



# Red Hat Enterprise Linux 8

## Gerenciamento de sistemas de arquivos

Criação, modificação e administração de sistemas de arquivo no Red Hat Enterprise Linux 8



# Red Hat Enterprise Linux 8 Gerenciamento de sistemas de arquivos

---

Criação, modificação e administração de sistemas de arquivo no Red Hat Enterprise Linux 8

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/Managing\_file\_systems.ent file | This material may only be distributed subject to the terms and conditions set forth in the GNU Free Documentation License (GFDL), V1.2 or later (the latest version is presently available at <http://www.gnu.org/licenses/fdl.txt>).

## Resumo

Esta coleção de documentação fornece instruções sobre como gerenciar efetivamente os sistemas de arquivo no Red Hat Enterprise Linux 8.

<b>TORNANDO O CÓDIGO ABERTO MAIS INCLUSIVO</b> .....	<b>9</b>
<b>FORNECENDO FEEDBACK SOBRE A DOCUMENTAÇÃO DA RED HAT</b> .....	<b>10</b>
<b>CAPÍTULO 1. VISÃO GERAL DOS SISTEMAS DE ARQUIVO DISPONÍVEIS</b> .....	<b>11</b>
1.1. TIPOS DE SISTEMAS DE ARQUIVO	11
1.2. SISTEMAS DE ARQUIVO LOCAIS	12
Sistemas de arquivo locais disponíveis	12
1.3. O SISTEMA DE ARQUIVO XFS	12
Características de desempenho	13
1.4. O SISTEMA DE ARQUIVO EXT4	14
1.5. COMPARAÇÃO DE XFS E EXT4	14
1.6. ESCOLHENDO UM SISTEMA DE ARQUIVO LOCAL	15
1.7. SISTEMAS DE ARQUIVOS EM REDE	16
Sistemas de arquivos em rede disponíveis	17
1.8. SISTEMAS DE ARQUIVOS DE ARMAZENAMENTO COMPARTILHADO	17
Comparação com sistemas de arquivos em rede	17
Concorrência	17
Características de desempenho	17
Sistemas de arquivos de armazenamento compartilhado disponíveis	18
1.9. ESCOLHENDO ENTRE SISTEMAS DE ARQUIVOS EM REDE E DE ARMAZENAMENTO COMPARTILHADO	18
1.10. SISTEMAS DE ARQUIVOS DE GERENCIAMENTO DE VOLUME	18
Sistemas de arquivos de gerenciamento de volume disponíveis	18
<b>CAPÍTULO 2. GERENCIAMENTO DO ARMAZENAMENTO LOCAL USANDO AS FUNÇÕES DO SISTEMA RHEL</b>	<b>20</b>
2.1. INTRODUÇÃO À FUNÇÃO DE ARMAZENAMENTO	20
2.2. PARÂMETROS QUE IDENTIFICAM UM DISPOSITIVO DE ARMAZENAMENTO NO PAPEL DO SISTEMA DE ARMAZENAMENTO	20
2.3. EXEMPLO LIVRO DE REPRODUÇÃO POSSÍVEL PARA CRIAR UM SISTEMA DE ARQUIVO XFS EM UM DISPOSITIVO DE BLOCO	21
2.4. EXEMPLO LIVRO DE REPRODUÇÃO POSSÍVEL PARA MONTAR PERSISTENTEMENTE UM SISTEMA DE ARQUIVO	22
2.5. EXEMPLO LIVRO DE EXERCÍCIOS POSSÍVEL PARA GERENCIAR VOLUMES LÓGICOS	22
2.6. EXEMPLO LIVRO DE REPRODUÇÃO POSSÍVEL PARA PERMITIR O DESCARTE EM BLOCO ONLINE	23
2.7. EXEMPLO LIVRO DE REPRODUÇÃO POSSÍVEL PARA CRIAR E MONTAR UM SISTEMA DE ARQUIVO EXT4	24
2.8. EXEMPLO LIVRO DE REPRODUÇÃO POSSÍVEL PARA CRIAR E MONTAR UM SISTEMA DE ARQUIVO EXT3	24
2.9. CONFIGURAÇÃO DE UM VOLUME RAID UTILIZANDO A FUNÇÃO DO SISTEMA DE ARMAZENAMENTO	25
2.10. CONFIGURAÇÃO DE UM POOL LVM COM RAID UTILIZANDO A FUNÇÃO DE SISTEMA DE ARMAZENAMENTO	26
2.11. CRIAÇÃO DE UM VOLUME CODIFICADO LUKS USANDO A FUNÇÃO DE ARMAZENAMENTO	28
<b>CAPÍTULO 3. MONTAGEM DE AÇÕES DA NFS</b> .....	<b>30</b>
3.1. INTRODUÇÃO AO NFS	30
3.2. VERSÕES NFS SUPORTADAS	30
Versão padrão da NFS	30
Características das versões menores do NFS	30
3.3. SERVIÇOS REQUERIDOS PELA NFS	31
Os serviços de RPC com NFSv4	32

3.4. FORMATOS DO NOME DO HOST NFS	32
3.5. INSTALANDO O NFS	33
3.6. DESCOBRINDO AS EXPORTAÇÕES DA NFS	33
3.7. MONTAGEM DE UM COMPARTILHAMENTO NFS COM MONTAGEM	34
3.8. OPÇÕES COMUNS DE MONTAGEM NFS	34
3.9. INFORMAÇÕES RELACIONADAS	36
<b>CAPÍTULO 4. EXPORTAÇÃO DE AÇÕES DA NFS</b>	<b>37</b>
4.1. INTRODUÇÃO AO NFS	37
4.2. VERSÕES NFS SUPORTADAS	37
Versão padrão da NFS	37
Características das versões menores do NFS	37
4.3. OS PROTOCOLOS TCP E UDP EM NFSV3 E NFSV4	38
4.4. SERVIÇOS REQUERIDOS PELA NFS	38
Os serviços de RPC com NFSv4	39
4.5. FORMATOS DO NOME DO HOST NFS	39
4.6. CONFIGURAÇÃO DO SERVIDOR NFS	40
4.6.1. O arquivo de configuração /etc/exporta	40
Entrada de exportação	40
Opções padrão	41
Opções padrão e anuladas	42
4.6.2. A utilidade das exportações	42
Opções comuns de exportação	42
4.7. NFS E RPCBIND	43
4.8. INSTALANDO O NFS	44
4.9. INICIANDO O SERVIDOR NFS	44
4.10. SOLUÇÃO DE PROBLEMAS DE NFS E RPCBIND	44
4.11. CONFIGURANDO O SERVIDOR NFS PARA RODAR ATRÁS DE UM FIREWALL	45
4.12. EXPORTANDO A COTA DE RPC ATRAVÉS DE UM FIREWALL	46
4.13. HABILITAÇÃO DO NFS SOBRE O RDMA (NFSORDMA)	47
4.14. CONFIGURAÇÃO DE UM SERVIDOR NFSV4 APENAS	48
4.14.1. Benefícios e desvantagens de um servidor NFSv4 somente para NFS	48
4.14.2. NFS e rpcbind	48
4.14.3. Configuração do servidor NFS para suportar apenas o NFSv4	49
4.14.4. Verificação da configuração apenas do NFSv4	49
4.15. INFORMAÇÕES RELACIONADAS	50
<b>CAPÍTULO 5. SEGURANÇA DO NFS</b>	<b>51</b>
5.1. SEGURANÇA NFS COM AUTH_SYS E CONTROLES DE EXPORTAÇÃO	51
5.2. SEGURANÇA NFS COM AUTH_GSS	51
5.3. CONFIGURAÇÃO DE UM SERVIDOR E CLIENTE NFS PARA USAR O KERBEROS	51
5.4. OPÇÕES DE SEGURANÇA NFSV4	52
5.5. PERMISSÕES DE ARQUIVO EM EXPORTAÇÕES NFS MONTADAS	53
<b>CAPÍTULO 6. HABILITANDO LAYOUTS PNFS SCSI EM NFS</b>	<b>54</b>
6.1. A TECNOLOGIA PNFS	54
6.2. LAYOUTS DO PNFS SCSI	54
Operações entre o cliente e o servidor	54
Reservas de dispositivos	55
6.3. VERIFICAÇÃO DE UM DISPOSITIVO SCSI COMPATÍVEL COM O PNFS	55
6.4. CONFIGURANDO O PNFS SCSI NO SERVIDOR	56
6.5. INSTALAÇÃO DO PNFS SCSI NO CLIENTE	56
6.6. LIBERAÇÃO DA RESERVA DO PNFS SCSI NO SERVIDOR	57
6.7. MONITORAMENTO DA FUNCIONALIDADE DOS LAYOUTS SCSI DO PNFS	58

6.7.1. Verificação das operações pNFS SCSI a partir do servidor usando o nfsstat	58
6.7.2. Verificação das operações pNFS SCSI por parte do cliente utilizando montarias	59
<b>CAPÍTULO 7. COMEÇANDO COM O FS-CACHE</b>	<b>60</b>
7.1. VISÃO GERAL DO FS-CACHE	60
7.2. GARANTIA DE DESEMPENHO	61
7.3. MONTANDO UM CACHE	61
7.4. USANDO O CACHE COM NFS	62
7.4.1. Configuração do compartilhamento de cache NFS	63
7.4.2. Limitações de cache com NFS	64
7.5. CONFIGURAÇÃO DOS LIMITES DE ABATE DE CACHES	65
7.6. OBTENÇÃO DE INFORMAÇÕES ESTATÍSTICAS DO MÓDULO FSCACHE KERNEL	65
7.7. REFERÊNCIAS DO FS-CACHE	66
<b>CAPÍTULO 8. MONTAGEM DE UMA AÇÃO SMB NO RED HAT ENTERPRISE LINUX</b>	<b>67</b>
8.1. VERSÕES DE PROTOCOLO SMB SUPOSTADAS	67
8.2. SUPORTE A EXTENSÕES UNIX	68
8.3. MONTAGEM MANUAL DE UMA AÇÃO SMB	68
8.4. MONTAGEM AUTOMÁTICA DE UM COMPARTILHAMENTO SMB QUANDO O SISTEMA INICIA	69
8.5. AUTENTICAÇÃO PARA UMA AÇÃO SMB USANDO UM ARQUIVO DE CREDENCIAIS	70
8.6. EXECUÇÃO DE UMA MONTAGEM SMB MULTIUSUÁRIO	70
8.6.1. Montagem de uma ação com a opção multiusuário	71
8.6.2. Verificar se uma ação SMB é montada com a opção multiusuário	71
8.6.3. Acesso a uma ação como um usuário	71
8.7. OPÇÕES DE MONTAGEM FREQUENTEMENTE UTILIZADAS	72
<b>CAPÍTULO 9. VISÃO GERAL DOS ATRIBUTOS DE NOMEAÇÃO PERSISTENTES</b>	<b>74</b>
9.1. DESVANTAGENS DOS ATRIBUTOS DE NOMENCLATURA NÃO-PERSISTENTES	74
9.2. SISTEMA DE ARQUIVOS E IDENTIFICADORES DE DISPOSITIVOS	75
Identificadores do sistema de arquivo	75
Identificadores de dispositivos	75
Recomendações	75
9.3. NOMES DE DISPOSITIVOS GERENCIADOS PELO MECANISMO UDEV EM /DEV/DISCO/	75
9.3.1. Identificadores do sistema de arquivo	76
O atributo UUID em /dev/disco/by-uuid/	76
O atributo Label in /dev/disk/by-label/	76
9.3.2. Identificadores de dispositivos	76
O atributo WWID em /dev/disco/by-id/	76
O atributo Partição UUID em /dev/disco/by-partuuid	77
O atributo Caminho em /dev/disco/by-path/	77
9.4. O IDENTIFICADOR MUNDIAL COM DM MULTIPATH	77
9.5. LIMITAÇÕES DA CONVENÇÃO DE NOMENCLATURA DE DISPOSITIVOS UDEV	78
9.6. LISTAGEM DE ATRIBUTOS DE NOMEAÇÃO PERSISTENTES	79
9.7. MODIFICAÇÃO DE ATRIBUTOS DE NOMEAÇÃO PERSISTENTES	80
<b>CAPÍTULO 10. COMEÇANDO COM AS DIVISÓRIAS</b>	<b>82</b>
10.1. VISUALIZAÇÃO DA MESA DIVISÓRIA	82
10.1.1. Visualizando a mesa divisória com separação	82
10.1.2. Exemplo de saída de parted print	82
10.2. CRIAÇÃO DE UMA TABELA DE PARTIÇÃO EM UM DISCO	83
10.2.1. Considerações antes de modificar as partições em um disco	83
O número máximo de divisórias	84
O tamanho máximo de uma divisória	84
Alinhamento de tamanhos	84

