



Red Hat Enterprise Linux 8

Configuração e gerenciamento de volumes lógicos

Um guia para a configuração e gerenciamento dos volumes lógicos da LVM

Red Hat Enterprise Linux 8 Configuração e gerenciamento de volumes lógicos

Um guia para a configuração e gerenciamento dos volumes lógicos da 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.

Nota Legal

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

Resumo

Esta coleção de documentação fornece instruções sobre como gerenciar volumes lógicos LVM no Red Hat Enterprise Linux 8.

TORNANDO O CÓDIGO ABERTO MAIS INCLUSIVO	6
FORNECENDO FEEDBACK SOBRE A DOCUMENTAÇÃO DA RED HAT	7
CAPÍTULO 1. VOLUMES LÓGICOS	8
1.1. VISÃO GERAL DA ARQUITETURA LVM	8
1.2. VOLUMES FÍSICOS	9
1.2.1. LVM layout de volume físico	10
1.2.2. Partições múltiplas em um disco	10
1.3. GRUPOS DE VOLUME	11
1.4. LVM VOLUMES LÓGICOS	11
1.4.1. Volumes lineares	11
1.4.2. Volumes lógicos listrados	13
1.4.3. Volumes lógicos RAID	14
1.4.4. Volumes lógicos de fornecimento fino (volumes finos)	15
1.4.5. Volumes instantâneos	16
1.4.6. Volumes de fotos de última geração	17
1.4.7. Volumes de Cache	18
CAPÍTULO 2. GERENCIAMENTO DO ARMAZENAMENTO LOCAL USANDO AS FUNÇÕES DO SISTEMA RHEL	19
2.1. INTRODUÇÃO À FUNÇÃO DE ARMAZENAMENTO	19
2.2. PARÂMETROS QUE IDENTIFICAM UM DISPOSITIVO DE ARMAZENAMENTO NO PAPEL DO SISTEMA DE ARMAZENAMENTO	19
2.3. EXEMPLO LIVRO DE REPRODUÇÃO POSSÍVEL PARA CRIAR UM SISTEMA DE ARQUIVO XFS EM UM DISPOSITIVO DE BLOCO	20
2.4. EXEMPLO LIVRO DE REPRODUÇÃO POSSÍVEL PARA MONTAR PERSISTENTEMENTE UM SISTEMA DE ARQUIVO	21
2.5. EXEMPLO LIVRO DE EXERCÍCIOS POSSÍVEL PARA GERENCIAR VOLUMES LÓGICOS	21
2.6. EXEMPLO LIVRO DE REPRODUÇÃO POSSÍVEL PARA PERMITIR O DESCARTE EM BLOCO ONLINE	22
2.7. EXEMPLO LIVRO DE REPRODUÇÃO POSSÍVEL PARA CRIAR E MONTAR UM SISTEMA DE ARQUIVO EXT4	23
2.8. EXEMPLO LIVRO DE REPRODUÇÃO POSSÍVEL PARA CRIAR E MONTAR UM SISTEMA DE ARQUIVO EXT3	23
2.9. CONFIGURAÇÃO DE UM VOLUME RAID UTILIZANDO A FUNÇÃO DO SISTEMA DE ARMAZENAMENTO	24
2.10. CONFIGURAÇÃO DE UM POOL LVM COM RAID UTILIZANDO A FUNÇÃO DE SISTEMA DE ARMAZENAMENTO	25
2.11. CRIAÇÃO DE UM VOLUME CODIFICADO LUKS USANDO A FUNÇÃO DE ARMAZENAMENTO	27
CAPÍTULO 3. IMPLANTAÇÃO DE LVM	29
3.1. CRIAÇÃO DE VOLUME FÍSICO LVM	29
3.2. CRIAÇÃO DE GRUPO DE VOLUME LVM	29
3.3. CRIAÇÃO DO VOLUME LÓGICO LVM	30
CAPÍTULO 4. CONFIGURAÇÃO DE VOLUMES LÓGICOS LVM	31
4.1. USANDO COMANDOS CLI	31
Especificação de unidades em um argumento de linha de comando	31
Especificação de grupos de volumes e volumes lógicos	31
Aumentando a verbosidade da produção	31
Exibindo ajuda para comandos LVM CLI	32
4.2. CRIAÇÃO DE UM VOLUME LÓGICO LVM EM TRÊS DISCOS	32
4.3. CRIAÇÃO DE UM VOLUME LÓGICO RAID (LISTRADO)	33
4.4. RENOMEANDO VOLUMES LÓGICOS LVM	35

4.5. REMOÇÃO DE UM DISCO DE UM VOLUME LÓGICO	36
4.5.1. Movendo extensões para os volumes físicos existentes	36
4.5.2. Movendo Extensões para um Novo Disco	37
4.6. CONFIGURAÇÃO DE NÚMEROS DE DISPOSITIVOS PERSISTENTES	38
4.7. ESPECIFICAÇÃO DO TAMANHO DA EXTENSÃO DA LVM	38
4.8. GERENCIANDO VOLUMES LÓGICOS LVM USANDO AS FUNÇÕES DO SISTEMA RHEL	38
4.8.1. Exemplo Livro de exercícios possível para gerenciar volumes lógicos	39
4.8.2. Recursos adicionais	40
4.9. REMOÇÃO DE VOLUMES LÓGICOS LVM	40
CAPÍTULO 5. MODIFICANDO O TAMANHO DE UM VOLUME LÓGICO	41
5.1. VOLUMES LÓGICOS CRESCENTES	41
5.2. CULTIVAR UM SISTEMA DE ARQUIVO EM UM VOLUME LÓGICO	41
5.3. VOLUMES LÓGICOS RETRÁTEIS	42
5.4. AMPLIAÇÃO DE UM VOLUME LÓGICO LISTRADO	43
CAPÍTULO 6. GERENCIAMENTO DE GRUPOS DE VOLUME LVM	45
6.1. GRUPOS DE VOLUME	45
6.2. EXIBIÇÃO DE GRUPOS DE VOLUME	45
6.3. COMBINANDO GRUPOS DE VOLUME	46
6.4. DIVISÃO DE UM GRUPO DE VOLUME	46
6.5. RENOMEANDO GRUPOS DE VOLUME LVM	47
6.6. MOVENDO UM GRUPO DE VOLUME PARA OUTRO SISTEMA	48
6.7. REMOÇÃO DE VOLUMES FÍSICOS DE UM GRUPO DE VOLUME	49
6.8. REMOÇÃO DE GRUPOS DE VOLUME LVM	50
6.9. RECURSOS ADICIONAIS	50
CAPÍTULO 7. GERENCIANDO VOLUMES FÍSICOS LVM	51
7.1. PROCURA DE DISPOSITIVOS DE BLOCO A SEREM UTILIZADOS COMO VOLUMES FÍSICOS	51
7.2. DEFINIÇÃO DO TIPO DE DIVISÓRIA PARA UM VOLUME FÍSICO	51
7.3. REDIMENSIONAMENTO DE UM VOLUME FÍSICO LVM	52
7.4. REMOÇÃO DE VOLUMES FÍSICOS	52
7.5. ADICIONANDO VOLUMES FÍSICOS A UM GRUPO DE VOLUME	52
7.6. REMOÇÃO DE VOLUMES FÍSICOS DE UM GRUPO DE VOLUME	52
CAPÍTULO 8. EXIBIÇÃO DE COMPONENTES LVM	54
8.1. EXIBIÇÃO DE INFORMAÇÕES LVM COM O COMANDO LVM	54
8.2. EXIBIÇÃO DE VOLUMES FÍSICOS	54
8.3. EXIBIÇÃO DE GRUPOS DE VOLUME	55
8.4. EXIBINDO VOLUMES LÓGICOS	56
CAPÍTULO 9. RELATÓRIOS PERSONALIZADOS PARA LVM	57
9.1. CONTROLE DO FORMATO DO VISOR DA LVM	57
9.2. CAMPOS DE EXIBIÇÃO DE OBJETOS LVM	58
9.3. ORDENAÇÃO DE RELATÓRIOS LVM	69
9.4. ESPECIFICAÇÃO DAS UNIDADES PARA A EXIBIÇÃO DE UM RELATÓRIO LVM	69
9.5. EXIBIÇÃO DA SAÍDA DO COMANDO LVM EM FORMATO JSON	70
9.6. EXIBIÇÃO DO LOG DE COMANDO LVM	71
CAPÍTULO 10. CONFIGURAÇÃO DE VOLUMES LÓGICOS RAID	73
10.1. VOLUMES LÓGICOS RAID	73
10.2. NÍVEIS RAID E SUPORTE LINEAR	73
10.3. TIPOS DE SEGMENTOS RAID LVM	75
10.4. CRIAÇÃO DE VOLUMES LÓGICOS RAID	77
10.5. CRIAÇÃO DE UM VOLUME LÓGICO RAID (LISTRADO)	77

10.6. USANDO INTEGRIDADE DM COM RAID LV	79
10.6.1. Proteção de dados com integridade de DM	80
10.6.1.1. Considerações ao adicionar a integridade de DM	80
10.6.2. Criando um RAID LV com integridade DM	81
10.6.3. Adicionando a integridade de DM a um RAID LV existente	81
10.6.4. Removendo a integridade de um RAID LV	82
10.6.5. Visualizando informações sobre integridade da DM	82
10.6.6. Recursos adicionais	84
10.7. CONTROLAR A TAXA DE INICIALIZAÇÃO DOS VOLUMES RAID	84
10.8. CONVERSÃO DE UM DISPOSITIVO LINEAR EM UM DISPOSITIVO RAID	84
10.9. CONVERSÃO DE UM VOLUME LÓGICO LVM RAID1 PARA UM VOLUME LÓGICO LINEAR LVM	85
10.10. CONVERSÃO DE UM DISPOSITIVO LVM ESPELHADO EM UM DISPOSITIVO RAID1	86
10.11. REDIMENSIONAMENTO DE UM VOLUME LÓGICO RAID	86
10.12. MUDANDO O NÚMERO DE IMAGENS EM UM DISPOSITIVO RAID1 EXISTENTE	87
10.13. SEPARAÇÃO DE UMA IMAGEM RAID COMO UM VOLUME LÓGICO SEPARADO	89
10.14. DIVIDINDO E FUNDINDO UMA IMAGEM RAID	90
10.15. ESTABELECENDO UMA POLÍTICA DE FALHAS RAID	92
10.15.1. A política de falhas de alocação RAID	92
10.15.2. A política de falhas RAID de alerta	93
10.16. SUBSTITUIÇÃO DE UM DISPOSITIVO RAID EM UM VOLUME LÓGICO	93
10.16.1. Substituindo um dispositivo RAID que não tenha falhado	94
10.16.2. Dispositivos falhados em RAID LVM	96
10.16.3. Recuperando um dispositivo RAID falho em um volume lógico	97
10.16.4. Substituição de um dispositivo RAID falho em um volume lógico	97
10.17. VERIFICAÇÃO DA COERÊNCIA DOS DADOS EM UM VOLUME LÓGICO RAID (SCRUBBING RAID)	99
10.18. CONVERSÃO DE UM NÍVEL RAID (AQUISIÇÃO RAID)	100
10.19. MUDANÇA DE ATRIBUTOS DE UM VOLUME RAID (REFORMULAÇÃO RAID)	101
10.20. CONTROLE DAS OPERAÇÕES DE E/S EM UM VOLUME LÓGICO RAID1	101
10.21. MUDANDO O TAMANHO DA REGIÃO EM UM VOLUME LÓGICO RAID	101
CAPÍTULO 11. VOLUMES LÓGICOS INSTANTÂNEOS	102
11.1. VOLUMES INSTANTÂNEOS	102
11.2. CRIAÇÃO DE VOLUMES DE INSTANTÂNEOS	103
11.3. FUSÃO DE VOLUMES DE FOTOS	105
CAPÍTULO 12. CRIAÇÃO E GERENCIAMENTO DE VOLUMES LÓGICOS DE POUCA MONTA (VOLUMES FINOS)	106
12.1. VOLUMES LÓGICOS DE FORNECIMENTO FINO (VOLUMES FINOS)	106
12.2. CRIAÇÃO DE VOLUMES LÓGICOS POUCO PREVISÍVEIS	106
12.3. VOLUMES DE FOTOS DE ÚLTIMA GERAÇÃO	109
12.4. CRIAÇÃO DE VOLUMES DE INSTANTÂNEOS POUCO PREVISÍVEIS	110
12.5. RASTREAMENTO E EXIBIÇÃO DE VOLUMES FINOS QUE FORAM REMOVIDOS	112
CAPÍTULO 13. PERMITINDO O CACHE PARA MELHORAR O DESEMPENHO DO VOLUME LÓGICO	116
13.1. MÉTODOS DE CACHE EM LVM	116
13.2. COMPONENTES DE CACHING LVM	116
13.3. PERMITINDO O CACHING DM-CACHE PARA UM VOLUME LÓGICO	116
13.4. POSSIBILITANDO O CACHE DM-CACHE COM UM CACHEPOOL PARA UM VOLUME LÓGICO	118
13.5. POSSIBILITANDO O CACHE DM-WRITECACHE PARA UM VOLUME LÓGICO	119
13.6. DESABILITANDO O CACHING PARA UM VOLUME LÓGICO	121
CAPÍTULO 14. ATIVAÇÃO DE VOLUME LÓGICO	122
14.1. CONTROLE DA AUTO-ATIVAÇÃO DE VOLUMES LÓGICOS	122
14.2. CONTROLE DA ATIVAÇÃO LÓGICA DO VOLUME	123

14.3. ATIVAÇÃO DE VOLUMES LÓGICOS COMPARTILHADOS	123
14.4. ATIVAÇÃO DE UM VOLUME LÓGICO COM DISPOSITIVOS AUSENTES	124
CAPÍTULO 15. CONTROLE DA VARREDURA DO DISPOSITIVO LVM	125
15.1. O FILTRO DO DISPOSITIVO LVM	125
15.2. EXEMPLOS DE CONFIGURAÇÕES DE FILTROS DE DISPOSITIVOS LVM	125
15.3. APLICAÇÃO DE UMA CONFIGURAÇÃO DE FILTRO DO DISPOSITIVO LVM	126
CAPÍTULO 16. SOBREPOSIÇÃO DE VOLUMES FÍSICOS LVM SOBRE VOLUMES LÓGICOS	127
CAPÍTULO 17. CONTROLE DA ALOCAÇÃO DA LVM	128
17.1. POLÍTICAS DE ALOCAÇÃO DA LVM	128
17.2. IMPEDINDO A ALOCAÇÃO EM UM VOLUME FÍSICO	129
17.3. AMPLIAÇÃO DE UM VOLUME LÓGICO COM A POLÍTICA DE ALOCAÇÃO CLING	129
17.4. DIFERENCIANDO ENTRE OBJETOS RAID LVM USANDO TAGS	131
CAPÍTULO 18. AGRUPANDO OBJETOS LVM COM TAGS	133
18.1. ETIQUETAS DE OBJETOS LVM	133
18.2. LISTAGEM DAS ETIQUETAS LVM	133
18.3. ADICIONANDO TAGS DE OBJETOS LVM	133
18.4. REMOÇÃO DAS ETIQUETAS DE OBJETOS LVM	134
18.5. DEFININDO AS ETIQUETAS DO HOSPEDEIRO LVM	134
18.6. CONTROLE DA ATIVAÇÃO LÓGICA DE VOLUME COM TAGS	134
CAPÍTULO 19. SOLUÇÃO DE PROBLEMAS DE LVM	136
19.1. COLETA DE DADOS DE DIAGNÓSTICO SOBRE LVM	136
19.2. EXIBIÇÃO DE INFORMAÇÕES SOBRE DISPOSITIVOS LVM COM FALHAS	137
19.3. REMOÇÃO DE VOLUMES FÍSICOS LVM PERDIDOS DE UM GRUPO DE VOLUME	138
19.4. RECUPERANDO UM VOLUME FÍSICO LVM COM METADADOS DANIFICADOS	139
19.4.1. Descobrir que um volume LVM tem metadados ausentes ou corrompidos	139
19.4.2. Encontrando os metadados de um volume físico LVM ausente	139
19.4.3. Restaurando metadados em um volume físico LVM	140
19.5. SUBSTITUIÇÃO DE UM VOLUME FÍSICO LVM AUSENTE	142
19.5.1. Encontrando os metadados de um volume físico LVM ausente	142
19.5.2. Restaurando metadados em um volume físico LVM	143
19.6. SOLUÇÃO DE PROBLEMAS DE RAID LVM	144
19.6.1. Verificação da coerência dos dados em um volume lógico RAID (scrubbing RAID)	144
19.6.2. Dispositivos falhados em RAID LVM	146
19.6.3. Recuperando um dispositivo RAID falho em um volume lógico	146
19.6.4. Substituição de um dispositivo RAID falho em um volume lógico	147
19.7. SOLUÇÃO DE PROBLEMAS COM EXTENSÕES LIVRES INSUFICIENTES PARA UM VOLUME LÓGICO	148
19.7.1. Grupos de volume	148
19.7.2. Erros de arredondamento na saída da LVM	149
19.7.3. Prevenir o erro de arredondamento ao criar um volume LVM	149
19.8. SOLUÇÃO DE PROBLEMAS DE AVISOS DE VOLUME FÍSICO DUPLICADO PARA DISPOSITIVOS LVM MULTIPATHED	150
19.8.1. Causa raiz das advertências de PV duplicadas	150
19.8.2. Casos de avisos PV duplicados	151
19.8.3. O filtro do dispositivo LVM	151
19.8.4. Exemplo de filtros de dispositivos LVM que impedem a duplicação de avisos PV	152
19.8.5. Aplicação de uma configuração de filtro do dispositivo LVM	152
19.8.6. Recursos adicionais	153

TORNANDO O CÓDIGO ABERTO MAIS INCLUSIVO

A Red Hat tem o compromisso de substituir a linguagem problemática em nosso código, documentação e propriedades da web. Estamos começando com estes quatro termos: master, slave, blacklist e whitelist. Por causa da enormidade deste esforço, estas mudanças serão implementadas gradualmente ao longo de vários lançamentos futuros. Para mais detalhes, veja a [mensagem de nosso CTO Chris Wright](#).

FORNECENDO FEEDBACK SOBRE A DOCUMENTAÇÃO DA RED HAT

Agradecemos sua contribuição em nossa documentação. Por favor, diga-nos como podemos melhorá-la. Para fazer isso:

- Para comentários simples sobre passagens específicas:
 1. Certifique-se de que você está visualizando a documentação no formato *Multi-page HTML*. Além disso, certifique-se de ver o botão **Feedback** no canto superior direito do documento.
 2. Use o cursor do mouse para destacar a parte do texto que você deseja comentar.
 3. Clique no pop-up **Add Feedback** que aparece abaixo do texto destacado.
 4. Siga as instruções apresentadas.
- Para enviar comentários mais complexos, crie um bilhete Bugzilla:
 1. Ir para o site da [Bugzilla](#).
 2. Como Componente, use **Documentation**.
 3. Preencha o campo **Description** com sua sugestão de melhoria. Inclua um link para a(s) parte(s) relevante(s) da documentação.
 4. Clique em **Submit Bug**.

CAPÍTULO 1. VOLUMES LÓGICOS

O gerenciamento de volume cria uma camada de abstração sobre o armazenamento físico, permitindo criar volumes lógicos de armazenamento. Isto proporciona uma flexibilidade muito maior em diversas formas do que a utilização direta do armazenamento físico. Além disso, a configuração do armazenamento de hardware é escondida do software para que ele possa ser redimensionado e movido sem parar aplicações ou desmontar sistemas de arquivos. Isto pode reduzir os custos operacionais.

Os volumes lógicos oferecem as seguintes vantagens em relação ao uso direto do armazenamento físico:

- Capacidade flexível
Ao utilizar volumes lógicos, os sistemas de arquivo podem se estender por vários discos, já que é possível agregar discos e partições em um único volume lógico.
- Piscinas de armazenamento redimensionáveis
Você pode estender os volumes lógicos ou reduzir os volumes lógicos em tamanho com comandos de software simples, sem reformatar e reparticionar os dispositivos de disco subjacentes.
- Relocalização de dados on-line
Para implantar subsistemas de armazenamento mais novos, mais rápidos ou mais resilientes, você pode mover os dados enquanto seu sistema estiver ativo. Os dados podem ser rearranjados em discos enquanto os discos estiverem em uso. Por exemplo, você pode esvaziar um disco permutável a quente antes de removê-lo.
- Nome conveniente do dispositivo
Os volumes de armazenamento lógico podem ser gerenciados em grupos personalizados e definidos pelo usuário.
- Tiras de disco
Você pode criar um volume lógico que risca os dados em dois ou mais discos. Isto pode aumentar drasticamente a produção.
- Espelhamento de volumes
Os volumes lógicos proporcionam uma forma conveniente de configurar um espelho para seus dados.
- Instantâneos de volume
Usando volumes lógicos, você pode tirar instantâneos do dispositivo para backups consistentes ou para testar o efeito das mudanças sem afetar os dados reais.
- Volumes finos
Os volumes lógicos podem ser provisionados de forma fina. Isto permite criar volumes lógicos maiores do que as extensões disponíveis.
- Volumes de cache
Um volume lógico de cache usa um pequeno volume lógico que consiste em dispositivos de blocos rápidos (como unidades SSD) para melhorar o desempenho de um volume lógico maior e mais lento, armazenando os blocos freqüentemente usados no volume lógico menor e mais rápido.

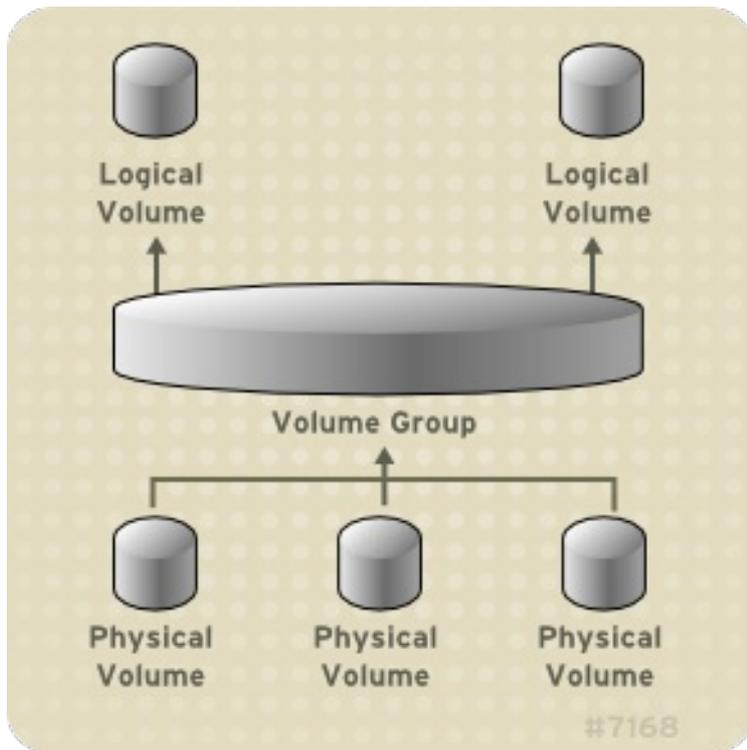
1.1. VISÃO GERAL DA ARQUITETURA LVM

A unidade de armazenamento físico subjacente de um volume lógico LVM é um dispositivo de bloco, como uma partição ou disco inteiro. Este dispositivo é inicializado como um LVM *physical volume* (PV).

Para criar um volume lógico LVM, os volumes físicos são combinados em um *volume group* (VG). Isto cria um pool de espaço em disco a partir do qual os volumes lógicos LVM (LVs) podem ser alocados. Este processo é análogo à maneira pela qual os discos são divididos em partições. Um volume lógico é usado por sistemas de arquivos e aplicações (como bancos de dados).

Figura 1.1, “Componentes lógicos de volume LVM” mostra os componentes de um simples volume lógico LVM:

Figura 1.1. Componentes lógicos de volume LVM



1.2. VOLUMES FÍSICOS

A unidade de armazenamento físico subjacente de um volume lógico LVM é um dispositivo de bloco, como uma partição ou disco inteiro. Para usar o dispositivo para um volume lógico LVM, o dispositivo deve ser inicializado como um volume físico (PV). A inicialização de um dispositivo de bloco como volume físico coloca uma etiqueta perto do início do dispositivo.

Por padrão, a etiqueta da LVM é colocada no segundo setor de 512 bytes. Você pode sobrescrever este padrão colocando a etiqueta em qualquer um dos 4 primeiros setores ao criar o volume físico. Isto permite que os volumes LVM coexistam com outros usuários destes setores, se necessário.

Uma etiqueta LVM fornece a identificação correta e o pedido de um dispositivo físico, uma vez que os dispositivos podem surgir em qualquer ordem quando o sistema é inicializado. Uma etiqueta LVM permanece persistente em todas as reinicializações e em todo um cluster.

A etiqueta LVM identifica o dispositivo como um volume físico LVM. Ela contém um identificador aleatório único (o UUID) para o volume físico. Ele também armazena o tamanho do dispositivo de bloco em bytes, e registra onde os metadados do LVM serão armazenados no dispositivo.

Os metadados do LVM contêm os detalhes de configuração dos grupos de volume do LVM em seu sistema. Por padrão, uma cópia idêntica dos metadados é mantida em cada área de metadados em cada volume físico dentro do grupo de volume. Os metadados do LVM são pequenos e armazenados como

ASCII.

Atualmente o LVM permite armazenar 0, 1 ou 2 cópias idênticas de seus metadados em cada volume físico. O padrão é 1 cópia. Uma vez configurado o número de cópias de metadados sobre o volume físico, não é possível alterar esse número posteriormente. A primeira cópia é armazenada no início do dispositivo, logo após a etiqueta. Se houver uma segunda cópia, ela é colocada no final do dispositivo. Se você acidentalmente sobrescrever a área no início do disco, escrevendo em um disco diferente do que você pretende, uma segunda cópia dos metadados no final do dispositivo lhe permitirá recuperar os metadados.

1.2.1. LVM layout de volume físico

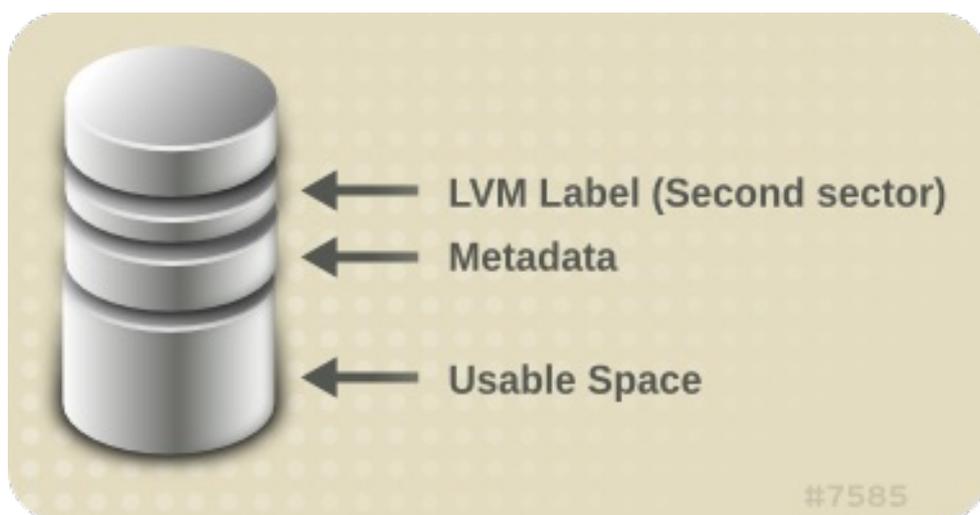
Figura 1.2, “Layout do volume físico” mostra o layout de um volume físico LVM. A etiqueta LVM está no segundo setor, seguida da área de metadados, seguida do espaço utilizável no dispositivo.



NOTA

No núcleo do Linux (e em todo este documento), os setores são considerados de 512 bytes de tamanho.

Figura 1.2. Layout do volume físico



1.2.2. Partições múltiplas em um disco

O LVM permite a criação de volumes físicos a partir de partições de disco. A Red Hat recomenda que você crie uma única partição que cubra o disco inteiro para rotular como um volume físico LVM pelas seguintes razões:

- Conveniência administrativa
É mais fácil acompanhar o hardware em um sistema se cada disco real aparecer apenas uma vez. Isto se torna particularmente verdadeiro se um disco falhar. Além disso, vários volumes físicos em um único disco podem causar um aviso do kernel sobre tipos de partição desconhecidos na inicialização.
- Desempenho de striping
A LVM não pode dizer que dois volumes físicos estão no mesmo disco físico. Se você criar um volume lógico listrado quando dois volumes físicos estão no mesmo disco físico, as faixas podem estar em partições diferentes no mesmo disco. Isto resultaria em uma diminuição do desempenho em vez de um aumento.

Embora não seja recomendado, pode haver circunstâncias específicas quando for necessário dividir um disco em volumes físicos LVM separados. Por exemplo, em um sistema com poucos discos pode ser necessário mover dados em torno de partições quando você estiver migrando um sistema existente para volumes LVM. Além disso, se você tiver um disco muito grande e quiser ter mais de um grupo de volumes para fins administrativos, então é necessário particionar o disco. Se você tiver um disco com mais de uma partição e ambas as partições estiverem no mesmo grupo de volumes, tenha o cuidado de especificar quais partições devem ser incluídas em um volume lógico ao criar volumes listrados.

1.3. GRUPOS DE VOLUME

Os volumes físicos são combinados em grupos de volume (VGs). Isto cria um pool de espaço em disco a partir do qual os volumes lógicos podem ser alocados.

Dentro de um grupo de volume, o espaço em disco disponível para alocação é dividido em unidades de tamanho fixo chamadas extensões. Uma extensão é a menor unidade de espaço que pode ser alocada. Dentro de um volume físico, as extensões são chamadas de extensões físicas.

Um volume lógico é alocado em extensões lógicas do mesmo tamanho que as extensões físicas. O tamanho da extensão é, portanto, o mesmo para todos os volumes lógicos do grupo de volume. O grupo de volume mapeia as extensões lógicas em extensões físicas.

1.4. LVM VOLUMES LÓGICOS

Na LVM, um grupo de volumes é dividido em volumes lógicos. Um administrador pode aumentar ou diminuir os volumes lógicos sem destruir os dados, ao contrário das partições de disco padrão. Se os volumes físicos em um grupo de volumes estiverem em unidades separadas ou matrizes RAID, então os administradores também podem espalhar um volume lógico pelos dispositivos de armazenamento.

Você pode perder dados se encolher um volume lógico para uma capacidade menor do que os dados sobre o volume exigem. Para garantir a máxima flexibilidade, crie volumes lógicos para atender às suas necessidades atuais e deixe o excesso de capacidade de armazenamento sem alocação. Você pode estender com segurança os volumes lógicos para usar o espaço não alocado, dependendo de suas necessidades.



IMPORTANTE

Nos servidores AMD, Intel, sistemas ARM e IBM Power Systems, o carregador de inicialização não consegue ler volumes LVM. Você deve fazer uma partição de disco padrão, não LVM, para sua partição /boot. No IBM Z, o carregador de inicialização zipl suporta /boot em volumes lógicos LVM com mapeamento linear. Por padrão, o processo de instalação sempre cria as partições / e swap dentro dos volumes LVM, com uma partição /boot separada em um volume físico.

As seções seguintes descrevem os diferentes tipos de volumes lógicos.

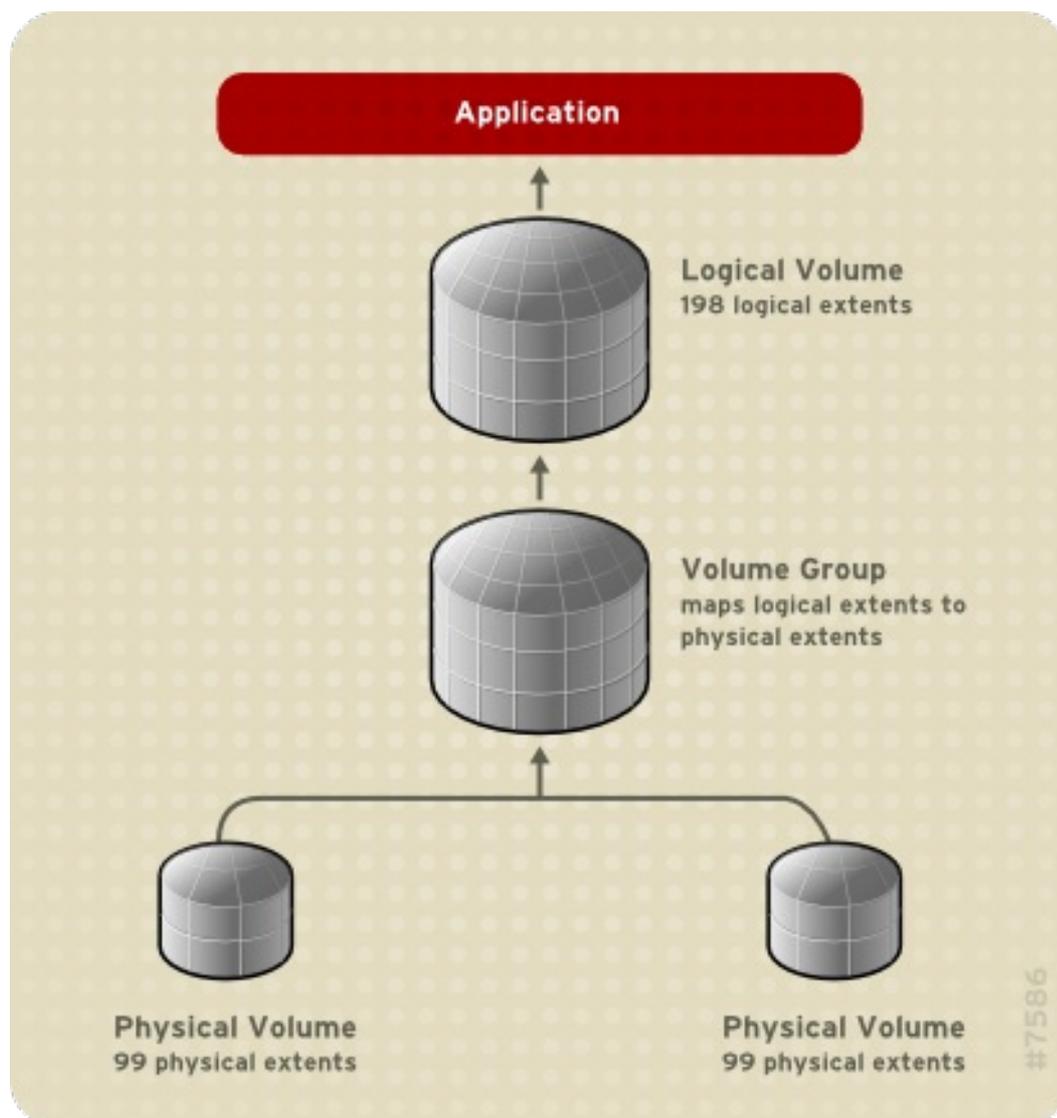
1.4.1. Volumes lineares

Um volume linear agrega espaço de um ou mais volumes físicos em um volume lógico. Por exemplo, se você tiver dois discos de 60GB, você pode criar um volume lógico de 120GB. O armazenamento físico é concatenado.

