

# If, while scanning the system for PVs, LVM2 encounters a device-mapper
# device that has its I/O suspended, it waits for it to become accessible.
# Set this to 1 to skip such devices. This should only be needed
# in recovery situations.
ignore_suspended_devices = 0
}

# This section that allows you to configure the nature of the
# information that LVM2 reports.
log {

    # Controls the messages sent to stdout or stderr.
    # There are three levels of verbosity, 3 being the most verbose.
    verbose = 0

    # Should we send log messages through syslog?
    # 1 is yes; 0 is no.
    syslog = 1

    # Should we log error and debug messages to a file?
    # By default there is no log file.
    #file = "/var/log/lvm2.log"

    # Should we overwrite the log file each time the program is run?
    # By default we append.
    overwrite = 0

    # What level of log messages should we send to the log file and/or syslog?
    # There are 6 syslog-like log levels currently in use - 2 to 7 inclusive.
    # 7 is the most verbose (LOG_DEBUG).
    level = 0

    # Format of output messages
    # Whether or not (1 or 0) to indent messages according to their severity
    indent = 1
}
```

```
# Whether or not (1 or 0) to display the command name on each line output
command_names = 0

# A prefix to use before the message text (but after the command name,
# if selected). Default is two spaces, so you can see/grep the severity
# of each message.
prefix = "  "

# To make the messages look similar to the original LVM tools use:
#   indent = 0
#   command_names = 1
#   prefix = " -- "

# Set this if you want log messages during activation.
# Don't use this in low memory situations (can deadlock).
# activation = 0
}

# Configuration of metadata backups and archiving. In LVM2 when we
# talk about a 'backup' we mean making a copy of the metadata for the
# *current* system. The 'archive' contains old metadata configurations.
# Backups are stored in a human readable text format.
backup {

    # Should we maintain a backup of the current metadata configuration ?
    # Use 1 for Yes; 0 for No.
    # Think very hard before turning this off!
    backup = 1

    # Where shall we keep it ?
    # Remember to back up this directory regularly!
    backup_dir = "/etc/lvm/backup"

    # Should we maintain an archive of old metadata configurations.
    # Use 1 for Yes; 0 for No.
    # On by default. Think very hard before turning this off.
    archive = 1

    # Where should archived files go ?
    # Remember to back up this directory regularly!
    archive_dir = "/etc/lvm/archive"

    # What is the minimum number of archive files you wish to keep ?
    retain_min = 10

    # What is the minimum time you wish to keep an archive file for ?
    retain_days = 30
}

# Settings for the running LVM2 in shell (readline) mode.
shell {

    # Number of lines of history to store in ~/.lvm_history
    history_size = 100
}

# Miscellaneous global LVM2 settings
global {
    library_dir = "/usr/lib64"

    # The file creation mask for any files and directories created.
    # Interpreted as octal if the first digit is zero.
    umask = 077

    # Allow other users to read the files
```

```
#umask = 022

# Enabling test mode means that no changes to the on disk metadata
# will be made. Equivalent to having the -t option on every
# command. Defaults to off.
test = 0

# Default value for --units argument
units = "h"

# Whether or not to communicate with the kernel device-mapper.
# Set to 0 if you want to use the tools to manipulate LVM metadata
# without activating any logical volumes.
# If the device-mapper kernel driver is not present in your kernel
# setting this to 0 should suppress the error messages.
activation = 1

# If we can't communicate with device-mapper, should we try running
# the LVM1 tools?
# This option only applies to 2.4 kernels and is provided to help you
# switch between device-mapper kernels and LVM1 kernels.
# The LVM1 tools need to be installed with .lvm1 suffices
# e.g. vgscan.lvm1 and they will stop working after you start using
# the new lvm2 on-disk metadata format.
# The default value is set when the tools are built.
# fallback_to_lvm1 = 0

# The default metadata format that commands should use - "lvm1" or "lvm2".
# The command line override is -M1 or -M2.
# Defaults to "lvm1" if compiled in, else "lvm2".
# format = "lvm1"

# Location of proc filesystem
proc = "/proc"

# Type of locking to use. Defaults to local file-based locking (1).
# Turn locking off by setting to 0 (dangerous: risks metadata corruption
# if LVM2 commands get run concurrently).
# Type 2 uses the external shared library locking_library.
# Type 3 uses built-in clustered locking.
locking_type = 3