10.2.2. Comparação dos tipos de mesas divisórias	85
10.2.3. Partições de disco MBR	85
10.2.4. Partições MBR estendidas	86
10.2.5. Tipos de divisórias MBR	86
10.2.6. Tabela de partição do GUID	87
10.2.7. Criação de uma tabela de partição em um disco com partição	89
10.3. CRIANDO UMA DIVISÓRIA	90
10.3.1. Considerações antes de modificar as partições em um disco	90
O número máximo de divisórias	90
O tamanho máximo de uma divisória	90
Alinhamento de tamanhos	90
10.3.2. Tipos de partição	91
Tipos de partição ou bandeiras	91
Tipo de sistema de arquivo de partição	91
10.3.3. Esquema de nomenclatura das partições	92
10.3.4. Pontos de montagem e partições de disco	92
10.3.5. Criação de uma divisória com separação	93
10.3.6. Definição de um tipo de divisória com fdisk	94
10.4. REMOÇÃO DE UMA DIVISÓRIA	95
10.4.1. Considerações antes de modificar as partições em um disco	95
O número máximo de divisórias	96
O tamanho máximo de uma divisória	96
Alinhamento de tamanhos	96
10.4.2. Remoção de uma divisória com separação	97
10.5. REDIMENSIONAMENTO DE UMA DIVISÓRIA	98
10.5.1. Considerações antes de modificar as partições em um disco	98
O número máximo de divisórias	98
O tamanho máximo de uma divisória	99
Alinhamento de tamanhos	99
10.5.2. Redimensionamento de uma divisória com separação	99
10.6. ESTRATÉGIAS PARA REPARTICIONAR UM DISCO	101
10.6.1. Utilização de espaço livre não particionado	101
10.6.2. Usando o espaço de uma divisória não utilizada	101
10.6.3. Usando o espaço livre de uma divisória ativa	102
10.6.3.1. Repartição destrutiva	102
10.6.3.2. Repartição não-destrutiva	103
10.6.3.2.1. Compressão de dados existentes	103
10.6.3.2.2. Redimensionamento da divisória existente	104
10.6.3.2.3. Criação de novas divisórias	104
<b>CAPÍTULO 11. COMEÇANDO COM XFS</b>	<b>106</b>
11.1. O SISTEMA DE ARQUIVO XFS	106
Características de desempenho	107
11.2. CRIAÇÃO DE UM SISTEMA DE ARQUIVOS XFS	107
11.2.1. Criando um sistema de arquivo XFS com mkfs.xfs	107
11.2.2. Criação de um sistema de arquivo XFS em um dispositivo de bloco usando funções do sistema RHEL	108
11.2.2.1. Exemplo Livro de reprodução possível para criar um sistema de arquivo XFS em um dispositivo de bloco	108
11.2.2.2. Recursos adicionais	109
11.3. CÓPIA DE SEGURANÇA DE UM SISTEMA DE ARQUIVO XFS	109
11.3.1. Características do backup do XFS	109
11.3.2. Cópia de segurança de um sistema de arquivo XFS com xfsdump	110

11.3.3. Recursos adicionais	111
11.4. RESTAURANDO UM SISTEMA DE ARQUIVO XFS A PARTIR DE BACKUP	111
11.4.1. Características da restauração do XFS a partir do backup	111
11.4.2. Restaurando um sistema de arquivo XFS a partir de backup com xfsrestore	111
11.4.3. Mensagens informativas na restauração de um backup XFS a partir de uma fita	113
11.4.4. Recursos adicionais	113
11.5. AUMENTANDO O TAMANHO DE UM SISTEMA DE ARQUIVO XFS	113
11.5.1. Aumentando o tamanho de um sistema de arquivo XFS com xfs_growfs	113
11.6. COMPARAÇÃO DAS FERRAMENTAS UTILIZADAS COM EXT4 E XFS	114
<b>CAPÍTULO 12. CONFIGURAÇÃO DO COMPORTAMENTO DE ERRO DO XFS</b>	<b>116</b>
12.1. MANUSEIO DE ERROS CONFIGURÁVEL EM XFS	116
12.2. ARQUIVOS DE CONFIGURAÇÃO PARA CONDIÇÕES DE ERRO XFS ESPECÍFICAS E INDEFINIDAS	116
12.3. DEFINIÇÃO DO COMPORTAMENTO XFS PARA CONDIÇÕES ESPECÍFICAS	117
12.4. DEFININDO O COMPORTAMENTO DO XFS PARA CONDIÇÕES INDEFINIDAS	117
12.5. COMPORTAMENTO DE DESMONTAGEM DO XFS	118
<b>CAPÍTULO 13. VERIFICAÇÃO E REPARO DE UM SISTEMA DE ARQUIVO</b>	<b>119</b>
13.1. CENÁRIOS QUE REQUEREM UMA VERIFICAÇÃO DO SISTEMA DE ARQUIVO	119
13.2. POTENCIAIS EFEITOS COLATERAIS DO FSCK EM FUNCIONAMENTO	120
13.3. MECANISMOS DE TRATAMENTO DE ERROS EM XFS	120
Montagens não limpas	120
Corrupção	120
13.4. VERIFICAÇÃO DE UM SISTEMA DE ARQUIVO XFS COM XFS_REPAIR	121
13.5. CONserto DE UM SISTEMA DE ARQUIVO XFS COM XFS_REPAIR	122
13.6. MECANISMOS DE TRATAMENTO DE ERROS EM EXT2, EXT3, E EXT4	123
13.7. VERIFICAÇÃO DE UM SISTEMA DE ARQUIVO EXT2, EXT3, OU EXT4 COM E2FSCK	124
13.8. CONserto DE UM SISTEMA DE ARQUIVO EXT2, EXT3, OU EXT4 COM E2FSCK	124
<b>CAPÍTULO 14. MONTAGEM DE SISTEMAS DE ARQUIVO</b>	<b>126</b>
14.1. O MECANISMO DE MONTAGEM LINUX	126
14.2. LISTAGEM DOS SISTEMAS DE ARQUIVO ATUALMENTE MONTADOS	126
14.3. MONTAGEM DE UM SISTEMA DE ARQUIVO COM MONTAGEM	127
14.4. MOVENDO UM PONTO DE MONTAGEM	128
14.5. DESMONTANDO UM SISTEMA DE ARQUIVO COM UMount	128
14.6. OPÇÕES COMUNS DE MONTAGEM	129
14.7. COMPARTILHAR UMA MONTAGEM EM VÁRIOS PONTOS DE MONTAGEM	130
14.7.1. Tipos de montagens compartilhadas	130
14.7.2. Criação de um duplicado de ponto de montagem privado	131
14.7.3. Criação de um duplicado de ponto de montagem compartilhado	132
14.7.4. Criação de um duplicado de ponto de montagem de escravos	133
14.7.5. Impedindo que um ponto de montagem seja duplicado	134
14.7.6. Informações relacionadas	135
14.8. MONTAGEM PERSISTENTE DE SISTEMAS DE ARQUIVO	135
14.8.1. O arquivo /etc/fstab	135
14.8.2. Adicionando um sistema de arquivo ao /etc/fstab	136
14.8.3. Montagem persistente de um sistema de arquivo usando os papéis do sistema RHEL	137
14.8.3.1. Exemplo Livro de reprodução possível para montar persistentemente um sistema de arquivo	137
14.8.3.2. Recursos adicionais	138
14.9. MONTAGEM DE SISTEMAS DE ARQUIVOS SOB DEMANDA	138
14.9.1. O serviço autofs	138
14.9.2. Os arquivos de configuração autofs	138
O arquivo do mapa principal	139
Arquivos de mapas	139

O formato amd map	140
14.9.3. Configuração de pontos de montagem autofs	140
14.9.4. Montagem automática dos diretórios domésticos dos usuários do servidor NFS com o serviço autofs	141
14.9.5. Substituição ou aumento dos arquivos de configuração do site Autofs	142
14.9.6. Usando o LDAP para armazenar mapas de montadoras	143
14.10. DEFINIÇÃO DE PERMISSÕES SOMENTE LEITURA PARA O SISTEMA DE ARQUIVOS RAIZ	145
14.10.1. Arquivos e diretórios que sempre retêm permissões de escrita	145
14.10.2. Configuração do sistema de arquivo raiz para montagem com permissões somente leitura na inicialização	146
<b>CAPÍTULO 15. LIMITANDO O USO DO ESPAÇO DE ARMAZENAMENTO COM COTAS</b>	<b>148</b>
15.1. COTAS DE DISCOS	148
15.1.1. A ferramenta xfs_quota	148
Recursos adicionais	148
15.2. GERENCIANDO QUOTAS DE DISCOS XFS	148
15.2.1. Gestão de quotas do sistema de arquivos em XFS	149
15.2.2. Habilitação de quotas de disco para XFS	149
15.2.3. Relatório de uso de XFS	150
Pré-requisitos	150
Procedimento	150
Recursos adicionais	150
15.2.4. Modificando os limites de quotas XFS	150
Pré-requisitos	150
Procedimento	150
Recursos adicionais	151
15.2.5. Estabelecendo os limites do projeto para XFS	151
Procedimento	151
Recursos adicionais	152
15.3. GERENCIANDO QUOTAS DE DISCOS EXT3 E EXT4	152
15.3.1. Instalando a ferramenta de cota	152
15.3.2. Possibilitando o recurso de cota na criação do sistema de arquivos	152
15.3.3. Habilitação do recurso de cota nos sistemas de arquivo existentes	153
15.3.4. Possibilitando a aplicação de cotas	153
15.3.5. Atribuição de cotas por usuário	154
15.3.6. Atribuição de cotas por grupo	155
15.3.7. Atribuição de cotas por projeto	156
15.3.8. Estabelecendo o período de carência para limites suaves	157
15.3.9. Desligamento das quotas do sistema de arquivos	157
15.3.10. Relatórios sobre quotas de discos	158
<b>CAPÍTULO 16. DESCARTANDO BLOCOS NÃO UTILIZADOS</b>	<b>160</b>
16.1. OPERAÇÕES DE DESCARTE EM BLOCO	160
Requisitos	160
16.2. TIPOS DE OPERAÇÕES DE DESCARTE EM BLOCO	160
Recomendações	160
16.3. EXECUÇÃO DE DESCARTE DE BLOCOS EM LOTE	160
16.4. PERMITINDO O DESCARTE EM BLOCO ONLINE	161
16.5. PERMITINDO O DESCARTE EM BLOCO ONLINE USANDO AS FUNÇÕES DO SISTEMA RHEL	162
16.5.1. Exemplo Livro de reprodução possível para permitir o descarte em bloco online	162
16.5.2. Recursos adicionais	162
16.6. POSSIBILITANDO O DESCARTE PERIÓDICO EM BLOCO	162
<b>CAPÍTULO 17. GERENCIAMENTO DE ARMAZENAMENTO LOCAL EM CAMADAS COM O STRATIS</b>	<b>164</b>

17.1. MONTAGEM DE SISTEMAS DE ARQUIVO STRATIS	164
17.1.1. O propósito e as características de Stratis	164
17.1.2. Componentes de um volume Stratis	164
17.1.3. Dispositivos de bloqueio utilizáveis com Stratis	165
Dispositivos com suporte	165
Dispositivos sem suporte	166
17.1.4. Instalando o Stratis	166
17.1.5. Criação de um pool Stratis	166
17.1.6. Criação de um sistema de arquivo Stratis	168
17.1.7. Montagem de um sistema de arquivo Stratis	169
17.1.8. Montagem persistente de um sistema de arquivo Stratis	170
17.1.9. Informações relacionadas	171
17.2. AMPLIAÇÃO DE UM VOLUME STRATIS COM DISPOSITIVOS DE BLOQUEIO ADICIONAIS	171
17.2.1. Componentes de um volume Stratis	171
17.2.2. Adicionando dispositivos de blocos a uma piscina Stratis	172
17.2.3. Informações relacionadas	172
17.3. MONITORAMENTO DOS SISTEMAS DE ARQUIVO STRATIS	172
17.3.1. Tamanhos Stratis relatados por diferentes utilidades	172
17.3.2. Exibindo informações sobre os volumes do Stratis	173
17.3.3. Informações relacionadas	173
17.4. USANDO SNAPSHOTS EM SISTEMAS DE ARQUIVO STRATIS	173
17.4.1. Características dos snapshots de Stratis	174
17.4.2. Criando um instantâneo do Stratis	174
17.4.3. Acesso ao conteúdo de uma foto de Stratis	174
17.4.4. Revertendo um sistema de arquivo Stratis para um instantâneo anterior	175
17.4.5. Removendo uma foto do Stratis	176
17.4.6. Informações relacionadas	176
17.5. REMOÇÃO DOS SISTEMAS DE ARQUIVO STRATIS	176
17.5.1. Componentes de um volume Stratis	176
17.5.2. Remoção de um sistema de arquivo Stratis	177
17.5.3. Remoção de uma piscina Stratis	178
17.5.4. Informações relacionadas	178
<b>CAPÍTULO 18. COMEÇANDO COM UM SISTEMA DE ARQUIVO EXT3</b>	<b>180</b>
18.1. CARACTERÍSTICAS DE UM SISTEMA DE ARQUIVO EXT3	180
18.2. CRIAÇÃO DE UM SISTEMA DE ARQUIVO EXT3	180
18.3. MONTAGEM DE UM SISTEMA DE ARQUIVO EXT3	181
18.4. REDIMENSIONAMENTO DE UM SISTEMA DE ARQUIVO EXT3	182
18.5. CRIAÇÃO E MONTAGEM DE SISTEMAS DE ARQUIVO EXT3 USANDO FUNÇÕES DO SISTEMA RHEL	183
18.5.1. Exemplo Livro de reprodução possível para criar e montar um sistema de arquivo ext3	183
18.5.2. Recursos adicionais	184
<b>CAPÍTULO 19. COMEÇANDO COM UM SISTEMA DE ARQUIVO EXT4</b>	<b>185</b>
19.1. CARACTERÍSTICAS DE UM SISTEMA DE ARQUIVO EXT4	185
19.2. CRIAÇÃO DE UM SISTEMA DE ARQUIVO EXT4	185
19.3. MONTAGEM DE UM SISTEMA DE ARQUIVO EXT4	187
19.4. REDIMENSIONAMENTO DE UM SISTEMA DE ARQUIVO EXT4	187
19.5. CRIAÇÃO E MONTAGEM DE SISTEMAS DE ARQUIVO EXT4 USANDO FUNÇÕES DO SISTEMA RHEL	189
19.5.1. Exemplo Livro de reprodução possível para criar e montar um sistema de arquivo Ext4	189
19.6. COMPARAÇÃO DAS FERRAMENTAS UTILIZADAS COM EXT4 E XFS	190



## 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. VISÃO GERAL DOS SISTEMAS DE ARQUIVO DISPONÍVEIS

A escolha do sistema de arquivo apropriado para sua aplicação é uma decisão importante devido ao grande número de opções disponíveis e as contrapartidas envolvidas. Este capítulo descreve alguns dos sistemas de arquivo que são embarcados com o Red Hat Enterprise Linux 8 e fornece histórico e recomendações sobre o sistema de arquivo correto para se adequar à sua aplicação.

