



# Red Hat Enterprise Linux 8

## Configuración y gestión de volúmenes lógicos

Una guía para la configuración y gestión de volúmenes lógicos LVM



# Red Hat Enterprise Linux 8 Configuración y gestión de volúmenes lógicos

---

Una guía para la configuración y gestión de volúmenes lógicos LVM

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

## Legal Notice

Copyright © 2021 | You need to change the HOLDER entity in the en-US/Configuring\_and\_managing\_logical\_volumes.ent file |.

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, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

## Resumen

Esta colección de documentación proporciona instrucciones sobre cómo gestionar volúmenes lógicos LVM en Red Hat Enterprise Linux 8.

## Table of Contents

<b>HACER QUE EL CÓDIGO ABIERTO SEA MÁS INCLUSIVO</b> .....	<b>6</b>
<b>PROPORCIONAR COMENTARIOS SOBRE LA DOCUMENTACIÓN DE RED HAT</b> .....	<b>7</b>
<b>CAPÍTULO 1. VOLÚMENES LÓGICOS</b> .....	<b>8</b>
1.1. VISIÓN GENERAL DE LA ARQUITECTURA LVM	8
1.2. VOLÚMENES FÍSICOS	9
1.2.1. Disposición del volumen físico LVM	10
1.2.2. Múltiples particiones en un disco	10
1.3. GRUPOS DE VOLUMEN	11
1.4. VOLÚMENES LÓGICOS LVM	11
1.4.1. Volúmenes lineales	11
1.4.2. Volúmenes lógicos rayados	13
1.4.3. Volúmenes lógicos RAID	14
1.4.4. Volúmenes lógicos de aprovisionamiento fino (volúmenes finos)	15
1.4.5. Volúmenes de instantáneas	16
1.4.6. Volúmenes instantáneos de aprovisionamiento ligero	17
1.4.7. Volúmenes de caché	18
<b>CAPÍTULO 2. GESTIÓN DEL ALMACENAMIENTO LOCAL MEDIANTE LOS ROLES DE SISTEMA DE RHEL</b>	<b>19</b>
2.1. INTRODUCCIÓN A LA FUNCIÓN DE ALMACENAMIENTO	19
2.2. PARÁMETROS QUE IDENTIFICAN UN DISPOSITIVO DE ALMACENAMIENTO EN EL ROL DE SISTEMA DE ALMACENAMIENTO	19
2.3. EJEMPLO DE PLAYBOOK DE ANSIBLE PARA CREAR UN SISTEMA DE ARCHIVOS XFS EN UN DISPOSITIVO DE BLOQUES	20
2.4. EJEMPLO DE PLAYBOOK DE ANSIBLE PARA MONTAR PERSISTENTEMENTE UN SISTEMA DE ARCHIVOS	21
2.5. EJEMPLO DE LIBRO DE JUGADAS DE ANSIBLE PARA GESTIONAR VOLÚMENES LÓGICOS	21
2.6. EJEMPLO DE LIBRO DE JUGADAS DE ANSIBLE PARA ACTIVAR EL DESCARTE DE BLOQUES EN LÍNEA	22
2.7. EJEMPLO DE PLAYBOOK ANSIBLE PARA CREAR Y MONTAR UN SISTEMA DE ARCHIVOS EXT4	23
2.8. EJEMPLO DE PLAYBOOK DE ANSIBLE PARA CREAR Y MONTAR UN SISTEMA DE ARCHIVOS EXT3	23
2.9. CONFIGURACIÓN DE UN VOLUMEN RAID MEDIANTE EL ROL DE SISTEMA DE ALMACENAMIENTO	24
2.10. CONFIGURACIÓN DE UN POOL LVM CON RAID UTILIZANDO EL ROL DE SISTEMA DE ALMACENAMIENTO	25
2.11. CREACIÓN DE UN VOLUMEN ENCRYPTADO LUKS UTILIZANDO EL ROL DE ALMACENAMIENTO	27
<b>CAPÍTULO 3. DESPLIEGUE DE LVM</b> .....	<b>29</b>
3.1. CREACIÓN DE UN VOLUMEN FÍSICO LVM	29
3.2. CREACIÓN DE UN GRUPO DE VOLÚMENES LVM	29
3.3. CREACIÓN DE UN VOLUMEN LÓGICO LVM	30
<b>CAPÍTULO 4. CONFIGURACIÓN DE VOLÚMENES LÓGICOS LVM</b> .....	<b>31</b>
4.1. USO DE LOS COMANDOS DE LA CLI	31
Especificación de unidades en un argumento de línea de comandos	31
Especificación de grupos de volúmenes y volúmenes lógicos	31
Aumentar la verbosidad de la salida	31
Visualización de la ayuda para los comandos de la CLI de LVM	32
4.2. CREACIÓN DE UN VOLUMEN LÓGICO LVM EN TRES DISCOS	32
4.3. CREACIÓN DE UN VOLUMEN LÓGICO RAID0 (RAYADO)	34
4.4. RENOMBRAR VOLÚMENES LÓGICOS LVM	36
4.5. ELIMINACIÓN DE UN DISCO DE UN VOLUMEN LÓGICO	36
4.5.1. Mover extensiones a volúmenes físicos existentes	36

4.5.2. Mover extensiones a un nuevo disco	37
4.6. CONFIGURACIÓN DE NÚMEROS DE DISPOSITIVOS PERSISTENTES	38
4.7. ESPECIFICACIÓN DEL TAMAÑO DE EXTENSIÓN DE LVM	39
4.8. GESTIÓN DE LOS VOLÚMENES LÓGICOS LVM MEDIANTE LOS ROLES DE SISTEMA DE RHEL	39
4.8.1. Ejemplo de libro de jugadas de Ansible para gestionar volúmenes lógicos	39
4.8.2. Recursos adicionales	40
4.9. ELIMINACIÓN DE VOLÚMENES LÓGICOS LVM	40
<b>CAPÍTULO 5. MODIFICAR EL TAMAÑO DE UN VOLUMEN LÓGICO</b>	<b>42</b>
5.1. CRECIMIENTO DE LOS VOLÚMENES LÓGICOS	42
5.2. CRECIMIENTO DE UN SISTEMA DE ARCHIVOS EN UN VOLUMEN LÓGICO	42
5.3. REDUCCIÓN DE VOLÚMENES LÓGICOS	43
5.4. AMPLIACIÓN DE UN VOLUMEN LÓGICO RAYADO	44
<b>CAPÍTULO 6. GESTIÓN DE GRUPOS DE VOLÚMENES LVM</b>	<b>46</b>
6.1. GRUPOS DE VOLUMEN	46
6.2. VISUALIZACIÓN DE LOS GRUPOS DE VOLUMEN	46
6.3. COMBINACIÓN DE GRUPOS DE VOLUMEN	47
6.4. DIVIDIR UN GRUPO DE VOLÚMENES	47
6.5. RENOMBRAR GRUPOS DE VOLÚMENES LVM	48
6.6. MOVER UN GRUPO DE VOLÚMENES A OTRO SISTEMA	49
6.7. ELIMINACIÓN DE VOLÚMENES FÍSICOS DE UN GRUPO DE VOLÚMENES	50
6.8. ELIMINACIÓN DE GRUPOS DE VOLÚMENES LVM	51
6.9. RECURSOS ADICIONALES	51
<b>CAPÍTULO 7. GESTIÓN DE VOLÚMENES FÍSICOS LVM</b>	<b>52</b>
7.1. BÚSQUEDA DE DISPOSITIVOS DE BLOQUE PARA UTILIZARLOS COMO VOLÚMENES FÍSICOS	52
7.2. ESTABLECER EL TIPO DE PARTICIÓN PARA UN VOLUMEN FÍSICO	52
7.3. CAMBIAR EL TAMAÑO DE UN VOLUMEN FÍSICO LVM	53
7.4. ELIMINACIÓN DE VOLÚMENES FÍSICOS	53
7.5. AÑADIR VOLÚMENES FÍSICOS A UN GRUPO DE VOLÚMENES	53
7.6. ELIMINACIÓN DE VOLÚMENES FÍSICOS DE UN GRUPO DE VOLÚMENES	53
<b>CAPÍTULO 8. VISUALIZACIÓN DE LOS COMPONENTES DE LVM</b>	<b>55</b>
8.1. VISUALIZACIÓN DE LA INFORMACIÓN DE LVM CON EL COMANDO LVM	55
8.2. VISUALIZACIÓN DE LOS VOLÚMENES FÍSICOS	55
8.3. VISUALIZACIÓN DE LOS GRUPOS DE VOLUMEN	56
8.4. VISUALIZACIÓN DE VOLÚMENES LÓGICOS	57
<b>CAPÍTULO 9. INFORMES PERSONALIZADOS PARA LVM</b>	<b>58</b>
9.1. CONTROLAR EL FORMATO DE LA PANTALLA LVM	58
9.2. CAMPOS DE VISUALIZACIÓN DEL OBJETO LVM	59
9.3. CLASIFICACIÓN DE LOS INFORMES LVM	70
9.4. ESPECIFICACIÓN DE LAS UNIDADES PARA LA VISUALIZACIÓN DE UN INFORME LVM	70
9.5. VISUALIZACIÓN DE LA SALIDA DEL COMANDO LVM EN FORMATO JSON	72
9.6. VISUALIZACIÓN DEL REGISTRO DE COMANDOS LVM	72
<b>CAPÍTULO 10. CONFIGURACIÓN DE VOLÚMENES LÓGICOS RAID</b>	<b>74</b>
10.1. VOLÚMENES LÓGICOS RAID	74
10.2. NIVELES RAID Y SOPORTE LINEAL	74
10.3. TIPOS DE SEGMENTOS RAID DE LVM	76
10.4. CREACIÓN DE VOLÚMENES LÓGICOS RAID	78
10.5. CREACIÓN DE UN VOLUMEN LÓGICO RAID (RAYADO)	79
10.6. USO DE LA INTEGRIDAD DE DM CON RAID LV	81
10.6.1. Proteger los datos con la integridad del DM	81

10.6.1.1. Consideraciones al añadir la integridad del DM	81
10.6.2. Creación de un RAID LV con integridad DM	82
10.6.3. Añadir integridad DM a un RAID LV existente	82
10.6.4. Eliminación de la integridad de un RAID LV	83
10.6.5. Visualización de la información sobre la integridad del DM	83
10.6.6. Recursos adicionales	85
10.7. CONTROL DE LA VELOCIDAD DE INICIALIZACIÓN DE LOS VOLÚMENES RAID	85
10.8. CONVERTIR UN DISPOSITIVO LINEAL EN UN DISPOSITIVO RAID	86
10.9. CONVERTIR UN VOLUMEN LÓGICO RAID1 DE LVM EN UN VOLUMEN LÓGICO LINEAL DE LVM	86
10.10. CONVERTIR UN DISPOSITIVO LVM EN ESPEJO EN UN DISPOSITIVO RAID1	87
10.11. REDIMENSIONAR UN VOLUMEN LÓGICO RAID	88
10.12. CAMBIO DEL NÚMERO DE IMÁGENES EN UN DISPOSITIVO RAID1 EXISTENTE	88
10.13. DIVIDIR UNA IMAGEN RAID COMO UN VOLUMEN LÓGICO SEPARADO	90
10.14. DIVIDIR Y FUSIONAR UNA IMAGEN RAID	91
10.15. ESTABLECIMIENTO DE UNA POLÍTICA DE FALLOS EN EL RAID	93
10.15.1. La política de fallos de RAID asignada	93
10.15.2. El aviso de la política de fallos del RAID	94
10.16. SUSTITUCIÓN DE UN DISPOSITIVO RAID EN UN VOLUMEN LÓGICO	95
10.16.1. Sustitución de un dispositivo RAID que no ha fallado	95
10.16.2. Dispositivos fallidos en RAID LVM	98
10.16.3. Recuperación de un dispositivo RAID fallido en un volumen lógico	98
10.16.4. Sustitución de un dispositivo RAID fallido en un volumen lógico	98
10.17. COMPROBACIÓN DE LA COHERENCIA DE LOS DATOS EN UN VOLUMEN LÓGICO RAID (RAID SCRUBBING)	100
10.18. CONVERSIÓN DE UN NIVEL RAID (ABSORCIÓN DE RAID)	102
10.19. CAMBIO DE ATRIBUTOS DE UN VOLUMEN RAID (RAID RESHAPE)	102
10.20. CONTROL DE LAS OPERACIONES DE E/S EN UN VOLUMEN LÓGICO RAID1	102
10.21. CAMBIAR EL TAMAÑO DE LA REGIÓN EN UN VOLUMEN LÓGICO RAID	103
<b>CAPÍTULO 11. VOLÚMENES LÓGICOS INSTANTÁNEOS</b>	<b>104</b>
11.1. VOLÚMENES DE INSTANTÁNEAS	104
11.2. CREACIÓN DE VOLÚMENES DE INSTANTÁNEAS	105
11.3. FUSIÓN DE VOLÚMENES DE INSTANTÁNEAS	107
<b>CAPÍTULO 12. CREACIÓN Y GESTIÓN DE VOLÚMENES LÓGICOS DE APROVISIONAMIENTO LIGERO (VOLÚMENES LIGEROS)</b>	<b>108</b>
12.1. VOLÚMENES LÓGICOS DE APROVISIONAMIENTO FINO (VOLÚMENES FINOS)	108
12.2. CREACIÓN DE VOLÚMENES LÓGICOS DE APROVISIONAMIENTO LIGERO	108
12.3. VOLÚMENES INSTANTÁNEOS DE APROVISIONAMIENTO LIGERO	111
12.4. CREACIÓN DE VOLÚMENES DE INSTANTÁNEAS CON APROVISIONAMIENTO LIGERO	112
12.5. SEGUIMIENTO Y VISUALIZACIÓN DE LOS VOLÚMENES FINOS DE INSTANTÁNEAS QUE SE HAN ELIMINADO	114
<b>CAPÍTULO 13. ACTIVACIÓN DE LA CACHÉ PARA MEJORAR EL RENDIMIENTO DEL VOLUMEN LÓGICO</b>	<b>118</b>
13.1. MÉTODOS DE ALMACENAMIENTO EN CACHÉ EN LVM	118
13.2. COMPONENTES DE CACHÉ LVM	118
13.3. ACTIVACIÓN DE LA CACHÉ DM-CACHE PARA UN VOLUMEN LÓGICO	118
13.4. ACTIVACIÓN DE LA CACHÉ DM-CACHE CON UN CACHEPOOL PARA UN VOLUMEN LÓGICO	120
13.5. ACTIVACIÓN DE LA CACHÉ DM-WRITECACHE PARA UN VOLUMEN LÓGICO	121
13.6. DESACTIVACIÓN DEL ALMACENAMIENTO EN CACHÉ DE UN VOLUMEN LÓGICO	123
<b>CAPÍTULO 14. ACTIVACIÓN DEL VOLUMEN LÓGICO</b>	<b>125</b>
14.1. CONTROL DE LA AUTOACTIVACIÓN DE LOS VOLÚMENES LÓGICOS	125
14.2. CONTROL DE LA ACTIVACIÓN DEL VOLUMEN LÓGICO	126

14.3. ACTIVACIÓN DE VOLÚMENES LÓGICOS COMPARTIDOS	127
14.4. ACTIVACIÓN DE UN VOLUMEN LÓGICO CON DISPOSITIVOS AUSENTES	127
<b>CAPÍTULO 15. CONTROL DE LA EXPLORACIÓN DE DISPOSITIVOS LVM</b>	<b>128</b>
15.1. EL FILTRO DE DISPOSITIVOS LVM	128
15.2. EJEMPLOS DE CONFIGURACIONES DE FILTROS DE DISPOSITIVOS LVM	128
15.3. APLICACIÓN DE UNA CONFIGURACIÓN DE FILTRO DE DISPOSITIVOS LVM	129
<b>CAPÍTULO 16. ESTRATIFICACIÓN DE VOLÚMENES FÍSICOS LVM SOBRE VOLÚMENES LÓGICOS</b>	<b>130</b>
<b>CAPÍTULO 17. CONTROL DE LA ASIGNACIÓN DE LVM</b>	<b>131</b>
17.1. POLÍTICAS DE ASIGNACIÓN DE LVM	131
17.2. IMPEDIR LA ASIGNACIÓN EN UN VOLUMEN FÍSICO	132
17.3. AMPLIACIÓN DE UN VOLUMEN LÓGICO CON LA POLÍTICA DE ASIGNACIÓN CLING	132
17.4. DIFERENCIACIÓN DE LOS OBJETOS RAID DE LVM MEDIANTE ETIQUETAS	134
<b>CAPÍTULO 18. AGRUPACIÓN DE OBJETOS LVM CON ETIQUETAS</b>	<b>136</b>
18.1. ETIQUETAS DE OBJETOS LVM	136
18.2. LISTADO DE ETIQUETAS LVM	136
18.3. AÑADIR ETIQUETAS DE OBJETOS LVM	136
18.4. ELIMINACIÓN DE ETIQUETAS DE OBJETOS LVM	137
18.5. DEFINICIÓN DE ETIQUETAS DE HOST LVM	137
18.6. CONTROL DE LA ACTIVACIÓN DE VOLÚMENES LÓGICOS CON ETIQUETAS	138
<b>CAPÍTULO 19. SOLUCIÓN DE PROBLEMAS DE LVM</b>	<b>139</b>
19.1. RECOGIDA DE DATOS DE DIAGNÓSTICO EN LVM	139
19.2. VISUALIZACIÓN DE INFORMACIÓN SOBRE DISPOSITIVOS LVM FALLIDOS	140
19.3. ELIMINACIÓN DE VOLÚMENES FÍSICOS LVM PERDIDOS DE UN GRUPO DE VOLÚMENES	141
19.4. RECUPERACIÓN DE UN VOLUMEN FÍSICO LVM CON METADATOS DAÑADOS	142
19.4.1. Descubrir que un volumen LVM tiene metadatos perdidos o corruptos	142
19.4.2. Encontrar los metadatos de un volumen físico LVM perdido	142
19.4.3. Restauración de metadatos en un volumen físico LVM	143
19.5. REEMPLAZAR UN VOLUMEN FÍSICO LVM PERDIDO	145
19.5.1. Encontrar los metadatos de un volumen físico LVM perdido	145
19.5.2. Restauración de metadatos en un volumen físico LVM	146
19.6. SOLUCIÓN DE PROBLEMAS DE RAID LVM	147
19.6.1. Comprobación de la coherencia de los datos en un volumen lógico RAID (RAID scrubbing)	147
19.6.2. Dispositivos fallidos en RAID LVM	149
19.6.3. Recuperación de un dispositivo RAID fallido en un volumen lógico	149
19.6.4. Sustitución de un dispositivo RAID fallido en un volumen lógico	150
19.7. SOLUCIÓN DE PROBLEMAS DE INSUFICIENCIA DE EXTENSIONES LIBRES PARA UN VOLUMEN LÓGICO	151
19.7.1. Grupos de volumen	151
19.7.2. Errores de redondeo en la salida de LVM	152
19.7.3. Evitar el error de redondeo al crear un volumen LVM	152
19.8. SOLUCIÓN DE PROBLEMAS DE ADVERTENCIAS DE VOLÚMENES FÍSICOS DUPLICADOS PARA DISPOSITIVOS LVM MULTIPATHED	153
19.8.1. Causa de la duplicación de los avisos de FV	153
19.8.2. Casos de duplicación de avisos de FV	154
19.8.3. El filtro de dispositivos LVM	154
19.8.4. Ejemplo de filtros de dispositivos LVM que evitan los avisos de PV duplicados	155
19.8.5. Aplicación de una configuración de filtro de dispositivos LVM	155
19.8.6. Recursos adicionales	156



## HACER QUE EL CÓDIGO ABIERTO SEA MÁS INCLUSIVO

Red Hat se compromete a sustituir el lenguaje problemático en nuestro código, documentación y propiedades web. Estamos empezando con estos cuatro términos: maestro, esclavo, lista negra y lista blanca. Debido a la enormidad de este esfuerzo, estos cambios se implementarán gradualmente a lo largo de varias versiones próximas. Para más detalles, consulte [el mensaje de nuestro CTO Chris Wright](#) .

## PROPORCIONAR COMENTARIOS SOBRE LA DOCUMENTACIÓN DE RED HAT

Agradecemos su opinión sobre nuestra documentación. Por favor, díganos cómo podemos mejorarla. Para ello:

- Para comentarios sencillos sobre pasajes concretos:
  1. Asegúrese de que está viendo la documentación en el formato *Multi-page HTML*. Además, asegúrese de ver el botón **Feedback** en la esquina superior derecha del documento.
  2. Utilice el cursor del ratón para resaltar la parte del texto que desea comentar.
  3. Haga clic en la ventana emergente **Add Feedback** que aparece debajo del texto resaltado.
  4. Siga las instrucciones mostradas.
- Para enviar comentarios más complejos, cree un ticket de Bugzilla:
  1. Vaya al sitio web [de Bugzilla](#).
  2. Como componente, utilice **Documentation**.
  3. Rellene el campo **Description** con su sugerencia de mejora. Incluya un enlace a la(s) parte(s) pertinente(s) de la documentación.
  4. Haga clic en **Submit Bug**.

# CAPÍTULO 1. VOLÚMENES LÓGICOS

La gestión de volúmenes crea una capa de abstracción sobre el almacenamiento físico, lo que permite crear volúmenes de almacenamiento lógico. Esto proporciona una flexibilidad mucho mayor en varios aspectos que el uso del almacenamiento físico directamente. Además, la configuración del almacenamiento de hardware se oculta al software, por lo que se puede redimensionar y mover sin detener las aplicaciones ni desmontar los sistemas de archivos. Esto puede reducir los costes operativos.

Los volúmenes lógicos ofrecen las siguientes ventajas sobre el uso directo del almacenamiento físico:

- **Capacidad flexible**  
Cuando se utilizan volúmenes lógicos, los sistemas de archivos pueden extenderse a través de varios discos, ya que se pueden agregar discos y particiones en un único volumen lógico.
- **Grupos de almacenamiento redimensionables**  
Puede ampliar los volúmenes lógicos o reducir su tamaño con simples comandos de software, sin necesidad de reformatear y reparticionar los dispositivos de disco subyacentes.
- **Reubicación de datos en línea**  
Para implementar subsistemas de almacenamiento más nuevos, más rápidos o más resistentes, puede mover los datos mientras el sistema está activo. Los datos se pueden reorganizar en los discos mientras éstos están en uso. Por ejemplo, puede vaciar un disco intercambiable en caliente antes de retirarlo.
- **Una cómoda denominación de los dispositivos**  
Los volúmenes de almacenamiento lógico se pueden gestionar en grupos definidos por el usuario y con nombres personalizados.
- **Separación de discos**  
Puede crear un volumen lógico que reparta los datos entre dos o más discos. Esto puede aumentar drásticamente el rendimiento.
- **Volúmenes en espejo**  
Los volúmenes lógicos proporcionan una forma conveniente de configurar una réplica para sus datos.
- **Instantáneas de volumen**  
Mediante el uso de volúmenes lógicos, puede tomar instantáneas de dispositivos para realizar copias de seguridad consistentes o para probar el efecto de los cambios sin afectar a los datos reales.
- **Volúmenes finos**  
Los volúmenes lógicos pueden tener un aprovisionamiento ligero. Esto le permite crear volúmenes lógicos que son más grandes que los extensiones disponibles.
- **Volúmenes de caché**  
Un volumen lógico de caché utiliza un pequeño volumen lógico formado por dispositivos de bloques rápidos (como las unidades SSD) para mejorar el rendimiento de un volumen lógico más grande y lento, almacenando los bloques más utilizados en el volumen lógico más pequeño y rápido.

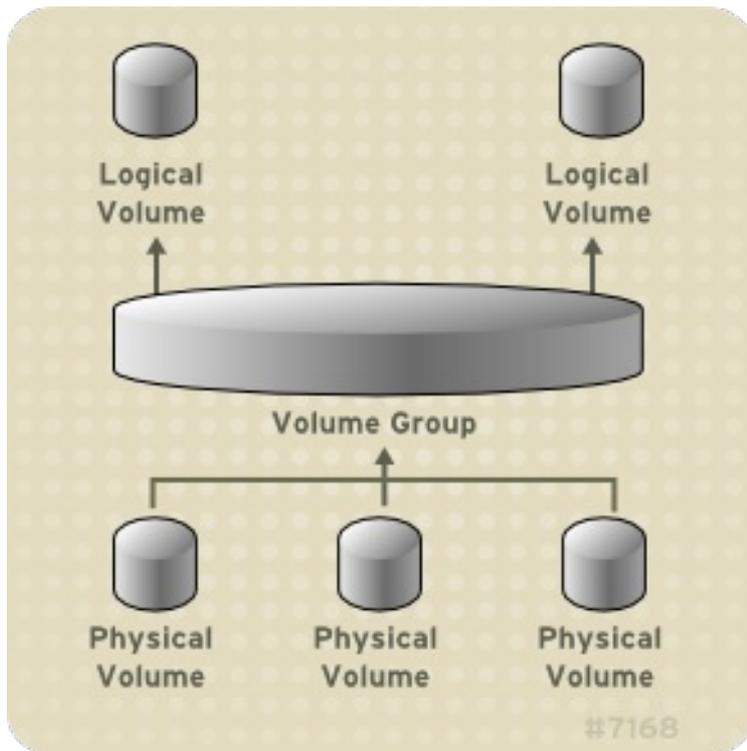
## 1.1. VISIÓN GENERAL DE LA ARQUITECTURA LVM

La unidad de almacenamiento físico subyacente de un volumen lógico LVM es un dispositivo de bloque como una partición o un disco completo. Este dispositivo se inicializa como un LVM *physical volume* (PV).

Para crear un volumen lógico LVM, los volúmenes físicos se combinan en un *volume group* (VG). Esto crea una reserva de espacio en disco a partir de la cual se pueden asignar volúmenes lógicos LVM (LVs). Este proceso es análogo a la forma en que los discos se dividen en particiones. Un volumen lógico es utilizado por los sistemas de archivos y las aplicaciones (como las bases de datos).

Figura 1.1, “Componentes del volumen lógico LVM” muestra los componentes de un volumen lógico LVM simple:

Figura 1.1. Componentes del volumen lógico LVM



## 1.2. VOLÚMENES FÍSICOS

La unidad de almacenamiento físico subyacente de un volumen lógico LVM es un dispositivo de bloques, como una partición o un disco completo. Para utilizar el dispositivo para un volumen lógico LVM, el dispositivo debe ser inicializado como un volumen físico (PV). La inicialización de un dispositivo de bloque como volumen físico coloca una etiqueta cerca del inicio del dispositivo.

Por defecto, la etiqueta LVM se coloca en el segundo sector de 512 bytes. Puede sobrescribir este valor por defecto colocando la etiqueta en cualquiera de los primeros 4 sectores cuando cree el volumen físico. Esto permite que los volúmenes LVM coexistan con otros usuarios de estos sectores, si es necesario.

Una etiqueta LVM proporciona una correcta identificación y ordenación de dispositivos para un dispositivo físico, ya que los dispositivos pueden aparecer en cualquier orden al arrancar el sistema. Una etiqueta LVM permanece persistente a través de los reinicios y a través de un clúster.

La etiqueta LVM identifica el dispositivo como un volumen físico LVM. Contiene un identificador único aleatorio (el UUID) para el volumen físico. También almacena el tamaño del dispositivo de bloque en bytes, y registra dónde se almacenarán los metadatos LVM en el dispositivo.

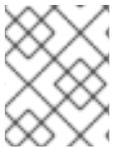
Los metadatos LVM contienen los detalles de configuración de los grupos de volúmenes LVM en su

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

Actualmente LVM permite almacenar 0, 1 o 2 copias idénticas de sus metadatos en cada volumen físico. El valor por defecto es 1 copia. Una vez que configure el número de copias de metadatos en el volumen físico, no podrá cambiar ese número posteriormente. La primera copia se almacena al principio del dispositivo, poco después de la etiqueta. Si hay una segunda copia, se coloca al final del dispositivo. Si accidentalmente sobrescribes el área al principio de tu disco escribiendo en un disco diferente al que pretendías, una segunda copia de los metadatos al final del dispositivo te permitirá recuperar los metadatos.

### 1.2.1. Disposición del volumen físico LVM

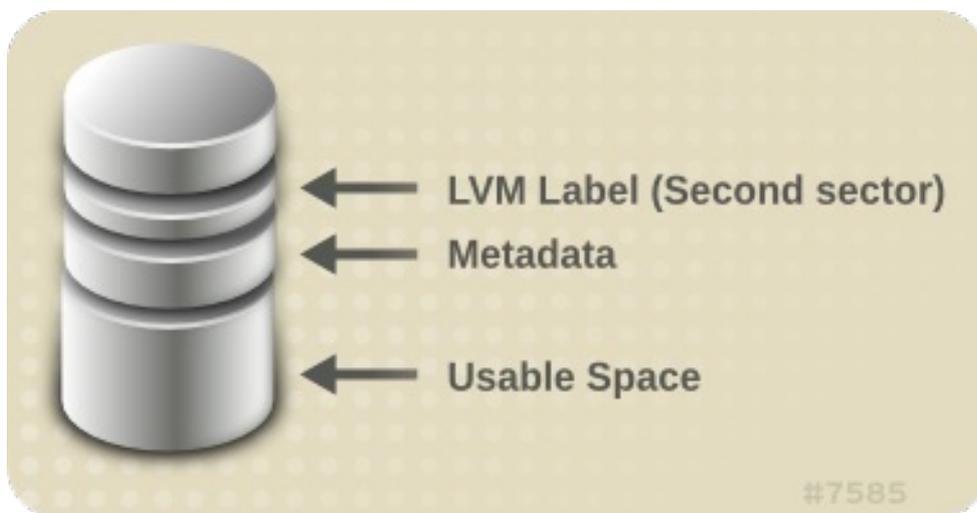
Figura 1.2, “Disposición del volumen físico” muestra la disposición de un volumen físico LVM. La etiqueta LVM está en el segundo sector, seguido por el área de metadatos, seguido por el espacio utilizable en el dispositivo.



#### NOTA

En el núcleo de Linux (y a lo largo de este documento), se considera que los sectores tienen un tamaño de 512 bytes.

Figura 1.2. Disposición del volumen físico



### 1.2.2. Múltiples particiones en un disco

LVM le permite crear volúmenes físicos a partir de particiones de disco. Red Hat recomienda que cree una sola partición que cubra todo el disco para etiquetarlo como un volumen físico LVM por las siguientes razones:

- **Conveniencia administrativa**  
Es más fácil hacer un seguimiento del hardware de un sistema si cada disco real sólo aparece una vez. Esto es particularmente cierto si un disco falla. Además, múltiples volúmenes físicos en un solo disco pueden causar una advertencia del kernel sobre tipos de partición desconocidos en el arranque.
- **Rendimiento de las franjas**  
LVM no puede decir que dos volúmenes físicos están en el mismo disco físico. Si crea un volumen lógico rayado cuando dos volúmenes físicos están en el mismo disco físico, las rayas

podrían estar en diferentes particiones del mismo disco. Esto resultaría en una disminución del rendimiento en lugar de un aumento.

Aunque no se recomienda, puede haber circunstancias específicas en las que necesite dividir un disco en volúmenes físicos LVM separados. Por ejemplo, en un sistema con pocos discos puede ser necesario mover los datos alrededor de las particiones cuando se está migrando un sistema existente a volúmenes LVM. Además, si tienes un disco muy grande y quieres tener más de un grupo de volúmenes para propósitos administrativos, entonces es 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 de especificar qué particiones se incluirán en un volumen lógico al crear volúmenes rayados.

### 1.3. GRUPOS DE VOLUMEN

Los volúmenes físicos se combinan en grupos de volúmenes (VG). Esto crea una reserva de espacio en disco a partir de la cual se pueden asignar volúmenes lógicos.

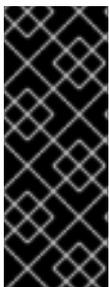
Dentro de un grupo de volúmenes, el espacio de disco disponible para la asignación se divide en unidades de tamaño fijo llamadas extensiones. Una extensión es la unidad más pequeña de espacio que se puede asignar. Dentro de un volumen físico, los extents se denominan extents físicos.

Un volumen lógico se asigna en extensiones lógicas del mismo tamaño que las extensiones físicas. El tamaño de la extensión es, por tanto, el mismo para todos los volúmenes lógicos del grupo de volúmenes. El grupo de volúmenes asigna las extensiones lógicas a extensiones físicas.

### 1.4. VOLÚMENES LÓGICOS LVM

En LVM, un grupo de volúmenes se divide en volúmenes lógicos. Un administrador puede aumentar o reducir los volúmenes lógicos sin destruir los datos, a diferencia de las particiones de disco estándar. Si los volúmenes físicos de un grupo de volúmenes están en unidades separadas o en matrices RAID, los administradores también pueden repartir un volumen lógico entre los dispositivos de almacenamiento.

Puede perder datos si reduce un volumen lógico a una capacidad inferior a la que requieren los datos del volumen. Para garantizar la máxima flexibilidad, cree volúmenes lógicos que satisfagan sus necesidades actuales y deje el exceso de capacidad de almacenamiento sin asignar. Puede ampliar de forma segura los volúmenes lógicos para utilizar el espacio no asignado, en función de sus necesidades.



#### IMPORTANTE

En los sistemas AMD, Intel, ARM y servidores IBM Power Systems, el gestor de arranque no puede leer volúmenes LVM. Debes hacer una partición de disco estándar, no LVM, para tu partición /boot. En IBM Z, el cargador de arranque zipl soporta /boot en volúmenes lógicos LVM con mapeo lineal. Por defecto, el proceso de instalación siempre crea las particiones / y swap dentro de volúmenes LVM, con una partición /boot separada en un volumen físico.

Las siguientes secciones describen los diferentes tipos de volúmenes lógicos.

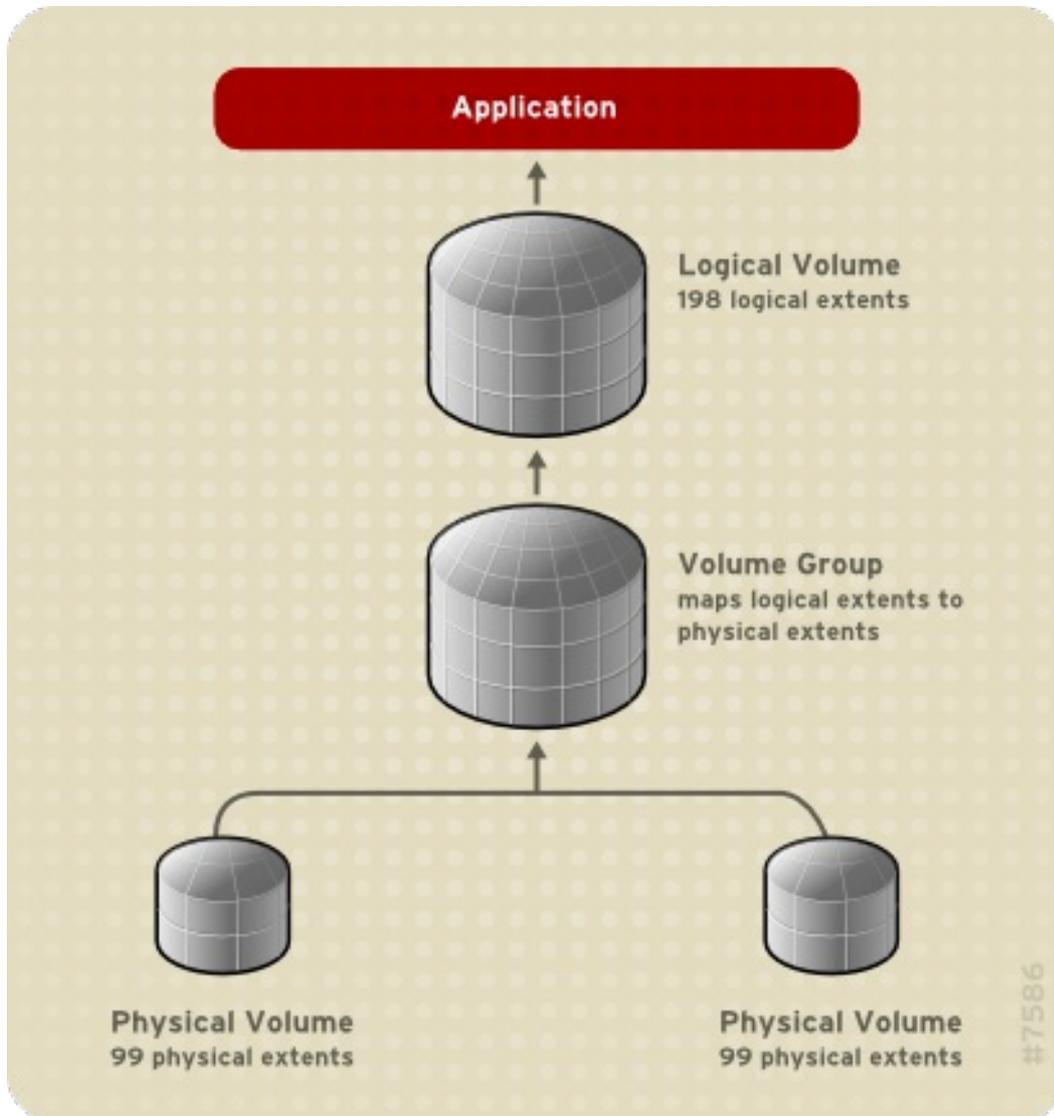
#### 1.4.1. Volúmenes lineales

Un volumen lineal agrega el espacio de uno o más volúmenes físicos en un volumen lógico. Por ejemplo, si tiene dos discos de 60 GB, puede crear un volumen lógico de 120 GB. El almacenamiento físico se concatena.

La creación de un volumen lineal asigna un rango de extensiones físicas a un área de un volumen lógico

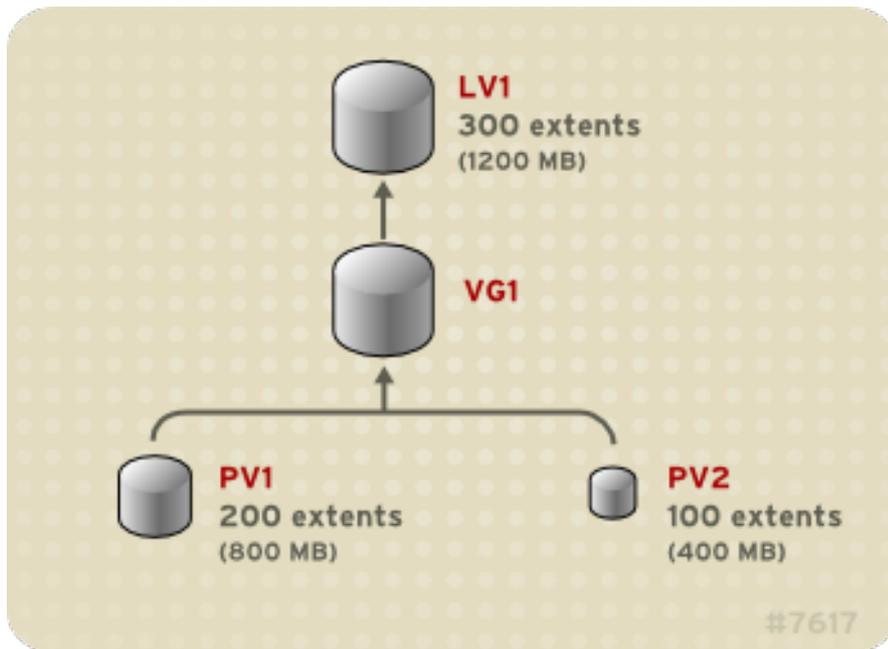
en orden. Por ejemplo, como se muestra en [Figura 1.3, "Mapa de extensión"](#) las extensiones lógicas 1 a 99 podrían asignarse a un volumen físico y las extensiones lógicas 100 a 198 podrían asignarse a un segundo volumen físico. Desde el punto de vista de la aplicación, hay un dispositivo que tiene 198 extensiones.

Figura 1.3. Mapa de extensión



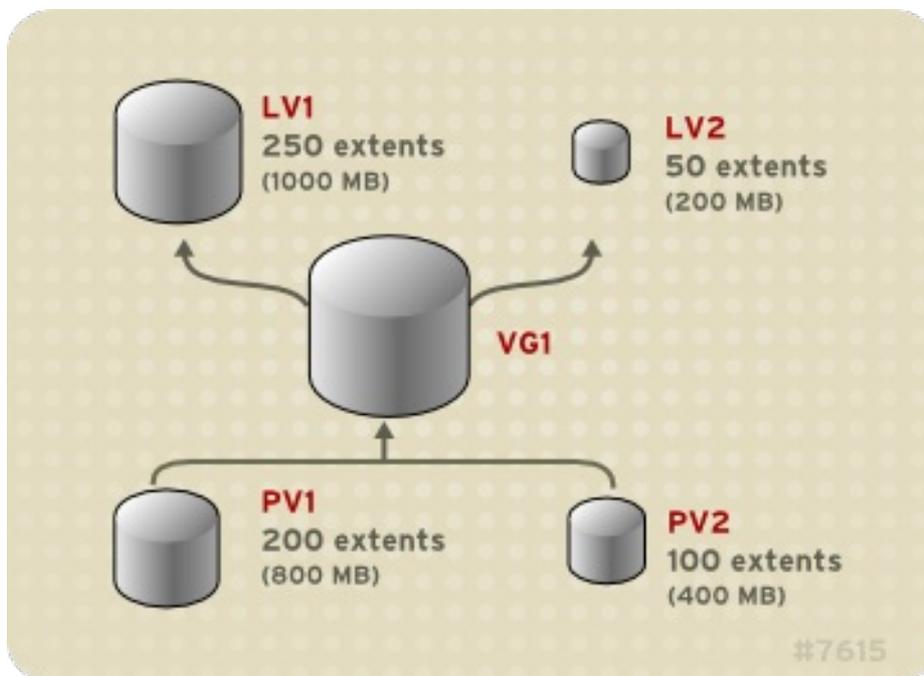
Los volúmenes físicos que componen un volumen lógico no tienen que ser del mismo tamaño. [Figura 1.4, "Volumen lineal con volúmenes físicos desiguales"](#) muestra el grupo de volúmenes **VG1** con un tamaño de extensión física de 4MB. Este grupo de volúmenes incluye 2 volúmenes físicos llamados **PV1** y **PV2**. Los volúmenes físicos están divididos en unidades de 4MB, ya que ese es el tamaño de extensión. En este ejemplo, **PV1** tiene un tamaño de 200 extensiones (800MB) y **PV2** tiene un tamaño de 100 extensiones (400MB). Puedes crear un volumen lineal de cualquier tamaño entre 1 y 300 extensiones (4MB a 1200MB). En este ejemplo, el volumen lineal llamado **LV1** tiene un tamaño de 300 extensiones.