# If using external locking (type 2) and initialisation fails,
# with this set to 1 an attempt will be made to use the built-in
# clustered locking.
# If you are using a customised locking_library you should set this to 0.
fallback_to_clustered_locking = 1

# If an attempt to initialise type 2 or type 3 locking failed, perhaps
# because cluster components such as clvmd are not running, with this set
# to 1 an attempt will be made to use local file-based locking (type 1).
# If this succeeds, only commands against local volume groups will proceed.
# Volume Groups marked as clustered will be ignored.
fallback_to_local_locking = 1

# Local non-LV directory that holds file-based locks while commands are
# in progress. A directory like /tmp that may get wiped on reboot is OK.
locking_dir = "/var/lock/lvm"

# Other entries can go here to allow you to load shared libraries
# e.g. if support for LVM1 metadata was compiled as a shared library use
# format_libraries = "liblvm2format1.so"
# Full pathnames can be given.

# Search this directory first for shared libraries.
# library_dir = "/lib"
```

```
# The external locking library to load if locking_type is set to 2.
#   locking_library = "liblvm2clusterlock.so"
}

activation {
    # How to fill in missing stripes if activating an incomplete volume.
    # Using "error" will make inaccessible parts of the device return
    # I/O errors on access. You can instead use a device path, in which
    # case, that device will be used to in place of missing stripes.
    # But note that using anything other than "error" with mirrored
    # or snapshotted volumes is likely to result in data corruption.
    missing_stripe_filler = "error"

    # How much stack (in KB) to reserve for use while devices suspended
    reserved_stack = 256

    # How much memory (in KB) to reserve for use while devices suspended
    reserved_memory = 8192

    # Nice value used while devices suspended
    process_priority = -18

    # If volume_list is defined, each LV is only activated if there is a
    # match against the list.
    #   "vgname" and "vgname/lvname" are matched exactly.
    #   "@tag" matches any tag set in the LV or VG.
    #   "@*" matches if any tag defined on the host is also set in the LV or VG
    #
    # volume_list = [ "vg1", "vg2/lvol1", "@tag1", "@*" ]

    # Size (in KB) of each copy operation when mirroring
    mirror_region_size = 512

    # Setting to use when there is no readahead value stored in the metadata.
    #
    # "none" - Disable readahead.
    # "auto" - Use default value chosen by kernel.
    readahead = "auto"

    # 'mirror_image_fault_policy' and 'mirror_log_fault_policy' define
    # how a device failure affecting a mirror is handled.
    # A mirror is composed of mirror images (copies) and a log.
    # A disk log ensures that a mirror does not need to be re-synced
    # (all copies made the same) every time a machine reboots or crashes.
    #
    # In the event of a failure, the specified policy will be used to
    # determine what happens:
    #
    # "remove" - Simply remove the faulty device and run without it. If
    #             the log device fails, the mirror would convert to using
    #             an in-memory log. This means the mirror will not
    #             remember its sync status across crashes/reboots and
    #             the entire mirror will be re-synced. If a
    #             mirror image fails, the mirror will convert to a
    #             non-mirrored device if there is only one remaining good
    #             copy.
    #
    # "allocate" - Remove the faulty device and try to allocate space on
    #               a new device to be a replacement for the failed device.
    #               Using this policy for the log is fast and maintains the
    #               ability to remember sync state through crashes/reboots.
    #               Using this policy for a mirror device is slow, as it
    #               requires the mirror to resynchronize the devices, but it
    #               will preserve the mirror characteristic of the device.
    #               This policy acts like "remove" if no suitable device and
    #               space can be allocated for the replacement.
    #               Currently this is not implemented properly and behaves
```

```
#           similarly to:
#
# "allocate_anywhere" - Operates like "allocate", but it does not
#           require that the new space being allocated be on a
#           device is not part of the mirror. For a log device
#           failure, this could mean that the log is allocated on
#           the same device as a mirror device. For a mirror
#           device, this could mean that the mirror device is
#           allocated on the same device as another mirror device.
#           This policy would not be wise for mirror devices
#           because it would break the redundant nature of the
#           mirror. This policy acts like "remove" if no suitable
#           device and space can be allocated for the replacement.

mirror_log_fault_policy = "allocate"
mirror_device_fault_policy = "remove"
}

#####
# Advanced section #
#####

# Metadata settings
#
# metadata {
#   Default number of copies of metadata to hold on each PV. 0, 1 or 2.
#   You might want to override it from the command line with 0
#   when running pvcreate on new PVs which are to be added to large VGs.