## 1.1. TIPOS DE SISTEMAS DE ARQUIVO

O Red Hat Enterprise Linux 8 suporta uma variedade de sistemas de arquivo (FS). Diferentes tipos de sistemas de arquivo resolvem diferentes tipos de problemas, e seu uso é específico da aplicação. No nível mais geral, os sistemas de arquivo disponíveis podem ser agrupados nos seguintes tipos principais:

**Tabela 1.1. Tipos de sistemas de arquivo e seus casos de uso**

Tipo	Sistema de arquivo	Atributos e casos de uso
Disco ou FS local	XFS	XFS é o sistema de arquivo padrão na RHEL. Como ele apresenta os arquivos como extensões, é menos vulnerável à fragmentação do que ext4. A Red Hat recomenda implantar o XFS como seu sistema de arquivo local, a menos que haja razões específicas para fazer o contrário: por exemplo, compatibilidade ou casos de canto em torno do desempenho.
	ext4	ext4 tem o benefício da longevidade no Linux. Portanto, ela é suportada por quase todas as aplicações Linux. Na maioria dos casos, ele rivaliza com o XFS no desempenho. ext4 é comumente usado para diretórios domésticos.
Rede ou cliente-e-servidor FS	NFS	Use NFS para compartilhar arquivos entre vários sistemas na mesma rede.
	SMB	Use SMB para compartilhamento de arquivos com sistemas Microsoft Windows.
Armazenamento compartilhado ou disco compartilhado FS	GFS2	O GFS2 fornece acesso compartilhado por escrito aos membros de um cluster de computação. A ênfase está na estabilidade e confiabilidade, com a experiência funcional de um sistema de arquivo local o mais possível. SAS Grid, Tibco MQ, IBM Websphere MQ, e Red Hat Active MQ foram implantados com sucesso no GFS2.

Tipo	Sistema de arquivo	Atributos e casos de uso
Gestão do volume FS	Stratis (Pré-visualização tecnológica)	Stratis é um gerente de volume construído sobre uma combinação de XFS e LVM. O objetivo do Stratis é emular as capacidades oferecidas pelos sistemas de arquivo de gerenciamento de volume como Btrfs e ZFS. É possível construir esta pilha manualmente, mas o Stratis reduz a complexidade da configuração, implementa as melhores práticas e consolida as informações de erro.

## 1.2. SISTEMAS DE ARQUIVO LOCAIS

Sistemas de arquivos locais são sistemas de arquivos que rodam em um único servidor local e estão diretamente ligados ao armazenamento.

Por exemplo, um sistema de arquivo local é a única opção para discos SATA ou SAS internos, e é usado quando seu servidor tem controladores RAID de hardware internos com unidades locais. Os sistemas de arquivo locais também são os sistemas de arquivo mais comuns usados no armazenamento anexado ao SAN quando o dispositivo exportado no SAN não é compartilhado.

Todos os sistemas de arquivo locais são compatíveis com POSIX e são totalmente compatíveis com todas as versões suportadas do Red Hat Enterprise Linux. Os sistemas de arquivo compatíveis com POSIX fornecem suporte a um conjunto bem definido de chamadas de sistema, tais como **read()**, **write()** e **seek()**.

Do ponto de vista do programador de aplicações, há relativamente poucas diferenças entre os sistemas de arquivo locais. As diferenças mais notáveis do ponto de vista do usuário estão relacionadas à escalabilidade e desempenho. Ao considerar a escolha de um sistema de arquivo, considere o tamanho que o sistema de arquivo precisa ter, que características únicas ele deve ter e como ele funciona sob sua carga de trabalho.

### Sistemas de arquivo locais disponíveis

- XFS
- ext4

## 1.3. O SISTEMA DE ARQUIVO XFS

O XFS é um sistema de arquivo de diário de 64 bits altamente escalável, de alto desempenho, robusto e maduro que suporta arquivos muito grandes e sistemas de arquivo em um único host. É o sistema de arquivo padrão no Red Hat Enterprise Linux 8. O XFS foi originalmente desenvolvido no início dos anos 90 pela SGI e tem um longo histórico de rodar em servidores e arrays de armazenamento extremamente grandes.

As características do XFS incluem:

### Confiabilidade

- Diário de Metadados, que garante a integridade do sistema de arquivo após uma falha do sistema, mantendo um registro das operações do sistema de arquivo que pode ser

reproduzido quando o sistema é reiniciado e o sistema de arquivo recontado

- Verificação extensiva da consistência dos metadados em tempo de execução
- Utilitários escaláveis e de reparo rápido
- Diário de cotas. Isto evita a necessidade de longas verificações de consistência de cotas após uma queda.

### Escalabilidade e desempenho

- Tamanho do sistema de arquivo suportado até 1024 TiB
- Capacidade de suportar um grande número de operações simultâneas
- Indexação B-tree para a escalabilidade da gestão do espaço livre
- Sofisticados algoritmos de leitura de metadados
- Otimizações para a carga de trabalho de streaming de vídeo

### Esquemas de alocação

- Alocação baseada na extensão
- Políticas de alocação de listras
- Atraso na alocação
- Pré-alocação de espaço
- Inódios alocados dinamicamente

### Outras características

- Cópias de arquivos com base no Refflink (novo no Red Hat Enterprise Linux 8)
- Utilitários de backup e restauração bem integrados
- Desfragmentação on-line
- Sistema de arquivo on-line crescendo
- Recursos abrangentes de diagnóstico
- Atributos estendidos (**xattr**). Isto permite que o sistema associe vários pares de nome/valor adicionais por arquivo.
- Cotas de projetos ou diretórios. Isto permite restrições de cotas sobre uma árvore de diretórios.
- Carimbos de tempo subseqüentes

### Características de desempenho

O XFS tem um alto desempenho em grandes sistemas com cargas de trabalho empresariais. Um sistema grande é aquele com um número relativamente alto de CPUs, múltiplos HBAs e conexões com matrizes de disco externas. O XFS também tem um bom desempenho em sistemas menores que têm uma carga

de trabalho de E/S paralela e multi-tarefa.

O XFS tem um desempenho relativamente baixo para cargas de trabalho com um único threaded e metadata intensivo: por exemplo, uma carga de trabalho que cria ou elimina um grande número de pequenos arquivos em um único thread.

## 1.4. O SISTEMA DE ARQUIVO EXT4

O sistema de arquivo ext4 é a quarta geração da família do sistema de arquivo ext4. Era o sistema de arquivo default do Red Hat Enterprise Linux 6.

O driver ext4 pode ler e escrever em sistemas de arquivo ext2 e ext3, mas o formato do sistema de arquivo ext4 não é compatível com os drivers ext2 e ext3.

ext4 acrescenta várias características novas e melhoradas, como por exemplo:

- Sistema de arquivo com suporte de até 50 TiB
- Metainformação baseada em extensão
- Atraso na alocação
- A soma de controle do diário
- Grande suporte de armazenamento

Os metadados baseados na extensão e os recursos de alocação atrasada proporcionam uma forma mais compacta e eficiente de rastrear o espaço utilizado em um sistema de arquivo. Estes recursos melhoram o desempenho do sistema de arquivos e reduzem o espaço consumido pelos metadados. A alocação atrasada permite que o sistema de arquivo adie a seleção do local permanente para os dados de usuários recém-escritos até que os dados sejam enviados para o disco. Isto permite um desempenho maior, pois pode permitir alocações maiores e mais contíguas, permitindo que o sistema de arquivo tome decisões com informações muito melhores.

O tempo de reparo do sistema de arquivos usando o utilitário **fsck** em ext4 é muito mais rápido do que em ext2 e ext3. Alguns reparos do sistema de arquivo têm demonstrado um aumento de até seis vezes no desempenho.

## 1.5. COMPARAÇÃO DE XFS E EXT4

XFS é o sistema de arquivo padrão na RHEL. Esta seção compara o uso e as características do XFS e ext4.

### Comportamento de erro dos metadados

No ext4, você pode configurar o comportamento quando o sistema de arquivos encontra erros de metadados. O comportamento padrão é simplesmente continuar a operação. Quando o XFS encontra um erro irreversível de metadados, ele desliga o sistema de arquivos e retorna o erro **EFSCORRUPTED**.

### Cotas

No ext4, você pode ativar cotas ao criar o sistema de arquivo ou mais tarde em um sistema de arquivo existente. Você pode então configurar a aplicação de cotas usando uma opção de montagem.

As cotas XFS não são uma opção remountable. Você deve ativar as cotas na montagem inicial.

A execução do comando **quotacheck** em um sistema de arquivos XFS não tem efeito. A primeira vez que você ativa a contabilização de cotas, o XFS verifica as cotas automaticamente.

### Redimensionamento do sistema de arquivo

O XFS não tem utilidade para reduzir o tamanho de um sistema de arquivo. Você só pode aumentar o tamanho de um sistema de arquivo XFS. Em comparação, ext4 suporta tanto a ampliação quanto a redução do tamanho de um sistema de arquivo.

### Números de inodo

O sistema de arquivo ext4 não suporta mais de 232 inodes.

O XFS aloca inodes dinamicamente. Um sistema de arquivo XFS não pode ficar sem inodes enquanto houver espaço livre no sistema de arquivo.

Certas aplicações não podem lidar corretamente com números de inode maiores que 232 em um sistema de arquivo XFS. Estas aplicações podem causar a falha de chamadas stat de 32 bits com o valor de retorno **Eoverflow**. O número de inode excede 232 sob as seguintes condições:

- O sistema de arquivo é maior do que 1 TiB com nós de 256 bytes.
- O sistema de arquivo é maior que 2 TiB com 512 bytes de inodes.

Se sua aplicação falhar com grandes números de inode, monte o sistema de arquivos XFS com a opção **-o inode32** para impor números de inode abaixo de 232. Note que o uso do **inode32** não afeta os inodes que já estão alocados com números de 64 bits.



#### IMPORTANTE

Faça *not* use a opção **inode32** a menos que um ambiente específico o exija. A opção **inode32** muda o comportamento de alocação. Como consequência, o erro **ENOSPC** pode ocorrer se não houver espaço disponível para alocar inodes nos blocos de disco inferiores.

## 1.6. ESCOLHENDO UM SISTEMA DE ARQUIVO LOCAL

Para escolher um sistema de arquivo que atenda às suas exigências de aplicação, você precisa entender o sistema alvo no qual você vai implantar o sistema de arquivo. Você pode usar as seguintes perguntas para informar sua decisão:

- Você tem um servidor grande?
- Você tem grandes exigências de armazenamento ou tem uma unidade SATA local e lenta?
- Que tipo de carga de trabalho de E/S você espera que sua aplicação apresente?
- Quais são seus requisitos de rendimento e latência?
- Qual é a estabilidade de seu servidor e hardware de armazenamento?
- Qual é o tamanho típico de seus arquivos e conjunto de dados?
- Se o sistema falhar, quanto tempo de inatividade você pode sofrer?

Se tanto seu servidor quanto seu dispositivo de armazenamento forem grandes, o XFS é a melhor escolha. Mesmo com matrizes de armazenamento menores, o XFS funciona muito bem quando os tamanhos médios dos arquivos são grandes (por exemplo, centenas de megabytes em tamanho).

Se sua carga de trabalho existente tiver funcionado bem com ext4, ficar com ext4 deve proporcionar a você e suas aplicações um ambiente muito familiar.

O sistema de arquivo ext4 tende a ter melhor desempenho em sistemas que têm capacidade limitada de E/S. Ele tem melhor desempenho em largura de banda limitada (menos de 200MB/s) e até cerca de 1000 IOPS. Para qualquer coisa com maior capacidade, o XFS tende a ser mais rápido.

O XFS consome cerca do dobro da operação CPU por metadados em relação ao ext4, portanto, se você tiver uma carga de trabalho vinculada à CPU com pouca concorrência, então o ext4 será mais rápido. Em geral, ext4 é melhor se uma aplicação usa um único fio de leitura/gravação e arquivos pequenos, enquanto o XFS brilha quando uma aplicação usa vários fios de leitura/gravação e arquivos maiores.

Não se pode encolher um sistema de arquivos XFS. Se você precisa ser capaz de encolher o sistema de arquivo, considere o uso do ext4, que suporta o encolhimento off-line.

Em geral, a Red Hat recomenda que você use XFS a menos que você tenha um caso de uso específico para ext4. Você também deve medir o desempenho de sua aplicação específica em seu servidor e sistema de armazenamento alvo para ter certeza de que você escolheu o tipo apropriado de sistema de arquivo.

**Tabela 1.2. Resumo das recomendações do sistema de arquivo local**

Cenário	Sistema de arquivo recomendado
Nenhum caso de uso especial	XFS
Grande servidor	XFS
Grandes dispositivos de armazenamento	XFS
Arquivos grandes	XFS
E/S com múltiplas roscas	XFS
E/S com rosca única	ext4
Capacidade limitada de E/S (menos de 1000 IOPS)	ext4
Largura de banda limitada (menos de 200MB/s)	ext4
Carga de trabalho vinculada à CPU	ext4
Apoio ao encolhimento off-line	ext4

## 1.7. SISTEMAS DE ARQUIVOS EM REDE

Os sistemas de arquivos em rede, também chamados de sistemas de arquivos cliente/servidor,

permitem que os sistemas clientes acessem arquivos que estão armazenados em um servidor compartilhado. Isto torna possível para múltiplos usuários em múltiplos sistemas compartilhar arquivos e recursos de armazenamento.

Tais sistemas de arquivo são construídos a partir de um ou mais servidores que exportam um conjunto de sistemas de arquivo para um ou mais clientes. Os nós do cliente não têm acesso ao armazenamento em bloco subjacente, mas interagem com o armazenamento usando um protocolo que permite um melhor controle de acesso.

### Sistemas de arquivos em rede disponíveis

- O sistema de arquivo cliente/servidor mais comum para clientes RHEL é o sistema de arquivo NFS. A RHEL fornece tanto um componente servidor NFS para exportar um sistema de arquivo local através da rede quanto um cliente NFS para importar estes sistemas de arquivo.
- A RHEL também inclui um cliente CIFS que suporta os populares servidores de arquivos Microsoft SMB para interoperabilidade com Windows. O servidor Samba do userspace fornece aos clientes Windows um serviço Microsoft SMB a partir de um servidor RHEL.