A criação de um volume linear atribui uma gama de extensões físicas a uma área de um volume lógico em ordem. Por exemplo, como mostrado em [Figura 1.3, "Mapeamento de Extensão"](#), extensões lógicas de 1 a 99 poderiam mapear um volume físico e extensões lógicas de 100 a 198 poderiam mapear um

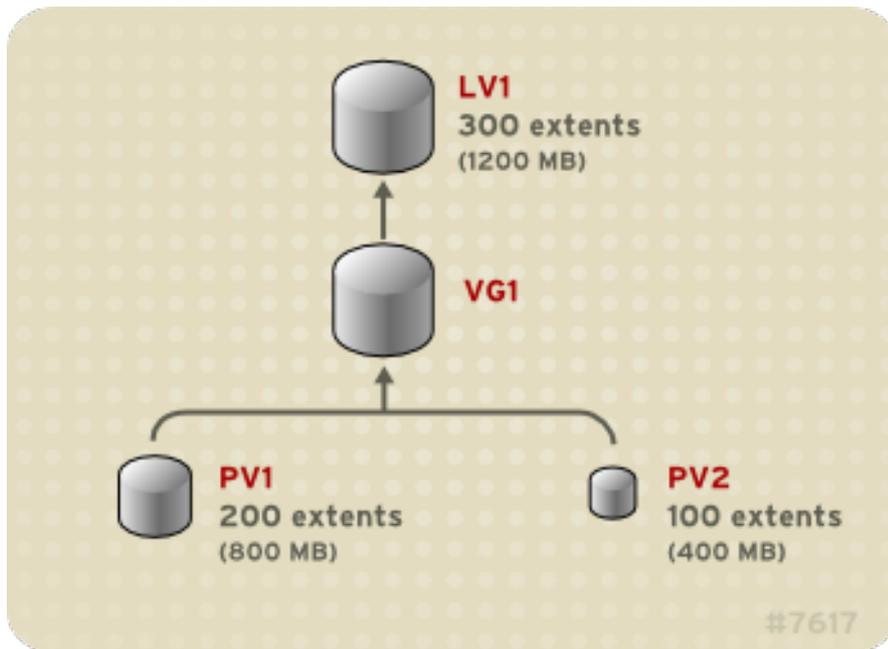
segundo volume físico. Do ponto de vista da aplicação, há um dispositivo que tem 198 extensões em tamanho.

Figura 1.3. Mapeamento de Extensão



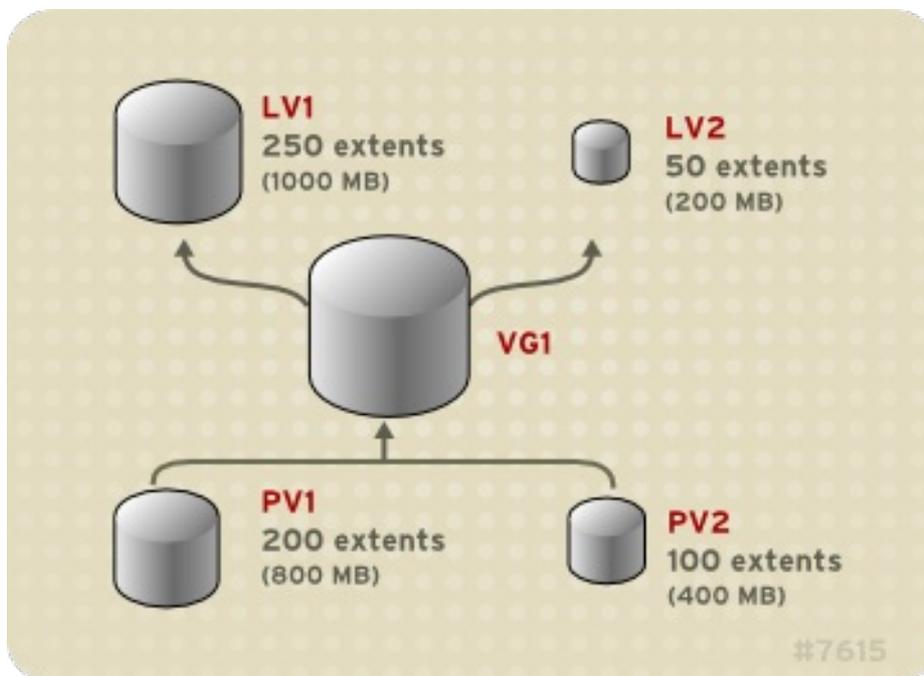
Os volumes físicos que compõem um volume lógico não precisam ter o mesmo tamanho. [Figura 1.4, "Volume linear com volumes físicos desiguais"](#) mostra o grupo de volume **VG1** com um tamanho de extensão física de 4MB. Este grupo de volume inclui 2 volumes físicos denominados **PV1** e **PV2**. Os volumes físicos são divididos em unidades de 4MB, já que esse é o tamanho da extensão. Neste exemplo, **PV1** tem 200 extensões em tamanho (800MB) e **PV2** tem 100 extensões em tamanho (400MB). Pode-se criar um volume linear de qualquer tamanho entre 1 e 300 extensões (4MB a 1200MB). Neste exemplo, o volume linear denominado **LV1** é de 300 extensões.

Figura 1.4. Volume linear com volumes físicos desiguais



Você pode configurar mais de um volume lógico linear de qualquer tamanho que desejar do pool de extensões físicas. [Figura 1.5, "Múltiplos volumes lógicos"](#) mostra o mesmo grupo de volume que em [Figura 1.4, "Volume linear com volumes físicos desiguais"](#), mas neste caso dois volumes lógicos foram esculpido do grupo de volume: **LV1**, que é 250 extensões em tamanho (1000MB) e **LV2** que é 50 extensões em tamanho (200MB).

Figura 1.5. Múltiplos volumes lógicos



1.4.2. Volumes lógicos listrados

Quando você escreve os dados em um volume lógico LVM, o sistema de arquivo estabelece os dados em todos os volumes físicos subjacentes. Você pode controlar a forma como os dados são escritos nos volumes físicos criando um volume lógico listrado. Para leituras e gravações sequenciais grandes, isto pode melhorar a eficiência da E/S dos dados.

A striping melhora o desempenho escrevendo dados para um número pré-determinado de volumes

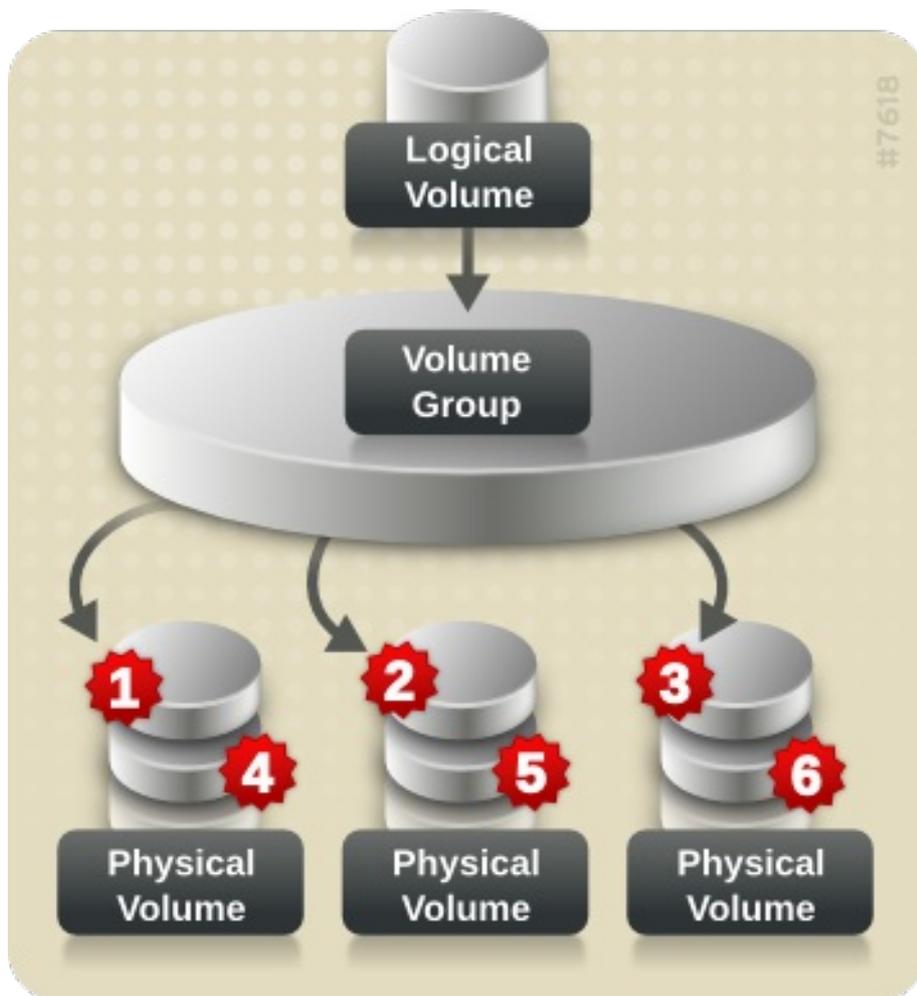
físicos de forma arredondada. Com a striping, as E/S podem ser feitas em paralelo. Em algumas situações, isto pode resultar em ganho de desempenho quase linear para cada volume físico adicional na lista.

A ilustração a seguir mostra os dados sendo listrados em três volumes físicos. Nesta figura:

- a primeira faixa de dados é escrita no primeiro volume físico
- a segunda faixa de dados é escrita para o segundo volume físico
- a terceira faixa de dados é escrita para o terceiro volume físico
- a quarta faixa de dados é escrita no primeiro volume físico

Em um volume lógico listrado, o tamanho da lista não pode exceder o tamanho de uma extensão.

Figura 1.6. Dados de striping em três PVs



Os volumes lógicos listrados podem ser estendidos concatenando outro conjunto de dispositivos no final do primeiro conjunto. Para estender um volume lógico listrado, entretanto, deve haver espaço livre suficiente no conjunto de volumes físicos subjacentes que compõem o grupo de volume para suportar a lista. Por exemplo, se você tiver uma faixa de duas vias que utiliza um grupo de volume inteiro, adicionar um único volume físico ao grupo de volume não lhe permitirá estender a faixa. Ao invés disso, você deve adicionar pelo menos dois volumes físicos ao grupo de volume.

1.4.3. Volumes lógicos RAID

O LVM suporta níveis RAID 0, 1, 4, 5, 6, e 10.

Um volume LVM RAID tem as seguintes características:

- Os volumes lógicos RAID criados e gerenciados pela LVM alavancam os drivers de kernel de múltiplos dispositivos (MD).
- Você pode dividir temporariamente as imagens RAID1 da matriz e fundi-las novamente na matriz mais tarde.
- Os volumes LVM RAID suportam snapshots.

Aglomerados

Os volumes lógicos RAID não são sensíveis a aglomerações.

Embora você possa criar e ativar volumes lógicos RAID exclusivamente em uma máquina, você não pode ativá-los simultaneamente em mais de uma máquina.

Subvolumes

Quando você cria um volume lógico RAID, o LVM cria um subvolume de metadados que é uma extensão em tamanho para cada dado ou subvolume de paridade na matriz.

Por exemplo, a criação de uma matriz RAID1 de duas vias resulta em dois subvolumes de metadados (**lv_rmeta_0** e **lv_rmeta_1**) e dois subvolumes de dados (**lv_rimage_0** e **lv_rimage_1**). Da mesma forma, criar uma faixa de 3 vias (mais 1 dispositivo de paridade implícita) RAID4 resulta em 4 subvolumes de metadados (**lv_rmeta_0**, **lv_rmeta_1**, **lv_rmeta_2**, e **lv_rmeta_3**) e 4 subvolumes de dados (**lv_rimage_0**, **lv_rimage_1**, **lv_rimage_2**, e **lv_rimage_3**).

Integridade

Você pode perder dados quando um dispositivo RAID falha ou quando ocorre corrupção suave. A corrupção suave no armazenamento de dados implica que os dados recuperados de um dispositivo de armazenamento são diferentes dos dados gravados naquele dispositivo. A adição de integridade a um RAID LV ajuda a mitigar ou prevenir a corrupção branda. Para saber mais sobre corrupção branda e como adicionar integridade a um RAID LV, consulte [Seção 10.6, "Usando integridade DM com RAID LV"](#).

1.4.4. Volumes lógicos de fornecimento fino (volumes finos)

Os volumes lógicos podem ser provisionados de forma fina. Isto permite criar volumes lógicos maiores do que as extensões disponíveis. Usando thin provisioning, você pode gerenciar um pool de armazenamento de espaço livre, conhecido como thin pool, que pode ser alocado a um número arbitrário de dispositivos quando necessário pelas aplicações. Você pode então criar dispositivos que podem ser vinculados ao thin pool para alocação posterior quando uma aplicação realmente escreve para o volume lógico. O thin pool pode ser expandido dinamicamente quando necessário para alocação econômica de espaço de armazenamento.



NOTA

Os volumes finos não são suportados através dos nós em um cluster. O pool fino e todos os seus volumes finos devem ser ativados exclusivamente em um único nó de cluster.

Ao utilizar thin provisioning, um administrador de armazenamento pode comprometer excessivamente o armazenamento físico, muitas vezes evitando a necessidade de comprar armazenamento adicional. Por exemplo, se cada dez usuários solicitarem um sistema de arquivo de 100GB para sua aplicação, o administrador de armazenamento pode criar o que parece ser um sistema de arquivo de 100GB para

cada usuário, mas que é apoiado por menos armazenamento real que é usado somente quando necessário. Ao utilizar thin provisioning, é importante que o administrador de armazenamento monitore o pool de armazenamento e acrescente mais capacidade se este começar a ficar cheio.

Para garantir que todo o espaço disponível possa ser utilizado, a LVM suporta o descarte de dados. Isto permite a reutilização do espaço que antes era utilizado por um arquivo descartado ou outro intervalo de blocos.

Os volumes finos fornecem suporte para uma nova implementação de volumes lógicos de cópia-em-escrita (COW), que permitem que muitos dispositivos virtuais compartilhem os mesmos dados no pool fino.

1.4.5. Volumes instantâneos

O recurso LVM snapshot oferece a capacidade de criar imagens virtuais de um dispositivo em um determinado instante sem causar uma interrupção do serviço. Quando uma mudança é feita no dispositivo original (a origem) após uma foto, o recurso de foto faz uma cópia da área de dados alterada como estava antes da mudança para que possa reconstruir o estado do dispositivo.



NOTA

A LVM apóia snapshots com pouco tempo de fornecimento.

Como um instantâneo copia apenas as áreas de dados que mudam depois que o instantâneo é criado, o recurso de instantâneo requer uma quantidade mínima de armazenamento. Por exemplo, com uma origem raramente atualizada, 3-5 % da capacidade da origem é suficiente para manter o instantâneo.



NOTA

As cópias instantâneas de um sistema de arquivo são cópias virtuais, não um backup de mídia real para um sistema de arquivo. Os instantâneos não fornecem um substituto para um procedimento de backup.

O tamanho do instantâneo rege a quantidade de espaço reservado para armazenar as mudanças no volume de origem. Por exemplo, se você fizer um instantâneo e depois sobrescrever completamente a origem, o instantâneo teria que ser pelo menos tão grande quanto o volume de origem para segurar as mudanças. Você precisa dimensionar um instantâneo de acordo com o nível de mudança esperado. Assim, por exemplo, um instantâneo de curta duração de um volume mais lido, como **/usr**, precisaria de menos espaço do que um instantâneo de longa duração de um volume que vê um número maior de escritas, como **/home**.

Se um instantâneo estiver cheio, o instantâneo se torna inválido, uma vez que não pode mais rastrear as mudanças no volume de origem. Você deve monitorar regularmente o tamanho do instantâneo. Os instantâneos são totalmente redimensionáveis, porém, se você tiver a capacidade de armazenamento, poderá aumentar o tamanho do volume do instantâneo para evitar que ele caia. Por outro lado, se você achar que o volume da foto é maior do que você precisa, você pode reduzir o tamanho do volume para liberar espaço que é necessário para outros volumes lógicos.

Quando você cria um sistema de arquivo instantâneo, o acesso completo de leitura e escrita à origem permanece possível. Se um pedaço de um instantâneo for alterado, esse pedaço é marcado e nunca é copiado do volume original.

Há vários usos para o recurso de instantâneo:

- Normalmente, um instantâneo é tirado quando é necessário fazer um backup em um volume lógico sem interromper o sistema ao vivo que está continuamente atualizando os dados.
- Você pode executar o comando **fsck** em um sistema de arquivo instantâneo para verificar a integridade do sistema de arquivo e determinar se o sistema de arquivo original requer reparo do sistema de arquivo.
- Como o instantâneo é lido/escrito, você pode testar aplicações contra os dados de produção tirando um instantâneo e executando testes contra o instantâneo, deixando os dados reais intocados.
- Você pode criar volumes LVM para uso com a Virtualização da Red Hat. Snapshots LVM podem ser usados para criar snapshots de imagens de convidados virtuais. Estes snapshots podem fornecer uma maneira conveniente para modificar os convidados existentes ou criar novos convidados com o mínimo de armazenamento adicional.

Você pode usar a opção **--merge** do comando **lvconvert** para fundir um instantâneo em seu volume de origem. Um uso para este recurso é realizar o rollback do sistema se você tiver perdido dados ou arquivos ou se precisar restaurar seu sistema a um estado anterior. Após fundir o volume de snapshot, o volume lógico resultante terá o nome do volume de origem, número menor e UUID e o snapshot fundido é removido.

1.4.6. Volumes de fotos de última geração

O Red Hat Enterprise Linux fornece suporte para volumes de instantâneos pouco fornecidos. Os volumes de snapshot finos permitem que muitos dispositivos virtuais sejam armazenados no mesmo volume de dados. Isto simplifica a administração e permite o compartilhamento de dados entre os volumes de snapshot.

Como para todos os volumes de instantâneos LVM, assim como para todos os volumes finos, os volumes de instantâneos finos não são suportados em todos os nós de um cluster. O volume de instantâneos deve ser ativado exclusivamente em um único nó de cluster.

Os volumes finos proporcionam os seguintes benefícios:

- Um volume fino de instantâneo pode reduzir o uso do disco quando há vários instantâneos com o mesmo volume de origem.
- Se houver vários instantâneos da mesma origem, então uma gravação na origem fará com que uma operação COW preserve os dados. Aumentar o número de instantâneos da origem não deve resultar em uma grande desaceleração.
- Os volumes finos podem ser usados como uma origem lógica de volume para outro instantâneo. Isto permite uma profundidade arbitrária de instantâneos recursivos (instantâneos de instantâneos de instantâneos de instantâneos...).
- Um instantâneo de um volume lógico fino também cria um volume lógico fino. Isto não consome espaço de dados até que uma operação COW seja necessária, ou até que o próprio instantâneo seja escrito.
- Um volume de instantâneo fino não precisa ser ativado com sua origem, portanto, um usuário pode ter apenas a origem ativa enquanto houver muitos volumes de instantâneos inativos da origem.
- Quando você apaga a origem de um volume de imagens de origem, cada imagem desse volume de origem se torna um volume independente de imagens de origem. Isto significa que, ao invés de fundir um instantâneo com seu volume de origem, você pode optar por excluir o volume de

origem e então criar um novo instantâneo com uma fina previsão usando esse volume independente como o volume de origem para o novo instantâneo.

Embora haja muitas vantagens no uso de volumes finos, há alguns casos de uso para os quais o recurso de volume de instantâneos LVM antigo pode ser mais apropriado às suas necessidades:

- Não se pode mudar o tamanho de um pedaço de uma piscina fina. Se a piscina fina tiver um pedaço de tamanho grande (por exemplo, 1MB) e você precisar de uma foto de curta duração para a qual um pedaço de tamanho tão grande não seja eficiente, você pode optar por usar o recurso de foto mais antigo.
- Não é possível limitar o tamanho de um volume fino; o instantâneo utilizará todo o espaço da piscina fina, se necessário. Isto pode não ser apropriado para suas necessidades.

Em geral, você deve considerar as exigências específicas de seu site ao decidir qual o formato de instantâneo a ser utilizado.

1.4.7. Volumes de Cache

LVM suporta o uso de dispositivos de blocos rápidos (como unidades SSD) como caches de gravação ou caches de gravação para dispositivos de blocos maiores e mais lentos. Os usuários podem criar volumes lógicos de cache para melhorar o desempenho de seus volumes lógicos existentes ou criar novos volumes lógicos de cache compostos de um dispositivo pequeno e rápido acoplado a um dispositivo grande e lento.

CAPÍTULO 2. GERENCIAMENTO DO ARMAZENAMENTO LOCAL USANDO AS FUNÇÕES DO SISTEMA RHEL

Para gerenciar LVM e sistemas de arquivos locais (FS) usando o Ansible, você pode usar a função **storage**, que é uma das funções do Sistema RHEL disponível no RHEL 8.

O uso da função **storage** permite automatizar a administração de sistemas de arquivos em discos e volumes lógicos em múltiplas máquinas e em todas as versões da RHEL, começando com a RHEL 7.7.

Para mais informações sobre os papéis do Sistema RHEL e como aplicá-los, consulte [Introdução aos papéis do Sistema RHEL](#).

2.1. INTRODUÇÃO À FUNÇÃO DE ARMAZENAMENTO

A função **storage** pode administrar:

- Sistemas de arquivos em discos que não foram particionados
- Grupos completos de volumes LVM incluindo seus volumes lógicos e sistemas de arquivo

Com o papel **storage** você pode realizar as seguintes tarefas:

- Criar um sistema de arquivo
- Remover um sistema de arquivo
- Montar um sistema de arquivo
- Desmontar um sistema de arquivo
- Criar grupos de volume LVM
- Remover grupos de volume LVM
- Criar volumes lógicos
- Remover volumes lógicos
- Criar volumes RAID
- Remover volumes RAID
- Criar pools LVM com RAID
- Remover as piscinas LVM com RAID

2.2. PARÂMETROS QUE IDENTIFICAM UM DISPOSITIVO DE ARMAZENAMENTO NO PAPEL DO SISTEMA DE ARMAZENAMENTO

Sua configuração de funções **storage** afeta apenas os sistemas de arquivos, volumes e pools que você lista nas seguintes variáveis.

storage_volumes

Lista de sistemas de arquivos em todos os discos não particionados a serem gerenciados. Atualmente, as partições não têm suporte.

storage_pools

Lista de piscinas a serem administradas.

Atualmente, o único tipo de piscina suportada é a LVM. Com LVM, os pools representam grupos de volume (VGs). Sob cada pool há uma lista de volumes a serem gerenciados pela função. Com o LVM, cada volume corresponde a um volume lógico (LV) com um sistema de arquivo.

2.3. EXEMPLO LIVRO DE REPRODUÇÃO POSSÍVEL PARA CRIAR UM SISTEMA DE ARQUIVO XFS EM UM DISPOSITIVO DE BLOCO

Esta seção fornece um exemplo de um livro de brincadeiras possível. Este playbook aplica o papel **storage** para criar um sistema de arquivos XFS em um dispositivo de bloco usando os parâmetros padrão.



ATENÇÃO

A função **storage** pode criar um sistema de arquivo somente em um disco não particionado, inteiro ou em um volume lógico (LV). Ele não pode criar o sistema de arquivo em uma partição.

Exemplo 2.1. Um playbook que cria XFS em /dev/sdb

```

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

```

- O nome do volume (**barefs** no exemplo) é atualmente arbitrária. A função **storage** identifica o volume pelo dispositivo de disco listado sob o atributo **disks:**.
- Você pode omitir a linha **fs_type: xfs** porque XFS é o sistema de arquivo padrão no RHEL 8.
- Para criar o sistema de arquivo em um LV, forneça a configuração LVM sob o atributo **disks:**, incluindo o grupo de volume envolvente. Para detalhes, veja [Exemplo Livro de exemplo para gerenciar volumes lógicos](#). Não forneça o caminho para o dispositivo LV.

Recursos adicionais

- Para detalhes sobre os parâmetros utilizados na função do sistema **storage**, consulte o arquivo `/usr/share/ansible/roles/rhel-system-roles.storage/README.md`.

2.4. EXEMPLO LIVRO DE REPRODUÇÃO POSSÍVEL PARA MONTAR PERSISTENTEMENTE UM SISTEMA DE ARQUIVO

Esta seção fornece um exemplo de um livro de brincadeiras possível. Este playbook aplica o papel **storage** para montar imediata e persistentemente um sistema de arquivos XFS.

Exemplo 2.2. Um playbook que monta um sistema de arquivo em `/dev/sdb` para `/mnt/dados`

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

- Este playbook adiciona o sistema de arquivo ao arquivo `/etc/fstab`, e monta o sistema de arquivo imediatamente.
- Se o sistema de arquivo no dispositivo `/dev/sdb` ou o diretório de pontos de montagem não existir, o playbook os cria.

Recursos adicionais

- Para detalhes sobre os parâmetros utilizados na função do sistema **storage**, consulte o arquivo `/usr/share/ansible/roles/rhel-system-roles.storage/README.md`.

2.5. EXEMPLO LIVRO DE EXERCÍCIOS POSSÍVEL PARA GERENCIAR VOLUMES LÓGICOS

Esta seção fornece um exemplo de um livro de brincadeiras possível. Este playbook aplica o papel **storage** para criar um volume lógico LVM em um grupo de volumes.

Exemplo 2.3. Um playbook que cria um volume lógico `mylv` no grupo de volume `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

```

- O grupo de volume **myvg** é composto pelos seguintes discos:
 - **/dev/sda**
 - **/dev/sdb**
 - **/dev/sdc**
- Se o grupo de volume **myvg** já existe, o playbook adiciona o volume lógico ao grupo de volume.
- Se o grupo de volume **myvg** não existe, o playbook o cria.
- O playbook cria um sistema de arquivo Ext4 no volume lógico **mylv** e monta persistentemente o sistema de arquivo em **/mnt**.

Recursos adicionais

- Para detalhes sobre os parâmetros utilizados na função do sistema **storage**, consulte o arquivo [/usr/share/ansible/roles/rhel-system-roles.storage/README.md](#).

2.6. EXEMPLO LIVRO DE REPRODUÇÃO POSSÍVEL PARA PERMITIR O DESCARTE EM BLOCO ONLINE

Esta seção fornece um exemplo de um livro de brincadeiras possível. Este playbook aplica o papel **storage** para montar um sistema de arquivo XFS com o descarte de blocos on-line habilitado.

Exemplo 2.4. Um playbook que permite o descarte de blocos online em `/mnt/dados/`

```

---
- 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 adicionais

- Este playbook também realiza todas as operações do exemplo de montagem persistente descrito em [Exemplo Um playbook possível para montar persistentemente um sistema de arquivo](#).
- Para detalhes sobre os parâmetros utilizados na função do sistema **storage**, consulte o arquivo `/usr/share/ansible/roles/rhel-system-roles.storage/README.md`.

2.7. EXEMPLO LIVRO DE REPRODUÇÃO POSSÍVEL PARA CRIAR E MONTAR UM SISTEMA DE ARQUIVO EXT4

Esta seção fornece um exemplo de um livro de brincadeiras possível. Este playbook aplica o papel **storage** para criar e montar um sistema de arquivos Ext4.

Exemplo 2.5. Um playbook que cria Ext4 em `/dev/sdb` e o monta em `/mnt/dados`

```
---
- 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
```

- O playbook cria o sistema de arquivos no disco `/dev/sdb`.
- O playbook monta persistentemente o sistema de arquivo no `/mnt/data` diretório.
- A etiqueta do sistema de arquivo é ***label-name***.

Recursos adicionais

- Para detalhes sobre os parâmetros utilizados na função do sistema **storage**, consulte o arquivo `/usr/share/ansible/roles/rhel-system-roles.storage/README.md`.

2.8. EXEMPLO LIVRO DE REPRODUÇÃO POSSÍVEL PARA CRIAR E MONTAR UM SISTEMA DE ARQUIVO EXT3

Esta seção fornece um exemplo de um livro de brincadeiras possível. Este playbook aplica o papel **storage** para criar e montar um sistema de arquivos Ext3.

Exemplo 2.6. Um playbook que cria Ext3 em `/dev/sdb` e o monta em `/mnt/dados`

```
---
- 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

```

- O playbook cria o sistema de arquivos no disco **/dev/sdb**.
- O playbook monta persistentemente o sistema de arquivo no **/mnt/data** diretório.
- A etiqueta do sistema de arquivo é **label-name**.

Recursos adicionais

- Para detalhes sobre os parâmetros utilizados na função do sistema **storage**, consulte o arquivo **/usr/share/ansible/roles/rhel-system-roles.storage/README.md**.

2.9. CONFIGURAÇÃO DE UM VOLUME RAID UTILIZANDO A FUNÇÃO DO SISTEMA DE ARMAZENAMENTO

Com o Sistema Função **storage**, você pode configurar um volume RAID na RHEL usando a Plataforma de Automação Possível Red Hat Ansible Automation. Nesta seção, você aprenderá como configurar um livro de jogo possível com os parâmetros disponíveis para configurar um volume RAID de acordo com suas necessidades.

Pré-requisitos

- Você tem o Red Hat Ansible Engine instalado no sistema a partir do qual você deseja executar o playbook.



NOTA

Você não precisa ter a Plataforma de Automação Possível da Red Hat instalada nos sistemas nos quais você deseja implantar a solução **storage**.

- Você tem o pacote **rhel-system-roles** instalado no sistema a partir do qual você deseja executar o playbook.
- Você tem um arquivo de inventário detalhando os sistemas nos quais você deseja implantar um volume RAID usando o sistema **storage** Função do sistema.

Procedimento

1. Criar um novo **playbook.yml** arquivo com o seguinte conteúdo:

```

- 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

```



ATENÇÃO

Os nomes dos dispositivos podem mudar em certas circunstâncias; por exemplo, quando você adiciona um novo disco a um sistema. Portanto, para evitar a perda de dados, não recomendamos o uso de nomes de disco específicos no livro de reprodução.

2. Opcional. Verificar a sintaxe do playbook.

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

3. Execute o playbook em seu arquivo de inventário:

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

Recursos adicionais

- Para mais informações sobre o RAID, consulte [Gerenciando RAID](#).
- Para detalhes sobre os parâmetros utilizados na função do sistema de armazenamento, consulte o arquivo `/usr/share/ansible/roles/rhel-system-roles.storage/README.md`.

2.10. CONFIGURAÇÃO DE UM POOL LVM COM RAID UTILIZANDO A FUNÇÃO DE SISTEMA DE ARMAZENAMENTO

Com o Sistema Função **storage**, você pode configurar um pool LVM com RAID na RHEL usando a Plataforma de Automação Possível da Red Hat. Nesta seção você aprenderá como configurar um playbook Ansible com os parâmetros disponíveis para configurar um pool LVM com RAID.

Pré-requisitos

- Você tem o Red Hat Ansible Engine instalado no sistema a partir do qual você deseja executar o playbook.

**NOTA**

Você não precisa ter a Plataforma de Automação Possível da Red Hat instalada nos sistemas nos quais você deseja implantar a solução **storage**.

- Você tem o pacote **rhel-system-roles** instalado no sistema a partir do qual você deseja executar o playbook.
- Você tem um arquivo de inventário detalhando os sistemas nos quais você deseja configurar um pool LVM com RAID usando o sistema **storage** Função do sistema.

Procedimento

1. Criar um novo **playbook.yml** arquivo com o seguinte conteúdo:

```
- 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 criar um pool LVM com RAID, você deve especificar o tipo de RAID usando o parâmetro **raid_level**.

2. Opcional. Verificar a sintaxe do playbook.

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

3. Execute o playbook em seu arquivo de inventário:

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

Recursos adicionais

- Para mais informações sobre o RAID, consulte [Gerenciando RAID](#).
- Para detalhes sobre os parâmetros utilizados na função do sistema de armazenamento, consulte o arquivo **/usr/share/ansible/roles/rhel-system-roles.storage/README.md**.

2.11. CRIAÇÃO DE UM VOLUME CODIFICADO LUKS USANDO A FUNÇÃO DE ARMAZENAMENTO

Você pode usar o papel **storage** para criar e configurar um volume criptografado com LUKS, executando um livro de brincadeiras Ansible playbook.

Pré-requisitos

- Você tem o Red Hat Ansible Engine instalado no sistema a partir do qual você deseja executar o playbook.



NOTA

Você não precisa ter a Plataforma de Automação Possível da Red Hat instalada nos sistemas nos quais você deseja criar o volume.

- Você tem o pacote **rhel-system-roles** instalado no controlador Ansible.
- Você tem um arquivo de inventário detalhando os sistemas nos quais você deseja implantar um volume codificado LUKS usando a função de sistema de armazenamento.

Procedimento

1. Criar um novo **playbook.yml** arquivo com o seguinte conteúdo:

```
- 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 a sintaxe do playbook:

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

3. Execute o playbook em seu arquivo de inventário:

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

Recursos adicionais

- Para mais informações sobre a LUKS, veja [17. Criptografando dispositivos de blocos usando LUKS...](#)

- Para detalhes sobre os parâmetros utilizados na função do sistema **storage**, consulte o arquivo **`/usr/share/ansible/roles/rhel-system-roles.storage/README.md`**.

Recursos adicionais

- Para mais informações, instale o pacote **rhel-system-roles** e veja os seguintes diretórios:
 - `/usr/share/doc/rhel-system-roles/storage/`
 - `/usr/share/ansible/roles/rhel-system-roles.storage/`

CAPÍTULO 3. IMPLANTAÇÃO DE LVM

Os seguintes procedimentos representam a configuração básica para volumes lógicos LVM em um sistema operacional recém-instalado.

3.1. CRIAÇÃO DE VOLUME FÍSICO LVM

Este procedimento descreve como criar e rotular volumes físicos LVM (PVs).