Figura 1.4. Volumen lineal con volúmenes físicos desiguales



Puedes configurar más de un volumen lógico lineal del tamaño que necesites a partir del pool de extensiones físicas. Figura 1.5, “Múltiples volúmenes lógicos” muestra el mismo grupo de volúmenes que en Figura 1.4, “Volumen lineal con volúmenes físicos desiguales”, pero en este caso se han extraído dos volúmenes lógicos del grupo de volúmenes: **LV1**, que tiene un tamaño de 250 extensiones (1000MB) y **LV2** que tiene un tamaño de 50 extensiones (200MB).

Figura 1.5. Múltiples volúmenes lógicos



### 1.4.2. Volúmenes lógicos rayados

Cuando se escriben datos en un volumen lógico LVM, el sistema de archivos distribuye los datos entre los volúmenes físicos subyacentes. Puedes controlar la forma en que se escriben los datos en los volúmenes físicos creando un volumen lógico dividido. Para grandes lecturas y escrituras secuenciales, esto puede mejorar la eficiencia de la E/S de los datos.

El striping mejora el rendimiento escribiendo los datos en un número predeterminado de volúmenes

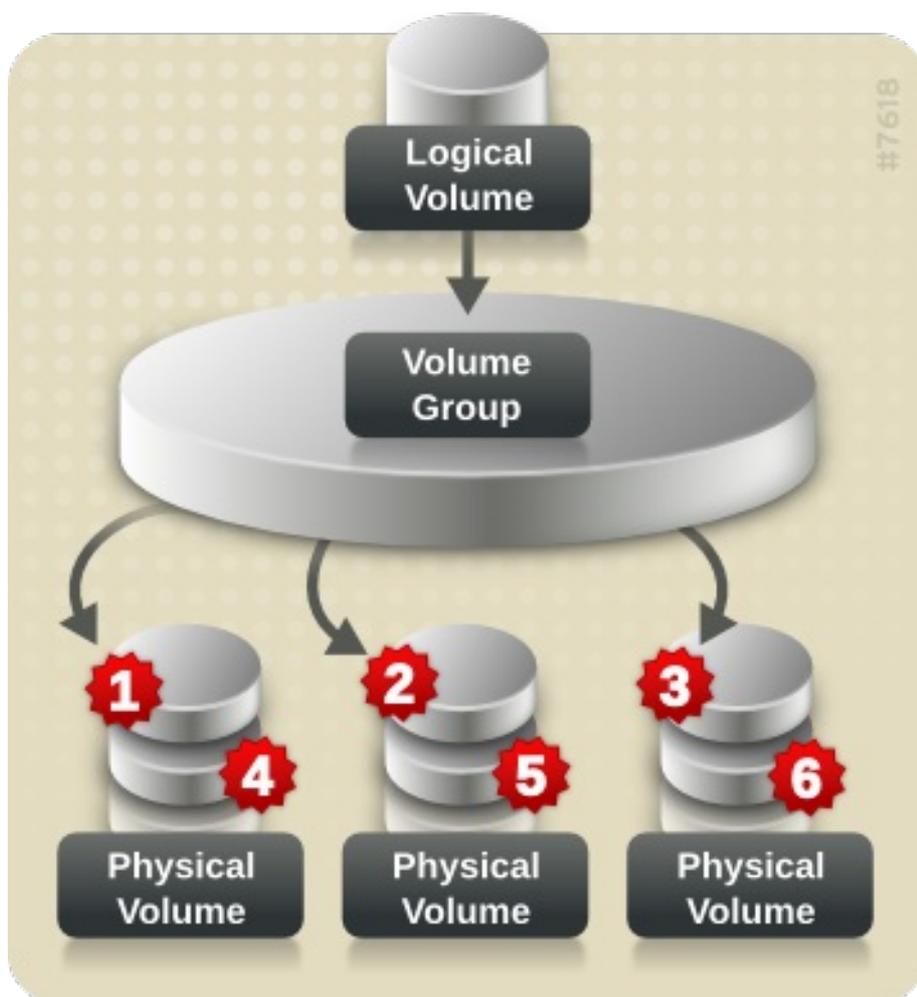
físicos en forma de round-robin. Con el striping, la E/S puede realizarse en paralelo. En algunas situaciones, esto puede resultar en una ganancia de rendimiento casi lineal por cada volumen físico adicional en la franja.

La siguiente ilustración muestra los datos que se están separando en tres volúmenes físicos. En esta figura:

- la primera franja de datos se escribe en el primer volumen físico
- la segunda franja de datos se escribe en el segundo volumen físico
- la tercera franja de datos se escribe en el tercer volumen físico
- la cuarta franja de datos se escribe en el primer volumen físico

En un volumen lógico rayado, el tamaño de la raya no puede superar el tamaño de una extensión.

Figura 1.6. Separación de datos en tres PV



Los volúmenes lógicos rayados pueden ampliarse concatenando otro conjunto de dispositivos al final del primer conjunto. Sin embargo, para ampliar un volumen lógico rayado, debe haber suficiente espacio libre en el conjunto de volúmenes físicos subyacentes que conforman el grupo de volúmenes para soportar la raya. Por ejemplo, si tienes una franja de dos direcciones que utiliza todo un grupo de volúmenes, añadir un solo volumen físico al grupo de volúmenes no te permitirá ampliar la franja. En su lugar, debes añadir al menos dos volúmenes físicos al grupo de volúmenes.

### 1.4.3. Volúmenes lógicos RAID

LVM soporta los niveles RAID 0, 1, 4, 5, 6 y 10.

Un volumen RAID LVM tiene las siguientes características:

- Los volúmenes lógicos RAID creados y gestionados por LVM aprovechan los controladores del kernel Multiple Devices (MD).
- Puede dividir temporalmente las imágenes RAID1 de la matriz y volver a fusionarlas en la matriz más tarde.
- Los volúmenes RAID de LVM admiten instantáneas.

## Agrupaciones

Los volúmenes lógicos RAID no son conscientes del cluster.

Aunque puede crear y activar volúmenes lógicos RAID exclusivamente en una máquina, no puede activarlos simultáneamente en más de una máquina.

## Subvolúmenes

Cuando se crea un volumen lógico RAID, LVM crea un subvolumen de metadatos de una extensión por cada subvolumen de datos o de paridad del array.

Por ejemplo, la creación de una matriz RAID1 de 2 vías da como resultado dos subvolúmenes de metadatos (**lv\_rmeta\_0** y **lv\_rmeta\_1**) y dos subvolúmenes de datos (**lv\_rimage\_0** y **lv\_rimage\_1**). Del mismo modo, la creación de una franja de 3 vías (más un dispositivo de paridad implícito) RAID4 da como resultado 4 subvolúmenes de metadatos (**lv\_rmeta\_0**, **lv\_rmeta\_1**, **lv\_rmeta\_2**, y **lv\_rmeta\_3**) y 4 subvolúmenes de datos (**lv\_rimage\_0**, **lv\_rimage\_1**, **lv\_rimage\_2**, y **lv\_rimage\_3**).

## Integridad

Se pueden perder datos cuando falla un dispositivo RAID o cuando se produce una corrupción blanda. La corrupción blanda en el almacenamiento de datos implica que los datos recuperados de un dispositivo de almacenamiento son diferentes de los datos escritos en ese dispositivo. Añadir integridad a un RAID LV ayuda a mitigar o prevenir la corrupción blanda. Para saber más sobre la corrupción blanda y cómo añadir integridad a un RAID LV, consulte [Sección 10.6, "Uso de la integridad de DM con RAID LV"](#).

### 1.4.4. Volúmenes lógicos de aprovisionamiento fino (volúmenes finos)

Los volúmenes lógicos pueden tener un aprovisionamiento ligero. Esto le permite crear volúmenes lógicos que son más grandes que las extensiones disponibles. Mediante el aprovisionamiento fino, puede gestionar un pool de almacenamiento de espacio libre, conocido como thin pool, que puede asignarse a un número arbitrario de dispositivos cuando lo necesiten las aplicaciones. A continuación, puede crear dispositivos que se pueden vincular al thin pool para su posterior asignación cuando una aplicación escriba realmente en el volumen lógico. El thin pool puede ampliarse dinámicamente cuando sea necesario para una asignación rentable del espacio de almacenamiento.



#### NOTA

Los volúmenes ligeros no son compatibles con los nodos de un clúster. La reserva delgada y todos sus volúmenes delgados deben activarse exclusivamente en un solo nodo del clúster.

Al utilizar el thin provisioning, un administrador de almacenamiento puede sobrecomprometer el almacenamiento físico, evitando a menudo la necesidad de comprar almacenamiento adicional. Por

ejemplo, si diez usuarios solicitan cada uno un sistema de archivos de 100 GB para su aplicación, el administrador de almacenamiento puede crear lo que parece ser un sistema de archivos de 100 GB para cada usuario, pero que está respaldado por menos almacenamiento real que se utiliza sólo cuando se necesita. Cuando se utiliza el thin provisioning, es importante que el administrador de almacenamiento supervise el pool de almacenamiento y añada más capacidad si empieza a llenarse.

Para asegurarse de que todo el espacio disponible puede ser utilizado, LVM soporta el descarte de datos. Esto permite reutilizar el espacio que antes utilizaba un archivo descartado u otro rango de bloques.

Los volúmenes delgados ofrecen soporte para una nueva implementación de volúmenes lógicos de copia en escritura (COW), que permiten que muchos dispositivos virtuales compartan los mismos datos en el grupo delgado.

### 1.4.5. Volúmenes de instantáneas

La función de instantáneas de LVM proporciona la capacidad de crear imágenes virtuales de un dispositivo en un instante determinado sin causar una interrupción del servicio. Cuando se realiza un cambio en el dispositivo original (el origen) después de tomar una instantánea, la función de instantánea hace una copia del área de datos modificada tal y como estaba antes del cambio para poder reconstruir el estado del dispositivo.



#### NOTA

LVM soporta instantáneas de aprovisionamiento ligero.

Dado que una instantánea sólo copia las áreas de datos que cambian después de la creación de la instantánea, la función de instantánea requiere una cantidad mínima de almacenamiento. Por ejemplo, con un origen que se actualiza raramente, el 3-5 % de la capacidad del origen es suficiente para mantener la instantánea.



#### NOTA

Las copias instantáneas de un sistema de archivos son copias virtuales, no una copia de seguridad real de un sistema de archivos. Las instantáneas no sustituyen a un procedimiento de copia de seguridad.

El tamaño de la instantánea gobierna la cantidad de espacio reservado para almacenar los cambios en el volumen de origen. Por ejemplo, si haces una instantánea y luego sobrescribes completamente el origen, la instantánea tendría que ser al menos tan grande como el volumen de origen para mantener los cambios. Es necesario dimensionar una instantánea de acuerdo con el nivel de cambio esperado. Así, por ejemplo, una instantánea de corta duración de un volumen de lectura, como `/usr`, necesitaría menos espacio que una instantánea de larga duración de un volumen que ve un mayor número de escrituras, como `/home`.

Si una instantánea se llena, la instantánea deja de ser válida, ya que no puede seguir los cambios en el volumen de origen. Debe controlar regularmente el tamaño de la instantánea. Sin embargo, las instantáneas son totalmente redimensionables, por lo que si tienes capacidad de almacenamiento puedes aumentar el tamaño del volumen de la instantánea para evitar que se pierda. A la inversa, si ves que el volumen de la instantánea es más grande de lo que necesitas, puedes reducir el tamaño del volumen para liberar espacio que necesitan otros volúmenes lógicos.

Cuando se crea un sistema de archivos de instantánea, el acceso de lectura y escritura completo al origen sigue siendo posible. Si se modifica un chunk en una instantánea, ese chunk se marca y nunca se copia del volumen original.

La función de instantáneas tiene varios usos:

- Normalmente, se toma una instantánea cuando se necesita realizar una copia de seguridad en un volumen lógico sin detener el sistema en vivo que está actualizando continuamente los datos.
- Puede ejecutar el comando **fsck** en un sistema de archivos de instantánea para comprobar la integridad del sistema de archivos y determinar si el sistema de archivos original requiere una reparación del sistema de archivos.
- Dado que la instantánea es de lectura/escritura, puede probar las aplicaciones contra los datos de producción tomando una instantánea y ejecutando pruebas contra la instantánea, dejando los datos reales intactos.
- Puede crear volúmenes LVM para utilizarlos con Red Hat Virtualization. Las instantáneas LVM se pueden utilizar para crear instantáneas de imágenes de huéspedes virtuales. Estas instantáneas pueden proporcionar una forma conveniente de modificar los huéspedes existentes o crear nuevos huéspedes con un mínimo de almacenamiento adicional.

Puedes utilizar la opción **--merge** del comando **lvconvert** para fusionar una instantánea con su volumen de origen. Una de las aplicaciones de esta función es realizar la reversión del sistema si has perdido datos o archivos o si necesitas restaurar tu sistema a un estado anterior. Después de fusionar el volumen de la instantánea, el volumen lógico resultante tendrá el nombre, el número menor y el UUID del volumen de origen y la instantánea fusionada se eliminará.

#### 1.4.6. Volúmenes instantáneos de aprovisionamiento ligero

Red Hat Enterprise Linux proporciona soporte para volúmenes de instantáneas de aprovisionamiento delgado. Los volúmenes de instantáneas delgadas permiten que muchos dispositivos virtuales sean almacenados en el mismo volumen de datos. Esto simplifica la administración y permite compartir datos entre volúmenes de instantáneas.

Como para todos los volúmenes de instantánea LVM, así como para todos los volúmenes finos, los volúmenes finos de instantánea no son compatibles con los nodos de un clúster. El volumen de instantánea debe activarse exclusivamente en un solo nodo del clúster.

Los volúmenes de instantáneas delgadas proporcionan los siguientes beneficios:

- Un volumen delgado de instantáneas puede reducir el uso del disco cuando hay varias instantáneas del mismo volumen de origen.
- Si hay varias instantáneas del mismo origen, una escritura en el origen provocará una operación COW para conservar los datos. Aumentar el número de instantáneas del origen no debería suponer una ralentización importante.
- Los volúmenes finos de instantáneas pueden utilizarse como origen de volumen lógico para otra instantánea. Esto permite una profundidad arbitraria de instantáneas recursivas (instantáneas de instantáneas de instantáneas...).
- Una instantánea de un volumen lógico delgado también crea un volumen lógico delgado. Esto no consume espacio de datos hasta que se requiera una operación COW, o hasta que se escriba la propia instantánea.

- Un volumen delgado de instantáneas no necesita ser activado con su origen, por lo que un usuario puede tener sólo el origen activo mientras hay muchos volúmenes de instantáneas inactivos del origen.
- Cuando se elimina el origen de un volumen de instantáneas de aprovisionamiento ligero, cada instantánea de ese volumen de origen se convierte en un volumen de aprovisionamiento ligero independiente. Esto significa que, en lugar de fusionar una instantánea con su volumen de origen, puedes optar por eliminar el volumen de origen y luego crear una nueva instantánea con aprovisionamiento ligero utilizando ese volumen independiente como volumen de origen para la nueva instantánea.

Aunque hay muchas ventajas en el uso de volúmenes delgados de instantáneas, hay algunos casos de uso para los que la antigua función de volumen de instantáneas LVM puede ser más apropiada para sus necesidades:

- No se puede cambiar el tamaño del chunk de un thin pool. Si el thin pool tiene un tamaño de chunk grande (por ejemplo, 1MB) y necesita una instantánea de corta duración para la que un tamaño de chunk tan grande no es eficiente, puede optar por utilizar la función de instantáneas antiguas.
- No se puede limitar el tamaño de un volumen de instantánea delgada; la instantánea utilizará todo el espacio de la reserva delgada, si es necesario. Esto puede no ser apropiado para sus necesidades.

En general, debe tener en cuenta los requisitos específicos de su sitio web a la hora de decidir qué formato de instantánea utilizar.

### 1.4.7. Volúmenes de caché

LVM admite el uso de dispositivos de bloque rápidos (como las unidades SSD) como cachés de retroescritura o escritura para dispositivos de bloque más grandes y lentos. Los usuarios pueden crear volúmenes lógicos de caché para mejorar el rendimiento de sus volúmenes lógicos existentes o crear nuevos volúmenes lógicos de caché compuestos por un dispositivo pequeño y rápido acoplado a un dispositivo grande y lento.

## CAPÍTULO 2. GESTIÓN DEL ALMACENAMIENTO LOCAL MEDIANTE LOS ROLES DE SISTEMA DE RHEL

Para gestionar LVM y sistemas de archivos locales (FS) mediante Ansible, puede utilizar el rol **storage**, que es uno de los roles de sistema RHEL disponibles en RHEL 8.

El uso del rol **storage** le permite automatizar la administración de sistemas de archivos en discos y volúmenes lógicos en múltiples máquinas y en todas las versiones de RHEL a partir de RHEL 7.7.

Para más información sobre los Roles del Sistema RHEL y cómo aplicarlos, vea [Introducción a los Roles del Sistema RHEL](#).

### 2.1. INTRODUCCIÓN A LA FUNCIÓN DE ALMACENAMIENTO

La función **storage** puede gestionar:

- Sistemas de archivos en discos que no han sido particionados
- Grupos de volúmenes LVM completos, incluyendo sus volúmenes lógicos y sistemas de archivos

Con el rol **storage** puede realizar las siguientes tareas:

- Crear un sistema de archivos
- Eliminar un sistema de archivos
- Montar un sistema de archivos
- Desmontar un sistema de archivos
- Crear grupos de volúmenes LVM
- Eliminar grupos de volúmenes LVM
- Crear volúmenes lógicos
- Eliminar volúmenes lógicos
- Crear volúmenes RAID
- Eliminar volúmenes RAID
- Crear pools LVM con RAID
- Eliminar pools LVM con RAID

### 2.2. PARÁMETROS QUE IDENTIFICAN UN DISPOSITIVO DE ALMACENAMIENTO EN EL ROL DE SISTEMA DE ALMACENAMIENTO

La configuración de su rol en **storage** afecta sólo a los sistemas de archivos, volúmenes y pools que se enumeran en las siguientes variables.

#### **storage\_volumes**

Lista de sistemas de archivos en todos los discos no particionados que se van a gestionar. Actualmente, las particiones no son compatibles.

## storage\_pools

Lista de piscinas a gestionar.

Actualmente el único tipo de pool soportado es LVM. Con LVM, los pools representan grupos de volúmenes (VGs). Bajo cada pool hay una lista de volúmenes que deben ser gestionados por el rol. Con LVM, cada volumen corresponde a un volumen lógico (LV) con un sistema de archivos.

## 2.3. EJEMPLO DE PLAYBOOK DE ANSIBLE PARA CREAR UN SISTEMA DE ARCHIVOS XFS EN UN DISPOSITIVO DE BLOQUES

Esta sección proporciona un ejemplo de libro de jugadas de Ansible. Este libro de jugadas aplica el rol **storage** para crear un sistema de archivos XFS en un dispositivo de bloques utilizando los parámetros predeterminados.



### AVISO

El rol **storage** puede crear un sistema de archivos sólo en un disco entero no particionado o en un volumen lógico (LV). No puede crear el sistema de archivos en una partición.

### Ejemplo 2.1. Un playbook que crea XFS en /dev/sdb

```

---
- hosts: all
  vars:
    storage_volumes:
      - name: barefs
        type: disk
        disks:
          - sdb
        fs_type: xfs
  roles:
    - rhel-system-roles.storage

```

- El nombre del volumen (**barefs** en el ejemplo) es actualmente arbitrario. El rol **storage** identifica el volumen por el dispositivo de disco listado bajo el atributo **disks:**.
- Puede omitir la línea **fs\_type: xfs** porque XFS es el sistema de archivos por defecto en RHEL 8.
- Para crear el sistema de archivos en un LV, proporcione la configuración de LVM bajo el atributo **disks:**, incluyendo el grupo de volúmenes que lo rodea. Para obtener más detalles, consulte [Ejemplo de libro de jugadas de Ansible para gestionar volúmenes lógicos](#) . No proporcione la ruta de acceso al dispositivo LV.

### Recursos adicionales

- Para más detalles sobre los parámetros utilizados en el rol de sistema **storage**, consulte el archivo `/usr/share/ansible/roles/rhel-system-roles.storage/README.md`.

## 2.4. EJEMPLO DE PLAYBOOK DE ANSIBLE PARA MONTAR PERSISTENTEMENTE UN SISTEMA DE ARCHIVOS

Esta sección proporciona un ejemplo de libro de jugadas de Ansible. Este libro de jugadas aplica el rol **storage** para montar de forma inmediata y persistente un sistema de archivos XFS.

### Ejemplo 2.2. Un playbook que monta un sistema de archivos en `/dev/sdb` a `/mnt/data`

```
---
- hosts: all
  vars:
    storage_volumes:
      - name: barefs
        type: disk
        disks:
          - sdb
        fs_type: xfs
        mount_point: /mnt/data
  roles:
    - rhel-system-roles.storage
```

- Este libro de jugadas añade el sistema de archivos al archivo `/etc/fstab`, y monta el sistema de archivos inmediatamente.
- Si el sistema de archivos del dispositivo `/dev/sdb` o el directorio del punto de montaje no existen, el libro de jugadas los crea.

### Recursos adicionales

- Para más detalles sobre los parámetros utilizados en el rol de sistema **storage**, consulte el archivo `/usr/share/ansible/roles/rhel-system-roles.storage/README.md`.

## 2.5. EJEMPLO DE LIBRO DE JUGADAS DE ANSIBLE PARA GESTIONAR VOLÚMENES LÓGICOS

Esta sección proporciona un ejemplo de libro de jugadas de Ansible. Este libro de jugadas aplica el rol **storage** para crear un volumen lógico LVM en un grupo de volúmenes.

### Ejemplo 2.3. Un libro de jugadas que crea un volumen lógico `mylv` en el grupo de volúmenes `myvg`

```
- hosts: all
  vars:
    storage_pools:
      - name: myvg
        disks:
          - sda
          - sdb
          - sdc
```

```

volumes:
  - name: mylv
    size: 2G
    fs_type: ext4
    mount_point: /mnt
roles:
  - rhel-system-roles.storage

```

- El grupo de volúmenes **myvg** está formado por los siguientes discos:
  - **/dev/sda**
  - **/dev/sdb**
  - **/dev/sdc**
- Si el grupo de volumen **myvg** ya existe, el libro de jugadas añade el volumen lógico al grupo de volumen.
- Si el grupo de volumen **myvg** no existe, el libro de jugadas lo crea.
- El libro de jugadas crea un sistema de archivos Ext4 en el volumen lógico **mylv**, y monta persistentemente el sistema de archivos en **/mnt**.

### Recursos adicionales

- Para más detalles sobre los parámetros utilizados en el rol de sistema **storage**, consulte el archivo `/usr/share/ansible/roles/rhel-system-roles.storage/README.md`.

## 2.6. EJEMPLO DE LIBRO DE JUGADAS DE ANSIBLE PARA ACTIVAR EL DESCARTE DE BLOQUES EN LÍNEA

Esta sección proporciona un ejemplo de libro de jugadas de Ansible. Este libro de jugadas aplica el rol **storage** para montar un sistema de archivos XFS con el descarte de bloques en línea activado.

### Ejemplo 2.4. Un libro de jugadas que permite descartar bloques en línea en `/mnt/data/`

```

---
- hosts: all
  vars:
    storage_volumes:
      - name: barefs
        type: disk
        disks:
          - sdb
        fs_type: xfs
        mount_point: /mnt/data
        mount_options: discard
  roles:
    - rhel-system-roles.storage

```

### Recursos adicionales

- Este libro [de jugadas](#) también realiza todas las operaciones del ejemplo de montaje persistente descrito en [Ejemplo de libro de jugadas de Ansible para montar persistentemente un sistema de archivos](#).
- Para más detalles sobre los parámetros utilizados en el rol de sistema **storage**, consulte el archivo `/usr/share/ansible/roles/rhel-system-roles.storage/README.md`.

## 2.7. EJEMPLO DE PLAYBOOK ANSIBLE PARA CREAR Y MONTAR UN SISTEMA DE ARCHIVOS EXT4

Esta sección proporciona un ejemplo de libro de jugadas de Ansible. Este libro de jugadas aplica el rol **storage** para crear y montar un sistema de archivos Ext4.

**Ejemplo 2.5. Un playbook que crea Ext4 en `/dev/sdb` y lo monta en `/mnt/data`**

```
---
- hosts: all
  vars:
    storage_volumes:
      - name: barefs
        type: disk
        disks:
          - sdb
        fs_type: ext4
        fs_label: label-name
        mount_point: /mnt/data
  roles:
    - rhel-system-roles.storage
```

- El libro de jugadas crea el sistema de archivos en el disco `/dev/sdb`.
- El libro de jugadas monta persistentemente el sistema de archivos en el directorio `/mnt/data` directorio.
- La etiqueta del sistema de archivos es ***label-name***.

### Recursos adicionales

- Para más detalles sobre los parámetros utilizados en el rol de sistema **storage**, consulte el archivo `/usr/share/ansible/roles/rhel-system-roles.storage/README.md`.

## 2.8. EJEMPLO DE PLAYBOOK DE ANSIBLE PARA CREAR Y MONTAR UN SISTEMA DE ARCHIVOS EXT3

Esta sección proporciona un ejemplo de libro de jugadas de Ansible. Este libro de jugadas aplica el rol **storage** para crear y montar un sistema de archivos Ext3.

**Ejemplo 2.6. Un playbook que crea Ext3 en `/dev/sdb` y lo monta en `/mnt/data`**

```
---
- hosts: all
  vars:
```

```

storage_volumes:
  - name: barefs
    type: disk
    disks:
      - sdb
    fs_type: ext3
    fs_label: label-name
    mount_point: /mnt/data
roles:
  - rhel-system-roles.storage

```

- El libro de jugadas crea el sistema de archivos en el disco **/dev/sdb**.
- El libro de jugadas monta persistentemente el sistema de archivos en el directorio **/mnt/data** directorio.
- La etiqueta del sistema de archivos es **label-name**.

### Recursos adicionales

- Para más detalles sobre los parámetros utilizados en el rol de sistema **storage**, consulte el archivo **/usr/share/ansible/roles/rhel-system-roles.storage/README.md**.

## 2.9. CONFIGURACIÓN DE UN VOLUMEN RAID MEDIANTE EL ROL DE SISTEMA DE ALMACENAMIENTO

Con el rol de sistema **storage**, puede configurar un volumen RAID en RHEL utilizando Red Hat Ansible Automation Platform. En esta sección aprenderá a configurar un playbook de Ansible con los parámetros disponibles para configurar un volumen RAID que se adapte a sus necesidades.

### Requisitos previos

- Tiene instalado Red Hat Ansible Engine en el sistema desde el que desea ejecutar el libro de jugadas.



#### NOTA

No es necesario tener Red Hat Ansible Automation Platform instalado en los sistemas en los que se desea implementar la solución **storage**.

- Tienes el paquete **rhel-system-roles** instalado en el sistema desde el que quieres ejecutar el playbook.
- Tienes un archivo de inventario que detalla los sistemas en los que quieres desplegar un volumen RAID usando el rol de sistema **storage**.

### Procedimiento

1. Cree un nuevo **playbook.yml** archivo con el siguiente contenido:

```

- hosts: all
  vars:

```

```

storage_safe_mode: false
storage_volumes:
  - name: data
    type: raid
    disks: [sdd, sde, sdf, sdg]
    raid_level: raid0
    raid_chunk_size: 32 KiB
    mount_point: /mnt/data
    state: present
roles:
  - name: rhel-system-roles.storage

```



### AVISO

Los nombres de los dispositivos pueden cambiar en determinadas circunstancias; por ejemplo, cuando se añade un nuevo disco a un sistema. Por lo tanto, para evitar la pérdida de datos, no se recomienda utilizar nombres de discos específicos en el libro de jugadas.

2. Opcional. Verificar la sintaxis del libro de jugadas.

```
# ansible-playbook --syntax-check playbook.yml
```

3. Ejecute el libro de jugadas en su archivo de inventario:

```
# ansible-playbook -i inventory.file /path/to/file/playbook.yml
```

### Recursos adicionales

- Para obtener más información sobre RAID, consulte [Gestión de RAID](#).
- Para obtener detalles sobre los parámetros utilizados en el rol del sistema de almacenamiento, consulte el archivo `/usr/share/ansible/roles/rhel-system-roles.storage/README.md`.

## 2.10. CONFIGURACIÓN DE UN POOL LVM CON RAID UTILIZANDO EL ROL DE SISTEMA DE ALMACENAMIENTO

Con el rol de sistema **storage**, puede configurar un pool LVM con RAID en RHEL utilizando Red Hat Ansible Automation Platform. En esta sección aprenderá a configurar un playbook Ansible con los parámetros disponibles para configurar un pool LVM con RAID.

### Requisitos previos

- Tiene instalado Red Hat Ansible Engine en el sistema desde el que desea ejecutar el libro de jugadas.

**NOTA**

No es necesario tener Red Hat Ansible Automation Platform instalado en los sistemas en los que se desea implementar la solución **storage**.

- Tienes el paquete **rhel-system-roles** instalado en el sistema desde el que quieres ejecutar el playbook.
- Tienes un archivo de inventario que detalla los sistemas en los que quieres configurar un pool LVM con RAID utilizando el rol de sistema **storage**.

**Procedimiento**

1. Cree un nuevo **playbook.yml** archivo con el siguiente contenido:

```
- hosts: all
  vars:
    storage_safe_mode: false
    storage_pools:
      - name: my_pool
        type: lvm
        disks: [sdh, sdi]
        raid_level: raid1
        volumes:
          - name: my_pool
            size: "1 GiB"
            mount_point: "/mnt/app/shared"
            fs_type: xfs
            state: present
  roles:
    - name: rhel-system-roles.storage
```

**NOTA**

Para crear un pool LVM con RAID, debes especificar el tipo de RAID utilizando el parámetro **raid\_level**.

2. Opcional. Verificar la sintaxis del libro de jugadas.

```
# ansible-playbook --syntax-check playbook.yml
```

3. Ejecute el libro de jugadas en su archivo de inventario:

```
# ansible-playbook -i inventory.file /path/to/file/playbook.yml
```

**Recursos adicionales**

- Para obtener más información sobre RAID, consulte [Gestión de RAID](#).
- Para obtener detalles sobre los parámetros utilizados en el rol del sistema de almacenamiento, consulte el archivo **/usr/share/ansible/roles/rhel-system-roles.storage/README.md**.

## 2.11. CREACIÓN DE UN VOLUMEN ENCRYPTADO LUKS UTILIZANDO EL ROL DE ALMACENAMIENTO

Puede utilizar el rol **storage** para crear y configurar un volumen encriptado con LUKS ejecutando un playbook de Ansible.

### Requisitos previos

- Tiene instalado Red Hat Ansible Engine en el sistema desde el que desea ejecutar el libro de jugadas.



### NOTA

No es necesario tener Red Hat Ansible Automation Platform instalado en los sistemas en los que se desea crear el volumen.

- Tiene el paquete **rhel-system-roles** instalado en el controlador Ansible.
- Dispone de un archivo de inventario en el que se detallan los sistemas en los que desea desplegar un volumen encriptado LUKS mediante el rol de sistema de almacenamiento.

### Procedimiento

1. Cree un nuevo **playbook.yml** archivo con el siguiente contenido:

```
- hosts: all
vars:
  storage_volumes:
    - name: barefs
      type: disk
      disks:
        - sdb
      fs_type: xfs
      fs_label: label-name
      mount_point: /mnt/data
      encryption: true
      encryption_password: your-password
roles:
  - rhel-system-roles.storage
```

2. Opcional. Verificar la sintaxis del libro de jugadas:

```
# ansible-playbook --syntax-check playbook.yml
```

3. Ejecute el libro de jugadas en su archivo de inventario:

```
# ansible-playbook -i inventory.file /path/to/file/playbook.yml
```

### Recursos adicionales

- Para más información sobre LUKS, véase [17. Cifrado de dispositivos de bloque mediante LUKS](#).

- Para más detalles sobre los parámetros utilizados en el rol de sistema **storage**, consulte el archivo `/usr/share/ansible/roles/rhel-system-roles.storage/README.md`.

### Recursos adicionales

- Para más información, instale el paquete **rhel-system-roles** y consulte los siguientes directorios:
  - `/usr/share/doc/rhel-system-roles/storage/`
  - `/usr/share/ansible/roles/rhel-system-roles.storage/`

## CAPÍTULO 3. DESPLIEGUE DE LVM

Los siguientes procedimientos representan la configuración básica para volúmenes lógicos LVM en un sistema operativo recién instalado.

### 3.1. CREACIÓN DE UN VOLUMEN FÍSICO LVM

Este procedimiento describe cómo crear y etiquetar volúmenes físicos (PV) LVM.

#### Requisitos previos

- El paquete **lvm2** está instalado.

#### Procedimiento

1. Para crear varios volúmenes físicos, utilice los nombres de los dispositivos delimitados por espacios como argumentos del comando **pvcreate**:

```
# pvcreate /dev/vdb1 /dev/vdb2
```

Esto coloca una etiqueta en `/dev/vdb1` y `/dev/vdb2`, marcándolos como volúmenes físicos pertenecientes a LVM.

Para más información, consulte la página de manual **pvcreate**.

### 3.2. CREACIÓN DE UN GRUPO DE VOLÚMENES LVM

Este procedimiento describe cómo crear un grupo de volúmenes LVM (VG).

#### Requisitos previos

- El paquete **lvm2** está instalado.
- Se crean uno o más volúmenes físicos. Consulte las instrucciones de cómo hacerlo en [Sección 3.1, "Creación de un volumen físico LVM"](#).

#### Procedimiento

1. Para crear un grupo de volúmenes, utilice el siguiente comando:

```
# vgcreate myvg /dev/vdb1 /dev/vdb2
```

Esto crea un VG con el nombre de `myvg`. Los PVs `/dev/vdb1` y `/dev/vdb2` son el nivel de almacenamiento base para el VG `myvg`.

Para más información, consulte la página de manual **vgcreate**.

Es posible ampliar la VG anterior con los PV más adelante. Para ampliar una VG, utilice el siguiente comando:

```
# vgextend myvg /dev/vdb3
```

### 3.3. CREACIÓN DE UN VOLUMEN LÓGICO LVM

Este procedimiento describe cómo crear un volumen lógico LVM (LV).

#### Requisitos previos

- El paquete **lvm2** está instalado.
- Se crea el grupo de volúmenes. Consulte las instrucciones sobre cómo hacerlo en [Sección 3.2, "Creación de un grupo de volúmenes LVM"](#).

#### Procedimiento

1. Para crear un volumen lógico, utilice el siguiente comando:

```
# lvcreate -n lv01 -L 500M myvg
```

La opción **-n** permite al usuario establecer el nombre del BT en *lv01*. La opción **-L** permite al usuario establecer el tamaño del BT en unidades de Mb en este ejemplo, pero es posible utilizar cualquier otra unidad. El tipo de BT es lineal por defecto, pero el usuario puede especificar el tipo deseado utilizando la opción **--type**.



#### IMPORTANTE

El comando fallará si el VG no tiene un número suficiente de extensiones físicas libres para el tamaño y tipo solicitados.

Para más información, consulte la página de manual **lvcreate**.

# CAPÍTULO 4. CONFIGURACIÓN DE VOLÚMENES LÓGICOS LVM

Los siguientes procedimientos proporcionan ejemplos de tareas básicas de administración de LVM.

## 4.1. USO DE LOS COMANDOS DE LA CLI

Las siguientes secciones describen algunas características operativas generales de los comandos de la CLI de LVM.

### Especificación de unidades en un argumento de línea de comandos

Cuando se requieren tamaños en un argumento de la línea de comandos, las unidades siempre se pueden especificar explícitamente. Si no se especifica una unidad, se asume una por defecto, normalmente KB o MB. Los comandos de la CLI de LVM no aceptan fracciones.

Cuando se especifican unidades en un argumento de la línea de comandos, LVM no distingue entre mayúsculas y minúsculas; especificar M o m es equivalente, por ejemplo, y se utilizan potencias de 2 (múltiplos de 1024). Sin embargo, cuando se especifica el argumento **--units** en un comando, las minúsculas indican que las unidades están en múltiplos de 1024 mientras que las mayúsculas indican que las unidades están en múltiplos de 1000.

### Especificación de grupos de volúmenes y volúmenes lógicos

Tenga en cuenta lo siguiente cuando especifique grupos de volúmenes o volúmenes lógicos en un comando CLI de LVM.

- Cuando los comandos toman como argumento nombres de grupos de volúmenes o volúmenes lógicos, el nombre completo de la ruta es opcional. Un volumen lógico llamado **lvol0** en un grupo de volúmenes llamado **vg0** puede ser especificado como **vg0/lvol0**.
- Cuando se requiera una lista de grupos de volúmenes pero se deje vacía, se sustituirá por una lista de todos los grupos de volúmenes.
- Cuando se requiere una lista de volúmenes lógicos pero se da un grupo de volúmenes, se sustituirá por una lista de todos los volúmenes lógicos de ese grupo de volúmenes. Por ejemplo, el comando **lvdisplay vg0** mostrará todos los volúmenes lógicos del grupo de volúmenes **vg0**.

### Aumentar la verbosidad de la salida

Todos los comandos LVM aceptan un argumento **-v**, que puede ser introducido varias veces para aumentar la verbosidad de la salida. Los siguientes ejemplos muestran la 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 comando muestra la salida del comando **lvcreate** con el argumento **-v**.

```
# lvcreate -v -L 50MB new_vg
Rounding up size to full physical extent 52.00 MB
Archiving volume group "new_vg" metadata (seqno 1).
Creating logical volume lvol0
Creating volume group backup "/etc/lvm/backup/new_vg" (seqno 2).
Activating logical volume new_vg/lvol0.
activation/volume_list configuration setting not defined: Checking only host tags for new_vg/lvol0.
```

```

Creating new_vg-lvol0
Loading table for new_vg-lvol0 (253:0).
Resuming new_vg-lvol0 (253:0).
Wiping known signatures on logical volume "new_vg/lvol0"
Initializing 4.00 KiB of logical volume "new_vg/lvol0" with value 0.
Logical volume "lvol0" created

```

Los argumentos **-vv**, **-vvv** y **-vvvv** muestran cada vez más detalles sobre la ejecución del comando. El argumento **-vvvv** proporciona la máxima cantidad de información en este momento. El siguiente ejemplo muestra las primeras líneas de salida del comando **lvcreate** con el argumento **-vvvv** especificado.

```

# 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
...

```

### Visualización de la ayuda para los comandos de la CLI de LVM

Puede mostrar la ayuda para cualquiera de los comandos de la CLI de LVM con el argumento **--help** del comando.

```
# commandname --help
```

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

```
# man commandname
```

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

## 4.2. CREACIÓN DE UN VOLUMEN LÓGICO LVM EN TRES DISCOS

Este procedimiento de ejemplo crea un volumen lógico LVM llamado **mylv** que consiste en los discos en **/dev/sda1**, **/dev/sdb1**, y **/dev/sdc1**.

1. Para utilizar los discos en un grupo de volúmenes, etiquételos como volúmenes físicos LVM con el comando **pvcreate**.

**AVISO**

Este comando destruye cualquier dato en **/dev/sda1**, **/dev/sdb1**, y **/dev/sdc1**.

```
# 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
```

2. Crea un grupo de volúmenes que consiste en los volúmenes físicos LVM que has creado. El siguiente comando crea el grupo de volúmenes **myvg**.

```
# vgcreate myvg /dev/sda1 /dev/sdb1 /dev/sdc1
Volume group "myvg" successfully created
```

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

```
# vgs
VG #PV #LV #SN Attr VSize VFree
myvg 3 0 0 wz--n- 51.45G 51.45G
```

3. Crea el volumen lógico a partir del grupo de volúmenes que has creado. El siguiente comando crea el volumen lógico **mylv** a partir del grupo de volumen **myvg**. Este ejemplo crea un volumen lógico que utiliza 2 gigabytes del grupo de volúmenes.

```
# lvcreate -L 2G -n mylv myvg
Logical volume "mylv" created
```

4. Crea un sistema de archivos en el volumen lógico. El siguiente comando crea un sistema de archivos **ext4** en el volumen lógico.

```
# mkfs.ext4 /dev/myvg/mylv
mke2fs 1.44.3 (10-July-2018)
Creating filesystem with 524288 4k blocks and 131072 inodes
Filesystem UUID: 616da032-8a48-4cd7-8705-bd94b7a1c8c4
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

Los siguientes comandos montan el volumen lógico e informan del uso del espacio en disco del sistema de archivos.

```
# mount /dev/myvg/mylv /mnt
# df
```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/mapper/myvg-mylv	1998672	6144	1871288	1%	/mnt

### 4.3. CREACIÓN DE UN VOLUMEN LÓGICO RAID0 (RAYADO)

Un volumen lógico RAID0 reparte los datos del volumen lógico entre varios subvolúmenes de datos en unidades de tamaño de franja.

El formato del comando para crear un volumen RAID0 es el siguiente.

```
lvcreate --type raid0[_meta] --stripes Stripes --stripesize StripeSize VolumeGroup
[PhysicalVolumePath...]
```

Tabla 4.1. Parámetros de creación del comando RAID0

Parámetro	Descripción
<b>--type raid0[_meta]</b>	Especificando <b>raid0</b> se crea un volumen RAID0 sin volúmenes de metadatos. Especificando <b>raid0_meta</b> se crea un volumen RAID0 con volúmenes de metadatos. Como el RAID 0 no es resistente, no tiene que almacenar ningún bloque de datos en espejo como el RAID1/10 ni calcular y almacenar ningún bloque de paridad como el RAID4/5/6. Por lo tanto, no necesita volúmenes de metadatos para mantener el estado sobre el progreso de la resincronización de los bloques en espejo o de paridad. Sin embargo, los volúmenes de metadatos se vuelven obligatorios en una conversión de RAID0 a RAID4/5/6/10, y al especificar <b>raid0_meta</b> se preasignan esos volúmenes de metadatos para evitar un fallo de asignación respectivo.
<b>--stripes <i>Stripes</i></b>	Especifica el número de dispositivos en los que se repartirá el volumen lógico.
<b>--stripesize <i>StripeSize</i></b>	Especifica el tamaño de cada franja en kilobytes. Es la cantidad de datos que se escribe en un dispositivo antes de pasar al siguiente.
<b><i>VolumeGroup</i></b>	Especifica el grupo de volúmenes a utilizar.
<b><i>PhysicalVolumePath...</i></b>	Especifica los dispositivos a utilizar. Si no se especifica, LVM elegirá el número de dispositivos especificados por la opción <i>Stripes</i> , uno para cada franja.

Este procedimiento de ejemplo crea un volumen lógico RAID0 de LVM llamado **mylv** que separa los datos en los discos en **/dev/sda1**, **/dev/sdb1**, y **/dev/sdc1**.

1. Etiquete los discos que utilizará en el grupo de volúmenes como volúmenes físicos LVM con el comando **pvcreate**.

**AVISO**

Este comando destruye cualquier dato en **/dev/sda1**, **/dev/sdb1**, y **/dev/sdc1**.