## 1.8. SISTEMAS DE ARQUIVOS DE ARMAZENAMENTO COMPARTILHADO

Sistemas de arquivo de armazenamento compartilhado, às vezes chamados de sistemas de arquivo de cluster, dão a cada servidor do cluster acesso direto a um dispositivo de bloco compartilhado através de uma rede de área de armazenamento local (SAN).

### Comparação com sistemas de arquivos em rede

Como os sistemas de arquivo cliente/servidor, os sistemas de arquivo de armazenamento compartilhado funcionam em um conjunto de servidores que são todos membros de um cluster. Ao contrário do NFS, no entanto, nenhum servidor único fornece acesso a dados ou metadados a outros membros: cada membro do cluster tem acesso direto ao mesmo dispositivo de armazenamento (o *shared storage*), e todos os nós de membros do cluster acessam o mesmo conjunto de arquivos.

### Concorrência

A coerência do cache é fundamental em um sistema de arquivo agrupado para garantir a consistência e integridade dos dados. Deve haver uma única versão de todos os arquivos em um cluster visível a todos os nós dentro de um cluster. O sistema de arquivo deve impedir que os membros do cluster atualizem o mesmo bloco de armazenamento ao mesmo tempo e causem corrupção de dados. Para fazer isso, os sistemas de arquivos de armazenamento compartilhado usam um mecanismo de amplo bloqueio de cluster para arbitrar o acesso ao armazenamento como um mecanismo de controle de simultaneidade. Por exemplo, antes de criar um novo arquivo ou escrever em um arquivo que é aberto em vários servidores, o componente do sistema de arquivo no servidor deve obter o bloqueio correto.

A exigência dos sistemas de arquivos de cluster é fornecer um serviço altamente disponível como um servidor web Apache. Qualquer membro do cluster terá uma visão totalmente coerente dos dados armazenados em seu sistema de arquivos em disco compartilhado, e todas as atualizações serão arbitradas corretamente pelos mecanismos de travamento.

### Características de desempenho

Os sistemas de arquivo em disco compartilhado nem sempre funcionam tão bem quanto os sistemas de arquivo locais rodando no mesmo sistema, devido ao custo computacional da sobrecarga de travamento. Sistemas de arquivos em disco compartilhado funcionam bem com cargas de trabalho onde cada nó escreve quase exclusivamente em um conjunto particular de arquivos que não são compartilhados com outros nós ou onde um conjunto de arquivos é compartilhado de forma quase exclusivamente de leitura através de um conjunto de nós. Isto resulta em um mínimo de invalidação de cache entre nós e pode maximizar o desempenho.

A configuração de um sistema de arquivo em disco compartilhado é complexa, e ajustar uma aplicação para ter um bom desempenho em um sistema de arquivo em disco compartilhado pode ser um desafio.

### Sistemas de arquivos de armazenamento compartilhado disponíveis

- O Red Hat Enterprise Linux fornece o sistema de arquivo GFS2. O GFS2 vem bem integrado com o Red Hat Enterprise Linux High Availability Add-On e o Resilient Storage Add-On. O Red Hat Enterprise Linux suporta GFS2 em clusters que variam em tamanho de 2 a 16 nós.

## 1.9. ESCOLHENDO ENTRE SISTEMAS DE ARQUIVOS EM REDE E DE ARMAZENAMENTO COMPARTILHADO

Ao escolher entre sistemas de arquivos em rede e de armazenamento compartilhado, considere os seguintes pontos

- Os sistemas de arquivos de rede baseados em NFS são uma escolha extremamente comum e popular para ambientes que fornecem servidores NFS.
- Os sistemas de arquivos de rede podem ser implantados utilizando tecnologias de rede de muito alto desempenho como Infiniband ou 10 Gigabit Ethernet. Isto significa que você não deve recorrer a sistemas de arquivos de armazenamento compartilhado apenas para obter largura de banda bruta para seu armazenamento. Se a velocidade de acesso é de suma importância, então use NFS para exportar um sistema de arquivo local como o XFS.
- Sistemas de arquivos de armazenamento compartilhado não são fáceis de configurar ou manter, portanto, você deve implantá-los somente quando não puder fornecer sua disponibilidade necessária com sistemas de arquivos locais ou em rede.
- Um sistema de arquivo de armazenamento compartilhado em um ambiente agrupado ajuda a reduzir o tempo de parada, eliminando as etapas necessárias para a desmontagem e montagem que precisam ser feitas durante um típico cenário de falha envolvendo a relocação de um serviço de alta disponibilidade.

A Red Hat recomenda que você utilize sistemas de arquivo em rede, a menos que você tenha um caso de uso específico para sistemas de arquivo de armazenamento compartilhado. Use sistemas de arquivo de armazenamento compartilhado principalmente para implantações que precisam fornecer serviços de alta disponibilidade com tempo mínimo de inatividade e com requisitos rigorosos de nível de serviço.

## 1.10. SISTEMAS DE ARQUIVOS DE GERENCIAMENTO DE VOLUME

Os sistemas de arquivo de gerenciamento de volume integram toda a pilha de armazenamento para fins de simplicidade e otimização na pilha.

### Sistemas de arquivos de gerenciamento de volume disponíveis

- O Red Hat Enterprise Linux 8 fornece o gerenciador de volume Stratis como uma prévia de tecnologia. Stratis usa XFS para a camada de sistema de arquivo e o integra com LVM, Device Mapper, e outros componentes. Stratis foi lançado pela primeira vez no Red Hat Enterprise Linux 8.0. Ele foi concebido para preencher a lacuna criada quando a Red Hat depreciou o Btrfs. Stratis 1.0 é um gerenciador de volume intuitivo, baseado em linha de comando, que pode realizar operações significativas de gerenciamento de armazenamento enquanto esconde a complexidade do usuário:
  - Gerenciamento de volume

- Criação de pool
- Piscinas de armazenagem finas
- Instantâneos
- Cache de leitura automatizado

Stratis oferece características poderosas, mas atualmente carece de certas capacidades de outras ofertas com as quais poderia ser comparado, tais como Btrfs ou ZFS. Mais notavelmente, ele não suporta CRCs com autocura.

## 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 [Seção 2.4, “Exemplo Livro de reprodução 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. MONTAGEM DE AÇÕES DA NFS

Como administrador do sistema, você pode montar ações NFS remotas em seu sistema para acessar dados compartilhados.

### 3.1. INTRODUÇÃO AO NFS

Esta seção explica os conceitos básicos do serviço NFS.

Um Sistema de Arquivo em Rede (NFS) permite que hosts remotos montem sistemas de arquivo em rede e interajam com esses sistemas de arquivo como se fossem montados localmente. Isto permite a consolidação de recursos em servidores centralizados na rede.

O servidor NFS se refere ao arquivo de configuração **/etc/exports** para determinar se o cliente tem permissão para acessar qualquer sistema de arquivo exportado. Uma vez verificadas, todas as operações de arquivo e diretório estão disponíveis para o usuário.

### 3.2. VERSÕES NFS SUPORTADAS

Esta seção lista versões do NFS suportadas no Red Hat Enterprise Linux e suas características.

Atualmente, o Red Hat Enterprise Linux 8 suporta as seguintes versões principais do NFS:

- O NFS versão 3 (NFSv3) suporta escritas assíncronas seguras e é mais robusto no manuseio de erros do que o NFSv2 anterior; ele também suporta tamanhos de arquivos de 64 bits e offsets, permitindo aos clientes acessar mais de 2 GB de dados de arquivos.
- O NFS versão 4 (NFSv4) funciona através de firewalls e na Internet, não requer mais um serviço **rpcbind**, suporta Listas de Controle de Acesso (ACLs), e utiliza operações estaduais.

O NFS versão 2 (NFSv2) não é mais suportado pela Red Hat.

#### Versão padrão da NFS

A versão default do NFS no Red Hat Enterprise Linux 8 é 4.2. Clientes NFS tentam montar usando o NFSv4.2 por default, e voltam ao NFSv4.1 quando o servidor não suporta o NFSv4.2. A montagem posteriormente cai de volta para o NFSv4.0 e depois para o NFSv3.

#### Características das versões menores do NFS

A seguir estão as características do NFSv4.2 no Red Hat Enterprise Linux 8:

##### Cópia do lado do servidor

Permite que o cliente NFS copie dados com eficiência sem desperdiçar recursos da rede usando a chamada do sistema **copy\_file\_range()**.

##### Arquivos esparsos

Permite que os arquivos tenham um ou mais *holes*, que são blocos de dados não alocados ou não inicializados, consistindo apenas em zeros. A operação **lseek()** no NFSv4.2 suporta **seek\_hole()** e **seek\_data()**, o que permite às aplicações mapear a localização de furos no arquivo esparsos.

##### Reserva de espaço

Permite que os servidores de armazenamento reservem espaço livre, o que proíbe que os servidores fiquem sem espaço. O NFSv4.2 suporta a operação **allocate()** para reservar espaço, a operação **deallocate()** para espaço sem reserva e a operação **fallocate()** para pré-alocar ou desalocar espaço em um arquivo.

##### Rotulado NFS

Impõe direitos de acesso aos dados e permite etiquetas SELinux entre um cliente e um servidor para arquivos individuais em um sistema de arquivos NFS.

### Melhorias de layout

Fornece a operação **layoutstats()**, que permite que alguns servidores Parallel NFS (pNFS) colem estatísticas de melhor desempenho.

A seguir estão as características do NFSv4.1:

- Aumenta o desempenho e a segurança da rede, e também inclui suporte do lado do cliente para o pNFS.
- Não é mais necessária uma conexão TCP separada para callbacks, o que permite que um servidor NFS conceda delegações mesmo quando não pode contatar o cliente: por exemplo, quando NAT ou um firewall interfere.
- Fornece exatamente uma vez a semântica (exceto para operações de reinício), evitando um problema anterior pelo qual certas operações às vezes retornavam um resultado impreciso se uma resposta fosse perdida e a operação fosse enviada duas vezes.

## 3.3. SERVIÇOS REQUERIDOS PELA NFS

Esta seção lista os serviços de sistema que são necessários para executar um servidor NFS ou montar ações NFS. O Red Hat Enterprise Linux inicia estes serviços automaticamente.

O Red Hat Enterprise Linux usa uma combinação de suporte em nível de kernel e processos de serviço para fornecer compartilhamento de arquivos NFS. Todas as versões NFS dependem de Remote Procedure Calls (RPC) entre clientes e servidores. Para compartilhar ou montar sistemas de arquivo NFS, os seguintes serviços trabalham em conjunto, dependendo de qual versão do NFS é implementada:

### **nfsd**

O módulo de kernel do servidor NFS que atende solicitações de sistemas de arquivos NFS compartilhados.

### **rpcbind**

Aceita reservas portuárias dos serviços locais de RPC. Estes portos são então disponibilizados (ou anunciados) para que os serviços de RPCs remotos correspondentes possam acessá-los. O serviço **rpcbind** responde às solicitações de serviços de RPC e estabelece conexões com o serviço de RPC solicitado. Isto não é usado com o NFSv4.

### **rpc.mountd**

Este processo é usado por um servidor NFS para processar solicitações de clientes NFSv3 em **MOUNT**. Ele verifica se o compartilhamento NFS solicitado é atualmente exportado pelo servidor NFS, e se o cliente tem permissão para acessá-lo. Se a solicitação de montagem for permitida, o serviço **nfs-mountd** responde com um status de Sucesso e fornece o File-Handle para este compartilhamento NFS de volta para o cliente NFS.

### **rpc.nfsd**

Este processo permite a definição de versões e protocolos NFS explícitos que o servidor anuncia. Ele trabalha com o kernel Linux para atender às demandas dinâmicas dos clientes NFS, tais como fornecer threads de servidor cada vez que um cliente NFS se conecta. Este processo corresponde ao serviço **nfs-server**.

### **lockd**

Esta é uma linha de kernel que roda tanto em clientes quanto em servidores. Ele implementa o protocolo Network Lock Manager (NLM), que permite aos clientes NFSv3 bloquear arquivos no servidor. Ele é iniciado automaticamente sempre que o servidor NFS é executado e sempre que um

sistema de arquivos NFS é montado.

### **rpc.statd**

Este processo implementa o protocolo Network Status Monitor (NSM) RPC, que notifica os clientes NFS quando um servidor NFS é reiniciado sem ser derrubado graciosamente. O serviço **rpc-statd** é iniciado automaticamente pelo serviço **nfs-server**, e não requer configuração do usuário. Isto não é usado com o NFSv4.

### **rpc.rquotad**

Este processo fornece informações de cota de usuário para usuários remotos. O serviço **rpc-rquotad** é iniciado automaticamente pelo serviço **nfs-server** e não requer configuração do usuário.

### **rpc.idmapd**

Este processo fornece upcalls de cliente e servidor NFSv4, que mapeiam entre os nomes NFSv4 on-the-wire (strings sob a forma de **user@domain**) e UIDs e GIDs locais. Para que **idmapd** funcione com NFSv4, o arquivo **/etc/idmapd.conf** deve ser configurado. No mínimo, deve ser especificado o parâmetro **Domain**, que define o domínio de mapeamento do NFSv4. Se o domínio de mapeamento do NFSv4 for o mesmo que o nome de domínio DNS, este parâmetro pode ser ignorado. O cliente e o servidor devem concordar sobre o domínio de mapeamento do NFSv4 para que o mapeamento de ID funcione corretamente.

Somente o servidor NFSv4 usa **rpc.idmapd**, que é iniciado pelo serviço **nfs-idmapd**. O cliente NFSv4 usa o utilitário **nfsidmap** baseado no keyring, que é chamado pelo kernel on-demand para realizar o mapeamento de ID. Se houver um problema com **nfsidmap**, o cliente volta a usar **rpc.idmapd**.