Pré-requisitos

- O pacote **lvm2** está instalado.

Procedimento

1. Para criar múltiplos volumes físicos, use os nomes de dispositivos com espaço delimitado como argumentos para o comando **pvcreate**:

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

Isto coloca uma etiqueta em `/dev/vdb1` e `/dev/vdb2`, marcando-os como volumes físicos pertencentes à LVM.

Para mais informações, consulte a página de manual **pvcreate**.

3.2. CRIAÇÃO DE GRUPO DE VOLUME LVM

Este procedimento descreve como criar um grupo de volume LVM (VG).

Pré-requisitos

- O pacote **lvm2** está instalado.
- São criados um ou mais volumes físicos. Veja as instruções de como fazê-lo em [Seção 3.1, "Criação de volume físico LVM"](#).

Procedimento

1. Para criar um grupo de volume, use o seguinte comando:

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

Isto cria uma VG com o nome de `myvg`. Os PVs `/dev/vdb1` e `/dev/vdb2` são o nível básico de armazenamento para a VG `myvg`.

Para mais informações, consulte a página de manual **vgcreate**.

É possível estender o VG acima com os PVs mais tarde. Para estender uma VG, use o seguinte comando:

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

3.3. CRIAÇÃO DO VOLUME LÓGICO LVM

Este procedimento descreve como criar o volume lógico LVM (LV).

Pré-requisitos

- O pacote **lvm2** está instalado.
- O grupo de volume é criado. Veja as instruções de como fazê-lo em [Seção 3.2, “Criação de grupo de volume LVM”](#).

Procedimento

1. Para criar um volume lógico, use o seguinte comando:

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

A opção **-n** permite que o usuário defina o nome LV para *lv01*. A opção **-L** permite que o usuário defina o tamanho do LV em unidades de Mb neste exemplo, mas é possível utilizar qualquer outra unidade. O tipo de BT é linear por padrão, mas o usuário pode especificar o tipo desejado usando a opção **--type**.



IMPORTANTE

O comando falhará se a VG não tiver um número suficiente de extensões físicas livres para o tamanho e tipo solicitados.

Para mais informações, consulte a página de manual **lvcreate**.

CAPÍTULO 4. CONFIGURAÇÃO DE VOLUMES LÓGICOS LVM

Os seguintes procedimentos fornecem exemplos de tarefas básicas de administração da LVM.

4.1. USANDO COMANDOS CLI

As seções seguintes descrevem algumas características operacionais gerais dos comandos LVM CLI.

Especificação de unidades em um argumento de linha de comando

Quando os tamanhos são exigidos em um argumento de linha de comando, as unidades podem sempre ser especificadas explicitamente. Se você não especificar uma unidade, então é assumido um padrão, geralmente KB ou MB. Os comandos LVM CLI não aceitam frações.

Ao especificar unidades em um argumento de linha de comando, LVM é insensível a maiúsculas e minúsculas; especificar M ou m é equivalente, por exemplo, e são usadas potências de 2 (múltiplos de 1024). Entretanto, ao especificar o argumento **--units** em um comando, a minúscula indica que as unidades estão em múltiplos de 1024, enquanto a maiúscula indica que as unidades estão em múltiplos de 1000.

Especificação de grupos de volumes e volumes lógicos

Observe o seguinte ao especificar grupos de volumes ou volumes lógicos em um comando LVM CLI.

- Quando os comandos tomam como argumentos nomes de grupos de volume ou de volumes lógicos, o nome completo do caminho é opcional. Um volume lógico chamado **lv010** em um grupo de volume chamado **vg0** pode ser especificado como **vg0/lv010**.
- Quando uma lista de grupos de volume for necessária, mas for deixada vazia, uma lista de todos os grupos de volume será substituída.
- Quando uma lista de volumes lógicos é necessária, mas um grupo de volumes é dado, uma lista de todos os volumes lógicos nesse grupo de volumes será substituída. Por exemplo, o comando **lvdisplay vg0** exibirá todos os volumes lógicos do grupo de volumes **vg0**.

Aumentando a verbosidade da produção

Todos os comandos LVM aceitam um argumento **-v**, que pode ser inserido várias vezes para aumentar a verbosidade de saída. Os exemplos a seguir mostram a saída padrão do comando **lvcreate**.

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

O seguinte comando mostra a saída do comando **lvcreate** com o 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 lv010
Creating volume group backup "/etc/lvm/backup/new_vg" (seqno 2).
Activating logical volume new_vg/lv010.
activation/volume_list configuration setting not defined: Checking only host tags for new_vg/lv010.
Creating new_vg-lv010
Loading table for new_vg-lv010 (253:0).
Resuming new_vg-lv010 (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
```

Os argumentos **-vv**, **-vvv** e **-vvvv** mostram cada vez mais detalhes sobre a execução do comando. O argumento **-vvvv** fornece a quantidade máxima de informações neste momento. O exemplo a seguir mostra as primeiras linhas de saída para o comando **lvcreate** com o 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
...
```

Exibindo ajuda para comandos LVM CLI

Você pode exibir ajuda para qualquer um dos comandos da LVM CLI com o argumento **--help** do comando.

```
# commandname --help
```

Para exibir a página de homem para um comando, execute o comando **man**:

```
# man commandname
```

O comando **man lvm** fornece informações gerais on-line sobre a LVM.

4.2. CRIAÇÃO DE UM VOLUME LÓGICO LVM EM TRÊS DISCOS

Este procedimento de exemplo cria um volume lógico LVM chamado **mylv** que consiste nos discos em **/dev/sda1**, **/dev/sdb1**, e **/dev/sdc1**.

1. Para utilizar discos em um grupo de volume, rotulá-los como volumes físicos LVM com o comando **pvcreate**.



ATENÇÃO

Este comando destrói quaisquer dados em **/dev/sda1**, **/dev/sdb1**, e **/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. Crie um grupo de volumes que consiste nos volumes físicos LVM que você criou. O seguinte comando cria o grupo de volume **myvg**.

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

Você pode usar o comando **vgs** para exibir os atributos do novo grupo de volume.

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

3. Crie o volume lógico a partir do grupo de volume que você criou. O seguinte comando cria o volume lógico **mylv** a partir do grupo de volume **myvg**. Este exemplo cria um volume lógico que utiliza 2 gigabytes do grupo de volume.

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

4. Criar um sistema de arquivo sobre o volume lógico. O seguinte comando cria um sistema de arquivo em **ext4** sobre o volume 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
```

Os seguintes comandos montam o volume lógico e informam o uso do espaço em disco do sistema de arquivos.

```
# 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. CRIAÇÃO DE UM VOLUME LÓGICO RAID0 (LISTRADO)

Um volume lógico RAID0 espalha dados lógicos de volume através de múltiplos subvolumes de dados em unidades de tamanho de faixa.

O formato para o comando para criar um volume RAID0 é o seguinte.

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

Tabela 4.1. Parâmetros de criação de comandos RAID0

Parâmetro	Descrição
--type raid0[_meta]	A especificação raid0 cria um volume RAID0 sem volumes de metadados. Especificando raid0_meta cria um volume RAID0 com volumes de metadados. Como o RAID0 não é resistente, ele não precisa armazenar nenhum bloco de dados espelhado como RAID1/10 ou calcular e armazenar qualquer bloco de paridade como o RAID4/5/6. Portanto, ele não precisa de volumes de metadados para manter o estado sobre o progresso da resincronização dos blocos espelhados ou de paridade. Os volumes de metadados tornam-se obrigatórios em uma conversão de RAID0 para RAID4/5/6/10, no entanto, e especificando raid0_meta pré-aloca esses volumes de metadados para evitar uma respectiva falha de alocação.
--stripes <i>Stripes</i>	Especifica o número de dispositivos para distribuir o volume lógico.
--stripesize <i>StripeSize</i>	Especifica o tamanho de cada listra em kilobytes. Esta é a quantidade de dados que é escrita em um dispositivo antes de passar para o dispositivo seguinte.
<i>VolumeGroup</i>	Especifica o grupo de volume a ser utilizado.
<i>PhysicalVolumePath...</i>	Especifica os dispositivos a serem utilizados. Se isto não for especificado, a LVM escolherá o número de dispositivos especificados pela opção <i>Stripes</i> , um para cada faixa.

Este procedimento de exemplo cria um volume lógico LVM RAID0 chamado **mylv** que risca os dados nos discos em **/dev/sda1**, **/dev/sdb1**, e **/dev/sdc1**.

1. Rotule os discos que você usará no grupo de volume como volumes físicos LVM com o comando **pvcreate**.



ATENÇÃO

Este comando destrói quaisquer dados em **/dev/sda1**, **/dev/sdb1**, e **/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. Criar o grupo de volume **myvg**. O seguinte comando cria o grupo de volume **myvg**.

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

Você pode usar o comando **vgs** para exibir os atributos do novo grupo de volume.

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

3. Crie um volume lógico RAID0 a partir do grupo de volume que você criou. O seguinte comando cria o volume RAID0 **mylv** a partir do grupo de volume **myvg**. Este exemplo cria um volume lógico de 2 gigabytes de tamanho, com três faixas e uma faixa 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. Criar um sistema de arquivo no volume lógico RAID0. O seguinte comando cria um sistema de arquivo em **ext4** no volume 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
```

Os seguintes comandos montam o volume lógico e informam o uso do espaço em disco do sistema de arquivos.

```
# 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. RENOMEANDO VOLUMES LÓGICOS LVM

Este procedimento renomeia um volume lógico existente usando a interface de linha de comando LVM.

Procedimento

1. Se o volume lógico estiver atualmente montado, desmonte o volume.
2. Se o volume lógico existir em um ambiente agrupado, desative o volume lógico em todos os nós onde ele está ativo. Use o seguinte comando em cada um desses nós:

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

3. Use o utilitário **lvrename** para renomear um volume lógico existente:

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

Opcionalmente, você pode especificar os caminhos completos para os dispositivos:

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

Recursos adicionais

- A página do homem **lvrename(8)**

4.5. REMOÇÃO DE UM DISCO DE UM VOLUME LÓGICO

Estes procedimentos de exemplo mostram como você pode remover um disco de um volume lógico existente, seja para substituir o disco ou para usar o disco como parte de um volume diferente. Para remover um disco, é necessário primeiro mover as extensões do volume físico do LVM para um disco ou conjunto de discos diferente.

4.5.1. Movendo extensões para os volumes físicos existentes

Neste exemplo, o volume lógico é distribuído por quatro volumes físicos no grupo de volume **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
```

Estes exemplos deslocam as extensões do **/dev/sdb1** para que ele possa ser removido do grupo de volume.

1. Se houver extensões livres suficientes nos outros volumes físicos do grupo de volume, você pode executar o comando **pvmove** no dispositivo que você deseja remover sem outras opções e as extensões serão distribuídas para os outros dispositivos.

Em um cluster, o comando **pvmove** pode mover somente o volume lógico que está ativo exclusivamente em um único nó.

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

Após o comando **pvmove** ter terminado a execução, a distribuição de extensões é a seguinte:

```
# 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
```

- Use o comando **vgreduce** para remover o volume físico **/dev/sdb1** do grupo de volume.

```
# 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
```

O disco pode agora ser removido fisicamente ou alocado a outros usuários.

4.5.2. Movendo Extensões para um Novo Disco

Neste exemplo, o volume lógico é distribuído em três volumes físicos no grupo de volume **myvg**, como segue:

```
# 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 procedimento de exemplo move as extensões do **/dev/sdb1** para um novo dispositivo, **/dev/sdd1**.

- Criar um novo volume físico a partir de **/dev/sdd1**.

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

- Adicionar o novo volume físico **/dev/sdd1** ao grupo de volume 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
```

- Use o comando **pvmove** para mover os dados de **/dev/sdb1** para **/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
```

4. Depois de ter retirado os dados de `/dev/sdb1`, você pode removê-los do grupo de volume.

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

Agora você pode realocar o disco para outro grupo de volume ou remover o disco do sistema.

4.6. CONFIGURAÇÃO DE NÚMEROS DE DISPOSITIVOS PERSISTENTES

Os números de dispositivos maiores e menores são alocados dinamicamente na carga do módulo. Algumas aplicações funcionam melhor se o dispositivo de bloco for sempre ativado com o mesmo número de dispositivo (maior e menor). Você pode especificá-los com os comandos **lvcreate** e **lvchange**, usando os seguintes argumentos:

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

Use um grande número menor para ter certeza de que ele não foi alocado a outro dispositivo de forma dinâmica.

Se você estiver exportando um sistema de arquivo usando NFS, especificar o parâmetro **fsid** no arquivo de exportação pode evitar a necessidade de definir um número de dispositivo persistente dentro do LVM.

4.7. ESPECIFICAÇÃO DO TAMANHO DA EXTENSÃO DA LVM

Quando volumes físicos são usados para criar um grupo de volume, seu espaço em disco é dividido em 4MB de extensão, por padrão. Esta extensão é a quantidade mínima pela qual o volume lógico pode ser aumentado ou diminuído em tamanho. Grandes números de extensões não terão impacto no desempenho de E/S do volume lógico.

Você pode especificar o tamanho de extensão com a opção **-s** para o comando **vgcreate** se o tamanho de extensão padrão não for adequado. Você pode colocar limites no número de volumes físicos ou lógicos que o grupo de volume pode ter usando os argumentos **-p** e **-l** do comando **vgcreate**.

4.8. GERENCIANDO VOLUMES LÓGICOS LVM USANDO AS FUNÇÕES DO SISTEMA RHEL

Esta seção descreve como aplicar a função **storage** para realizar as seguintes tarefas:

- Criar um volume lógico LVM em um grupo de volume composto de vários discos.
- Criar um sistema de arquivo ext4 com uma determinada etiqueta no volume lógico.
- Montar de forma persistente o sistema de arquivo ext4.

Pré-requisitos

- Um livro de brincadeiras possível, incluindo o papel **storage**

Para obter informações sobre como aplicar um caderno de atividades, consulte [Aplicando um papel](#).

4.8.1. Exemplo Livro de exercícios possível para gerenciar volumes lógicos

Esta seção fornece um exemplo de um livro de brincadeiras possível. Este playbook aplica o papel **storage** para criar um volume lógico LVM em um grupo de volumes.

Exemplo 4.1. Um playbook que cria um volume lógico mylv no grupo de volume 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
```

- O grupo de volume **myvg** é composto pelos seguintes discos:
 - **/dev/sda**
 - **/dev/sdb**
 - **/dev/sdc**
- Se o grupo de volume **myvg** já existe, o playbook adiciona o volume lógico ao grupo de volume.
- Se o grupo de volume **myvg** não existe, o playbook o cria.
- O playbook cria um sistema de arquivo Ext4 no volume lógico **mylv** e monta persistentemente o sistema de arquivo em **/mnt**.

Recursos adicionais

- Para detalhes sobre os parâmetros utilizados na função do sistema **storage**, consulte o arquivo [/usr/share/ansible/roles/rhel-system-roles.storage/README.md](#).

4.8.2. Recursos adicionais

- Para mais informações sobre a função **storage**, consulte [Gerenciando o armazenamento local usando as funções do Sistema RHEL](#).

4.9. REMOÇÃO DE VOLUMES LÓGICOS LVM

Este procedimento remove um volume lógico existente usando a interface de linha de comando LVM.

Os seguintes comandos removem o volume lógico **/dev/vg-name/lv-name** do grupo de volume **vg-name**.

Procedimento

1. Se o volume lógico estiver atualmente montado, desmonte o volume.
2. Se o volume lógico existir em um ambiente agrupado, desative o volume lógico em todos os nós onde ele está ativo. Use o seguinte comando em cada um desses nós:

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

3. Remova o volume lógico usando o utilitário **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

Neste caso, o volume lógico não foi desativado. Se você desativasse explicitamente o volume lógico antes de removê-lo, você não veria a solicitação de verificar se deseja remover um volume lógico ativo.

Recursos adicionais

- A página do homem **lvremove(8)**

CAPÍTULO 5. MODIFICANDO O TAMANHO DE UM VOLUME LÓGICO

Após ter criado um volume lógico, você pode modificar o tamanho do volume.

5.1. VOLUMES LÓGICOS CRESCENTES

Para aumentar o tamanho de um volume lógico, use o comando **lvextend**.

Ao estender o volume lógico, você pode indicar quanto deseja estender o volume, ou quão grande você quer que seja depois de estendê-lo.

O seguinte comando estende o volume lógico **/dev/myvg/homevol** para 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
```

O seguinte comando adiciona outro gigabyte ao volume 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
```

Como no comando **lvcreate**, você pode usar o argumento **-l** do comando **lvextend** para especificar o número de extensões pelas quais aumentar o tamanho do volume lógico. Você também pode usar este argumento para especificar uma porcentagem do grupo de volume, ou uma porcentagem do espaço livre restante no grupo de volume. O seguinte comando estende o volume lógico chamado **testlv** para preencher todo o espaço não alocado no grupo de volume **myvg**.

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

Depois de estender o volume lógico, é necessário aumentar o tamanho do sistema de arquivo para corresponder.

Por padrão, a maioria das ferramentas de redimensionamento do sistema de arquivos aumentará o tamanho do sistema de arquivos para ser o tamanho do volume lógico subjacente, de modo que você não precisa se preocupar em especificar o mesmo tamanho para cada um dos dois comandos.

5.2. CULTIVAR UM SISTEMA DE ARQUIVO EM UM VOLUME LÓGICO

Para cultivar um sistema de arquivo em um volume lógico, execute os seguintes passos:

1. Determinar se há espaço não alocado suficiente no grupo de volume existente para estender o volume lógico. Caso contrário, executar o procedimento a seguir:
 - a. Crie um novo volume físico com o comando **pvcreate**.

- b. Use o comando **vgextend** para ampliar o grupo de volume que contém o volume lógico com o sistema de arquivo que você está crescendo para incluir o novo volume físico.
2. Quando o grupo de volume for suficientemente grande para incluir o sistema de arquivo maior, estenda o volume lógico com o comando **lvresize**.
3. Redimensionar o sistema de arquivo no volume lógico.

Note que você pode usar a opção **-r** do comando **lvresize** para estender o volume lógico e redimensionar o sistema de arquivo subjacente com um único comando

5.3. VOLUMES LÓGICOS RETRÁTEIS

Você pode reduzir o tamanho de um volume lógico com o comando **lvreduce**.



NOTA

A retração não é suportada em um sistema de arquivo GFS2 ou XFS, portanto não se pode reduzir o tamanho de um volume lógico que contenha um sistema de arquivo GFS2 ou XFS.

Se o volume lógico que você está reduzindo contém um sistema de arquivo, para evitar a perda de dados, você deve garantir que o sistema de arquivo não está usando o espaço no volume lógico que está sendo reduzido. Por este motivo, recomenda-se usar a opção **--resizefs** do comando **lvreduce** quando o volume lógico contiver um sistema de arquivo. Quando você usa esta opção, o comando **lvreduce** tenta reduzir o sistema de arquivo antes de reduzir o volume lógico. Se o encolhimento do sistema de arquivo falhar, como pode ocorrer se o sistema de arquivo estiver cheio ou se o sistema de arquivo não suportar o encolhimento, então o comando **lvreduce** falhará e não tentará encolher o volume lógico.



ATENÇÃO

Na maioria dos casos, o comando **lvreduce** averte sobre uma possível perda de dados e pede uma confirmação. Entretanto, você não deve confiar nestes avisos de confirmação para evitar a perda de dados porque em alguns casos você não verá estes avisos, tais como quando o volume lógico estiver inativo ou quando a opção **--resizefs** não for utilizada.

Note que o uso da opção **--test** do comando **lvreduce** não indica onde a operação é segura, pois esta opção não verifica o sistema de arquivo ou não testa o redimensionamento do sistema de arquivo.

O seguinte comando reduz o volume lógico **lvvol1** no grupo de volume **vg00** para 64 megabytes. Neste exemplo, **lvvol1** contém um sistema de arquivo, que este comando redimensiona junto com o volume lógico. Este exemplo mostra a saída para o comando.

```
# lvreduce --resizefs -L 64M vg00/lvol1
fsck from util-linux 2.23.2
/dev/mapper/vg00-lvol1: 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.
```

A especificação do sinal - antes do valor de redimensionamento indica que o valor será subtraído do tamanho real do volume lógico. O exemplo seguinte mostra o comando que você usaria se, em vez de encolher um volume lógico para um tamanho absoluto de 64 megabytes, você quisesse encolher o volume em um valor de 64 megabytes.

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

5.4. AMPLIAÇÃO DE UM VOLUME LÓGICO LISTRADO

A fim de aumentar o tamanho de um volume lógico listrado, deve haver espaço livre suficiente nos volumes físicos subjacentes que compõem o grupo de volume para suportar a lista. Por exemplo, se você tiver uma faixa de duas vias que utilize um grupo de volume inteiro, adicionar um único volume físico ao grupo de volume não lhe permitirá estender a faixa. Ao invés disso, você deve adicionar pelo menos dois volumes físicos ao grupo de volume.

Por exemplo, considere um grupo de volumes **vg** que consiste de dois volumes físicos subjacentes, como exibido com o seguinte comando **vgs**.

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

Você pode criar uma faixa usando todo o espaço do grupo de volume.

```
# 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)
```

Note que o grupo de volume agora não tem mais espaço livre.

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

O seguinte comando acrescenta outro volume físico ao grupo de volume, que então tem 135 gigabytes de espaço 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
```

Neste ponto você não pode estender o volume lógico listrado até o tamanho total do grupo de volume, porque dois dispositivos subjacentes são necessários para riscar os dados.

```
# 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 estender o volume lógico listrado, adicionar outro volume físico e depois estender o volume lógico. Neste exemplo, tendo adicionado dois volumes físicos ao grupo de volume, podemos estender o volume lógico até o tamanho total do grupo de volume.

```
# 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
```

Se você não tiver dispositivos físicos subjacentes suficientes para estender o volume lógico listrado, é possível estender o volume de qualquer forma, se não importar que a extensão não seja listrada, o que pode resultar em um desempenho desigual. Ao adicionar espaço ao volume lógico, a operação padrão é usar os mesmos parâmetros de striping do último segmento do volume lógico existente, mas você pode sobrepor esses parâmetros. O exemplo a seguir estende o volume lógico em tiras existente para usar o espaço livre restante após o comando inicial **lvextend** falhar.

```
# 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. GERENCIAMENTO DE GRUPOS DE VOLUME LVM

Esta seção descreve os comandos que executam os vários aspectos da administração de grupos de volume.

6.1. GRUPOS DE VOLUME

Os volumes físicos são combinados em grupos de volume (VGs). Isto cria um pool de espaço em disco a partir do qual os volumes lógicos podem ser alocados.

Dentro de um grupo de volume, o espaço em disco disponível para alocação é dividido em unidades de tamanho fixo chamadas extensões. Uma extensão é a menor unidade de espaço que pode ser alocada. Dentro de um volume físico, as extensões são chamadas de extensões físicas.

Um volume lógico é alocado em extensões lógicas do mesmo tamanho que as extensões físicas. O tamanho da extensão é, portanto, o mesmo para todos os volumes lógicos do grupo de volume. O grupo de volume mapeia os extensões lógicas em extensões físicas.

6.2. EXIBIÇÃO DE GRUPOS DE VOLUME

Há dois comandos que você pode usar para exibir as propriedades dos grupos de volume LVM: **vgs** e **vgdisplay**. O comando **vgscan**, que verifica todos os dispositivos de bloco LVM suportados no sistema para grupos de volume, também pode ser usado para exibir os grupos de volume existentes.

O comando **vgs** fornece informações de grupo de volume em uma forma configurável, exibindo uma linha por grupo de volume. O comando **vgs** fornece um grande controle de formato, e é útil para a criação de scripts.

O comando **vgdisplay** exibe as propriedades do grupo de volume (tais como tamanho, extensões, número de volumes físicos, etc.) em uma forma fixa. O exemplo a seguir mostra a saída do comando **vgdisplay** para o grupo de volume **new_vg**. Se você não especificar um grupo de volume, todos os grupos de volume existentes são exibidos.

```
# 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
```

O exemplo a seguir mostra a saída do 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. COMBINANDO GRUPOS DE VOLUME

Para combinar dois grupos de volume em um único grupo de volume, use o comando **vgmerge**. Você pode fundir um volume inativo com um volume ativo ou inativo de destino se a extensão física do volume for igual e os resumos de volume físico e lógico de ambos os grupos de volume se encaixarem nos limites dos grupos de volume de destino.

O seguinte comando funde o grupo de volume inativo **my_vg** com o grupo de volume ativo ou inativo **databases** dando informações verbosas sobre o tempo de execução.

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

6.4. DIVISÃO DE UM GRUPO DE VOLUME

Neste procedimento de exemplo, um grupo de volume existente consiste em três volumes físicos. Se houver espaço suficiente não utilizado nos volumes físicos, um novo grupo de volumes pode ser criado sem a adição de novos discos.

Na montagem inicial, o volume lógico **mylv** é esculpido do grupo de volume **myvg**, que por sua vez consiste nos três volumes físicos, **/dev/sda1**, **/dev/sdb1**, e **/dev/sdc1**.

Após a conclusão deste procedimento, o grupo de volume **myvg** será composto por **/dev/sda1** e **/dev/sdb1**. Um segundo grupo de volume, **yourvg**, consistirá de **/dev/sdc1**.

1. Use o comando **pvscan** para determinar quanto espaço livre está atualmente disponível no grupo de volume.

```
# 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. Mova todas as extensões físicas utilizadas em **/dev/sdc1** para **/dev/sdb1** com o comando **pvmove**. O comando **pvmove** pode levar muito tempo para ser executado. Em um cluster, o comando **pvmove** pode mover somente o volume lógico que está ativo exclusivamente em um único nó.

```
# 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%

```

Depois de mover os dados, você pode ver que todo o espaço em **/dev/sdc1** é gratuito.

```

# 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 criar o novo grupo de volume **yourvg**, use o comando **vgsplit** para dividir o grupo de volume **myvg**.

O seguinte comando divide o grupo de volume **yourvg** do grupo de volume **myvg**, movendo o volume físico **/dev/sdc1** para o novo grupo de volume **yourvg**.

```

# lvchange -a n /dev/myvg/mylv
# vgsplit myvg yourvg /dev/sdc1
Volume group "yourvg" successfully split from "myvg"

```

Você pode usar o comando **vgs** para ver os atributos dos dois grupos de volumes.

```

# 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

```

- Após criar o novo grupo de volume, crie o novo volume lógico **yourlv**.

```

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

```

- Crie um sistema de arquivo no novo volume lógico e monte-o.

```

# 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. RENOMEANDO GRUPOS DE VOLUME LVM

Este procedimento renomeia um grupo de volume existente usando a interface de linha de comando LVM.

Procedimento

1. Se o grupo de volume existir em um ambiente agrupado, desative o grupo de volume em todos os nós onde ele está ativo. Use o seguinte comando em cada um desses nós:

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

2. Use o utilitário **vgrename** para renomear um grupo de volume existente:

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

Opcionalmente, você pode especificar os caminhos completos para os dispositivos:

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

Recursos adicionais

- A página do homem **vgrename(8)**

6.6. MOVENDO UM GRUPO DE VOLUME PARA OUTRO SISTEMA

Você pode mover todo um grupo de volume LVM para outro sistema. É recomendado que você use os comandos **vgexport** e **vgimport** quando fizer isso.



NOTA

Você pode usar o argumento **--force** do comando **vgimport**. Isto permite importar grupos de volumes que estão faltando volumes físicos e posteriormente executar o comando **vgreduce --removemissing**.

O comando **vgexport** torna inacessível ao sistema um grupo de volume inativo, o que permite destacar seus volumes físicos. O comando **vgimport** torna um grupo de volume acessível a uma máquina novamente após o comando **vgexport** tê-lo tornado inativo.

Para mover um grupo de volume de um sistema para outro, execute os seguintes passos:

1. Certifique-se de que nenhum usuário esteja acessando os arquivos nos volumes ativos do grupo de volumes, depois desmonte os volumes lógicos.
2. Use o argumento **-a n** do comando **vgchange** para marcar o grupo de volume como inativo, o que impede qualquer outra atividade no grupo de volume.
3. Use o comando **vgexport** para exportar o grupo de volume. Isto impede que ele seja acessado pelo sistema do qual você está removendo-o.

Depois de exportar o grupo de volume, o volume físico aparecerá como estando em um grupo de volume exportado quando você executar o comando **pvscan**, como no exemplo a seguir.

```
# 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]
```

```
...
```

Quando o sistema for desligado em seguida, você pode desconectar os discos que constituem o grupo de volume e conectá-los ao novo sistema.

4. Quando os discos estiverem conectados ao novo sistema, use o comando **vgimport** para importar o grupo de volume, tornando-o acessível ao novo sistema.
5. Ative o grupo de volume com o argumento **-a y** do comando **vgchange**.
6. Montar o sistema de arquivo para torná-lo disponível para uso.

6.7. REMOÇÃO DE VOLUMES FÍSICOS DE UM GRUPO DE VOLUME

Para remover volumes físicos não utilizados de um grupo de volume, use o comando **vgreduce**. O comando **vgreduce** diminui a capacidade de um grupo de volume ao remover um ou mais volumes físicos vazios. Isto libera esses volumes físicos para serem usados em diferentes grupos de volume ou para serem removidos do sistema.

Antes de remover um volume físico de um grupo de volume, você pode ter certeza de que o volume físico não é usado por nenhum volume lógico, usando o 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
```

Se o volume físico ainda estiver sendo usado, você terá que migrar os dados para outro volume físico usando o comando **pvmove**. Em seguida, use o comando **vgreduce** para remover o volume físico.

O seguinte comando remove o volume físico **/dev/hda1** do grupo de volume **my_volume_group**.

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

Se um volume lógico contém um volume físico que falha, você não pode usar esse volume lógico. Para remover volumes físicos ausentes de um grupo de volumes, você pode usar o parâmetro **--removemissing** do comando **vgreduce**, se não houver volumes lógicos alocados nos volumes físicos ausentes.

Se o volume físico que falhar contém uma imagem de espelho de um volume lógico de um segmento do tipo **mirror**, você pode remover essa imagem do espelho com o comando **vgreduce --removemissing -mirroronly --force**. Isto remove apenas os volumes lógicos que são imagens-espelho do volume físico.

6.8. REMOÇÃO DE GRUPOS DE VOLUME LVM

Este procedimento remove um grupo de volume existente usando a interface de linha de comando LVM.

Pré-requisitos

- O grupo de volume não contém volumes lógicos. Para remover volumes lógicos de um grupo de volume, ver [Seção 4.9, "Remoção de volumes lógicos LVM"](#) .

Procedimento

1. Se o grupo de volume existir em um ambiente agrupado, pare o espaço de bloqueio do grupo de volume em todos os outros nós. Use o seguinte comando em todos os nós, exceto no nó onde você está realizando a remoção:

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

Aguarde que o cadeado pare.

2. Para remover o grupo de volume, use o utilitário **vgremove**:

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

Recursos adicionais

- A página do homem **vgremove(8)**

6.9. RECURSOS ADICIONAIS

- A página do homem **vgchange(8)**

CAPÍTULO 7. GERENCIANDO VOLUMES FÍSICOS LVM

Há uma variedade de comandos e procedimentos que você pode usar para gerenciar os volumes físicos da LVM.

7.1. PROCURA DE DISPOSITIVOS DE BLOCO A SEREM UTILIZADOS COMO VOLUMES FÍSICOS

Você pode procurar dispositivos de bloco que podem ser usados como volumes físicos com o comando **lvmdiskscan**, como mostrado no exemplo a seguir.

```
# 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. DEFINIÇÃO DO TIPO DE DIVISÓRIA PARA UM VOLUME FÍSICO

Se você estiver usando um dispositivo de disco inteiro para seu volume físico, o disco não deve ter uma tabela de partições. Para partições de disco DOS, o id da partição deve ser ajustado para 0x8e usando o comando **fdisk** ou **cdisk** ou um equivalente. Para dispositivos de disco inteiro, somente a tabela de partição deve ser apagada, o que efetivamente destruirá todos os dados naquele disco. Você pode remover uma tabela de partição existente zerando o primeiro setor com o seguinte comando:

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

7.3. REDIMENSIONAMENTO DE UM VOLUME FÍSICO LVM

Se você precisar alterar o tamanho de um dispositivo de bloco subjacente por qualquer motivo, use o comando **pvresize** para atualizar o LVM com o novo tamanho. Você pode executar este comando enquanto o LVM estiver usando o volume físico.

7.4. REMOÇÃO DE VOLUMES FÍSICOS

Se um dispositivo não for mais necessário para uso pela LVM, você pode remover a etiqueta da LVM com o comando **pvremove**. A execução do comando **pvremove** zera os metadados do LVM em um volume físico vazio.

Se o volume físico que você deseja remover atualmente faz parte de um grupo de volume, você deve removê-lo do grupo de volume com o comando **vgreduce**.

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

7.5. ADICIONANDO VOLUMES FÍSICOS A UM GRUPO DE VOLUME

Para adicionar volumes físicos adicionais a um grupo de volume existente, use o comando **vgextend**. O comando **vgextend** aumenta a capacidade de um grupo de volume ao adicionar um ou mais volumes físicos livres.

O seguinte comando adiciona o volume físico **/dev/sdf1** ao grupo de volume **vg1**.

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

7.6. REMOÇÃO DE VOLUMES FÍSICOS DE UM GRUPO DE VOLUME

Para remover volumes físicos não utilizados de um grupo de volume, use o comando **vgreduce**. O comando **vgreduce** diminui a capacidade de um grupo de volume ao remover um ou mais volumes físicos vazios. Isto libera esses volumes físicos para serem usados em diferentes grupos de volume ou para serem removidos do sistema.

Antes de remover um volume físico de um grupo de volume, você pode ter certeza de que o volume físico não é usado por nenhum volume lógico, usando o comando **pvdisk**.