```
# 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
```

2. Crear el grupo de volumen **myvg**. El siguiente comando crea el grupo de volúmenes **myvg**.

```
# vgcreate myvg /dev/sda1 /dev/sdb1 /dev/sdc1
Volume group "myvg" successfully created
```

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

```
# vgs
VG #PV #LV #SN Attr VSize VFree
myvg 3 0 0 wz--n- 51.45G 51.45G
```

3. Crea un volumen lógico RAID0 a partir del grupo de volúmenes que has creado. El siguiente comando crea el volumen RAID0 **mylv** a partir del grupo de volúmenes **myvg**. Este ejemplo crea un volumen lógico que tiene un tamaño de 2 gigabytes, con tres franjas y un tamaño de franja de 4 kilobytes.

```
# lvcreate --type raid0 -L 2G --stripes 3 --stripesize 4 -n mylv myvg
Rounding size 2.00 GiB (512 extents) up to stripe boundary size 2.00 GiB(513 extents).
Logical volume "mylv" created.
```

4. Cree un sistema de archivos en el volumen lógico RAID0. El siguiente comando crea un sistema de archivos **ext4** en el volumen lógico.

```
# mkfs.ext4 /dev/myvg/mylv
mke2fs 1.44.3 (10-July-2018)
Creating filesystem with 525312 4k blocks and 131376 inodes
Filesystem UUID: 9d4c0704-6028-450a-8b0a-8875358c0511
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

Los siguientes comandos montan el volumen lógico e informan del uso del espacio en disco del sistema de archivos.

```
# mount /dev/myvg/mylv /mnt
# df
Filesystem          1K-blocks  Used Available Use% Mounted on
/dev/mapper/myvg-mylv 2002684   6168  1875072  1% /mnt
```

## 4.4. RENOMBRAR VOLÚMENES LÓGICOS LVM

Este procedimiento cambia el nombre de un volumen lógico existente utilizando la interfaz LVM de la línea de comandos.

### Procedimiento

1. Si el volumen lógico está actualmente montado, desmonte el volumen.
2. Si el volumen lógico existe en un entorno de clúster, desactive el volumen lógico en todos los nodos en los que esté activo. Utilice el siguiente comando en cada uno de esos nodos:

```
[root@node-n]# lvchange --activate n vg-name/lv-name
```

3. Utilice la utilidad **lvrename** para renombrar un volumen lógico existente:

```
# lvrename vg-name original-lv-name new-lv-name
```

Opcionalmente, puede especificar las rutas completas de los dispositivos:

```
# lvrename /dev/vg-name/original-lv-name /dev/vg-name/new-lv-name
```

### Recursos adicionales

- La página de manual **lvrename(8)**

## 4.5. ELIMINACIÓN DE UN DISCO DE UN VOLUMEN LÓGICO

Estos procedimientos de ejemplo muestran cómo se puede eliminar un disco de un volumen lógico existente, ya sea para reemplazar el disco o para utilizar el disco como parte de un volumen diferente. Para eliminar un disco, primero debes mover las extensiones del volumen físico LVM a un disco o conjunto de discos diferente.

### 4.5.1. Mover extensiones a volúmenes físicos existentes

En este ejemplo, el volumen lógico se distribuye en cuatro volúmenes físicos en el grupo de volúmenes **myvg**.

```
# 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
```

Estos ejemplos mueven las extensiones fuera de **/dev/sdb1** para que pueda ser eliminado del grupo de volúmenes.

1. Si hay suficientes extensiones libres en los otros volúmenes físicos del grupo de volúmenes, puedes ejecutar el comando **pvmove** en el dispositivo que quieres eliminar sin otras opciones y las extensiones se distribuirán a los otros dispositivos.

En un clúster, el comando **pvmove** sólo puede mover el volumen lógico que está activo exclusivamente en un solo nodo.

```
# pvmove /dev/sdb1
/dev/sdb1: Moved: 2.0%
...
/dev/sdb1: Moved: 79.2%
...
/dev/sdb1: Moved: 100.0%
```

Después de que el comando **pvmove** haya terminado de ejecutarse, la distribución de extensiones es la siguiente:

```
# 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
```

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

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

Ahora el disco puede ser eliminado físicamente o asignado a otros usuarios.

#### 4.5.2. Mover extensiones a un nuevo disco

En este ejemplo, el volumen lógico se distribuye en tres volúmenes físicos en el grupo de volúmenes **myvg** como sigue:

```
# 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
```

Este procedimiento de ejemplo mueve las extensiones de **/dev/sdb1** a un nuevo dispositivo, **/dev/sdd1**.

1. Crear un nuevo volumen físico desde **/dev/sdd1**.

```
# pvcreate /dev/sdd1
Physical volume "/dev/sdd1" successfully created
```

- Añade el nuevo volumen físico **/dev/sdd1** al grupo de volumen existente **myvg**.

```
# vgextend myvg /dev/sdd1
Volume group "myvg" successfully extended
# 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
```

- Utilice el comando **pvmove** para mover los datos de **/dev/sdb1** a **/dev/sdd1**.

```
# pvmove /dev/sdb1 /dev/sdd1
/dev/sdb1: Moved: 10.0%
...
/dev/sdb1: Moved: 79.7%
...
/dev/sdb1: Moved: 100.0%

# 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
```

- Después de haber movido los datos fuera de **/dev/sdb1**, puede eliminarlos del grupo de volúmenes.

```
# vgreduce myvg /dev/sdb1
Removed "/dev/sdb1" from volume group "myvg"
```

Ahora puede reasignar el disco a otro grupo de volúmenes o eliminar el disco del sistema.

## 4.6. CONFIGURACIÓN DE NÚMEROS DE DISPOSITIVOS PERSISTENTES

Los números de dispositivo mayor y menor se asignan dinámicamente al cargar el módulo. Algunas aplicaciones funcionan mejor si el dispositivo de bloque se activa siempre con el mismo número de dispositivo (mayor y menor). Puede especificarlos con los comandos **lvcreate** y **lvchange** utilizando los siguientes argumentos:

```
--persistente y --mayor major --menor minor
```

Utilice un número menor grande para estar seguro de que no se ha asignado ya a otro dispositivo de forma dinámica.

Si está exportando un sistema de archivos usando NFS, especificar el parámetro **fsid** en el archivo de exportación puede evitar la necesidad de establecer un número de dispositivo persistente dentro de LVM.

## 4.7. ESPECIFICACIÓN DEL TAMAÑO DE EXTENSIÓN DE LVM

Cuando se utilizan volúmenes físicos para crear un grupo de volúmenes, su espacio de disco se divide en extensiones de 4MB, por defecto. Esta extensión es la cantidad mínima por la que el volumen lógico puede aumentar o disminuir su tamaño. Un gran número de extensiones no tendrá ningún impacto en el rendimiento de E/S del volumen lógico.

Puedes especificar el tamaño de extensión con la opción **-s** del comando **vgcreate** si el tamaño de extensión por defecto no es adecuado. Puedes poner límites al número de volúmenes físicos o lógicos que puede tener el grupo de volúmenes utilizando los argumentos **-p** y **-l** del comando **vgcreate**.

## 4.8. GESTIÓN DE LOS VOLÚMENES LÓGICOS LVM MEDIANTE LOS ROLES DE SISTEMA DE RHEL

Esta sección describe cómo aplicar el rol **storage** para realizar las siguientes tareas:

- Crear un volumen lógico LVM en un grupo de volúmenes compuesto por varios discos.
- Crea un sistema de archivos ext4 con una etiqueta determinada en el volumen lógico.
- Montar persistentemente el sistema de archivos ext4.

### Requisitos previos

- Un libro de jugadas de Ansible que incluye el rol **storage**

Para obtener información sobre cómo aplicar un libro de jugadas de Ansible, consulte [Aplicar un rol](#).

### 4.8.1. Ejemplo de libro de jugadas de Ansible para gestionar volúmenes lógicos

Esta sección proporciona un ejemplo de libro de jugadas de Ansible. Este libro de jugadas aplica el rol **storage** para crear un volumen lógico LVM en un grupo de volúmenes.

#### Ejemplo 4.1. Un libro de jugadas que crea un volumen lógico mylv en el grupo de volúmenes myvg

```
- hosts: all
vars:
  storage_pools:
    - name: myvg
      disks:
        - sda
        - sdb
        - sdc
      volumes:
        - name: mylv
          size: 2G
          fs_type: ext4
```

```

    mount_point: /mnt
roles:
  - rhel-system-roles.storage

```

- El grupo de volúmenes **myvg** está formado por los siguientes discos:
  - **/dev/sda**
  - **/dev/sdb**
  - **/dev/sdc**
- Si el grupo de volumen **myvg** ya existe, el libro de jugadas añade el volumen lógico al grupo de volumen.
- Si el grupo de volumen **myvg** no existe, el libro de jugadas lo crea.
- El libro de jugadas crea un sistema de archivos Ext4 en el volumen lógico **mylv**, y monta persistentemente el sistema de archivos en **/mnt**.

#### Recursos adicionales

- Para más detalles sobre los parámetros utilizados en el rol de sistema **storage**, consulte el archivo `/usr/share/ansible/roles/rhel-system-roles.storage/README.md`.

#### 4.8.2. Recursos adicionales

- Para obtener más información sobre el rol **storage**, consulte [Gestión del almacenamiento local mediante los roles de sistema de RHEL](#).

## 4.9. ELIMINACIÓN DE VOLÚMENES LÓGICOS LVM

Este procedimiento elimina un volumen lógico existente utilizando la interfaz LVM de la línea de comandos.

Los siguientes comandos eliminan el volumen lógico **/dev/vg-name/lv-name** del grupo de volúmenes **vg-name**.

#### Procedimiento

1. Si el volumen lógico está actualmente montado, desmonte el volumen.
2. Si el volumen lógico existe en un entorno de clúster, desactive el volumen lógico en todos los nodos en los que esté activo. Utilice el siguiente comando en cada uno de esos nodos:

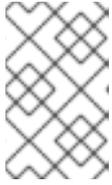
```
[root@node-n]# lvchange --activate n vg-name/lv-name
```

3. Elimine el volumen lógico utilizando la utilidad **lvremove**:

```

# lvremove /dev/vg-name/lv-name
Do you really want to remove active logical volume "lv-name"? [y/n]: y
Logical volume "lv-name" successfully removed

```



## NOTA

En este caso, el volumen lógico no ha sido desactivado. Si se desactiva explícitamente el volumen lógico antes de eliminarlo, no se vería el aviso que verifica si se quiere eliminar un volumen lógico activo.

### Recursos adicionales

- La página de manual **lvremove(8)**

## CAPÍTULO 5. MODIFICAR EL TAMAÑO DE UN VOLUMEN LÓGICO

Después de crear un volumen lógico, puedes modificar el tamaño del volumen.

### 5.1. CRECIMIENTO DE LOS VOLÚMENES LÓGICOS

Para aumentar el tamaño de un volumen lógico, utilice el comando **lvextend**.

Al ampliar el volumen lógico, puedes indicar cuánto quieres ampliar el volumen, o qué tamaño quieres que tenga después de ampliarlo.

El siguiente comando amplía el volumen lógico **/dev/myvg/homevol** a 12 gigabytes.

```
# 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
```

Al igual que con el comando **lvcreate**, puedes utilizar el argumento **-l** del comando **lvextend** para especificar el número de extensiones en las que aumentar el tamaño del volumen lógico. También puedes utilizar este argumento para especificar un porcentaje del grupo de volúmenes, o un porcentaje del espacio libre restante en el grupo de volúmenes. El siguiente comando extiende el volumen lógico llamado **testlv** para llenar todo el espacio no asignado en el grupo de volumen **myvg**.

```
# lvextend -l +100%FREE /dev/myvg/testlv
Extending logical volume testlv to 68.59 GB
Logical volume testlv successfully resized
```

Después de haber ampliado el volumen lógico es necesario aumentar el tamaño del sistema de archivos para que coincida.

Por defecto, la mayoría de las herramientas de redimensionamiento de sistemas de archivos aumentarán el tamaño del sistema de archivos hasta el tamaño del volumen lógico subyacente, por lo que no hay que preocuparse de especificar el mismo tamaño para cada uno de los dos comandos.

### 5.2. CRECIMIENTO DE UN SISTEMA DE ARCHIVOS EN UN VOLUMEN LÓGICO

Para hacer crecer un sistema de archivos en un volumen lógico, realiza los siguientes pasos:

1. Determine si hay suficiente espacio sin asignar en el grupo de volúmenes existente para ampliar el volumen lógico. Si no es así, realice el siguiente procedimiento:
  - a. Cree un nuevo volumen físico con el comando **pvcreate**.

- b. Utilice el comando **vgextend** para ampliar el grupo de volúmenes que contiene el volumen lógico con el sistema de archivos que está creciendo para incluir el nuevo volumen físico.
2. Una vez que el grupo de volúmenes sea lo suficientemente grande como para incluir el sistema de archivos más grande, amplíe el volumen lógico con el comando **lvresize**.
3. Redimensiona el sistema de archivos del volumen lógico.

Tenga en cuenta que puede utilizar la opción **-r** del comando **lvresize** para ampliar el volumen lógico y redimensionar el sistema de archivos subyacente con un solo comando

### 5.3. REDUCCIÓN DE VOLÚMENES LÓGICOS

Puede reducir el tamaño de un volumen lógico con el comando **lvreduce**.



#### NOTA

La reducción no es compatible con un sistema de archivos GFS2 o XFS, por lo que no se puede reducir el tamaño de un volumen lógico que contenga un sistema de archivos GFS2 o XFS.

Si el volumen lógico que está reduciendo contiene un sistema de archivos, para evitar la pérdida de datos debe asegurarse de que el sistema de archivos no está utilizando el espacio del volumen lógico que se está reduciendo. Por este motivo, se recomienda utilizar la opción **--resizefs** del comando **lvreduce** cuando el volumen lógico contenga un sistema de archivos. Cuando se utiliza esta opción, el comando **lvreduce** intenta reducir el sistema de archivos antes de reducir el volumen lógico. Si la reducción del sistema de archivos falla, como puede ocurrir si el sistema de archivos está lleno o el sistema de archivos no admite la reducción, el comando **lvreduce** fallará y no intentará reducir el volumen lógico.



#### AVISO

En la mayoría de los casos, el comando **lvreduce** advierte sobre la posible pérdida de datos y pide una confirmación. Sin embargo, no debes confiar en estos avisos de confirmación para evitar la pérdida de datos porque en algunos casos no verás estos avisos, como cuando el volumen lógico está inactivo o no se utiliza la opción **--resizefs**.

Tenga en cuenta que el uso de la opción **--test** del comando **lvreduce** no indica si la operación es segura, ya que esta opción no comprueba el sistema de archivos ni prueba el redimensionamiento del sistema de archivos.

El siguiente comando reduce el volumen lógico **lv1** en el grupo de volúmenes **vg00** a 64 megabytes. En este ejemplo, **lv1** contiene un sistema de archivos, que este comando redimensiona junto con el volumen lógico. Este ejemplo muestra la salida del comando.

```
# lvreduce --resizefs -L 64M vg00/lv1
fsck from util-linux 2.23.2
/dev/mapper/vg00-lv1: clean, 11/25688 files, 8896/102400 blocks
resize2fs 1.42.9 (28-Dec-2013)
```

```
Resizing the filesystem on /dev/mapper/vg00-lvol1 to 65536 (1k) blocks.
The filesystem on /dev/mapper/vg00-lvol1 is now 65536 blocks long.
```

```
Size of logical volume vg00/lvol1 changed from 100.00 MiB (25 extents) to 64.00 MiB (16 extents).
Logical volume vg00/lvol1 successfully resized.
```

Especificar el signo - antes del valor de redimensionamiento indica que el valor se restará del tamaño real del volumen lógico. El siguiente ejemplo muestra el comando que utilizarías si, en lugar de reducir un volumen lógico a un tamaño absoluto de 64 megabytes, quisieras reducir el volumen en un valor de 64 megabytes.

```
# lvreduce --resizefs -L -64M vg00/lvol1
```

## 5.4. AMPLIACIÓN DE UN VOLUMEN LÓGICO RAYADO

Para aumentar el tamaño de un volumen lógico rayado, debe haber suficiente espacio libre en los volúmenes físicos subyacentes que componen el grupo de volúmenes para soportar la raya. Por ejemplo, si tienes una franja de dos direcciones que utiliza todo un grupo de volúmenes, añadir un solo volumen físico al grupo de volúmenes no te permitirá ampliar la franja. En su lugar, debes añadir al menos dos volúmenes físicos al grupo de volúmenes.

Por ejemplo, considere un grupo de volumen **vg** que consiste en dos volúmenes físicos subyacentes, 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
```

Puedes crear una franja utilizando todo el espacio del 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)
```

Observe que el grupo de volúmenes ya no tiene espacio libre.

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

El siguiente comando agrega otro volumen físico al grupo de volúmenes, que entonces tiene 135 gigabytes 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 punto no se puede ampliar el volumen lógico rayado hasta el tamaño completo del grupo de volúmenes, porque se necesitan dos dispositivos subyacentes para rayar 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 ampliar el volumen lógico rayado, añade otro volumen físico y luego amplíe el volumen lógico. En este ejemplo, habiendo añadido dos volúmenes físicos al grupo de volúmenes podemos ampliar el volumen lógico hasta el 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/stripe1 -L 542G
```

```
Using stripesize of last segment 64.00 KB
```

```
Extending logical volume stripe1 to 542.00 GB
```

```
Logical volume stripe1 successfully resized
```

Si no tiene suficientes dispositivos físicos subyacentes para ampliar el volumen lógico rayado, es posible ampliar el volumen de todas formas si no importa que la ampliación no esté rayada, lo que puede dar lugar a un rendimiento desigual. Cuando se añade espacio al volumen lógico, la operación por defecto es utilizar los mismos parámetros de striping del último segmento del volumen lógico existente, pero se pueden anular esos parámetros. El siguiente ejemplo amplía el volumen lógico rayado existente para utilizar el espacio libre restante después de que falle el comando inicial **lvextend**.

```
# 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
```

```
# lvextend -i1 -l+100%FREE vg/stripe1
```

## CAPÍTULO 6. GESTIÓN DE GRUPOS DE VOLÚMENES LVM

Esta sección describe los comandos que realizan los distintos aspectos de la administración de grupos de volumen.

### 6.1. GRUPOS DE VOLUMEN

Los volúmenes físicos se combinan en grupos de volúmenes (VG). Esto crea una reserva de espacio en disco a partir de la cual se pueden asignar volúmenes lógicos.

Dentro de un grupo de volúmenes, el espacio de disco disponible para la asignación se divide en unidades de tamaño fijo llamadas extensiones. Una extensión es la unidad más pequeña de espacio que se puede asignar. Dentro de un volumen físico, los extents se denominan extents físicos.

Un volumen lógico se asigna en extensiones lógicas del mismo tamaño que las extensiones físicas. El tamaño de la extensión es, por tanto, el mismo para todos los volúmenes lógicos del grupo de volúmenes. El grupo de volúmenes asigna las extensiones lógicas a extensiones físicas.

### 6.2. VISUALIZACIÓN DE LOS GRUPOS DE VOLUMEN

Hay dos comandos que puedes utilizar para mostrar las propiedades de los grupos de volumen LVM: **vgs** y **vgdisplay**. El comando **vgscan**, que escanea todos los dispositivos de bloque LVM soportados en el sistema en busca de grupos de volúmenes, también se puede utilizar para mostrar los grupos de volúmenes existentes.

El comando **vgs** proporciona información de grupos de volúmenes de forma configurable, mostrando una línea por grupo de volúmenes. El comando **vgs** proporciona un gran control del formato, y es útil para la creación de scripts.

El comando **vgdisplay** muestra las propiedades del grupo de volúmenes (como tamaño, extensiones, número de volúmenes físicos, etc.) de forma fija. El siguiente ejemplo muestra la salida del comando **vgdisplay** para el grupo de volumen **new\_vg**. Si no se especifica un grupo de volúmenes, se muestran todos los grupos de volúmenes existentes.

```
# 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-OpMO-0118-nlwO-wwqd-fD5D32
```

El siguiente ejemplo muestra 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
```

### 6.3. COMBINACIÓN DE GRUPOS DE VOLUMEN

Para combinar dos grupos de volúmenes en un único grupo de volúmenes, utiliza el comando **vgmerge**. Puedes combinar un volumen inactivo "de origen" con un volumen activo o inactivo "de destino" si los tamaños de extensión física del volumen son iguales y los resúmenes de volumen físico y lógico de ambos grupos de volumen se ajustan a los límites de los grupos de volumen de destino.

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

```
# vgmerge -v databases my_vg
```

### 6.4. DIVIDIR UN GRUPO DE VOLÚMENES

En este procedimiento de ejemplo, un grupo de volumen existente consta de tres volúmenes físicos. Si hay suficiente espacio no utilizado en los volúmenes físicos, se puede crear un nuevo grupo de volúmenes sin añadir nuevos discos.

En la configuración inicial, el volumen lógico **mylv** se extrae del grupo de volúmenes **myvg**, que a su vez consta de tres volúmenes físicos, **/dev/sda1**, **/dev/sdb1** y **/dev/sdc1**.

Tras completar este procedimiento, el grupo de volúmenes **myvg** estará formado por **/dev/sda1** y **/dev/sdb1**. Un segundo grupo de volúmenes, **yourvg**, estará formado por **/dev/sdc1**.

1. Utilice el comando **pvscan** para determinar cuánto espacio libre hay actualmente en el grupo de volúmenes.

```
# 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 ]
```

2. Mueve todas las extensiones físicas utilizadas en **/dev/sdc1** a **/dev/sdb1** con el comando **pvmove**. El comando **pvmove** puede tardar mucho en ejecutarse.

En un clúster, el comando **pvmove** sólo puede mover el volumen lógico que está activo exclusivamente en un solo nodo.

```
# 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 mover los datos, puede ver que todo el espacio en **/dev/sdc1** está libre.

```

# 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 ]

```

- Para crear el nuevo grupo de volumen **yourvg**, utilice el comando **vgsplit** para dividir el grupo de volumen **myvg**.

El siguiente comando divide el grupo de volumen **yourvg** del grupo de volumen **myvg**, moviendo el volumen físico **/dev/sdc1** al nuevo grupo de volumen **yourvg**.

```

# lvchange -a n /dev/myvg/mylv
# 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.

```

# 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

```

- Después de crear el nuevo grupo de volúmenes, cree el nuevo volumen lógico **yourlv**.

```

# lvcreate -L 5G -n yourlv yourvg
Logical volume "yourlv" created

```

- Crea un sistema de archivos en el nuevo volumen lógico y móntalo.

```

# mkfs.ext4 /dev/yourvg/yourlv
mke2fs 1.44.3 (10-July-2018)
Creating filesystem with 524288 4k blocks and 131072 inodes
Filesystem UUID: 616da032-8a48-4cd7-8705-bd94b7a1c8c4
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

# mount /dev/yourvg/yourlv /mnt

```

## 6.5. RENOMBRAR GRUPOS DE VOLÚMENES LVM

Este procedimiento cambia el nombre de un grupo de volúmenes existente utilizando la interfaz LVM de la línea de comandos.

## Procedimiento

1. Si el grupo de volúmenes existe en un entorno de clúster, desactive el grupo de volúmenes en todos los nodos en los que esté activo. Utilice el siguiente comando en cada uno de esos nodos:

```
[root@node-n]# vgchange --activate n vg-name
```

2. Utilice la utilidad **vgrename** para renombrar un grupo de volúmenes existente:

```
# vgrename original-vg-name new-vg-name
```

Opcionalmente, puede especificar las rutas completas de los dispositivos:

```
# vgrename /dev/original-vg-name /dev/new-vg-name
```

## Recursos adicionales

- La página de manual **vgrename(8)**

## 6.6. MOVER UN GRUPO DE VOLÚMENES A OTRO SISTEMA

Puedes mover todo un grupo de volúmenes LVM a otro sistema. Se recomienda utilizar los comandos **vgexport** y **vgimport** cuando se hace esto.



### NOTA

Puede utilizar el argumento **--force** del comando **vgimport**. Esto le permite importar grupos de volúmenes a los que les faltan volúmenes físicos y ejecutar posteriormente el comando **vgreduce --removemissing**.

El comando **vgexport** hace que un grupo de volúmenes inactivo sea inaccesible para el sistema, lo que permite separar sus volúmenes físicos. El comando **vgimport** hace que un grupo de volúmenes vuelva a ser accesible para una máquina después de que el comando **vgexport** lo haya dejado inactivo.

Para mover un grupo de volúmenes de un sistema a otro, realice los siguientes pasos:

1. Asegúrese de que ningún usuario accede a los archivos de los volúmenes activos del grupo de volúmenes y, a continuación, desmonte los volúmenes lógicos.
2. Utilice el argumento **-a n** del comando **vgchange** para marcar el grupo de volumen como inactivo, lo que impide cualquier otra actividad en el grupo de volumen.
3. Utilice el comando **vgexport** para exportar el grupo de volúmenes. Esto evita que el sistema del que se está eliminando pueda acceder a él.

Después de exportar el grupo de volúmenes, el volumen físico aparecerá como si estuviera en un grupo de volúmenes exportado cuando ejecute el comando **pvscan**, como en el siguiente ejemplo.

```
# 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]
...
```

La próxima vez que se apague el sistema, puede desconectar los discos que constituyen el grupo de volúmenes y conectarlos al nuevo sistema.

4. Cuando los discos estén conectados al nuevo sistema, utilice el comando **vgimport** para importar el grupo de volúmenes, haciéndolo accesible al nuevo sistema.
5. Active el grupo de volumen con el argumento **-a y** del comando **vgchange**.
6. Montar el sistema de archivos para que esté disponible para su uso.

## 6.7. ELIMINACIÓN DE VOLÚMENES FÍSICOS DE UN GRUPO DE VOLÚMENES

Para eliminar los volúmenes físicos no utilizados de un grupo de volúmenes, utilice el comando **vgreduce**. El comando **vgreduce** reduce la capacidad de un grupo de volúmenes eliminando uno o más volúmenes físicos vacíos. Esto libera esos volúmenes físicos para utilizarlos en otros grupos de volúmenes o para eliminarlos del sistema.

Antes de eliminar un volumen físico de un grupo de volúmenes, puedes asegurarte de que el volumen físico no es utilizado por ningún volumen lógico utilizando 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-MOkn-76iP-iftz-OVSen7
```

Si el volumen físico sigue siendo utilizado, tendrás que migrar los datos a otro volumen físico utilizando el comando **pvmove**. A continuación, utilice el comando **vgreduce** para eliminar el volumen físico.

El siguiente comando elimina el volumen físico **/dev/hda1** del grupo de volúmenes **my\_volume\_group**.

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

Si un volumen lógico contiene un volumen físico que falla, no podrás utilizar ese volumen lógico. Para eliminar los volúmenes físicos que faltan de un grupo de volúmenes, puede utilizar el parámetro **--removemissing** del comando **vgreduce**, si no hay volúmenes lógicos que estén asignados en los volúmenes físicos que faltan.

Si el volumen físico que falla contiene una imagen en espejo de un volumen lógico de un tipo de segmento **mirror**, puedes eliminar esa imagen del espejo con el comando **vgreduce --removemissing - -mirroronly --force**. Esto elimina sólo los volúmenes lógicos que son imágenes en espejo del volumen físico.

## 6.8. ELIMINACIÓN DE GRUPOS DE VOLÚMENES LVM

Este procedimiento elimina un grupo de volúmenes existente utilizando la interfaz LVM de la línea de comandos.

### Requisitos previos

- El grupo de volúmenes no contiene volúmenes lógicos. Para eliminar volúmenes lógicos de un grupo de volúmenes, consulte [Sección 4.9, “Eliminación de volúmenes lógicos LVM”](#) .

### Procedimiento

1. Si el grupo de volúmenes existe en un entorno de clúster, detenga el espacio de bloqueo del grupo de volúmenes en todos los demás nodos. Utilice el siguiente comando en todos los nodos excepto en el nodo en el que está realizando la eliminación:

```
[root@node-n]# vgchange --lockstop vg-name
```

Espere a que se detenga el bloqueo.

2. Para eliminar el grupo de volúmenes, utilice la utilidad **vgremove**:

```
# vgremove vg-name  
Volume group "vg-name" successfully removed
```

### Recursos adicionales

- La página de manual **vgremove(8)**

## 6.9. RECURSOS ADICIONALES

- La página de manual **vgchange(8)**

## CAPÍTULO 7. GESTIÓN DE VOLÚMENES FÍSICOS LVM

Hay una variedad de comandos y procedimientos que puede utilizar para gestionar los volúmenes físicos LVM.

### 7.1. BÚSQUEDA DE DISPOSITIVOS DE BLOQUE PARA UTILIZARLOS COMO VOLÚMENES FÍSICOS

Puede buscar dispositivos de bloque que puedan ser utilizados como 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/pkl_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
```

### 7.2. ESTABLECER EL TIPO DE PARTICIÓN PARA UN VOLUMEN FÍSICO

Si está utilizando un dispositivo de disco completo para su volumen físico, el disco no debe tener ninguna tabla de particiones. Para las particiones de disco de DOS, el id de la partición debe establecerse en 0x8e utilizando el comando **fdisk** o **cfdisk** o un equivalente. En el caso de los dispositivos de disco completo, sólo debe borrarse la tabla de particiones, lo que destruirá efectivamente todos los datos de ese disco. Puede eliminar una tabla de particiones existente poniendo a cero el primer sector con el siguiente comando:

■

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

### 7.3. CAMBIAR EL TAMAÑO DE UN VOLUMEN FÍSICO LVM

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

### 7.4. ELIMINACIÓN DE VOLÚMENES FÍSICOS

Si un dispositivo ya no es necesario para su uso por LVM, puedes eliminar la etiqueta LVM con el comando **pvremove**. La ejecución del comando **pvremove** pone a cero los metadatos LVM en un volumen físico vacío.

Si el volumen físico que quieres eliminar forma parte actualmente de un grupo de volúmenes, debes eliminarlo del grupo de volúmenes con el comando **vgreduce**.

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

### 7.5. AÑADIR VOLÚMENES FÍSICOS A UN GRUPO DE VOLÚMENES

Para añadir volúmenes físicos adicionales a un grupo de volúmenes existente, utilice el comando **vgextend**. El comando **vgextend** aumenta la capacidad de un grupo de volúmenes añadiendo uno o más volúmenes físicos libres.

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

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

### 7.6. ELIMINACIÓN DE VOLÚMENES FÍSICOS DE UN GRUPO DE VOLÚMENES

Para eliminar los volúmenes físicos no utilizados de un grupo de volúmenes, utilice el comando **vgreduce**. El comando **vgreduce** reduce la capacidad de un grupo de volúmenes eliminando uno o más volúmenes físicos vacíos. Esto libera esos volúmenes físicos para utilizarlos en otros grupos de volúmenes o para eliminarlos del sistema.

Antes de eliminar un volumen físico de un grupo de volúmenes, puedes asegurarte de que el volumen físico no es utilizado por ningún volumen lógico utilizando 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-MOkn-76iP-iftz-OVSen7
```

Si el volumen físico sigue siendo utilizado, tendrás que migrar los datos a otro volumen físico utilizando el comando **pvmove**. A continuación, utilice el comando **vgreduce** para eliminar el volumen físico.

El siguiente comando elimina el volumen físico **/dev/hda1** del grupo de volúmenes **my\_volume\_group**.

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

Si un volumen lógico contiene un volumen físico que falla, no podrás utilizar ese volumen lógico. Para eliminar los volúmenes físicos que faltan de un grupo de volúmenes, puede utilizar el parámetro **--removemissing** del comando **vgreduce**, si no hay volúmenes lógicos que estén asignados en los volúmenes físicos que faltan.

Si el volumen físico que falla contiene una imagen en espejo de un volumen lógico de un tipo de segmento **mirror**, puedes eliminar esa imagen del espejo con el comando **vgreduce --removemissing - -mirroronly --force**. Esto elimina sólo los volúmenes lógicos que son imágenes en espejo del volumen físico.

## CAPÍTULO 8. VISUALIZACIÓN DE LOS COMPONENTES DE LVM

LVM proporciona una variedad de formas para mostrar los componentes de LVM, así como para personalizar la pantalla. Esta sección resume el uso de los comandos básicos de visualización de LVM.

### 8.1. VISUALIZACIÓN DE LA INFORMACIÓN DE LVM CON EL COMANDO LVM

El comando **lvm** proporciona varias opciones incorporadas que puede utilizar para mostrar información sobre el soporte y la configuración de LVM.

- **lvm devtypes**  
Muestra los tipos de dispositivos de bloque incorporados reconocidos
- **lvm formats**  
Muestra los formatos de metadatos reconocidos.
- **lvm help**  
Muestra el texto de ayuda de LVM.
- **lvm segtypes**  
Muestra los tipos de segmento de volumen lógico reconocidos.
- **lvm tags**  
Muestra cualquier etiqueta definida en este host.
- **lvm version**  
Muestra la información de la versión actual.

### 8.2. VISUALIZACIÓN DE LOS VOLÚMENES FÍSICOS

Hay tres comandos que puede utilizar para mostrar las propiedades de los volúmenes físicos LVM: **pvs**, **pvdisplay**, y **pvscan**.

El comando **pvs** proporciona información sobre el volumen físico de forma configurable, mostrando una línea por volumen físico. El comando **pvs** proporciona un gran control del formato, y es útil para la creación de scripts.

El comando **pvdisplay** proporciona una salida verbosa 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 del comando **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-ldwM-01S9-XO8M-mcpsVe
```

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

El siguiente comando muestra todos los dispositivos físicos encontrados:

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

Puede definir un filtro en el archivo **lvm.conf** para que este comando evite escanear volúmenes físicos específicos.

### 8.3. VISUALIZACIÓN DE LOS GRUPOS DE VOLUMEN

Hay dos comandos que puedes utilizar para mostrar las propiedades de los grupos de volumen LVM: **vgs** y **vgdisplay**. El comando **vgscan**, que escanea todos los dispositivos de bloque LVM soportados en el sistema en busca de grupos de volúmenes, también se puede utilizar para mostrar los grupos de volúmenes existentes.

El comando **vgs** proporciona información de grupos de volúmenes de forma configurable, mostrando una línea por grupo de volúmenes. El comando **vgs** proporciona un gran control del formato, y es útil para la creación de scripts.

El comando **vgdisplay** muestra las propiedades del grupo de volúmenes (como tamaño, extensiones, número de volúmenes físicos, etc.) de forma fija. El siguiente ejemplo muestra la salida del comando **vgdisplay** para el grupo de volumen **new\_vg**. Si no se especifica un grupo de volúmenes, se muestran todos los grupos de volúmenes existentes.

```
# 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-OpMO-0118-nlwO-wwqd-fD5D32
```

El siguiente ejemplo muestra 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
```

## 8.4. VISUALIZACIÓN DE VOLÚMENES LÓGICOS

Hay tres comandos que puede utilizar para mostrar las propiedades de los volúmenes lógicos LVM: **lvs**, **lvdisplay**, y **lvscan**.

El comando **lvs** proporciona información sobre el volumen lógico de forma configurable, mostrando una línea por volumen lógico. El comando **lvs** proporciona un gran control del formato, y es útil para la creación de scripts.

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

El siguiente comando muestra los atributos de **lvol2** en **vg00**. Si se han creado volúmenes lógicos instantáneos para este volumen lógico original, este comando muestra una lista de todos los volúmenes lógicos instantáneos y su estado (activo o inactivo) también.

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

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

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

## CAPÍTULO 9. INFORMES PERSONALIZADOS PARA LVM

LVM proporciona una amplia gama de opciones de configuración y de línea de comandos para producir informes personalizados y para filtrar la salida del informe. Para una descripción completa de las características y capacidades de los informes de LVM, consulte la página de manual **lvmreport(7)**.

Puede producir informes concisos y personalizados de los objetos LVM con los comandos **pvs**, **lvs**, y **vgs**. Los informes que generan estos comandos incluyen una línea de salida para cada objeto. Cada línea contiene una lista ordenada de campos de propiedades relacionadas con el objeto. Hay cinco maneras de seleccionar los objetos a reportar: por volumen físico, grupo de volumen, volumen lógico, segmento de volumen físico y segmento de volumen lógico.

Puedes reportar información sobre volúmenes físicos, grupos de volúmenes, volúmenes lógicos, segmentos de volúmenes físicos y segmentos de volúmenes lógicos a la vez con el comando **lvm fullreport**. Para obtener información sobre este comando y sus capacidades, consulte la página de manual **lvm-fullreport(8)**.

LVM admite informes de registro, que contienen un registro de operaciones, mensajes y el estado de cada objeto con la identificación completa del mismo, recogidos durante la ejecución de los comandos LVM. Para más información sobre el informe de registro de LVM, consulte la página de manual **lvmreport(7)**.

### 9.1. CONTROLAR EL FORMATO DE LA PANTALLA LVM

El uso de los comandos **pvs**, **lvs**, o **vgs** determina el conjunto de campos que se muestran por defecto y el orden de clasificación. Puede controlar la salida de estos comandos con los siguientes argumentos:

- Puede cambiar los campos que se muestran a algo distinto de lo predeterminado utilizando el argumento **-o**. Por ejemplo, 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 a la salida con el signo más ( **+** ), que se utiliza en combinación con el argumento **-o**. El siguiente ejemplo muestra el UUID del volumen físico además de los campos por defecto.

```
# 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-ldwM-01S9-X08M-mcpsVe
/dev/sdd1 new_vg lvm2 a- 17.14G 17.14G yfvfZK-Cf31-j75k-dECm-0RZ3-0dGW-UqkCS
```

- Añadir el argumento **-v** a un comando incluye algunos campos adicionales. Por ejemplo, el comando **pvs -v** mostrará los campos **DevSize** y **PV UUID** además de los campos por defecto.

```
# 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-ldwM-01S9-XO8M-
mcpSve
/dev/sdd1 new_vg lvm2 a- 17.14G 17.14G 17.14G yfvZK-Cf31-j75k-dECm-0RZ3-0dGW-
tUqkCS
```

- El argumento **--noheadings** suprime la línea de encabezamiento. Esto puede ser útil para escribir scripts.

El siguiente ejemplo utiliza el argumento **--noheadings** en combinación con el argumento **pv\_name**, que generará una lista de todos los volúmenes físicos.

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

- El argumento **--separator separator** utiliza *separator* para separar cada campo. El siguiente ejemplo separa los campos de salida por defecto del comando **pvs** con un 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 al utilizar el argumento **separator**, utilice el argumento **separator** junto con el argumento **--aligned**.

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

Puede utilizar el argumento **-P** del comando **lvs** o **vgs** para mostrar información sobre un volumen que ha fallado y que de otro modo no aparecería en la salida.

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

Los campos de los grupos de volúmenes pueden mezclarse con los campos de los volúmenes físicos (y los segmentos de los volúmenes físicos) o con los campos de los volúmenes lógicos (y los segmentos de los volúmenes lógicos), pero los campos de los volúmenes físicos y los volúmenes lógicos 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
```

## 9.2. CAMPOS DE VISUALIZACIÓN DEL OBJETO LVM

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

Por comodidad, se puede omitir el prefijo de un nombre de campo si coincide con el predeterminado para el comando. Por ejemplo, con el comando **pvs**, **name** significa **pv\_name**, pero con el comando **vgs**, **name** se interpreta como **vg\_name**.

Ejecutar el siguiente comando es el equivalente a ejecutar **pvs -o pv\_free**.

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



## NOTA

El número de caracteres en los campos de atributos en la salida de **pvs**, **vgs**, y **lvs** puede aumentar en versiones posteriores. Los campos de caracteres existentes no cambiarán de posición, pero es posible que se añadan nuevos campos al final. Debe tener esto en cuenta cuando escriba scripts que busquen determinados caracteres de atributos, buscando el carácter en función de su posición relativa al principio del campo, pero no por su posición relativa al final del campo. Por ejemplo, para buscar el carácter **p** en el noveno bit del campo **lv\_attr**, podría buscar la cadena `"^/.....p/"`, pero no debería buscar la cadena `"/*p$/"`.

Tabla 9.1, “Los campos de visualización del comando **pvs**” enumera los argumentos de visualización del comando **pvs**, junto con el nombre del campo tal y como aparece en la visualización de la cabecera y una descripción del campo.