## **Os serviços de RPC com NFSv4**

Os protocolos de montagem e travamento foram incorporados ao protocolo NFSv4. O servidor também ouve na conhecida porta TCP 2049. Como tal, o NFSv4 não precisa interagir com **rpcbind**, **lockd**, e **rpc-statd** serviços. O serviço **nfs-mountd** ainda é necessário no servidor NFS para configurar as exportações, mas não está envolvido em nenhuma operação de "over-the-wire".

### **Recursos adicionais**

- Para configurar um servidor somente NFSv4, que não requer **rpcbind**, ver [Seção 4.14](#), "Configuração de um servidor NFSv4 apenas".

## **3.4. FORMATOS DO NOME DO HOST NFS**

Esta seção descreve diferentes formatos que você pode usar para especificar um host ao montar ou exportar uma ação NFS.

Você pode especificar o anfitrião nos seguintes formatos:

### **Máquina única**

Qualquer uma das seguintes opções:

- Um nome de domínio totalmente qualificado (que pode ser resolvido pelo servidor)
- Nome do host (que pode ser resolvido pelo servidor)
- Um endereço IP.

### **Redes IP**

Qualquer um dos seguintes formatos é válido:

- ***a.b.c.d/z*** onde ***a.b.c.d*** é a rede e ***z*** é o número de bits na máscara de rede; por exemplo **192.168.0.0/24**.
- ***a.b.c.d/netmask*** onde ***a.b.c.d*** é a rede e ***netmask*** é a máscara de rede; por exemplo, **192.168.100.8/255.255.255.0**.

### Netgroups

O **@group-name** formato , onde **group-name** é o nome do NIS netgroup.

## 3.5. INSTALANDO O NFS

Este procedimento instala todos os pacotes necessários para montar ou exportar ações da NFS.

### Procedimento

- Instale o pacote **nfs-utils**:

```
# yum instalar nfs-utils
```

## 3.6. DESCOBRINDO AS EXPORTAÇÕES DA NFS

Este procedimento descobre quais sistemas de arquivo um determinado servidor NFSv3 ou NFSv4 exporta.

### Procedimento

- Com qualquer servidor que suporte NFSv3, use o utilitário **showmount**:

```
$ showmount --exports my-server
```

```
Export list for my-server
/exports/foo
/exports/bar
```

- Com qualquer servidor que suporte NFSv4, monte o diretório raiz e olhe ao redor:

```
# mount my-server:/ /mnt/
# ls /mnt/
```

```
exports
```

```
# ls /mnt/exports/
```

```
foo
bar
```

Em servidores que suportam tanto NFSv4 quanto NFSv3, ambos os métodos funcionam e dão os mesmos resultados.

### Recursos adicionais

- A página do homem **showmount(8)**.

## 3.7. MONTAGEM DE UM COMPARTILHAMENTO NFS COM MONTAGEM

Este procedimento monta uma parte NFS exportada de um servidor usando o utilitário **mount**.

### Procedimento

- Para montar um compartilhamento NFS, use o seguinte comando:

```
# montagem -t nfs -o options host:/remote/export /local/directory
```

Este comando utiliza as seguintes variáveis:

#### *options*

Uma lista delimitada por vírgulas de opções de montagem.

#### *host*

O nome do host, endereço IP, ou nome de domínio totalmente qualificado do servidor que exporta o sistema de arquivos que você deseja montar.

#### */remote/export*

O sistema de arquivo ou diretório sendo exportado do servidor, ou seja, o diretório que você deseja montar.

#### */local/directory*

O local do cliente onde */remote/export* é montado.

### Recursos adicionais

- [Seção 3.8, "Opções comuns de montagem NFS"](#)
- [Seção 3.4, "Formatos do nome do host NFS"](#)
- [Seção 14.3, "Montagem de um sistema de arquivo com montagem"](#)
- A página do homem **mount(8)**

## 3.8. OPÇÕES COMUNS DE MONTAGEM NFS

Esta seção lista as opções comumente usadas na montagem de ações NFS. Estas opções podem ser usadas com comandos de montagem manual, configurações **/etc/fstab**, e **autofs**.

### **lookupcache=mode**

Especifica como o kernel deve gerenciar seu cache de entradas de diretório para um determinado ponto de montagem. Argumentos válidos para *mode* são **all**, **none**, ou **positive**.

### **nfsvers=version**

Especifica qual versão do protocolo NFS deve ser usada, onde *version* é **3**, **4**, **4.0**, **4.1**, ou **4.2**. Isto é útil para hosts que rodam vários servidores NFS, ou para desativar a re-instalação de uma montagem com versões inferiores. Se nenhuma versão for especificada, o NFS usa a versão mais alta suportada pelo kernel e o utilitário **mount**.

A opção **vers** é idêntica a **nfsvers**, e está incluída neste release por razões de compatibilidade.

### **noacl**

Desliga todo o processamento de ACL. Isto pode ser necessário ao fazer interface com versões mais antigas do Red Hat Enterprise Linux, Red Hat Linux ou Solaris, porque a tecnologia ACL mais recente não é compatível com sistemas mais antigos.

### **nolock**

Desativa o bloqueio de arquivos. Esta configuração é às vezes necessária quando se conecta a servidores NFS muito antigos.

### **noexec**

Impede a execução de binários em sistemas de arquivos montados. Isto é útil se o sistema estiver montando um sistema de arquivo não-Linux contendo binários incompatíveis.

### **nosuid**

Desativa os bits **set-user-identifier** e **set-group-identifier**. Isto impede que usuários remotos ganhem privilégios mais altos ao executar um programa **setuid**.

### **port=num**

Especifica o valor numérico da porta do servidor NFS. Se *num* é **0** (o valor padrão), então **mount** consulta o serviço **rpcbind** no host remoto para que o número da porta seja utilizado. Se o serviço NFS no host remoto não estiver registrado com seu serviço **rpcbind**, o número padrão da porta NFS do TCP 2049 é usado em seu lugar.

### **rsize=num e wsize=num**

Estas opções definem o número máximo de bytes a serem transferidos em uma única operação de leitura ou escrita NFS.

Não há valor padrão fixo para **rsize** e **wsize**. Por padrão, o NFS usa o maior valor possível tanto para o servidor quanto para o suporte ao cliente. No Red Hat Enterprise Linux 8, o cliente e o servidor têm um máximo de 1.048.576 bytes. Para mais detalhes, veja a seção [Quais são os valores default e máximo para rsize e wsize com montagens do NFS?](#) Artigo do KBase.

### **sec=flavors**

Sabores de segurança a serem usados para acessar arquivos na exportação montada. O *flavors* é uma lista separada por dois pontos de um ou mais sabores de segurança.

Por padrão, o cliente tenta encontrar um sabor de segurança que tanto o cliente quanto o servidor suportam. Se o servidor não suportar nenhum dos sabores selecionados, a operação de montagem falha.

Sabores disponíveis:

- **sec=sys** utiliza UNIX UIDs e GIDs locais. Estes usam **AUTH\_SYS** para autenticar as operações NFS.
- **sec=krb5** usa Kerberos V5 em vez de UIDs e GIDs UNIX locais para autenticar os usuários.
- **sec=krb5i** utiliza o Kerberos V5 para autenticação do usuário e realiza a verificação de integridade das operações NFS utilizando checksums seguros para evitar a adulteração de dados.
- **sec=krb5p** utiliza o Kerberos V5 para autenticação do usuário, verificação de integridade e criptografia do tráfego NFS para evitar o farejamento do tráfego. Esta é a configuração mais segura, mas também envolve a maior sobrecarga de desempenho.

### **tcp**

Instrui o suporte NFS a usar o protocolo TCP.

## **Recursos adicionais**

- A página do homem **mount(8)**
- A página do homem **nfs(5)**

### 3.9. INFORMAÇÕES RELACIONADAS

- O wiki Linux NFS: [https://linux-nfs.org/wiki/index.php/Main\\_Page](https://linux-nfs.org/wiki/index.php/Main_Page)
- Para montar ações da NFS de forma persistente, ver [Seção 14.8, “Montagem persistente de sistemas de arquivo”](#).
- Para montar ações da NFS sob demanda, ver [Seção 14.9, “Montagem de sistemas de arquivos sob demanda”](#).

## CAPÍTULO 4. EXPORTAÇÃO DE AÇÕES DA NFS

Como administrador do sistema, você pode usar o servidor NFS para compartilhar um diretório em seu sistema através da rede.

### 4.1. INTRODUÇÃO AO NFS

Esta seção explica os conceitos básicos do serviço NFS.

Um Sistema de Arquivo em Rede (NFS) permite que hosts remotos montem sistemas de arquivo em rede e interajam com esses sistemas de arquivo como se fossem montados localmente. Isto permite a consolidação de recursos em servidores centralizados na rede.

O servidor NFS se refere ao arquivo de configuração **/etc/exports** para determinar se o cliente tem permissão para acessar qualquer sistema de arquivo exportado. Uma vez verificadas, todas as operações de arquivo e diretório estão disponíveis para o usuário.

### 4.2. VERSÕES NFS SUPORTADAS

Esta seção lista versões do NFS suportadas no Red Hat Enterprise Linux e suas características.

Atualmente, o Red Hat Enterprise Linux 8 suporta as seguintes versões principais do NFS:

- O NFS versão 3 (NFSv3) suporta escritas assíncronas seguras e é mais robusto no manuseio de erros do que o NFSv2 anterior; ele também suporta tamanhos de arquivos de 64 bits e offsets, permitindo aos clientes acessar mais de 2 GB de dados de arquivos.
- O NFS versão 4 (NFSv4) funciona através de firewalls e na Internet, não requer mais um serviço **rpcbind**, suporta Listas de Controle de Acesso (ACLs), e utiliza operações estaduais.

O NFS versão 2 (NFSv2) não é mais suportado pela Red Hat.

#### Versão padrão da NFS

A versão default do NFS no Red Hat Enterprise Linux 8 é 4.2. Clientes NFS tentam montar usando o NFSv4.2 por default, e voltam ao NFSv4.1 quando o servidor não suporta o NFSv4.2. A montagem posteriormente cai de volta para o NFSv4.0 e depois para o NFSv3.

#### Características das versões menores do NFS

A seguir estão as características do NFSv4.2 no Red Hat Enterprise Linux 8:

##### Cópia do lado do servidor

Permite que o cliente NFS copie dados com eficiência sem desperdiçar recursos da rede usando a chamada do sistema **copy\_file\_range()**.

##### Arquivos esparsos

Permite que os arquivos tenham um ou mais *holes*, que são blocos de dados não alocados ou não inicializados, consistindo apenas em zeros. A operação **lseek()** no NFSv4.2 suporta **seek\_hole()** e **seek\_data()**, o que permite às aplicações mapear a localização de furos no arquivo esparsos.

##### Reserva de espaço

Permite que os servidores de armazenamento reservem espaço livre, o que proíbe que os servidores fiquem sem espaço. O NFSv4.2 suporta a operação **allocate()** para reservar espaço, a operação **deallocate()** para espaço sem reserva e a operação **fallocate()** para pré-alocar ou desalocar espaço em um arquivo.

##### Rotulado NFS

Impõe direitos de acesso aos dados e permite etiquetas SELinux entre um cliente e um servidor para arquivos individuais em um sistema de arquivos NFS.

### Melhorias de layout

Fornecer a operação **layoutstats()**, que permite que alguns servidores Parallel NFS (pNFS) colem estatísticas de melhor desempenho.

A seguir estão as características do NFSv4.1:

- Aumenta o desempenho e a segurança da rede, e também inclui suporte do lado do cliente para o pNFS.
- Não é mais necessária uma conexão TCP separada para callbacks, o que permite que um servidor NFS conceda delegações mesmo quando não pode contatar o cliente: por exemplo, quando NAT ou um firewall interfere.
- Fornece exatamente uma vez a semântica (exceto para operações de reinício), evitando um problema anterior pelo qual certas operações às vezes retornavam um resultado impreciso se uma resposta fosse perdida e a operação fosse enviada duas vezes.

## 4.3. OS PROTOCOLOS TCP E UDP EM NFSV3 E NFSV4

O NFSv4 requer o Protocolo de Controle de Transmissão (TCP) rodando sobre uma rede IP.

O NFSv3 também poderia usar o User Datagram Protocol (UDP) nas versões anteriores do Red Hat Enterprise Linux. No Red Hat Enterprise Linux 8, o NFS sobre o UDP não é mais suportado. Por default, o UDP é desativado no servidor NFS.

## 4.4. SERVIÇOS REQUERIDOS PELA NFS

Esta seção lista os serviços de sistema que são necessários para executar um servidor NFS ou montar ações NFS. O Red Hat Enterprise Linux inicia estes serviços automaticamente.

O Red Hat Enterprise Linux usa uma combinação de suporte em nível de kernel e processos de serviço para fornecer compartilhamento de arquivos NFS. Todas as versões NFS dependem de Remote Procedure Calls (RPC) entre clientes e servidores. Para compartilhar ou montar sistemas de arquivo NFS, os seguintes serviços trabalham em conjunto, dependendo de qual versão do NFS é implementada:

### **nfsd**

O módulo de kernel do servidor NFS que atende solicitações de sistemas de arquivos NFS compartilhados.

### **rpcbind**

Aceita reservas portuárias dos serviços locais de RPC. Estes portos são então disponibilizados (ou anunciados) para que os serviços de RPCs remotos correspondentes possam acessá-los. O serviço **rpcbind** responde às solicitações de serviços de RPC e estabelece conexões com o serviço de RPC solicitado. Isto não é usado com o NFSv4.

### **rpc.mountd**

Este processo é usado por um servidor NFS para processar solicitações de clientes NFSv3 em **MOUNT**. Ele verifica se o compartilhamento NFS solicitado é atualmente exportado pelo servidor NFS, e se o cliente tem permissão para acessá-lo. Se a solicitação de montagem for permitida, o serviço **nfs-mountd** responde com um status de Sucesso e fornece o File-Handle para este compartilhamento NFS de volta para o cliente NFS.