```
# pvdisk /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
```

Se o volume físico ainda estiver sendo usado, você terá que migrar os dados para outro volume físico usando o comando **pvmove**. Em seguida, use o comando **vgreduce** para remover o volume físico.

O seguinte comando remove o volume físico **/dev/hda1** do grupo de volume **my_volume_group**.

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

Se um volume lógico contém um volume físico que falha, você não pode usar esse volume lógico. Para remover volumes físicos ausentes de um grupo de volumes, você pode usar o parâmetro **--removemissing** do comando **vgreduce**, se não houver volumes lógicos alocados nos volumes físicos ausentes.

Se o volume físico que falhar contém uma imagem de espelho de um volume lógico de um segmento do tipo **mirror**, você pode remover essa imagem do espelho com o comando **vgreduce --removemissing - -mirroronly --force**. Isto remove apenas os volumes lógicos que são imagens-espelho do volume físico.

CAPÍTULO 8. EXIBIÇÃO DE COMPONENTES LVM

LVM oferece uma variedade de maneiras de exibir os componentes LVM, bem como de personalizar a exibição. Esta seção resume o uso dos comandos básicos de exibição do LVM.

8.1. EXIBIÇÃO DE INFORMAÇÕES LVM COM O COMANDO LVM

O comando **lvm** fornece várias opções embutidas que você pode usar para exibir informações sobre suporte e configuração de LVM.

- **lvm devtypes**
Exibe os tipos reconhecidos de dispositivos de blocos incorporados
- **lvm formats**
Exibe formatos de metadados reconhecidos.
- **lvm help**
Exibe o texto de ajuda da LVM.
- **lvm segtypes**
Exibe os tipos de segmentos de volume lógicos reconhecidos.
- **lvm tags**
Exibe quaisquer tags definidas neste host.
- **lvm version**
Exibe as informações da versão atual.

8.2. EXIBIÇÃO DE VOLUMES FÍSICOS

Há três comandos que você pode usar para exibir as propriedades dos volumes físicos da LVM: **pvs**, **pvdisplay**, e **pvscan**.

O comando **pvs** fornece informações de volume físico em uma forma configurável, exibindo uma linha por volume físico. O comando **pvs** fornece uma grande quantidade de controle de formato, e é útil para o script.

O comando **pvdisplay** fornece uma saída verbosa de múltiplas linhas para cada volume físico. Ele exibe as propriedades físicas (tamanho, extensões, grupo de volume, etc.) em um formato fixo.

O exemplo a seguir mostra a saída do comando **pvdisplay** para um único volume 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
```

O comando **pvscan** escaneia todos os dispositivos de bloco LVM suportados no sistema em busca de volumes físicos.

O comando a seguir mostra todos os 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 VG vg0 lvm2 [964.84 MB]
Total: 3 [2.83 GB] / in use: 2 [1.88 GB] / in no VG: 1 [964.84 MB]
```

Você pode definir um filtro no arquivo **lvm.conf** para que este comando evite a leitura de volumes físicos específicos.

8.3. EXIBIÇÃO DE GRUPOS DE VOLUME

Há dois comandos que você pode usar para exibir as propriedades dos grupos de volume LVM: **vgs** e **vgdisplay**. O comando **vgscan**, que verifica todos os dispositivos de bloco LVM suportados no sistema para grupos de volume, também pode ser usado para exibir os grupos de volume existentes.

O comando **vgs** fornece informações de grupo de volume em uma forma configurável, exibindo uma linha por grupo de volume. O comando **vgs** fornece um grande controle de formato, e é útil para a criação de scripts.

O comando **vgdisplay** exibe as propriedades do grupo de volume (tais como tamanho, extensões, número de volumes físicos, etc.) em uma forma fixa. O exemplo a seguir mostra a saída do comando **vgdisplay** para o grupo de volume **new_vg**. Se você não especificar um grupo de volume, todos os grupos de volume existentes são exibidos.

```
# 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
```

O exemplo a seguir mostra a saída do 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. EXIBINDO VOLUMES LÓGICOS

Há três comandos que você pode usar para exibir as propriedades dos volumes lógicos da LVM: **lvs**, **lvdisplay**, e **lvscan**.

O comando **lvs** fornece informações de volume lógico em uma forma configurável, exibindo uma linha por volume lógico. O comando **lvs** fornece uma grande quantidade de controle de formato, e é útil para o scripting.

O comando **lvdisplay** exibe as propriedades lógicas de volume (como tamanho, layout e mapeamento) em um formato fixo.

O seguinte comando mostra os atributos de **lvol2** em **vg00**. Se os volumes lógicos instantâneos foram criados para este volume lógico original, este comando mostra uma lista de todos os volumes lógicos instantâneos e seu status (ativo ou inativo) também.

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

O comando **lvscan** procura todos os volumes lógicos do sistema e os lista, como no exemplo a seguir.

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

CAPÍTULO 9. RELATÓRIOS PERSONALIZADOS PARA LVM

LVM fornece uma ampla gama de opções de configuração e linha de comando para produzir relatórios personalizados e para filtrar a saída do relatório. Para uma descrição completa dos recursos e capacidades do relatório LVM, consulte a página de manual **lvmreport(7)**.

Você pode produzir relatórios concisos e personalizáveis de objetos LVM com os comandos **pvs**, **lvs**, e **vgs**. Os relatórios que estes comandos geram incluem uma linha de saída para cada objeto. Cada linha contém uma lista ordenada de campos de propriedades relacionadas com o objeto. Há cinco maneiras de selecionar os objetos a serem relatados: por volume físico, grupo de volume, volume lógico, segmento de volume físico, e segmento de volume lógico.

Você pode relatar informações sobre volumes físicos, grupos de volume, volumes lógicos, segmentos de volume físico e segmentos lógicos de volume de uma só vez com o comando **lvm fullreport**. Para obter informações sobre este comando e suas capacidades, consulte a página de manual **lvm-fullreport(8)**.

LVM suporta relatórios de registro, que contêm um registro de operações, mensagens e status por objeto com identificação completa do objeto coletado durante a execução do comando LVM. Para maiores informações sobre o relatório de log LVM, veja a página de manual **lvmreport(7)**.

9.1. CONTROLE DO FORMATO DO VISOR DA LVM

Se você usa o comando **pvs**, **lvs**, ou **vgs** determina o conjunto padrão de campos exibidos e a ordem de ordenação. Você pode controlar a saída destes comandos com os seguintes argumentos:

- Você pode mudar quais campos são exibidos para algo diferente do padrão usando o argumento **-o**. Por exemplo, o seguinte comando exibe apenas o nome e o tamanho do volume físico.

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

- Você pode anexar um campo à saída com o sinal de mais (), que é usado em combinação com o argumento **-o**. O exemplo a seguir mostra a UUID do volume físico, além dos campos padrão.

```
# 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 yvfvZK-Cf31-j75k-dECm-0RZ3-0dGW-UqkCS
```

- A adição do argumento **-v** a um comando inclui alguns campos extras. Por exemplo, o comando **pvs -v** exibirá os campos **DevSize** e **PV UUID**, além dos campos padrão.

```
# 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
```

- O argumento **--noheadings** suprime a linha de títulos. Isto pode ser útil para escrever roteiros. O exemplo seguinte utiliza o argumento **--noheadings** em combinação com o argumento **pv_name**, que gerará uma lista de todos os volumes físicos.

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

- O **--separator separator** o argumento usa *separator* para separar cada campo. O exemplo a seguir separa os campos de saída padrão do comando **pvs** com um sinal de igual (=).

```
# 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 manter os campos alinhados ao usar o argumento **separator**, use o argumento **separator** em conjunto com o 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
```

Você pode usar o argumento **-P** do comando **lvs** ou **vgs** para exibir informações sobre um volume falho que de outra forma não apareceria na saída.

Para uma lista completa de argumentos de exibição, consulte as páginas de manual **pvs(8)**, **vgs(8)** e **lvs(8)**.

Os campos de grupo de volume podem ser misturados com campos de volume físico (e segmento de volume físico) ou com campos de volume lógico (e segmento de volume lógico), mas os campos de volume físico e de volume lógico não podem ser misturados. Por exemplo, o seguinte comando exibirá uma linha de saída para cada volume 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 EXIBIÇÃO DE OBJETOS LVM

Esta seção fornece uma série de tabelas que listam as informações que você pode exibir sobre os objetos LVM com os comandos **pvs**, **vgs**, e **lvs**.

Por conveniência, um prefixo de nome de campo pode ser descartado se corresponder ao padrão para o comando. Por exemplo, com o comando **pvs, name** significa **pv_name**, mas com o comando **vgs, name** é interpretado como **vg_name**.

Executar o seguinte comando é o equivalente a executar **pvs -o pv_free**.

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



NOTA

O número de caracteres nos campos de atributos em **pvs**, **vgs**, e **lvs** pode aumentar em lançamentos posteriores. Os campos de caracteres existentes não mudarão de posição, mas novos campos podem ser adicionados ao final. Você deve levar isto em consideração ao escrever scripts que buscam caracteres de atributo específicos, buscando o caractere com base em sua posição relativa até o início do campo, mas não por sua posição relativa até o final do campo. Por exemplo, para pesquisar o caractere **p** no nono bit do campo **lv_attr**, você poderia pesquisar a string `"^/.....p/"`, mas não deve pesquisar a string `"/*p$/"`.

Tabela 9.1, “Os campos de exibição do comando **pvs**” lista os argumentos de exibição do comando **pvs**, juntamente com o nome do campo como aparece na exibição do cabeçalho e uma descrição do campo.

Tabela 9.1. Os campos de exibição do comando **pvs**

Argumento	Cabeçalho	Descrição
dev_size	DevSize	O tamanho do dispositivo subjacente sobre o qual o volume físico foi criado
pe_start	1º PE	Offset até o início da primeira extensão física no dispositivo subjacente
pv_attr	Attr	Status do volume físico: (a)llocalizável ou e(x)portado.
pv_fmt	Fmt	O formato dos metadados do volume físico (lvm2 ou lvm1)
pv_free	PFree	O espaço livre restante no volume físico
pv_name	PV	O nome do volume físico
pv_pe_alloc_count	Atribuir	Número de extensões físicas utilizadas

Argumento	Cabeçalho	Descrição
pv_pe_count	PE	Número de extensões físicas
pvseg_size	SSize	O tamanho do segmento do volume físico
pvseg_start	Início	A extensão física inicial do segmento de volume físico
pv_size	PSize	O tamanho do volume físico
pv_tags	Etiquetas PV	Etiquetas LVM anexadas ao volume físico
pv_used	Usado	A quantidade de espaço atualmente utilizada no volume físico
pv_uuid	UUID PV	A UUID do volume físico

O comando **pvs** exibe os seguintes campos por padrão: **pv_name**, **vg_name**, **pv_fmt**, **pv_attr**, **pv_size**, **pv_free**. A exibição é 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
```

Usando o argumento **-v** com o comando **pvs** adiciona os seguintes campos à exibição padrão: **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 yvfvZK-Cf31-j75k-dECm-0RZ3-0dGW-tUqkCS
```

Você pode usar o argumento **--segments** do comando **pvs** para exibir informações sobre cada segmento de volume físico. Um segmento é um grupo de extensões. A visualização de um segmento pode ser útil se você quiser ver se seu volume lógico está fragmentado.

O comando **pvs --segments** exibe os seguintes campos por padrão: **pv_name**, **vg_name**, **pv_fmt**, **pv_attr**, **pv_size**, **pv_free**, **pvseg_start**, **pvseg_size**. A exibição é ordenada por **pv_name** e **pvseg_size** dentro do volume físico.

```
# pvs --segments
PV      VG    Fmt Attr PSize PFree Start SSize
```

```

/dev/hda2 VolGroup00 lvm2 a- 37.16G 32.00M 0 1172
/dev/hda2 VolGroup00 lvm2 a- 37.16G 32.00M 1172 16
/dev/hda2 VolGroup00 lvm2 a- 37.16G 32.00M 1188 1
/dev/sda1 vg lvm2 a- 17.14G 16.75G 0 26
/dev/sda1 vg lvm2 a- 17.14G 16.75G 26 24
/dev/sda1 vg lvm2 a- 17.14G 16.75G 50 26
/dev/sda1 vg lvm2 a- 17.14G 16.75G 76 24
/dev/sda1 vg lvm2 a- 17.14G 16.75G 100 26
/dev/sda1 vg lvm2 a- 17.14G 16.75G 126 24
/dev/sda1 vg lvm2 a- 17.14G 16.75G 150 22
/dev/sda1 vg lvm2 a- 17.14G 16.75G 172 4217
/dev/sdb1 vg lvm2 a- 17.14G 17.14G 0 4389
/dev/sdc1 vg lvm2 a- 17.14G 17.14G 0 4389
/dev/sdd1 vg lvm2 a- 17.14G 17.14G 0 4389
/dev/sde1 vg lvm2 a- 17.14G 17.14G 0 4389
/dev/sdf1 vg lvm2 a- 17.14G 17.14G 0 4389
/dev/sdg1 vg lvm2 a- 17.14G 17.14G 0 4389

```

Você pode usar o comando **pvs -a** para ver dispositivos detectados pela LVM que não foram inicializados como volumes físicos da LVM.

```

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

```

Tabela 9.2, “campos de exibição vgs” lista os argumentos de exibição do comando **vgs**, juntamente com o nome do campo como aparece na exibição do cabeçalho e uma descrição do campo.

Tabela 9.2. campos de exibição vgs

Argumento	Cabeçalho	Descrição
lv_count	#LV	O número de volumes lógicos que o grupo de volumes contém

Argumento	Cabeçalho	Descrição
max_lv	MaxLV	O número máximo de volumes lógicos permitidos no grupo de volume (0 se ilimitado)
max_pv	MaxPV	O número máximo de volumes físicos permitidos no grupo de volume (0 se ilimitado)
pv_count	#PV	O número de volumes físicos que definem o grupo de volumes
snap_count	#SN	O número de instantâneos que o grupo de volume contém
vg_attr	Attr	Status do grupo de volume: (w)riteable, (r)eadonly, resi(z)eable, e(x)ported, (p)artial e (c)lustered.
vg_extent_count	#Extremo	O número de extensões físicas no grupo de volume
vg_extent_size	Ext	O tamanho das extensões físicas no grupo de volume
vg_fmt	Fmt	O formato dos metadados do grupo de volume (lvm2 ou lvm1)
vg_free	VFree	Tamanho do espaço livre restante no grupo de volume
vg_free_count	Grátis	Número de extensões físicas livres no grupo de volume
vg_name	VG	O nome do grupo de volume
vg_seqno	Seq	Número que representa a revisão do grupo de volume
vg_size	VSize	O tamanho do grupo de volume
vg_sysid	SYS ID	LVM1 ID do sistema
vg_tags	Tags VG	Etiquetas LVM anexadas ao grupo de volume

Argumento	Cabeçalho	Descrição
vg_uuid	VG UUID	A UUID do grupo de volume

O comando **vgs** exibe os seguintes campos por padrão: **vg_name**, **pv_count**, **lv_count**, **snap_count**, **vg_attr**, **vg_size**, **vg_free**. A exibição é ordenada por **vg_name**.

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

Usando o argumento **-v** com o comando **vgs** adiciona os seguintes campos à exibição padrão: **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
```

Tabela 9.3, “campos de exibição de lvs” lista os argumentos de exibição do comando **lvs**, juntamente com o nome do campo como aparece na exibição do cabeçalho e uma descrição do campo.



NOTA

Em versões posteriores do Red Hat Enterprise Linux, a saída do comando **lvs** pode ser diferente, com campos adicionais na saída. A ordem dos campos, entretanto, permanecerá a mesma e quaisquer campos adicionais aparecerão no final da exibição.

Tabela 9.3. campos de exibição de lvs

Argumento	Cabeçalho	Descrição
* chunksize * chunk_size	Pedaço	Tamanho da unidade em um volume instantâneo
copy_percent	Cópia%	A porcentagem de sincronização de um volume lógico espelhado; também usada quando os extensões físicas estão sendo movidos com o comando pv_move
devices	Dispositivos	Os dispositivos subjacentes que compõem o volume lógico: os volumes físicos, os volumes lógicos e os extensões físicas e lógicas iniciais

Argumento	Cabeçalho	Descrição
lv_ancestors	Ancestrais	Para fotos de piscinas finas, os antepassados do volume lógico
lv_descendants	Descendentes	Para fotos de piscinas finas, os descendentes do volume lógico
lv_attr	Attr	<p>O status do volume lógico. Os bits do atributo volume lógico são os seguintes:</p> <p>* Bit 1: Tipo de volume: (m)irritado, (M)irritado sem sincronia inicial, (o)rigin, (O)rigin com foto de fusão, (r)aid, ®aid sem sincronia inicial, (s)napshot, (S)napshot, (p)vmove, (v)irtual, espelho ou raid (i)mage, espelho ou raid (l)mage out-of-sync, espelho (l)og dispositivo, sob (c)onversion, thin (V)olume, (t)hin pool, (T)hin pool data, raid ou thin pool m(e)tadata ou pool metadata spare,</p> <p>* Bit 2: Permissões: (w)riteable, (r)ead-only, ®ead-only activation of non-read-only volume</p> <p>* Bit 3: Política de alocação: (a)nywhere, (c)ontiguous, (i)nherited, c(l)ing, (n)ormal. Isto é capitalizado se o volume estiver atualmente bloqueado contra mudanças de alocação, por exemplo, durante a execução do comando pvmove.</p> <p>* Bit 4: fixo (m)inor</p> <p>* Bit 5: Estado: (a)ctive, (s)uspended, (l)invalid snapshot, invalid (S)uspended snapshot, snapshot (m)erge failed, suspended snapshot (M)erge failed, mapped (d)evice present without tables, mapped device present with (i)nactive table</p> <p>* Bit 6: dispositivo (o)caneta</p> <p>* Bit 7: Tipo de alvo: (m)irritor, (r)aid, (s)napshot, (t)hin,</p>

Argumento	Cabeçalho	Descrição
		<p>(u)known, (v)irtual. Isto agrupa volumes lógicos relacionados com o mesmo alvo central. Assim, por exemplo, imagens de espelho, logs de espelho assim como os próprios espelhos aparecem como (m) se eles usarem o driver do kernel do dispositivo original, enquanto que os equivalentes de raid usando o driver do kernel do md raid aparecem todos como (r). Instantâneos usando o driver do dispositivo original aparecem como (s), enquanto que instantâneos de volumes finos usando o driver de provisionamento fino aparecem como (t).</p> <p>* Bit 8: Os blocos de dados recém-alocados são sobrescritos com blocos de (z)eros antes do uso.</p> <p>* Bit 9: Volume Saúde: (p)artial, (r)efresh needed, (m)ismatches exist, (w)ritemostly. (p)artial significa que um ou mais dos Volumes Físicos que este Volume Lógico utiliza está faltando no sistema. (r)efresh significa que um ou mais dos Volumes Físicos que este Volume Lógico RAID utiliza sofreu um erro de escrita. O erro de escrita pode ser devido a uma falha temporária desse Volume Físico ou a uma indicação de que ele está falhando. O dispositivo deve ser atualizado ou substituído. (m)ismatches significa que o volume lógico RAID tem partes da matriz que não são coerentes. Inconsistências são descobertas ao iniciar uma operação em check em um volume lógico RAID. (As operações de scrubbing, check e repair, podem ser realizadas em um volume lógico RAID por meio do comando lvchange) (w)ritemostly significa os dispositivos em um volume lógico RAID 1 que foram marcados como sendo os mais gravados.</p>

Argumento	Cabeçalho	Descrição
		* Bit 10: ativação s(k)ip: este volume é sinalizado para ser pulado durante a ativação.
lv_kernel_major	KMaj	Número real do dispositivo principal do volume lógico (-1 se inativo)
lv_kernel_minor	KMIN	Número real do dispositivo menor do volume lógico (-1 se inativo)
lv_major	Maj	O número do dispositivo principal persistente do volume lógico (-1 se não especificado)
lv_minor	Min	O número de dispositivo menor persistente do volume lógico (-1 se não especificado)
lv_name	LV	O nome do volume lógico
lv_size	LSize	O tamanho do volume lógico
lv_tags	Tags LV	Etiquetas LVM anexadas ao volume lógico
lv_uuid	UUID LV	A UUID do volume lógico.
mirror_log	Log	Dispositivo sobre o qual reside o registro espelho
modules	Módulos	Objetivo correspondente do gerador de dispositivos do núcleo, necessário para usar este volume lógico
move_pv	Mudança	Fonte volume físico de um volume lógico temporário criado com o comando pvmove
origin	Origem	O dispositivo de origem de um volume instantâneo
* regionsize * region_size	Região	O tamanho da unidade de um volume lógico espelhado
seg_count	#Seg	O número de segmentos no volume lógico

Argumento	Cabeçalho	Descrição
seg_size	SSize	O tamanho dos segmentos no volume lógico
seg_start	Início	Deslocamento do segmento no volume lógico
seg_tags	Etiquetas de segmento	Etiquetas LVM anexadas aos segmentos do volume lógico
segtype	Tipo	O tipo de segmento de um volume lógico (por exemplo: espelho, listrado, linear)
snap_percent	Snap%	Porcentagem atual de um volume de instantâneo que está em uso
stripes	#Str	Número de listras ou espelhos em um volume lógico
* stripesize * stripe_size	Listra	Tamanho da unidade da listra em um volume lógico listrado

O comando **lvs** fornece a seguinte exibição por padrão. A exibição padrão é ordenada por **vg_name** e **lv_name** dentro do grupo de volume.

```
# 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
```

Um uso comum do comando **lvs** é anexar **devices** ao comando para exibir os dispositivos subjacentes que compõem o volume lógico. Este exemplo também especifica a opção **-a** para exibir os volumes internos que são componentes dos volumes lógicos, tais como espelhos RAID, entre parênteses. Este exemplo inclui um volume RAID, um volume listrado, e um volume ligeiramente polarizado.

```
# 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-a-or--- 1.00g /dev/sdf1(7041)
[raid1_rmeta_0] VG ewi-a-or--- 4.00m /dev/sde1(7040)
[raid1_rmeta_1] VG ewi-a-or--- 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)

```

Usando o argumento **-v** com o comando **lvs** adiciona os seguintes campos à exibição padrão: **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
lv00    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  lv00    0.20          1ye1OU-1clu-
o79k-20h2-ZGF0-qCJm-Cfbslx

```

Você pode usar o argumento **--segments** do comando **lvs** para exibir informações com colunas padrão que enfatizam as informações do segmento. Quando você usa o argumento **segments**, o prefixo **seg** é opcional. O comando **lvs --segments** exibe os seguintes campos por padrão: **lv_name**, **vg_name**, **lv_attr**, **stripes**, **segtype**, **seg_size**. A exibição padrão é ordenada por **vg_name**, **lv_name** dentro do grupo de volume, e **seg_start** dentro do volume lógico. Se os volumes lógicos estivessem fragmentados, a saída deste comando mostraria isso.

```

# 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

```

Usando o argumento **-v** com o comando **lvs --segments** adiciona os seguintes campos à exibição padrão: **seg_start**, **stripesize**, **chunksize**.

```

# lvs -v --segments
Finding all logical volumes
LV      VG      Attr  Start SSize #Str Type  Stripe Chunk
lv00    new_vg  owi-a- 0 52.00M 1 linear 0 0
newvgsnap1 new_vg  swi-a- 0 8.00M 1 linear 0 8.00K

```

O exemplo a seguir mostra a saída padrão do comando **lvs** em um sistema com um volume lógico configurado, seguido pela saída padrão do comando **lvs** com o 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. ORDENAÇÃO DE RELATÓRIOS LVM

Normalmente toda a saída do comando **lvs**, **vgs**, ou **pvs** tem que ser gerada e armazenada internamente antes de poder ser classificada e as colunas alinhadas corretamente. Você pode especificar o argumento **--unbuffered** para exibir a saída não classificada assim que ela for gerada.

Para especificar uma lista alternativa ordenada de colunas a serem ordenadas, use o argumento **-O** de qualquer um dos comandos do relatório. Não é necessário incluir estes campos dentro da própria saída.

O exemplo a seguir mostra a saída do comando **pvs** que exibe o nome do volume físico, tamanho e espaço livre.

```
# 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
```

O exemplo a seguir mostra a mesma saída, ordenada pelo campo de espaço livre.

```
# 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
```

O exemplo a seguir mostra que você não precisa exibir o campo no qual você está classificando.

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

Para exibir uma ordenação inversa, preceda um campo especificado após o argumento **-O** com o caracter **-**.

```
# 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. ESPECIFICAÇÃO DAS UNIDADES PARA A EXIBIÇÃO DE UM RELATÓRIO LVM

Para especificar as unidades para a exibição do relatório da LVM, use o argumento **--units** do comando do relatório. Você pode especificar (b)ytes, (k)ilobytes, (m)egabytes, (g)igabytes, (t)erabytes, (e)xabytes, (p)etabytes, e (h)uman-readable. A exibição padrão é legível por humanos. Você pode anular o padrão definindo o parâmetro **units** na seção **global** do arquivo **/etc/lvm/lvm.conf**.

O exemplo a seguir especifica a saída do comando **pvs** em megabytes ao invés dos gigabytes padrão.

```
# 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 padrão, as unidades são exibidas em potências de 2 (múltiplos de 1024). Pode-se especificar que as unidades sejam exibidas em múltiplos de 1000, capitalizando a especificação da unidade (B, K, M, G, T, H).

O seguinte comando exibe a saída como um múltiplo de 1024, o comportamento padrão.

```
# 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
```

O seguinte comando exibe a saída como um 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
```

Você também pode especificar (s)ectores (definidos como 512 bytes) ou unidades personalizadas.

O exemplo a seguir mostra a saída do comando **pvs** como um número de setores.

```
# 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
```

O exemplo a seguir mostra a saída do comando **pvs** em 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. EXIBIÇÃO DA SAÍDA DO COMANDO LVM EM FORMATO JSON

Você pode usar a opção **--reportformat** dos comandos de exibição da LVM para exibir a saída no formato JSON.

O exemplo a seguir mostra a saída do **lvs** no formato padrão padrão.

```
# 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

```

O comando a seguir mostra a saída da mesma configuração LVM quando você especifica o 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":""}
      ]
    }
  ]
}

```

Você também pode definir o formato do relatório como uma opção de configuração no arquivo `/etc/lvm/lvm.conf`, usando a configuração `output_format`. A configuração `--reportformat` da linha de comando, entretanto, tem precedência sobre esta configuração.

9.6. EXIBIÇÃO DO LOG DE COMANDO LVM

Tanto os comandos LVM orientados a relatórios como os orientados a processos podem relatar o log de comando se isso for ativado com a configuração `log/report_command_log`. Você pode determinar o conjunto de campos a serem exibidos e ordenados para este relatório.

Os exemplos a seguir configuram o LVM para gerar um relatório de registro completo para os comandos LVM. Neste exemplo, você pode ver que ambos os volumes lógicos `lvvol0` e `lvvol1` foram processados com sucesso, assim como o grupo de volumes `VG` que contém os volumes.

lvmconfig --type full log/command_log_selection

```
command_log_selection="all"
```

lvs

```
Logical Volume
```

```
=====
```

```
LV   LSize Cpy%Sync
```

```
lvvol1 4.00m 100.00
```

```
lvvol0 4.00m
```

```
Command Log
```

```
=====
```

```
Seq LogType Context  ObjType ObjName ObjGrp  Msg  Errno RetCode
```

```
1 status processing lv  lvvol0 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 mais informações sobre a configuração de relatórios e logs de comando LVM, consulte a página de manual **lvreport**.

CAPÍTULO 10. CONFIGURAÇÃO DE VOLUMES LÓGICOS RAID

Você pode criar, ativar, alterar, remover, exibir e usar volumes LVM RAID.

10.1. VOLUMES LÓGICOS RAID

O LVM suporta níveis RAID 0, 1, 4, 5, 6, e 10.

Um volume LVM RAID tem as seguintes características:

- Os volumes lógicos RAID criados e gerenciados pela LVM alavancam os drivers de kernel de múltiplos dispositivos (MD).
- Você pode dividir temporariamente as imagens RAID1 da matriz e fundi-las novamente na matriz mais tarde.
- Os volumes LVM RAID suportam snapshots.

Aglomerados

Os volumes lógicos RAID não são sensíveis a aglomerações.

Embora você possa criar e ativar volumes lógicos RAID exclusivamente em uma máquina, você não pode ativá-los simultaneamente em mais de uma máquina.

Subvolumes

Quando você cria um volume lógico RAID, o LVM cria um subvolume de metadados que é uma extensão em tamanho para cada dado ou subvolume de paridade na matriz.

Por exemplo, a criação de uma matriz RAID1 de duas vias resulta em dois subvolumes de metadados (**lv_rmeta_0** e **lv_rmeta_1**) e dois subvolumes de dados (**lv_rimage_0** e **lv_rimage_1**). Da mesma forma, criar uma faixa de 3 vias (mais 1 dispositivo de paridade implícita) RAID4 resulta em 4 subvolumes de metadados (**lv_rmeta_0**, **lv_rmeta_1**, **lv_rmeta_2**, e **lv_rmeta_3**) e 4 subvolumes de dados (**lv_rimage_0**, **lv_rimage_1**, **lv_rimage_2**, e **lv_rimage_3**).

Integridade

Você pode perder dados quando um dispositivo RAID falha ou quando ocorre corrupção suave. A corrupção suave no armazenamento de dados implica que os dados recuperados de um dispositivo de armazenamento são diferentes dos dados gravados naquele dispositivo. A adição de integridade a um RAID LV ajuda a mitigar ou prevenir a corrupção branda. Para saber mais sobre corrupção branda e como adicionar integridade a um RAID LV, consulte [Seção 10.6, "Usando integridade DM com RAID LV"](#).

10.2. NÍVEIS RAID E SUPORTE LINEAR

RAID suporta várias configurações, incluindo níveis 0, 1, 4, 5, 6, 10, e linear. Estes tipos de RAID são definidos como segue:

Nível 0

RAID nível 0, muitas vezes chamado *striping*, é uma técnica de mapeamento de dados com faixas orientadas para o desempenho. Isto significa que os dados que estão sendo escritos na matriz são divididos em faixas e escritos nos discos dos membros da matriz, permitindo um alto desempenho de E/S a baixo custo inerente, mas não oferece redundância.

Muitas implementações de nível 0 RAID apenas riscam os dados através dos dispositivos membros até o tamanho do menor dispositivo da matriz. Isto significa que se você tiver vários dispositivos com

tamanhos ligeiramente diferentes, cada dispositivo é tratado como se fosse do mesmo tamanho que o menor drive. Portanto, a capacidade de armazenamento comum de uma matriz de nível 0 é igual à capacidade do menor disco membro em um RAID de Hardware ou a capacidade da menor partição membro em um RAID de Software multiplicada pelo número de discos ou partições da matriz.

Nível 1

RAID nível 1, ou *mirroring*, fornece redundância ao escrever dados idênticos para cada disco membro da matriz, deixando uma cópia "espelhada" em cada disco. O espelhamento permanece popular devido a sua simplicidade e alto nível de disponibilidade de dados. O nível 1 opera com dois ou mais discos, e oferece muito boa confiabilidade dos dados e melhora o desempenho para aplicações de leitura intensiva, mas a um custo relativamente alto.

O nível RAID 1 tem um custo elevado porque você escreve as mesmas informações em todos os discos da matriz, fornece confiabilidade de dados, mas de uma maneira muito menos eficiente em termos de espaço do que os níveis RAID baseados na paridade, como o nível 5. Entretanto, esta ineficiência de espaço vem com um benefício de desempenho: os níveis RAID baseados em paridade consomem consideravelmente mais energia da CPU para gerar a paridade enquanto o nível RAID 1 simplesmente grava os mesmos dados mais de uma vez para os múltiplos membros RAID com muito pouca sobrecarga de CPU. Como tal, o nível RAID 1 pode superar os níveis RAID baseados na paridade em máquinas onde o RAID por software é empregado e os recursos da CPU na máquina são tributados consistentemente com operações que não sejam atividades RAID.

A capacidade de armazenamento da matriz de nível 1 é igual à capacidade do menor disco rígido espelhado em um RAID de hardware ou a menor partição espelhada em um RAID de software. A redundância de nível 1 é a maior possível entre todos os tipos de RAID, sendo que a matriz pode operar com apenas um único disco presente.

Nível 4

O nível 4 utiliza a paridade concentrada em um único drive de disco para proteger os dados. A informação da paridade é calculada com base no conteúdo do resto dos discos membros da matriz. Esta informação pode então ser usada para reconstruir os dados quando um disco da matriz falhar. Os dados reconstruídos podem então ser usados para satisfazer as solicitações de E/S para o disco com falha antes de ser substituído e para repovoar o disco com falha após a sua substituição. Como o disco de paridade dedicado representa um gargalo inerente em todas as transações de gravação no array RAID, o nível 4 raramente é usado sem tecnologias de acompanhamento, como o cache de gravação, ou em circunstâncias específicas onde o administrador do sistema está intencionalmente projetando o dispositivo RAID de software com este gargalo em mente (como um array que terá poucas ou nenhuma transações de gravação uma vez que o array estiver preenchido com dados). O RAID nível 4 é tão raramente usado que não está disponível como opção no Anaconda. Entretanto, ele pode ser criado manualmente pelo usuário, se realmente necessário.

A capacidade de armazenamento do Hardware RAID nível 4 é igual à capacidade da menor partição membro multiplicada pelo número de partições *minus one*. O desempenho de uma matriz RAID nível 4 é sempre assimétrico, o que significa que a leitura é melhor que a escrita. Isto porque as gravações consomem CPU extra e largura de banda da memória principal ao gerar paridade, e também consomem largura de banda extra do barramento ao escrever os dados reais em discos, porque você está escrevendo não apenas os dados, mas também a paridade. As leituras só precisam ler os dados e não a paridade, a menos que a matriz esteja em um estado degradado. Como resultado, as leituras geram menos tráfego para os discos e através dos ônibus do computador para a mesma quantidade de transferência de dados em condições normais de operação.

Nível 5

Este é o tipo mais comum de RAID. Ao distribuir a paridade entre todas as unidades de disco de um array, o RAID nível 5 elimina o gargalo de gravação inerente ao nível 4. O único gargalo de desempenho é o próprio processo de cálculo da paridade. Com CPUs e software RAID modernos,

isso normalmente não é um gargalo, uma vez que as CPUs modernas podem gerar paridade muito rapidamente. Entretanto, se você tiver um número suficientemente grande de dispositivos membros em uma matriz RAID5 de software, de tal forma que a velocidade de transferência de dados combinados entre todos os dispositivos seja suficientemente alta, então este gargalo pode começar a entrar em jogo.

Assim como no nível 4, o nível 5 tem um desempenho assimétrico e lê textos com um desempenho substancialmente superior ao desempenho. A capacidade de armazenamento do RAID nível 5 é calculada da mesma forma que com o nível 4.

Nível 6

Este é um nível comum de RAID quando a redundância e preservação de dados, e não o desempenho, são as principais preocupações, mas onde a ineficiência de espaço do nível 1 não é aceitável. O nível 6 utiliza um esquema de paridade complexo para ser capaz de se recuperar da perda de quaisquer dois drives da matriz. Este esquema de paridade complexo cria uma carga de CPU significativamente maior nos dispositivos RAID de software e também impõe uma carga maior durante as transações de gravação. Como tal, o nível 6 é consideravelmente mais assimétrico no desempenho do que os níveis 4 e 5.