Tabla 9.1. Los campos de visualización del comando **pvs**

Argumento	Cabecera	Descripción
<b>dev_size</b>	DevSize	El tamaño del dispositivo subyacente en el que se creó el volumen físico
<b>pe_start</b>	1er PE	Desplazamiento al inicio de la primera extensión física en el dispositivo subyacente
<b>pv_attr</b>	Attr	Estado del volumen físico: (a)locable o e(x)portado.
<b>pv_fmt</b>	Fmt	El formato de los metadatos del volumen físico ( <b>lvm2</b> o <b>lvm1</b> )
<b>pv_free</b>	PFree	El espacio libre que queda en el volumen físico
<b>pv_name</b>	PV	El nombre del volumen físico

Argumento	Cabecera	Descripción
<b>pv_pe_alloc_count</b>	Asignar	Número de extensiones físicas utilizadas
<b>pv_pe_count</b>	PE	Número de extensiones físicas
<b>pvseg_size</b>	Tamaño de la SS	El tamaño del segmento del volumen físico
<b>pvseg_start</b>	Inicie	La extensión física inicial del segmento de volumen físico
<b>pv_size</b>	PSize	El tamaño del volumen físico
<b>pv_tags</b>	Etiquetas PV	Etiquetas LVM adjuntas al volumen físico
<b>pv_used</b>	Usado	La cantidad de espacio utilizado actualmente en el volumen físico
<b>pv_uuid</b>	PV UUID	El UUID del volumen físico

El comando **pvs** muestra por defecto los siguientes campos: **pv\_name**, **vg\_name**, **pv\_fmt**, **pv\_attr**, **pv\_size**, **pv\_free**. La visualización está ordenada por **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
```

El uso del argumento **-v** con el comando **pvs** añade los siguientes campos a la visualización por defecto: **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-ldwM-01S9-XO8M-mcpsVe
/dev/sdd1 new_vg lvm2 a- 17.14G 17.13G 17.14G yfvZK-Cf31-j75k-dECm-0RZ3-0dGW-tUqkCS
```

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

El comando **pvs --segments** muestra por defecto los siguientes campos: **pv\_name**, **vg\_name**, **pv\_fmt**, **pv\_attr**, **pv\_size**, **pv\_free**, **pvseg\_start**, **pvseg\_size**. La visualización está ordenada por **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/sdgl1 vg      lvm2 a- 17.14G 17.14G 0 4389

```

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

**# pvs -a**

```

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

```

Tabla 9.2, “campos de visualización de vgs” enumera los argumentos de visualización del comando **vgs**, junto con el nombre del campo tal y como aparece en la visualización de la cabecera y una descripción del campo.

Tabla 9.2. campos de visualización de vgs

Argumento	Cabecera	Descripción
<b>lv_count</b>	#LV	El número de volúmenes lógicos que contiene el grupo de volúmenes

Argumento	Cabecera	Descripción
<b>max_lv</b>	MaxLV	El número máximo de volúmenes lógicos permitidos en el grupo de volúmenes (0 si es ilimitado)
<b>max_pv</b>	MaxPV	El número máximo de volúmenes físicos permitidos en el grupo de volúmenes (0 si es ilimitado)
<b>pv_count</b>	#PV	El número de volúmenes físicos que definen el grupo de volúmenes
<b>snap_count</b>	#SN	El número de instantáneas que contiene el grupo de volúmenes
<b>vg_attr</b>	Attr	Estado del grupo de volumen: (w)riteable, (r)eadonly, resi(z)eable, e(x)ported, (p)artial y (c)lustered.
<b>vg_extent_count</b>	#Ext	El número de extensiones físicas en el grupo de volúmenes
<b>vg_extent_size</b>	Ext	El tamaño de las extensiones físicas del grupo de volúmenes
<b>vg_fmt</b>	Fmt	El formato de los metadatos del grupo de volúmenes ( <b>lvm2</b> o <b>lvm1</b> )
<b>vg_free</b>	VFree	Tamaño del espacio libre que queda en el grupo de volúmenes
<b>vg_free_count</b>	Gratis	Número de extensiones físicas libres en el grupo de volúmenes
<b>vg_name</b>	VG	El nombre del grupo de volumen
<b>vg_seqno</b>	Seq	Número que representa la revisión del grupo de volúmenes
<b>vg_size</b>	VSize	El tamaño del grupo de volumen
<b>vg_sysid</b>	IDENTIFICACIÓN DEL SISTEMA	ID del sistema LVM1
<b>vg_tags</b>	Etiquetas VG	Etiquetas LVM adjuntas al grupo de volúmenes

Argumento	Cabecera	Descripción
<b>vg_uuid</b>	VG UUID	El UUID del grupo de volúmenes

El comando **vgs** muestra por defecto los siguientes campos: **vg\_name**, **pv\_count**, **lv\_count**, **snap\_count**, **vg\_attr**, **vg\_size**, **vg\_free**. La visualización está ordenada por **vg\_name**.

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

El uso del argumento **-v** con el comando **vgs** añade los siguientes campos a la visualización por defecto: **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-OpMO-0118-nlwO-wwqd-fD5D32
```

Tabla 9.3, “campos de visualización de lvs” enumera los argumentos de visualización del comando **lvs**, junto con el nombre del campo tal y como aparece en la visualización de la cabecera y una descripción del campo.



#### NOTA

En versiones posteriores de Red Hat Enterprise Linux, la salida del comando **lvs** puede diferir, con campos adicionales en la salida. El orden de los campos, sin embargo, seguirá siendo el mismo y cualquier campo adicional aparecerá al final de la pantalla.

Tabla 9.3. campos de visualización de lvs

Argumento	Cabecera	Descripción
<b>* chunksize</b> <b>* chunk_size</b>	Chunk	Tamaño de la unidad en un volumen de instantáneas
<b>copy_percent</b>	Copiar%	El porcentaje de sincronización de un volumen lógico reflejado; también se utiliza cuando se mueven extensiones físicas con el comando <b>pv_move</b>
<b>devices</b>	Dispositivos	Los dispositivos subyacentes que componen el volumen lógico: los volúmenes físicos, los volúmenes lógicos y los extensiones físicas y lógicas de inicio

Argumento	Cabecera	Descripción
<b>lv_ancestors</b>	Ancestros	En el caso de las instantáneas del grupo ligero, los ancestros del volumen lógico
<b>lv_descendants</b>	Descendientes	En el caso de las instantáneas del grupo ligero, los descendientes del volumen lógico
<b>lv_attr</b>	Attr	<p>El estado del volumen lógico. Los bits de atributos del volumen lógico son los siguientes:</p> <p>* Bit 1: Tipo de volumen: (m)irrored, (M)irrored sin sincronización inicial, (o)rigin, (O)rigin con snapshot de fusión, (r)aid, ®aid sin sincronización inicial, (s)napshot, (S)napshot de fusión, (p)vmove, (v)irtual, mirror o raid (i)mage, mirror o raid (l)mage out-of-sync, mirror (l)og device, under (c)onversion, thin (V)olume, (t)hin pool, (T)hin pool data, raid o thin pool m(e)tadata o pool metadata spare,</p> <p>* Bit 2: Permisos: (w)riteable, (r)ead-only, ®ead-only activación del volumen de no-lectura</p> <p>* Bit 3: Política de asignación: (a)nywhere, (c)ontiguous, (i)nherited, c(l)ing, (n)ormal. Se escribe con mayúsculas si el volumen está actualmente bloqueado contra los cambios de asignación, por ejemplo mientras se ejecuta el comando <b>pvmove</b>.</p> <p>* Bit 4: (m)inor fijo</p> <p>* Bit 5: Estado: (a)ctivo, (s)uspendido, instantánea (l)nválida, instantánea (S)uspendida inválida, instantánea (m)erge fallada, instantánea suspendida (M)erge fallada, dispositivo (d)evice mapeado presente sin tablas, dispositivo mapeado presente con tabla (i)nactiva</p>

Argumento	Cabecera	* Bit 6: dispositivo (o)pen Descripción
		<p>* Bit 7: Tipo de objetivo: (m)irror, (r)aid, (s)napshot, (t)hin, (u)ncached, (v)irtual. Esto agrupa los volúmenes lógicos relacionados con el mismo objetivo del kernel. Así, por ejemplo, las imágenes de espejo, los registros de espejo así como los espejos mismos aparecen como (m) si usan el controlador original del kernel de espejo device-mapper, mientras que los equivalentes de raid que usan el controlador del kernel md raid aparecen todos como (r). Las instantáneas que utilizan el controlador original del mapeador de dispositivos aparecen como (s), mientras que las instantáneas de los volúmenes finos que utilizan el controlador de aprovisionamiento fino aparecen como (t).</p> <p>* Bit 8: Los bloques de datos recién asignados se sobrescriben con bloques de (z)eros antes de su uso.</p> <p>* Bit 9: Salud del volumen: (p)artificial, (r)efresh needed, (m)ismatches exist, (w)ritemostly. (p)artial significa que uno o más de los volúmenes físicos que utiliza este volumen lógico no está en el sistema. (r)efresh significa que uno o más de los volúmenes físicos que utiliza este volumen lógico RAID ha sufrido un error de escritura. El error de escritura puede deberse a un fallo temporal de ese Volumen Físico o una indicación de que está fallando. El dispositivo debe ser refrescado o reemplazado. (m)ismatches significa que el volumen lógico RAID tiene porciones del array que no son coherentes. Las incoherencias se descubren iniciando una operación <b>check</b> en un volumen lógico RAID. (Las operaciones de depuración, <b>check</b> y <b>repair</b>, pueden realizarse en un volumen lógico RAID mediante el comando</p>

Argumento	Cabecera	Descripción
		<p><b>lvchange</b> (w)ritemostly significa los dispositivos de un volumen lógico RAID 1 que han sido marcados como de escritura.</p> <p>* Bit 10: activación s(k)ip: este volumen está marcado para ser omitido durante la activación.</p>
<b>lv_kernel_major</b>	KMaj	Número de dispositivo principal real del volumen lógico (-1 si está inactivo)
<b>lv_kernel_minor</b>	KMIN	Número de dispositivo menor real del volumen lógico (-1 si está inactivo)
<b>lv_major</b>	Mayor	El número de dispositivo mayor persistente del volumen lógico (-1 si no se especifica)
<b>lv_minor</b>	Min	El número de dispositivo menor persistente del volumen lógico (-1 si no se especifica)
<b>lv_name</b>	LV	El nombre del volumen lógico
<b>lv_size</b>	LSize	El tamaño del volumen lógico
<b>lv_tags</b>	Etiquetas LV	Etiquetas LVM adjuntas al volumen lógico
<b>lv_uuid</b>	LV UUID	El UUID del volumen lógico.
<b>mirror_log</b>	Registro	Dispositivo en el que reside el registro de réplica
<b>modules</b>	Módulos	El correspondiente objetivo del kernel device-mapper necesario para utilizar este volumen lógico
<b>move_pv</b>	Mover	Volumen físico de origen de un volumen lógico temporal creado con el comando <b>pvmove</b>
<b>origin</b>	Origen	El dispositivo de origen de un volumen de instantáneas

Argumento	Cabecera	Descripción
* <b>regionsize</b> * <b>region_size</b>	Región	El tamaño de la unidad de un volumen lógico reflejado
<b>seg_count</b>	#Seg	El número de segmentos del volumen lógico
<b>seg_size</b>	Tamaño de la SS	El tamaño de los segmentos del volumen lógico
<b>seg_start</b>	Inicie	Desplazamiento del segmento en el volumen lógico
<b>seg_tags</b>	Seg Tags	Etiquetas LVM adjuntas a los segmentos del volumen lógico
<b>segtype</b>	Tipo	El tipo de segmento de un volumen lógico (por ejemplo: espejo, rayado, lineal)
<b>snap_percent</b>	Snap%	Porcentaje actual de un volumen de instantáneas que está en uso
<b>stripes</b>	#Str	Número de franjas o espejos en un volumen lógico
* <b>stripesize</b> * <b>stripe_size</b>	Raya	Tamaño unitario de la franja en un volumen lógico rayado

El comando **lvs** proporciona la siguiente visualización por defecto. La visualización por defecto está ordenada por **vg\_name** y **lv\_name** dentro del grupo de volúmenes.

```
# lvs
LV VG Attr LSize Pool Origin Data% Meta% Move Log Cpy%Sync Convert
origin VG owi-a-s--- 1.00g
snap VG swi-a-s--- 100.00m origin 0.00
```

Un uso común del comando **lvs** es añadir **devices** al comando para mostrar los dispositivos subyacentes que componen el volumen lógico. Este ejemplo también especifica la opción **-a** para mostrar los volúmenes internos que son componentes de los volúmenes lógicos, como las réplicas RAID, encerradas entre paréntesis. Este ejemplo incluye un volumen RAID, un volumen rayado y un volumen thinly-pooled.

```
# lvs -a -o +devices
LV VG Attr LSize Pool Origin Data% Meta% Move Log Cpy%Sync Convert
Devices
raid1 VG rwi-a-r--- 1.00g 100.00
raid1_rimage_0(0),raid1_rimage_1(0)
[raid1_rimage_0] VG iwi-a-or--- 1.00g /dev/sde1(7041)
```

```

[raid1_rimage_1] VG      iwi-aor--- 1.00g                /dev/sdf1(7041)
[raid1_rmeta_0] VG      ewi-aor--- 4.00m                /dev/sde1(7040)
[raid1_rmeta_1] VG      ewi-aor--- 4.00m                /dev/sdf1(7040)
stripe1      VG      -wi-a----- 99.95g            /dev/sde1(0),/dev/sdf1(0)
stripe1      VG      -wi-a----- 99.95g            /dev/sdd1(0)
stripe1      VG      -wi-a----- 99.95g            /dev/sdc1(0)
[lvol0_pmspare] rhel_host-083 ewi----- 4.00m                /dev/vda2(0)
pool00      rhel_host-083 twi-aotz-- <4.79g            72.90 54.69
pool00_tdata(0)
[pool00_tdata] rhel_host-083 Twi-ao---- <4.79g                /dev/vda2(1)
[pool00_tmeta] rhel_host-083 ewi-ao---- 4.00m                /dev/vda2(1226)
root        rhel_host-083 Vwi-aotz-- <4.79g pool00      72.90
swap        rhel_host-083 -wi-ao---- 820.00m            /dev/vda2(1227)

```

El uso del argumento **-v** con el comando **lvs** añade los siguientes campos a la visualización por defecto: **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  Convert  LV
UUID
lvol0   new_vg  1 owi-a- 52.00M -1 -1 253 3                LBy1Tz-sr23-Ojsl-LT03-
nHLC-y8XW-EhCI78
newvgsnap1 new_vg  1 swi-a- 8.00M -1 -1 253 5  lvol0  0.20                1ye1OU-1clu-
o79k-20h2-ZGF0-qCJm-Cfbslx

```

Puede utilizar el argumento **--segments** del comando **lvs** para mostrar información con columnas predeterminadas que enfatizan la información del segmento. Cuando se utiliza el argumento **segments**, el prefijo **seg** es opcional. El comando **lvs --segments** muestra por defecto los siguientes campos: **lv\_name**, **vg\_name**, **lv\_attr**, **stripes**, **segtype**, **seg\_size**. La visualización por defecto está ordenada por **vg\_name**, **lv\_name** dentro del grupo de volúmenes, y **seg\_start** dentro del volumen lógico. Si los volúmenes lógicos estuvieran fragmentados, la salida de este comando lo mostraría.

```

# 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

```

El uso del argumento **-v** con el comando **lvs --segments** añade los siguientes campos a la visualización por defecto: **seg\_start**, **stripesize**, **chunksize**.

```

# lvs -v --segments
Finding all logical volumes
LV      VG      Attr Start SSize #Str Type  Stripe Chunk
lvol0   new_vg owi-a- 0 52.00M 1 linear 0 0
newvgsnap1 new_vg swi-a- 0 8.00M 1 linear 0 8.00K

```

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

```
# lvs
LV VG Attr LSize Origin Snap% Move Log Copy%
lv00 new_vg -wi-a- 52.00M
# lvs --segments
LV VG Attr #Str Type SSize
lv00 new_vg -wi-a- 1 linear 52.00M
```

### 9.3. CLASIFICACIÓN DE LOS INFORMES LVM

Normalmente la salida completa del comando **lvs**, **vgs**, o **pvs** tiene que ser generada y almacenada internamente antes de que pueda ser ordenada y las columnas alineadas correctamente. Puede especificar el argumento **--unbuffered** para mostrar la salida sin clasificar tan pronto como se genere.

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

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

```
# 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 por el campo de espacio libre.

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

El siguiente ejemplo muestra que no es necesario mostrar el campo en el que se está clasificando.

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

Para mostrar una ordenación inversa, preceda un campo que especifique después del argumento **-O** con el carácter **-**.

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

### 9.4. ESPECIFICACIÓN DE LAS UNIDADES PARA LA VISUALIZACIÓN DE UN INFORME LVM

Para especificar las unidades para la visualización del informe LVM, utilice el argumento **--units** del comando informe. Puede especificar (b)ytes, (k)ilobytes, (m)egabytes, (g)igabytes, (t)erabytes, (e)xabytes, (p)etabytes y (h)uman-readable. La visualización por defecto es legible para el ser humano. Puede anular el valor predeterminado estableciendo el parámetro **units** en la sección **global** del archivo **/etc/lvm/lvm.conf**.

El siguiente ejemplo especifica la salida del comando **pvs** en megabytes en lugar de los gigabytes por defecto.

```
# pvs --units m
PV      VG   Fmt Attr PSize  PFree
/dev/sda1    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 en potencias de 2 (múltiplos de 1024). Puede especificar que las unidades se muestren en múltiplos de 1000 poniendo en mayúsculas la especificación de la unidad (B, K, M, G, T, H).

El siguiente comando muestra la salida como un múltiplo de 1024, el comportamiento por defecto.

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

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

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

También puede especificar (s)ectores (definidos como 512 bytes) o unidades personalizadas.

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

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

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

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

## 9.5. VISUALIZACIÓN DE LA SALIDA DEL COMANDO LVM EN FORMATO JSON

Puede utilizar la opción **--reportformat** de los comandos de visualización de LVM para mostrar la salida en formato JSON.

El siguiente ejemplo muestra la salida del **lvs** en formato estándar por defecto.

```
# lvs
LV VG Attr LSize Pool Origin Data% Meta% Move Log Cpy%Sync Convert
my_raid my_vg Rwi-a-r--- 12.00m 100.00
root rhel_host-075 -wi-ao---- 6.67g
swap rhel_host-075 -wi-ao---- 820.00m
```

El siguiente comando muestra la salida de la misma configuración LVM cuando se especifica el formato JSON.

```
# lvs --reportformat json
{
  "report": [
    {
      "lv": [
        {"lv_name":"my_raid", "vg_name":"my_vg", "lv_attr":"Rwi-a-r---", "lv_size":"12.00m",
"pool_lv":"","origin":"","data_percent":"","metadata_percent":"","move_pv":"","mirror_log":"","copy_percent":"100.00", "convert_lv":""},
        {"lv_name":"root", "vg_name":"rhel_host-075", "lv_attr":"-wi-ao----", "lv_size":"6.67g",
"pool_lv":"","origin":"","data_percent":"","metadata_percent":"","move_pv":"","mirror_log":"","copy_percent":"","convert_lv":""},
        {"lv_name":"swap", "vg_name":"rhel_host-075", "lv_attr":"-wi-ao----", "lv_size":"820.00m",
"pool_lv":"","origin":"","data_percent":"","metadata_percent":"","move_pv":"","mirror_log":"","copy_percent":"","convert_lv":""}
      ]
    }
  ]
}
```

También puede establecer el formato del informe como una opción de configuración en el archivo `/etc/lvm/lvm.conf`, utilizando el ajuste **output\_format**. Sin embargo, el ajuste **--reportformat** de la línea de comandos tiene prioridad sobre este ajuste.

## 9.6. VISUALIZACIÓN DEL REGISTRO DE COMANDOS LVM

Tanto los comandos LVM orientados al informe como los orientados al procesamiento pueden informar del registro de comandos si se habilita con el ajuste de configuración **log/report\_command\_log**. Se puede determinar el conjunto de campos a mostrar y a ordenar para este informe.

Los siguientes ejemplos configuran LVM para generar un informe de registro completo para los comandos LVM. En este ejemplo, se puede ver que ambos volúmenes lógicos **lv010** y **lv011** fueron procesados con éxito, al igual que el grupo de volumen **VG** que contiene los volúmenes.

```
# lvmconfig --type full log/command_log_selection
command_log_selection="all"

# lvs
```

## Logical Volume

=====

LV LSize Cpy%Sync

lvol1 4.00m 100.00

lvol0 4.00m

## Command Log

=====

Seq LogType Context ObjType ObjName ObjGrp Msg Errno RetCode

1 status processing lv lvol0 vg success 0 1

2 status processing lv lvol1 vg success 0 1

3 status processing vg vg success 0 1

# **lvchange -an vg/lvol1**

## Command Log

=====

Seq LogType Context ObjType ObjName ObjGrp Msg Errno RetCode

1 status processing lv lvol1 vg success 0 1

2 status processing vg vg success 0 1

Para más información sobre la configuración de los informes de LVM y los registros de comandos, consulte la página de manual **lvreport**.

# CAPÍTULO 10. CONFIGURACIÓN DE VOLÚMENES LÓGICOS RAID

Puede crear, activar, modificar, eliminar, visualizar y utilizar volúmenes RAID LVM.

## 10.1. VOLÚMENES LÓGICOS RAID

LVM soporta los niveles RAID 0, 1, 4, 5, 6 y 10.

Un volumen RAID LVM tiene las siguientes características:

- Los volúmenes lógicos RAID creados y gestionados por LVM aprovechan los controladores del kernel Multiple Devices (MD).
- Puede dividir temporalmente las imágenes RAID1 de la matriz y volver a fusionarlas en la matriz más tarde.
- Los volúmenes RAID de LVM admiten instantáneas.

### Agrupaciones

Los volúmenes lógicos RAID no son conscientes del cluster.

Aunque puede crear y activar volúmenes lógicos RAID exclusivamente en una máquina, no puede activarlos simultáneamente en más de una máquina.

### Subvolúmenes

Cuando se crea un volumen lógico RAID, LVM crea un subvolumen de metadatos de una extensión por cada subvolumen de datos o de paridad del array.

Por ejemplo, la creación de una matriz RAID1 de 2 vías da como resultado dos subvolúmenes de metadatos (**lv\_rmeta\_0** y **lv\_rmeta\_1**) y dos subvolúmenes de datos (**lv\_rimage\_0** y **lv\_rimage\_1**). Del mismo modo, la creación de una franja de 3 vías (más un dispositivo de paridad implícito) RAID4 da como resultado 4 subvolúmenes de metadatos (**lv\_rmeta\_0**, **lv\_rmeta\_1**, **lv\_rmeta\_2**, y **lv\_rmeta\_3**) y 4 subvolúmenes de datos (**lv\_rimage\_0**, **lv\_rimage\_1**, **lv\_rimage\_2**, y **lv\_rimage\_3**).

### Integridad

Se pueden perder datos cuando falla un dispositivo RAID o cuando se produce una corrupción blanda. La corrupción blanda en el almacenamiento de datos implica que los datos recuperados de un dispositivo de almacenamiento son diferentes de los datos escritos en ese dispositivo. Añadir integridad a un RAID LV ayuda a mitigar o prevenir la corrupción blanda. Para saber más sobre la corrupción blanda y cómo añadir integridad a un RAID LV, consulte [Sección 10.6, "Uso de la integridad de DM con RAID LV"](#).

## 10.2. NIVELES RAID Y SOPORTE LINEAL

RAID soporta varias configuraciones, incluyendo los niveles 0, 1, 4, 5, 6, 10 y lineal. Estos tipos de RAID se definen como sigue:

### Nivel 0

El nivel 0 de RAID, a menudo llamado *striping*, es una técnica de asignación de datos en franjas orientada al rendimiento. Esto significa que los datos que se escriben en la matriz se dividen en franjas y se escriben en los discos miembros de la matriz, lo que permite un alto rendimiento de E/S

con un bajo coste inherente, pero no proporciona redundancia.

Muchas implementaciones de RAID de nivel 0 sólo dividen los datos entre los dispositivos miembros hasta el tamaño del dispositivo más pequeño de la matriz. Esto significa que si tiene varios dispositivos con tamaños ligeramente diferentes, cada dispositivo se trata como si tuviera el mismo tamaño que la unidad más pequeña. Por lo tanto, la capacidad de almacenamiento común de un array de nivel 0 es igual a la capacidad del disco miembro más pequeño en un RAID de hardware o la capacidad de la partición miembro más pequeña en un RAID de software multiplicada por el número de discos o particiones del array.

### Nivel 1

El nivel 1 de RAID, o *mirroring*, proporciona redundancia escribiendo datos idénticos en cada disco miembro de la matriz, dejando una copia "en espejo" en cada disco. El mirroring sigue siendo popular debido a su simplicidad y a su alto nivel de disponibilidad de datos. El nivel 1 funciona con dos o más discos y proporciona una muy buena fiabilidad de los datos y mejora el rendimiento de las aplicaciones de lectura intensiva, pero a un coste relativamente alto.

El nivel RAID 1 tiene un coste elevado porque escribe la misma información en todos los discos de la matriz, lo que proporciona fiabilidad a los datos, pero de una forma mucho menos eficiente en cuanto a espacio que los niveles RAID basados en paridad, como el nivel 5. Sin embargo, esta ineficiencia de espacio viene acompañada de un beneficio de rendimiento: los niveles RAID basados en la paridad consumen considerablemente más potencia de la CPU para generar la paridad, mientras que el nivel RAID 1 simplemente escribe los mismos datos más de una vez en los múltiples miembros del RAID con muy poca sobrecarga de la CPU. Por ello, el nivel RAID 1 puede superar a los niveles RAID basados en la paridad en máquinas en las que se emplea el software RAID y los recursos de la CPU de la máquina se ven sometidos a una carga constante de operaciones distintas de las actividades RAID.

La capacidad de almacenamiento de la matriz de nivel 1 es igual a la capacidad del disco duro duplicado más pequeño en un RAID de hardware o de la partición duplicada más pequeña en un RAID de software. La redundancia de nivel 1 es la más alta posible entre todos los tipos de RAID, ya que la matriz puede funcionar con un solo disco presente.

### Nivel 4

El nivel 4 utiliza la paridad concentrada en una sola unidad de disco para proteger los datos. La información de paridad se calcula basándose en el contenido del resto de los discos miembros de la matriz. Esta información puede utilizarse para reconstruir los datos cuando falla un disco de la matriz. Los datos reconstruidos pueden utilizarse entonces para satisfacer las solicitudes de E/S al disco que ha fallado antes de que sea sustituido y para repoblar el disco que ha fallado después de que haya sido sustituido.

Debido a que el disco de paridad dedicado representa un cuello de botella inherente en todas las transacciones de escritura en la matriz RAID, el nivel 4 rara vez se utiliza sin tecnologías de acompañamiento como el almacenamiento en caché de escritura, o en circunstancias específicas en las que el administrador del sistema está diseñando intencionadamente el dispositivo RAID por software teniendo en cuenta este cuello de botella (como una matriz que tendrá pocas o ninguna transacción de escritura una vez que la matriz esté llena de datos). El nivel 4 de RAID se utiliza tan raramente que no está disponible como opción en Anaconda. Sin embargo, puede ser creado manualmente por el usuario si es realmente necesario.

La capacidad de almacenamiento del hardware RAID de nivel 4 es igual a la capacidad de la partición miembro más pequeña multiplicada por el número de particiones *minus one*. El rendimiento de una matriz RAID de nivel 4 es siempre asimétrico, lo que significa que las lecturas superan a las escrituras. Esto se debe a que las escrituras consumen un ancho de banda extra de la CPU y de la memoria principal cuando se genera la paridad, y también consumen un ancho de banda extra del bus cuando se escriben los datos reales en los discos, porque se están escribiendo no sólo los datos, sino también la paridad. Las lecturas sólo necesitan leer los datos y no la paridad, a menos que la matriz esté en un

estado degradado. Como resultado, las lecturas generan menos tráfico en los discos y en los buses del ordenador para la misma cantidad de transferencia de datos en condiciones normales de funcionamiento.

### Nivel 5

Este es el tipo más común de RAID. Al distribuir la paridad entre todas las unidades de disco miembros de una matriz, el nivel RAID 5 elimina el cuello de botella de escritura inherente al nivel 4. El único cuello de botella en el rendimiento es el propio proceso de cálculo de la paridad. Con las CPUs modernas y el RAID por software, esto no suele ser un cuello de botella en absoluto, ya que las CPUs modernas pueden generar la paridad muy rápidamente. Sin embargo, si tiene un número suficientemente grande de dispositivos miembros en una matriz RAID 5 por software, de forma que la velocidad de transferencia de datos combinada entre todos los dispositivos sea lo suficientemente alta, entonces este cuello de botella puede empezar a entrar en juego.

Al igual que el nivel 4, el nivel 5 tiene un rendimiento asimétrico, y las lecturas superan sustancialmente a las escrituras. La capacidad de almacenamiento del nivel RAID 5 se calcula de la misma manera que la del nivel 4.

### Nivel 6

Este es un nivel común de RAID cuando la redundancia y la preservación de los datos, y no el rendimiento, son las principales preocupaciones, pero cuando la ineficiencia de espacio del nivel 1 no es aceptable. El nivel 6 utiliza un complejo esquema de paridad para poder recuperarse de la pérdida de dos unidades cualesquiera de la matriz. Este complejo esquema de paridad crea una carga de CPU significativamente mayor en los dispositivos RAID por software y también impone una mayor carga durante las transacciones de escritura. Como tal, el nivel 6 es considerablemente más asimétrico en rendimiento que los niveles 4 y 5.

La capacidad total de una matriz RAID de nivel 6 se calcula de forma similar a la de las matrices RAID de nivel 5 y 4, con la salvedad de que hay que restar 2 dispositivos (en lugar de 1) del recuento de dispositivos para el espacio de almacenamiento de paridad adicional.

### Nivel 10

Este nivel RAID intenta combinar las ventajas de rendimiento del nivel 0 con la redundancia del nivel 1. También ayuda a aliviar parte del espacio desperdiciado en las matrices de nivel 1 con más de 2 dispositivos. Con el nivel 10, es posible, por ejemplo, crear una matriz de 3 unidades configurada para almacenar sólo 2 copias de cada dato, lo que permite que el tamaño total de la matriz sea 1,5 veces el tamaño de los dispositivos más pequeños, en lugar de ser sólo igual al dispositivo más pequeño (como ocurriría con una matriz de 3 dispositivos de nivel 1). Esto evita el uso de procesos de la CPU para calcular la paridad como en el nivel 6 de RAID, pero es menos eficiente en cuanto a espacio.

La creación de un RAID de nivel 10 no es posible durante la instalación. Es posible crear uno manualmente después de la instalación.

### RAID lineal

El RAID lineal es una agrupación de unidades para crear una unidad virtual más grande.

En el RAID lineal, los trozos se asignan secuencialmente desde una unidad miembro, pasando a la siguiente unidad sólo cuando la primera está completamente llena. Esta agrupación no proporciona ninguna ventaja de rendimiento, ya que es poco probable que las operaciones de E/S se dividan entre las unidades miembro. El RAID lineal tampoco ofrece redundancia y disminuye la fiabilidad. Si una de las unidades miembro falla, no se puede utilizar toda la matriz. La capacidad es el total de todos los discos miembros.

## 10.3. TIPOS DE SEGMENTOS RAID DE LVM

Para crear un volumen lógico RAID, se especifica un tipo de raid como argumento **--type** del comando **lvcreate**. La siguiente tabla describe los posibles tipos de segmento RAID.

Para la mayoría de los usuarios, especificar uno de los cinco tipos primarios disponibles (**raid1**, **raid4**, **raid5**, **raid6**, **raid10**) debería ser suficiente.

Tabla 10.1. Tipos de segmentos RAID de LVM

Tipo de segmento	Descripción
<b>raid1</b>	Espejo RAID1. Este es el valor por defecto para el argumento <b>--type</b> del comando <b>lvcreate</b> cuando se especifica el <b>-m</b> pero no se especifica el striping.
<b>raid4</b>	Disco de paridad dedicado RAID4
<b>raid5</b>	Igual que <b>raid5_ls</b>
<b>raid5_la</b>	<ul style="list-style-type: none"> <li>● RAID5 asimétrico izquierdo.</li> <li>● Rotación de la paridad 0 con continuación de datos</li> </ul>
<b>raid5_ra</b>	<ul style="list-style-type: none"> <li>● RAID5 asimétrico derecho.</li> <li>● Paridad rotativa N con continuación de datos</li> </ul>
<b>raid5_ls</b>	<ul style="list-style-type: none"> <li>● RAID5 simétrico izquierdo.</li> <li>● Rotación de la paridad 0 con reinicio de datos</li> </ul>
<b>raid5_rs</b>	<ul style="list-style-type: none"> <li>● RAID5 simétrico derecho.</li> <li>● Paridad rotativa N con reinicio de datos</li> </ul>
<b>raid6</b>	Igual que <b>raid6_zr</b>
<b>raid6_zr</b>	<ul style="list-style-type: none"> <li>● Reinicio cero de RAID6</li> <li>● Rotación de la paridad cero (de izquierda a derecha) con reinicio de datos</li> </ul>

Tipo de segmento	Descripción
<b>raid6_nr</b>	<ul style="list-style-type: none"> <li>● Reinicio del RAID6 N</li> <li>● Rotación de la paridad N (de izquierda a derecha) con reinicio de datos</li> </ul>
<b>raid6_nc</b>	<ul style="list-style-type: none"> <li>● RAID6 N continuar</li> <li>● Rotación de la paridad N (de izquierda a derecha) con continuación de datos</li> </ul>
<b>raid10</b>	<ul style="list-style-type: none"> <li>● Espejos a rayas. Este es el valor por defecto del argumento <b>--type</b> del comando <b>lvcreate</b> si se especifica el <b>-m</b> y se especifica un número de franjas mayor que 1.</li> <li>● Rayado de conjuntos de espejos</li> </ul>
<b>raid0/raid0_meta</b>	Striping. El RAID 0 reparte los datos del volumen lógico entre varios subvolúmenes de datos en unidades de tamaño de franja. Esto se utiliza para aumentar el rendimiento. Los datos del volumen lógico se perderán si alguno de los subvolúmenes de datos falla.

## 10.4. CREACIÓN DE VOLÚMENES LÓGICOS RAID

Esta sección proporciona ejemplos de comandos que crean diferentes tipos de volumen lógico RAID.

Puedes crear matrices RAID1 con diferentes números de copias según el valor que especifiques para el argumento **-m**. Del mismo modo, puedes especificar el número de franjas para un volumen lógico RAID 4/5/6 con el argumento **-i argument**. También puedes especificar el tamaño de las franjas con el argumento **-l**.

El siguiente comando crea una matriz RAID1 de 2 vías llamada **my\_lv** en el grupo de volúmenes **my\_vg** que tiene un tamaño de un gigabyte.

```
# lvcreate --type raid1 -m 1 -L 1G -n my_lv my_vg
```

El siguiente comando crea una matriz RAID5 (3 rayas 1 unidad de paridad implícita) llamada **my\_lv** en el grupo de volumen **my\_vg** que tiene un tamaño de un gigabyte. Tenga en cuenta que debe especificar el número de franjas tal y como lo hace para un volumen rayado LVM; el número correcto de unidades de paridad se añade automáticamente.

```
# lvcreate --type raid5 -i 3 -L 1G -n my_lv my_vg
```

El siguiente comando crea una matriz RAID6 (3 rayas 2 unidades de paridad implícita) llamada **my\_lv** en el grupo de volumen **my\_vg** que tiene un tamaño de un gigabyte.

```
# lvcreate --type raid6 -i 3 -L 1G -n my_lv my_vg
```

## 10.5. CREACIÓN DE UN VOLUMEN LÓGICO RAID0 (RAYADO)

Un volumen lógico RAID0 reparte los datos del volumen lógico entre varios subvolúmenes de datos en unidades de tamaño de franja.

El formato del comando para crear un volumen RAID0 es el siguiente.

```
lvcreate --type raid0[_meta] --stripes Stripes --stripesize StripeSize VolumeGroup
[PhysicalVolumePath...]
```

Tabla 10.2. Parámetros de creación del comando RAID0

Parámetro	Descripción
<b>--type raid0[_meta]</b>	Especificando <b>raid0</b> se crea un volumen RAID0 sin volúmenes de metadatos. Especificando <b>raid0_meta</b> se crea un volumen RAID0 con volúmenes de metadatos. Como el RAID 0 no es resistente, no tiene que almacenar ningún bloque de datos en espejo como el RAID1/10 ni calcular y almacenar ningún bloque de paridad como el RAID4/5/6. Por lo tanto, no necesita volúmenes de metadatos para mantener el estado sobre el progreso de la resincronización de los bloques en espejo o de paridad. Sin embargo, los volúmenes de metadatos se vuelven obligatorios en una conversión de RAID0 a RAID4/5/6/10, y al especificar <b>raid0_meta</b> se preasignan esos volúmenes de metadatos para evitar un fallo de asignación respectivo.
<b>--stripes <i>Stripes</i></b>	Especifica el número de dispositivos en los que se repartirá el volumen lógico.
<b>--stripesize <i>StripeSize</i></b>	Especifica el tamaño de cada franja en kilobytes. Es la cantidad de datos que se escribe en un dispositivo antes de pasar al siguiente.
<b><i>VolumeGroup</i></b>	Especifica el grupo de volúmenes a utilizar.
<b><i>PhysicalVolumePath ...</i></b>	Especifica los dispositivos a utilizar. Si no se especifica, LVM elegirá el número de dispositivos especificados por la opción <i>Stripes</i> , uno para cada franja.

Este procedimiento de ejemplo crea un volumen lógico RAID0 de LVM llamado **mylv** que separa los datos en los discos en **/dev/sda1**, **/dev/sdb1**, y **/dev/sdc1**.

1. Etiquete los discos que utilizará en el grupo de volúmenes como volúmenes físicos LVM con el comando **pvcreate**.

**AVISO**

Este comando destruye cualquier dato en **/dev/sda1**, **/dev/sdb1**, y **/dev/sdc1**.

```
# 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
```

2. Crear el grupo de volumen **myvg**. El siguiente comando crea el grupo de volúmenes **myvg**.

```
# vgcreate myvg /dev/sda1 /dev/sdb1 /dev/sdc1
Volume group "myvg" successfully created
```

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

```
# vgs
VG #PV #LV #SN Attr VSize VFree
myvg 3 0 0 wz--n- 51.45G 51.45G
```

3. Crea un volumen lógico RAID0 a partir del grupo de volúmenes que has creado. El siguiente comando crea el volumen RAID0 **mylv** a partir del grupo de volúmenes **myvg**. Este ejemplo crea un volumen lógico que tiene un tamaño de 2 gigabytes, con tres franjas y un tamaño de franja de 4 kilobytes.

```
# lvcreate --type raid0 -L 2G --stripes 3 --stripesize 4 -n mylv myvg
Rounding size 2.00 GiB (512 extents) up to stripe boundary size 2.00 GiB(513 extents).
Logical volume "mylv" created.
```

4. Cree un sistema de archivos en el volumen lógico RAID0. El siguiente comando crea un sistema de archivos **ext4** en el volumen lógico.

```
# mkfs.ext4 /dev/myvg/mylv
mke2fs 1.44.3 (10-July-2018)
Creating filesystem with 525312 4k blocks and 131376 inodes
Filesystem UUID: 9d4c0704-6028-450a-8b0a-8875358c0511
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

Los siguientes comandos montan el volumen lógico e informan del uso del espacio en disco del sistema de archivos.

```
# mount /dev/myvg/mylv /mnt
```

```
# df
Filesystem          1K-blocks  Used Available Use% Mounted on
/dev/mapper/myvg-mylv 2002684   6168  1875072  1% /mnt
```

## 10.6. USO DE LA INTEGRIDAD DE DM CON RAID LV

Mientras que el RAID ayuda a prevenir la pérdida de datos cuando un dispositivo falla, el uso de la integridad del mapeador de dispositivos (DM) con un RAID LV reduce el riesgo de pérdida de datos cuando los datos de un dispositivo RAID se corrompen. Lea esta sección para saber más sobre cómo puede proteger sus datos de la corrupción suave con la integridad DM.

### 10.6.1. Proteger los datos con la integridad del DM

Según el tipo de configuración, un RAID LV evita la pérdida de datos cuando falla un dispositivo. Si un dispositivo que forma parte de una matriz RAID falla, los datos pueden recuperarse de otros dispositivos que forman parte de ese RAID LV. Sin embargo, una configuración RAID no asegura la integridad de los datos en sí. Corrupción blanda, corrupción silenciosa, errores blandos y errores silenciosos son términos que describen los datos que se han corrompido, aunque el diseño y el software del sistema sigan funcionando como se esperaba.

La corrupción blanda en el almacenamiento de datos implica que los datos recuperados de un dispositivo de almacenamiento son diferentes de los datos escritos en ese dispositivo. Los datos corruptos pueden existir indefinidamente en los dispositivos de almacenamiento. Es posible que no descubra estos datos corruptos hasta que los recupere e intente utilizarlos.

La integridad se utiliza con los niveles RAID 1, 4, 5, 6 y 10 para ayudar a mitigar o prevenir la pérdida de datos debido a la corrupción blanda. La capa RAID garantiza que una copia no corrupta de los datos pueda reparar los errores de corrupción blanda. La capa de integridad se sitúa por encima de cada imagen RAID, mientras que un sub LV adicional almacena los metadatos de integridad (sumas de comprobación de datos) para cada imagen RAID. Cuando se recuperan datos de un RAID LV con integridad, las sumas de comprobación de los datos de integridad analizan los datos en busca de corrupción. Si se detecta corrupción, la capa de integridad devuelve un mensaje de error y la capa RAID recupera una copia no corrupta de los datos de otra imagen RAID. La capa RAID reescribe automáticamente los datos no corruptos sobre los datos corruptos para reparar la corrupción blanda.

Puede añadir la integridad de DM a un RAID LV cuando lo crea, o puede añadir la integridad de DM a un RAID LV que ya existe. Cuando se crea un RAID LV con integridad, o se añade integridad a uno existente, se requiere espacio de almacenamiento adicional para los metadatos de integridad. Para cada imagen RAID, cada 500 MB de datos requiere 4 MB de espacio de almacenamiento adicional para guardar los metadatos de integridad.

#### 10.6.1.1. Consideraciones al añadir la integridad del DM

Cuando se crea un nuevo RAID LV con integridad DM o se añade integridad a un RAID LV existente, se aplican ciertas consideraciones:

- La integridad DM requiere espacio de almacenamiento adicional porque añade sumas de comprobación a los datos.
- Aunque algunas configuraciones RAID se ven más afectadas que otras, añadir la integridad de la DM afecta al rendimiento debido a la latencia al acceder a los datos. Una configuración RAID1 suele ofrecer mejor rendimiento que RAID5 o sus variantes.
- El tamaño del bloque de integridad del RAID también afecta al rendimiento. Configurar un tamaño de bloque de integridad RAID mayor ofrece un mejor rendimiento. Sin embargo, un

tamaño de bloque de integridad RAID menor ofrece una mayor compatibilidad con versiones anteriores.

- Hay dos modos de integridad disponibles: mapa de bits o diario. El modo de integridad de mapa de bits suele ofrecer un mejor rendimiento que el modo de diario.

## SUGERENCIA

Si experimenta problemas de rendimiento, le recomendamos que utilice RAID1 con integridad o que pruebe el rendimiento de una configuración RAID concreta para asegurarse de que cumple sus requisitos.

### 10.6.2. Creación de un RAID LV con integridad DM

Cuando se crea un RAID LV, la adición de la integridad DM ayuda a mitigar el riesgo de perder datos debido a la corrupción suave.

#### Requisitos previos

- Debes tener acceso a la raíz.

#### Procedimiento

- Para crear un RAID LV con integridad DM, ejecute el siguiente comando:  
**# lvcreate --type <raid-level> --raidintegrity y -L <usable-size> -n <logical-volume> <volume-group>**

donde

#### **<raid-level>**

Especifica el nivel RAID del RAID LV que se desea crear.

#### **<usable-size>**

Especifica el tamaño utilizable en MB.

#### **<logical-volume>**

Especifica el nombre del BT que quieres crear.

#### **<volume-group>**

Especifica el nombre del grupo de volúmenes bajo el que quieres crear el RAID LV.

En el siguiente ejemplo, creamos un RAID LV con integridad llamado **test-lv** en el grupo de volúmenes **test-vg**, con un tamaño utilizable de 256M y nivel RAID 1.

#### Ejemplo de RAID LV con integridad

```
# lvcreate --type raid1 --raidintegrity y -L256M -n test-lv test-vg
Creating integrity metadata LV test-lv_rimage_0_imeta with size 8.00 MiB.
Logical volume "test-lv_rimage_0_imeta" created.
Creating integrity metadata LV test-lv_rimage_1_imeta with size 8.00 MiB.
Logical volume "test-lv_rimage_1_imeta" created.
Logical volume "test-lv" created.
```

### 10.6.3. Añadir integridad DM a un RAID LV existente

Puede añadir la integridad DM a un RAID LV existente para ayudar a mitigar el riesgo de perder datos debido a la corrupción suave.