### **rpc.nfsd**

Este processo permite a definição de versões e protocolos NFS explícitos que o servidor anuncia. Ele trabalha com o kernel Linux para atender às demandas dinâmicas dos clientes NFS, tais como fornecer threads de servidor cada vez que um cliente NFS se conecta. Este processo corresponde ao serviço **nfs-server**.

### lockd

Esta é uma linha de kernel que roda tanto em clientes quanto em servidores. Ele implementa o protocolo Network Lock Manager (NLM), que permite aos clientes NFSv3 bloquear arquivos no servidor. Ele é iniciado automaticamente sempre que o servidor NFS é executado e sempre que um sistema de arquivos NFS é montado.

### rpc.statd

Este processo implementa o protocolo Network Status Monitor (NSM) RPC, que notifica os clientes NFS quando um servidor NFS é reiniciado sem ser derrubado graciosamente. O serviço **rpc-statd** é iniciado automaticamente pelo serviço **nfs-server**, e não requer configuração do usuário. Isto não é usado com o NFSv4.

### rpc.rquotad

Este processo fornece informações de cota de usuário para usuários remotos. O serviço **rpc-rquotad** é iniciado automaticamente pelo serviço **nfs-server** e não requer configuração do usuário.

### rpc.idmapd

Este processo fornece upcalls de cliente e servidor NFSv4, que mapeiam entre os nomes NFSv4 on-the-wire (strings sob a forma de **user@domain**) e UIDs e GIDs locais. Para que **idmapd** funcione com NFSv4, o arquivo **/etc/idmapd.conf** deve ser configurado. No mínimo, deve ser especificado o parâmetro **Domain**, que define o domínio de mapeamento do NFSv4. Se o domínio de mapeamento do NFSv4 for o mesmo que o nome de domínio DNS, este parâmetro pode ser ignorado. O cliente e o servidor devem concordar sobre o domínio de mapeamento do NFSv4 para que o mapeamento de ID funcione corretamente.

Somente o servidor NFSv4 usa **rpc.idmapd**, que é iniciado pelo serviço **nfs-idmapd**. O cliente NFSv4 usa o utilitário **nfsidmap** baseado no keyring, que é chamado pelo kernel on-demand para realizar o mapeamento de ID. Se houver um problema com **nfsidmap**, o cliente volta a usar **rpc.idmapd**.

## Os serviços de RPC com NFSv4

Os protocolos de montagem e travamento foram incorporados ao protocolo NFSv4. O servidor também ouve na conhecida porta TCP 2049. Como tal, o NFSv4 não precisa interagir com **rpcbind**, **lockd**, e **rpc-statd** serviços. O serviço **nfs-mountd** ainda é necessário no servidor NFS para configurar as exportações, mas não está envolvido em nenhuma operação de "over-the-wire".

### Recursos adicionais

- Para configurar um servidor somente NFSv4, que não requer **rpcbind**, ver [Seção 4.14](#), "Configuração de um servidor NFSv4 apenas".

## 4.5. FORMATOS DO NOME DO HOST NFS

Esta seção descreve diferentes formatos que você pode usar para especificar um host ao montar ou exportar uma ação NFS.

Você pode especificar o anfitrião nos seguintes formatos:

### Máquina única

Qualquer uma das seguintes opções:

- Um nome de domínio totalmente qualificado (que pode ser resolvido pelo servidor)

- Nome do host (que pode ser resolvido pelo servidor)
- Um endereço IP.

## Redes IP

Qualquer um dos seguintes formatos é válido:

- ***a.b.c.d/z*** onde ***a.b.c.d*** é a rede e ***z*** é o número de bits na máscara de rede; por exemplo **192.168.0.0/24**.
- ***a.b.c.d/netmask*** onde ***a.b.c.d*** é a rede e ***netmask*** é a máscara de rede; por exemplo, **192.168.100.8/255.255.255.0**.

## Netgroups

O **@*group-name*** formato, onde ***group-name*** é o nome do NIS netgroup.

## 4.6. CONFIGURAÇÃO DO SERVIDOR NFS

Esta seção descreve a sintaxe e as opções de duas maneiras de configurar as exportações em um servidor NFS:

- Edição manual do arquivo de configuração **/etc/exports**
- Usando o utilitário **exportfs** na linha de comando

### 4.6.1. O arquivo de configuração /etc/exporta

O arquivo **/etc/exports** controla quais sistemas de arquivo são exportados para hosts remotos e especifica opções. Ele segue as seguintes regras de sintaxe:

- As linhas em branco são ignoradas.
- Para acrescentar um comentário, inicie uma linha com a marca hash (**#**).
- Você pode enrolar longas linhas com uma contrabarra (**\**).
- Cada sistema de arquivo exportado deve estar em sua própria linha individual.
- Qualquer lista de hosts autorizados colocada após um sistema de arquivo exportado deve ser separada por caracteres de espaço.
- As opções para cada um dos anfitriões devem ser colocadas entre parênteses diretamente após o identificador do anfitrião, sem nenhum espaço que separe o anfitrião do primeiro parêntese.

### Entrada de exportação

Cada entrada para um sistema de arquivo exportado tem a seguinte estrutura:

```
export host(options)
```

Também é possível especificar múltiplos anfitriões, juntamente com opções específicas para cada anfitrião. Para isso, liste-os na mesma linha de uma lista delimitada por espaço, com cada nome de host seguido de suas respectivas opções (entre parênteses), como em:

```
export host1(options1) host2(options2) host3(options3)
```

Nesta estrutura:

### **export**

O diretório que está sendo exportado

### **host**

O anfitrião ou rede para a qual a exportação está sendo compartilhada

### **options**

As opções a serem utilizadas para hospedar

## Exemplo 4.1. Um simples arquivo `/etc/exportação`

Em sua forma mais simples, o arquivo `/etc/exports` especifica apenas o diretório exportado e os anfitriões autorizados a acessá-lo:

```
/exportado/directório bob.example.com
```

Aqui, **bob.example.com** pode montar `/exported/directory/` a partir do servidor NFS. Como nenhuma opção está especificada neste exemplo, o NFS usa opções padrão.

## IMPORTANTE

O formato do arquivo `/etc/exports` é muito preciso, particularmente no que diz respeito ao uso do caráter espacial. Lembre-se de sempre separar os sistemas de arquivo exportados dos sistemas hospedeiros e hospedeiros uns dos outros com um caractere de espaço. Entretanto, não deve haver outros caracteres de espaço no arquivo, exceto nas linhas de comentário.

Por exemplo, as duas linhas a seguir não significam a mesma coisa:

```
/home bob.example.com(rw)
/home bob.example.com (rw)
```

A primeira linha permite apenas aos usuários do site **bob.example.com** o acesso de leitura e escrita ao diretório `/home`. A segunda linha permite aos usuários de **bob.example.com** montar o diretório como somente leitura (o padrão), enquanto o resto do mundo pode montá-lo como leitura/escrita.

## Opções padrão

As opções padrão para uma entrada de exportação são:

### **ro**

O sistema de arquivo exportado é somente leitura. Os hosts remotos não podem alterar os dados compartilhados no sistema de arquivo. Para permitir que os hosts façam mudanças no sistema de arquivo (ou seja, leitura e escrita), especifique a opção `rw`.

### **sync**

O servidor NFS não responderá às solicitações antes que as alterações feitas por solicitações anteriores sejam gravadas em disco. Para habilitar as escritas assíncronas, em vez disso, especifique a opção **async**.

### **wdelay**

O servidor NFS irá atrasar a gravação no disco se suspeitar que outra solicitação de gravação está iminente. Isto pode melhorar o desempenho, pois reduz o número de vezes que o disco deve ser acessado por comandos de gravação separados, reduzindo assim a sobrecarga de gravação. Para desativar isto, especifique a opção **no\_wdelay**, que está disponível somente se a opção de sincronização padrão também for especificada.

### root\_squash

Isso impede que usuários root conectados remotamente (ao contrário de localmente) tenham privilégios de root; em vez disso, o servidor NFS atribui a eles o ID de usuário **nobody**. Isto efetivamente "esmaga" o poder do usuário root remoto para o usuário local mais baixo, impedindo possíveis escritas não autorizadas no servidor remoto. Para desabilitar o squashing da raiz, especifique a opção **no\_root\_squash**.

Para esmagar cada usuário remoto (incluindo a raiz), use a opção **all\_squash**. Para especificar os IDs de usuário e grupo que o servidor NFS deve atribuir aos usuários remotos de um determinado host, use as opções **anonuid** e **anongid**, respectivamente, como em:

```
export host(anonuid=uid,anongid=gid)
```

Aqui, *uid* e *gid* são o número de identificação do usuário e o número de identificação do grupo, respectivamente. As opções **anonuid** e **anongid** permitem a criação de uma conta especial de usuário e grupo para usuários de NFS remoto para compartilhar.

Por default, as listas de controle de acesso (ACLs) são suportadas pelo NFS sob o Red Hat Enterprise Linux. Para desabilitar este recurso, especifique a opção **no\_acl** ao exportar o sistema de arquivo.

### Opções padrão e anuladas

Cada padrão para cada sistema de arquivo exportado deve ser explicitamente anulado. Por exemplo, se a opção **rw** não for especificada, então o sistema de arquivo exportado é compartilhado como somente leitura. O seguinte é uma linha de exemplo de **/etc/exports** que anula duas opções padrão:

```
/another/exported/directory 192.168.0.3(rw,async)
```

Neste exemplo, **192.168.0.3** pode montar **/another/exported/directory/** ler e escrever, e todas as gravações em disco são assíncronas.

## 4.6.2. A utilidade das exportações

O utilitário **exportfs** permite ao usuário root exportar ou exportar diretórios seletivamente sem reiniciar o serviço NFS. Quando são dadas as opções adequadas, o utilitário **exportfs** escreve os sistemas de arquivos exportados para **/var/lib/nfs/xtab**. Como o serviço **nfs-mountd** refere-se ao arquivo **xtab** ao decidir privilégios de acesso a um sistema de arquivo, as mudanças na lista de sistemas de arquivo exportados entram em vigor imediatamente.

### Opções comuns de exportação

A seguir está uma lista das opções mais utilizadas disponíveis para **exportfs**:

**-r**

Faz com que todos os diretórios listados em **/etc/exports** sejam exportados através da construção de uma nova lista de exportação em **/etc/lib/nfs/xtab**. Esta opção efetivamente atualiza a lista de exportação com quaisquer alterações feitas em **/etc/exports**.

**-a**

Faz com que todos os diretórios sejam exportados ou não, dependendo de quais outras opções sejam passadas para **exportfs**. Se nenhuma outra opção for especificada, **exportfs** exporta todos os sistemas de arquivos especificados em **/etc/exports**.

#### **-o file-systems**

Especifica diretórios a serem exportados que não estão listados em **/etc/exports**. Substituir *file-systems* com sistemas de arquivo adicionais a serem exportados. Estes sistemas de arquivo devem ser formatados da mesma forma que estão especificados em **/etc/exports**. Esta opção é frequentemente usada para testar um sistema de arquivo exportado antes de adicioná-lo permanentemente à lista de sistemas de arquivo exportados.

#### **-i**

Ignora **/etc/exports**; apenas as opções dadas pela linha de comando são utilizadas para definir os sistemas de arquivo exportados.

#### **-u**

Desexporta todos os diretórios compartilhados. O comando **exportfs -ua** suspende o compartilhamento de arquivos NFS enquanto mantém todos os serviços NFS ativos. Para reativar o compartilhamento do NFS, use **exportfs -r**.

#### **-v**

Operação verbosa, onde os sistemas de arquivo sendo exportados ou não exportados são exibidos com mais detalhes quando o comando **exportfs** é executado.

Se nenhuma opção for passada para o utilitário **exportfs**, ele exibe uma lista dos sistemas de arquivos atualmente exportados.

#### Recursos adicionais

- Para informações sobre diferentes métodos para especificar nomes de anfitriões, ver [Seção 4.5, “Formatos do nome do host NFS”](#).
- Para uma lista completa de opções de exportação, consulte a página de manual **exports(5)**.
- Para mais informações sobre a utilidade **exportfs**, consulte a página de manual **exportfs(8)**.

## 4.7. NFS E RPCBIND

Esta seção explica o objetivo do serviço **rpcbind**, que é exigido pelo NFSv3.

O serviço **rpcbind** mapeia os serviços de Remote Procedure Call (RPC) para os portos nos quais eles escutam. Os processos de RPC notificam **rpcbind** quando iniciam, registrando as portas que estão escutando e os números dos programas de RPC que esperam servir. O sistema cliente então contata **rpcbind** no servidor com um determinado número de programa RPC. O serviço **rpcbind** redireciona o cliente para o número de porta apropriado para que ele possa se comunicar com o serviço solicitado.

Como os serviços baseados em RPC contam com **rpcbind** para fazer todas as conexões com pedidos de clientes entrantes, **rpcbind** deve estar disponível antes de qualquer um desses serviços começar.

As regras de controle de acesso para **rpcbind** afetam todos os serviços baseados em RPC. Alternativamente, é possível especificar regras de controle de acesso para cada um dos daemons RPC do NFS.

#### Recursos adicionais

- Para a sintaxe precisa das regras de controle de acesso, consulte as páginas de manual **rpc.mountd(8)** e **rpc.statd(8)**.

## 4.8. INSTALANDO O NFS

Este procedimento instala todos os pacotes necessários para montar ou exportar ações da NFS.

### Procedimento

- Instale o pacote **nfs-utils**:

```
# yum instalar nfs-utils
```

## 4.9. INICIANDO O SERVIDOR NFS

Este procedimento descreve como iniciar o servidor NFS, que é necessário para exportar ações NFS.