A capacidade total de uma matriz RAID nível 6 é calculada de forma semelhante ao RAID nível 5 e 4, exceto que você deve subtrair 2 dispositivos (ao invés de 1) da contagem do dispositivo para o espaço de armazenamento da paridade extra.

Nível 10

Este nível RAID tenta combinar as vantagens de desempenho do nível 0 com a redundância do nível 1. Ele também ajuda a aliviar parte do espaço desperdiçado em matrizes de nível 1 com mais de 2 dispositivos. Com o nível 10, é possível, por exemplo, criar uma matriz de 3 drives configurada para armazenar apenas 2 cópias de cada pedaço de dados, o que então permite que o tamanho total da matriz seja 1,5 vezes o tamanho dos menores dispositivos em vez de apenas igualar o menor dispositivo (como seria com um array de 3 dispositivos, nível 1). Isto evita o uso do processo de CPU para calcular a paridade como com o RAID nível 6, mas é menos eficiente em termos de espaço. A criação de RAID nível 10 não é suportada durante a instalação. É possível criar um manualmente após a instalação.

RAID Linear

O RAID linear é um agrupamento de drives para criar um drive virtual maior.

Em RAID linear, os pedaços são alocados sequencialmente de um drive membro, indo para o próximo drive somente quando o primeiro é completamente preenchido. Este agrupamento não oferece nenhum benefício de desempenho, pois é improvável que qualquer operação de E/S seja dividida entre os drives membros. O RAID linear também não oferece redundância e diminui a confiabilidade. Se algum drive membro falhar, não será possível utilizar todo o conjunto. A capacidade é o total de todos os discos dos membros.

10.3. TIPOS DE SEGMENTOS RAID LVM

Para criar um volume lógico RAID, você especifica um tipo de raid como o argumento **--type** do comando **lvcreate**. A tabela a seguir descreve os possíveis tipos de segmentos RAID.

Para a maioria dos usuários, especificar um dos cinco tipos primários disponíveis (**raid1**, **raid4**, **raid5**, **raid6**, **raid10**) deve ser suficiente.

Tabela 10.1. Tipos de segmentos RAID LVM

Tipo de segmento	Descrição
raid1	Espelhamento RAID1. Este é o valor padrão para o argumento --type do comando lvcreate quando você especifica o -m , mas não especifica o striping.
raid4	Disco de paridade dedicado RAID4
raid5	O mesmo que raid5_ls
raid5_la	<ul style="list-style-type: none"> ● RAID5 assimétrico esquerdo. ● Paridade rotativa 0 com continuação de dados
raid5_ra	<ul style="list-style-type: none"> ● RAID5 assimétrico direito. ● Paridade rotativa N com continuação de dados
raid5_ls	<ul style="list-style-type: none"> ● RAID5 saiu simétrico. ● Paridade rotativa 0 com reinício dos dados
raid5_rs	<ul style="list-style-type: none"> ● RAID5 simétrico à direita. ● Paridade rotativa N com reinício dos dados
raid6	O mesmo que raid6_zr
raid6_zr	<ul style="list-style-type: none"> ● RAID6 reinício zero ● Paridade rotativa zero (da esquerda para a direita) com reinício dos dados
raid6_nr	<ul style="list-style-type: none"> ● RAID6 N reinício ● Paridade rotativa N (da esquerda para a direita) com reinício dos dados
raid6_nc	<ul style="list-style-type: none"> ● RAID6 N continuar ● Paridade rotativa N (da esquerda para a direita) com continuação dos dados

Tipo de segmento	Descrição
raid10	<ul style="list-style-type: none"> • Espelhos listrados. Este é o valor padrão para o argumento --type do comando lvcreate se você especificar o -m e especificar um número de listras que seja maior que 1. • Listração de conjuntos de espelhos
raid0/raid0_meta	Striping. RAID0 espalha dados lógicos de volume através de múltiplos subvolumes de dados em unidades de tamanho de listra. Isto é usado para aumentar o desempenho. Os dados de volume lógico serão perdidos se qualquer um dos subvolumes de dados falhar.

10.4. CRIAÇÃO DE VOLUMES LÓGICOS RAID

Esta seção fornece exemplos de comandos que criam diferentes tipos de volume lógico RAID.

Você pode criar matrizes RAID1 com diferentes números de cópias de acordo com o valor especificado para o argumento **-m**. Da mesma forma, você especifica o número de faixas para um volume lógico RAID 4/5/6 com o **-i argument**. Você também pode especificar o tamanho das faixas com o argumento **-l**.

O seguinte comando cria uma matriz RAID1 de 2 vias chamada **my_lv** no grupo de volume **my_vg** que é de um gigabyte em tamanho.

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

O seguinte comando cria uma matriz RAID5 (3 stripes 1 unidade de paridade implícita) chamada **my_lv** no grupo de volume **my_vg** que é um gigabyte em tamanho. Observe que você especifica o número de faixas exatamente como faz para um volume de faixas LVM; o número correto de unidades de paridade é adicionado automaticamente.

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

O seguinte comando cria um array RAID6 (3 stripes 2 unidades de paridade implícita) chamado **my_lv** no grupo de volume **my_vg** que é um gigabyte em tamanho.

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

10.5. CRIAÇÃO DE UM VOLUME LÓGICO RAID0 (LISTRADO)

Um volume lógico RAID0 espalha dados lógicos de volume através de múltiplos subvolumes de dados em unidades de tamanho de faixa.

O formato para o comando para criar um volume RAID0 é o seguinte.

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

Tabela 10.2. Parâmetros de criação de comandos RAID0

Parâmetro	Descrição
--type raid0[_meta]	A especificação raid0 cria um volume RAID0 sem volumes de metadados. Especificando raid0_meta cria um volume RAID0 com volumes de metadados. Como o RAID0 não é resistente, ele não precisa armazenar nenhum bloco de dados espelhado como RAID1/10 ou calcular e armazenar qualquer bloco de paridade como o RAID4/5/6. Portanto, ele não precisa de volumes de metadados para manter o estado sobre o progresso da ressincronização dos blocos espelhados ou de paridade. Os volumes de metadados tornam-se obrigatórios em uma conversão de RAID0 para RAID4/5/6/10, no entanto, e especificando raid0_meta pré-aloca esses volumes de metadados para evitar uma respectiva falha de alocação.
--stripes <i>Stripes</i>	Especifica o número de dispositivos para distribuir o volume lógico.
--stripesize <i>StripeSize</i>	Especifica o tamanho de cada listra em kilobytes. Esta é a quantidade de dados que é escrita em um dispositivo antes de passar para o dispositivo seguinte.
<i>VolumeGroup</i>	Especifica o grupo de volume a ser utilizado.
<i>PhysicalVolumePath...</i>	Especifica os dispositivos a serem utilizados. Se isto não for especificado, a LVM escolherá o número de dispositivos especificados pela opção <i>Stripes</i> , um para cada faixa.

Este procedimento de exemplo cria um volume lógico LVM RAID0 chamado **mylv** que risca os dados nos discos em **/dev/sda1**, **/dev/sdb1**, e **/dev/sdc1**.

1. Rotule os discos que você usará no grupo de volume como volumes físicos LVM com o comando **pvcreate**.



ATENÇÃO

Este comando destrói quaisquer dados em **/dev/sda1**, **/dev/sdb1**, e **/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. Criar o grupo de volume **myvg**. O seguinte comando cria o grupo de volume **myvg**.

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

Você pode usar o comando **vgs** para exibir os atributos do novo grupo de volume.

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

3. Crie um volume lógico RAID0 a partir do grupo de volume que você criou. O seguinte comando cria o volume RAID0 **mylv** a partir do grupo de volume **myvg**. Este exemplo cria um volume lógico de 2 gigabytes de tamanho, com três faixas e uma faixa 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. Criar um sistema de arquivo no volume lógico RAID0. O seguinte comando cria um sistema de arquivo em **ext4** no volume 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
```

Os seguintes comandos montam o volume lógico e informam o uso do espaço em disco do sistema de arquivos.

```
# 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. USANDO INTEGRIDADE DM COM RAID LV

Enquanto o RAID ajuda a evitar a perda de dados quando um dispositivo falha, o uso da integridade do mapeador de dispositivos (DM) com um RAID LV reduz o risco de perda de dados quando os dados em um dispositivo RAID são corrompidos. Leia esta seção para saber mais sobre como você pode proteger seus dados contra corrupção suave com a integridade do DM.

10.6.1. Proteção de dados com integridade de DM

Dependendo do tipo de configuração, um RAID LV evita a perda de dados quando um dispositivo falha. Se um dispositivo que compreende uma matriz RAID falhar, os dados podem ser recuperados de outros dispositivos que fazem parte desse RAID LV. Entretanto, uma configuração RAID não garante a integridade dos dados em si. Corrupção leve, corrupção silenciosa, erros leves e erros silenciosos são termos que descrevem dados que se tornaram corrompidos, mesmo que o projeto do sistema e o software continuem a funcionar como esperado.

A corrupção suave no armazenamento de dados implica que os dados recuperados de um dispositivo de armazenamento são diferentes dos dados escritos para aquele dispositivo. Os dados corrompidos podem existir indefinidamente nos dispositivos de armazenamento. Você pode não descobrir estes dados corrompidos até que recupere e tente usar estes dados.

A integridade é usada com níveis RAID 1, 4, 5, 6 e 10 para ajudar a mitigar ou prevenir a perda de dados devido à corrupção suave. A camada RAID garante que uma cópia não corrompida dos dados possa corrigir os erros de corrupção suave. A camada de integridade fica acima de cada imagem RAID enquanto um sub LV extra armazena os metadados de integridade (checksums de dados) para cada imagem RAID. Quando se recupera dados de um RAID LV com integridade, os checksums dos dados de integridade analisam os dados quanto à corrupção. Se for detectada corrupção, a camada de integridade retorna uma mensagem de erro, e a camada RAID recupera uma cópia não corrompida dos dados de outra imagem RAID. A camada RAID reescreve automaticamente os dados não corrompidos sobre os dados corrompidos para reparar a corrupção suave.

Você pode adicionar integridade de DM a um RAID LV ao criá-lo, ou pode adicionar integridade de DM a um RAID LV que já existe. Ao criar um RAID LV com integridade, ou adicionar integridade a um já existente, é necessário espaço de armazenamento adicional para os metadados de integridade. Para cada imagem RAID, cada 500MB de dados requer 4MB de espaço de armazenamento adicional para armazenar os metadados de integridade.

10.6.1.1. Considerações ao adicionar a integridade de DM

Ao criar um novo RAID LV com integridade DM ou adicionar integridade a um RAID LV existente, certas considerações se aplicam:

- A integridade do DM requer espaço de armazenamento adicional porque adiciona checksums aos dados.
- Enquanto algumas configurações RAID são mais impactadas do que outras, adicionar a integridade da DM afeta o desempenho devido à latência ao acessar os dados. Uma configuração RAID1 normalmente oferece melhor desempenho do que o RAID5 ou suas variantes.
- O tamanho do bloco de integridade RAID também impacta o desempenho. A configuração de um bloco de integridade RAID de tamanho maior oferece melhor desempenho. Entretanto, um tamanho de bloco de integridade RAID menor oferece maior retrocompatibilidade.
- Há dois modos de integridade disponíveis: bitmap ou diário. O modo de integridade bitmap normalmente oferece melhor desempenho do que o modo de diário.

DICA

Se você tiver problemas de desempenho, recomendamos que você use RAID1 com integridade ou que teste o desempenho de uma configuração RAID particular para garantir que ela atenda às suas exigências.

10.6.2. Criando um RAID LV com integridade DM

Quando você cria um RAID LV, adicionar a integridade do DM ajuda a mitigar o risco de perda de dados devido à corrupção suave.

Pré-requisitos

- Você deve ter acesso à raiz.

Procedimento

- Para criar um RAID LV com integridade DM, execute o seguinte comando:
lvcreate --type <raid-level> --raidintegrity y -L <usable-size> -n <logical-volume> <volume-group>

onde

<raid-level>

Especifica o nível RAID do RAID LV que você deseja criar.

<usable-size>

Especifica o tamanho utilizável em MB.

<logical-volume>

Especifica o nome do LV que você deseja criar.

<volume-group>

Especifica o nome do grupo de volume que você deseja criar o RAID LV sob.

No exemplo a seguir, criamos um RAID LV com integridade chamado **test-lv** no grupo de volume **test-vg**, com um tamanho utilizável de 256M e RAID nível 1.

Exemplo RAID LV com integridade

```
# 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. Adicionando a integridade de DM a um RAID LV existente

Você pode adicionar integridade de DM a um RAID LV existente para ajudar a mitigar o risco de perda de dados devido à corrupção suave.

Pré-requisitos

- Você deve ter acesso à raiz.

Procedimento

- Para adicionar integridade de DM a um RAID LV existente, execute o seguinte comando:
lvconvert --raidintegrity y <volume-group>/<logical-volume>

onde

<volume-group>

Especifica o nome do grupo de volume que você deseja criar o RAID LV sob.

<logical-volume>

Especifica o nome do LV que você deseja criar.

10.6.4. Removendo a integridade de um RAID LV

A adição de integridade a um RAID LV limita o número de operações que você pode realizar nesse RAID LV. Portanto, você deve remover a integridade antes de executar determinadas operações.

Pré-requisitos

- Você deve ter acesso à raiz.

Procedimento

Para remover a integridade de um RAID LV, execute o seguinte comando:

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

<volume-group>

Especifica o nome do grupo de volume que você deseja criar o RAID LV sob.

<logical-volume>

Especifica o nome do LV que você deseja criar.

10.6.5. Visualizando informações sobre integridade da DM

Quando você criar um RAID LV com integridade ou quando adicionar integridade a um RAID LV existente, use o seguinte comando para visualizar informações sobre a integridade:

lvs -a <volume-group>

onde <volume-group> é o nome do grupo de volume que contém o RAID LV com integridade.

O exemplo a seguir mostra informações sobre o **test-lv** RAID LV que foi criado no grupo de volume **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-aor--- 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-aor--- 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-aor--- 4.00m
[test-lv_rmeta_1] test-vg ewi-aor--- 4.00m
```

Sincronização

Quando você criar um RAID LV com integridade ou adicionar integridade a um RAID LV existente, recomendamos que você aguarde a sincronização da integridade e os metadados do RAID para completar antes de usar o LV. Caso contrário, a inicialização em segundo plano pode afetar o desempenho do LV. A coluna **Cpy%Sync** indica o progresso da sincronização tanto para o RAID LV de nível superior quanto para cada imagem RAID. A imagem RAID é indicada na coluna LV por **raid_image_N**. Consulte a coluna LV para garantir que o progresso da sincronização exiba **100%** para o nível superior do RAID LV e para cada imagem RAID.

Imagens RAID usando integridade

O atributo **g** nos atributos listados sob a coluna **Attr** indica que a imagem RAID está usando integridade. Os checksums de integridade são armazenados no **_imeta** RAID LV.

Para exibir o tipo para cada RAID LV, adicione a opção **-o segtype** ao 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 integridade

Há um contador incremental que conta o número de desajustes detectados em cada imagem RAID. Para visualizar os descasamentos de dados detectados pela integridade em uma determinada imagem RAID, execute o seguinte comando:

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

onde

<volume-group>

Especifica o nome do grupo de volume que você deseja criar o RAID LV sob.

<logical-volume>

Especifica o nome do LV que você deseja criar.

<n>

Especifica a imagem RAID para a qual você deseja visualizar as informações de descoordenação de integridade.

Você deve executar o comando para cada imagem RAID que você deseja visualizar. No exemplo a seguir, veremos os desajustes de dados de **rimage_0** sob **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-a-or--- 256.00m [test-lv_rimage_0_iorig] 100.00          0
```

Podemos ver que a integridade não detectou nenhum desajuste de dados e, portanto, o contador **IntegMismatches** mostra zero (0).

Desajustes de integridade nos logs de mensagens do kernel

Você também pode encontrar informações de integridade de dados nos logs de mensagens do kernel, como mostrado nos exemplos a seguir.

Example of dm-integrity mismatches from the kernel message logs

```
dispositivo-mestre: integridade: dm-12: Checksum falhou no setor 0x24e7
```

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

```
md/raid1:mdX: erro de leitura corrigido (8 setores em 9448 em dm-16)
```

10.6.6. Recursos adicionais

- Para mais informações sobre todas as opções disponíveis, consulte a(s) página(s) de comando **lvraid**.

10.7. CONTROLAR A TAXA DE INICIALIZAÇÃO DOS VOLUMES RAID

Quando você cria volumes lógicos RAID10, as E/S de fundo necessárias para inicializar os volumes lógicos com uma operação **sync** podem excluir outras operações de E/S para dispositivos LVM, tais como atualizações de metadados de grupos de volumes, particularmente quando você está criando muitos volumes lógicos RAID. Isto pode fazer com que as outras operações LVM diminuam a velocidade.

Você pode controlar a taxa na qual um volume lógico RAID é inicializado, implementando a aceleração da recuperação. Você controla a taxa na qual as operações do **sync** são realizadas definindo a taxa mínima e máxima de E/S para essas operações com as opções **--minrecoveryrate** e **--maxrecoveryrate** do comando **lvcreate**. Você especifica estas opções da seguinte forma.

- **--maxrecoveryrate *Rate*[bBsSkKmMgG]**
Define a taxa máxima de recuperação para um volume lógico RAID, de modo que não exclua as operações de E/S nominais. O *Rate* é especificado como uma quantidade por segundo para cada dispositivo da matriz. Se nenhum sufixo for dado, então o kiB/sec/dispositivo é assumido. Ajustar a taxa de recuperação para 0 significa que ela não será limitada.
- **--minrecoveryrate *Rate*[bBsSkKmMgG]**
Define a taxa mínima de recuperação para um volume lógico RAID para garantir que as operações de E/S para **sync** alcancem um rendimento mínimo, mesmo quando a E/S nominal pesada está presente. O *Rate* é especificado como uma quantidade por segundo para cada dispositivo da matriz. Se nenhum sufixo for dado, então o kiB/sec/dispositivo é assumido.

O seguinte comando cria uma matriz RAID10 de 2 vias com 3 faixas de tamanho de 10 gigabytes com uma taxa máxima de recuperação de 128 kiB/seg/dispositivo. O array é denominado **my_lv** e está no grupo de volume **my_vg**.

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

Você também pode especificar taxas de recuperação mínimas e máximas para uma operação de lavagem RAID.

10.8. CONVERSÃO DE UM DISPOSITIVO LINEAR EM UM DISPOSITIVO RAID

Você pode converter um volume lógico linear existente para um dispositivo RAID usando o argumento **--type** do comando **lvconvert**.

O seguinte comando converte o volume lógico linear **my_lv** no grupo de volume **my_vg** para uma matriz RAID1 de 2 vias.

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

Como os volumes lógicos RAID são compostos de pares de metadados e subvolume de dados, quando você converte um dispositivo linear para uma matriz RAID1, um novo subvolume de metadados é criado e associado ao volume lógico original em (um dos) mesmos volumes físicos em que o volume linear está. As imagens adicionais são adicionadas em pares de metadados/subvolume de dados. Por exemplo, se o dispositivo original for o seguinte:

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

Após a conversão para uma matriz RAID1 de 2 vias, o dispositivo contém os seguintes pares de subvolumes de dados e metadados:

```
# 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)
```

Se a imagem dos metadados que se parecia com o volume lógico original não puder ser colocada no mesmo volume físico, o **lvconvert** falhará.

10.9. CONVERSÃO DE UM VOLUME LÓGICO LVM RAID1 PARA UM VOLUME LÓGICO LINEAR LVM

Você pode converter um volume lógico RAID1 LVM existente para um volume lógico linear LVM com o comando **lvconvert**, especificando o argumento **-m0**. Isto remove todos os subvolumes de dados RAID e todos os subvolumes de metadados RAID que compõem a matriz RAID, deixando a imagem RAID1 de nível superior como o volume lógico linear.

O exemplo a seguir mostra um volume 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)
```

O seguinte comando converte o volume lógico LVM RAID1 **my_vg/my_lv** para um dispositivo linear LVM.

■

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

Ao converter um volume lógico LVM RAID1 para um volume linear LVM, é possível especificar quais volumes físicos devem ser removidos. O seguinte exemplo mostra o layout de um volume lógico LVM RAID1 composto de duas imagens: `/dev/sda1` e `/dev/sdb1`. Neste exemplo, o comando **lvconvert** especifica que você deseja remover `/dev/sda1`, deixando `/dev/sdb1` como o volume físico que compõe o dispositivo linear.

```
# 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. CONVERSÃO DE UM DISPOSITIVO LVM ESPELHADO EM UM DISPOSITIVO RAID1

Você pode converter um dispositivo LVM espelhado existente com um tipo de segmento **mirror** para um dispositivo LVM RAID1 com o comando **lvconvert**, especificando o argumento **--type raid1**. Isto renomeia os subvolumes espelhados (**mimage**) para subvolumes RAID (**rimage**). Além disso, o registro espelho é removido e os subvolumes de metadados (**rmeta**) são criados para os subvolumes de dados nos mesmos volumes físicos que os subvolumes de dados correspondentes.

O exemplo a seguir mostra o layout de um volume lógico espelhado **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)
```

O seguinte comando converte o volume lógico espelhado **my_vg/my_lv** para um volume 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. REDIMENSIONAMENTO DE UM VOLUME LÓGICO RAID

Você pode redimensionar um volume lógico RAID das seguintes maneiras;

- Você pode aumentar o tamanho de um volume lógico RAID de qualquer tipo com o comando **lvresize** ou **lvextend**. Isto não altera o número de imagens RAID. Para volumes lógicos RAID listrados aplicam-se as mesmas restrições de arredondamento de listras que quando se cria um volume lógico RAID listrado.
- Você pode reduzir o tamanho de um volume lógico RAID de qualquer tipo com o comando **lvresize** ou **lvreduce**. Isto não altera o número de imagens RAID. Como com o comando **lvextend**, aplicam-se as mesmas restrições de arredondamento de listras que quando se cria um volume lógico RAID listrado.
- Você pode alterar o número de faixas em um volume lógico RAID listrado (**raid4/5/6/10**) com o parâmetro **--stripes N** do comando **lvconvert**. Isto aumenta ou reduz o tamanho do volume lógico RAID pela capacidade das listras adicionadas ou removidas. Observe que os volumes **raid10** são capazes apenas de adicionar listras. Esta capacidade é parte do recurso RAID *reshaping* que permite alterar os atributos de um volume lógico RAID mantendo o mesmo nível de RAID. Para informações sobre remodelação RAID e exemplos de como usar o comando **lvconvert** para remodelar um volume lógico RAID, consulte a página de manual **lvraid(7)**.

10.12. MUDANDO O NÚMERO DE IMAGENS EM UM DISPOSITIVO RAID1 EXISTENTE

Você pode alterar o número de imagens em uma matriz RAID1 existente, assim como pode alterar o número de imagens na implementação anterior do espelhamento LVM. Use o comando **lvconvert** para especificar o número de pares de metadados/dados adicionais a serem adicionados ou removidos.

Quando você adiciona imagens a um dispositivo RAID1 com o comando **lvconvert**, você pode especificar o número total de imagens para o dispositivo resultante, ou pode especificar quantas imagens adicionar ao dispositivo. Opcionalmente, você também pode especificar em quais volumes físicos os novos pares de metadados/dados de imagem residirão.

Subvolumes de metadados (denominados **rmeta**) sempre existem nos mesmos dispositivos físicos que suas contrapartes de subvolume de dados (**rimage**). Os pares de metadados/subvolume de dados não serão criados nos mesmos volumes físicos que os de outro par de metadados/subvolume de dados na matriz RAID (a menos que você especifique **--alloc anywhere**).

O formato para o comando de adicionar imagens a um volume RAID1 é o seguinte:

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

Por exemplo, o seguinte comando exibe o dispositivo LVM **my_vg/my_lv**, que é uma matriz RAID1 de 2 vias:

```
# 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)
```

O seguinte comando converte o dispositivo RAID1 de 2 vias **my_vg/my_lv** para um dispositivo RAID1 de 3 vias:

```
# 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)
```

Quando você adiciona uma imagem a uma matriz RAID1, você pode especificar quais volumes físicos usar para a imagem. O seguinte comando converte o dispositivo RAID1 de 2 vias **my_vg/my_lv** para um dispositivo RAID1 de 3 vias, especificando que o volume físico **/dev/sdd1** seja usado para a matriz:

```
# 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 remover imagens de uma matriz RAID1, use o seguinte comando. Quando você remover imagens de um dispositivo RAID1 com o comando **lvconvert**, você pode especificar o número total de imagens para o dispositivo resultante, ou você pode especificar quantas imagens a serem removidas do dispositivo. Você também pode especificar opcionalmente os volumes físicos dos quais remover o dispositivo.

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

Além disso, quando uma imagem e seu subvolume de metadados associado são removidos, quaisquer imagens de números mais altos serão deslocadas para baixo para preencher o espaço. Se você remover **lv_rimage_1** de uma matriz RAID1 de 3 vias que consiste de **lv_rimage_0**, **lv_rimage_1** e **lv_rimage_2**, isto resulta em uma matriz RAID1 que consiste de **lv_rimage_0** e **lv_rimage_1**. O subvolume **lv_rimage_2** será renomeado e assumirá o slot vazio, tornando-se **lv_rimage_1**.

O exemplo a seguir mostra o layout de um volume lógico RAID1 de 3 vias **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)
```

O seguinte comando converte o volume lógico RAID1 de 3 vias em um volume lógico RAID1 de 2 vias.

```
# 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)
```

O seguinte comando converte o volume lógico RAID1 de 3 vias em um volume lógico RAID1 de 2 vias, especificando o volume físico que contém a imagem a ser removida 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. SEPARAÇÃO DE UMA IMAGEM RAID COMO UM VOLUME LÓGICO SEPARADO

Você pode dividir uma imagem de um volume lógico RAID para formar um novo volume lógico.

O formato do comando de divisão de uma imagem RAID é o seguinte:

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

Assim como quando você está removendo uma imagem RAID de um volume lógico RAID1 existente, quando você remove um subvolume de dados RAID (e seu subvolume de metadados associado) do meio do dispositivo quaisquer imagens numeradas superiores serão deslocadas para baixo para preencher o slot. Os números índices nos volumes lógicos que compõem uma matriz RAID serão assim uma seqüência ininterrupta de números inteiros.



NOTA

Você não pode dividir uma imagem RAID se a matriz RAID1 ainda não estiver em sincronia.

O exemplo seguinte divide um volume lógico RAID1 de 2 vias, **my_lv**, em dois volumes lógicos lineares, **my_lv** e **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)

```

O exemplo a seguir divide um volume lógico RAID1 de 3 vias, **my_lv**, em um volume lógico RAID1 de 2 vias, **my_lv**, e um volume lógico linear, **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. DIVIDINDO E FUNDINDO UMA IMAGEM RAID

Você pode dividir temporariamente uma imagem de uma matriz RAID1 para uso apenas de leitura enquanto mantém o controle de quaisquer mudanças usando o argumento **--trackchanges** em conjunto com o argumento **--splitmirrors** do comando **lvconvert**. Isto permite que você funda a imagem de volta ao array mais tarde enquanto sincroniza apenas as partes do array que mudaram desde que a imagem foi dividida.

O formato do comando **lvconvert** para dividir uma imagem RAID é o seguinte.

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

Quando você divide uma imagem RAID com o argumento **--trackchanges**, você pode especificar qual imagem dividir, mas não pode mudar o nome do volume a ser dividido. Além disso, os volumes resultantes têm as seguintes restrições.

- O novo volume que você cria é somente leitura.
- Não se pode redimensionar o novo volume.
- Não é possível renomear a matriz restante.

- Não é possível redimensionar a matriz restante.
- Você pode ativar o novo volume e a matriz restante independentemente.

Você pode fundir uma imagem que foi dividida com o argumento **--trackchanges** especificado, executando um comando **lvconvert** subsequente com o argumento **--merge**. Quando você funde a imagem, apenas as partes da matriz que mudaram desde que a imagem foi dividida são sincronizadas novamente.

O formato do comando **lvconvert** para fundir uma imagem RAID é o seguinte.

```
lvconvert --merge raid_image
```

O exemplo a seguir cria um volume lógico RAID1 e depois separa uma imagem desse volume enquanto acompanha as mudanças na 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)
```

O exemplo seguinte separa uma imagem de um volume RAID1 enquanto rastreia as mudanças na matriz restante, depois funde o volume de volta à 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. ESTABELECENDO UMA POLÍTICA DE FALHAS RAID

LVM RAID trata as falhas de dispositivos de forma automática com base nas preferências definidas pelo campo **raid_fault_policy** no arquivo **lvm.conf**.

- Se o campo **raid_fault_policy** estiver configurado para **allocate**, o sistema tentará substituir o dispositivo falhado por um dispositivo de reposição do grupo de volume. Se não houver um dispositivo de reposição disponível, isto será relatado ao registro do sistema.
- Se o campo **raid_fault_policy** estiver configurado para **warn**, o sistema produzirá um aviso e o registro indicará que um dispositivo falhou. Isto permite que o usuário determine o curso de ação a ser tomado.

Enquanto houver dispositivos suficientes para suportar a usabilidade, o volume lógico RAID continuará a operar.

10.15.1. A política de falhas de alocação RAID

No exemplo a seguir, o campo **raid_fault_policy** foi ajustado para **allocate** no arquivo **lvm.conf**. O volume lógico RAID está disposto da seguinte forma.

```
# 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)
```

Se o dispositivo **/dev/sde** falhar, o log do sistema exibirá mensagens de erro.

```
# 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.
```

Desde que o campo **raid_fault_policy** foi ajustado para **allocate**, o dispositivo falhado é substituído por um novo dispositivo do grupo de volume.

```
# 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)
```

Observe que mesmo que o dispositivo com defeito tenha sido substituído, o visor ainda indica que a LVM não conseguiu encontrar o dispositivo com defeito. Isto porque, embora o dispositivo com falha tenha sido removido do volume lógico RAID, o dispositivo com falha ainda não foi removido do grupo de volume. Para remover o dispositivo com falha do grupo de volume, você pode executar **vgreduce --removemissing VG**.

Se o **raid_fault_policy** foi configurado para **allocate**, mas não há dispositivos de reserva, a alocação falhará, deixando o volume lógico como está. Se a alocação falhar, você tem a opção de consertar o drive, iniciando então a recuperação do dispositivo que falhou com a opção **--refresh** do comando **lvchange**. Alternativamente, você pode substituir o dispositivo que falhou.

10.15.2. A política de falhas RAID de alerta

No exemplo a seguir, o campo **raid_fault_policy** foi ajustado para **warn** no arquivo **lvm.conf**. O volume lógico RAID está disposto da seguinte forma.

```
# 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)
```

Se o dispositivo **/dev/sdh** falhar, o log do sistema exibirá mensagens de erro. Neste caso, entretanto, o LVM não tentará reparar o dispositivo RAID automaticamente, substituindo uma das imagens. Ao invés disso, se o dispositivo falhar, você pode substituir o dispositivo com o argumento **--repair** do comando **lvconvert**.

10.16. SUBSTITUIÇÃO DE UM DISPOSITIVO RAID EM UM VOLUME LÓGICO

Você pode substituir um dispositivo RAID em um volume lógico.

- Caso não tenha havido nenhuma falha no dispositivo RAID, siga [Seção 10.16.1, "Substituindo um dispositivo RAID que não tenha falhado"](#).
- Caso o dispositivo RAID tenha falhado, siga [Seção 10.16.4, "Substituição de um dispositivo RAID falho em um volume lógico"](#).

10.16.1. Substituindo um dispositivo RAID que não tenha falhado

Para substituir um dispositivo RAID em um volume lógico, use o argumento **--replace** do comando **lvconvert**.

Pré-requisitos

- O dispositivo RAID não falhou. Os seguintes comandos não funcionarão se o dispositivo RAID falhar.

Procedimento

- Substitua o dispositivo RAID:

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

- Substitua *dev_to_remove* com o caminho para o volume físico que você deseja substituir.
- Substitua *vg/lv* com o grupo de volume e o nome lógico do volume da matriz RAID.
- Substitua *possible_replacements* com o caminho para o volume físico que você deseja usar como substituto.

Exemplo 10.1. Substituição de um dispositivo RAID1

O exemplo seguinte cria um volume lógico RAID1 e depois substitui um dispositivo nesse volume.

1. Criar a matriz RAID1:

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

Logical volume "my_lv" created
```

2. Examine a 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. Substitua o volume físico **/dev/sdb2**:

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

4. Examine a matriz RAID1 com a substituição:

```
# 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)
```

Exemplo 10.2. Especificação do volume físico de reposição

O exemplo seguinte cria um volume lógico RAID1 e depois substitui um dispositivo nesse volume, especificando qual volume físico usar para a substituição.

1. Criar a matriz RAID1:

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

Logical volume "my_lv" created
```

2. Examine a 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. Examinar os volumes 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. Substituir o volume físico **/dev/sdb1** por **/dev/sdd1**:

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

5. Examine a matriz RAID1 com a substituição:

```
# 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)
```

Exemplo 10.3. Substituição de múltiplos dispositivos RAID

Você pode substituir mais de um dispositivo RAID de cada vez especificando vários argumentos **replace**, como no exemplo a seguir.

1. Criar uma matriz RAID1:

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

Logical volume "my_lv" created
```

2. Examine a 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. Substitua os volumes físicos **/dev/sdb1** e **/dev/sdc1**:

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

4. Examine a matriz RAID1 com os substitutos:

```
# 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 falhados em RAID LVM

RAID não é como o tradicional espelhamento LVM. O espelhamento LVM exigia a remoção de dispositivos com falhas ou o volume lógico do espelhamento seria pendurado. As matrizes RAID podem continuar a funcionar com dispositivos com falhas. Na verdade, para outros tipos de RAID além do RAID1, remover um dispositivo significaria converter para um RAID de nível inferior (por exemplo, de RAID6 para RAID5, ou de RAID4 ou RAID5 para RAID0).

Portanto, ao invés de remover incondicionalmente e potencialmente alocar um dispositivo falhado, o LVM permite substituir um dispositivo falhado em um volume RAID em uma solução de uma etapa, usando o argumento **--repair** do comando **lvconvert**.

10.16.3. Recuperando um dispositivo RAID falho em um volume lógico

Se a falha do dispositivo LVM RAID for uma falha transitória ou se você for capaz de reparar o dispositivo que falhou, você pode iniciar a recuperação do dispositivo que falhou.