### Requisitos previos

- Debes tener acceso a la raíz.

### Procedimiento

- Para añadir la integridad DM a un RAID LV existente, ejecute el siguiente comando:  
**# lvconvert --raidintegrity y <volume-group>/<logical-volume>**

donde

**<volume-group>**

Especifica el nombre del grupo de volúmenes bajo el que quieres crear el RAID LV.

**<logical-volume>**

Especifica el nombre del BT que quieres crear.

## 10.6.4. Eliminación de la integridad de un RAID LV

La adición de integridad a un RAID LV limita el número de operaciones que puede realizar en ese RAID LV. Por lo tanto, debe eliminar la integridad antes de realizar determinadas operaciones.

### Requisitos previos

- Debes tener acceso a la raíz.

### Procedimiento

Para eliminar la integridad de un RAID LV, ejecute el siguiente comando:

- **# lvconvert --raidintegrity n <volume-group>/<logical-volume>**

donde

**<volume-group>**

Especifica el nombre del grupo de volúmenes bajo el que quieres crear el RAID LV.

**<logical-volume>**

Especifica el nombre del BT que quieres crear.

## 10.6.5. Visualización de la información sobre la integridad del DM

Cuando cree un RAID LV con integridad o cuando añada integridad a un RAID LV existente, utilice el siguiente comando para ver información sobre la integridad:

**# lvs -a <volume-group>**

donde *<volume-group>* es el nombre del grupo de volúmenes que contiene el RAID LV con integridad.

El siguiente ejemplo muestra información sobre el LV RAID **test-lv** que fue creado en el grupo de volúmenes **test-vg**.

```
# lvs -a test-vg
```

```

LV          VG   Attr   LSize  Origin              Cpy%Sync
test-lv     test-vg rwi-a-r--- 256.00m          2.10
[test-lv_rimage_0] test-vg gwi-a-or--- 256.00m [test-lv_rimage_0_iorig] 93.75
[test-lv_rimage_0_imeta] test-vg ewi-ao---- 8.00m
[test-lv_rimage_0_iorig] test-vg -wi-ao---- 256.00m
[test-lv_rimage_1] test-vg gwi-a-or--- 256.00m [test-lv_rimage_1_iorig] 85.94
[test-lv_rimage_1_imeta] test-vg ewi-ao---- 8.00m
[test-lv_rimage_1_iorig] test-vg -wi-ao---- 256.00m
[test-lv_rmeta_0] test-vg ewi-a-or--- 4.00m
[test-lv_rmeta_1] test-vg ewi-a-or--- 4.00m

```

## Sincronización

Cuando cree un RAID LV con integridad o añada integridad a un RAID LV existente, le recomendamos que espere a que se complete la sincronización de la integridad y los metadatos del RAID antes de utilizar el LV. De lo contrario, la inicialización en segundo plano podría afectar al rendimiento del RAID. La columna **Cpy%Sync** indica el progreso de la sincronización tanto para el RAID LV de nivel superior como para cada imagen RAID. La imagen RAID se indica en la columna LV mediante **raid\_image\_N**. Consulte la columna LV para asegurarse de que el progreso de la sincronización muestra **100%** para el RAID LV de nivel superior y para cada imagen RAID.

## Imágenes RAID con integridad

El atributo **g** en los atributos listados bajo la columna **Attr** indica que la imagen RAID está usando integridad. Las sumas de comprobación de la integridad se almacenan en el LV del RAID **\_imeta**.

Para mostrar el tipo de cada RAID LV, añada la opción **-o segtype** al comando **lvs**:

```

# lvs -a my-vg -o+segtype
LV          VG   Attr   LSize  Origin              Cpy%Sync Type
test-lv     test-vg rwi-a-r--- 256.00m          87.96 raid1
[test-lv_rimage_0] test-vg gwi-a-or--- 256.00m [test-lv_rimage_0_iorig] 100.00 integrity
[test-lv_rimage_0_imeta] test-vg ewi-ao---- 8.00m          linear
[test-lv_rimage_0_iorig] test-vg -wi-ao---- 256.00m          linear
[test-lv_rimage_1] test-vg gwi-a-or--- 256.00m [test-lv_rimage_1_iorig] 100.00 integrity
[test-lv_rimage_1_imeta] test-vg ewi-ao---- 8.00m          linear
[test-lv_rimage_1_iorig] test-vg -wi-ao---- 256.00m          linear
[test-lv_rmeta_0] test-vg ewi-a-or--- 4.00m          linear
[test-lv_rmeta_1] test-vg ewi-a-or--- 4.00m          linear

```

## Desajustes de integridad

Hay un contador incremental que cuenta el número de desajustes detectados en cada imagen RAID. Para ver los desajustes de datos detectados por la integridad en una imagen RAID concreta, ejecute el siguiente comando:

```
# lvs -o integritymismatches <volume-group>/<logical-volume>_raid-image_<n>
```

donde

**<volume-group>**

Especifica el nombre del grupo de volúmenes bajo el que quieres crear el RAID LV.

**<logical-volume>**

Especifica el nombre del BT que quieres crear.

**<n>**

Especifica la imagen RAID de la que desea ver la información de desajuste de integridad.

Debe ejecutar el comando para cada imagen RAID que desee ver. En el siguiente ejemplo, veremos los desajustes de datos de **rimage\_0** en **test-vg/test-lv**.

```
# lvs -o+integritymismatches test-vg/test-lv_rimage_0
LV          VG   Attr   LSize   Origin              Cpy%Sync IntegMismatches
[test-lv_rimage_0] test-vg gwi-aor--- 256.00m [test-lv_rimage_0_iorig] 100.00      0
```

Podemos ver que la integridad no ha detectado ninguna falta de coincidencia de datos y, por lo tanto, el contador **IntegMismatches** muestra cero (0).

### Desajustes de integridad en los registros de mensajes del núcleo

También puede encontrar información sobre la integridad de los datos en los registros de mensajes del núcleo, como se muestra en los siguientes ejemplos.

#### Example of dm-integrity mismatches from the kernel message logs

```
device-mapper: integrity: dm-12: Checksum failed at sector 0x24e7
```

#### Example of dm-integrity data corrections from the kernel message logs

```
md/raid1:mdX: error de lectura corregido (8 sectores en 9448 en dm-16)
```

### 10.6.6. Recursos adicionales

- Para más información sobre todas las opciones disponibles, consulte la(s) página(s) del comando **lvraid**.

## 10.7. CONTROL DE LA VELOCIDAD DE INICIALIZACIÓN DE LOS VOLÚMENES RAID

Cuando creas volúmenes lógicos RAID10, la E/S en segundo plano requerida para inicializar los volúmenes lógicos con una operación **sync** puede desplazar otras operaciones de E/S a los dispositivos LVM, como las actualizaciones de los metadatos del grupo de volúmenes, particularmente cuando estás creando muchos volúmenes lógicos RAID. Esto puede causar que las otras operaciones LVM se ralenticen.

Puedes controlar la velocidad a la que se inicializa un volumen lógico RAID implementando el estrangulamiento de la recuperación. Puede controlar la velocidad a la que se realizan las operaciones de **sync** estableciendo la velocidad de E/S mínima y máxima para dichas operaciones con las opciones **--minrecoveryrate** y **--maxrecoveryrate** del comando **lvcreate**. Estas opciones se especifican de la siguiente manera.

- **--maxrecoveryrate *Rate*[bBsSkKmMgG]**  
Establece la tasa máxima de recuperación de un volumen lógico RAID para que no desborde las operaciones de E/S nominales. El *Rate* se especifica como una cantidad por segundo para cada dispositivo de la matriz. Si no se da ningún sufijo, se asume kiB/seg/dispositivo. Establecer la tasa de recuperación a 0 significa que será ilimitada.
- **--minrecoveryrate *Rate*[bBsSkKmMgG]**  
Establece la tasa mínima de recuperación de un volumen lógico RAID para garantizar que la E/S de las operaciones de **sync** alcance un rendimiento mínimo, incluso cuando haya una E/S

nominal intensa. El *Rate* se especifica como una cantidad por segundo para cada dispositivo en el array. Si no se da ningún sufijo, se asume kiB/seg/dispositivo.

El siguiente comando crea un array RAID10 de 2 vías con 3 bandas que tiene un tamaño de 10 gigabytes con una tasa de recuperación máxima de 128 kiB/seg/dispositivo. El array se llama **my\_lv** y está en el grupo de volúmenes **my\_vg**.

```
# lvcreate --type raid10 -i 2 -m 1 -L 10G --maxrecoveryrate 128 -n my_lv my_vg
```

También puede especificar las tasas de recuperación mínimas y máximas para una operación de depuración de RAID.

## 10.8. CONVERTIR UN DISPOSITIVO LINEAL EN UN DISPOSITIVO RAID

Puede convertir un volumen lógico lineal existente en un dispositivo RAID utilizando el argumento **--type** del comando **lvconvert**.

El siguiente comando convierte el volumen lógico lineal **my\_lv** en el grupo de volumen **my\_vg** en un array RAID1 de dos vías.

```
# lvconvert --type raid1 -m 1 my_vg/my_lv
```

Dado que los volúmenes lógicos RAID se componen de pares de subvolúmenes de metadatos y datos, cuando se convierte un dispositivo lineal en una matriz RAID1, se crea un nuevo subvolumen de metadatos y se asocia con el volumen lógico original en (uno de) los mismos volúmenes físicos en los que se encuentra el volumen lineal. Las imágenes adicionales se añaden en pares de subvolúmenes de metadatos/datos. Por ejemplo, si el dispositivo original es el siguiente

```
# lvs -a -o name,copy_percent,devices my_vg
LV      Copy%  Devices
my_lv   /dev/sde1(0)
```

Tras la conversión a una matriz RAID1 de 2 vías, el dispositivo contiene los siguientes pares de subvolúmenes de datos y metadatos:

```
# lvconvert --type raid1 -m 1 my_vg/my_lv
# lvs -a -o name,copy_percent,devices my_vg
LV      Copy%  Devices
my_lv   6.25  my_lv_rimage_0(0),my_lv_rimage_1(0)
[my_lv_rimage_0] /dev/sde1(0)
[my_lv_rimage_1] /dev/sdf1(1)
[my_lv_rmeta_0]  /dev/sde1(256)
[my_lv_rmeta_1]  /dev/sdf1(0)
```

Si la imagen de metadatos que se empareja con el volumen lógico original no puede colocarse en el mismo volumen físico, el **lvconvert** fallará.

## 10.9. CONVERTIR UN VOLUMEN LÓGICO RAID1 DE LVM EN UN VOLUMEN LÓGICO LINEAL DE LVM

Puede convertir un volumen lógico RAID1 LVM existente en un volumen lógico lineal LVM con el comando **lvconvert** especificando el argumento **-m0**. Esto elimina todos los subvolúmenes de datos RAID y todos los subvolúmenes de metadatos RAID que componen el conjunto RAID, dejando la imagen

RAID1 de nivel superior como volumen lógico lineal.

El siguiente ejemplo muestra un volumen lógico LVM RAID1 existente.

```
# lvs -a -o name,copy_percent,devices my_vg
LV          Copy%  Devices
my_lv       100.00  my_lv_rimage_0(0),my_lv_rimage_1(0)
[my_lv_rimage_0]  /dev/sde1(1)
[my_lv_rimage_1]  /dev/sdf1(1)
[my_lv_rmeta_0]   /dev/sde1(0)
[my_lv_rmeta_1]   /dev/sdf1(0)
```

El siguiente comando convierte el volumen lógico RAID1 de LVM **my\_vg/my\_lv** en un dispositivo lineal de LVM.

```
# lvconvert -m0 my_vg/my_lv
# lvs -a -o name,copy_percent,devices my_vg
LV  Copy%  Devices
my_lv  /dev/sde1(1)
```

Cuando se convierte un volumen lógico RAID1 LVM en un volumen lineal LVM, se puede especificar qué volúmenes físicos eliminar. El siguiente ejemplo muestra la disposición de un volumen lógico RAID1 LVM compuesto por dos imágenes: **/dev/sda1** y **/dev/sdb1**. En este ejemplo, el comando **lvconvert** especifica que se quiere eliminar **/dev/sda1**, dejando **/dev/sdb1** como el volumen físico que compone el dispositivo lineal.

```
# lvs -a -o name,copy_percent,devices my_vg
LV          Copy%  Devices
my_lv       100.00  my_lv_rimage_0(0),my_lv_rimage_1(0)
[my_lv_rimage_0]  /dev/sda1(1)
[my_lv_rimage_1]  /dev/sdb1(1)
[my_lv_rmeta_0]   /dev/sda1(0)
[my_lv_rmeta_1]   /dev/sdb1(0)
# lvconvert -m0 my_vg/my_lv /dev/sda1
# lvs -a -o name,copy_percent,devices my_vg
LV  Copy%  Devices
my_lv  /dev/sdb1(1)
```

## 10.10. CONVERTIR UN DISPOSITIVO LVM EN ESPEJO EN UN DISPOSITIVO RAID1

Puede convertir un dispositivo LVM reflejado existente con un tipo de segmento de **mirror** a un dispositivo LVM RAID1 con el comando **lvconvert** especificando el argumento **--type raid1**. Esto cambia el nombre de los subvolúmenes en espejo (**mimage**) a subvolúmenes RAID (**rimage**). Además, se elimina el registro de réplica y se crean subvolúmenes de metadatos (**rmeta**) se crean para los subvolúmenes de datos en los mismos volúmenes físicos que los correspondientes subvolúmenes de datos.

El siguiente ejemplo muestra la disposición de un volumen lógico en espejo **my\_vg/my\_lv**.

```
# lvs -a -o name,copy_percent,devices my_vg
LV          Copy%  Devices
my_lv       15.20  my_lv_mimage_0(0),my_lv_mimage_1(0)
```

```
[my_lv_mimage_0]    /dev/sde1(0)
[my_lv_mimage_1]    /dev/sdf1(0)
[my_lv_mlog]        /dev/sdd1(0)
```

El siguiente comando convierte el volumen lógico en espejo **my\_vg/my\_lv** en un volumen lógico RAID1.

```
# lvconvert --type raid1 my_vg/my_lv
# lvs -a -o name,copy_percent,devices my_vg
LV          Copy%  Devices
my_lv       100.00 my_lv_rimage_0(0),my_lv_rimage_1(0)
[my_lv_rimage_0]    /dev/sde1(0)
[my_lv_rimage_1]    /dev/sdf1(0)
[my_lv_rmeta_0]     /dev/sde1(125)
[my_lv_rmeta_1]     /dev/sdf1(125)
```

## 10.11. REDIMENSIONAR UN VOLUMEN LÓGICO RAID

Puede redimensionar un volumen lógico RAID de las siguientes maneras;

- Puede aumentar el tamaño de un volumen lógico RAID de cualquier tipo con el comando **lvresize** o **lvextend**. Esto no cambia el número de imágenes RAID. Para los volúmenes lógicos RAID rayados se aplican las mismas restricciones de redondeo de franjas que cuando se crea un volumen lógico RAID rayado.
- Puede reducir el tamaño de un volumen lógico RAID de cualquier tipo con el comando **lvresize** o **lvreduce**. Esto no cambia el número de imágenes RAID. Al igual que con el comando **lvextend**, se aplican las mismas restricciones de redondeo de franjas que cuando se crea un volumen lógico RAID en franjas.
- Puede cambiar el número de franjas de un volumen lógico RAID rayado (**raid4/5/6/10**) con el parámetro **--stripes N** del comando **lvconvert**. Esto aumenta o reduce el tamaño del volumen lógico RAID por la capacidad de las franjas añadidas o eliminadas. Tenga en cuenta que los volúmenes de **raid10** sólo pueden añadir franjas. Esta capacidad es parte de la función RAID *reshaping* que permite cambiar los atributos de un volumen lógico RAID manteniendo el mismo nivel RAID. Para obtener información sobre la remodelación de RAID y ejemplos de uso del comando **lvconvert** para remodelar un volumen lógico RAID, consulte la página de manual **lvraid(7)**.

## 10.12. CAMBIO DEL NÚMERO DE IMÁGENES EN UN DISPOSITIVO RAID1 EXISTENTE

Puede cambiar el número de imágenes en una matriz RAID1 existente, al igual que puede cambiar el número de imágenes en la implementación anterior de la duplicación LVM. Utilice el comando **lvconvert** para especificar el número de pares de subvolúmenes de metadatos/datos adicionales que se van a añadir o eliminar.

Cuando añada imágenes a un dispositivo RAID1 con el comando **lvconvert**, puede especificar el número total de imágenes para el dispositivo resultante, o puede especificar cuántas imágenes añadir al dispositivo. También puede especificar opcionalmente en qué volúmenes físicos residirán los nuevos pares de imágenes de metadatos/datos.

Los subvolúmenes de metadatos (denominados **rmeta**) siempre existen en los mismos dispositivos físicos que sus homólogos de subvolumen de datos (**rimage**). Los pares de subvolúmenes de metadatos/datos no se crearán en los mismos volúmenes físicos que los de otro par de subvolúmenes

de metadatos/datos en la matriz RAID (a menos que especifique **--alloc anywhere**).

El formato del comando para añadir imágenes a un volumen RAID1 es el siguiente:

```
lvconvert -m new_absolute_count vg/lv [removable_PVs]  
lvconvert -m +num_additional_images vg/lv [removable_PVs]
```

Por ejemplo, el siguiente comando muestra el dispositivo LVM **my\_vg/my\_lv**, que es una matriz RAID1 de 2 vías:

```
# lvs -a -o name,copy_percent,devices my_vg  
LV          Copy%  Devices  
my_lv       6.25  my_lv_rimage_0(0),my_lv_rimage_1(0)  
[my_lv_rimage_0]  /dev/sde1(0)  
[my_lv_rimage_1]  /dev/sdf1(1)  
[my_lv_rmeta_0]   /dev/sde1(256)  
[my_lv_rmeta_1]   /dev/sdf1(0)
```

El siguiente comando convierte el dispositivo RAID1 de 2 vías **my\_vg/my\_lv** en un dispositivo RAID1 de 3 vías:

```
# lvconvert -m 2 my_vg/my_lv  
# lvs -a -o name,copy_percent,devices my_vg  
LV          Copy%  Devices  
my_lv       6.25  my_lv_rimage_0(0),my_lv_rimage_1(0),my_lv_rimage_2(0)  
[my_lv_rimage_0]  /dev/sde1(0)  
[my_lv_rimage_1]  /dev/sdf1(1)  
[my_lv_rimage_2]  /dev/sdg1(1)  
[my_lv_rmeta_0]   /dev/sde1(256)  
[my_lv_rmeta_1]   /dev/sdf1(0)  
[my_lv_rmeta_2]   /dev/sdg1(0)
```

Cuando añades una imagen a un array RAID1, puedes especificar qué volúmenes físicos utilizar para la imagen. El siguiente comando convierte el dispositivo RAID1 de 2 vías **my\_vg/my\_lv** en un dispositivo RAID1 de 3 vías, especificando que se utilice el volumen físico **/dev/sdd1** para el array:

```
# lvs -a -o name,copy_percent,devices my_vg  
LV          Copy%  Devices  
my_lv       56.00 my_lv_rimage_0(0),my_lv_rimage_1(0)  
[my_lv_rimage_0]  /dev/sda1(1)  
[my_lv_rimage_1]  /dev/sdb1(1)  
[my_lv_rmeta_0]   /dev/sda1(0)  
[my_lv_rmeta_1]   /dev/sdb1(0)  
# lvconvert -m 2 my_vg/my_lv /dev/sdd1  
# lvs -a -o name,copy_percent,devices my_vg  
LV          Copy%  Devices  
my_lv       28.00 my_lv_rimage_0(0),my_lv_rimage_1(0),my_lv_rimage_2(0)  
[my_lv_rimage_0]  /dev/sda1(1)  
[my_lv_rimage_1]  /dev/sdb1(1)  
[my_lv_rimage_2]  /dev/sdd1(1)  
[my_lv_rmeta_0]   /dev/sda1(0)  
[my_lv_rmeta_1]   /dev/sdb1(0)  
[my_lv_rmeta_2]   /dev/sdd1(0)
```

Para eliminar imágenes de una matriz RAID1, utilice el siguiente comando. Cuando elimine imágenes de un dispositivo RAID1 con el comando **lvconvert**, puede especificar el número total de imágenes para el dispositivo resultante, o puede especificar cuántas imágenes eliminar del dispositivo. También puede especificar opcionalmente los volúmenes físicos de los que eliminar el dispositivo.

```
lvconvert -m new_absolute_count vg/lv [removable_PVs]
lvconvert -m -num_fewer_images vg/lv [removable_PVs]
```

Además, cuando se elimina una imagen y su volumen de subvolumen de metadatos asociado, cualquier imagen con un número superior se desplazará hacia abajo para llenar el hueco. Si elimina **lv\_rimage\_1** de una matriz RAID1 de 3 vías que consiste en **lv\_rimage\_0**, **lv\_rimage\_1**, y **lv\_rimage\_2**, esto resulta en una matriz RAID1 que consiste en **lv\_rimage\_0** y **lv\_rimage\_1**. El subvolumen **lv\_rimage\_2** será renombrado y ocupará la ranura vacía, convirtiéndose en **lv\_rimage\_1**.

El siguiente ejemplo muestra la disposición de un volumen lógico RAID1 de 3 vías **my\_vg/my\_lv**.

```
# lvs -a -o name,copy_percent,devices my_vg
LV          Copy%  Devices
my_lv       100.00 my_lv_rimage_0(0),my_lv_rimage_1(0),my_lv_rimage_2(0)
[my_lv_rimage_0]  /dev/sde1(1)
[my_lv_rimage_1]  /dev/sdf1(1)
[my_lv_rimage_2]  /dev/sdg1(1)
[my_lv_rmeta_0]   /dev/sde1(0)
[my_lv_rmeta_1]   /dev/sdf1(0)
[my_lv_rmeta_2]   /dev/sdg1(0)
```

El siguiente comando convierte el volumen lógico RAID1 de 3 vías en un volumen lógico RAID1 de 2 vías.

```
# lvconvert -m1 my_vg/my_lv
# lvs -a -o name,copy_percent,devices my_vg
LV          Copy%  Devices
my_lv       100.00 my_lv_rimage_0(0),my_lv_rimage_1(0)
[my_lv_rimage_0]  /dev/sde1(1)
[my_lv_rimage_1]  /dev/sdf1(1)
[my_lv_rmeta_0]   /dev/sde1(0)
[my_lv_rmeta_1]   /dev/sdf1(0)
```

El siguiente comando convierte el volumen lógico RAID1 de 3 vías en un volumen lógico RAID1 de 2 vías, especificando el volumen físico que contiene la imagen a eliminar como **/dev/sde1**.

```
# lvconvert -m1 my_vg/my_lv /dev/sde1
# lvs -a -o name,copy_percent,devices my_vg
LV          Copy%  Devices
my_lv       100.00 my_lv_rimage_0(0),my_lv_rimage_1(0)
[my_lv_rimage_0]  /dev/sdf1(1)
[my_lv_rimage_1]  /dev/sdg1(1)
[my_lv_rmeta_0]   /dev/sdf1(0)
[my_lv_rmeta_1]   /dev/sdg1(0)
```

## 10.13. DIVIDIR UNA IMAGEN RAID COMO UN VOLUMEN LÓGICO SEPARADO

Puede dividir una imagen de un volumen lógico RAID para formar un nuevo volumen lógico.

El formato del comando para dividir una imagen RAID es el siguiente:

```
lvconvert --splitmirrors count -n splitname vg/lv [removable_PVs]
```

Al igual que cuando se elimina una imagen RAID de un volumen lógico RAID1 existente, cuando se elimina un subvolumen de datos RAID (y su subvolumen de metadatos asociado) del centro del dispositivo, cualquier imagen con un número superior se desplazará hacia abajo para llenar el hueco. Los números de índice de los volúmenes lógicos que componen una matriz RAID serán, por tanto, una secuencia ininterrumpida de números enteros.



## NOTA

No se puede dividir una imagen RAID si la matriz RAID1 aún no está sincronizada.

El siguiente ejemplo divide un volumen lógico RAID1 de dos vías, **my\_lv**, en dos volúmenes lógicos lineales, **my\_lv** y **new**.

```
# lvs -a -o name,copy_percent,devices my_vg
LV          Copy%  Devices
my_lv       12.00  my_lv_rimage_0(0),my_lv_rimage_1(0)
[my_lv_rimage_0]  /dev/sde1(1)
[my_lv_rimage_1]  /dev/sdf1(1)
[my_lv_rmeta_0]   /dev/sde1(0)
[my_lv_rmeta_1]   /dev/sdf1(0)
# lvconvert --splitmirror 1 -n new my_vg/my_lv
# lvs -a -o name,copy_percent,devices my_vg
LV  Copy%  Devices
my_lv  /dev/sde1(1)
new    /dev/sdf1(1)
```

El siguiente ejemplo divide un volumen lógico RAID1 de 3 vías, **my\_lv**, en un volumen lógico RAID1 de 2 vías, **my\_lv**, y un volumen lógico lineal, **new**

```
# lvs -a -o name,copy_percent,devices my_vg
LV          Copy%  Devices
my_lv       100.00  my_lv_rimage_0(0),my_lv_rimage_1(0),my_lv_rimage_2(0)
[my_lv_rimage_0]  /dev/sde1(1)
[my_lv_rimage_1]  /dev/sdf1(1)
[my_lv_rimage_2]  /dev/sdg1(1)
[my_lv_rmeta_0]   /dev/sde1(0)
[my_lv_rmeta_1]   /dev/sdf1(0)
[my_lv_rmeta_2]   /dev/sdg1(0)
# lvconvert --splitmirror 1 -n new my_vg/my_lv
# lvs -a -o name,copy_percent,devices my_vg
LV  Copy%  Devices
my_lv  100.00  my_lv_rimage_0(0),my_lv_rimage_1(0)
[my_lv_rimage_0]  /dev/sde1(1)
[my_lv_rimage_1]  /dev/sdf1(1)
[my_lv_rmeta_0]   /dev/sde1(0)
[my_lv_rmeta_1]   /dev/sdf1(0)
new    /dev/sdg1(1)
```

## 10.14. DIVIDIR Y FUSIONAR UNA IMAGEN RAID

Puedes dividir temporalmente una imagen de una matriz RAID1 para uso de sólo lectura mientras mantienes el seguimiento de cualquier cambio utilizando el argumento **--trackchanges** junto con el argumento **--splitmirrors** del comando **lvconvert**. Esto le permite fusionar la imagen de nuevo en la matriz en un momento posterior mientras resincroniza sólo aquellas partes de la matriz que han cambiado desde que la imagen fue dividida.

El formato del comando **lvconvert** para dividir una imagen RAID es el siguiente.

```
lvconvert --splitmirrors count --trackchanges vg/lv [removable_PVs]
```

Cuando se divide una imagen RAID con el argumento **--trackchanges**, se puede especificar qué imagen se va a dividir, pero no se puede cambiar el nombre del volumen que se está dividiendo. Además, los volúmenes resultantes tienen las siguientes restricciones.

- El nuevo volumen creado es de sólo lectura.
- No se puede cambiar el tamaño del nuevo volumen.
- No se puede cambiar el nombre de la matriz restante.
- No se puede cambiar el tamaño de la matriz restante.
- Puedes activar el nuevo volumen y el resto del array de forma independiente.

Puede fusionar una imagen que se dividió con el argumento **--trackchanges** especificado ejecutando un comando posterior **lvconvert** con el argumento **--merge**. Cuando se fusiona la imagen, sólo se resincronizan las partes de la matriz que han cambiado desde que se dividió la imagen.

El formato del comando **lvconvert** para fusionar una imagen RAID es el siguiente.

```
lvconvert --merge raid_image
```

El siguiente ejemplo crea un volumen lógico RAID1 y luego separa una imagen de ese volumen mientras rastrea los cambios en la matriz restante.

```
# lvcreate --type raid1 -m 2 -L 1G -n my_lv my_vg
Logical volume "my_lv" created
# lvs -a -o name,copy_percent,devices my_vg
LV          Copy%  Devices
my_lv       100.00 my_lv_rimage_0(0),my_lv_rimage_1(0),my_lv_rimage_2(0)
[my_lv_rimage_0]  /dev/sdb1(1)
[my_lv_rimage_1]  /dev/sdc1(1)
[my_lv_rimage_2]  /dev/sdd1(1)
[my_lv_rmeta_0]   /dev/sdb1(0)
[my_lv_rmeta_1]   /dev/sdc1(0)
[my_lv_rmeta_2]   /dev/sdd1(0)
# lvconvert --splitmirrors 1 --trackchanges my_vg/my_lv
my_lv_rimage_2 split from my_lv for read-only purposes.
Use 'lvconvert --merge my_vg/my_lv_rimage_2' to merge back into my_lv
# lvs -a -o name,copy_percent,devices my_vg
LV          Copy%  Devices
my_lv       100.00 my_lv_rimage_0(0),my_lv_rimage_1(0),my_lv_rimage_2(0)
[my_lv_rimage_0]  /dev/sdb1(1)
[my_lv_rimage_1]  /dev/sdc1(1)
my_lv_rimage_2  /dev/sdd1(1)
```

```
[my_lv_rmeta_0]    /dev/sdb1(0)
[my_lv_rmeta_1]    /dev/sdc1(0)
[my_lv_rmeta_2]    /dev/sdd1(0)
```

El siguiente ejemplo separa una imagen de un volumen RAID1 mientras rastrea los cambios en la matriz restante, y luego fusiona el volumen de nuevo en la matriz.

```
# lvconvert --splitmirrors 1 --trackchanges my_vg/my_lv
lv_rimage_1 split from my_lv for read-only purposes.
Use 'lvconvert --merge my_vg/my_lv_rimage_1' to merge back into my_lv
# lvs -a -o name,copy_percent,devices my_vg
LV          Copy%  Devices
my_lv       100.00 my_lv_rimage_0(0),my_lv_rimage_1(0)
[my_lv_rimage_0]    /dev/sdc1(1)
[my_lv_rimage_1]    /dev/sdd1(1)
[my_lv_rmeta_0]     /dev/sdc1(0)
[my_lv_rmeta_1]     /dev/sdd1(0)
# lvconvert --merge my_vg/my_lv_rimage_1
my_vg/my_lv_rimage_1 successfully merged back into my_vg/my_lv
# lvs -a -o name,copy_percent,devices my_vg
LV          Copy%  Devices
my_lv       100.00 my_lv_rimage_0(0),my_lv_rimage_1(0)
[my_lv_rimage_0]    /dev/sdc1(1)
[my_lv_rimage_1]    /dev/sdd1(1)
[my_lv_rmeta_0]     /dev/sdc1(0)
[my_lv_rmeta_1]     /dev/sdd1(0)
```

## 10.15. ESTABLECIMIENTO DE UNA POLÍTICA DE FALLOS EN EL RAID

El RAID LVM maneja los fallos de los dispositivos de forma automática basándose en las preferencias definidas por el campo **raid\_fault\_policy** en el archivo **lvm.conf**.

- Si el campo **raid\_fault\_policy** está configurado como **allocate**, el sistema intentará sustituir el dispositivo que ha fallado por un dispositivo de repuesto del grupo de volúmenes. Si no hay ningún dispositivo de repuesto disponible, se informará al registro del sistema.
- Si el campo **raid\_fault\_policy** está configurado como **warn**, el sistema producirá una advertencia y el registro indicará que un dispositivo ha fallado. Esto permite al usuario determinar el curso de acción a seguir.

Mientras queden suficientes dispositivos para soportar la usabilidad, el volumen lógico RAID seguirá funcionando.

### 10.15.1. La política de fallos de RAID asignada

En el siguiente ejemplo, el campo **raid\_fault\_policy** se ha configurado como **allocate** en el archivo **lvm.conf**. El volumen lógico RAID está dispuesto de la siguiente manera.

```
# lvs -a -o name,copy_percent,devices my_vg
LV          Copy%  Devices
my_lv       100.00 my_lv_rimage_0(0),my_lv_rimage_1(0),my_lv_rimage_2(0)
[my_lv_rimage_0]    /dev/sde1(1)
[my_lv_rimage_1]    /dev/sdf1(1)
[my_lv_rimage_2]    /dev/sdg1(1)
```

```
[my_lv_rmeta_0]    /dev/sde1(0)
[my_lv_rmeta_1]    /dev/sdf1(0)
[my_lv_rmeta_2]    /dev/sdg1(0)
```

Si el dispositivo **/dev/sde** falla, el registro del sistema mostrará mensajes de error.

```
# grep lvm /var/log/messages
```

```
Jan 17 15:57:18 bp-01 lvm[8599]: Device #0 of raid1 array, my_vg-my_lv, has failed.
Jan 17 15:57:18 bp-01 lvm[8599]: /dev/sde1: read failed after 0 of 2048 at
250994294784: Input/output error
Jan 17 15:57:18 bp-01 lvm[8599]: /dev/sde1: read failed after 0 of 2048 at
250994376704: Input/output error
Jan 17 15:57:18 bp-01 lvm[8599]: /dev/sde1: read failed after 0 of 2048 at 0:
Input/output error
Jan 17 15:57:18 bp-01 lvm[8599]: /dev/sde1: read failed after 0 of 2048 at
4096: Input/output error
Jan 17 15:57:19 bp-01 lvm[8599]: Couldn't find device with uuid
3lugiV-3eSP-AFAR-sdrP-H20O-wM2M-qdMANy.
Jan 17 15:57:27 bp-01 lvm[8599]: raid1 array, my_vg-my_lv, is not in-sync.
Jan 17 15:57:36 bp-01 lvm[8599]: raid1 array, my_vg-my_lv, is now in-sync.
```

Dado que el campo **raid\_fault\_policy** se ha establecido en **allocate**, el dispositivo que ha fallado se sustituye por un nuevo dispositivo del grupo de volúmenes.

```
# lvs -a -o name,copy_percent,devices vg
```

```
Couldn't find device with uuid 3lugiV-3eSP-AFAR-sdrP-H20O-wM2M-qdMANy.
LV          Copy%  Devices
lv          100.00 lv_rimage_0(0),lv_rimage_1(0),lv_rimage_2(0)
[lv_rimage_0]    /dev/sdh1(1)
[lv_rimage_1]    /dev/sdf1(1)
[lv_rimage_2]    /dev/sdg1(1)
[lv_rmeta_0]     /dev/sdh1(0)
[lv_rmeta_1]     /dev/sdf1(0)
[lv_rmeta_2]     /dev/sdg1(0)
```

Tenga en cuenta que aunque el dispositivo fallado haya sido reemplazado, la pantalla sigue indicando que LVM no pudo encontrar el dispositivo fallado. Esto se debe a que, aunque el dispositivo fallido ha sido eliminado del volumen lógico RAID, el dispositivo fallido aún no ha sido eliminado del grupo de volúmenes. Para eliminar el dispositivo fallido del grupo de volúmenes, puede ejecutar **vgreduce --removemissing VG**.

Si el **raid\_fault\_policy** se ha establecido en **allocate** pero no hay dispositivos de repuesto, la asignación fallará, dejando el volumen lógico como está. Si la asignación falla, tienes la opción de arreglar la unidad y luego iniciar la recuperación del dispositivo fallido con la opción **--refresh** del comando **lvchange**. Alternativamente, puede reemplazar el dispositivo fallado.

### 10.15.2. El aviso de la política de fallos del RAID

En el siguiente ejemplo, el campo **raid\_fault\_policy** se ha configurado como **warn** en el archivo **lvm.conf**. El volumen lógico RAID está dispuesto de la siguiente manera.

```
# lvs -a -o name,copy_percent,devices my_vg
LV          Copy%  Devices
my_lv       100.00 my_lv_rimage_0(0),my_lv_rimage_1(0),my_lv_rimage_2(0)
[my_lv_rimage_0]    /dev/sdh1(1)
```

```
[my_lv_rimage_1]    /dev/sdf1(1)
[my_lv_rimage_2]    /dev/sdg1(1)
[my_lv_rmeta_0]     /dev/sdh1(0)
[my_lv_rmeta_1]     /dev/sdf1(0)
[my_lv_rmeta_2]     /dev/sdg1(0)
```

Si el dispositivo **/dev/sdh** falla, el registro del sistema mostrará mensajes de error. En este caso, sin embargo, LVM no intentará reparar automáticamente el dispositivo RAID reemplazando una de las imágenes. En su lugar, si el dispositivo ha fallado, puede reemplazar el dispositivo con el argumento **--repair** del comando **lvconvert**.

## 10.16. SUSTITUCIÓN DE UN DISPOSITIVO RAID EN UN VOLUMEN LÓGICO

Puede sustituir un dispositivo RAID en un volumen lógico.

- Si no se ha producido ningún fallo en el dispositivo RAID, siga [Sección 10.16.1, “Sustitución de un dispositivo RAID que no ha fallado”](#).
- Si el dispositivo RAID ha fallado, siga [Sección 10.16.4, “Sustitución de un dispositivo RAID fallido en un volumen lógico”](#).

### 10.16.1. Sustitución de un dispositivo RAID que no ha fallado

Para sustituir un dispositivo RAID en un volumen lógico, utilice el argumento **--replace** del comando **lvconvert**.

#### Requisitos previos

- El dispositivo RAID no ha fallado. Los siguientes comandos no funcionarán si el dispositivo RAID ha fallado.

#### Procedimiento

- Sustituya el dispositivo RAID:

```
# lvconvert --replace dev_to_remove vg/lv possible_replacements
```

- Sustituya *dev\_to\_remove* con la ruta del volumen físico que desea reemplazar.
- Sustituya *vg/lv* con el grupo de volúmenes y el nombre del volumen lógico de la matriz RAID.
- Sustituya *possible\_replacements* con la ruta del volumen físico que desea utilizar como reemplazo.

#### Ejemplo 10.1. Sustitución de un dispositivo RAID1

El siguiente ejemplo crea un volumen lógico RAID1 y luego reemplaza un dispositivo en ese volumen.

1. Cree la matriz RAID1:

```
# lvcreate --type raid1 -m 2 -L 1G -n my_lv my_vg
```

```
Logical volume "my_lv" created
```

2. Examine la matriz RAID1:

```
# lvs -a -o name,copy_percent,devices my_vg
```

```
LV          Copy%  Devices
my_lv       100.00 my_lv_rimage_0(0),my_lv_rimage_1(0),my_lv_rimage_2(0)
[my_lv_rimage_0] /dev/sdb1(1)
[my_lv_rimage_1] /dev/sdb2(1)
[my_lv_rimage_2] /dev/sdc1(1)
[my_lv_rmeta_0] /dev/sdb1(0)
[my_lv_rmeta_1] /dev/sdb2(0)
[my_lv_rmeta_2] /dev/sdc1(0)
```

3. Sustituya el volumen físico **/dev/sdb2**:

```
# lvconvert --replace /dev/sdb2 my_vg/my_lv
```

4. Examine la matriz RAID1 con el reemplazo:

```
# lvs -a -o name,copy_percent,devices my_vg
```

```
LV          Copy%  Devices
my_lv       37.50 my_lv_rimage_0(0),my_lv_rimage_1(0),my_lv_rimage_2(0)
[my_lv_rimage_0] /dev/sdb1(1)
[my_lv_rimage_1] /dev/sdc2(1)
[my_lv_rimage_2] /dev/sdc1(1)
[my_lv_rmeta_0] /dev/sdb1(0)
[my_lv_rmeta_1] /dev/sdc2(0)
[my_lv_rmeta_2] /dev/sdc1(0)
```

## Ejemplo 10.2. Especificación del volumen físico de sustitución

El siguiente ejemplo crea un volumen lógico RAID1 y luego reemplaza un dispositivo en ese volumen, especificando qué volumen físico usar para el reemplazo.

1. Cree la matriz RAID1:

```
# lvcreate --type raid1 -m 1 -L 100 -n my_lv my_vg
```

```
Logical volume "my_lv" created
```

2. Examine la matriz RAID1:

```
# lvs -a -o name,copy_percent,devices my_vg
```

```
LV          Copy%  Devices
my_lv       100.00 my_lv_rimage_0(0),my_lv_rimage_1(0)
[my_lv_rimage_0] /dev/sda1(1)
```

```
[my_lv_rimage_1]    /dev/sdb1(1)
[my_lv_rmeta_0]    /dev/sda1(0)
[my_lv_rmeta_1]    /dev/sdb1(0)
```

3. Examine los volúmenes físicos:

```
# pvs

PV      VG      Fmt Attr PSize  PFree
/dev/sda1 my_vg  lvm2 a-- 1020.00m 916.00m
/dev/sdb1 my_vg  lvm2 a-- 1020.00m 916.00m
/dev/sdc1 my_vg  lvm2 a-- 1020.00m 1020.00m
/dev/sdd1 my_vg  lvm2 a-- 1020.00m 1020.00m
```

4. Sustituya el volumen físico **/dev/sdb1** por **/dev/sdd1**:

```
# lvconvert --replace /dev/sdb1 my_vg/my_lv /dev/sdd1
```

5. Examine la matriz RAID1 con el reemplazo:

```
# lvs -a -o name,copy_percent,devices my_vg

LV          Copy% Devices
my_lv       28.00 my_lv_rimage_0(0),my_lv_rimage_1(0)
[my_lv_rimage_0]    /dev/sda1(1)
[my_lv_rimage_1]    /dev/sdd1(1)
[my_lv_rmeta_0]     /dev/sda1(0)
[my_lv_rmeta_1]     /dev/sdd1(0)
```

### Ejemplo 10.3. Sustitución de varios dispositivos RAID

Puede reemplazar más de un dispositivo RAID a la vez especificando varios argumentos **replace**, como en el siguiente ejemplo.

1. Cree una matriz RAID1:

```
# lvcreate --type raid1 -m 2 -L 100 -n my_lv my_vg

Logical volume "my_lv" created
```

2. Examine la matriz RAID1:

```
# lvs -a -o name,copy_percent,devices my_vg

LV          Copy% Devices
my_lv       100.00 my_lv_rimage_0(0),my_lv_rimage_1(0),my_lv_rimage_2(0)
[my_lv_rimage_0]    /dev/sda1(1)
[my_lv_rimage_1]    /dev/sdb1(1)
[my_lv_rimage_2]    /dev/sdc1(1)
[my_lv_rmeta_0]     /dev/sda1(0)
[my_lv_rmeta_1]     /dev/sdb1(0)
[my_lv_rmeta_2]     /dev/sdc1(0)
```

3. Sustituya los volúmenes físicos **/dev/sdb1** y **/dev/sdc1**:

```
# lvconvert --replace /dev/sdb1 --replace /dev/sdc1 my_vg/my_lv
```

4. Examine la matriz RAID1 con los reemplazos:

```
# lvs -a -o name,copy_percent,devices my_vg

LV          Copy%  Devices
my_lv       60.00  my_lv_rimage_0(0),my_lv_rimage_1(0),my_lv_rimage_2(0)
[my_lv_rimage_0]  /dev/sda1(1)
[my_lv_rimage_1]  /dev/sdd1(1)
[my_lv_rimage_2]  /dev/sde1(1)
[my_lv_rmeta_0]   /dev/sda1(0)
[my_lv_rmeta_1]   /dev/sdd1(0)
[my_lv_rmeta_2]   /dev/sde1(0)
```

### 10.16.2. Dispositivos fallidos en RAID LVM

RAID no es como el mirroring LVM tradicional. El mirroring LVM requería la eliminación de los dispositivos que fallaban o el volumen lógico reflejado se colgaba. Las matrices RAID pueden seguir funcionando con dispositivos fallidos. De hecho, para los tipos de RAID que no sean RAID1, eliminar un dispositivo significaría convertirlo a un nivel inferior de RAID (por ejemplo, de RAID6 a RAID5, o de RAID4 o RAID5 a RAID0).