#   pvmetadacopies = 1

#   Approximate default size of on-disk metadata areas in sectors.
#   You should increase this if you have large volume groups or
#   you want to retain a large on-disk history of your metadata changes.

#   pvmetadatasize = 255

#   List of directories holding live copies of text format metadata.
#   These directories must not be on logical volumes!
#   It's possible to use LVM2 with a couple of directories here,
#   preferably on different (non-LV) filesystems, and with no other
#   on-disk metadata (pvmetadacopies = 0). Or this can be in
#   addition to on-disk metadata areas.
#   The feature was originally added to simplify testing and is not
#   supported under low memory situations - the machine could lock up.
#
#   Never edit any files in these directories by hand unless you
#   you are absolutely sure you know what you are doing! Use
#   the supplied toolset to make changes (e.g. vgcfgrestore).

#   dirs = [ "/etc/lvm/metadata", "/mnt/disk2/lvm/metadata2" ]
#}

# Event daemon
#
# dmeventd {
#   mirror_library is the library used when monitoring a mirror device.
#
#   "libdevmapper-event-lvm2mirror.so" attempts to recover from
#   failures. It removes failed devices from a volume group and
#   reconfigures a mirror as necessary. If no mirror library is
#   provided, mirrors are not monitored through dmeventd.

#   mirror_library = "libdevmapper-event-lvm2mirror.so"

#   snapshot_library is the library used when monitoring a snapshot device.
```

```
#  
# "libdevmapper-event-lvm2snapshot.so" monitors the filling of  
# snapshots and emits a warning through syslog, when the use of  
# snapshot exceeds 80%. The warning is repeated when 85%, 90% and  
# 95% of the snapshot are filled.  
  
snapshot_library = "libdevmapper-event-lvm2snapshot.so"  
}
```

Apéndice C. 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.

Las etiquetas de LVM son cadenas que utilizan hasta 128 de los siguientes caracteres: [A-Za-z0-9_+.-]. Las etiquetas no pueden iniciar con un guión.

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 remover 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**.

C.2. Etiquetas de host

In a cluster configuration, you can define host tags in the configuration files. If you set **hosttags = 1** in the **tags** section, a host tag is automatically defined using the machine's hostname. This allow you to use a common configuration file which can be replicated on all your machines so they hold identical copies of the file, but the behavior can differ between machines according to the hostname.

For information on the configuration files, see [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 sólo 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" ] }
```

There is a special match "@" that causes a match only if any metadata tag matches any host tag on that machine.

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/lvol2** únicamente en el host **db2**, haga lo siguiente:

1. Ejecute **lvchange --addtag @db2 vg1/lvol2** desde cualquier host en el cluster.
2. Ejecute **lvchange -ay vg1/lvol2**.

En esta solución los nombres de host se almacenan en los metadatos del grupo de volúmenes.