Pré-requisitos

- O dispositivo anteriormente falhado está agora funcionando.

Procedimento

- Atualize o volume lógico que contém o dispositivo RAID:

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

Etapas de verificação

- Examine o volume lógico com o dispositivo recuperado:

```
# lvs -- todas --opções nome,dispositivos,lv_attr,lv_health_status my_vg
```

10.16.4. Substituição de um dispositivo RAID falho em um volume lógico

Este procedimento substitui um dispositivo falho que serve como volume físico em um volume lógico LVM RAID.

Pré-requisitos

- O grupo de volume inclui um volume físico que fornece capacidade livre suficiente para substituir o dispositivo avariado.
Se nenhum volume físico com extensões livres suficientes estiver disponível no grupo de volume, adicione um novo volume físico suficientemente grande usando o utilitário **vgextend**.

Procedimento

1. No exemplo a seguir, um volume lógico RAID é apresentado como se segue:

```
# 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. Se o dispositivo **/dev/sdc** falhar, a saída do comando **lvs** é a seguinte:

```
# 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. Substitua o dispositivo que falhou e exiba o volume 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 o volume físico que substitui o dispositivo avariado, adicionar o volume físico no final do comando:

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

4. Examine o volume lógico com o substituto:

```
# 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)
```

Até que você remova o dispositivo defeituoso do grupo de volume, as utilidades da LVM ainda indicam que a LVM não consegue encontrar o dispositivo defeituoso.

5. Remover o dispositivo com defeito do grupo de volume:

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

10.17. VERIFICAÇÃO DA COERÊNCIA DOS DADOS EM UM VOLUME LÓGICO RAID (SCRUBBING RAID)

O LVM fornece suporte à lavagem de volumes lógicos RAID. A depuração RAID é o processo de leitura de todos os dados e blocos de paridade em uma matriz e verificação para ver se eles são coerentes.

Procedimento

1. Opcional: Limite a largura de banda de E/S que o processo de depuração utiliza. Quando você realiza uma operação de scrubbing RAID, as E/S de fundo exigidas pelas operações do **sync** podem excluir outras E/S para dispositivos LVM, tais como atualizações de metadados de grupos de volume. Isto pode fazer com que as outras operações LVM diminuam a velocidade. Você pode controlar a taxa da operação de scrubbing implementando o estrangulamento de recuperação.

Adicione as seguintes opções aos comandos **lvchange --syncaction** nos próximos passos:

--maxrecoveryrate *Rate*[bBsSkKmMgG]

Define a taxa máxima de recuperação de modo que a operação exclua as operações de E/S nominais. Ajustar a taxa de recuperação para 0 significa que a operação não está vinculada.

--minrecoveryrate *Rate*[bBsSkKmMgG]

Define a taxa mínima de recuperação para garantir que as E/S para as operações do site **sync** atinjam um rendimento mínimo, mesmo quando a E/S nominal pesada está presente.

Especifique o valor *Rate* como um valor por segundo para cada dispositivo da matriz. Se você não fornecer nenhum sufixo, as opções assumem kiB por segundo por dispositivo.

2. Mostrar o número de discrepâncias na matriz, sem repará-las:

```
# lvchange - verificação de sincronia vg/raid_lv
```

3. Corrigir as discrepâncias na matriz:

```
# lvchange --syncaction repair vg/raid_lv
```



NOTA

A operação **lvchange --syncaction repair** não desempenha a mesma função que a operação **lvconvert --repair**:

- A operação **lvchange --syncaction repair** inicia uma operação de sincronização de fundo na matriz.
- A operação **lvconvert --repair** repara ou substitui dispositivos falhos em um espelho ou em um volume lógico RAID.

4. Opcional: Exibir informações sobre a operação de depuração:

```
# lvs -o raid_sync_action,raid_mismatch_count vg/lv
```

- O campo **raid_sync_action** exibe a operação de sincronização atual que o volume RAID está realizando. Ele pode ser um dos seguintes valores:

idle

Todas as operações de sincronização concluídas (não fazer nada)

resync

Inicialização de uma matriz ou recuperação após uma falha da máquina

recover

Substituição de um dispositivo na matriz

check

Procurando por inconsistências de matriz

repair

Procura e reparo de inconsistências

- O campo **raid_mismatch_count** mostra o número de discrepâncias encontradas durante uma operação **check**.
- O campo **Cpy%Sync** mostra o progresso das operações do **sync**.
- O campo **lv_attr** fornece indicadores adicionais. O bit 9 deste campo exibe a saúde do volume lógico, e suporta os seguintes indicadores:
 - **m** (desajustes) indica que há discrepâncias em um volume lógico RAID. Este caráter é mostrado após uma operação de scrubbing ter detectado que partes do RAID não são coerentes.
 - **r** (refresh) indica que um dispositivo em um array RAID sofreu uma falha e o kernel o considera uma falha, mesmo que o LVM possa ler a etiqueta do dispositivo e considere o dispositivo como operacional. Atualize o volume lógico para notificar o kernel de que o dispositivo está agora disponível, ou substitua o dispositivo se você suspeitar que ele falhou.

Recursos adicionais

- Para mais informações, consulte as páginas de manual **lvchange(8)** e **lvraid(7)**.

10.18. CONVERSÃO DE UM NÍVEL RAID (AQUISIÇÃO RAID)

LVM suporta *Raid takeover*, o que significa converter um volume lógico RAID de um nível RAID para outro (tal como de RAID 5 para RAID 6). A mudança do nível RAID é geralmente feita para aumentar ou diminuir a resistência a falhas de dispositivos ou para restringir os volumes lógicos. Você usa o **lvconvert** para a aquisição do RAID. Para informações sobre a aquisição do RAID e para exemplos de utilização do **lvconvert** para converter um volume lógico RAID, consulte a página de manual **lvraid(7)**.

10.19. MUDANÇA DE ATRIBUTOS DE UM VOLUME RAID (REFORMULAÇÃO RAID)

RAID *reshaping* significa alterar os atributos de um volume lógico RAID mantendo o mesmo nível de RAID. Alguns atributos que você pode alterar incluem o layout RAID, tamanho das faixas e número de faixas. Para informações sobre remodelação RAID e exemplos de uso do comando **lvconvert** para remodelar um volume lógico RAID, consulte a página de manual **lvraid(7)**.

10.20. CONTROLE DAS OPERAÇÕES DE E/S EM UM VOLUME LÓGICO RAID1

Você pode controlar as operações de E/S de um dispositivo em um volume lógico RAID1, usando os parâmetros **--writemostly** e **--writebehind** do comando **lvchange**. O formato para usar estes parâmetros é o seguinte.

- **--[raid]writemostly *PhysicalVolume*[:{t|y|n}]**
Marca um dispositivo em um volume lógico RAID1 como **write-mostly**. Todas as leituras para estes drives serão evitadas, a menos que seja necessário. A definição deste parâmetro mantém o número de operações de E/S para o drive no mínimo. Por padrão, o atributo **write-mostly** é definido como sim para o volume físico especificado no volume lógico. É possível remover a bandeira **write-mostly**, anexando **:n** ao volume físico ou alternar o valor especificando **:t**. O argumento **--writemostly** pode ser especificado mais de uma vez em um único comando, tornando possível alternar de uma só vez os atributos de escrita para todos os volumes físicos em um volume lógico.
- **--[raid]writebehind *IOCount***
Especifica o número máximo de escritas pendentes que são permitidas aos dispositivos em um volume lógico RAID1 que são marcados como **write-mostly**. Uma vez ultrapassado este valor, as gravações tornam-se síncronas, fazendo com que todas as gravações nos dispositivos constituintes sejam concluídas antes que a matriz sinalize que a gravação foi concluída. A definição do valor a zero limpa a preferência e permite que o sistema escolha o valor arbitrariamente.

10.21. MUDANDO O TAMANHO DA REGIÃO EM UM VOLUME LÓGICO RAID

Quando você cria um volume lógico RAID, o tamanho da região para o volume lógico será o valor do parâmetro **raid_region_size** no arquivo **/etc/lvm/lvm.conf**. Você pode substituir este valor padrão com a opção **-R** do comando **lvcreate**.

Após ter criado um volume lógico RAID, você pode alterar o tamanho da região do volume com a opção **-R** do comando **lvconvert**. O exemplo a seguir muda o tamanho da região do volume lógico **vg/raidlv** para 4096K. O volume RAID deve ser sincronizado a fim de mudar o tamanho da região.

```
# 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. VOLUMES LÓGICOS INSTANTÂNEOS

O recurso LVM snapshot oferece a capacidade de criar imagens virtuais de um dispositivo em um determinado instante sem causar uma interrupção do serviço.

11.1. VOLUMES INSTANTÂNEOS

O recurso LVM snapshot oferece a capacidade de criar imagens virtuais de um dispositivo em um determinado instante sem causar uma interrupção do serviço. Quando uma mudança é feita no dispositivo original (a origem) após uma foto, o recurso de foto faz uma cópia da área de dados alterada como estava antes da mudança para que possa reconstruir o estado do dispositivo.



NOTA

A LVM apóia snapshots com pouco tempo de fornecimento.

Como um instantâneo copia apenas as áreas de dados que mudam depois que o instantâneo é criado, o recurso de instantâneo requer uma quantidade mínima de armazenamento. Por exemplo, com uma origem raramente atualizada, 3-5 % da capacidade da origem é suficiente para manter o instantâneo.



NOTA

As cópias instantâneas de um sistema de arquivo são cópias virtuais, não um backup de mídia real para um sistema de arquivo. Os instantâneos não fornecem um substituto para um procedimento de backup.

O tamanho do instantâneo rege a quantidade de espaço reservado para armazenar as mudanças no volume de origem. Por exemplo, se você fizer um instantâneo e depois sobrescrever completamente a origem, o instantâneo teria que ser pelo menos tão grande quanto o volume de origem para segurar as mudanças. Você precisa dimensionar um instantâneo de acordo com o nível de mudança esperado. Assim, por exemplo, um instantâneo de curta duração de um volume mais lido, como **/usr**, precisaria de menos espaço do que um instantâneo de longa duração de um volume que vê um número maior de escritas, como **/home**.

Se um instantâneo estiver cheio, o instantâneo se torna inválido, uma vez que não pode mais rastrear as mudanças no volume de origem. Você deve monitorar regularmente o tamanho do instantâneo. Os instantâneos são totalmente redimensionáveis, porém, se você tiver a capacidade de armazenamento, poderá aumentar o tamanho do volume do instantâneo para evitar que ele caia. Por outro lado, se você achar que o volume da foto é maior do que você precisa, você pode reduzir o tamanho do volume para liberar espaço que é necessário para outros volumes lógicos.

Quando você cria um sistema de arquivo instantâneo, o acesso completo de leitura e escrita à origem permanece possível. Se um pedaço de um instantâneo for alterado, esse pedaço é marcado e nunca é copiado do volume original.

Há vários usos para o recurso de instantâneo:

- Normalmente, um instantâneo é tirado quando é necessário fazer um backup em um volume lógico sem interromper o sistema ao vivo que está continuamente atualizando os dados.
- Você pode executar o comando **fsck** em um sistema de arquivo instantâneo para verificar a integridade do sistema de arquivo e determinar se o sistema de arquivo original requer reparo do sistema de arquivo.

- Como o instantâneo é lido/escrito, você pode testar aplicações contra os dados de produção tirando um instantâneo e executando testes contra o instantâneo, deixando os dados reais intocados.
- Você pode criar volumes LVM para uso com a Virtualização da Red Hat. Snapshots LVM podem ser usados para criar snapshots de imagens de convidados virtuais. Estes snapshots podem fornecer uma maneira conveniente para modificar os convidados existentes ou criar novos convidados com o mínimo de armazenamento adicional.

Você pode usar a opção **--merge** do comando **lvconvert** para fundir um instantâneo em seu volume de origem. Um uso para este recurso é realizar o rollback do sistema se você tiver perdido dados ou arquivos ou se precisar restaurar seu sistema a um estado anterior. Após fundir o volume de snapshot, o volume lógico resultante terá o nome do volume de origem, número menor e UUID e o snapshot fundido é removido.

11.2. CRIAÇÃO DE VOLUMES DE INSTANTÂNEOS

Use o argumento **-s** do comando **lvcreate** para criar um volume de instantâneo. Um volume de instantâneo pode ser escrito.



NOTA

Os instantâneos de LVM não são suportados através dos nós em um cluster. Não se pode criar um volume instantâneo em um grupo de volume compartilhado. Entretanto, se você precisar criar um backup consistente de dados em um volume lógico compartilhado, você pode ativar o volume exclusivamente e depois criar o instantâneo.



NOTA

Os snapshots são suportados para volumes lógicos RAID.

O LVM não permite criar um volume instantâneo maior do que o tamanho do volume de origem mais os metadados necessários para o volume. Se você especificar um volume de instantâneo maior que este, o sistema criará um volume de instantâneo que é apenas tão grande quanto será necessário para o tamanho da origem.

Por padrão, um volume de instantâneo é pulado durante os comandos normais de ativação.

O seguinte procedimento cria um volume lógico de origem chamado **origin** e um volume instantâneo do volume original chamado **snap**.

1. Criar um volume lógico chamado **origin** a partir do grupo de volume **VG**.

```
# lvcreate -L 1G -n origin VG
Logical volume "origin" created.
```

2. Crie um volume lógico instantâneo de **/dev/VG/origin** que tenha 100 MB de tamanho, chamado **snap**. Se o volume lógico original contiver um sistema de arquivo, você pode montar o volume lógico instantâneo em um diretório arbitrário, a fim de acessar o conteúdo do sistema de arquivo para executar um backup enquanto o sistema de arquivo original continua a ser atualizado.

```
# lvcreate --size 100M --snapshot --name snap /dev/VG/origin
Logical volume "snap" created.
```

- Mostrar o status do volume lógico **/dev/VG/origin**, mostrando todos os volumes lógicos instantâneos e seu status (ativo ou inativo).

```
# 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
```

- O comando **lvs**, por padrão, exibe o volume de origem e a porcentagem atual do volume de instantâneo que está sendo utilizado. O exemplo a seguir mostra a saída padrão para o comando **lvs** após ter criado o volume de instantâneo, com um display que inclui os dispositivos que constituem os volumes 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)
```



ATENÇÃO

Como a foto aumenta de tamanho conforme o volume de origem muda, é importante monitorar regularmente a porcentagem do volume da foto com o comando **lvs** para ter certeza de que ela não preencha. Um instantâneo de 100 tiros é completamente perdido, pois uma gravação em partes inalteradas da origem não teria sucesso sem corromper o instantâneo.

Além da própria foto ser invalidada quando cheia, qualquer sistema de arquivo montado naquele dispositivo de foto é forçado a não ser montado, evitando os inevitáveis erros do sistema de arquivo ao acessar o ponto de montagem. Além disso, você pode especificar a opção **snapshot_autoextend_threshold** no arquivo **lvm.conf**. Esta opção permite a extensão automática de um snapshot sempre que o espaço de snapshot restante cair abaixo do limite que você definir. Esta característica requer que haja espaço não alocado no grupo de volume.

O LVM não permite criar um volume instantâneo maior do que o tamanho do volume de origem mais os

metadados necessários para o volume. Da mesma forma, a extensão automática de um instantâneo não aumentará o tamanho de um volume de instantâneo além do tamanho máximo calculado necessário para o instantâneo. Uma vez que um instantâneo tenha crescido o suficiente para cobrir a origem, ele não será mais monitorado para extensão automática.

As informações sobre a configuração **snapshot_autoextend_threshold** e **snapshot_autoextend_percent** são fornecidas no próprio arquivo **/etc/lvm/lvm.conf**.

11.3. FUSÃO DE VOLUMES DE FOTOS

Você pode usar a opção **--merge** do comando **lvconvert** para fundir um instantâneo em seu volume de origem. Se tanto a origem quanto o volume de snapshot não estiverem abertos, a fusão começará imediatamente. Caso contrário, a fusão começará na primeira vez que a origem ou instantâneo forem ativados e ambos forem fechados. A fusão de um instantâneo em uma origem que não pode ser fechada, por exemplo, um sistema de arquivo raiz, é adiada até a próxima vez em que o volume de origem for ativado. Quando a fusão começa, o volume lógico resultante terá o nome da origem, número menor e UUID. Enquanto a fusão está em andamento, as leituras ou escritas para a origem aparecem como foram direcionadas para o instantâneo que está sendo fundido. Quando a fusão termina, o instantâneo fundido é removido.

O seguinte comando funde o volume de instantâneos **vg00/lvol1_snap** em sua origem.

```
# lvconvert --merge vg00/lvol1_snap
```

Você pode especificar vários instantâneos na linha de comando, ou pode usar tags de objetos LVM para especificar que vários instantâneos sejam fundidos em suas respectivas origens. No exemplo a seguir, os volumes lógicos **vg00/lvol1**, **vg00/lvol2**, e **vg00/lvol3** estão todos marcados com a tag **@some_tag**. O seguinte comando funde os volumes lógicos instantâneos para os três volumes em série: **vg00/lvol1**, depois **vg00/lvol2**, depois **vg00/lvol3**. Se a opção **--background** fosse utilizada, todas as fusões de volumes lógicos instantâneos começariam em paralelo.

```
# lvconvert --merge @some_tag
```

Para mais informações sobre o comando **lvconvert --merge**, consulte a página de manual **lvconvert(8)**.

CAPÍTULO 12. CRIAÇÃO E GERENCIAMENTO DE VOLUMES LÓGICOS DE POUCA MONTA (VOLUMES FINOS)

Os volumes lógicos podem ser provisionados de forma fina. Isto permite criar volumes lógicos maiores do que as extensões disponíveis.

12.1. VOLUMES LÓGICOS DE FORNECIMENTO FINO (VOLUMES FINOS)

Os volumes lógicos podem ser provisionados de forma fina. Isto permite criar volumes lógicos maiores do que as extensões disponíveis. Usando thin provisioning, você pode gerenciar um pool de armazenamento de espaço livre, conhecido como thin pool, que pode ser alocado a um número arbitrário de dispositivos quando necessário pelas aplicações. Você pode então criar dispositivos que podem ser vinculados ao thin pool para alocação posterior quando uma aplicação realmente escreve para o volume lógico. O thin pool pode ser expandido dinamicamente quando necessário para alocação econômica de espaço de armazenamento.



NOTA

Os volumes finos não são suportados através dos nós em um cluster. O pool fino e todos os seus volumes finos devem ser ativados exclusivamente em um único nó de cluster.

Ao utilizar thin provisioning, um administrador de armazenamento pode comprometer excessivamente o armazenamento físico, muitas vezes evitando a necessidade de comprar armazenamento adicional. Por exemplo, se cada dez usuários solicitarem um sistema de arquivo de 100GB para sua aplicação, o administrador de armazenamento pode criar o que parece ser um sistema de arquivo de 100GB para cada usuário, mas que é apoiado por menos armazenamento real que é usado somente quando necessário. Ao utilizar thin provisioning, é importante que o administrador de armazenamento monitore o pool de armazenamento e acrescente mais capacidade se este começar a ficar cheio.

Para garantir que todo o espaço disponível possa ser utilizado, a LVM suporta o descarte de dados. Isto permite a reutilização do espaço que antes era utilizado por um arquivo descartado ou outro intervalo de blocos.

Os volumes finos fornecem suporte para uma nova implementação de volumes lógicos de cópia-em-escrita (COW), que permitem que muitos dispositivos virtuais compartilhem os mesmos dados no pool fino.

12.2. CRIAÇÃO DE VOLUMES LÓGICOS POUCO PREVISÍVEIS

Este procedimento fornece uma visão geral dos comandos básicos que você usa para criar e crescer volumes lógicos pouco previsíveis. Para informações detalhadas sobre o provisionamento de LVM thin provisioning, bem como informações sobre o uso dos comandos e utilitários LVM com volumes lógicos thinly-provisioned, consulte a página de manual `lvminthin(7)`.

Para criar um volume fino, execute as seguintes tarefas:

1. Criar um grupo de volume com o comando **vgcreate**.
2. Criar um pool fino com o comando **lvcreate**.
3. Crie um volume fino no pool fino com o comando **lvcreate**.

Você pode usar a opção **-T** (ou **--thin**) do comando **lvcreate** para criar um pool fino ou um volume fino. Você também pode usar a opção **-T** do comando **lvcreate** para criar tanto um thin pool quanto um volume thin nesse pool ao mesmo tempo com um único comando.

O seguinte comando usa a opção **-T** do comando **lvcreate** para criar um pool fino chamado **mythinpool** no grupo de volume **vg001** e que tem um tamanho de 100M. Note que, como você está criando um pool de espaço físico, você deve especificar o tamanho do pool. A opção **-T** do comando **lvcreate** não aceita um argumento; ela deduz o tipo de dispositivo a ser criado das outras opções que o comando especifica.

```
# 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
```

O seguinte comando usa a opção **-T** do comando **lvcreate** para criar um volume fino chamado **thinvolume** no pool fino **vg001/mythinpool**. Note que neste caso você está especificando um tamanho virtual, e que você está especificando um tamanho virtual para o volume que é maior do que o do pool que o contém.

```
# 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
```

O seguinte comando usa a opção **-T** do comando **lvcreate** para criar um pool fino e um volume fino nesse pool, especificando tanto um argumento de tamanho como um argumento de tamanho virtual para o comando **lvcreate**. Este comando cria um thin pool chamado **mythinpool** no grupo de volume **vg001** e também cria um thin volume chamado **thinvolume** nesse pool.

```
# 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
```

Você também pode criar um pool fino especificando o parâmetro **--thinpool** do comando **lvcreate**. Ao contrário da opção **-T**, o parâmetro **--thinpool** requer um argumento, que é o nome do volume lógico do thin pool que você está criando. O exemplo a seguir especifica o parâmetro **--thinpool** do comando

lvcreate para criar um thin pool chamado **mythinpool** no grupo de volume **vg001** e que tem um tamanho 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
```

Use os seguintes critérios para utilizar o tamanho do pedaço:

- Um pedaço menor requer mais metadados e dificulta o desempenho, mas proporciona uma melhor utilização do espaço com instantâneos.
- Um pedaço maior requer menos manipulação de metadados, mas torna o instantâneo menos eficiente em termos de espaço.

Por padrão, **lvm2** começa com um pedaço de tamanho 64KiB e aumenta seu valor quando o tamanho resultante do dispositivo de metadados do pool fino cresce acima de 128MiB, isto mantém o tamanho dos metadados compacto. No entanto, isto pode resultar em alguns valores de tamanho de pedaços grandes, que são menos eficientes em termos de espaço para o uso de instantâneos. Nesses casos, um pedaço menor e um tamanho maior de metadados é uma opção melhor.

Se o tamanho dos dados de volume estiver na faixa de TiB, use ~15,8GiB como o tamanho dos metadados, que é o tamanho máximo suportado, e defina o tamanho do pedaço de acordo com sua necessidade. Mas, note que não é possível aumentar o tamanho dos metadados se você precisar ampliar o tamanho dos dados do volume e ter um pequeno pedaço de tamanho.



ATENÇÃO

A Red Hat não recomenda definir um pedaço de tamanho menor do que o valor padrão. Se o tamanho do pedaço for muito pequeno e seu volume ficar sem espaço para metadados, o volume é incapaz de criar dados. Monitore seus volumes lógicos para garantir que eles sejam expandidos, ou crie mais armazenamento antes que os volumes dos metadados fiquem completamente cheios. Assegure-se de configurar seu pool fino com um pedaço suficientemente grande para que não fiquem sem espaço para os metadados.

O Striping é apoiado para a criação de pools. O seguinte comando cria um pool fino de 100M chamado **pool** no grupo de volume **vg001** com duas faixas de 64 kB e um pedaço de 256 kB. Ele também cria um volume fino de 1T, **vg00/thin_lv**.

```
# lvcreate -i 2 -l 64 -c 256 -L 100M -T vg00/pool -V 1T --name thin_lv
```

Você pode ampliar o tamanho de um volume fino com o comando **lvextend**. Você não pode, entretanto, reduzir o tamanho de uma piscina fina.

O seguinte comando redimensiona uma piscina fina existente de 100M de tamanho, ampliando-a com mais 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
```

Como com outros tipos de volumes lógicos, você pode renomear o volume com o **lvrename**, pode remover o volume com o **lvremove**, e pode exibir informações sobre o volume com os comandos **lvs** e **lvdisplay**.

By default, the **lvcreate** command sets the size of the thin pool's metadata logical volume according to the formula (Pool_LV_size / 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.

Você pode usar o parâmetro **--thinpool** do comando **lvconvert** para converter um volume lógico existente em um volume de pool fino. Ao converter um volume lógico existente para um volume de thin pool, você deve usar o parâmetro **--poolmetadata** em conjunto com o parâmetro **--thinpool** do **lvconvert** para converter um volume lógico existente para o volume de metadados do thin pool.



NOTA

A conversão de um volume lógico em um volume de pool fino ou um volume de metadados de pool fino destrói o conteúdo do volume lógico, já que neste caso o **lvconvert** não preserva o conteúdo dos dispositivos, mas, em vez disso, sobrescreve o conteúdo.

O exemplo a seguir converte o volume lógico existente **lv1** no grupo de volume **vg001** para um volume de pool fino e converte o volume lógico existente **lv2** no grupo de volume **vg001** para o volume de metadados para esse volume de pool fino.

```
# lvconvert --thinpool vg001/lv1 --poolmetadata vg001/lv2
```

```
Converted vg001/lv1 to thin pool.
```

12.3. VOLUMES DE FOTOS DE ÚLTIMA GERAÇÃO

O Red Hat Enterprise Linux fornece suporte para volumes de instantâneos pouco fornecidos. Os volumes de snapshot finos permitem que muitos dispositivos virtuais sejam armazenados no mesmo volume de dados. Isto simplifica a administração e permite o compartilhamento de dados entre os volumes de snapshot.

Como para todos os volumes de instantâneos LVM, assim como para todos os volumes finos, os volumes de instantâneos finos não são suportados em todos os nós de um cluster. O volume de instantâneos deve ser ativado exclusivamente em um único nó de cluster.

Os volumes finos proporcionam os seguintes benefícios:

- Um volume fino de instantâneo pode reduzir o uso do disco quando há vários instantâneos com o mesmo volume de origem.

- Se houver vários instantâneos da mesma origem, então uma gravação na origem fará com que uma operação COW preserve os dados. Aumentar o número de instantâneos da origem não deve resultar em uma grande desaceleração.
- Os volumes finos podem ser usados como uma origem lógica de volume para outro instantâneo. Isto permite uma profundidade arbitrária de instantâneos recursivos (instantâneos de instantâneos de instantâneos de instantâneos...).
- Um instantâneo de um volume lógico fino também cria um volume lógico fino. Isto não consome espaço de dados até que uma operação COW seja necessária, ou até que o próprio instantâneo seja escrito.
- Um volume de instantâneo fino não precisa ser ativado com sua origem, portanto, um usuário pode ter apenas a origem ativa enquanto houver muitos volumes de instantâneos inativos da origem.
- Quando você apaga a origem de um volume de imagens de origem, cada imagem desse volume de origem se torna um volume independente de imagens de origem. Isto significa que, ao invés de fundir um instantâneo com seu volume de origem, você pode optar por excluir o volume de origem e então criar um novo instantâneo com uma fina previsão usando esse volume independente como o volume de origem para o novo instantâneo.

Embora haja muitas vantagens no uso de volumes finos, há alguns casos de uso para os quais o recurso de volume de instantâneos LVM antigo pode ser mais apropriado às suas necessidades:

- Não se pode mudar o tamanho de um pedaço de uma piscina fina. Se a piscina fina tiver um pedaço de tamanho grande (por exemplo, 1MB) e você precisar de uma foto de curta duração para a qual um pedaço de tamanho tão grande não seja eficiente, você pode optar por usar o recurso de foto mais antigo.
- Não é possível limitar o tamanho de um volume fino; o instantâneo utilizará todo o espaço da piscina fina, se necessário. Isto pode não ser apropriado para suas necessidades.

Em geral, você deve considerar as exigências específicas de seu site ao decidir qual o formato de instantâneo a ser utilizado.

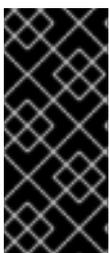
12.4. CRIAÇÃO DE VOLUMES DE INSTANTÂNEOS POUCO PREVISÍVEIS

O Red Hat Enterprise Linux fornece suporte para volumes de instantâneos pouco fornecidos.



NOTA

Esta seção fornece uma visão geral dos comandos básicos que você usa para criar e crescer volumes de instantâneos pouco previsíveis. Para informações detalhadas sobre provisão fina de LVM, bem como informações sobre o uso dos comandos e utilitários LVM com volumes lógicos pouco previsíveis, consulte a página de manual **lvmtthin(7)**.



IMPORTANTE

Ao criar um volume fino, não se especifica o tamanho do volume. Se você especificar um parâmetro de tamanho, o snapshot que será criado não será um volume de snapshot fino e não utilizará o pool fino para armazenar dados. Por exemplo, o comando **lvcreate -s vg/thinvolume -L10M** não criará um instantâneo fino, mesmo que o volume de origem seja um volume fino.

Podem ser criados instantâneos finos para volumes de origem pouco fornecidos, ou para volumes de origem que não são pouco fornecidos.

Você pode especificar um nome para o volume de instantâneos com a opção **--name** do comando **lvcreate**. O comando a seguir cria um volume de instantâneo do volume lógico de pouco volume **vg001/thinvolume** que é chamado **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
```

Um volume fino tem as mesmas características que qualquer outro volume fino. Você pode independentemente ativar o volume, estender o volume, renomear o volume, remover o volume e até mesmo fotografar o volume.

Por padrão, um volume de instantâneo é pulado durante os comandos normais de ativação. Para informações sobre como controlar a ativação de um volume lógico, consulte [Ativação do volume lógico](#).

Você também pode criar um instantâneo de um volume lógico não previsto. Uma vez que o volume lógico não previsto não está contido dentro de um pool fino, ele é referido como um *external origin*. Os volumes de origem externa podem ser utilizados e compartilhados por muitos volumes de imagens de origem fina, mesmo de diferentes piscinas finas. A origem externa deve ser inativa e somente de leitura no momento em que o instantâneo com *thinly-provisioned* é criado.

Para criar um instantâneo de uma origem externa, você deve especificar a opção **--thinpool**. O seguinte comando cria um pequeno volume do volume inativo somente de leitura **origin_volume**. O nome do volume fino é **mythinsnap**. O volume lógico **origin_volume** torna-se então a origem externa do volume thin snapshot **mythinsnap** no grupo de volume **vg001** que utilizará o pool thin existente **vg001/pool**. Como o volume de origem deve estar no mesmo grupo de volume que o volume de instantâneos, não é necessário especificar o grupo de volume ao especificar o volume lógico de origem.

```
# lvcreate -s --thinpool vg001/pool origin_volume --name mythinsnap
```

Você pode criar um segundo volume de instantâneo do primeiro volume, como no seguinte comando.

```
# lvcreate -s vg001/mythinsnap --name my2ndthinsnap
```

Você pode exibir uma lista de todos os antepassados e descendentes de um volume lógico de um pequeno instantâneo especificando os campos de relatório **lv_ancestors** e **lv_descendants** do comando **lvs**.

No exemplo a seguir:

- **stack1** é um volume de origem em grupo de volume **vg001**.
- **stack2** é um retrato de **stack1**
- **stack3** é um retrato de **stack2**
- **stack4** é um retrato de **stack3**

Além disso:

- **stack5** também é um instantâneo de **stack2**
- **stack6** é um retrato 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

Os campos **lv_ancestors** e **lv_descendants** mostram as dependências existentes, mas não rastreiam as entradas removidas que podem quebrar uma cadeia de dependência se a entrada foi removida do meio da cadeia. Por exemplo, se você remover o volume lógico **stack3** desta configuração de amostra, a exibição é a seguinte.

```
$ 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
```

Você pode configurar seu sistema para rastrear e exibir os volumes lógicos que foram removidos, e pode exibir toda a cadeia de dependência que inclui esses volumes, especificando os campos **lv_ancestors_full** e **lv_descendants_full**.

12.5. RASTREAMENTO E EXIBIÇÃO DE VOLUMES FINOS QUE FORAM REMOVIDOS

Você pode configurar seu sistema para rastrear imagens e volumes lógicos finos que foram removidos, ativando a opção de metadados **record_lvs_history** no arquivo de configuração **lvm.conf**. Isto permite que você exiba uma cadeia completa de dependência *thinshot* que inclui volumes lógicos que foram removidos da cadeia de dependência original e que se tornaram *historical* volumes lógicos.

Você pode configurar seu sistema para reter volumes históricos por um período de tempo definido especificando o tempo de retenção, em segundos, com a opção **lvs_history_retention_time** metadados no arquivo de configuração **lvm.conf**.

Um volume lógico histórico mantém uma representação simplificada do volume lógico que foi removido, incluindo os seguintes campos de relatório para o volume:

- **lv_time_removed**: o tempo de remoção do volume lógico
- **lv_time**: o tempo de criação do volume lógico
- **lv_name**: o nome do volume lógico

- **lv_uuid**: a UUID do volume lógico
- **vg_name**: o grupo de volume que contém o volume lógico.

Quando um volume é removido, o nome histórico lógico do volume adquire um hífen como prefixo. Por exemplo, quando se remove o volume lógico **lvvol1**, o nome do volume histórico é **-lvvol1**. Um volume lógico histórico não pode ser reativado.

Mesmo quando a opção de metadados **record_lvs_history** estiver ativada, você pode evitar a retenção de volumes lógicos históricos em uma base individual ao remover um volume lógico especificando a opção **--nohistory** do comando **lvremove**.

Para incluir volumes lógicos históricos na exibição do volume, você especifica a opção **-H|--history** de um comando de exibição LVM. Você pode exibir uma cadeia completa de dependência fina que inclui volumes históricos, especificando os campos **lv_full_ancestors** e **lv_full_descendants**, juntamente com a opção **-H**.

A série de comandos a seguir fornece exemplos de como você pode exibir e gerenciar volumes lógicos históricos.

1. Assegure-se de que os volumes históricos lógicos sejam retidos, definindo **record_lvs_history=1** no arquivo **lv.conf**. Esta opção de metadados não é ativada por padrão.
2. Digite o seguinte comando para exibir uma fina cadeia de imagens provisionadas. Neste exemplo:

- **lvvol1** é um volume de origem, o primeiro volume da cadeia.
- **lvvol2** é um instantâneo de **lvvol1**.
- **lvvol3** é um instantâneo de **lvvol2**.
- **lvvol4** é um instantâneo de **lvvol3**.
- **lvvol5** é também um instantâneo de **lvvol3**.

Observe que, embora o comando de exibição do exemplo **lvs** inclua a opção **-H**, ainda não foi removido nenhum volume fino e não há volumes lógicos históricos a serem exibidos.