Por lo tanto, en lugar de eliminar un dispositivo fallado incondicionalmente y potencialmente asignar un reemplazo, LVM le permite reemplazar un dispositivo fallado en un volumen RAID en una solución de un solo paso utilizando el argumento **--repair** del comando **lvconvert**.

### 10.16.3. Recuperación de un dispositivo RAID fallido en un volumen lógico

Si el fallo del dispositivo RAID LVM es un fallo transitorio o puede reparar el dispositivo que ha fallado, puede iniciar la recuperación del dispositivo que ha fallado.

#### Requisitos previos

- El dispositivo que antes fallaba ahora funciona.

#### Procedimiento

- Actualiza el volumen lógico que contiene el dispositivo RAID:

```
# lvchange --refresh my_vg/my_lv
```

#### Pasos de verificación

- Examine el volumen lógico con el dispositivo recuperado:

```
# lvs --all --options name,devices,lv_attr,lv_health_status my_vg
```

### 10.16.4. Sustitución de un dispositivo RAID fallido en un volumen lógico

Este procedimiento reemplaza un dispositivo fallado que sirve como volumen físico en un volumen lógico RAID LVM.

### Requisitos previos

- El grupo de volúmenes incluye un volumen físico que proporciona suficiente capacidad libre para reemplazar el dispositivo que ha fallado.  
Si no hay ningún volumen físico con suficientes extensiones libres en el grupo de volúmenes, añada un nuevo volumen físico lo suficientemente grande utilizando la utilidad **vgextend**.

### Procedimiento

1. En el siguiente ejemplo, un volumen lógico RAID está dispuesto de la siguiente manera:

```
# lvs --all --options name,copy_percent,devices my_vg

LV          Cpy%Sync Devices
my_lv       100.00 my_lv_rimage_0(0),my_lv_rimage_1(0),my_lv_rimage_2(0)
[my_lv_rimage_0]    /dev/sde1(1)
[my_lv_rimage_1]    /dev/sdc1(1)
[my_lv_rimage_2]    /dev/sdd1(1)
[my_lv_rmeta_0]     /dev/sde1(0)
[my_lv_rmeta_1]     /dev/sdc1(0)
[my_lv_rmeta_2]     /dev/sdd1(0)
```

2. Si el dispositivo **/dev/sdc** falla, la salida del comando **lvs** es la siguiente:

```
# lvs --all --options name,copy_percent,devices my_vg

/dev/sdc: open failed: No such device or address
Couldn't find device with uuid A4kRI2-vlZA-uyCb-cci7-bOod-H5tX-lzH4Ee.
WARNING: Couldn't find all devices for LV my_vg/my_lv_rimage_1 while checking used and
assumed devices.
WARNING: Couldn't find all devices for LV my_vg/my_lv_rmeta_1 while checking used and
assumed devices.
LV          Cpy%Sync Devices
my_lv       100.00 my_lv_rimage_0(0),my_lv_rimage_1(0),my_lv_rimage_2(0)
[my_lv_rimage_0]    /dev/sde1(1)
[my_lv_rimage_1]    [unknown](1)
[my_lv_rimage_2]    /dev/sdd1(1)
[my_lv_rmeta_0]     /dev/sde1(0)
[my_lv_rmeta_1]     [unknown](0)
[my_lv_rmeta_2]     /dev/sdd1(0)
```

3. Sustituya el dispositivo que ha fallado y visualice el volumen lógico:

```
# lvconvert --repair my_vg/my_lv

/dev/sdc: open failed: No such device or address
Couldn't find device with uuid A4kRI2-vlZA-uyCb-cci7-bOod-H5tX-lzH4Ee.
WARNING: Couldn't find all devices for LV my_vg/my_lv_rimage_1 while checking used and
assumed devices.
WARNING: Couldn't find all devices for LV my_vg/my_lv_rmeta_1 while checking used and
```

```
assumed devices.
```

```
Attempt to replace failed RAID images (requires full device resync)? [y/n]: y
```

```
Faulty devices in my_vg/my_lv successfully replaced.
```

Opcional: Para especificar manualmente el volumen físico que sustituye al dispositivo que ha fallado, añada el volumen físico al final del comando:

```
# lvconvert --repair my_vg/my_lv replacement_pv
```

4. Examine el volumen lógico con la sustitución:

```
# lvs --all --options name,copy_percent,devices my_vg

/dev/sdc: open failed: No such device or address
/dev/sdc1: open failed: No such device or address
Couldn't find device with uuid A4kRI2-vlZA-uyCb-cci7-bOod-H5tX-lzH4Ee.
LV          Cpy%Sync Devices
my_lv       43.79  my_lv_rimage_0(0),my_lv_rimage_1(0),my_lv_rimage_2(0)
[my_lv_rimage_0]  /dev/sde1(1)
[my_lv_rimage_1]  /dev/sdb1(1)
[my_lv_rimage_2]  /dev/sdd1(1)
[my_lv_rmeta_0]   /dev/sde1(0)
[my_lv_rmeta_1]   /dev/sdb1(0)
[my_lv_rmeta_2]   /dev/sdd1(0)
```

Hasta que no elimine el dispositivo fallado del grupo de volúmenes, las utilidades LVM siguen indicando que LVM no puede encontrar el dispositivo fallado.

5. Elimine el dispositivo que ha fallado del grupo de volúmenes:

```
# vgreduce --removemissing VG
```

## 10.17. COMPROBACIÓN DE LA COHERENCIA DE LOS DATOS EN UN VOLUMEN LÓGICO RAID (RAID SCRUBBING)

LVM proporciona soporte de scrubbing para volúmenes lógicos RAID. El scrubbing de RAID es el proceso de lectura de todos los datos y bloques de paridad de un array y la comprobación de su coherencia.

### Procedimiento

1. Opcional: Limitar el ancho de banda de E/S que utiliza el proceso de depuración. Cuando se realiza una operación de depuración de RAID, la E/S en segundo plano requerida por las operaciones de **sync** puede desplazar otras E/S a los dispositivos LVM, como las actualizaciones de los metadatos del grupo de volúmenes. Esto puede hacer que las otras operaciones LVM se ralenticen. Se puede controlar la tasa de la operación de depuración implementando el estrangulamiento de la recuperación.

Añade las siguientes opciones a los comandos de **lvchange --syncaction** en los siguientes pasos:

```
--maxrecoveryrate Rate[bBsSkKmMgG]
```

Establece la tasa de recuperación máxima para que la operación no desplace las operaciones de E/S nominales. Establecer la tasa de recuperación en 0 significa que la operación no tiene límites.

### **--minrecoveryrate** *Rate*[bBsSkKmMgG]

Establece la tasa de recuperación mínima para garantizar que la E/S de las operaciones de **sync** alcance un rendimiento mínimo, incluso cuando haya una E/S nominal intensa.

Especifique el valor de *Rate* como una cantidad por segundo para cada dispositivo de la matriz. Si no proporciona ningún sufijo, las opciones asumen kiB por segundo por dispositivo.

2. Muestra el número de discrepancias en la matriz, sin repararlas:

```
# lvchange --syncaction check vg/raid_lv
```

3. Corregir las discrepancias en la matriz:

```
# lvchange --syncaction repair vg/raid_lv
```



### NOTA

La operación **lvchange --syncaction repair** no realiza la misma función que la operación **lvconvert --repair**:

- La operación **lvchange --syncaction repair** inicia una operación de sincronización en segundo plano en la matriz.
- La operación **lvconvert --repair** repara o sustituye los dispositivos que fallan en un espejo o volumen lógico RAID.

4. Opcional: Muestra información sobre la operación de lavado:

```
# lvs -o raid_sync_action,raid_mismatch_count vg/lv
```

- El campo **raid\_sync\_action** muestra la operación de sincronización actual que está realizando el volumen RAID. Puede ser uno de los siguientes valores:

#### **idle**

Todas las operaciones de sincronización completadas (sin hacer nada)

#### **resync**

Inicialización de una matriz o recuperación tras un fallo de la máquina

#### **recover**

Sustitución de un dispositivo en la matriz

#### **check**

Buscar incoherencias en las matrices

#### **repair**

Buscar y reparar incoherencias

- El campo **raid\_mismatch\_count** muestra el número de discrepancias encontradas durante una operación de **check**.
- El campo **Cpy%Sync** muestra el progreso de las operaciones de **sync**.

- El campo **lv\_attr** proporciona indicadores adicionales. El bit 9 de este campo muestra la salud del volumen lógico, y admite los siguientes indicadores:
  - **m** (mismatches) indica que hay discrepancias en un volumen lógico RAID. Este carácter se muestra después de que una operación de depuración haya detectado que partes del RAID no son coherentes.
  - **r** (refresh) indica que un dispositivo en un array RAID ha sufrido un fallo y el kernel lo considera como fallido, aunque LVM puede leer la etiqueta del dispositivo y considera que el dispositivo está operativo. Refresca el volumen lógico para notificar al kernel que el dispositivo está ahora disponible, o reemplaza el dispositivo si sospechas que ha fallado.

### Recursos adicionales

- Para más información, consulte las páginas de manual **lvchange(8)** y **lvraid(7)**.

## 10.18. CONVERSIÓN DE UN NIVEL RAID (ABSORCIÓN DE RAID)

LVM soporta *Raid takeover*, lo que significa convertir un volumen lógico RAID de un nivel RAID a otro (como de RAID 5 a RAID 6). El cambio de nivel RAID se hace normalmente para aumentar o disminuir la resistencia a los fallos de los dispositivos o para restripear los volúmenes lógicos. Se utiliza la dirección **lvconvert** para la absorción de RAID. Para obtener información sobre la adquisición de RAID y para ver ejemplos de uso de **lvconvert** para convertir un volumen lógico RAID, consulte la página de manual **lvraid(7)**.

## 10.19. CAMBIO DE ATRIBUTOS DE UN VOLUMEN RAID (RAID RESHAPE)

RAID *reshaping* significa cambiar los atributos de un volumen lógico RAID manteniendo el mismo nivel RAID. Algunos de los atributos que se pueden cambiar son la disposición del RAID, el tamaño de las franjas y el número de franjas. Para obtener información sobre la remodelación de RAID y ejemplos de uso del comando **lvconvert** para remodelar un volumen lógico RAID, consulte la página de manual **lvraid(7)**.

## 10.20. CONTROL DE LAS OPERACIONES DE E/S EN UN VOLUMEN LÓGICO RAID1

Puedes controlar las operaciones de E/S de un dispositivo en un volumen lógico RAID1 utilizando los parámetros **--writemostly** y **--writebehind** del comando **lvchange**. El formato para utilizar estos parámetros es el siguiente.

- **--[raid]writemostly *PhysicalVolume*[:{t|y|n}]**  
 Marca un dispositivo en un volumen lógico RAID1 como **write-mostly**. Todas las lecturas a estas unidades se evitarán a menos que sean necesarias. La configuración de este parámetro mantiene el número de operaciones de E/S en la unidad al mínimo. Por defecto, el atributo **write-mostly** se establece en sí para el volumen físico especificado en el volumen lógico. Es posible eliminar la bandera **write-mostly** añadiendo **:n** al volumen físico o alternar el valor especificando **:t**. El argumento **--writemostly** se puede especificar más de una vez en un solo comando, lo que hace posible alternar los atributos de sólo escritura para todos los volúmenes físicos en un volumen lógico a la vez.
- **--[raid]writebehind *IOCount***  
 Especifica el número máximo de escrituras pendientes que se permiten en los dispositivos de un

volumen lógico RAID1 que están marcados como **write-mostly**. Una vez que se excede este valor, las escrituras se vuelven sincrónicas, haciendo que todas las escrituras en los dispositivos constituyentes se completen antes de que el array señale que la escritura se ha completado. Poner el valor a cero borra la preferencia y permite al sistema elegir el valor de forma arbitraria.

## 10.21. CAMBIAR EL TAMAÑO DE LA REGIÓN EN UN VOLUMEN LÓGICO RAID

Cuando se crea un volumen lógico RAID, el tamaño de la región para el volumen lógico será el valor del parámetro **raid\_region\_size** en el archivo **/etc/lvm/lvm.conf**. Puedes anular este valor por defecto con la opción **-R** del comando **lvcreate**.

Después de crear un volumen lógico RAID, puede cambiar el tamaño de la región del volumen con la opción **-R** del comando **lvconvert**. El siguiente ejemplo cambia el tamaño de región del volumen lógico **vg/raidlv** a 4096K. El volumen RAID debe estar sincronizado para poder cambiar el tamaño de la región.

```
# lvconvert -R 4096K vg/raid1
```

```
Do you really want to change the region_size 512.00 KiB of LV vg/raid1 to 4.00 MiB? [y/n]: y
```

```
Changed region size on RAID LV vg/raid1 to 4.00 MiB.
```

# CAPÍTULO 11. VOLÚMENES LÓGICOS INSTANTÁNEOS

La función de instantáneas LVM proporciona la capacidad de crear imágenes virtuales de un dispositivo en un instante determinado sin causar una interrupción del servicio.

## 11.1. VOLÚMENES DE INSTANTÁNEAS

La función de instantáneas de LVM proporciona la capacidad de crear imágenes virtuales de un dispositivo en un instante determinado sin causar una interrupción del servicio. Cuando se realiza un cambio en el dispositivo original (el origen) después de tomar una instantánea, la función de instantánea hace una copia del área de datos modificada tal y como estaba antes del cambio para poder reconstruir el estado del dispositivo.



### NOTA

LVM soporta instantáneas de aprovisionamiento ligero.

Dado que una instantánea sólo copia las áreas de datos que cambian después de la creación de la instantánea, la función de instantánea requiere una cantidad mínima de almacenamiento. Por ejemplo, con un origen que se actualiza raramente, el 3-5 % de la capacidad del origen es suficiente para mantener la instantánea.



### NOTA

Las copias instantáneas de un sistema de archivos son copias virtuales, no una copia de seguridad real de un sistema de archivos. Las instantáneas no sustituyen a un procedimiento de copia de seguridad.

El tamaño de la instantánea gobierna la cantidad de espacio reservado para almacenar los cambios en el volumen de origen. Por ejemplo, si haces una instantánea y luego sobrescribes completamente el origen, la instantánea tendría que ser al menos tan grande como el volumen de origen para mantener los cambios. Es necesario dimensionar una instantánea de acuerdo con el nivel de cambio esperado. Así, por ejemplo, una instantánea de corta duración de un volumen de lectura, como **/usr**, necesitaría menos espacio que una instantánea de larga duración de un volumen que ve un mayor número de escrituras, como **/home**.

Si una instantánea se llena, la instantánea deja de ser válida, ya que no puede seguir los cambios en el volumen de origen. Debe controlar regularmente el tamaño de la instantánea. Sin embargo, las instantáneas son totalmente redimensionables, por lo que si tienes capacidad de almacenamiento puedes aumentar el tamaño del volumen de la instantánea para evitar que se pierda. A la inversa, si ves que el volumen de la instantánea es más grande de lo que necesitas, puedes reducir el tamaño del volumen para liberar espacio que necesitan otros volúmenes lógicos.

Cuando se crea un sistema de archivos de instantánea, el acceso de lectura y escritura completo al origen sigue siendo posible. Si se modifica un chunk en una instantánea, ese chunk se marca y nunca se copia del volumen original.

La función de instantáneas tiene varios usos:

- Normalmente, se toma una instantánea cuando se necesita realizar una copia de seguridad en un volumen lógico sin detener el sistema en vivo que está actualizando continuamente los datos.

- Puede ejecutar el comando **fsck** en un sistema de archivos de instantánea para comprobar la integridad del sistema de archivos y determinar si el sistema de archivos original requiere una reparación del sistema de archivos.
- Dado que la instantánea es de lectura/escritura, puede probar las aplicaciones contra los datos de producción tomando una instantánea y ejecutando pruebas contra la instantánea, dejando los datos reales intactos.
- Puede crear volúmenes LVM para utilizarlos con Red Hat Virtualization. Las instantáneas LVM se pueden utilizar para crear instantáneas de imágenes de huéspedes virtuales. Estas instantáneas pueden proporcionar una forma conveniente de modificar los huéspedes existentes o crear nuevos huéspedes con un mínimo de almacenamiento adicional.

Puedes utilizar la opción **--merge** del comando **lvconvert** para fusionar una instantánea con su volumen de origen. Una de las aplicaciones de esta función es realizar la reversión del sistema si has perdido datos o archivos o si necesitas restaurar tu sistema a un estado anterior. Después de fusionar el volumen de la instantánea, el volumen lógico resultante tendrá el nombre, el número menor y el UUID del volumen de origen y la instantánea fusionada se eliminará.

## 11.2. CREACIÓN DE VOLÚMENES DE INSTANTÁNEAS

Utilice el argumento **-s** del comando **lvcreate** para crear un volumen de instantánea. Un volumen de instantánea es escribible.



### NOTA

Las instantáneas LVM no son compatibles con los nodos de un clúster. No se puede crear un volumen de instantánea en un grupo de volumen compartido. Sin embargo, si necesitas crear una copia de seguridad consistente de los datos en un volumen lógico compartido, puedes activar el volumen exclusivamente y luego crear la instantánea.



### NOTA

Las instantáneas son compatibles con los volúmenes lógicos RAID.

LVM no permite crear un volumen de instantánea que sea mayor que el tamaño del volumen de origen más los metadatos necesarios para el volumen. Si especifica un volumen de instantánea que es mayor que esto, el sistema creará un volumen de instantánea que es sólo tan grande como se necesitará para el tamaño del origen.

Por defecto, un volumen de instantánea se omite durante los comandos de activación normales.

El siguiente procedimiento crea un volumen lógico de origen llamado **origin** y un volumen instantáneo del volumen original llamado **snap**.

1. Cree un volumen lógico llamado **origin** a partir del grupo de volúmenes **VG**.

```
# lvcreate -L 1G -n origin VG
Logical volume "origin" created.
```

2. Cree un volumen lógico de instantánea de **/dev/VG/origin** que tenga un tamaño de 100 MB llamado **snap**. Si el volumen lógico original contiene un sistema de archivos, puedes montar el volumen lógico de instantánea en un directorio arbitrario para acceder al contenido del sistema de archivos y ejecutar una copia de seguridad mientras el sistema de archivos original sigue actualizándose.

```
# lvcreate --size 100M --snapshot --name snap /dev/VG/origin
Logical volume "snap" created.
```

3. Muestra el estado del volumen lógico **/dev/VG/origin**, mostrando todos los volúmenes lógicos instantáneos y su estado (activo o inactivo).

```
# lvsdisplay /dev/VG/origin
--- Logical volume ---
LV Path          /dev/VG/origin
LV Name          origin
VG Name          VG
LV UUID          EsFoBp-CB9H-Epl5-pUO4-Yevi-EdFS-xtFnaF
LV Write Access  read/write
LV Creation host, time host-083.virt.lab.msp.redhat.com, 2019-04-11 14:45:06 -0500
LV snapshot status source of
                  snap [active]
LV Status        available
# open           0
LV Size          1.00 GiB
Current LE       256
Segments         1
Allocation       inherit
Read ahead sectors auto
- currently set to 8192
Block device     253:6
```

4. El comando **lvs**, por defecto, muestra el volumen de origen y el porcentaje actual del volumen de instantánea que se está utilizando. El siguiente ejemplo muestra la salida por defecto del comando **lvs** después de haber creado el volumen de instantánea, con una visualización que incluye los dispositivos que constituyen los volúmenes lógicos.

```
# lvs -a -o +devices
LV      VG      Attr      LSize  Pool  Origin Data%  Meta%  Move Log Cpy%Sync Convert
Devices
origin  VG      owi-a-s--- 1.00g                /dev/sde1(0)
snap    VG      swi-a-s--- 100.00m  origin 0.00                /dev/sde1(256)
```



### AVISO

Dado que la instantánea aumenta de tamaño a medida que el volumen de origen cambia, es importante controlar el porcentaje del volumen de la instantánea regularmente con el comando **lvs** para asegurarse de que no se llena. Una instantánea que se llena 100 veces se pierde por completo, ya que una escritura en partes del origen que no han cambiado no podría tener éxito sin corromper la instantánea.

Además de que la propia instantánea se invalida cuando está llena, cualquier sistema de archivos montado en ese dispositivo de instantánea se desmonta a la fuerza, evitando los inevitables errores del sistema de archivos al acceder al punto de montaje. Además, puede especificar la opción

**snapshot\_autoextend\_threshold** en el archivo **lvm.conf**. Esta opción permite la extensión automática de una instantánea siempre que el espacio restante de la instantánea caiga por debajo del umbral que hayas establecido. Esta función requiere que haya espacio sin asignar en el grupo de volúmenes.

LVM no permite crear un volumen de instantánea que sea mayor que el tamaño del volumen de origen más los metadatos necesarios para el volumen. Del mismo modo, la extensión automática de una instantánea no aumentará el tamaño de un volumen de instantánea más allá del tamaño máximo calculado que es necesario para la instantánea. Una vez que una instantánea ha crecido lo suficiente como para cubrir el origen, ya no se supervisa la extensión automática.

La información sobre la configuración de **snapshot\_autoextend\_threshold** y **snapshot\_autoextend\_percent** se proporciona en el propio archivo **/etc/lvm/lvm.conf**.

### 11.3. FUSIÓN DE VOLÚMENES DE INSTANTÁNEAS

Puedes utilizar la opción **--merge** del comando **lvconvert** para fusionar una instantánea con su volumen de origen. Si tanto el volumen de origen como el de la instantánea no están abiertos, la fusión se iniciará inmediatamente. En caso contrario, la fusión se iniciará la primera vez que el origen o la instantánea se activen y ambos se cierren. La fusión de una instantánea en un origen que no puede cerrarse, por ejemplo un sistema de archivos raíz, se aplaza hasta la próxima vez que se active el volumen de origen. Cuando se inicia la fusión, el volumen lógico resultante tendrá el nombre del origen, el número menor y el UUID. Mientras la fusión está en curso, las lecturas o escrituras en el origen aparecen como si fueran dirigidas a la instantánea que se está fusionando. Cuando la fusión termina, la instantánea fusionada se elimina.

El siguiente comando fusiona el volumen snapshot **vg00/lvol1\_snap** en su origen.

```
# lvconvert --merge vg00/lvol1_snap
```

Puedes especificar múltiples instantáneas en la línea de comandos, o puedes usar etiquetas de objetos LVM para especificar que varias instantáneas se fusionen con sus respectivos orígenes. En el siguiente ejemplo, los volúmenes lógicos **vg00/lvol1**, **vg00/lvol2**, y **vg00/lvol3** están todos etiquetados con la etiqueta **@some\_tag**. El siguiente comando fusiona los volúmenes lógicos de instantáneas para los tres volúmenes en serie: **vg00/lvol1**, luego **vg00/lvol2**, luego **vg00/lvol3**. Si se utilizara la opción **--background**, todas las fusiones de volúmenes lógicos de instantáneas se iniciarían en paralelo.

```
# lvconvert --merge @some_tag
```

Para más información sobre el comando **lvconvert --merge**, consulte la página de manual **lvconvert(8)**.

## CAPÍTULO 12. CREACIÓN Y GESTIÓN DE VOLÚMENES LÓGICOS DE APROVISIONAMIENTO LIGERO (VOLÚMENES LIGEROS)

Los volúmenes lógicos pueden tener un aprovisionamiento ligero. Esto le permite crear volúmenes lógicos que son más grandes que los extensiones disponibles.

### 12.1. VOLÚMENES LÓGICOS DE APROVISIONAMIENTO FINO (VOLÚMENES FINOS)

Los volúmenes lógicos pueden tener un aprovisionamiento ligero. Esto le permite crear volúmenes lógicos que son más grandes que los extensiones disponibles. Mediante el aprovisionamiento fino, puede gestionar un pool de almacenamiento de espacio libre, conocido como thin pool, que puede asignarse a un número arbitrario de dispositivos cuando lo necesiten las aplicaciones. A continuación, puede crear dispositivos que se pueden vincular al thin pool para su posterior asignación cuando una aplicación escriba realmente en el volumen lógico. El thin pool puede ampliarse dinámicamente cuando sea necesario para una asignación rentable del espacio de almacenamiento.



#### NOTA

Los volúmenes ligeros no son compatibles con los nodos de un clúster. La reserva delgada y todos sus volúmenes delgados deben activarse exclusivamente en un solo nodo del clúster.

Al utilizar el thin provisioning, un administrador de almacenamiento puede sobrecomprometer el almacenamiento físico, evitando a menudo la necesidad de comprar almacenamiento adicional. Por ejemplo, si diez usuarios solicitan cada uno un sistema de archivos de 100 GB para su aplicación, el administrador de almacenamiento puede crear lo que parece ser un sistema de archivos de 100 GB para cada usuario, pero que está respaldado por menos almacenamiento real que se utiliza sólo cuando se necesita. Cuando se utiliza el thin provisioning, es importante que el administrador de almacenamiento supervise el pool de almacenamiento y añada más capacidad si empieza a llenarse.

Para asegurarse de que todo el espacio disponible puede ser utilizado, LVM soporta el descarte de datos. Esto permite reutilizar el espacio que antes utilizaba un archivo descartado u otro rango de bloques.

Los volúmenes delgados ofrecen soporte para una nueva implementación de volúmenes lógicos de copia en escritura (COW), que permiten que muchos dispositivos virtuales compartan los mismos datos en el grupo delgado.

### 12.2. CREACIÓN DE VOLÚMENES LÓGICOS DE APROVISIONAMIENTO LIGERO

Este procedimiento proporciona una visión general de los comandos básicos que se utilizan para crear y aumentar los volúmenes lógicos de aprovisionamiento ligero. Para obtener información detallada sobre el aprovisionamiento ligero de LVM, así como información sobre el uso de los comandos y utilidades de LVM con volúmenes lógicos de aprovisionamiento ligero, consulte la página de manual [lvmthin\(7\)](#).

Para crear un volumen ligero, realice las siguientes tareas:

1. Cree un grupo de volúmenes con el comando **vgcreate**.

2. Cree un grupo ligero con el comando **lvcreate**.
3. Cree un volumen delgado en la reserva delgada con el comando **lvcreate**.

Puedes utilizar la opción **-T** (o **--thin**) del comando **lvcreate** para crear un pool delgado o un volumen delgado. También puedes utilizar la opción **-T** del comando **lvcreate** para crear un pool delgado y un volumen delgado en ese pool al mismo tiempo con un solo comando.

El siguiente comando utiliza la opción **-T** del comando **lvcreate** para crear un thin pool llamado **mythinpool** en el grupo de volúmenes **vg001** y que tiene un tamaño de 100M. Ten en cuenta que como estás creando un pool de espacio físico, debes especificar el tamaño del pool. La opción **-T** del comando **lvcreate** no toma un argumento; deduce qué tipo de dispositivo se va a crear a partir de las otras opciones que especifica el comando.

```
# lvcreate -L 100M -T vg001/mythinpool
Thin pool volume with chunk size 64.00 KiB can address at most 15.81 TiB of data.
Logical volume "mythinpool" created.
# lvs
LV      VG      Attr      LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
mythinpool vg001 twi-a-tz-- 100.00m          0.00  10.84
```

El siguiente comando utiliza la opción **-T** del comando **lvcreate** para crear un volumen delgado llamado **thinvolume** en el pool delgado **vg001/mythinpool**. Ten en cuenta que en este caso estás especificando el tamaño virtual, y que estás especificando un tamaño virtual para el volumen que es mayor que el pool que lo contiene.

```
# lvcreate -V 1G -T vg001/mythinpool -n thinvolume
WARNING: Sum of all thin volume sizes (1.00 GiB) exceeds the size of thin pool vg001/mythinpool (100.00 MiB).
WARNING: You have not turned on protection against thin pools running out of space.
WARNING: Set activation/thin_pool_autoextend_threshold below 100 to trigger automatic extension of thin pools before they get full.
Logical volume "thinvolume" created.
# lvs
LV      VG      Attr      LSize   Pool      Origin Data%  Move Log Copy% Convert
mythinpool vg001 twi-a-tz 100.00m          0.00
thinvolume vg001 Vwi-a-tz 1.00g mythinpool      0.00
```

El siguiente comando utiliza la opción **-T** del comando **lvcreate** para crear una reserva delgada y un volumen delgado en esa reserva especificando tanto un argumento de tamaño como de tamaño virtual para el comando **lvcreate**. Este comando crea una reserva delgada llamada **mythinpool** en el grupo de volúmenes **vg001** y también crea un volumen delgado llamado **thinvolume** en esa reserva.

```
# lvcreate -L 100M -T vg001/mythinpool -V 1G -n thinvolume
Thin pool volume with chunk size 64.00 KiB can address at most 15.81 TiB of data.
WARNING: Sum of all thin volume sizes (1.00 GiB) exceeds the size of thin pool vg001/mythinpool (100.00 MiB).
WARNING: You have not turned on protection against thin pools running out of space.
WARNING: Set activation/thin_pool_autoextend_threshold below 100 to trigger automatic extension of thin pools before they get full.
Logical volume "thinvolume" created.
# lvs
LV      VG      Attr      LSize   Pool      Origin Data%  Meta%  Move Log Cpy%Sync Convert
mythinpool vg001 twi-aotz-- 100.00m          0.00  10.94
thinvolume vg001 Vwi-a-tz-- 1.00g mythinpool      0.00
```

También puedes crear un thin pool especificando el parámetro **--thinpool** del comando **lvcreate**. A diferencia de la opción **-T**, el parámetro **--thinpool** requiere un argumento, que es el nombre del volumen lógico del thin pool que se está creando. El siguiente ejemplo especifica el parámetro **--thinpool** del comando **lvcreate** para crear un thin pool llamado **mythinpool** en el grupo de volúmenes **vg001** y que tiene un tamaño de 100M:

```
# lvcreate -L 100M --thinpool mythinpool vg001
Thin pool volume with chunk size 64.00 KiB can address at most 15.81 TiB of data.
Logical volume "mythinpool" created.
# lvs
LV      VG   Attr   LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
mythinpool vg001 twi-a-tz-- 100.00m          0.00  10.84
```

Utiliza los siguientes criterios para utilizar el tamaño de los trozos:

- Un tamaño de trozo más pequeño requiere más metadatos y dificulta el rendimiento, pero proporciona una mejor utilización del espacio con las instantáneas.
- Un tamaño de trozo mayor requiere menos manipulación de metadatos, pero hace que la instantánea sea menos eficiente en cuanto a espacio.

Por defecto, **lvm2** comienza con un tamaño de chunk de 64KiB y aumenta su valor cuando el tamaño resultante del dispositivo de metadatos del thin pool crece por encima de los 128MiB, esto mantiene el tamaño de los metadatos compacto. Sin embargo, esto puede resultar en algunos valores de tamaño de chunk grandes, que son menos eficientes en cuanto a espacio para el uso de instantáneas. En estos casos, un tamaño de chunk más pequeño y un tamaño de metadatos más grande es una mejor opción.

Si el tamaño de los datos del volumen está en el rango de TiB, utilice ~15.8GiB como tamaño de los metadatos, que es el tamaño máximo soportado, y establezca el tamaño de los trozos según sus necesidades. Sin embargo, tenga en cuenta que no es posible aumentar el tamaño de los metadatos si necesita ampliar el tamaño de los datos del volumen y tiene un tamaño de trozo pequeño.



## AVISO

Red Hat no recomienda establecer un tamaño de chunk menor que el valor por defecto. Si el tamaño del chunk es demasiado pequeño y su volumen se queda sin espacio para los metadatos, el volumen no podrá crear datos. Supervise sus volúmenes lógicos para asegurarse de que se amplían, o cree más almacenamiento antes de que los volúmenes de metadatos se llenen por completo. Asegúrese de configurar su thin pool con un tamaño de trozo lo suficientemente grande para que no se queden sin espacio para los metadatos.

El striping es soportado para la creación de pools. El siguiente comando crea una reserva delgada de 100M llamada **pool** en el grupo de volúmenes **vg001** con dos franjas de 64 kB y un tamaño de trozo de 256 kB. También crea un volumen fino de 1T, **vg00/thin\_lv**.

```
# lvcreate -i 2 -l 64 -c 256 -L 100M -T vg00/pool -V 1T --name thin_lv
```

Puede ampliar el tamaño de un volumen ligero con el comando **lvextend**. Sin embargo, no se puede reducir el tamaño de un thin pool.

El siguiente comando redimensiona un thin pool existente que tiene un tamaño de 100M ampliándolo otros 100M.

```
# lvextend -L+100M vg001/mythinpool
Extending logical volume mythinpool to 200.00 MiB
Logical volume mythinpool successfully resized
# lvs
LV      VG      Attr  LSize  Pool   Origin Data%  Move Log Copy%  Convert
mythinpool  vg001  twi-a-tz 200.00m          0.00
thinvolume  vg001  Vwi-a-tz 1.00g mythinpool      0.00
```

Al igual que con otros tipos de volúmenes lógicos, puedes renombrar el volumen con el comando **lvrename**, puedes eliminar el volumen con el comando **lvremove**, y puedes mostrar información sobre el volumen con los comandos **lvs** y **lvdisplay**.

By default, the **lvcreate** command sets the size of the thin pool's metadata logical volume according to the formula  $(\text{Pool\_LV\_size} / \text{Pool\_LV\_chunk\_size} * 64)$ . If you will have large numbers of snapshots or if you have small chunk sizes for your thin pool and thus expect significant growth of the size of the thin pool at a later time, you may need to increase the default value of the thin pool's metadata volume with the **--poolmetadatasize** parameter of the **lvcreate** command. The supported value for the thin pool's metadata logical volume is in the range between 2MiB and 16GiB.

Puede utilizar el parámetro **--thinpool** del comando **lvconvert** para convertir un volumen lógico existente en un volumen de reserva delgada. Cuando convierta un volumen lógico existente en un volumen de reserva delgada, debe utilizar el parámetro **--poolmetadata** junto con el parámetro **--thinpool** del comando **lvconvert** para convertir un volumen lógico existente en el volumen de metadatos del volumen de reserva delgada.



#### NOTA

La conversión de un volumen lógico en un volumen thin pool o en un volumen de metadatos thin pool destruye el contenido del volumen lógico, ya que en este caso el **lvconvert** no conserva el contenido de los dispositivos, sino que lo sobrescribe.

El siguiente ejemplo convierte el volumen lógico existente **lv1** en el grupo de volúmenes **vg001** en un volumen thin pool y convierte el volumen lógico existente **lv2** en el grupo de volúmenes **vg001** en el volumen de metadatos para ese volumen thin pool.

```
# lvconvert --thinpool vg001/lv1 --poolmetadata vg001/lv2
Converted vg001/lv1 to thin pool.
```

## 12.3. VOLÚMENES INSTANTÁNEOS DE APROVISIONAMIENTO LIGERO

Red Hat Enterprise Linux proporciona soporte para volúmenes de instantáneas de aprovisionamiento delgado. Los volúmenes de instantáneas delgadas permiten que muchos dispositivos virtuales sean almacenados en el mismo volumen de datos. Esto simplifica la administración y permite compartir datos entre volúmenes de instantáneas.

Como para todos los volúmenes de instantánea LVM, así como para todos los volúmenes finos, los volúmenes finos de instantánea no son compatibles con los nodos de un clúster. El volumen de instantánea debe activarse exclusivamente en un solo nodo del clúster.

Los volúmenes de instantáneas delgadas proporcionan los siguientes beneficios:

- Un volumen delgado de instantáneas puede reducir el uso del disco cuando hay varias instantáneas del mismo volumen de origen.
- Si hay varias instantáneas del mismo origen, una escritura en el origen provocará una operación COW para conservar los datos. Aumentar el número de instantáneas del origen no debería suponer una ralentización importante.
- Los volúmenes finos de instantáneas pueden utilizarse como origen de volumen lógico para otra instantánea. Esto permite una profundidad arbitraria de instantáneas recursivas (instantáneas de instantáneas de instantáneas...).
- Una instantánea de un volumen lógico delgado también crea un volumen lógico delgado. Esto no consume espacio de datos hasta que se requiera una operación COW, o hasta que se escriba la propia instantánea.
- Un volumen delgado de instantáneas no necesita ser activado con su origen, por lo que un usuario puede tener sólo el origen activo mientras hay muchos volúmenes de instantáneas inactivos del origen.
- Cuando se elimina el origen de un volumen de instantáneas de aprovisionamiento ligero, cada instantánea de ese volumen de origen se convierte en un volumen de aprovisionamiento ligero independiente. Esto significa que, en lugar de fusionar una instantánea con su volumen de origen, puedes optar por eliminar el volumen de origen y luego crear una nueva instantánea con aprovisionamiento ligero utilizando ese volumen independiente como volumen de origen para la nueva instantánea.

Aunque hay muchas ventajas en el uso de volúmenes delgados de instantáneas, hay algunos casos de uso para los que la antigua función de volumen de instantáneas LVM puede ser más apropiada para sus necesidades:

- No se puede cambiar el tamaño del chunk de un thin pool. Si el thin pool tiene un tamaño de chunk grande (por ejemplo, 1MB) y necesita una instantánea de corta duración para la que un tamaño de chunk tan grande no es eficiente, puede optar por utilizar la función de instantáneas antiguas.
- No se puede limitar el tamaño de un volumen de instantánea delgada; la instantánea utilizará todo el espacio de la reserva delgada, si es necesario. Esto puede no ser apropiado para sus necesidades.

En general, debe tener en cuenta los requisitos específicos de su sitio web a la hora de decidir qué formato de instantánea utilizar.

## 12.4. CREACIÓN DE VOLÚMENES DE INSTANTÁNEAS CON APROVISIONAMIENTO LIGERO

Red Hat Enterprise Linux ofrece soporte para volúmenes de instantáneas de aprovisionamiento ligero.



### NOTA

Esta sección proporciona una visión general de los comandos básicos que se utilizan para crear y hacer crecer los volúmenes instantáneos de aprovisionamiento ligero. Para obtener información detallada sobre el aprovisionamiento ligero de LVM, así como información sobre el uso de los comandos y utilidades de LVM con volúmenes lógicos de aprovisionamiento ligero, consulte la página de manual **lvmtthin(7)**.



## IMPORTANTE

Cuando se crea un volumen de instantánea delgada, no se especifica el tamaño del volumen. Si se especifica un parámetro de tamaño, la instantánea que se creará no será un volumen de instantánea delgada y no utilizará el pool delgado para almacenar datos. Por ejemplo, el comando **lvcreate -s vg/thinvolume -L10M** no creará una instantánea delgada, aunque el volumen de origen sea un volumen delgado.

Las instantáneas delgadas pueden crearse para volúmenes de origen con aprovisionamiento delgado o para volúmenes de origen que no tienen aprovisionamiento delgado.

Puedes especificar un nombre para el volumen instantáneo con la opción **--name** del comando **lvcreate**. El siguiente comando crea un volumen de instantánea de aprovisionamiento ligero del volumen lógico de aprovisionamiento ligero **vg001/thinvolume** que se llama **mynsnapshot1**.

```
# lvcreate -s --name mynsnapshot1 vg001/thinvolume
Logical volume "mynsnapshot1" created
# lvs
LV      VG      Attr  LSize Pool   Origin  Data% Move Log Copy% Convert
mynsnapshot1 vg001  Vwi-a-tz 1.00g mythinpool thinvolume 0.00
mythinpool vg001  twi-a-tz 100.00m          0.00
thinvolume vg001  Vwi-a-tz 1.00g mythinpool          0.00
```

Un volumen delgado de instantánea tiene las mismas características que cualquier otro volumen delgado. Puedes activar el volumen de forma independiente, ampliar el volumen, renombrar el volumen, eliminar el volumen e incluso hacer una instantánea del volumen.

Por defecto, un volumen instantáneo se omite durante los comandos de activación normales. Para obtener información sobre el control de la activación de un volumen lógico, consulte [Activación de volúmenes lógicos](#).

También puede crear una instantánea con aprovisionamiento ligero de un volumen lógico sin aprovisionamiento ligero. Dado que el volumen lógico no aprovisionado en capa fina no está contenido en un fondo común, se denomina *external origin*. Los volúmenes de origen externo pueden ser utilizados y compartidos por muchos volúmenes instantáneos de aprovisionamiento fino, incluso de diferentes pools finos. El origen externo debe estar inactivo y ser de sólo lectura en el momento en que se crea la instantánea de aprovisionamiento fino.

Para crear una instantánea con aprovisionamiento ligero de un origen externo, debes especificar la opción **--thinpool**. El siguiente comando crea un volumen de instantánea delgada del volumen inactivo de sólo lectura **origin\_volume**. El volumen thin snapshot se llama **mythinsnap**. El volumen lógico **origin\_volume** se convierte entonces en el origen externo delgado para el volumen instantáneo delgado **mythinsnap** en el grupo de volúmenes **vg001** que utilizará la reserva delgada existente **vg001/pool**. Debido a que el volumen de origen debe estar en el mismo grupo de volúmenes que el volumen de instantánea, no es necesario especificar el grupo de volúmenes cuando se especifica el volumen lógico de origen.

```
# lvcreate -s --thinpool vg001/pool origin_volume --name mythinsnap
```

Puedes crear un segundo volumen de instantánea con aprovisionamiento ligero del primer volumen de instantánea, como en el siguiente comando.

```
# lvcreate -s vg001/mythinsnap --name my2ndthinsnap
```

Puedes mostrar una lista de todos los ancestros y descendientes de un volumen lógico de instantánea delgada especificando los campos de información **lv\_ancestors** y **lv\_descendants** del comando **lvs**.