### Pré-requisitos

- Para servidores que suportam conexões NFSv2 ou NFSv3, o serviço **rpcbind** deve estar em execução. Para verificar se **rpcbind** está ativo, use o seguinte comando:

```
$ systemctl status rpcbind
```

Se o serviço for interrompido, inicie e habilite-o:

```
$ systemctl habilitado --agora rpcbind
```

### Procedimento

- Para iniciar o servidor NFS e habilitá-lo a iniciar automaticamente na inicialização, use o seguinte comando:

```
# systemctl enable --now nfs-server
```

### Recursos adicionais

- Para configurar um servidor somente NFSv4, que não requer **rpcbind**, ver [Seção 4.14, “Configuração de um servidor NFSv4 apenas”](#).

## 4.10. SOLUÇÃO DE PROBLEMAS DE NFS E RPCBIND

Como o serviço **rpcbind** oferece coordenação entre os serviços de RPC e os números de porta usados para se comunicar com eles, é útil visualizar o status dos serviços de RPC atuais usando **rpcbind** na resolução de problemas. O utilitário **rpcinfo** mostra cada serviço baseado em RPC com números de porta, um número de programa RPC, um número de versão e um tipo de protocolo IP (TCP ou UDP).

### Procedimento

1. Para ter certeza de que os serviços apropriados baseados no NFS RPC estão habilitados para **rpcbind**, use o seguinte comando:

```
# rpcinfo -p
```

### Exemplo 4.2. rpcinfo -p saída de comando

A seguir, uma amostra da saída deste comando:

```

program vers proto  port  service
100000  4  tcp  111  portmapper
100000  3  tcp  111  portmapper
100000  2  tcp  111  portmapper
100000  4  udp  111  portmapper
100000  3  udp  111  portmapper
100000  2  udp  111  portmapper
100005  1  udp  20048 mountd
100005  1  tcp  20048 mountd
100005  2  udp  20048 mountd
100005  2  tcp  20048 mountd
100005  3  udp  20048 mountd
100005  3  tcp  20048 mountd
100024  1  udp  37769 status
100024  1  tcp  49349 status
100003  3  tcp  2049  nfs
100003  4  tcp  2049  nfs
100227  3  tcp  2049  nfs_acl
100021  1  udp  56691 nlockmgr
100021  3  udp  56691 nlockmgr
100021  4  udp  56691 nlockmgr
100021  1  tcp  46193 nlockmgr
100021  3  tcp  46193 nlockmgr
100021  4  tcp  46193 nlockmgr

```

Se um dos serviços NFS não iniciar corretamente, **rpcbind** não poderá mapear as solicitações de RPC dos clientes para esse serviço até a porta correta.

2. Em muitos casos, se o NFS não estiver presente na saída de **rpcinfo**, reiniciar o NFS faz com que o serviço se registre corretamente em **rpcbind** e comece a funcionar:

```
# systemctl restart nfs-server
```

### Recursos adicionais

- Para mais informações e uma lista de opções de **rpcinfo**, consulte a página de manual **rpcinfo(8)**.
- Para configurar um servidor somente NFSv4, que não requer **rpcbind**, ver [Seção 4.14, "Configuração de um servidor NFSv4 apenas"](#).

## 4.11. CONFIGURANDO O SERVIDOR NFS PARA RODAR ATRÁS DE UM FIREWALL

NFS requer o serviço **rpcbind**, que atribui dinamicamente portas para serviços RPC e pode causar problemas para a configuração de regras de firewall. Este procedimento descreve como configurar o servidor NFS para funcionar por trás de um firewall.

## Procedimento

1. Para permitir que os clientes acessem compartilhamentos NFS atrás de um firewall, defina em quais portas os serviços RPC rodam na seção **[mountd]** do arquivo **/etc/nfs.conf**:

```
[mountd]
port=port-number
```

Isto acrescenta o **-p *port-number*** opção para a linha de comando **rpc.mount rpc.mount -p *port-number***.

2. Para permitir que os clientes acessem compartilhamentos NFS atrás de um firewall, configure o firewall executando os seguintes comandos no servidor NFS:

```
firewall-cmd --permanent --add-service mountd
firewall-cmd --permanent --add-service rpc-bind
firewall-cmd --permanent --add-service nfs
firewall-cmd --permanent --add-port=<mountd-port>/tcp
firewall-cmd --permanent --add-port=<mountd-port>/udp
firewall-cmd --reload
```

Nos comandos, substitua *<mountd-port>* pela porta pretendida ou um intervalo de portas. Ao especificar um intervalo de portas, use a sintaxe **--add-port=<*mountd-port*>-<*mountd-port*>/udp**.

3. Para permitir que NFSv4.0 callbacks passem por firewalls, configure **/proc/sys/fs/nfs/nfs\_callback\_tcpport** e permita que o servidor se conecte a essa porta no cliente.  
Esta etapa não é necessária para o NFSv4.1 ou superior, e os outros portos para **mountd**, **statd** e **lockd** não são necessários em um ambiente NFSv4 puro.
4. Para especificar as portas a serem utilizadas pelo serviço de RPC **nlockmgr**, defina o número da porta para as opções **nlm\_tcpport** e **nlm\_udpport** no arquivo **/etc/modprobe.d/lockd.conf**.
5. Reinicie o servidor NFS:

```
# systemctl restart nfs-server
```

Se a NFS não conseguir iniciar, verifique **/var/log/messages**. Geralmente, o NFS não inicia se você especificar um número de porta que já esteja em uso.

6. Confirmar que as mudanças entraram em vigor:

```
# rpcinfo -p
```

## Recursos adicionais

- Para configurar um servidor somente NFSv4, que não requer **rpcbind**, ver [Seção 4.14, "Configuração de um servidor NFSv4 apenas"](#).

## 4.12. EXPORTANDO A COTA DE RPC ATRAVÉS DE UM FIREWALL

Se você exportar um sistema de arquivo que usa cotas de disco, você pode usar o serviço de Chamada de Procedimento Remoto (RPC) de cotas para fornecer dados de cotas de disco para clientes NFS.

### Procedimento

1. Habilite e inicie o serviço **rpc-rquotad**:

```
# systemctl habilitado --agora rpc-rquotad
```



#### NOTA

O serviço **rpc-rquotad**, se ativado, é iniciado automaticamente após o início do serviço **nfs-server**.

2. Para tornar o serviço RPC de cota acessível atrás de um firewall, a porta TCP (ou UDP, se UDP estiver habilitada) 875 precisa estar aberta. O número da porta padrão é definido no arquivo **/etc/services**.

Você pode substituir o número da porta padrão anexando **-p port-number** à variável **RPCRQUOTADOPTS** no arquivo **/etc/sysconfig/rpc-rquotad**.

3. Por padrão, os anfitriões remotos só podem ler cotas. Se você quiser permitir que os clientes definam cotas, anexe a opção **-S** à variável **RPCRQUOTADOPTS** no arquivo **/etc/sysconfig/rpc-rquotad**.
4. Reinicie **rpc-rquotad** para que as mudanças no arquivo **/etc/sysconfig/rpc-rquotad** entrem em vigor:

```
# systemctl restart rpc-rquotad
```

## 4.13. HABILITAÇÃO DO NFS SOBRE O RDMA (NFSORDMA)

O serviço de acesso remoto direto à memória (RDMA) funciona automaticamente no Red Hat Enterprise Linux 8 se houver hardware com capacidade RDMA presente.

### Procedimento

1. Instale o pacote **rdma-core**:

```
# yum instalar rdma-core
```

2. Para permitir o carregamento automático dos módulos NFSoRDMA *server*, adicione a opção **SVCRDMA\_LOAD=yes** em uma nova linha no arquivo de configuração **/etc/rdma/rdma.conf**. A opção **rdma=20049** na seção **[nfsd]** do arquivo **/etc/nfs.conf** especifica o número da porta na qual o serviço NFSoRDMA ouve os clientes. A norma RFC 5667 especifica que os servidores devem escutar na porta **20049** quando prestam serviços NFSv4 sobre RDMA.

O arquivo **/etc/rdma/rdma.conf** contém uma linha que define por padrão a opção **XPRTRDMA\_LOAD=yes**, que solicita o serviço **rdma** para carregar o módulo NFSoRDMA *client*.

3. Reinicie o serviço **nfs-server**:

```
# systemctl restart nfs-server
```

## Recursos adicionais

- A norma RFC 5667: <https://tools.ietf.org/html/rfc5667>.

## 4.14. CONFIGURAÇÃO DE UM SERVIDOR NFSV4 APENAS

Como administrador do servidor NFS, você pode configurar o servidor NFS para suportar apenas o NFSv4, o que minimiza o número de portas abertas e serviços em execução no sistema.

### 4.14.1. Benefícios e desvantagens de um servidor NFSv4 somente para NFS

Esta seção explica os benefícios e desvantagens de configurar o servidor NFS para suportar apenas o NFSv4.

Por default, o servidor NFS suporta conexões NFSv3 e NFSv4 no Red Hat Enterprise Linux 8. No entanto, você também pode configurar o NFS para suportar apenas a versão 4.0 e posterior do NFS. Isto minimiza o número de portas abertas e serviços em execução no sistema, porque o NFSv4 não requer o serviço **rpcbind** para ouvir na rede.

Quando seu servidor NFS é configurado como NFSv4 somente, os clientes que tentam montar ações usando NFSv3 falham com um erro como o seguinte:

A versão solicitada do NFS ou protocolo de transporte não é suportada.

Opcionalmente, você também pode desativar a escuta das chamadas ao protocolo **RPCBIND**, **MOUNT** e **NSM**, que não são necessárias apenas no caso do NFSv4.

Os efeitos de desativar essas opções adicionais são:

- Os clientes que tentam montar ações a partir de seu servidor usando NFSv3 tornam-se insensíveis.
- O próprio servidor NFS é incapaz de montar sistemas de arquivo NFSv3.

### 4.14.2. NFS e rpcbind

Esta seção explica o objetivo do serviço **rpcbind**, que é exigido pelo NFSv3.

O serviço **rpcbind** mapeia os serviços de Remote Procedure Call (RPC) para os portos nos quais eles escutam. Os processos de RPC notificam **rpcbind** quando iniciam, registrando as portas que estão escutando e os números dos programas de RPC que esperam servir. O sistema cliente então contata **rpcbind** no servidor com um determinado número de programa RPC. O serviço **rpcbind** redireciona o cliente para o número de porta apropriado para que ele possa se comunicar com o serviço solicitado.

Como os serviços baseados em RPC contam com **rpcbind** para fazer todas as conexões com pedidos de clientes entrantes, **rpcbind** deve estar disponível antes de qualquer um desses serviços começar.

As regras de controle de acesso para **rpcbind** afetam todos os serviços baseados em RPC. Alternativamente, é possível especificar regras de controle de acesso para cada um dos daemons RPC do NFS.

## Recursos adicionais

- Para a sintaxe precisa das regras de controle de acesso, consulte as páginas de manual **rpc.mountd(8)** e **rpc.statd(8)**.

### 4.14.3. Configuração do servidor NFS para suportar apenas o NFSv4

Este procedimento descreve como configurar seu servidor NFS para suportar somente NFS versão 4.0 e posterior.

#### Procedimento

1. Desative o NFSv3 adicionando as seguintes linhas à seção **[nfsd]** do arquivo de configuração **/etc/nfs.conf**:

```
[nfsd]
vers3=no
```

2. Opcionalmente, desativar a escuta das chamadas ao protocolo **RPCBIND**, **MOUNT** e **NSM**, que não são necessárias apenas no caso do NFSv4. Desativar os serviços relacionados:

```
# máscara systemctl --now rpc-statd.service rpcbind.service rpcbind.socket
```

3. Reinicie o servidor NFS:

```
# systemctl restart nfs-server
```

As mudanças entram em vigor assim que você inicia ou reinicia o servidor NFS.

### 4.14.4. Verificação da configuração apenas do NFSv4

Este procedimento descreve como verificar se seu servidor NFS está configurado no modo somente NFSv4, usando o utilitário **netstat**.

#### Procedimento

- Use o utilitário **netstat** para listar os serviços de escuta nos protocolos TCP e UDP:

```
# netstat --listening --tcp --udp
```

#### Exemplo 4.3. Saída em um servidor NFSv4 apenas

O exemplo a seguir é uma saída **netstat** em um servidor somente NFSv4; ouvir para **RPCBIND**, **MOUNT** e **NSM** também é desativado. Aqui, **nfs** é o único serviço de escuta do NFS:

```
# netstat --listening --tcp --udp

Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp    0      0 0.0.0.0:ssh           0.0.0.0:*              LISTEN
tcp    0      0 0.0.0.0:nfs           0.0.0.0:*              LISTEN
tcp6   0      0 [::]:ssh             [::]:*                 LISTEN
tcp6   0      0 [::]:nfs             [::]:*                 LISTEN
udp    0      0 localhost.locald:bootpc 0.0.0.0:*
```

**Exemplo 4.4. Saída antes de configurar um servidor NFSv4 apenas**

Em comparação, a saída **netstat** antes de configurar um servidor somente NFSv4 inclui os serviços **sunrpc** e **mountd**:

```
# netstat --listening --tcp --udp

Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp    0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN
tcp    0      0 0.0.0.0:40189           0.0.0.0:*               LISTEN
tcp    0      0 0.0.0.0:46813           0.0.0.0:*               LISTEN
tcp    0      0 0.0.0.0:nfs             0.0.0.0:*               LISTEN
tcp    0      0 0.0.0.0:sunrpc          0.0.0.0:*               LISTEN
tcp    0      0 0.0.0.0:mountd          0.0.0.0:*               LISTEN
tcp6   0      0 [::]:ssh                [::]:*                  LISTEN
tcp6   0      0 [::]:51227               [::]:*                  LISTEN
tcp6   0      0 [::]:nfs                  [::]:*                  LISTEN
tcp6   0      0 [::]:sunrpc               [::]:*                  LISTEN
tcp6   0      0 [::]:mountd               [::]:*                  LISTEN
tcp6   0      0 [::]:45043                [::]:*                  LISTEN
udp    0      0 localhost:1018           0.0.0.0:*
udp    0      0 localhost.locald:bootpc 0.0.0.0:*
udp    0      0 0.0.0.0:mountd           0.0.0.0:*
udp    0      0 0.0.0.0:46672            0.0.0.0:*
udp    0      0 0.0.0.0:sunrpc           0.0.0.0:*
udp    0      0 0.0.0.0:33494            0.0.0.0:*
udp6   0      0 [::]:33734                [::]:*
udp6   0      0 [::]:mountd               [::]:*
udp6   0      0 [::]:sunrpc               [::]:*
udp6   0      0 [::]:40243                [::]:*
```

## 4.15. INFORMAÇÕES RELACIONADAS

- O wiki Linux NFS: <https://linux-nfs.org>

## CAPÍTULO 5. SEGURANÇA DO NFS

Para minimizar os riscos de segurança NFS e proteger os dados no servidor, considere as seguintes seções ao exportar sistemas de arquivos NFS em um servidor ou montá-los em um cliente.