```
# lvs -H -o name,full_ancestors,full_descendants
LV  FAncestors    FDescendants
lvvol1          lvvol2,lvvol3,lvvol4,lvvol5
lvvol2 lvvol1      lvvol3,lvvol4,lvvol5
lvvol3 lvvol2,lvvol1  lvvol4,lvvol5
lvvol4 lvvol3,lvvol2,lvvol1
lvvol5 lvvol3,lvvol2,lvvol1
pool
```

3. Remova o volume lógico **lvvol3** da cadeia de instantâneos, depois execute novamente o seguinte comando **lvs** para ver como os volumes lógicos históricos são exibidos, juntamente com seus antepassados e descendentes.

```
# lvremove -f vg/lvol3
Logical volume "lvvol3" successfully removed
# 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
```

4. Você pode usar o campo **lv_time_removed** para exibir a hora em que um volume histórico foi removido.

```
# lvs -H -o name,full_ancestors,full_descendants,time_removed
LV  FAncestors      FDescendants      RTime
lvol1          lvol2,-lvol3,lvol4,lvol5
lvol2 lvol1          -lvol3,lvol4,lvol5
-lvol3 lvol2,lvol1  lvol4,lvol5      2016-03-14 14:14:32 +0100
lvol4 -lvol3,lvol2,lvol1
lvol5 -lvol3,lvol2,lvol1
pool
```

5. Você pode fazer referência a volumes lógicos históricos individualmente em um comando de exibição, especificando o formato *vgname/lvname*, como no exemplo a seguir. Observe que o quinto bit no campo **lv_attr** está configurado para **h** para indicar que o volume é um volume histórico.

```
# lvs -H vg/-lvol3
LV  VG  Attr  LSize
-lvol3 vg  ----h-----  0
```

6. A LVM não mantém volumes lógicos históricos se o volume não tiver descendente vivo. Isto significa que se você remover um volume lógico no final de uma cadeia de instantâneos, o volume lógico não é retido como um volume lógico histórico.

```
# lvremove -f vg/lvol5
Automatically removing historical logical volume vg/-lvol5.
Logical volume "lvol5" successfully removed
# lvs -H -o name,full_ancestors,full_descendants
LV  FAncestors      FDescendants
lvol1          lvol2,-lvol3,lvol4
lvol2 lvol1          -lvol3,lvol4
-lvol3 lvol2,lvol1  lvol4
lvol4 -lvol3,lvol2,lvol1
pool
```

7. Execute os seguintes comandos para remover o volume **lvvol1** e **lvvol2** e para ver como o comando **lvs** exibe os volumes uma vez que eles foram removidos.

```
# lvremove -f vg/lvol1 vg/lvol2
Logical volume "lvol1" successfully removed
Logical volume "lvol2" successfully removed
# lvs -H -o name,full_ancestors,full_descendants
LV  FAncestors      FDescendants
-lvol1          -lvol2,-lvol3,lvol4
-lvol2 -lvol1          -lvol3,lvol4
```

```
-lvol3 -lvol2,-lvol1    lvol4
lvol4 -lvol3,-lvol2,-lvol1
pool
```

8. Para remover completamente um volume lógico histórico, você pode executar novamente o comando **lvremove**, especificando o nome do volume histórico que agora inclui o hífen, como no exemplo a seguir.

```
# lvremove -f vg/-lvol3
Historical logical volume "lvol3" successfully removed
# lvs -H -o name,full_ancestors,full_descendants
LV  FAncestors  FDescendants
-lvol1          -lvol2,lvol4
-lvol2 -lvol1    lvol4
lvol4 -lvol2,-lvol1
pool
```

9. Um volume lógico histórico é retido enquanto houver uma cadeia que inclua volumes vivos em seus descendentes. Isto significa que a remoção de um volume lógico histórico também remove todos os volumes lógicos da cadeia se nenhum descendente existente estiver ligado a eles, como mostrado no exemplo a seguir.

```
# lvremove -f vg/lvol4
Automatically removing historical logical volume vg/-lvol1.
Automatically removing historical logical volume vg/-lvol2.
Automatically removing historical logical volume vg/-lvol4.
Logical volume "lvol4" successfully removed
```

CAPÍTULO 13. PERMITINDO O CACHE PARA MELHORAR O DESEMPENHO DO VOLUME LÓGICO

Você pode adicionar caching a um volume lógico LVM para melhorar o desempenho. O LVM então armazena as operações de E/S para o volume lógico usando um dispositivo rápido, como um SSD.

Os seguintes procedimentos criam um LV especial a partir do dispositivo rápido, e anexam este LV especial ao LV original para melhorar o desempenho.

13.1. MÉTODOS DE CACHE EM LVM

A LVM fornece os seguintes tipos de caching. Cada um deles é adequado para diferentes tipos de padrões de E/S sobre o volume lógico.

dm-cache

Este método acelera o acesso aos dados frequentemente utilizados, armazenando-os em cache no volume mais rápido. O método armazena as operações de leitura e escrita.

O método **dm-cache** cria volumes lógicos do tipo **cache**.

dm-writecache

Este método só permite escrever operações de escrita. O volume mais rápido armazena as operações de gravação e depois as migra para o disco mais lento em segundo plano. O volume mais rápido é geralmente um disco SSD ou um disco de memória persistente (PMEM).

O método **dm-writecache** cria volumes lógicos do tipo **writecache**.

13.2. COMPONENTES DE CACHING LVM

Quando você ativa o cache para um volume lógico, a LVM renomeia e esconde os volumes originais, e apresenta um novo volume lógico que é composto dos volumes lógicos originais. A composição do novo volume lógico depende do método de caching e se você está usando a opção **cachevol** ou **cachepool**.

As opções **cachevol** e **cachepool** expõem diferentes níveis de controle sobre a colocação dos componentes do caching:

- Com a opção **cachevol**, o dispositivo mais rápido armazena tanto as cópias em cache dos blocos de dados quanto os metadados para o gerenciamento do cache.
- Com a opção **cachepool**, dispositivos separados podem armazenar as cópias em cache dos blocos de dados e os metadados para o gerenciamento do cache.
O método **dm-writecache** não é compatível com **cachepool**.

Em todas as configurações, o LVM expõe um único dispositivo resultante, que agrupa todos os componentes do cache. O dispositivo resultante tem o mesmo nome que o volume lógico lento original.

13.3. PERMITINDO O CACHING DM-CACHE PARA UM VOLUME LÓGICO

Este procedimento permite o cache de dados comumente usados em um volume lógico usando o método **dm-cache**.

Pré-requisitos

- Um volume lógico lento que você deseja acelerar usando **dm-cache** existe em seu sistema.
- O grupo de volume que contém o volume lógico lento também contém um volume físico não utilizado em um dispositivo de bloqueio rápido.

Procedimento

1. Crie um volume **cachevol** no dispositivo rápido:

```
# lvcreate --size cachevol-size --nome fastvol vg /dev/fast-pv
```

Substituir os seguintes valores:

cachevol-size

O tamanho do volume **cachevol**, tais como **5G**

fastvol

Um nome para o volume **cachevol**

vg

O nome do grupo de volume

/dev/fast-pv

O caminho para o dispositivo de bloqueio rápido, tais como **/dev/sdf1**

2. Anexe o volume **cachevol** ao volume lógico principal para iniciar o caching:

```
# lvconvert -- tipo cache --cachevol fastvol vg /main-lv
```

Substituir os seguintes valores:

fastvol

O nome do volume **cachevol**

vg

O nome do grupo de volume

main-lv

O nome do volume lógico lento

Etapas de verificação

- Examine os dispositivos recém-criados:

```
# 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 adicionais

- Para informações sobre este procedimento e outros detalhes, incluindo exemplos administrativos, consulte a página de manual **lvmmcache(7)**.

13.4. POSSIBILITANDO O CACHE DM-CACHE COM UM CACHEPOOL PARA UM VOLUME LÓGICO

Este procedimento permite criar os dados do cache e os volumes lógicos dos metadados do cache individualmente e depois combinar os volumes em um pool de cache.

Pré-requisitos

- Um volume lógico lento que você deseja acelerar usando **dm-cache** existe em seu sistema.
- O grupo de volume que contém o volume lógico lento também contém um volume físico não utilizado em um dispositivo de bloqueio rápido.

Procedimento

1. Crie um volume **cachepool** no dispositivo rápido:

```
# lvcreate --type cache-pool --size cachepool-size --name fastpool vg /dev/fast
```

Substituir os seguintes valores:

cachepool-size

O tamanho do **cachepool**, tais como **5G**

fastpool

Um nome para o volume **cachepool**

vg

O nome do grupo de volume

/dev/fast

O caminho para o dispositivo de bloqueio rápido, tais como **/dev/sdf1**



NOTA

Você pode usar a opção **--poolmetadata** para especificar a localização dos metadados do pool ao criar o cache-pool.

2. Anexe o **cachepool** ao volume lógico principal para iniciar o caching:

```
# lvconvert -- tipo cache --cachepool fastpool vg/main
```

Substituir os seguintes valores:

fastpool

O nome do volume **cachepool**

vg

O nome do grupo de volume

main

O nome do volume lógico lento

Etapas de verificação

- Examine os dispositivos recém-criados:

```
# 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 adicionais

- A página do homem **lvcreate(8)**.
- A página do homem **lvmcache(7)**.
- A página do homem **lvconvert(8)**.

13.5. POSSIBILITANDO O CACHE DM-WRITECACHE PARA UM VOLUME LÓGICO

Este procedimento permite o cache de operações de E/S escritas a um volume lógico usando o método **dm-writecache**.

Pré-requisitos

- Um volume lógico lento que você deseja acelerar usando **dm-writecache** existe em seu sistema.
- O grupo de volume que contém o volume lógico lento também contém um volume físico não utilizado em um dispositivo de bloqueio rápido.

Procedimento

1. Se o volume lógico lento estiver ativo, desative-o:

```
# lvchange -- ativar n vg/main-lv
```

Substituir os seguintes valores:

vg

O nome do grupo de volume

main-lv

O nome do volume lógico lento

2. Criar um volume de **cachevol** desativado no dispositivo rápido:

```
# lvcreate --ativar n --size cachevol-size --nome fastvol vg /dev/fast-pv
```

Substituir os seguintes valores:

cachevol-size

O tamanho do volume **cachevol**, tais como **5G**

fastvol

Um nome para o volume **cachevol**

vg

O nome do grupo de volume

/dev/fast-pv

O caminho para o dispositivo de bloqueio rápido, tais como **/dev/sdf1**

- Anexe o volume **cachevol** ao volume lógico principal para iniciar o caching:

```
# lvconvert --type writecache --cachevol fastvol vg /main-lv
```

Substituir os seguintes valores:

fastvol

O nome do volume **cachevol**

vg

O nome do grupo de volume

main-lv

O nome do volume lógico lento

- Ativar o volume lógico resultante:

```
# lvchange -- ativar y vg/main-lv
```

Substituir os seguintes valores:

vg

O nome do grupo de volume

main-lv

O nome do volume lógico lento

Etapas de verificação

- Examine os dispositivos recém-criados:

```
# 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_cv] [main-lv_wcorig] 0.00
main-lv_wcorig(0)
[fastvol_cv] vg Cwi-aoC--- 252.00m
/dev/sdc1(0)
[main-lv_wcorig] vg owi-aoC--- 500.00m
/dev/sdb1(0)
```

Recursos adicionais

- Para informações, incluindo exemplos administrativos, consulte a página de manual **lvmdcache(7)**.

13.6. DESABILITANDO O CACHING PARA UM VOLUME LÓGICO

Este procedimento desabilita **dm-cache** ou **dm-writecache** caching que está atualmente habilitado em um volume lógico.

Pré-requisitos

- O cache é ativado em um volume lógico.

Procedimento

1. Desativar o volume lógico:

```
# lvchange -- ativar n vg/main-lv
```

Substituir os seguintes valores:

vg

O nome do grupo de volume

main-lv

O nome do volume lógico onde o caching é permitido

2. Desmonte o volume **cachevol** ou **cachepool**:

```
# lvconvert --splitcache vg/main-lv
```

Substituir os seguintes valores:

vg

O nome do grupo de volume

main-lv

O nome do volume lógico onde o caching é permitido

Etapas de verificação

- Verifique se os volumes lógicos não estão mais presos juntos:

```
# lvs --all --options +devices [replaceable]_vg_
```

```
LV   Attr   Type  Devices
fastvol -wi----- linear /dev/fast-pv
main-lv -wi----- linear /dev/slow-pv
```

Recursos adicionais

- A página do homem **lvmdcache(7)**

CAPÍTULO 14. ATIVAÇÃO DE VOLUME LÓGICO

Um volume lógico que é um estado ativo pode ser usado através de um dispositivo de bloco. Um volume lógico que é ativado é acessível e está sujeito a mudanças. Quando se cria um volume lógico, ele é ativado por padrão.

Há várias circunstâncias para as quais você precisa tornar um volume lógico individual inativo e, portanto, desconhecido para o núcleo. Você pode ativar ou desativar o volume lógico individual com a opção **-a** do comando **lvchange**.

O formato do comando para desativar um volume lógico individual é o seguinte.

```
lvchange -an vg/lv
```

O formato do comando para ativar um volume lógico individual é o seguinte.

```
lvchange -ay vg/lv
```

Você pode ativar ou desativar todos os volumes lógicos em um grupo de volumes com a opção **-a** do comando **vgchange**. Isto é o equivalente a executar o comando **lvchange -a** em cada volume lógico individual do grupo de volume.

O formato do comando para desativar todos os volumes lógicos em um grupo de volumes é o seguinte.

```
vgchange -an vg
```

O formato do comando para ativar todos os volumes lógicos em um grupo de volumes é o seguinte.

```
vgchange -ay vg
```

14.1. CONTROLE DA AUTO-ATIVAÇÃO DE VOLUMES LÓGICOS

A ativação automática de um volume lógico refere-se à ativação automática baseada em eventos de um volume lógico durante a inicialização do sistema. Conforme os dispositivos ficam disponíveis no sistema (eventos on-line do dispositivo), **systemd/udev** executa o serviço **lvm2-pvscan** para cada dispositivo. Este serviço executa o **pvscan --cache -aay device** que lê o dispositivo nomeado. Se o dispositivo pertence a um grupo de volume, o comando **pvscan** verificará se todos os volumes físicos para aquele grupo de volume estão presentes no sistema. Se assim for, o comando ativará os volumes lógicos nesse grupo de volumes.

Você pode usar as seguintes opções de configuração no arquivo de configuração **/etc/lvm/lvm.conf** para controlar a auto-ativação dos volumes lógicos.

- **global/event_activation**
Quando **event_activation** estiver desativado, **systemd/udev** ativará automaticamente o volume lógico somente em qualquer volume físico presente durante a inicialização do sistema. Se todos os volumes físicos ainda não aparecerem, então alguns volumes lógicos podem não ser auto-ativados.
- **activation/auto_activation_volume_list**
A configuração do **auto_activation_volume_list** para uma lista vazia desativa totalmente a ativação automática. O ajuste **auto_activation_volume_list** para volumes lógicos específicos e grupos de volumes limita a auto-ativação a esses volumes lógicos.

Para informações sobre a configuração destas opções, consulte o arquivo de configuração `/etc/lvm/lvm.conf`.

14.2. CONTROLE DA ATIVAÇÃO LÓGICA DO VOLUME

Você pode controlar a ativação do volume lógico das seguintes maneiras:

- Através da configuração `activation/volume_list` no arquivo `/etc/lvm/conf`. Isto permite especificar quais volumes lógicos são ativados. Para informações sobre o uso desta opção, consulte o arquivo de configuração `/etc/lvm/lvm.conf`.
- Por meio da bandeira de pulo de ativação para um volume lógico. Quando esta bandeira é definida para um volume lógico, o volume é pulado durante os comandos normais de ativação.

Você pode colocar a bandeira de pulo de ativação em um volume lógico das seguintes maneiras.

- Você pode desativar a bandeira de pulo de ativação ao criar um volume lógico, especificando a opção `-kn` ou `--setactivationskip n` do comando `lvcreate`.
- Você pode desativar a bandeira de pulo de ativação para um volume lógico existente especificando a opção `-kn` ou `--setactivationskip n` do comando `lvchange`.
- Você pode ligar a bandeira de pulo de ativação novamente para um volume onde foi desligado com a opção `-ky` ou `--setactivationskip y` do comando `lvchange`.

Para determinar se a bandeira de pulo de ativação está definida para um volume lógico, execute o comando `lvs`, que exibe o atributo `k`, como no exemplo a seguir.

```
# lvs vg/thin1s1
LV      VG Attr   LSize Pool Origin
thin1s1  vg Vwi---tz-k 1.00t pool0 thin1
```

Você pode ativar um volume lógico com o atributo `k` definido usando a opção `-K` ou `--ignoreactivationskip`, além da opção padrão `-ay` ou `--activate y`.

Por padrão, os volumes de fotos finas são marcados para ativação quando são criados. Você pode controlar a configuração padrão de salto de ativação em novos volumes de instantâneos finos com a configuração `auto_set_activation_skip` no arquivo `/etc/lvm/lvm.conf`.

O seguinte comando ativa um fino volume lógico de instantâneo que tem a bandeira de pulo de ativação definida.

```
# lvchange -ay -K VG/SnapLV
```

O seguinte comando cria uma fina foto sem a bandeira de pulo de ativação

```
# lvcreate --type thin -n SnapLV -kn -s ThinLV --thinpool VG/ThinPoolLV
```

O seguinte comando remove a bandeira de pulo de ativação de um volume lógico de instantâneo.

```
# lvchange -kn VG/SnapLV
```

14.3. ATIVAÇÃO DE VOLUMES LÓGICOS COMPARTILHADOS

Você pode controlar a ativação lógica de um volume lógico compartilhado com a opção **-a** dos comandos **lvchange** e **vgchange**, como a seguir.

Comando	Ativação
lvchange -ay e	Ativar o volume lógico compartilhado em modo exclusivo, permitindo que apenas um único host ative o volume lógico. Se a ativação falhar, como aconteceria se o volume lógico estivesse ativo em outro host, um erro é relatado.
lvchange -asy	Ativar o volume lógico compartilhado em modo compartilhado, permitindo que vários hospedeiros ativem o volume lógico ao mesmo tempo. Se a ativação falhar, como aconteceria se o volume lógico estivesse ativo exclusivamente em outro host, um erro é relatado. Se o tipo lógico proibir o acesso compartilhado, como um instantâneo, o comando irá reportar um erro e falhar. Os tipos de volume lógico que não podem ser usados simultaneamente a partir de múltiplos hosts incluem thin, cache, raid e snapshot.
lvchange -an	Desativar o volume lógico.

14.4. ATIVAÇÃO DE UM VOLUME LÓGICO COM DISPOSITIVOS AUSENTES

Você pode configurar quais volumes lógicos com dispositivos ausentes são ativados ao definir o parâmetro **activation_mode** com o comando **lvchange** para um dos seguintes valores.

Modo de ativação	Significado
completo	Permite ativar somente os volumes lógicos, sem a falta de volumes físicos. Este é o modo mais restritivo.
degradado	Permite que volumes lógicos RAID com volumes físicos ausentes sejam ativados.
parcial	Permite que qualquer volume lógico com volumes físicos ausentes seja ativado. Esta opção deve ser usada somente para recuperação ou reparo.

O valor padrão de **activation_mode** é determinado pela configuração **activation_mode** no arquivo **/etc/lvm/lvm.conf**. Para maiores informações, consulte a página de manual **lvraid(7)**.

CAPÍTULO 15. CONTROLE DA VARREDURA DO DISPOSITIVO LVM

Você pode controlar a leitura do dispositivo LVM configurando os filtros no arquivo `/etc/lvm/lvm.conf`. Os filtros no arquivo `lvm.conf` consistem de uma série de expressões regulares simples que são aplicadas aos nomes dos dispositivos no diretório `/dev` para decidir se aceita ou rejeita cada dispositivo de bloco encontrado.

15.1. O FILTRO DO DISPOSITIVO LVM

As ferramentas LVM procuram por dispositivos no diretório `/dev` e verificam todos os dispositivos lá em busca de metadados LVM. Um filtro no arquivo `/etc/lvm/lvm.conf` controla quais dispositivos LVM escaneia.

O filtro é uma lista de padrões que a LVM aplica a cada dispositivo encontrado por uma varredura do diretório `/dev`, ou o diretório especificado pela palavra-chave `dir` no arquivo `/etc/lvm/lvm.conf`. Os padrões são expressões regulares delimitadas por qualquer caracter e precedidas por `a` para *accept* ou `r` para *reject*. A primeira expressão regular na lista que corresponde a um dispositivo determina se a LVM aceita ou rejeita (ignora) o dispositivo. A LVM aceita dispositivos que não correspondem a nenhum padrão.

A seguir está a configuração padrão do filtro, que escaneia todos os dispositivos:

```
filtro = [ \ a/.*/ ]
```

15.2. EXEMPLOS DE CONFIGURAÇÕES DE FILTROS DE DISPOSITIVOS LVM

Os exemplos a seguir mostram o uso de filtros para controlar quais dispositivos LVM scans.



ATENÇÃO

Alguns dos exemplos aqui apresentados podem combinar involuntariamente dispositivos extras no sistema e podem não representar uma prática recomendada para o seu sistema. Por exemplo, `a/loop/` é equivalente a `a.*loop.*/` e seria igual a `/dev/solooperation/lvol1`.

- O seguinte filtro adiciona todos os dispositivos descobertos, que é o comportamento padrão porque nenhum filtro é configurado no arquivo de configuração:

```
filtro = [ \ a/.*/ ]
```

- O seguinte filtro remove o dispositivo `cdrom` a fim de evitar atrasos se a unidade não contiver mídia:

```
filtro = [ "r|^/dev/cdrom$" ]
```

- O seguinte filtro adiciona todos os dispositivos de laço e remove todos os outros dispositivos de bloco:

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

- O seguinte filtro adiciona todos os dispositivos de laço e IDE e remove todos os outros dispositivos de bloco:

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

- O seguinte filtro adiciona apenas a partição 8 na primeira unidade IDE e remove todos os outros dispositivos de bloco:

```
filtro = [ \ a/^dev/hda8$, "r.*//____ " ]
```

15.3. APLICAÇÃO DE UMA CONFIGURAÇÃO DE FILTRO DO DISPOSITIVO LVM

Este procedimento altera a configuração do filtro do dispositivo LVM, que controla os dispositivos que o LVM escaneia.

Pré-requisitos

- Prepare o padrão de filtro do dispositivo que você deseja usar.

Procedimento

1. Teste o padrão de filtragem de seu dispositivo sem modificar o arquivo `/etc/lvm/lvm.conf`. Use um comando LVM com o `--config 'devices{ filter = [your device filter pattern] }` opção. Por exemplo:

```
# lvs --configurar 'dispositivos (filtro = [ a|dev/emcpower.*", "r|.*" ]'
```

2. Edite a opção **filter** no arquivo de configuração `/etc/lvm/lvm.conf` para usar seu novo padrão de filtro de dispositivo.
3. Verifique se não estão faltando volumes físicos ou grupos de volume que você deseja utilizar com a nova configuração:

```
# pvscan
```

```
# vgscan
```

4. Reconstruir o sistema de arquivos **initramfs** para que a LVM escaneie somente os dispositivos necessários após a reinicialização:

```
# dracut --force --verbose
```

CAPÍTULO 16. SOBREPOSIÇÃO DE VOLUMES FÍSICOS LVM SOBRE VOLUMES LÓGICOS

Você pode configurar o LVM para que seja possível criar volumes físicos em cima de volumes lógicos.

Por padrão, os comandos LVM não digitalizam os volumes lógicos em seu sistema. Este comportamento padrão oferece as seguintes vantagens:

- Se houver muitos volumes lógicos ativos no sistema, cada comando LVM exigiria tempo adicional, impactando negativamente o desempenho e causando atrasos ou interrupções indesejáveis.
- Se os volumes lógicos contiverem volumes físicos de uma imagem VM hóspede, o host geralmente não quer escanear ou usar aqueles volumes físicos em camadas que pertencem ao hóspede. Note, entretanto, que nos casos em que o volume físico de uma VM hóspede existe diretamente em um dispositivo SCSI visível ao host, a fim de evitar que a LVM no host acesse esses volumes físicos, será necessário configurar um filtro, como descrito em [Capítulo 15, Controle da varredura do dispositivo LVM](#).

A digitalização de volumes lógicos pode ser necessária quando a sobreposição de volumes físicos em cima de volumes lógicos é intencional. Isto permitirá que o comando **pvcreate** seja executado sobre um volume lógico. Para configurar o LVM para escanear todos os volumes lógicos, defina a opção de configuração **scan_lvs** no arquivo **/etc/lvm/lvm.conf** para **scan_lvs=1**. Para restringir quais volumes lógicos os comandos LVM podem ser escaneados, pode-se então configurar filtros de dispositivos no arquivo de configuração **/etc/lvm/lvm.conf**, conforme descrito em [Capítulo 15, Controle da varredura do dispositivo LVM](#).

CAPÍTULO 17. CONTROLE DA ALOCAÇÃO DA LVM

Por padrão, um grupo de volume aloca extensões físicas de acordo com regras de senso comum, tais como não colocar faixas paralelas no mesmo volume físico. Esta é a política de alocação do **normal**. Você pode usar o argumento **--alloc** do comando **vgcreate** para especificar uma política de alocação de **contiguous**, **anywhere**, ou **cling**. Em geral, as políticas de alocação que não **normal** são necessárias somente em casos especiais onde é necessário especificar uma alocação de extensão incomum ou não padronizada.

17.1. POLÍTICAS DE ALOCAÇÃO DA LVM

Quando uma operação LVM precisa alocar extensões físicas para um ou mais volumes lógicos, a alocação procede da seguinte forma:

- O conjunto completo de extensões físicas não alocadas no grupo de volume é gerado para consideração. Se você fornecer qualquer faixa de extensões físicas no final da linha de comando, somente extensões físicas não alocadas dentro dessas faixas nos volumes físicos especificados são consideradas.
- Cada política de alocação é tentada por sua vez, começando com a política mais rigorosa (**contiguous**) e terminando com a política de alocação especificada usando a opção **--alloc** ou definida como padrão para o volume ou grupo de volume lógico particular. Para cada política, trabalhando a partir da extensão lógica mais baixa do espaço de volume lógico vazio que precisa ser preenchido, é alocado o máximo de espaço possível, de acordo com as restrições impostas pela política de alocação. Se for necessário mais espaço, a LVM passa para a política seguinte.

As restrições da política de alocação são as seguintes:

- Uma política de alocação do **contiguous** requer que a localização física de qualquer extensão lógica que não seja a primeira extensão lógica de um volume lógico seja adjacente à localização física da extensão lógica imediatamente anterior a ela.
Quando um volume lógico é listrado ou espelhado, a restrição de alocação **contiguous** é aplicada independentemente a cada listra ou imagem de espelho (perna) que necessita de espaço.
- Uma política de alocação do **cling** exige que o volume físico usado para qualquer extensão lógica seja adicionado a um volume lógico existente que já esteja em uso pelo menos uma extensão lógica mais cedo nesse volume lógico. Se o parâmetro de configuração **allocation/cling_tag_list** for definido, então dois volumes físicos são considerados iguais se alguma das tags listadas estiver presente em ambos os volumes físicos. Isto permite que grupos de volumes físicos com propriedades similares (como sua localização física) sejam etiquetados e tratados como equivalentes para fins de alocação.
Quando um volume lógico é listrado ou espelhado, a restrição de alocação **cling** é aplicada independentemente a cada listra ou imagem de espelho (perna) que necessita de espaço.
- Uma política de alocação do **normal** não escolherá uma extensão física que compartilhe o mesmo volume físico que uma extensão lógica já alocada a um volume lógico paralelo (ou seja, uma faixa ou imagem/perna espelhada diferente) com o mesmo offset dentro desse volume lógico paralelo.
Ao alocar um registro espelho ao mesmo tempo que os volumes lógicos para armazenar os dados espelho, uma política de alocação do **normal** tentará primeiro selecionar diferentes volumes físicos para o registro e os dados. Se isso não for possível e o parâmetro de configuração **allocation/mirror_logs_require_separate_pvs** for definido como 0, ele então permitirá que o log compartilhe volume(s) físico(s) com parte dos dados.

Da mesma forma, ao alocar metadados do thin pool, uma política de alocação de **normal** seguirá as mesmas considerações que para a alocação de um registro espelho, com base no valor do parâmetro de configuração **allocation/thin_pool_metadata_require_separate_pvs**.

- Se houver extensões livres suficientes para satisfazer um pedido de alocação, mas uma política de alocação **normal** não as utilizaria, a política de alocação **anywhere** irá, mesmo que isso reduza o desempenho, colocando duas faixas no mesmo volume físico.

As políticas de alocação podem ser alteradas usando o comando **vgchange**.



NOTA

Se você confiar em qualquer comportamento de layout além daquele documentado nesta seção de acordo com as políticas de alocação definidas, você deve observar que isto pode mudar em versões futuras do código. Por exemplo, se você fornecer na linha de comando dois volumes físicos vazios que tenham um número idêntico de extensões físicas livres disponíveis para alocação, a LVM atualmente considera usar cada um deles na ordem em que estão listados; não há garantia de que futuras liberações manterão essa propriedade. Se for importante obter um layout específico para um determinado volume lógico, então você deve construí-lo através de uma seqüência de etapas de **lvcreate** e **lvconvert**, de modo que as políticas de alocação aplicadas a cada etapa não deixem à LVM nenhum arbítrio sobre o layout.

Para ver a forma como o processo de alocação funciona atualmente em qualquer caso específico, você pode ler a saída de registro de depuração, por exemplo, adicionando a opção **-vvvv** a um comando.

17.2. IMPEDINDO A ALOCAÇÃO EM UM VOLUME FÍSICO

Você pode impedir a alocação de extensões físicas no espaço livre de um ou mais volumes físicos com o comando **pvchange**. Isto pode ser necessário se houver erros de disco, ou se você estiver removendo o volume físico.

O seguinte comando desativa a alocação de extensões físicas em **/dev/sdk1**.

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

Você também pode usar os argumentos **-xy** do comando **pvchange** para permitir a alocação onde ela havia sido previamente proibida.

17.3. AMPLIAÇÃO DE UM VOLUME LÓGICO COM A POLÍTICA DE ALOCAÇÃO CLING

Ao estender um volume LVM, você pode usar a opção **--alloc cling** do comando **lvextend** para especificar a política de alocação **cling**. Esta política escolherá espaço nos mesmos volumes físicos que o último segmento do volume lógico existente. Se não houver espaço suficiente nos volumes físicos e uma lista de tags estiver definida no arquivo **/etc/lvm/lvm.conf**, a LVM verificará se alguma das tags está anexada aos volumes físicos e procurará combinar essas tags de volume físico entre as extensões existentes e as novas extensões.

Por exemplo, se você tiver volumes lógicos que são espelhados entre dois sites dentro de um único grupo de volumes, você pode etiquetar os volumes físicos de acordo com o local onde eles estão situados, etiquetando os volumes físicos com as tags **@site1** e **@site2**. Você pode então especificar a seguinte linha no arquivo **lvm.conf**:

■

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

No exemplo a seguir, o arquivo **lvm.conf** foi modificado para conter a seguinte linha:

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

Também neste exemplo, foi criado um grupo de volumes **taft** que consiste nos volumes físicos **/dev/sdb1**, **/dev/sdc1**, **/dev/sdd1**, **/dev/sde1**, **/dev/sdf1**, **/dev/sdg1**, e **/dev/sdh1**. Estes volumes físicos foram marcados com tags **A**, **B**, e **C**. O exemplo não usa a tag **C**, mas isto mostrará que a LVM usa as tags para selecionar quais volumes físicos usar para as pernas do espelho.

```
# 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
```

O seguinte comando cria um volume espelhado de 10 gigabytes do grupo de volume **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
```

O seguinte comando mostra quais dispositivos são usados para as pernas do espelho e os subvolumes de metadados 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)
```

O seguinte comando estende o tamanho do volume espelhado, usando a política de alocação **cling** para indicar que as pernas do espelho devem ser estendidas usando volumes físicos com a mesma 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
```

O seguinte comando de exibição mostra que as pernas do espelho foram estendidas usando volumes físicos com a mesma etiqueta que a perna. Note que os volumes físicos com a mesma tag de **C** foram 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. DIFERENCIANDO ENTRE OBJETOS RAID LVM USANDO TAGS

Você pode atribuir tags aos objetos LVM RAID para agrupá-los, de modo que você possa automatizar o controle do comportamento LVM RAID, como a ativação, por grupo.

As tags de volume físico (PV) são responsáveis pelo controle da alocação no ataque LVM, ao contrário das tags de volume lógico (LV) ou de grupo de volume (VG), pois a alocação em lvm ocorre no nível PV com base nas políticas de alocação. Para distinguir os tipos de armazenamento por suas diferentes propriedades, etiquetá-los adequadamente (por exemplo, NVMe, SSD, HDD). A Red Hat recomenda que você marque cada novo PV apropriadamente depois de adicioná-lo a um VG.

Este procedimento adiciona tags de objeto a seus volumes lógicos, assumindo que **/dev/sda** é uma SSD, e **/dev/sd[b-f]** são HDDs com uma partição.

Pré-requisitos

- O pacote **lvm2** está instalado.
- Estão disponíveis dispositivos de armazenamento para uso como PVs.

Procedimento

1. Criar um grupo de volume.

```
# vgcreate MyVG /dev/sd[a-f]1
```

2. Adicione etiquetas a seus volumes físicos.

```
# pvchange --addtag ssds /dev/sda1
# pvchange --addtag hdds /dev/sd[b-f]1
```

3. Criar um volume lógico RAID6.

```
# lvcreate --type raid6 --stripes 3 -L1G -nr6 MyVG @hdds
```

4. Criar um volume de cache linear.

```
# lvcreate -nr6pool -L512m MyVG @ssds
```

5. Converter o volume RAID6 para ser colocado em cache.

```
# lvconvert -- tipo cache --cachevol MyVG/r6pool MyVG/r6
```

Recursos adicionais

- As páginas de manual **lvcreate(8)**, **lvconvert(8)**, **lvraid(7)** e **lvncache(7)**.