En el siguiente ejemplo:

- **stack1** es un volumen de origen en el grupo de volúmenes **vg001**.
- **stack2** es una instantánea de **stack1**
- **stack3** es una instantánea de **stack2**
- **stack4** es una instantánea de **stack3**

Además:

- **stack5** es también una instantánea de **stack2**
- **stack6** es una instantánea de **stack5**

```
$ lvs -o name,lv_ancestors,lv_descendants vg001
LV   Ancestors          Descendants
stack1                               stack2,stack3,stack4,stack5,stack6
stack2 stack1           stack3,stack4,stack5,stack6
stack3 stack2,stack1    stack4
stack4 stack3,stack2,stack1
stack5 stack2,stack1     stack6
stack6 stack5,stack2,stack1
pool
```

## NOTA

Los campos **lv\_ancestors** y **lv\_descendants** muestran las dependencias existentes, pero no hacen un seguimiento de las entradas eliminadas, lo que puede romper una cadena de dependencia si la entrada se eliminó de la mitad de la cadena. Por ejemplo, si se elimina el volumen lógico **stack3** de esta configuración de ejemplo, la pantalla es la siguiente.

```
$ lvs -o name,lv_ancestors,lv_descendants vg001
LV   Ancestors          Descendants
stack1                               stack2,stack5,stack6
stack2 stack1           stack5,stack6
stack4
stack5 stack2,stack1     stack6
stack6 stack5,stack2,stack1
pool
```

Puede configurar su sistema para rastrear y mostrar los volúmenes lógicos que han sido eliminados, y puede mostrar la cadena de dependencia completa que incluye esos volúmenes especificando los campos **lv\_ancestors\_full** y **lv\_descendants\_full**.

## 12.5. SEGUIMIENTO Y VISUALIZACIÓN DE LOS VOLÚMENES FINOS DE INSTANTÁNEAS QUE SE HAN ELIMINADO

Puede configurar su sistema para realizar un seguimiento de los volúmenes lógicos finos y de instantáneas finas que se han eliminado activando la opción de metadatos **record\_lvs\_history** en el

archivo de configuración **lvm.conf**. Esto le permite mostrar una cadena de dependencia de instantáneas delgadas completa que incluye los volúmenes lógicos que se han eliminado de la cadena de dependencia original y se han convertido en volúmenes lógicos de *historical*.

Puede configurar su sistema para retener los volúmenes históricos durante un periodo de tiempo definido especificando el tiempo de retención, en segundos, con la opción de metadatos **lvs\_history\_retention\_time** en el archivo de configuración **lvm.conf**.

Un volumen lógico histórico conserva una representación simplificada del volumen lógico que ha sido eliminado, incluyendo los siguientes campos de información para el volumen:

- **lv\_time\_removed**: el tiempo de eliminación del volumen lógico
- **lv\_time**: la hora de creación del volumen lógico
- **lv\_name**: el nombre del volumen lógico
- **lv\_uuid**: el UUID del volumen lógico
- **vg\_name**: el grupo de volumen que contiene el volumen lógico.

Cuando se elimina un volumen, el nombre del volumen lógico histórico adquiere un guión como prefijo. Por ejemplo, cuando se elimina el volumen lógico **lvvol1**, el nombre del volumen histórico es **-lvvol1**. Un volumen lógico histórico no puede ser reactivado.

Incluso con la opción de metadatos **record\_lvs\_history** activada, puede evitar la retención de volúmenes lógicos históricos de forma individual cuando elimine un volumen lógico especificando la opción **--nohistory** del comando **lvremove**.

Para incluir los volúmenes lógicos históricos en la visualización del volumen, se especifica la opción **-H|--history** de un comando de visualización de LVM. Puede mostrar una cadena de dependencia de instantáneas delgadas completa que incluya volúmenes históricos especificando los campos de información **lv\_full\_ancestors** y **lv\_full\_descendants** junto con la opción **-H**.

La siguiente serie de comandos proporciona ejemplos de cómo puede visualizar y gestionar los volúmenes lógicos históricos.

1. Asegúrese de que los volúmenes lógicos históricos se conservan estableciendo **record\_lvs\_history=1** en el archivo **lvm.conf**. Esta opción de metadatos no está activada por defecto.
2. Introduzca el siguiente comando para mostrar una cadena de instantáneas con aprovisionamiento ligero.  
En este ejemplo:

- **lvvol1** es un volumen de origen, el primer volumen de la cadena.
- **lvvol2** es una instantánea de **lvvol1**.
- **lvvol3** es una instantánea de **lvvol2**.
- **lvvol4** es una instantánea de **lvvol3**.
- **lvvol5** es también una instantánea de **lvvol3**.

Tenga en cuenta que, aunque el comando de visualización de **lvs** de ejemplo incluye la opción **-H**, todavía no se ha eliminado ningún volumen delgado de instantáneas y no hay volúmenes lógicos históricos que mostrar.

```
# lvs -H -o name,full_ancestors,full_descendants
LV  FAncestors    FDescendants
lvol1          lvol2,lvol3,lvol4,lvol5
lvol2 lvol1      lvol3,lvol4,lvol5
lvol3 lvol2,lvol1  lvol4,lvol5
lvol4 lvol3,lvol2,lvol1
lvol5 lvol3,lvol2,lvol1
pool
```

- Elimine el volumen lógico **lvvol3** de la cadena de instantáneas y, a continuación, vuelva a ejecutar el siguiente comando **lvs** para ver cómo se muestran los volúmenes lógicos históricos, junto con sus ancestros y descendientes.

```
# lvremove -f vg/lvol3
Logical volume "lvvol3" successfully removed
# lvs -H -o name,full_ancestors,full_descendants
LV  FAncestors    FDescendants
lvol1          lvol2,-lvvol3,lvol4,lvol5
lvol2 lvol1      -lvvol3,lvol4,lvol5
-lvvol3 lvol2,lvol1  lvol4,lvol5
lvol4 -lvvol3,lvol2,lvol1
lvol5 -lvvol3,lvol2,lvol1
pool
```

- Puede utilizar el campo de informes **lv\_time\_removed** para mostrar la hora en que se eliminó un volumen histórico.

```
# lvs -H -o name,full_ancestors,full_descendants,time_removed
LV  FAncestors    FDescendants    RTime
lvol1          lvol2,-lvvol3,lvol4,lvol5
lvol2 lvol1      -lvvol3,lvol4,lvol5
-lvvol3 lvol2,lvol1  lvol4,lvol5    2016-03-14 14:14:32 +0100
lvol4 -lvvol3,lvol2,lvol1
lvol5 -lvvol3,lvol2,lvol1
pool
```

- Puedes referenciar volúmenes lógicos históricos individualmente en un comando de visualización especificando el formato *vgname/lvname*, como en el siguiente ejemplo. Tenga en cuenta que el quinto bit del campo **lv\_attr** se establece en **h** para indicar que el volumen es un volumen histórico.

```
# lvs -H vg/-lvvol3
LV  VG  Attr  LSize
-lvvol3 vg  ----h----- 0
```

- LVM no conserva los volúmenes lógicos históricos si el volumen no tiene ningún descendiente vivo. Esto significa que si se elimina un volumen lógico al final de una cadena de instantáneas, el volumen lógico no se conserva como volumen lógico histórico.

```
# lvremove -f vg/lvol5
Automatically removing historical logical volume vg/-lvvol5.
Logical volume "lvvol5" successfully removed
# lvs -H -o name,full_ancestors,full_descendants
LV  FAncestors    FDescendants
```

```
lvol1          lvol2,-lvol3,lvoll4
lvoll2 lvol1    -lvol3,lvoll4
-lvol3 lvol2,lvoll1    lvol4
lvoll4 -lvol3,lvoll2,lvoll1
pool
```

- Ejecute los siguientes comandos para eliminar el volumen **lvoll1** y **lvoll2** y para ver cómo el comando **lvs** muestra los volúmenes una vez eliminados.

```
# lvremove -f vg/lvoll1 vg/lvoll2
Logical volume "lvoll1" successfully removed
Logical volume "lvoll2" successfully removed
# lvs -H -o name,full_ancestors,full_descendants
LV  FAncestors      FDescendants
-lvoll1          -lvoll2,-lvoll3,lvoll4
-lvoll2 -lvoll1    -lvoll3,lvoll4
-lvoll3 -lvoll2,-lvoll1    lvol4
lvoll4 -lvoll3,-lvoll2,-lvoll1
pool
```

- Para eliminar un volumen lógico histórico por completo, puedes ejecutar el comando **lvremove** de nuevo, especificando el nombre del volumen histórico que ahora incluye el guión, como en el siguiente ejemplo.

```
# lvremove -f vg/-lvoll3
Historical logical volume "lvoll3" successfully removed
# lvs -H -o name,full_ancestors,full_descendants
LV  FAncestors      FDescendants
-lvoll1          -lvoll2,lvoll4
-lvoll2 -lvoll1    lvol4
lvoll4 -lvoll2,-lvoll1
pool
```

- Un volumen lógico histórico se conserva mientras exista una cadena que incluya volúmenes vivos en sus descendientes. Esto significa que la eliminación de un volumen lógico histórico también elimina todos los volúmenes lógicos de la cadena si no hay ningún descendiente vinculado a ellos, como se muestra en el siguiente ejemplo.

```
# lvremove -f vg/lvoll4
Automatically removing historical logical volume vg/-lvoll1.
Automatically removing historical logical volume vg/-lvoll2.
Automatically removing historical logical volume vg/-lvoll4.
Logical volume "lvoll4" successfully removed
```

## CAPÍTULO 13. ACTIVACIÓN DE LA CACHÉ PARA MEJORAR EL RENDIMIENTO DEL VOLUMEN LÓGICO

Puedes añadir caché a un volumen lógico LVM para mejorar el rendimiento. LVM almacena en caché las operaciones de E/S en el volumen lógico utilizando un dispositivo rápido, como un SSD.

Los siguientes procedimientos crean un LV especial del dispositivo rápido, y adjuntan este LV especial al LV original para mejorar el rendimiento.

### 13.1. MÉTODOS DE ALMACENAMIENTO EN CACHÉ EN LVM

LVM proporciona los siguientes tipos de caché. Cada uno es adecuado para diferentes tipos de patrones de E/S en el volumen lógico.

#### **dm-cache**

Este método acelera el acceso a los datos de uso frecuente al almacenarlos en caché en el volumen más rápido. El método almacena en caché tanto las operaciones de lectura como las de escritura.

El método **dm-cache** crea volúmenes lógicos del tipo **cache**.

#### **dm-writecache**

Este método sólo almacena en caché las operaciones de escritura. El volumen más rápido almacena las operaciones de escritura y luego las migra al disco más lento en segundo plano. El volumen más rápido suele ser un SSD o un disco de memoria persistente (PMEM).

El método **dm-writecache** crea volúmenes lógicos del tipo **writecache**.

### 13.2. COMPONENTES DE CACHÉ LVM

Cuando se habilita el almacenamiento en caché para un volumen lógico, LVM renombra y oculta los volúmenes originales, y presenta un nuevo volumen lógico que está compuesto por los volúmenes lógicos originales. La composición del nuevo volumen lógico depende del método de almacenamiento en caché y de si se utiliza la opción **cachevol** o **cachepool**.

Las opciones **cachevol** y **cachepool** exponen diferentes niveles de control sobre la colocación de los componentes de la caché:

- Con la opción **cachevol**, el dispositivo más rápido almacena tanto las copias en caché de los bloques de datos como los metadatos para gestionar la caché.
- Con la opción **cachepool**, dispositivos separados pueden almacenar las copias en caché de los bloques de datos y los metadatos para gestionar la caché.

El método **dm-writecache** no es compatible con **cachepool**.

En todas las configuraciones, LVM expone un único dispositivo resultante, que agrupa todos los componentes de la caché. El dispositivo resultante tiene el mismo nombre que el volumen lógico lento original.

### 13.3. ACTIVACIÓN DE LA CACHÉ DM-CACHE PARA UN VOLUMEN LÓGICO

Este procedimiento permite almacenar en caché los datos más utilizados en un volumen lógico mediante el método **dm-cache**.

## Requisitos previos

- En su sistema existe un volumen lógico lento que desea acelerar utilizando **dm-cache**.
- El grupo de volumen que contiene el volumen lógico lento también contiene un volumen físico no utilizado en un dispositivo de bloque rápido.

## Procedimiento

1. Cree un volumen **cachevol** en el dispositivo rápido:

```
# lvcreate --size cachevol-size --name fastvol vg /dev/fast-pv
```

Sustituya los siguientes valores:

### ***cachevol-size***

El tamaño del volumen de **cachevol**, como **5G**

### ***fastvol***

Un nombre para el volumen **cachevol**

### ***vg***

El nombre del grupo de volumen

### ***/dev/fast-pv***

La ruta de acceso al dispositivo de bloque rápido, como **/dev/sdf1**

2. Adjunte el volumen **cachevol** al volumen lógico principal para comenzar el almacenamiento en caché:

```
# lvconvert --type cache --cachevol fastvol vg /main-lv
```

Sustituya los siguientes valores:

### ***fastvol***

El nombre del volumen **cachevol**

### ***vg***

El nombre del grupo de volumen

### ***main-lv***

El nombre del volumen lógico lento

## Pasos de verificación

- Examine los dispositivos recién creados:

```
# lvs --all --options +devices vg

LV          Pool          Type  Devices
main-lv     [fastvol_cvol] cache  main-lv_corig(0)
[fastvol_cvol]          linear /dev/fast-pv
[main-lv_corig]          linear /dev/slow-pv
```

## Recursos adicionales

- Para obtener información sobre este procedimiento y otros detalles, incluyendo ejemplos administrativos, consulte la página man **lvmcache(7)**.

## 13.4. ACTIVACIÓN DE LA CACHÉ DM-CACHE CON UN CACHEPOOL PARA UN VOLUMEN LÓGICO

Este procedimiento permite crear los volúmenes lógicos de datos de caché y de metadatos de caché individualmente y luego combinar los volúmenes en un pool de caché.

### Requisitos previos

- En su sistema existe un volumen lógico lento que desea acelerar utilizando **dm-cache**.
- El grupo de volumen que contiene el volumen lógico lento también contiene un volumen físico no utilizado en un dispositivo de bloque rápido.

### Procedimiento

1. Cree un volumen **cachepool** en el dispositivo rápido:

```
# lvcreate --type cache-pool --size cachepool-size --name fastpool vg /dev/fast
```

Sustituya los siguientes valores:

#### ***cachepool-size***

El tamaño de la **cachepool**, como **5G**

#### ***fastpool***

Un nombre para el volumen **cachepool**

#### ***vg***

El nombre del grupo de volumen

#### ***/dev/fast***

La ruta de acceso al dispositivo de bloque rápido, como **/dev/sdf1**



### NOTA

Puede utilizar la opción **--poolmetadata** para especificar la ubicación de los metadatos del pool al crear el cache-pool.

2. Adjunte el **cachepool** al volumen lógico principal para comenzar a almacenar en caché:

```
# lvconvert --type cache --cachepool fastpool vg/main
```

Sustituya los siguientes valores:

#### ***fastpool***

El nombre del volumen **cachepool**

#### ***vg***

El nombre del grupo de volumen

#### ***main***

El nombre del volumen lógico lento

### Pasos de verificación

- Examine los dispositivos recién creados:

```
# lvs --all --options +devices vg

LV          Pool          Type    Devices
[fastpool_cpoo]          cache-pool fastpool_pool_cdata(0)
[fastpool_cpoo_cdata]          linear   /dev/sdf1(4)
[fastpool_cpoo_cmeta]          linear   /dev/sdf1(2)
[lvol0_pmspare]          linear   /dev/sdf1(0)
main        [fastpool_cpoo] cache      main_corig(0)
[main_corig]          linear   /dev/sdf1(0)
```

### Recursos adicionales

- La página de manual **lvcreate(8)**.
- La página de manual **lvocache(7)**.
- La página de manual **lvconvert(8)**.

## 13.5. ACTIVACIÓN DE LA CACHÉ DM-WRITECACHE PARA UN VOLUMEN LÓGICO

Este procedimiento permite almacenar en caché las operaciones de E/S de escritura en un volumen lógico utilizando el método **dm-writecache**.

### Requisitos previos

- En su sistema existe un volumen lógico lento que desea acelerar utilizando **dm-writecache**.
- El grupo de volumen que contiene el volumen lógico lento también contiene un volumen físico no utilizado en un dispositivo de bloque rápido.

### Procedimiento

1. Si el volumen lógico lento está activo, desactívalo:

```
# lvchange --activate n vg/main-lv
```

Sustituya los siguientes valores:

**vg**

El nombre del grupo de volumen

**main-lv**

El nombre del volumen lógico lento

2. Cree un volumen **cachevol** desactivado en el dispositivo rápido:

```
# lvcreate --activate n --tamaño cachevol-size --nombre fastvol vg /dev/fast-pv
```

-

Sustituya los siguientes valores:

**cachevol-size**

El tamaño del volumen de **cachevol**, como **5G**

**fastvol**

Un nombre para el volumen **cachevol**

**vg**

El nombre del grupo de volumen

**/dev/fast-pv**

La ruta de acceso al dispositivo de bloque rápido, como **/dev/sdf1**

3. Adjunte el volumen **cachevol** al volumen lógico principal para comenzar el almacenamiento en caché:

```
# lvconvert --type writecache --cachevol fastvol vg /main-lv
```

Sustituya los siguientes valores:

**fastvol**

El nombre del volumen **cachevol**

**vg**

El nombre del grupo de volumen

**main-lv**

El nombre del volumen lógico lento

4. Active el volumen lógico resultante:

```
# lvchange --activate y vg/main-lv
```

Sustituya los siguientes valores:

**vg**

El nombre del grupo de volumen

**main-lv**

El nombre del volumen lógico lento

## Pasos de verificación

- Examine los dispositivos recién creados:

```
# lvs --all --options +devices vg
LV          VG Attr  LSize Pool           Origin        Data% Meta% Move Log
Cpy%Sync Convert Devices
main-lv     vg Cwi-a-C--- 500.00m [fastvol_cvol] [main-lv_worig] 0.00
main-lv_worig(0)
[fastvol_cvol] vg Cwi-aoC--- 252.00m
```

```
/dev/sdc1(0)
[main-lv_wcorig] vg owi-aoC--- 500.00m
/dev/sdb1(0)
```

### Recursos adicionales

- Para obtener información, incluyendo ejemplos administrativos, consulte la página man **lvmdcache(7)**.

## 13.6. DESACTIVACIÓN DEL ALMACENAMIENTO EN CACHÉ DE UN VOLUMEN LÓGICO

Este procedimiento desactiva el almacenamiento en caché de **dm-cache** o **dm-writetocache** que está actualmente activado en un volumen lógico.

### Requisitos previos

- El almacenamiento en caché está activado en un volumen lógico.

### Procedimiento

1. Desactivar el volumen lógico:

```
# lvchange --activate n vg/main-lv
```

Sustituya los siguientes valores:

**vg**

El nombre del grupo de volumen

**main-lv**

El nombre del volumen lógico en el que se ha habilitado el almacenamiento en caché

2. Separe el volumen **cachevol** o **cachepool**:

```
# lvconvert --splitcache vg/main-lv
```

Sustituya los siguientes valores:

**vg**

El nombre del grupo de volumen

**main-lv**

El nombre del volumen lógico en el que se ha habilitado el almacenamiento en caché

### Pasos de verificación

- Compruebe que los volúmenes lógicos ya no están unidos:

```
# lvs --all --options +devices [replaceable]_vg_
LV   Attr   Type  Devices
```

```
fastvol -wi----- linear /dev/fast-pv  
main-lv -wi----- linear /dev/slow-pv
```

### Recursos adicionales

- La página de manual **lvmetache(7)**

## CAPÍTULO 14. ACTIVACIÓN DEL VOLUMEN LÓGICO

Un volumen lógico en estado activo puede ser utilizado a través de un dispositivo de bloque. Un volumen lógico que está activado es accesible y está sujeto a cambios. Cuando se crea un volumen lógico se activa por defecto.

Hay varias circunstancias por las que necesitas hacer que un volumen lógico individual esté inactivo y por lo tanto sea desconocido para el kernel. Puede activar o desactivar un volumen lógico individual con la opción **-a** del comando **lvchange**.

El formato del comando para desactivar un volumen lógico individual es el siguiente.

```
lvchange -un vg/lv
```

El formato del comando para activar un volumen lógico individual es el siguiente.

```
lvchange -ay vg/lv
```

Puedes activar o desactivar todos los volúmenes lógicos de un grupo de volúmenes con la opción **-a** del comando **vgchange**. Esto es el equivalente a ejecutar el comando **lvchange -a** en cada volumen lógico individual del grupo de volúmenes.

El formato del comando para desactivar todos los volúmenes lógicos de un grupo de volúmenes es el siguiente.

```
vgchange -an vg
```

El formato del comando para activar todos los volúmenes lógicos de un grupo de volúmenes es el siguiente.

```
vgchange -ay vg
```

### 14.1. CONTROL DE LA AUTOACTIVACIÓN DE LOS VOLÚMENES LÓGICOS

La autoactivación de un volumen lógico se refiere a la activación automática basada en eventos de un volumen lógico durante el inicio del sistema. A medida que los dispositivos están disponibles en el sistema (eventos de dispositivos en línea), **systemd/udev** ejecuta el servicio **lvm2-pvscan** para cada dispositivo. Este servicio ejecuta el comando **pvscan --cache -aay device** que lee el dispositivo nombrado. Si el dispositivo pertenece a un grupo de volúmenes, el comando **pvscan** comprobará si todos los volúmenes físicos de ese grupo de volúmenes están presentes en el sistema. Si es así, el comando activará los volúmenes lógicos de ese grupo de volúmenes.

Puede utilizar las siguientes opciones de configuración en el archivo de configuración **/etc/lvm/lvm.conf** para controlar la autoactivación de los volúmenes lógicos.

- **global/event\_activation**  
Cuando **event\_activation** está deshabilitado, **systemd/udev** autoactivará el volumen lógico sólo en los volúmenes físicos que estén presentes durante el arranque del sistema. Si aún no han aparecido todos los volúmenes físicos, es posible que algunos volúmenes lógicos no se autoactiven.
- **activation/auto\_activation\_volume\_list**

Si se establece **auto\_activation\_volume\_list** en una lista vacía, se desactiva la autoactivación por completo. Si se establece **auto\_activation\_volume\_list** en volúmenes lógicos y grupos de volúmenes específicos, se limita la autoactivación a esos volúmenes lógicos.

Para obtener información sobre la configuración de estas opciones, consulte el archivo de configuración **/etc/lvm/lvm.conf**.

## 14.2. CONTROL DE LA ACTIVACIÓN DEL VOLUMEN LÓGICO

Puede controlar la activación del volumen lógico de las siguientes maneras:

- A través de la configuración de **activation/volume\_list** en el archivo **/etc/lvm/conf**. Esto permite especificar qué volúmenes lógicos se activan. Para obtener información sobre el uso de esta opción, consulte el archivo de configuración **/etc/lvm/lvm.conf**.
- Mediante el indicador de omisión de activación para un volumen lógico. Cuando se establece este indicador para un volumen lógico, el volumen se omite durante los comandos de activación normales.

Puede establecer el indicador de omisión de activación en un volumen lógico de las siguientes maneras.

- Puede desactivar el indicador de salto de activación al crear un volumen lógico especificando la opción **-kn** o **--setactivationskip n** del comando **lvcreate**.
- Puede desactivar el indicador de salto de activación para un volumen lógico existente especificando la opción **-kn** o **--setactivationskip n** del comando **lvchange**.
- Puede volver a activar el indicador de salto de activación para un volumen en el que se haya desactivado con la opción **-ky** o **--setactivationskip y** del comando **lvchange**.

Para determinar si el indicador de omisión de activación está establecido para un volumen lógico, ejecute el comando **lvs**, que muestra el atributo **k** como en el siguiente ejemplo.

```
# lvs vg/thin1s1
LV      VG Attr   LSize Pool Origin
thin1s1  vg Vwi---tz-k 1.00t pool0 thin1
```

Puede activar un volumen lógico con el atributo **k** establecido utilizando la opción **-K** o **--ignoreactivationskip** además de la opción estándar **-ay** o **--activate y**.

Por defecto, los volúmenes finos de instantáneas se marcan para omitir la activación cuando se crean. Puede controlar el ajuste de omisión de activación por defecto en los nuevos volúmenes ligeros de instantánea con el ajuste **auto\_set\_activation\_skip** en el archivo **/etc/lvm/lvm.conf**.

El siguiente comando activa un volumen lógico de instantánea delgada que tiene el indicador de omisión de activación establecido.

```
# lvchange -ay -K VG/SnapLV
```

El siguiente comando crea una instantánea fina sin el indicador de omisión de activación

```
# lvcreate --type thin -n SnapLV -kn -s ThinLV --thinpool VG/ThinPoolLV
```

El siguiente comando elimina el indicador de omisión de activación de un volumen lógico de instantánea.

# **lvchange -kn VG/SnapLV**

### 14.3. ACTIVACIÓN DE VOLÚMENES LÓGICOS COMPARTIDOS

Puedes controlar la activación de un volumen lógico compartido con la opción **-a** de los comandos **lvchange** y **vgchange**, de la siguiente manera.

Comando	Activación
<b>lvchange -ay e</b>	Activar el volumen lógico compartido en modo exclusivo, permitiendo que sólo un único host active el volumen lógico. Si la activación falla, como ocurriría si el volumen lógico está activo en otro host, se informa de un error.
<b>lvchange -asy</b>	Activar el volumen lógico compartido en modo compartido, permitiendo que varios hosts activen el volumen lógico simultáneamente. Si la activación falla, como sucedería si el volumen lógico está activo exclusivamente en otro host, se informa de un error. Si el tipo lógico prohíbe el acceso compartido, como una instantánea, el comando informará de un error y fallará. Los tipos de volúmenes lógicos que no pueden ser utilizados simultáneamente desde múltiples hosts incluyen thin, cache, raid y snapshot.
<b>lvchange -an</b>	Desactivar el volumen lógico.

### 14.4. ACTIVACIÓN DE UN VOLUMEN LÓGICO CON DISPOSITIVOS AUSENTES

Puede configurar qué volúmenes lógicos con dispositivos ausentes se activan estableciendo el parámetro **activation\_mode** con el comando **lvchange** a uno de los siguientes valores.

Modo de activación	Significado
completa	Permite activar sólo los volúmenes lógicos sin volúmenes físicos ausentes. Este es el modo más restrictivo.
degradado	Permite activar volúmenes lógicos RAID con volúmenes físicos ausentes.
parcial	Permite activar cualquier volumen lógico al que le falten volúmenes físicos. Esta opción debe utilizarse únicamente para la recuperación o reparación.

El valor por defecto de **activation\_mode** está determinado por la configuración de **activation\_mode** en el archivo **/etc/lvm/lvm.conf**. Para más información, consulte la página de manual **lvraid(7)**.

## CAPÍTULO 15. CONTROL DE LA EXPLORACIÓN DE DISPOSITIVOS LVM

Puedes controlar el escaneo de dispositivos LVM configurando filtros en el archivo `/etc/lvm/lvm.conf`. Los filtros en el archivo `lvm.conf` consisten en una serie de expresiones regulares simples que se aplican a los nombres de los dispositivos en el directorio `/dev` para decidir si se acepta o rechaza cada dispositivo de bloque encontrado.

### 15.1. EL FILTRO DE DISPOSITIVOS LVM

Las herramientas LVM escanean los dispositivos en el directorio `/dev` y comprueban cada uno de ellos en busca de metadatos LVM. Un filtro en el archivo `/etc/lvm/lvm.conf` controla qué dispositivos escanea LVM.

El filtro es una lista de patrones que LVM aplica a cada dispositivo encontrado por una exploración del directorio `/dev`, o el directorio especificado por la palabra clave `dir` en el archivo `/etc/lvm/lvm.conf`. Los patrones son expresiones regulares delimitadas por cualquier carácter y precedidas por `a` para *accept* o `r` para *reject*. La primera expresión regular de la lista que coincide con un dispositivo determina si LVM acepta o rechaza (ignora) el dispositivo. LVM acepta los dispositivos que no coinciden con ningún patrón.

La siguiente es la configuración por defecto del filtro, que escanea todos los dispositivos:

```
filtro = [ \ "a/*" ]
```

### 15.2. EJEMPLOS DE CONFIGURACIONES DE FILTROS DE DISPOSITIVOS LVM

Los siguientes ejemplos muestran el uso de filtros para controlar qué dispositivos escanea LVM.



#### AVISO

Algunos de los ejemplos presentados aquí podrían coincidir involuntariamente con dispositivos adicionales en el sistema y podrían no representar la práctica recomendada para su sistema. Por ejemplo, `a/loop/` es equivalente a `a.*loop.*/` y coincidiría con `/dev/solooperation/lvol1`.

- El siguiente filtro añade todos los dispositivos descubiertos, que es el comportamiento por defecto porque no se ha configurado ningún filtro en el archivo de configuración:

```
filtro = [ \ "a/*" ]
```

- El siguiente filtro elimina el dispositivo `cdrom` para evitar retrasos si la unidad no contiene medios:

```
filter = [ "r!^/dev/cdrom$" ]
```

- El siguiente filtro añade todos los dispositivos de bucle y elimina todos los demás dispositivos de bloque:

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

- El siguiente filtro añade todos los dispositivos de bucle e IDE y elimina todos los demás dispositivos de bloque:

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

- El siguiente filtro añade sólo la partición 8 en la primera unidad IDE y elimina todos los demás dispositivos de bloque:

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

### 15.3. APLICACIÓN DE UNA CONFIGURACIÓN DE FILTRO DE DISPOSITIVOS LVM

Este procedimiento cambia la configuración del filtro de dispositivos LVM, que controla los dispositivos que escanea LVM.

#### Requisitos previos

- Prepare el patrón de filtro del dispositivo que desea utilizar.

#### Procedimiento

1. Pruebe el patrón de filtrado de su dispositivo sin modificar el archivo `/etc/lvm/lvm.conf`. Utilice un comando LVM con la opción `--config 'devices{ filter = [ your device filter pattern ] }'` opción. Por ejemplo:

```
# lvs --config 'devices{ filter = [ !"a|dev/emcpower.*|", "r.*" ]'
```

2. Edite la opción **filter** en el archivo de configuración `/etc/lvm/lvm.conf` para utilizar su nuevo patrón de filtrado de dispositivos.
3. Compruebe que no faltan volúmenes físicos o grupos de volúmenes que desee utilizar con la nueva configuración:

```
# pvscan
```

```
# vgscan
```

4. Reconstruya el sistema de archivos **initramfs** para que LVM escanee sólo los dispositivos necesarios al reiniciar:

```
# dracut --force --verbose
```

## CAPÍTULO 16. ESTRATIFICACIÓN DE VOLÚMENES FÍSICOS LVM SOBRE VOLÚMENES LÓGICOS

Puedes configurar LVM para que sea posible crear volúmenes físicos sobre volúmenes lógicos.

Por defecto, los comandos LVM no escanean los volúmenes lógicos de su sistema. Este comportamiento por defecto proporciona las siguientes ventajas:

- Si hay muchos volúmenes lógicos activos en el sistema, cada comando LVM requeriría un tiempo adicional, impactando negativamente en el rendimiento y causando retrasos o tiempos de espera no deseados.
- Si los volúmenes lógicos contienen volúmenes físicos de la imagen de una VM invitada, el host normalmente no quiere escanear o utilizar esos volúmenes físicos en capas que pertenecen al invitado. Tenga en cuenta, sin embargo, que en los casos en los que el volumen físico de una VM invitada existe directamente en un dispositivo SCSI visible para el host, para evitar que LVM en el host acceda a esos volúmenes físicos tendrá que configurar un filtro, como se describe en [Capítulo 15, Control de la exploración de dispositivos LVM](#).

El escaneo de volúmenes lógicos puede ser necesario cuando la superposición de volúmenes físicos sobre volúmenes lógicos es intencional. Esto permitirá ejecutar el comando **pvcreate** en un volumen lógico. Para configurar LVM para escanear todos los volúmenes lógicos, establezca la opción de configuración **scan\_lvs** en el archivo **/etc/lvm/lvm.conf** a **scan\_lvs=1**. Para restringir los volúmenes lógicos que los comandos LVM escanean, puedes configurar filtros de dispositivo en el archivo de configuración **/etc/lvm/lvm.conf**, como se describe en [Capítulo 15, Control de la exploración de dispositivos LVM](#).

## CAPÍTULO 17. CONTROL DE LA ASIGNACIÓN DE LVM

Por defecto, un grupo de volúmenes asigna extensiones físicas de acuerdo a reglas de sentido común como no colocar franjas paralelas en el mismo volumen físico. Esta es la política de asignación **normal**. Puedes usar el argumento **--alloc** del comando **vgcreate** para especificar una política de asignación de **contiguous**, **anywhere**, o **cling**. En general, las políticas de asignación diferentes a **normal** son requeridas sólo en casos especiales donde se necesita especificar una asignación de extensión inusual o no estándar.

### 17.1. POLÍTICAS DE ASIGNACIÓN DE LVM

Cuando una operación LVM necesita asignar extensiones físicas para uno o más volúmenes lógicos, la asignación procede como sigue:

- Se genera el conjunto completo de extensiones físicas no asignadas en el grupo de volúmenes para su consideración. Si proporciona cualquier rango de extensiones físicas al final de la línea de comandos, sólo se considerarán las extensiones físicas no asignadas dentro de esos rangos en los volúmenes físicos especificados.
- Cada política de asignación se intenta por turnos, empezando por la política más estricta (**contiguous**) y terminando con la política de asignación especificada mediante la opción **--alloc** o establecida por defecto para el volumen lógico o grupo de volúmenes concreto. Para cada política, trabajando desde la extensión lógica más baja del espacio del volumen lógico vacío que necesita ser llenado, se asigna tanto espacio como sea posible, de acuerdo con las restricciones impuestas por la política de asignación. Si se necesita más espacio, LVM pasa a la siguiente política.

Las restricciones de la política de asignación son las siguientes:

- Una política de asignación de **contiguous** requiere que la ubicación física de cualquier extensión lógica que no sea la primera extensión lógica de un volumen lógico sea adyacente a la ubicación física de la extensión lógica inmediatamente anterior.  
Cuando un volumen lógico está dividido en franjas o en espejo, la restricción de asignación **contiguous** se aplica independientemente a cada franja o imagen en espejo (pierna) que necesite espacio.
- Una política de asignación de **cling** requiere que el volumen físico utilizado para cualquier extensión lógica se añada a un volumen lógico existente que ya esté en uso por al menos una extensión lógica anterior en ese volumen lógico. Si se define el parámetro de configuración **allocation/cling\_tag\_list**, se considera que dos volúmenes físicos coinciden si alguna de las etiquetas enumeradas está presente en ambos volúmenes físicos. Esto permite que grupos de volúmenes físicos con propiedades similares (como su ubicación física) sean etiquetados y tratados como equivalentes a efectos de asignación.  
Cuando un Volumen Lógico es rayado o reflejado, la restricción de asignación **cling** se aplica independientemente a cada raya o imagen reflejada (pierna) que necesite espacio.
- Una política de asignación de **normal** no elegirá una extensión física que comparta el mismo volumen físico que una extensión lógica ya asignada a un volumen lógico paralelo (es decir, una franja o imagen espejo diferente) en el mismo desplazamiento dentro de ese volumen lógico paralelo.  
Cuando se asigna un registro en espejo al mismo tiempo que volúmenes lógicos para contener los datos en espejo, una política de asignación de **normal** intentará primero seleccionar volúmenes físicos diferentes para el registro y los datos. Si eso no es posible y el parámetro de configuración **allocation/mirror\_logs\_require\_separate\_pvs** se establece en 0, entonces permitirá que el registro comparta volumen(s) físico(s) con parte de los datos.

Del mismo modo, al asignar los metadatos del thin pool, una política de asignación de **normal** seguirá las mismas consideraciones que para la asignación de un mirror log, basándose en el valor del parámetro de configuración **allocation/thin\_pool\_metadata\_require\_separate\_pvs**.

- Si hay suficientes extensiones libres para satisfacer una solicitud de asignación pero una política de asignación de **normal** no las utilizaría, la política de asignación de **anywhere** lo hará, incluso si eso reduce el rendimiento al colocar dos franjas en el mismo volumen físico.

Las políticas de asignación pueden modificarse mediante el comando **vgchange**.



## NOTA

Si confía en cualquier comportamiento de distribución más allá de lo documentado en esta sección según las políticas de asignación definidas, debe tener en cuenta que esto podría cambiar en futuras versiones del código. Por ejemplo, si proporcionas en la línea de comandos dos volúmenes físicos vacíos que tienen un número idéntico de extensiones físicas libres disponibles para la asignación, LVM actualmente considera usar cada uno de ellos en el orden en que están listados; no hay garantía de que las futuras versiones mantengan esa propiedad. Si es importante obtener un diseño específico para un Volumen Lógico en particular, entonces deberías construirlo a través de una secuencia de pasos de **lvcreate** y **lvconvert** de tal manera que las políticas de asignación aplicadas a cada paso no dejen a LVM ninguna discreción sobre el diseño.

Para ver cómo funciona actualmente el proceso de asignación en cualquier caso específico, puede leer la salida del registro de depuración, por ejemplo, añadiendo la opción **-vvvv** a un comando.

## 17.2. IMPEDIR LA ASIGNACIÓN EN UN VOLUMEN FÍSICO

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

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

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

También puede utilizar los argumentos de **-xy** del comando **pvchange** para permitir la asignación donde antes no se permitía.

## 17.3. AMPLIACIÓN DE UN VOLUMEN LÓGICO CON LA POLÍTICA DE ASIGNACIÓN CLING

Cuando se amplía un volumen LVM, se puede utilizar la opción **--alloc cling** del comando **lvextend** para especificar la política de asignación **cling**. Esta política elegirá espacio en los mismos volúmenes físicos que el último segmento del volumen lógico existente. Si no hay suficiente espacio en los volúmenes físicos y se define una lista de etiquetas en el archivo **/etc/lvm/lvm.conf**, LVM comprobará si alguna de las etiquetas está unida a los volúmenes físicos y tratará de hacer coincidir esas etiquetas de volúmenes físicos entre los extensiones existentes y las nuevas extensiones.

Por ejemplo, si tiene volúmenes lógicos que se reflejan entre dos sitios dentro de un mismo grupo de volúmenes, puede etiquetar los volúmenes físicos según su ubicación etiquetando los volúmenes físicos con las etiquetas **@site1** y **@site2**. Entonces puede especificar la siguiente línea en el archivo **lvm.conf**:

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

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

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

También en este ejemplo, se ha creado un grupo de volúmenes **taft** que consta de los volúmenes físicos **/dev/sdb1**, **/dev/sdc1**, **/dev/sdd1**, **/dev/sde1**, **/dev/sdf1**, **/dev/sdg1**, y **/dev/sdh1**. Estos volúmenes físicos han sido etiquetados con las etiquetas **A**, **B**, y **C**. El ejemplo no utiliza la etiqueta **C**, pero esto mostrará que LVM utiliza las etiquetas para seleccionar qué volúmenes físicos utilizar para las patas de la réplica.

```
# pvs -a -o +pv_tags /dev/sd[bcdefgh]
PV      VG  Fmt Attr PSize PFree PV Tags
/dev/sdb1 taft lvm2 a-- 15.00g 15.00g A
/dev/sdc1 taft lvm2 a-- 15.00g 15.00g B
/dev/sdd1 taft lvm2 a-- 15.00g 15.00g B
/dev/sde1 taft lvm2 a-- 15.00g 15.00g C
/dev/sdf1 taft lvm2 a-- 15.00g 15.00g C
/dev/sdg1 taft lvm2 a-- 15.00g 15.00g A
/dev/sdh1 taft lvm2 a-- 15.00g 15.00g A
```

El siguiente comando crea un volumen duplicado de 10 gigabytes a partir del grupo de volúmenes **taft**.

```
# lvcreate --type raid1 -m 1 -n mirror --nosync -L 10G taft
WARNING: New raid1 won't be synchronised. Don't read what you didn't write!
Logical volume "mirror" created
```

El siguiente comando muestra qué dispositivos se utilizan para los tramos de réplica y los subvolúmenes de metadatos RAID.

```
# lvs -a -o +devices
LV      VG  Attr      LSize Log Cpy%Sync Devices
mirror  taft Rwi-a-r--- 10.00g 100.00 mirror_rimage_0(0),mirror_rimage_1(0)
[mirror_rimage_0] taft iwi-aor--- 10.00g /dev/sdb1(1)
[mirror_rimage_1] taft iwi-aor--- 10.00g /dev/sdc1(1)
[mirror_rmeta_0] taft ewi-aor--- 4.00m /dev/sdb1(0)
[mirror_rmeta_1] taft ewi-aor--- 4.00m /dev/sdc1(0)
```

El siguiente comando amplía el tamaño del volumen reflejado, utilizando la política de asignación **cling** para indicar que los tramos reflejados deben ampliarse utilizando volúmenes físicos con la misma etiqueta.

```
# lvextend --alloc cling -L +10G taft/mirror
Extending 2 mirror images.
Extending logical volume mirror to 20.00 GiB
Logical volume mirror successfully resized
```

El siguiente comando de visualización muestra que los tramos de réplica se han ampliado utilizando volúmenes físicos con la misma etiqueta que el tramo. Tenga en cuenta que los volúmenes físicos con una etiqueta de **C** fueron ignorados.

```
# lvs -a -o +devices
```

```

LV          VG Attr   LSize Log Cpy%Sync Devices
mirror     taft Rwi-a-r--- 20.00g 100.00 mirror_rimage_0(0),mirror_rimage_1(0)
[mirror_rimage_0] taft iwi-aor--- 20.00g /dev/sdb1(1)
[mirror_rimage_0] taft iwi-aor--- 20.00g /dev/sdg1(0)
[mirror_rimage_1] taft iwi-aor--- 20.00g /dev/sdc1(1)
[mirror_rimage_1] taft iwi-aor--- 20.00g /dev/sdd1(0)
[mirror_rmeta_0] taft ewi-aor--- 4.00m /dev/sdb1(0)
[mirror_rmeta_1] taft ewi-aor--- 4.00m /dev/sdc1(0)

```

## 17.4. DIFERENCIACIÓN DE LOS OBJETOS RAID DE LVM MEDIANTE ETIQUETAS

Puede asignar etiquetas a los objetos RAID de LVM para agruparlos, de modo que pueda automatizar el control del comportamiento de los RAID de LVM, como la activación, por grupo.

Las etiquetas de volumen físico (PV) son las responsables del control de la asignación en el raid LVM, a diferencia de las etiquetas de volumen lógico (LV) o de grupo de volúmenes (VG), ya que la asignación en lvm se produce a nivel de PV basándose en las políticas de asignación. Para distinguir los tipos de almacenamiento por sus diferentes propiedades, etiquételos apropiadamente (por ejemplo, NVMe, SSD, HDD). Red Hat recomienda que etiquete cada nuevo PV apropiadamente después de agregarlo a un VG.

Este procedimiento añade etiquetas de objeto a sus volúmenes lógicos, asumiendo que **/dev/sda** es un SSD, y **/dev/sd[b-f]** son HDDs con una partición.

### Requisitos previos

- El paquete **lvm2** está instalado.
- Existen dispositivos de almacenamiento para utilizar como FV.