Apéndice D. 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 `--metadatacopies 0` del comando `pvccreate`. 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 `pvccreate`. 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 `pvccreate` 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-1DHq-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
            ]
        }
    }
}
}

```

--

Apéndice E. Historial de revisión

Revisión 1.0 Thu Jan 29 2009

Índice

A

- activating logical volumes
 - individual nodes, 41
- activating volume groups, 27
 - individual nodes, 27
 - local node only, 27
- administrative procedures, 15
- allocation
 - policy, 23
 - preventing, 22
- archive file, 16, 28

B

- backup
 - file, 16
 - metadata, 16, 28
- backup file, 28
- block device
 - scanning, 21

C

- cache file
 - building, 26
- cluster environment, 3, 15
- CLVM
 - definition, 3
- clvmd daemon, 3
- command line units, 19
- configuration examples, 53
- creating
 - logical volume, 30
 - logical volume, example, 53
 - LVM volumes in a cluster, 15
 - physical volumes, 20
 - striped logical volume, example, 54
 - volume group, clustered, 24
 - volume groups, 23
- creating LVM volumes
 - overview, 15

D

- data relocation, online, 41
- deactivating volume groups, 27
 - exclusive on one node, 27
 - local node only, 27
- device numbers
 - major, 34
 - minor, 34
 - persistent, 34
- device path names, 19

- device scan filters, 40
- device size, maximum, 24
- device special file directory, 24
- display
 - sorting output, 49
- displaying
 - logical volumes, 35, 47
 - physical volumes, 22, 44
 - volume groups, 25, 46

E

- extent
 - allocation, 23
 - definition, 8, 23

F

- failed devices
 - displaying, 61
- feedback, viii, viii
- file system
 - growing on a logical volume, 16
- filters, 40

G

- growing file system
 - logical volume, 16

H

- help display, 20

I

- initializing
 - partitions, 21
 - physical volumes, 21
- Insufficient Free Extents message, 67

L

- linear logical volume
 - converting to mirrored, 33
 - creation, 30
 - definition, 9
- logging, 17
- logical volume
 - administration, general, 30
 - changing parameters, 34
 - creation, 30
 - creation example, 53
 - definition, 1, 9
 - displaying, 35, 42, 47
 - exclusive access, 41
 - extending, 36
 - growing, 36
 - linear, 30

- local access, 41
- lvs display arguments, 47
- mirrored, 32
- reducing, 38
- removing, 35
- renaming, 35
- resizing, 34
- shrinking, 38
- snapshot, 38
- striped, 31
- lvchange command, 34
- lvconvert command, 33
- lvcreate command, 30
- lvdisplay command, 35
- lvextend command, 36
- LVM
 - architecture overview, 2
 - clustered, 3
 - components, 2, 7
 - custom report format, 42
 - directory structure, 24
 - help, 20
 - history, 2
 - label, 7
 - logging, 17
 - logical volume administration, 30
 - physical volume administration, 20
 - physical volume, definition, 7
 - volume group, definition, 8
- LVM1, 2
- LVM2, 2
- lvmdiskscan command, 21
- lvreduce command, 34, 38
- lvremove command, 35
- lvrename command, 35
- lvs command, 42, 47
 - display arguments, 47
- lvscan command, 36

M

- man page display, 20
- metadata
 - backup, 16, 28
 - recovery, 65
- mirrored logical volume
 - converting to linear, 33
 - creation, 32
 - definition, 11
 - failure recovery, 62
 - reconfiguration, 33

O

- online data relocation, 41

P

- partition type, setting, 21
- partitions
 - multiple, 8
- path names, 19
- persistent device numbers, 34
- physical extent
 - preventing allocation, 22
- physical volume
 - adding to a volume group, 25
 - administration, general, 20
 - creating, 20
 - definition, 7
 - display, 44
 - displaying, 22, 42
 - illustration, 7
 - initializing, 21
 - layout, 7
 - pvs display arguments, 44
 - recovery, 66
 - removing, 23
 - removing from volume group, 26
 - removing lost volume, 67
 - resizing, 23
- pvdiskscan command, 22
- pvmove command, 41
- pvremove command, 23
- pvresize command, 23
- pvs command, 42
 - display arguments, 44
- pvscan command, 22

R

- removing
 - disk from a logical volume, 58
 - logical volume, 35
 - physical volumes, 23
- renaming
 - logical volume, 35
 - volume group, 28
- report format, LVM devices, 42
- resizing
 - logical volume, 34
 - physical volume, 23

S

- scanning
 - block devices, 21
- scanning devices, filters, 40
- snapshot logical volume
 - creation, 38
- snapshot volume
 - definition, 12

- striped logical volume
 - creation, 31
 - creation example, 54
 - definition, 10
 - extending, 37
 - growing, 37

T

- troubleshooting, 61

U

- units, command line, 19

V

- verbose output, 19
- vgcfbackup command, 28
- vgcfrestore command, 28
- vgchange command, 27
- vgcreate command, 23, 24
- vgdisplay command, 25
- vgexport command, 29
- vgextend command, 25
- vgimport command, 29
- vgmerge command, 28
- vgmknodes command, 29
- vgreduce command, 26
- vgrename command, 28
- vgs command, 42
 - display arguments, 46
- vgscan command, 26
- vgsplit command, 28
- volume group
 - activating, 27
 - administration, general, 23
 - changing parameters, 27
 - combining, 28
 - creating, 23
 - creating in a cluster, 24
 - deactivating, 27
 - definition, 8
 - displaying, 25, 42, 46
 - extending, 25
 - growing, 25
 - merging, 28
 - moving between systems, 29
 - reducing, 26
 - removing, 27
 - renaming, 28
 - shrinking, 26
 - splitting, 28
 - example procedure, 55
 - vgs display arguments, 46