CAPÍTULO 18. AGRUPANDO OBJETOS LVM COM TAGS

Como administrador do sistema, você pode atribuir tags aos objetos LVM para agrupá-los, para que você possa automatizar o controle do comportamento do LVM, como a ativação, por grupo.

18.1. ETIQUETAS DE OBJETOS LVM

Uma etiqueta LVM é uma palavra que é usada para agrupar objetos LVM2 do mesmo tipo. As tags são anexadas a objetos como volumes físicos, grupos de volumes e volumes lógicos, assim como a anfitriões em uma configuração de cluster.

As etiquetas são dadas na linha de comando no lugar dos argumentos PV, VG ou LV. As etiquetas devem ser prefixadas com @ para evitar ambigüidade. Cada etiqueta é expandida substituindo-a por todos os objetos que possuem aquela etiqueta, que são do tipo esperado por sua posição na linha de comando.

As etiquetas LVM são cordas de até 1024 caracteres. As tags LVM não podem começar com um hífen.

Uma etiqueta válida consiste de uma gama limitada de caracteres apenas. Os caracteres permitidos são **A-Z a-z 0-9 _ . - / = ! : # &**.

Somente objetos em um grupo de volume podem ser etiquetados. Os volumes físicos perdem suas tags se forem removidos de um grupo de volume; isto porque as tags são armazenadas como parte dos metadados do grupo de volume e isso é excluído quando um volume físico é removido.

18.2. LISTAGEM DAS ETIQUETAS LVM

O exemplo a seguir mostra como listar as etiquetas LVM.

Procedimento

- Use o seguinte comando para listar todos os volumes lógicos com a tag **database**:

```
# lvs @database
```

- Use o seguinte comando para listar as tags do host atualmente ativas:

```
# lvm tags
```

18.3. ADICIONANDO TAGS DE OBJETOS LVM

Este procedimento descreve como adicionar tags de objetos LVM.

Pré-requisitos

- O pacote **lvm2** está instalado.
- São criados um ou mais volumes físicos, grupos de volumes ou volumes lógicos.

Procedimento

- Para criar uma tag de objeto, adicione a opção **--addtag** a um comando LVM:

- Para criar tags a partir de volumes físicos, adicione a opção ao comando **pvchange**.
- Para criar tags a partir de grupos de volume, adicione a opção aos comandos **vgchange** ou **vgcreate**.
- Para criar tags a partir de volumes lógicos, adicione a opção aos comandos **lvchange** ou **lvcreate**.

18.4. REMOÇÃO DAS ETIQUETAS DE OBJETOS LVM

Este procedimento descreve como remover as etiquetas dos objetos LVM.

Pré-requisitos

- O pacote **lvm2** está instalado.
- São criadas etiquetas de objetos em volumes físicos, grupos de volumes ou volumes lógicos.

Procedimento

- Para excluir uma etiqueta de objeto, adicione a opção **--deltag** a um comando LVM:
 - Para excluir tags de volumes físicos, adicione a opção ao comando **pvchange**.
 - Para excluir tags de grupos de volume, adicione a opção aos comandos **vgchange** ou **vgcreate**.
 - Para excluir tags dos volumes lógicos, adicione a opção aos comandos **lvchange** ou **lvcreate**.

18.5. DEFININDO AS ETIQUETAS DO HOSPEDEIRO LVM

Este procedimento descreve como definir as tags do host LVM em uma configuração de cluster. É possível definir as tags de host nos arquivos de configuração.

Procedimento

- Defina **hosttags = 1** na seção **tags** para definir automaticamente o host tag usando o nome do host da máquina.
Isto permite que você utilize um arquivo de configuração comum que pode ser replicado em todas as suas máquinas para que elas mantenham cópias idênticas do arquivo, mas o comportamento pode diferir entre as máquinas de acordo com o nome do host.

Para cada tag do host, um arquivo de configuração extra é lido, caso exista: **lvm_hosttag.conf**. Se esse arquivo definir novas tags, então outros arquivos de configuração serão anexados à lista de arquivos a serem lidos.

Por exemplo, a seguinte entrada no arquivo de configuração sempre define **tag1**, e define **tag2** se o nome do host for **host1**:

```
tags { tag1 { } tag2 { host_list = ["host1"] } }
```

18.6. CONTROLE DA ATIVAÇÃO LÓGICA DE VOLUME COM TAGS

Este procedimento descreve como especificar no arquivo de configuração que somente determinados volumes lógicos devem ser ativados naquele host.

Pré-requisitos

- Uma lista de condições que devem ser satisfeitas antes que o usuário comece a seguir esta montagem.
- Também é possível vincular-se a outros módulos ou montagens que o usuário deve seguir antes de iniciar esta montagem.
- Eliminar o título da seção e as balas se a assembléia não tiver pré-requisitos.

Procedimento

Por exemplo, a seguinte entrada atua como um filtro para solicitações de ativação (como **vgchange -ay**) e só ativa **vg1/lvol0** e quaisquer volumes lógicos ou grupos de volume com a tag **database** nos metadados daquele host:

```
ativação { lista_de_volume = [vg1/lvol0], "@base_de_dados"
```

A correspondência especial **@*** que causa uma correspondência somente se qualquer tag de metadados corresponder a qualquer tag do host naquela máquina.

Como outro exemplo, considere uma situação em que cada máquina do cluster tem a seguinte entrada no arquivo de configuração:

```
tags { hosttags = 1 }
```

Se você quiser ativar **vg1/lvol2** somente no host **db2**, faça o seguinte:

1. Execute **lvchange --addtag @db2 vg1/lvol2** a partir de qualquer anfitrião do cluster.
2. Rodar **lvchange -ay vg1/lvol2**.

Esta solução envolve o armazenamento de nomes de host dentro dos metadados do grupo de volume.

CAPÍTULO 19. SOLUÇÃO DE PROBLEMAS DE LVM

Você pode usar ferramentas LVM para solucionar uma variedade de problemas em volumes e grupos de LVM.

19.1. COLETA DE DADOS DE DIAGNÓSTICO SOBRE LVM

Se um comando LVM não estiver funcionando como esperado, você pode reunir diagnósticos das seguintes maneiras.

Procedimento

- Use os seguintes métodos para coletar diferentes tipos de dados de diagnóstico:
 - Adicione o argumento **-vvvv** a qualquer comando LVM para aumentar o nível de verbosidade da saída do comando.
 - Na seção **log** do arquivo de configuração **/etc/lvm/lvm.conf**, aumente o valor da opção **level**. Isto faz com que a LVM forneça mais detalhes no log do sistema.
 - Se o problema estiver relacionado com a ativação lógica do volume, habilite a LVM a registrar mensagens durante a ativação:
 - i. Defina a opção **activation = 1** na seção **log** do arquivo de configuração **/etc/lvm/lvm.conf**.
 - ii. Execute o comando LVM com a opção **-vvvv**.
 - iii. Examine a saída do comando.
 - iv. Redefinir a opção **activation** para **0**.
Se você não reiniciar a opção para **0**, o sistema pode ficar sem resposta durante situações de pouca memória.

- Exibir uma lixeira de informações para fins de diagnóstico:

```
# lvmdump
```

- Exibir informações adicionais do sistema:

```
# lvs -v
```

```
# pvs -- todos
```

```
# dmsetup info --colunas
```

- Examine o último backup dos metadados da LVM no diretório **/etc/lvm/backup/** e as versões arquivadas no diretório **/etc/lvm/archive/**.

- Verifique as informações de configuração atuais:

```
# lvmconfig
```

- Verifique o arquivo cache **/run/lvm/hints** para saber quais dispositivos têm volumes físicos.

Recursos adicionais

- A página do homem **lvmdump(8)**

19.2. EXIBIÇÃO DE INFORMAÇÕES SOBRE DISPOSITIVOS LVM COM FALHAS

Você pode exibir informações sobre um volume LVM falho que podem ajudá-lo a determinar por que o volume falhou.

Procedimento

- Mostrar os volumes falhados usando o utilitário **vgs** ou **lvs**.

Exemplo 19.1. Grupos de volume fracassados

Neste exemplo, um dos dispositivos que compunham o grupo de volume **vg** falhou. O grupo de volume é inutilizável, mas você pode ver informações sobre o dispositivo que falhou.

```
# 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/stripe 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)
```

Exemplo 19.2. Falha no LV linear e listrado

Neste exemplo, o dispositivo falhado fez com que tanto um volume linear quanto um volume lógico listrado no grupo de volume falhassem. A saída do comando mostra os volumes lógicos que falharam.

```
# 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/stripe 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)

```

Exemplo 19.3. Perna com defeito de um volume lógico espelhado

Os exemplos a seguir mostram a saída do comando dos utilitários **vgs** e **lvs** quando uma perna de um volume lógico espelhado falhou.

```

# 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. REMOÇÃO DE VOLUMES FÍSICOS LVM PERDIDOS DE UM GRUPO DE VOLUME

Se um volume físico falhar, você pode ativar os volumes físicos restantes no grupo de volume e remover todos os volumes lógicos que usaram esse volume físico do grupo de volume.

Procedimento

1. Ativar os volumes físicos restantes no grupo de volume:

```
# vgchange -- ativar y -parcial volume-group
```

2. Verifique quais volumes lógicos serão removidos:

```
# vgreduce -- destituição --teste volume-group
```

3. Remover todos os volumes lógicos que utilizaram o volume físico perdido do grupo de volume:

```
# vgreduce -- retirada da força --force volume-group
```

4. Opcional: Se você removeu acidentalmente volumes lógicos que queria manter, você pode reverter a operação **vgreduce**:

■

```
# vgcfgrestore volume-group
```



ATENÇÃO

Se você removeu um pool fino, a LVM não pode reverter a operação.

19.4. RECUPERANDO UM VOLUME FÍSICO LVM COM METADADOS DANIFICADOS

Se a área de metadados do grupo de volume de um volume físico for acidentalmente sobrescrita ou destruída, você recebe uma mensagem de erro indicando que a área de metadados está incorreta, ou que o sistema não foi capaz de encontrar um volume físico com um UUID específico. Você pode ser capaz de recuperar os dados do volume físico, reescrevendo a área de metadados no volume físico.

19.4.1. Descobrimo que um volume LVM tem metadados ausentes ou corrompidos

O exemplo a seguir mostra a saída de comando que você pode ver se a área de metadados em um volume físico está faltando ou corrompida.

Procedimento

- Tente listar os volumes lógicos:

```
# lvs -- todos -- dispositivos de opção
```

Exemplo 19.4. Saída com metadados ausentes ou corrompidos

Neste exemplo, certos volumes lógicos estão localizados em um volume físico que tem metadados ausentes ou corrompidos.

```
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. Encontrando os metadados de um volume físico LVM ausente

Este procedimento encontra os últimos metadados arquivados de um volume físico que está faltando ou corrompido.

Procedimento

1. Encontre o arquivo de metadados arquivado do grupo de volume que contém o volume físico.

Os arquivos de metadados arquivados estão localizados no **/etc/lvm/archive/volume-group-name_backup-number.vg** caminho. Selecione o último arquivo de metadados válido conhecido, que tem o maior número para o grupo de volume.

2. Encontre o UUID do volume físico. Use um dos seguintes métodos.

- Liste os volumes lógicos:

```
# lvs --all --options +devices

Couldn't find device with uuid 'FmGRh3-zhok-iVI8-7qTD-S5BI-MAEN-NYM5SK.
```

- Examine o arquivo de metadados arquivado. Encontre a UUID como o valor rotulado **id =** na seção **physical_volumes** da configuração do grupo de volume.

- Desativar o grupo de volume usando a opção **--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. Restaurando metadados em um volume físico LVM

Este procedimento restaura metadados em um volume físico que está corrompido ou substituído por um novo dispositivo.



ATENÇÃO

Não tente este procedimento em um volume lógico LVM funcional. Você perderá seus dados se você especificar a UUID incorreta.

Pré-requisitos

- Você identificou os metadados do volume físico em falta. Para maiores detalhes, veja [Seção 19.4.2, “Encontrando os metadados de um volume físico LVM ausente”](#).

Procedimento

1. Restaurar os metadados sobre o volume físico:

```
# pvcreate --uuid physical-volume-uuid \
  --restorefile /etc/lvm/archive/volume-group-name_backup-number.vg \
  block-device
```



NOTA

O comando substitui apenas as áreas de metadados LVM e não afeta as áreas de dados existentes.

Exemplo 19.5. Restaurando um volume físico em /dev/sdh1

O seguinte exemplo rotula o dispositivo **/dev/sdh1** como um volume físico com as seguintes propriedades:

- A UUID de **FmGRh3-zhok-iVI8-7qTD-S5BI-MAEN-NYM5Sk**
- As informações de metadados contidas em **VG_00050.vg**, que é o mais recente bom metadados arquivados para o grupo de volume

```
# 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 os metadados do grupo de volume:

```
# vgcfgrestore volume-group-name

Restored volume group volume-group-name
```

3. Mostrar os volumes lógicos no grupo de volume:

```
# lvs -- todos -- dispositivos de opção volume-group-name
```

Os volumes lógicos estão atualmente inativos. Por exemplo, os volumes lógicos estão atualmente inativos:

```
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. Se o tipo de segmento dos volumes lógicos for RAID ou espelho, resincronize os volumes lógicos:

```
# lvchange --resync volume-group-name/logical-volume-name
```

5. Ativar os volumes lógicos:

```
# lvchange -- ativar y /dev/volume-group-name/logical-volume-name
```

6. Se os metadados LVM em disco ocuparem pelo menos tanto espaço quanto o que os ultrapassa, este procedimento pode recuperar o volume físico. Se o que ultrapassou os metadados passou pela área dos metadados, os dados sobre o volume podem ter sido afetados. Você pode ser capaz de usar o comando `fsck` para recuperar esses dados.

Etapas de verificação

- Mostrar os volumes lógicos ativos:

```
# 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. SUBSTITUIÇÃO DE UM VOLUME FÍSICO LVM AUSENTE

Se um volume físico falhar ou precisar ser substituído, você pode rotular um novo volume físico para substituir aquele que foi perdido no grupo de volume existente.

Pré-requisitos

- Você substituiu o volume físico por um novo dispositivo de armazenamento.
TODO: Reavaliar a colocação desta etapa.

19.5.1. Encontrando os metadados de um volume físico LVM ausente

Este procedimento encontra os últimos metadados arquivados de um volume físico que está faltando ou corrompido.

Procedimento

- Encontre o arquivo de metadados arquivado do grupo de volume que contém o volume físico. Os arquivos de metadados arquivados estão localizados no **/etc/lvm/archive/volume-group-name_backup-number.vg** caminho. Selecione o último arquivo de metadados válido conhecido, que tem o maior número para o grupo de volume.
- Encontre o UUID do volume físico. Use um dos seguintes métodos.

- Liste os volumes lógicos:

```
# lvs --all --options +devices

Couldn't find device with uuid 'FmGRh3-zhok-iVI8-7qTD-S5BI-MAEN-NYM5SK.'
```

- Examinar o arquivo de metadados arquivado. Encontre a UUID como o valor rotulado **id =** na seção **physical_volumes** da configuração do grupo de volume.
- Desativar o grupo de volume usando a opção **--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. Restaurando metadados em um volume físico LVM

Este procedimento restaura metadados em um volume físico que está corrompido ou substituído por um novo dispositivo.



ATENÇÃO

Não tente este procedimento em um volume lógico LVM funcional. Você perderá seus dados se você especificar a UUID incorreta.

Pré-requisitos

- Você identificou os metadados do volume físico em falta. Para maiores detalhes, veja [Seção 19.5.1, “Encontrando os metadados de um volume físico LVM ausente”](#).

Procedimento

1. Restaurar os metadados sobre o volume físico:

```
# pvcreate --uuid physical-volume-uuid \
  --restorefile /etc/lvm/archive/volume-group-name_backup-number.vg \
  block-device
```



NOTA

O comando substitui apenas as áreas de metadados LVM e não afeta as áreas de dados existentes.

Exemplo 19.6. Restaurando um volume físico em /dev/sdh1

O seguinte exemplo rotula o dispositivo **/dev/sdh1** como um volume físico com as seguintes propriedades:

- A UUID de **FmGRh3-zhok-iVI8-7qTD-S5BI-MAEN-NYM5Sk**
- As informações de metadados contidas em **VG_00050.vg**, que é o mais recente bom metadados arquivados para o grupo de volume

```
# 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 os metadados do grupo de volume:

```
# vgcfgrestore volume-group-name

Restored volume group volume-group-name
```

3. Mostrar os volumes lógicos no grupo de volume:

```
# lvs -- todos -- dispositivos de opção volume-group-name
```

Os volumes lógicos estão atualmente inativos. Por exemplo, os volumes lógicos estão atualmente inativos:

```
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. Se o tipo de segmento dos volumes lógicos for RAID ou espelho, resincronize os volumes lógicos:

```
# lvchange --resync volume-group-name/logical-volume-name
```

5. Ativar os volumes lógicos:

```
# lvchange -- ativar y /dev/volume-group-name/logical-volume-name
```

6. Se os metadados LVM em disco ocuparem pelo menos tanto espaço quanto o que os ultrapassa, este procedimento pode recuperar o volume físico. Se o que ultrapassou os metadados passou pela área dos metadados, os dados sobre o volume podem ter sido afetados. Você pode ser capaz de usar o comando `fsck` para recuperar esses dados.

Etapas de verificação

- Mostrar os volumes lógicos ativos:

```
# 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. SOLUÇÃO DE PROBLEMAS DE RAID LVM

Você pode solucionar vários problemas em dispositivos LVM RAID para corrigir erros de dados, recuperar dispositivos ou substituir dispositivos falhos.

19.6.1. Verificação da coerência dos dados em um volume lógico RAID (scrubbing RAID)

O LVM fornece suporte à lavagem de volumes lógicos RAID. A depuração RAID é o processo de leitura de todos os dados e blocos de paridade em uma matriz e verificação para ver se eles são coerentes.

Procedimento

1. Opcional: Limite a largura de banda de E/S que o processo de depuração utiliza. Quando você realiza uma operação de scrubbing RAID, as E/S de fundo exigidas pelas operações do **sync** podem excluir outras E/S para dispositivos LVM, tais como atualizações de metadados de grupos de volume. Isto pode fazer com que as outras operações LVM diminuam a velocidade. Você pode controlar a taxa da operação de scrubbing implementando o estrangulamento de recuperação.

Adicione as seguintes opções aos comandos **lvchange --syncaction** nos próximos passos:

--maxrecoveryrate *Rate*[bBsSkKmMgG]

Define a taxa máxima de recuperação de modo que a operação exclua as operações de E/S nominais. Ajustar a taxa de recuperação para 0 significa que a operação não está vinculada.

--minrecoveryrate *Rate*[bBsSkKmMgG]

Define a taxa mínima de recuperação para garantir que as E/S para as operações do site **sync** atinjam um rendimento mínimo, mesmo quando a E/S nominal pesada está presente.

Especifique o valor *Rate* como um valor por segundo para cada dispositivo da matriz. Se você não fornecer nenhum sufixo, as opções assumem kiB por segundo por dispositivo.

2. Mostrar o número de discrepâncias na matriz, sem repará-las:

```
# lvchange - verificação de sincronia vg/raid_lv
```

3. Corrigir as discrepâncias na matriz:

```
# lvchange --syncaction repair vg/raid_lv
```



NOTA

A operação **lvchange --syncaction repair** não desempenha a mesma função que a operação **lvconvert --repair**:

- A operação **lvchange --syncaction repair** inicia uma operação de sincronização de fundo na matriz.
- A operação **lvconvert --repair** repara ou substitui dispositivos falhos em um espelho ou em um volume lógico RAID.

4. Opcional: Exibir informações sobre a operação de depuração:

```
# lvs -o raid_sync_action,raid_mismatch_count vg/lv
```

- O campo **raid_sync_action** exibe a operação de sincronização atual que o volume RAID está realizando. Ele pode ser um dos seguintes valores:

idle

Todas as operações de sincronização concluídas (não fazer nada)

resync

Inicialização de uma matriz ou recuperação após uma falha da máquina

recover

Substituição de um dispositivo na matriz

check

Procurando por inconsistências de matriz

repair

Procura e reparo de inconsistências

- O campo **raid_mismatch_count** mostra o número de discrepâncias encontradas durante uma operação **check**.
- O campo **Cpy%Sync** mostra o progresso das operações do **sync**.
- O campo **lv_attr** fornece indicadores adicionais. O bit 9 deste campo exibe a saúde do volume lógico, e suporta os seguintes indicadores:
 - **m** (desajustes) indica que há discrepâncias em um volume lógico RAID. Este caráter é mostrado após uma operação de scrubbing ter detectado que partes do RAID não são coerentes.
 - **r** (refresh) indica que um dispositivo em um array RAID sofreu uma falha e o kernel o considera uma falha, mesmo que o LVM possa ler a etiqueta do dispositivo e considere o dispositivo como operacional. Atualize o volume lógico para notificar o kernel de que o dispositivo está agora disponível, ou substitua o dispositivo se você suspeitar que ele falhou.

Recursos adicionais

- Para mais informações, consulte as páginas de manual **lvchange(8)** e **lvraid(7)**.

19.6.2. Dispositivos falhados em RAID LVM

RAID não é como o tradicional espelhamento LVM. O espelhamento LVM exigia a remoção de dispositivos com falhas ou o volume lógico do espelhamento seria pendurado. As matrizes RAID podem continuar a funcionar com dispositivos com falhas. Na verdade, para outros tipos de RAID além do RAID1, remover um dispositivo significaria converter para um RAID de nível inferior (por exemplo, de RAID6 para RAID5, ou de RAID4 ou RAID5 para RAID0).

Portanto, ao invés de remover incondicionalmente e potencialmente alocar um dispositivo falhado, o LVM permite substituir um dispositivo falhado em um volume RAID em uma solução de uma etapa, usando o argumento **--repair** do comando **lvconvert**.

19.6.3. Recuperando um dispositivo RAID falho em um volume lógico

Se a falha do dispositivo LVM RAID for uma falha transitória ou se você for capaz de reparar o dispositivo que falhou, você pode iniciar a recuperação do dispositivo que falhou.

Pré-requisitos

- O dispositivo anteriormente falhado está agora funcionando.

Procedimento

- Atualize o volume lógico que contém o dispositivo RAID:

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

Etapas de verificação

- Examine o volume lógico com o dispositivo recuperado:

```
# lvs -- todas --opções nome,dispositivos,lv_attr,lv_health_status my_vg
```

19.6.4. Substituição de um dispositivo RAID falho em um volume lógico

Este procedimento substitui um dispositivo falho que serve como volume físico em um volume lógico LVM RAID.

Pré-requisitos

- O grupo de volume inclui um volume físico que fornece capacidade livre suficiente para substituir o dispositivo avariado.
Se nenhum volume físico com extensões livres suficientes estiver disponível no grupo de volume, adicione um novo volume físico suficientemente grande usando o utilitário **vgextend**.

Procedimento

1. No exemplo a seguir, um volume lógico RAID é apresentado como se segue:

```
# 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. Se o dispositivo **/dev/sdc** falhar, a saída do comando **lvs** é a seguinte:

```
# 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. Substitua o dispositivo que falhou e exiba o volume 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 o volume físico que substitui o dispositivo avariado, adicionar o volume físico no final do comando:

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

4. Examine o volume lógico com o substituto:

```
# 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)
```

Até que você remova o dispositivo defeituoso do grupo de volume, as utilidades da LVM ainda indicam que a LVM não consegue encontrar o dispositivo defeituoso.

5. Remover o dispositivo com defeito do grupo de volume:

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

19.7. SOLUÇÃO DE PROBLEMAS COM EXTENSÕES LIVRES INSUFICIENTES PARA UM VOLUME LÓGICO

Você pode receber a mensagem de erro **Insufficient free extents** ao tentar criar um volume lógico, mesmo quando você acha que o grupo de volume tem espaço livre suficiente. Você pode solucionar este erro para poder criar um volume lógico no grupo de volume.

19.7.1. Grupos de volume

Os volumes físicos são combinados em grupos de volume (VGs). Isto cria um pool de espaço em disco a partir do qual os volumes lógicos podem ser alocados.

Dentro de um grupo de volume, o espaço em disco disponível para alocação é dividido em unidades de tamanho fixo chamadas extensões. Uma extensão é a menor unidade de espaço que pode ser alocada. Dentro de um volume físico, as extensões são chamadas de extensões físicas.

Um volume lógico é alocado em extensões lógicas do mesmo tamanho que as extensões físicas. O tamanho da extensão é, portanto, o mesmo para todos os volumes lógicos do grupo de volume. O grupo de volume mapeia as extensões lógicas em extensões físicas.

19.7.2. Erros de arredondamento na saída da LVM

Os comandos LVM que informam o uso do espaço em grupos de volume arredondam o número informado a 2 casas decimais para fornecer uma saída legível para humanos. Isto inclui os utilitários **vgdisplay** e **vgs**.

Como resultado do arredondamento, o valor relatado de espaço livre pode ser maior do que o que as extensões físicas no grupo de volume proporcionam. Se você tentar criar um volume lógico do tamanho do espaço livre relatado, você pode obter o seguinte erro:

Extensões livres insuficientes

Para contornar o erro, você deve examinar o número de extensões físicas livres no grupo de volume, que é o valor exato do espaço livre. Você pode então usar o número de extensões para criar o volume lógico com sucesso.

19.7.3. Prevenir o erro de arredondamento ao criar um volume LVM

Ao criar um volume lógico LVM, você pode especificar o tamanho do volume lógico para que não ocorra nenhum erro de arredondamento.

Procedimento

1. Encontre o número de extensões físicas livres no grupo de volume:

```
# vgdisplay volume-group-name
```

Exemplo 19.7. Extensões livres em um grupo de volume

Por exemplo, o seguinte grupo de volume tem 8780 extensões físicas livres:

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

2. Criar o volume lógico. Digite o tamanho do volume em extensões em vez de bytes.

Exemplo 19.8. Criação de um volume lógico, especificando o número de extensões

```
# lvcreate --extents 8780 --nome testlv testvg
```

Exemplo 19.9. Criação de um volume lógico para ocupar todo o espaço restante

Alternativamente, você pode estender o volume lógico para usar uma porcentagem do espaço livre restante no grupo de volume. Por exemplo:

```
# lvcreate --extents 100EE --nome testlv2 testvg
```

Etapas de verificação

- Verifique o número de extensões que o grupo de volume agora usa:

```
# vgs --options +vg_free_count,vg_extents_count

VG   #PV #LV #SN Attr   VSize  VFree Free #Ext
testvg 2  1  0 wz--n- 34.30G  0    0 8780
```

19.8. SOLUÇÃO DE PROBLEMAS DE AVISOS DE VOLUME FÍSICO DUPLICADO PARA DISPOSITIVOS LVM MULTIPATHED

Ao usar LVM com armazenamento multipath, comandos LVM que listam um grupo de volume ou volume lógico podem exibir mensagens como as seguintes:

```
Found duplicate PV GDjTZf7Y03GJHjteqOwrye2dcSCjdaUi: using /dev/dm-5 not /dev/sdd
Found duplicate PV GDjTZf7Y03GJHjteqOwrye2dcSCjdaUi: using /dev/emcpowerb not /dev/sde
Found duplicate PV GDjTZf7Y03GJHjteqOwrye2dcSCjdaUi: using /dev/sddlmap not /dev/sdf
```

Você pode solucionar estes avisos para entender por que a LVM os exibe, ou para esconder os avisos.

19.8.1. Causa raiz das advertências de PV duplicadas

Quando um software multipath como o Device Mapper Multipath (DM Multipath), EMC PowerPath ou Hitachi Dynamic Link Manager (HDLM) gerencia dispositivos de armazenamento no sistema, cada caminho para uma unidade lógica específica (LUN) é registrado como um dispositivo SCSI diferente. O software multipath cria então um novo dispositivo que mapeia para esses caminhos individuais. Como cada LUN tem múltiplos nós de dispositivos no diretório **/dev** que apontam para os mesmos dados subjacentes, todos os nós de dispositivos contêm os mesmos metadados do LVM.

Tabela 19.1. Exemplos de mapeamentos de dispositivos em diferentes softwares multipath

Software multipath	SCSI caminhos para um LUN	Mapeamento de dispositivos multicaminhos para caminhos
DM Multipath	/dev/sdb e /dev/sdc	/dev/mapper/mpath1 ou /dev/mapper/mpatha
EMC PowerPath		/dev/emcpowera
HDLM		/dev/sddlmap

Como resultado dos múltiplos nós de dispositivos, as ferramentas LVM encontram os mesmos metadados várias vezes e os relatam como duplicados.

19.8.2. Casos de avisos PV duplicados

A LVM exibe os avisos PV duplicados em um dos seguintes casos:

- Os dois dispositivos exibidos na saída são ambos caminhos individuais para o mesmo dispositivo.
- Os dois dispositivos exibidos na saída são ambos mapas multipath.

Caminhos individuais para o mesmo dispositivo

O exemplo a seguir mostra um aviso PV duplicado, no qual os dispositivos duplicados são ambos caminhos únicos para o mesmo dispositivo.

Encontrado duplicado de PV GDjTZf7Y03GJHjteqOwrye2dcSCjdaUi: usando /dev/sdd não /dev/sdf

Se você listar a topologia DM Multipath atual usando o comando **multipath -ll**, você pode encontrar tanto **/dev/sdd** como **/dev/sdf** sob o mesmo mapa multipath.

Estas mensagens duplicadas são apenas avisos e não significam que a operação da LVM falhou. Ao contrário, elas estão alertando que a LVM usa apenas um dos dispositivos como um volume físico e ignora os outros.

Se as mensagens indicarem que a LVM escolhe o dispositivo incorreto ou se os avisos forem perturbadores para os usuários, você pode aplicar um filtro. O filtro configura o LVM para procurar somente os dispositivos necessários para volumes físicos, e para deixar de fora quaisquer caminhos subjacentes a dispositivos multipath. Como resultado, as advertências não aparecem mais.

Mapas multicaminhos

Os exemplos a seguir mostram um aviso PV duplicado para dois dispositivos que são ambos mapas multipath. Os volumes físicos duplicados estão localizados em dois dispositivos diferentes e não em dois caminhos diferentes para o mesmo dispositivo.

Encontrado duplicado de PV GDjTZf7Y03GJHjteqOwrye2dcSCjdaUi: usando /dev/mapper/mpatha não /dev/mapper/mpathc

Encontrado duplicado de PV GDjTZf7Y03GJHjteqOwrye2dcSCjdaUi: usando /dev/emcpowera não /dev/emcpowerh

Esta situação é mais grave do que a duplicação de avisos para dispositivos que são ambos caminhos únicos para o mesmo dispositivo. Estes avisos muitas vezes significam que a máquina está acessando dispositivos que não deveria acessar: por exemplo, clones LUN ou espelhos.

A menos que você saiba claramente quais dispositivos você deve remover da máquina, esta situação pode ser irrecuperável. A Red Hat recomenda que você entre em contato com o Suporte Técnico da Red Hat para resolver este problema.

19.8.3. O filtro do dispositivo LVM

As ferramentas LVM procuram por dispositivos no diretório **/dev** e verificam todos os dispositivos lá em busca de metadados LVM. Um filtro no arquivo **/etc/lvm/lvm.conf** controla quais dispositivos LVM escaneia.

O filtro é uma lista de padrões que a LVM aplica a cada dispositivo encontrado por uma varredura do diretório **/dev**, ou o diretório especificado pela palavra-chave **dir** no arquivo **/etc/lvm/lvm.conf**. Os padrões são expressões regulares delimitadas por qualquer caracter e precedidas por **a** para *accept* ou **r**

para *reject*. A primeira expressão regular na lista que corresponde a um dispositivo determina se a LVM aceita ou rejeita (ignora) o dispositivo. A LVM aceita dispositivos que não correspondem a nenhum padrão.

A seguir está a configuração padrão do filtro, que escaneia todos os dispositivos:

```
filtro = [ \ a/.*/ ]
```

19.8.4. Exemplo de filtros de dispositivos LVM que impedem a duplicação de avisos PV

Os exemplos seguintes mostram filtros de dispositivos LVM que evitam a duplicação de avisos de volume físico que são causados por múltiplos caminhos de armazenamento para uma única unidade lógica (LUN).

O filtro que você configura deve incluir todos os dispositivos que o LVM precisa ser verificado quanto a metadados, como o disco rígido local com o grupo de volume raiz nele e quaisquer dispositivos multipathed. Ao rejeitar os caminhos subjacentes a um dispositivo multipath (como `/dev/sdb`, `/dev/sdd`, etc.), você pode evitar estes avisos de PV duplicados, porque o LVM encontra cada área de metadados única uma vez no próprio dispositivo multipath.

- Este filtro aceita a segunda partição no primeiro disco rígido e quaisquer dispositivos DM Multipath, mas rejeita todo o resto:

```
filtro = [ a|/dev/sda2$, "a|/dev/mapper/mpath.*", "r|.*/ ]
```

- Este filtro aceita todos os controladores HP SmartArray e quaisquer dispositivos EMC PowerPath:

```
filtro = [ a|/dev/cciss/.*/, "a|/dev/emcpower.*", "r|.*/ ]
```

- Este filtro aceita quaisquer partições no primeiro drive IDE e quaisquer dispositivos multipath:

```
filtro = [ a|/dev/hda.*", "a|/dev/mapper/mpath.*", "r|.*/ ]
```

19.8.5. Aplicação de uma configuração de filtro do dispositivo LVM

Este procedimento altera a configuração do filtro do dispositivo LVM, que controla os dispositivos que o LVM escaneia.

Pré-requisitos

- Prepare o padrão de filtro do dispositivo que você deseja usar.

Procedimento

1. Teste o padrão de filtragem de seu dispositivo sem modificar o arquivo `/etc/lvm/lvm.conf`. Use um comando LVM com o `--config 'devices{ filter = [your device filter pattern] }` opção. Por exemplo:

```
# lvs --configurar 'dispositivos (filtro = [ a|/dev/emcpower.*", "r|.*/ ]'
```

2. Edite a opção **filter** no arquivo de configuração **/etc/lvm/lvm.conf** para usar seu novo padrão de filtro de dispositivo.
3. Verifique se não estão faltando volumes físicos ou grupos de volume que você deseja utilizar com a nova configuração:

```
# pvscan
```

```
# vgscan
```

4. Reconstruir o sistema de arquivos **initramfs** para que a LVM escaneie somente os dispositivos necessários após a reinicialização:

```
# dracut --force --verbose
```

19.8.6. Recursos adicionais

- [Capítulo 15, Controle da varredura do dispositivo LVM](#)