### Procedimiento

1. Crea un grupo de volúmenes.

```
# vgcreate MyVG /dev/sd[a-f]1
```

2. Añade etiquetas a tus volúmenes físicos.

```
# pvchange --addtag ssds /dev/sda1
```

```
# pvchange --addtag hdds /dev/sd[b-f]1
```

3. Cree un volumen lógico RAID6.

```
# lvcreate --type raid6 --stripes 3 -L1G -nr6 MyVG @hdds
```

4. Cree un volumen de reserva de caché lineal.

```
# lvcreate -nr6pool -L512m MyVG @ssds
```

5. Convierte el volumen RAID6 en caché.

```
# lvconvert --type cache --cachevol MyVG/r6pool MyVG/r6
```

### Recursos adicionales

- Las páginas de manual **lvcreate(8)**, **lvconvert(8)**, **lvraid(7)** y **lvncache(7)**.

## CAPÍTULO 18. AGRUPACIÓN DE OBJETOS LVM CON ETIQUETAS

Como administrador del sistema, puede asignar etiquetas a los objetos LVM para agruparlos, de modo que pueda automatizar el control del comportamiento de LVM, como la activación, por grupo.

### 18.1. ETIQUETAS DE OBJETOS LVM

Una etiqueta LVM es una palabra que se utiliza para agrupar objetos LVM2 del mismo tipo. Las etiquetas se adjuntan a objetos como volúmenes físicos, grupos de volúmenes y volúmenes lógicos, así como a hosts en una configuración de clúster.

Las etiquetas se dan en la línea de comandos en lugar de los argumentos PV, VG o LV. Las etiquetas deben ir precedidas de @ para evitar ambigüedades. Cada etiqueta se expande sustituyéndola por todos los objetos que posean esa etiqueta y que sean del tipo esperado por su posición en la línea de comandos.

Las etiquetas LVM son cadenas de hasta 1024 caracteres. Las etiquetas LVM no pueden comenzar con un guión.

Una etiqueta válida está formada únicamente por un rango limitado de caracteres. Los caracteres permitidos son **A-Z a-z 0-9 \_ . - / = ! : # &**.

Sólo los objetos de un grupo de volúmenes pueden ser etiquetados. Los volúmenes físicos pierden sus etiquetas si se eliminan de un grupo de volúmenes; esto se debe a que las etiquetas se almacenan como parte de los metadatos del grupo de volúmenes y eso se elimina cuando se elimina un volumen físico.

### 18.2. LISTADO DE ETIQUETAS LVM

El siguiente ejemplo muestra cómo listar las etiquetas LVM.

#### Procedimiento

- Utilice el siguiente comando para listar todos los volúmenes lógicos con la etiqueta **database**:

```
# lvs @database
```

- Utilice el siguiente comando para listar las etiquetas de host actualmente activas:

```
# etiquetas lvm
```

### 18.3. AÑADIR ETIQUETAS DE OBJETOS LVM

Este procedimiento describe cómo añadir etiquetas de objetos LVM.

#### Requisitos previos

- El paquete **lvm2** está instalado.
- Se crean uno o más volúmenes físicos, grupos de volúmenes o volúmenes lógicos.

#### Procedimiento

- Para crear una etiqueta de objeto, añada la opción **--addtag** a un comando LVM:
  - Para crear etiquetas a partir de volúmenes físicos, añada la opción al comando **pvchange**.
  - Para crear etiquetas a partir de grupos de volumen, añada la opción a los comandos **vgchange** o **vgcreate**.
  - Para crear etiquetas a partir de volúmenes lógicos, añada la opción a los comandos **lvchange** o **lvcreate**.

## 18.4. ELIMINACIÓN DE ETIQUETAS DE OBJETOS LVM

Este procedimiento describe cómo eliminar las etiquetas de objetos LVM.

### Requisitos previos

- El paquete **lvm2** está instalado.
- Se crean etiquetas de objetos en volúmenes físicos, grupos de volúmenes o volúmenes lógicos.

### Procedimiento

- Para eliminar una etiqueta de objeto, añada la opción **--deltag** a un comando LVM:
  - Para eliminar las etiquetas de los volúmenes físicos, añada la opción al comando **pvchange**.
  - Para eliminar las etiquetas de los grupos de volumen, añada la opción a los comandos **vgchange** o **vgcreate**.
  - Para eliminar las etiquetas de los volúmenes lógicos, añada la opción a los comandos **lvchange** o **lvcreate**.

## 18.5. DEFINICIÓN DE ETIQUETAS DE HOST LVM

Este procedimiento describe cómo definir las etiquetas de host LVM en una configuración de cluster. Puede definir las etiquetas de host en los archivos de configuración.

### Procedimiento

- Establezca **hosttags = 1** en la sección **tags** para definir automáticamente la etiqueta de host utilizando el nombre de host de la máquina.  
Esto le permite utilizar un archivo de configuración común que puede ser replicado en todas sus máquinas para que tengan copias idénticas del archivo, pero el comportamiento puede diferir entre las máquinas según el nombre del host.

Para cada etiqueta de host, se lee un archivo de configuración adicional si existe: **lvm\_hosttag.conf**. Si ese archivo define nuevas etiquetas, se añadirán más archivos de configuración 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 del host es **host1**:

```
tags { tag1 { } tag2 { host_list = ["host1"] } }
```

## 18.6. CONTROL DE LA ACTIVACIÓN DE VOLÚMENES LÓGICOS CON ETIQUETAS

Este procedimiento describe cómo especificar en el archivo de configuración que sólo ciertos volúmenes lógicos deben ser activados en ese host.

### Requisitos previos

- Una lista con viñetas de las condiciones que deben cumplirse antes de que el usuario comience a seguir este montaje.
- También puede enlazar con otros módulos o montajes que el usuario debe seguir antes de iniciar este montaje.
- Borre el título de la sección y las viñetas si la asamblea no tiene requisitos previos.

### Procedimiento

Por ejemplo, la siguiente entrada actúa como un filtro para las solicitudes de activación (como **vgchange -ay**) y sólo activa **vg1/lvol0** y cualquier volumen lógico o grupo de volúmenes con la etiqueta **database** en los metadatos de ese host:

```
activación { volume_list = ["vg1/lvol0", "@database" ] }
```

La coincidencia especial **@\*** que causa una coincidencia sólo si cualquier etiqueta de metadatos coincide con cualquier etiqueta de host en esa máquina.

Como otro ejemplo, considere una situación en la que cada máquina en el clúster tiene la siguiente entrada en el archivo de configuración:

```
tags { hosttags = 1 }
```

Si quiere activar **vg1/lvol2** sólo en el host **db2**, haga lo siguiente:

1. Ejecute **lvchange --addtag @db2 vg1/lvol2** desde cualquier host del clúster.
2. Ejecutar **lvchange -ay vg1/lvol2**.

Esta solución consiste en almacenar los nombres de los hosts dentro de los metadatos del grupo de volúmenes.

## CAPÍTULO 19. SOLUCIÓN DE PROBLEMAS DE LVM

Puede utilizar las herramientas de LVM para solucionar una variedad de problemas en los volúmenes y grupos de LVM.

### 19.1. RECOGIDA DE DATOS DE DIAGNÓSTICO EN LVM

Si un comando LVM no está funcionando como se espera, puede reunir diagnósticos de las siguientes maneras.

#### Procedimiento

- Utilice los siguientes métodos para reunir diferentes tipos de datos de diagnóstico:
  - Añade el argumento **-vvvv** a cualquier comando LVM para aumentar el nivel de verbosidad de la salida del comando.
  - En la sección **log** del archivo de configuración **/etc/lvm/lvm.conf**, aumenta el valor de la opción **level**. Esto hace que LVM proporcione más detalles en el registro del sistema.
  - Si el problema está relacionado con la activación del volumen lógico, habilite a LVM para que registre los mensajes durante la activación:
    - i. Establezca la opción **activation = 1** en la sección **log** del archivo de configuración **/etc/lvm/lvm.conf**.
    - ii. Ejecute el comando LVM con la opción **-vvvv**.
    - iii. Examine la salida del comando.
    - iv. Restablece la opción **activation** en **0**.  
Si no restablece la opción a **0**, el sistema podría dejar de responder en situaciones de poca memoria.
  - Muestra un volcado de información con fines de diagnóstico:

```
# lvmdump
```

- Muestra información adicional del sistema:

```
# lvs -v
```

```
# pvs --all
```

```
# dmsetup info --columns
```

- Examine la última copia de seguridad de los metadatos LVM en el directorio **/etc/lvm/backup/** y las versiones archivadas en el directorio **/etc/lvm/archive/**.
- Compruebe la información de configuración actual:

```
# lvmconfig
```

- Compruebe el archivo de caché `/run/lvm/hints` para obtener un registro de los dispositivos que tienen volúmenes físicos en ellos.

## Recursos adicionales

- La página de manual `lvmdump(8)`

## 19.2. VISUALIZACIÓN DE INFORMACIÓN SOBRE DISPOSITIVOS LVM FALLIDOS

Puede mostrar información sobre un volumen LVM fallado que puede ayudarle a determinar por qué ha fallado el volumen.

### Procedimiento

- Visualice los volúmenes que han fallado utilizando la utilidad `vgs` o `lvs`.

#### Ejemplo 19.1. Grupos de volumen fallidos

En este ejemplo, uno de los dispositivos que componen el grupo de volumen `vg` ha fallado. El grupo de volúmenes está inutilizado pero puedes ver información sobre el dispositivo que ha fallado.

```
# vgs --options +devices

/dev/sdb: open failed: No such device or address
/dev/sdb: open failed: No such device or address
WARNING: Couldn't find device with uuid 42B7bu-YCMp-CEVD-CmKH-2rk6-fiO9-z1lf4s.
WARNING: VG vg is missing PV 42B7bu-YCMp-CEVD-CmKH-2rk6-fiO9-z1lf4s (last written to /dev/sdb1).
WARNING: Couldn't find all devices for LV vg/linear while checking used and assumed devices.
WARNING: Couldn't find all devices for LV vg/striped while checking used and assumed devices.
VG #PV #LV #SN Attr VSize VFree Devices
vg  2  2  0 wz-pn- <3.64t <3.60t [unknown](0)
vg  2  2  0 wz-pn- <3.64t <3.60t [unknown](5120),/dev/sdc1(0)
```

#### Ejemplo 19.2. LV lineal y rayado fallido

En este ejemplo, el dispositivo que ha fallado ha provocado el fallo de un volumen lógico lineal y otro rayado en el grupo de volúmenes. La salida del comando muestra los volúmenes lógicos que han fallado.

```
# lvs --all --options +devices

/dev/sdb: open failed: No such device or address
/dev/sdb: open failed: No such device or address
WARNING: Couldn't find device with uuid 42B7bu-YCMp-CEVD-CmKH-2rk6-fiO9-z1lf4s.
WARNING: VG vg is missing PV 42B7bu-YCMp-CEVD-CmKH-2rk6-fiO9-z1lf4s (last written to /dev/sdb1).
```

```
WARNING: Couldn't find all devices for LV vg/linear while checking used and assumed devices.
```

```
WARNING: Couldn't find all devices for LV vg/stripes while checking used and assumed devices.
```

```
LV   VG Attr   LSize Pool Origin Data% Meta% Move Log Cpy%Sync Convert
Devices
linear vg -wi-a---p- 20.00g                [unknown](0)
stripe vg -wi-a---p- 20.00g                [unknown]
(5120),/dev/sdc1(0)
```

### Ejemplo 19.3. Fallo en el tramo de un volumen lógico reflejado

Los siguientes ejemplos muestran la salida de comandos de las utilidades **vgs** y **lvs** cuando un tramo de un volumen lógico reflejado ha fallado.

```
# vgs --all --options +devices
```

```
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)
```

```
# lvs --all --options +devices
```

```
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)
```

## 19.3. ELIMINACIÓN DE VOLÚMENES FÍSICOS LVM PERDIDOS DE UN GRUPO DE VOLÚMENES

Si un volumen físico falla, puedes activar los volúmenes físicos restantes en el grupo de volúmenes y eliminar del grupo de volúmenes todos los volúmenes lógicos que utilizaban ese volumen físico.

### Procedimiento

1. Active los volúmenes físicos restantes en el grupo de volúmenes:

```
# vgchange --activate y --partial volume-group
```

2. Compruebe qué volúmenes lógicos se eliminarán:

```
# vgreduce --removemissing --test volume-group
```

3. Eliminar del grupo de volúmenes todos los volúmenes lógicos que utilizaban el volumen físico perdido:

```
# vgreduce --removemissing --force volume-group
```

- Opcional: Si ha eliminado accidentalmente volúmenes lógicos que quería conservar, puede revertir la operación **vgrestore**:

```
# vgcfgrestore volume-group
```



#### AVISO

Si has eliminado un thin pool, LVM no puede revertir la operación.

## 19.4. RECUPERACIÓN DE UN VOLUMEN FÍSICO LVM CON METADATOS DAÑADOS

Si el área de metadatos del grupo de volúmenes de un volumen físico se sobrescribe accidentalmente o se destruye de alguna manera, se obtiene un mensaje de error que indica que el área de metadatos es incorrecta, o que el sistema no pudo encontrar un volumen físico con un UUID particular. Es posible que puedas recuperar los datos del volumen físico reescribiendo el área de metadatos en el volumen físico.

### 19.4.1. Descubrir que un volumen LVM tiene metadatos perdidos o corruptos

El siguiente ejemplo muestra la salida del comando que podría ver si el área de metadatos de un volumen físico falta o está dañada.

#### Procedimiento

- Intenta listar los volúmenes lógicos:

```
# lvs --all --options devices
```

#### Ejemplo 19.4. Salida con metadatos faltantes o corruptos

En este ejemplo, algunos volúmenes lógicos se encuentran en un volumen físico al que le faltan metadatos o están corruptos.

```
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.
...
```

### 19.4.2. Encontrar los metadatos de un volumen físico LVM perdido

Este procedimiento encuentra los últimos metadatos archivados de un volumen físico que falta o está dañado.

## Procedimiento

1. Busca el archivo de metadatos del grupo de volúmenes que contiene el volumen físico.  
Los archivos de metadatos archivados se encuentran en la **/etc/lvm/archive/volume-group-name\_backup-number.vg** ruta. Seleccione el último archivo de metadatos válido conocido, que tiene el número más alto para el grupo de volúmenes.
2. Encuentre el UUID del volumen físico. Utilice uno de los siguientes métodos.

- Enumerar los volúmenes lógicos:

```
# lvs --all --options +devices

Couldn't find device with uuid 'FmGRh3-zhok-iVI8-7qTD-S5BI-MAEN-NYM5SK.
```

- Examine el archivo de metadatos archivado. Encuentre el UUID como el valor etiquetado **id =** en la sección **physical\_volumes** de la configuración del grupo de volumen.
- Desactivar el grupo de volumen utilizando la opción **--partial**:

```
# vgchange --activate n --partial volume-group-name

PARTIAL MODE. Incomplete logical volumes will be processed.
WARNING: Couldn't find device with uuid 42B7bu-YCMp-CEVD-CmKH-2rk6-fiO9-z1lf4s.
WARNING: VG raid_sanity is missing PV 42B7bu-YCMp-CEVD-CmKH-2rk6-fiO9-z1lf4s
(last written to /dev/sdb1).
0 logical volume(s) in volume group "raid_sanity" now active
```

### 19.4.3. Restauración de metadatos en un volumen físico LVM

Este procedimiento restaura los metadatos de un volumen físico dañado o sustituido por un nuevo dispositivo.



#### AVISO

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

#### Requisitos previos

- Ha identificado los metadatos del volumen físico que falta. Para más detalles, consulte [Sección 19.4.2, "Encontrar los metadatos de un volumen físico LVM perdido"](#) .

## Procedimiento

1. Restaurar los metadatos en el volumen físico:

```
# pvcreate --uuid physical-volume-uuid \
  --restorefile /etc/lvm/archive/volume-group-name_backup-number.vg \
  block-device
```

**NOTA**

El comando sólo sobrescribe las áreas de metadatos de LVM y no afecta a las áreas de datos existentes.

**Ejemplo 19.5. Restauración de un volumen físico en /dev/sdh1**

El siguiente ejemplo etiqueta el dispositivo **/dev/sdh1** como un volumen físico con las siguientes propiedades:

- El UUID de **FmGRh3-zhok-iVI8-7qTD-S5BI-MAEN-NYM5Sk**
- La información de metadatos contenida en **VG\_00050.vg**, que son los metadatos buenos más recientes archivados para el grupo de volúmenes

```
# pvcreate --uuid "FmGRh3-zhok-iVI8-7qTD-S5BI-MAEN-NYM5Sk" \
  --restorefile /etc/lvm/archive/VG_00050.vg \
  /dev/sdh1

...
Physical volume "/dev/sdh1" successfully created
```

2. Restaurar los metadatos del grupo de volúmenes:

```
# vgcfgrestore volume-group-name

Restored volume group volume-group-name
```

3. Muestra los volúmenes lógicos del grupo de volúmenes:

```
# lvs --all --options devices volume-group-name
```

Los volúmenes lógicos están actualmente inactivos. Por ejemplo:

```
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)
```

4. Si el tipo de segmento de los volúmenes lógicos es RAID o espejo, resincronice los volúmenes lógicos:

```
# lvchange --resync volume-group-name/logical-volume-name
```

5. Activar los volúmenes lógicos:

```
# lvchange --activate y /dev/volume-group-name/logical-volume-name
```

- Si los metadatos LVM en el disco ocupan al menos el mismo espacio que lo que los sobrepasó, este procedimiento puede recuperar el volumen físico. Si lo que anuló los metadatos fue más allá del área de metadatos, los datos en el volumen pueden haber sido afectados. Es posible que puedas utilizar el comando `fsck` para recuperar esos datos.

### Pasos de verificación

- Muestra los volúmenes lógicos activos:

```
# lvs --all --options +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)
```

## 19.5. REEMPLAZAR UN VOLUMEN FÍSICO LVM PERDIDO

Si un volumen físico falla o necesita ser reemplazado, puedes etiquetar un nuevo volumen físico para reemplazar el que se ha perdido en el grupo de volumen existente.

### Requisitos previos

- Ha sustituido el volumen físico por un nuevo dispositivo de almacenamiento.  
TODO: Reevaluar la ubicación de este paso.

### 19.5.1. Encontrar los metadatos de un volumen físico LVM perdido

Este procedimiento encuentra los últimos metadatos archivados de un volumen físico que falta o está dañado.

#### Procedimiento

- Busca el archivo de metadatos del grupo de volúmenes que contiene el volumen físico. Los archivos de metadatos archivados se encuentran en la **`/etc/lvm/archive/volume-group-name_backup-number.vg`** ruta. Seleccione el último archivo de metadatos válido conocido, que tiene el número más alto para el grupo de volúmenes.

- Encuentre el UUID del volumen físico. Utilice uno de los siguientes métodos.

- Enumerar los volúmenes lógicos:

```
# lvs --all --options +devices

Couldn't find device with uuid 'FmGRh3-zhok-iVI8-7qTD-S5BI-MAEN-NYM5SK.'
```

- Examine el archivo de metadatos archivado. Encuentre el UUID como el valor etiquetado **id =** en la sección **physical\_volumes** de la configuración del grupo de volumen.
- Desactivar el grupo de volumen utilizando la opción **--partial**:

```
# vgchange --activate n --partial volume-group-name

PARTIAL MODE. Incomplete logical volumes will be processed.
WARNING: Couldn't find device with uuid 42B7bu-YCMp-CEVD-CmKH-2rk6-fiO9-
```

*z1lf4s.*

WARNING: VG *raid\_sanity* is missing PV *42B7bu-YCMp-CEVD-CmKH-2rk6-fiO9-z1lf4s* (last written to */dev/sdb1*).

0 logical volume(s) in volume group "*raid\_sanity*" now active

## 19.5.2. Restauración de metadatos en un volumen físico LVM

Este procedimiento restaura los metadatos de un volumen físico dañado o sustituido por un nuevo dispositivo.



### AVISO

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

### Requisitos previos

- Ha identificado los metadatos del volumen físico que falta. Para más detalles, consulte [Sección 19.5.1, "Encontrar los metadatos de un volumen físico LVM perdido"](#) .

### Procedimiento

1. Restaurar los metadatos en el volumen físico:

```
# pvcreate --uuid physical-volume-uuid \
  --restorefile /etc/lvm/archive/volume-group-name_backup-number.vg \
  block-device
```



### NOTA

El comando sólo sobrescribe las áreas de metadatos de LVM y no afecta a las áreas de datos existentes.

### Ejemplo 19.6. Restauración de un volumen físico en */dev/sdh1*

El siguiente ejemplo etiqueta el dispositivo */dev/sdh1* como un volumen físico con las siguientes propiedades:

- El UUID de **FmGRh3-zhok-iVl8-7qTD-S5BI-MAEN-NYM5Sk**
- La información de metadatos contenida en **VG\_00050.vg**, que son los metadatos buenos más recientes archivados para el grupo de volúmenes

```
# pvcreate --uuid "FmGRh3-zhok-iVl8-7qTD-S5BI-MAEN-NYM5Sk" \
  --restorefile /etc/lvm/archive/VG_00050.vg \
  /dev/sdh1
```

```
...
Physical volume "/dev/sdh1" successfully created
```

- Restaurar los metadatos del grupo de volúmenes:

```
# vgcfgrestore volume-group-name

Restored volume group volume-group-name
```

- Muestra los volúmenes lógicos del grupo de volúmenes:

```
# lvs --all --options devices volume-group-name
```

Los volúmenes lógicos están actualmente inactivos. Por ejemplo:

```
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)
```

- Si el tipo de segmento de los volúmenes lógicos es RAID o espejo, resincronice los volúmenes lógicos:

```
# lvchange --resync volume-group-name/logical-volume-name
```

- Activar los volúmenes lógicos:

```
# lvchange --activate y /dev/volume-group-name/logical-volume-name
```

- Si los metadatos LVM en el disco ocupan al menos el mismo espacio que lo que los sobrepasó, este procedimiento puede recuperar el volumen físico. Si lo que anuló los metadatos fue más allá del área de metadatos, los datos en el volumen pueden haber sido afectados. Es posible que puedas utilizar el comando `fsck` para recuperar esos datos.

## Pasos de verificación

- Muestra los volúmenes lógicos activos:

```
# lvs --all --options +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)
```

## 19.6. SOLUCIÓN DE PROBLEMAS DE RAID LVM

Puede solucionar varios problemas en los dispositivos RAID LVM para corregir errores de datos, recuperar dispositivos o reemplazar dispositivos fallidos.

### 19.6.1. Comprobación de la coherencia de los datos en un volumen lógico RAID (RAID scrubbing)

LVM proporciona soporte de scrubbing para volúmenes lógicos RAID. El scrubbing de RAID es el proceso de lectura de todos los datos y bloques de paridad de un array y la comprobación de su coherencia.

## Procedimiento

1. Opcional: Limitar el ancho de banda de E/S que utiliza el proceso de depuración.  
Cuando se realiza una operación de depuración de RAID, la E/S en segundo plano requerida por las operaciones de **sync** puede desplazar otras E/S a los dispositivos LVM, como las actualizaciones de los metadatos del grupo de volúmenes. Esto puede hacer que las otras operaciones LVM se ralenticen. Se puede controlar la tasa de la operación de depuración implementando el estrangulamiento de la recuperación.

Añade las siguientes opciones a los comandos de **lvchange --syncaction** en los siguientes pasos:

### **--maxrecoveryrate** *Rate*[bBsSkKmMgG]

Establece la tasa de recuperación máxima para que la operación no desplace las operaciones de E/S nominales. Establecer la tasa de recuperación en 0 significa que la operación no tiene límites.

### **--minrecoveryrate** *Rate*[bBsSkKmMgG]

Establece la tasa de recuperación mínima para garantizar que la E/S de las operaciones de **sync** alcance un rendimiento mínimo, incluso cuando haya una E/S nominal intensa.

Especifique el valor de *Rate* como una cantidad por segundo para cada dispositivo de la matriz. Si no proporciona ningún sufijo, las opciones asumen kiB por segundo por dispositivo.

2. Muestra el número de discrepancias en la matriz, sin repararlas:

```
# lvchange --syncaction check vg/raid_lv
```

3. Corregir las discrepancias en la matriz:

```
# lvchange --syncaction repair vg/raid_lv
```



## NOTA

La operación **lvchange --syncaction repair** no realiza la misma función que la operación **lvconvert --repair**:

- La operación **lvchange --syncaction repair** inicia una operación de sincronización en segundo plano en la matriz.
- La operación **lvconvert --repair** repara o sustituye los dispositivos que fallan en un espejo o volumen lógico RAID.

4. Opcional: Muestra información sobre la operación de lavado:

```
# lvs -o raid_sync_action,raid_mismatch_count vg/lv
```

- El campo **raid\_sync\_action** muestra la operación de sincronización actual que está realizando el volumen RAID. Puede ser uno de los siguientes valores:

**idle**

Todas las operaciones de sincronización completadas (sin hacer nada)

### **resync**

Inicialización de una matriz o recuperación tras un fallo de la máquina

### **recover**

Sustitución de un dispositivo en la matriz

### **check**

Buscar incoherencias en las matrices

### **repair**

Buscar y reparar incoherencias

- El campo **raid\_mismatch\_count** muestra el número de discrepancias encontradas durante una operación de **check**.
- El campo **Cpy%Sync** muestra el progreso de las operaciones de **sync**.
- El campo **lv\_attr** proporciona indicadores adicionales. El bit 9 de este campo muestra la salud del volumen lógico, y admite los siguientes indicadores:
  - **m** (mismatches) indica que hay discrepancias en un volumen lógico RAID. Este carácter se muestra después de que una operación de depuración haya detectado que partes del RAID no son coherentes.
  - **r** (refresh) indica que un dispositivo en un array RAID ha sufrido un fallo y el kernel lo considera como fallido, aunque LVM puede leer la etiqueta del dispositivo y considera que el dispositivo está operativo. Refresca el volumen lógico para notificar al kernel que el dispositivo está ahora disponible, o reemplaza el dispositivo si sospechas que ha fallado.

### **Recursos adicionales**

- Para más información, consulte las páginas de manual **lvchange(8)** y **lvraid(7)**.

## **19.6.2. Dispositivos fallidos en RAID LVM**

RAID no es como el mirroring LVM tradicional. El mirroring LVM requería la eliminación de los dispositivos que fallaban o el volumen lógico reflejado se colgaba. Las matrices RAID pueden seguir funcionando con dispositivos fallidos. De hecho, para los tipos de RAID que no sean RAID1, eliminar un dispositivo significaría convertirlo a un nivel inferior de RAID (por ejemplo, de RAID6 a RAID5, o de RAID4 o RAID5 a RAID0).

Por lo tanto, en lugar de eliminar un dispositivo fallido incondicionalmente y potencialmente asignar un reemplazo, LVM le permite reemplazar un dispositivo fallido en un volumen RAID en una solución de un solo paso utilizando el argumento **--repair** del comando **lvconvert**.

## **19.6.3. Recuperación de un dispositivo RAID fallido en un volumen lógico**

Si el fallo del dispositivo RAID LVM es un fallo transitorio o puede reparar el dispositivo que ha fallado, puede iniciar la recuperación del dispositivo que ha fallado.

### **Requisitos previos**

- El dispositivo que antes fallaba ahora funciona.

## Procedimiento

- Actualiza el volumen lógico que contiene el dispositivo RAID:

```
# lvchange --refresh my_vg/my_lv
```

## Pasos de verificación

- Examine el volumen lógico con el dispositivo recuperado:

```
# lvs --all --options name,devices,lv_attr,lv_health_status my_vg
```

## 19.6.4. Sustitución de un dispositivo RAID fallido en un volumen lógico

Este procedimiento reemplaza un dispositivo fallado que sirve como volumen físico en un volumen lógico RAID LVM.

### Requisitos previos

- El grupo de volúmenes incluye un volumen físico que proporciona suficiente capacidad libre para reemplazar el dispositivo que ha fallado.  
Si no hay ningún volumen físico con suficientes extensiones libres en el grupo de volúmenes, añada un nuevo volumen físico lo suficientemente grande utilizando la utilidad **vgextend**.

## Procedimiento

1. En el siguiente ejemplo, un volumen lógico RAID está dispuesto de la siguiente manera:

```
# lvs --all --options name,copy_percent,devices my_vg

LV          Cpy%Sync Devices
my_lv       100.00 my_lv_rimage_0(0),my_lv_rimage_1(0),my_lv_rimage_2(0)
[my_lv_rimage_0] /dev/sde1(1)
[my_lv_rimage_1] /dev/sdc1(1)
[my_lv_rimage_2] /dev/sdd1(1)
[my_lv_rmeta_0] /dev/sde1(0)
[my_lv_rmeta_1] /dev/sdc1(0)
[my_lv_rmeta_2] /dev/sdd1(0)
```

2. Si el dispositivo **/dev/sdc** falla, la salida del comando **lvs** es la siguiente:

```
# lvs --all --options name,copy_percent,devices my_vg

/dev/sdc: open failed: No such device or address
Couldn't find device with uuid A4kRI2-vlZA-uyCb-cci7-bOod-H5tX-lzH4Ee.
WARNING: Couldn't find all devices for LV my_vg/my_lv_rimage_1 while checking used and
assumed devices.
WARNING: Couldn't find all devices for LV my_vg/my_lv_rmeta_1 while checking used and
assumed devices.
LV          Cpy%Sync Devices
my_lv       100.00 my_lv_rimage_0(0),my_lv_rimage_1(0),my_lv_rimage_2(0)
[my_lv_rimage_0] /dev/sde1(1)
[my_lv_rimage_1] [unknown](1)
[my_lv_rimage_2] /dev/sdd1(1)
```

```
[my_lv_rmeta_0]    /dev/sde1(0)
[my_lv_rmeta_1]    [unknown](0)
[my_lv_rmeta_2]    /dev/sdd1(0)
```

3. Sustituya el dispositivo que ha fallado y visualice el volumen lógico:

```
# lvconvert --repair my_vg/my_lv

/dev/sdc: open failed: No such device or address
Couldn't find device with uuid A4kRI2-vlZA-uyCb-cci7-bOod-H5tX-lzH4Ee.
WARNING: Couldn't find all devices for LV my_vg/my_lv_rimage_1 while checking used and
assumed devices.
WARNING: Couldn't find all devices for LV my_vg/my_lv_rmeta_1 while checking used and
assumed devices.
Attempt to replace failed RAID images (requires full device resync)? [y/n]: y
Faulty devices in my_vg/my_lv successfully replaced.
```

Opcional: Para especificar manualmente el volumen físico que sustituye al dispositivo que ha fallado, añada el volumen físico al final del comando:

```
# lvconvert --repair my_vg/my_lv replacement_pv
```

4. Examine el volumen lógico con la sustitución:

```
# lvs --all --options name,copy_percent,devices my_vg

/dev/sdc: open failed: No such device or address
/dev/sdc1: open failed: No such device or address
Couldn't find device with uuid A4kRI2-vlZA-uyCb-cci7-bOod-H5tX-lzH4Ee.
LV          Cpy%Sync Devices
my_lv       43.79  my_lv_rimage_0(0),my_lv_rimage_1(0),my_lv_rimage_2(0)
[my_lv_rimage_0]    /dev/sde1(1)
[my_lv_rimage_1]    /dev/sdb1(1)
[my_lv_rimage_2]    /dev/sdd1(1)
[my_lv_rmeta_0]     /dev/sde1(0)
[my_lv_rmeta_1]     /dev/sdb1(0)
[my_lv_rmeta_2]     /dev/sdd1(0)
```

Hasta que no elimine el dispositivo fallado del grupo de volúmenes, las utilidades LVM siguen indicando que LVM no puede encontrar el dispositivo fallado.

5. Elimine el dispositivo que ha fallado del grupo de volúmenes:

```
# vgreduce --removemissing VG
```

## 19.7. SOLUCIÓN DE PROBLEMAS DE INSUFICIENCIA DE EXTENSIONES LIBRES PARA UN VOLUMEN LÓGICO

Es posible que recibas el mensaje de error **Insufficient free extents** al intentar crear un volumen lógico, incluso cuando crees que el grupo de volúmenes tiene suficiente espacio libre. Puedes solucionar este error para poder crear un volumen lógico en el grupo de volúmenes.

### 19.7.1. Grupos de volumen

Los volúmenes físicos se combinan en grupos de volúmenes (VG). Esto crea una reserva de espacio en disco a partir de la cual se pueden asignar volúmenes lógicos.

Dentro de un grupo de volúmenes, el espacio de disco disponible para la asignación se divide en unidades de tamaño fijo llamadas extensiones. Una extensión es la unidad más pequeña de espacio que se puede asignar. Dentro de un volumen físico, los extents se denominan extents físicos.

Un volumen lógico se asigna en extensiones lógicas del mismo tamaño que las extensiones físicas. El tamaño de la extensión es, por tanto, el mismo para todos los volúmenes lógicos del grupo de volúmenes. El grupo de volúmenes asigna las extensiones lógicas a extensiones físicas.

### 19.7.2. Errores de redondeo en la salida de LVM

Los comandos de LVM que informan del uso de espacio en los grupos de volúmenes redondean el número informado a 2 decimales para proporcionar una salida legible para los humanos. Esto incluye las utilidades **vgdisplay** y **vgs**.

Como resultado del redondeo, el valor reportado de espacio libre podría ser mayor que el que proporcionan las extensiones físicas del grupo de volúmenes. Si intentas crear un volumen lógico del tamaño del espacio libre informado, podrías obtener el siguiente error:

**Insuficientes extensiones libres**

Para solucionar el error, debe examinar el número de extensiones físicas libres en el grupo de volúmenes, que es el valor exacto del espacio libre. A continuación, puede utilizar el número de extensiones para crear el volumen lógico con éxito.

### 19.7.3. Evitar el error de redondeo al crear un volumen LVM

Al crear un volumen lógico LVM, puede especificar el tamaño del volumen lógico para que no se produzca un error de redondeo.

#### Procedimiento

1. Encuentre el número de extensiones físicas libres en el grupo de volúmenes:

```
# vgdisplay volume-group-name
```

#### Ejemplo 19.7. Extensiones libres en un grupo de volúmenes

Por ejemplo, el siguiente grupo de volúmenes tiene 8780 extensiones físicas libres:

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

2. Cree el volumen lógico. Introduzca el tamaño del volumen en extensiones en lugar de bytes.

#### Ejemplo 19.8. Creación de un volumen lógico especificando el número de extensiones

```
# lvcreate --extents 8780 --name testlv testvg
```

**Ejemplo 19.9. Creación de un volumen lógico para ocupar todo el espacio restante**

Alternativamente, puedes ampliar el volumen lógico para utilizar un porcentaje del espacio libre restante en el grupo de volúmenes. Por ejemplo:

```
# lvcreate --extents 100EE --name testlv2 testvg
```

**Pasos de verificación**

- Comprueba el número de extensiones que utiliza ahora el grupo de volúmenes:

```
# vgs --options +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
```

**19.8. SOLUCIÓN DE PROBLEMAS DE ADVERTENCIAS DE VOLÚMENES FÍSICOS DUPLICADOS PARA DISPOSITIVOS LVM MULTIPATHED**

Cuando se utiliza LVM con almacenamiento multipista, los comandos LVM que enumeran un grupo de volúmenes o un volumen lógico pueden mostrar mensajes como el siguiente:

```
Found duplicate PV GDjTZf7Y03GJHjtecOwrye2dcSCjdaUi: using /dev/dm-5 not /dev/sdd
Found duplicate PV GDjTZf7Y03GJHjtecOwrye2dcSCjdaUi: using /dev/emcpowerb not /dev/sde
Found duplicate PV GDjTZf7Y03GJHjtecOwrye2dcSCjdaUi: using /dev/sddlmap not /dev/sdf
```

Puede solucionar estos avisos para entender por qué LVM los muestra, o para ocultar los avisos.

**19.8.1. Causa de la duplicación de los avisos de FV**

Cuando un software de multirruta como Device Mapper Multipath (DM Multipath), EMC PowerPath o Hitachi Dynamic Link Manager (HDLM) gestiona los dispositivos de almacenamiento en el sistema, cada ruta a una unidad lógica (LUN) concreta se registra como un dispositivo SCSI diferente. A continuación, el software multipath crea un nuevo dispositivo que se asigna a esas rutas individuales. Como cada LUN tiene varios nodos de dispositivo en el directorio **/dev** que apuntan a los mismos datos subyacentes, todos los nodos de dispositivo contienen los mismos metadatos LVM.

**Tabla 19.1. Ejemplo de mapeo de dispositivos en diferentes softwares multirruta**

Software multirrayectoria	Rutas SCSI a un LUN	Asignación de dispositivos multirruta a las rutas
DM Multipath	<b>/dev/sdb y /dev/sdc</b>	<b>/dev/mapper/mpath1 o /dev/mapper/mpatha</b>
EMC PowerPath		<b>/dev/emcpowera</b>
HDLM		<b>/dev/sddlmap</b>

Como resultado de los múltiples nodos de dispositivos, las herramientas LVM encuentran los mismos metadatos varias veces y los reportan como duplicados.

### 19.8.2. Casos de duplicación de avisos de FV

LVM muestra las advertencias de PV duplicado en cualquiera de los siguientes casos:

- Los dos dispositivos que se muestran en la salida son rutas únicas hacia el mismo dispositivo.
- Los dos dispositivos que se muestran en la salida son mapas multitrayectoria.

#### Camino individuales hacia el mismo dispositivo

El siguiente ejemplo muestra una advertencia de PV duplicada en la que los dispositivos duplicados son ambas rutas únicas hacia el mismo dispositivo.

```
Encontrado duplicado PV GDjTZf7Y03GJHjteqOwrye2dcSCjdaUi: usando /dev/sdd no /dev/sdf
```

Si se lista la topología actual de DM Multipath usando el comando **multipath -ll**, se puede encontrar tanto **/dev/sdd** como **/dev/sdf** bajo el mismo mapa multipath.

Estos mensajes duplicados son sólo advertencias y no significan que la operación LVM haya fallado. Más bien, le están alertando de que LVM utiliza sólo uno de los dispositivos como volumen físico e ignora los otros.

Si los mensajes indican que LVM elige el dispositivo incorrecto o si las advertencias son molestas para los usuarios, puede aplicar un filtro. El filtro configura LVM para que busque sólo los dispositivos necesarios para los volúmenes físicos, y para que omita cualquier ruta subyacente a los dispositivos multipath. Como resultado, las advertencias ya no aparecen.

#### Mapas multitrayectoria

Los siguientes ejemplos muestran una advertencia de volúmenes físicos duplicados para dos dispositivos que son ambos mapas multirruta. Los volúmenes físicos duplicados se encuentran en dos dispositivos diferentes y no en dos rutas diferentes hacia el mismo dispositivo.

```
Encontrado duplicado PV GDjTZf7Y03GJHjteqOwrye2dcSCjdaUi: usando /dev/mapper/mpatha no /dev/mapper/mpathc
```

```
Encontrado duplicado PV GDjTZf7Y03GJHjteqOwrye2dcSCjdaUi: usando /dev/emcpowera no /dev/emcpowerh
```

Esta situación es más grave que las advertencias de duplicación de dispositivos que son ambas rutas únicas al mismo dispositivo. Estas advertencias suelen significar que la máquina está accediendo a dispositivos a los que no debería acceder: por ejemplo, clones LUN o réplicas.

A menos que sepa claramente qué dispositivos debe eliminar de la máquina, esta situación podría ser irreparable. Red Hat recomienda que se ponga en contacto con el Soporte Técnico de Red Hat para solucionar este problema.

### 19.8.3. El filtro de dispositivos LVM

Las herramientas LVM escanean los dispositivos en el directorio **/dev** y comprueban cada uno de ellos en busca de metadatos LVM. Un filtro en el archivo **/etc/lvm/lvm.conf** controla qué dispositivos escanea LVM.

El filtro es una lista de patrones que LVM aplica a cada dispositivo encontrado por una exploración del directorio `/dev`, o el directorio especificado por la palabra clave `dir` en el archivo `/etc/lvm/lvm.conf`. Los patrones son expresiones regulares delimitadas por cualquier carácter y precedidas por `a` para *accept* o `r` para *reject*. La primera expresión regular de la lista que coincide con un dispositivo determina si LVM acepta o rechaza (ignora) el dispositivo. LVM acepta los dispositivos que no coinciden con ningún patrón.

La siguiente es la configuración por defecto del filtro, que escanea todos los dispositivos:

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

#### 19.8.4. Ejemplo de filtros de dispositivos LVM que evitan los avisos de PV duplicados

Los siguientes ejemplos muestran filtros de dispositivos LVM que evitan las advertencias de volúmenes físicos duplicados que son causadas por múltiples rutas de almacenamiento a una sola unidad lógica (LUN).

El filtro que configure debe incluir todos los dispositivos que LVM necesita para comprobar los metadatos, como el disco duro local con el grupo de volumen raíz en él y cualquier dispositivo multipathed. Rechazando las rutas subyacentes a un dispositivo multirruta (como `/dev/sdb`, `/dev/sdd`, y así sucesivamente), puede evitar estas advertencias de PV duplicado, porque LVM encuentra cada área de metadatos única una vez en el propio dispositivo multirruta.

- Este filtro acepta la segunda partición del primer disco duro y cualquier dispositivo DM Multipath, pero rechaza todo lo demás:

```
filter = [ \ "a/dev/sda2$|\", \ "a/dev/mapper/mpath.*|", \ "r|.*)" ]
```

- Este filtro acepta todos los controladores HP SmartArray y cualquier dispositivo EMC PowerPath:

```
filter = [ \ "a/dev/cciss.*|", \ "a/dev/emcpower.*|", \ "r|.*)" ]
```

- Este filtro acepta cualquier partición en la primera unidad IDE y cualquier dispositivo multipath:

```
filter = [ |"a/dev/hda.*|", |"a/dev/mapper/mpath.*|", |"r|.*)" ]
```

#### 19.8.5. Aplicación de una configuración de filtro de dispositivos LVM

Este procedimiento cambia la configuración del filtro de dispositivos LVM, que controla los dispositivos que escanea LVM.

##### Requisitos previos

- Prepare el patrón de filtro del dispositivo que desea utilizar.

##### Procedimiento

1. Pruebe el patrón de filtrado de su dispositivo sin modificar el archivo `/etc/lvm/lvm.conf`. Utilice un comando LVM con la opción `--config 'devices{ filter = [ your device filter pattern ]}'` opción. Por ejemplo:

```
# lvs --config 'devices{ filter = [ |"a/dev/emcpower.*|", |"r|.*)" ]'
```

2. Edite la opción **filter** en el archivo de configuración `/etc/lvm/lvm.conf` para utilizar su nuevo patrón de filtrado de dispositivos.
3. Compruebe que no faltan volúmenes físicos o grupos de volúmenes que desee utilizar con la nueva configuración:

```
# pvscan
```

```
# vgscan
```

4. Reconstruya el sistema de archivos **initramfs** para que LVM escanee sólo los dispositivos necesarios al reiniciar:

```
# dracut --force --verbose
```

### 19.8.6. Recursos adicionales

- [Capítulo 15, Control de la exploración de dispositivos LVM](#)