### 5.1. SEGURANÇA NFS COM AUTH\_SYS E CONTROLES DE EXPORTAÇÃO

A NFS oferece as seguintes opções tradicionais para controlar o acesso aos arquivos exportados:

- O servidor restringe quais hosts estão autorizados a montar quais sistemas de arquivo por endereço IP ou pelo nome do host.
- O servidor aplica as permissões do sistema de arquivos para usuários em clientes NFS da mesma forma que o faz para usuários locais. Tradicionalmente, o NFS faz isto usando a mensagem de chamada **AUTH\_SYS** (também chamada **AUTH\_UNIX**), que depende do cliente para declarar a UID e GIDs do usuário. Esteja ciente de que isto significa que um cliente malicioso ou mal configurado pode facilmente errar isto e permitir que um usuário tenha acesso a arquivos que não deveria.

Para limitar os riscos potenciais, os administradores muitas vezes limitam o acesso a permissões de usuário somente leitura ou de esmagamento a um usuário comum e a uma identificação de grupo. Infelizmente, estas soluções impedem que o compartilhamento NFS seja utilizado da forma originalmente pretendida.

Além disso, se um atacante ganhar o controle do servidor DNS utilizado pelo sistema de exportação do sistema de arquivos NFS, ele pode apontar o sistema associado a um determinado nome de máquina ou nome de domínio totalmente qualificado para uma máquina não autorizada. Neste ponto, a máquina não autorizada *is* o sistema permite a montagem do compartilhamento NFS, porque nenhuma informação de nome de usuário ou senha é trocada para fornecer segurança adicional para a montagem do NFS.

Os curingas devem ser usados com parcimônia na exportação de diretórios através do NFS, pois é possível que o escopo do curinga englobe mais sistemas do que o pretendido.

#### Recursos adicionais

- Para garantir o NFS e **rpcbind**, utilize, por exemplo, **nftables** e **firewalld**. Para detalhes sobre a configuração dessas estruturas, consulte as páginas de manual **nft(8)** e **firewalld-cmd(1)**.

### 5.2. SEGURANÇA NFS COM AUTH\_GSS

Todas as versões do suporte NFS RPCSEC\_GSS e o mecanismo Kerberos.

Ao contrário do AUTH\_SYS, com o mecanismo RPCSEC\_GSS Kerberos, o servidor não depende do cliente para representar corretamente qual usuário está acessando o arquivo. Em vez disso, a criptografia é usada para autenticar usuários no servidor, o que impede que um cliente malicioso imite um usuário sem ter as credenciais Kerberos desse usuário. O uso do mecanismo Kerberos RPCSEC\_GSS é a maneira mais simples de garantir montagens porque, após configurar o Kerberos, não é necessária nenhuma configuração adicional.

### 5.3. CONFIGURAÇÃO DE UM SERVIDOR E CLIENTE NFS PARA USAR O KERBEROS

Kerberos é um sistema de autenticação de rede que permite que clientes e servidores se autenticuem uns aos outros usando criptografia simétrica e um terceiro de confiança, o KDC. A Red Hat recomenda o uso do Gerenciamento de Identidade (IdM) para a configuração do Kerberos.

### Pré-requisitos

- O Centro de Distribuição de Chaves Kerberos (**KDC**) está instalado e configurado.

### Procedimento

1.
  - Criar o **nfs/hostname.domain@REALM** principal no lado do servidor NFS.
  - Criar o **host/hostname.domain@REALM** principal tanto do lado do servidor como do lado do cliente.
  - Adicione as chaves correspondentes às fichas-chave para o cliente e o servidor.
2. No lado do servidor, use a opção **sec=** para habilitar os sabores de segurança desejados. Para habilitar todos os sabores de segurança, bem como as montagens não criptográficas:

```
/exportar *(seg=sys:krb5:krb5i:krb5p)
```

Os sabores de segurança válidos para usar com a opção **sec=** são:

- **sys**: sem proteção criptográfica, o padrão
  - **krb5**: somente autenticação
  - **krb5i**: proteção da integridade
  - **krb5p**: proteção de privacidade
3. No lado do cliente, adicionar **sec=krb5** (ou **sec=krb5i**, ou **sec=krb5p**, dependendo da configuração) às opções de montagem:

```
# mount -o sec=krb5 server:/exportar /mnt
```

### Recursos adicionais

- Se você precisar escrever arquivos como raiz no compartilhamento NFS protegido por Kerberos e manter a propriedade raiz nesses arquivos, veja <https://access.redhat.com/articles/4040141>. Note que esta configuração não é recomendada.
- Para mais informações sobre a configuração do NFS, consulte as páginas de manual **exports(5)** e **nfs(5)**.

## 5.4. OPÇÕES DE SEGURANÇA NFSV4

O NFSv4 inclui suporte ACL baseado no modelo Microsoft Windows NT, não no modelo POSIX, devido às características do modelo Microsoft Windows NT e sua ampla implementação.

Outra importante característica de segurança do NFSv4 é a remoção do uso do protocolo **MOUNT** para montagem de sistemas de arquivos. O protocolo **MOUNT** apresentou um risco de segurança devido à forma como o protocolo processava o arquivo.

## 5.5. PERMISSÕES DE ARQUIVO EM EXPORTAÇÕES NFS MONTADAS

Uma vez que o sistema de arquivo NFS é montado como leitura ou leitura e escrita por um host remoto, a única proteção que cada arquivo compartilhado tem é suas permissões. Se dois usuários que compartilham o mesmo valor de ID de usuário montarem o mesmo sistema de arquivo NFS em sistemas clientes diferentes, eles podem modificar os arquivos um do outro. Além disso, qualquer pessoa logada como **root** no sistema cliente pode usar o comando **su -** para acessar qualquer arquivo com o compartilhamento NFS.

Por default, as listas de controle de acesso (ACLs) são suportadas pelo NFS sob o Red Hat Enterprise Linux. A Red Hat recomenda manter este recurso ativado.

Por padrão, a NFS usa *root squashing* ao exportar um sistema de arquivo. Isto define o ID de usuário de qualquer pessoa que acesse o compartilhamento do NFS como usuário **root** em sua máquina local para **nobody**. O esmagamento de raízes é controlado pela opção padrão **root\_squash**; para mais informações sobre esta opção, veja [Seção 4.6, "Configuração do servidor NFS"](#).

Ao exportar uma ação da NFS como somente leitura, considere o uso da opção **all\_squash**. Esta opção faz com que cada usuário que acesse o sistema de arquivo exportado pegue o ID de usuário do usuário **nobody**.

## CAPÍTULO 6. HABILITANDO LAYOUTS PNFS SCSI EM NFS

É possível configurar o servidor e cliente NFS para usar o layout pNFS SCSI para acessar dados. pNFS SCSI é benéfico em casos de uso que envolvam acesso a um arquivo de um único cliente de duração mais longa.

### Pré-requisitos

- Tanto o cliente quanto o servidor devem ser capazes de enviar comandos SCSI para o mesmo dispositivo de bloco. Ou seja, o dispositivo de bloco deve estar em um barramento SCSI compartilhado.
- O dispositivo de bloco deve conter um sistema de arquivo XFS.
- O dispositivo SCSI deve suportar as Reservas Persistentes SCSI, conforme descrito na especificação dos Comandos Primários SCSI-3.

### 6.1. A TECNOLOGIA PNFS

A arquitetura pNFS melhora a escalabilidade do NFS. Quando um servidor implementa o pNFS, o cliente é capaz de acessar dados através de múltiplos servidores simultaneamente. Isto pode levar a melhorias de desempenho.

o pNFS suporta os seguintes protocolos ou layouts de armazenamento no RHEL:

- Arquivos
- Flexfiles
- SCSI

### 6.2. LAYOUTS DO PNFS SCSI

O layout SCSI se baseia no trabalho de layouts de blocos pNFS. O layout é definido através dos dispositivos SCSI. Ele contém uma série sequencial de blocos de tamanho fixo como unidades lógicas (LUs) que devem ser capazes de suportar reservas persistentes SCSI. Os dispositivos LU são identificados por sua identificação de dispositivos SCSI.

pNFS SCSI tem um bom desempenho em casos de uso que envolvem acesso de um único cliente a um arquivo com duração mais longa. Um exemplo pode ser um servidor de e-mail ou uma máquina virtual que abriga um cluster.

#### Operações entre o cliente e o servidor

Quando um cliente NFS lê de um arquivo ou escreve para ele, o cliente realiza uma operação em **LAYOUTGET**. O servidor responde com a localização do arquivo no dispositivo SCSI. O cliente pode precisar realizar uma operação adicional de **GETDEVICEINFO** para determinar qual dispositivo SCSI usar. Se estas operações funcionarem corretamente, o cliente pode emitir solicitações de E/S diretamente para o dispositivo SCSI em vez de enviar **READ** e **WRITE** operações para o servidor.

Erros ou contendas entre os clientes podem fazer com que o servidor lembre os layouts ou não os emita para os clientes. Nesses casos, os clientes voltam a emitir operações em **READ** e **WRITE** para o servidor em vez de enviar pedidos de E/S diretamente para o dispositivo SCSI.

Para monitorar as operações, ver [Seção 6.7, “Monitoramento da funcionalidade dos layouts SCSI do pNFS”](#).

## Reservas de dispositivos

pNFS SCSI lida com as vedações através da atribuição de reservas. Antes de o servidor emitir layouts para os clientes, ele reserva o dispositivo SCSI para garantir que somente clientes registrados possam acessar o dispositivo. Se um cliente pode emitir comandos para esse dispositivo SCSI mas não está registrado com o dispositivo, muitas operações do cliente nesse dispositivo falham. Por exemplo, o comando **blkid** no cliente não mostra a UUID do sistema de arquivos XFS se o servidor não tiver dado um layout para aquele dispositivo ao cliente.

O servidor não remove sua própria reserva persistente. Isto protege os dados dentro do sistema de arquivos no dispositivo através de reinicializações de clientes e servidores. A fim de redirecionar o dispositivo SCSI, talvez seja necessário remover manualmente a reserva persistente no servidor NFS.

## 6.3. VERIFICAÇÃO DE UM DISPOSITIVO SCSI COMPATÍVEL COM O PNFS

Este procedimento verifica se um dispositivo SCSI suporta o layout pNFS SCSI.

### Pré-requisitos

- Instale o pacote **sg3\_utils**:

```
# yum instalar sg3_utils
```

### Procedimento

- Tanto no servidor quanto no cliente, verifique o suporte adequado do dispositivo SCSI:

```
# sg_persist --in --reportar-capacidades --verbose path-to-scsi-device
```

Certifique-se de que o bit *Persist Through Power Loss Active* (**PTPL\_A**) esteja definido.

#### Exemplo 6.1. Um dispositivo SCSI que suporta o pNFS SCSI

A seguir, um exemplo de saída **sg\_persist** para um dispositivo SCSI que suporta o pNFS SCSI. O relatório de bit **PTPL\_A** informa **1**.

```
inquiry cdb: 12 00 00 00 24 00
Persistent Reservation In cmd: 5e 02 00 00 00 00 00 20 00 00
LIO-ORG block11      4.0
Peripheral device type: disk
Report capabilities response:
Compatible Reservation Handling(CRH): 1
Specify Initiator Ports Capable(SIP_C): 1
All Target Ports Capable(ATP_C): 1
Persist Through Power Loss Capable(PTPL_C): 1
Type Mask Valid(TMV): 1
Allow Commands: 1
Persist Through Power Loss Active(PTPL_A): 1
Support indicated in Type mask:
  Write Exclusive, all registrants: 1
  Exclusive Access, registrants only: 1
  Write Exclusive, registrants only: 1
```

```
Exclusive Access: 1
Write Exclusive: 1
Exclusive Access, all registrants: 1
```

### Recursos adicionais

- A página do homem `sg_persist(8)`

## 6.4. CONFIGURANDO O PNFS SCSI NO SERVIDOR

Este procedimento configura um servidor NFS para exportar um layout SCSI pNFS.

### Procedimento

1. No servidor, monte o sistema de arquivos XFS criado no dispositivo SCSI.
2. Configurar o servidor NFS para exportar a versão NFS 4.1 ou superior. Configurar a seguinte opção na seção `[nfsd]` do arquivo `/etc/nfs.conf`:

```
[nfsd]
vers4.1=y
```

3. Configurar o servidor NFS para exportar o sistema de arquivos XFS sobre o NFS com a opção `pnfs`:

#### Exemplo 6.2. Uma entrada em `/etc/exportação` para exportar pNFS SCSI

A seguinte entrada no arquivo de configuração `/etc/exports` exporta o sistema de arquivo montado em `/exportado/directório/` para o cliente `allowed.example.com` como um layout pNFS SCSI:

```
/exportado/diretório permitido.example.com(pnfs)
```

### Recursos adicionais

- Para mais informações sobre a configuração de um servidor NFS, veja [Capítulo 4, Exportação de ações da NFS](#).

## 6.5. INSTALAÇÃO DO PNFS SCSI NO CLIENTE

Este procedimento configura um cliente NFS para montar um layout SCSI pNFS.

### Pré-requisitos

- O servidor NFS é configurado para exportar um sistema de arquivos XFS sobre pNFS SCSI. Veja [Seção 6.4, "Configurando o pNFS SCSI no servidor"](#).

### Procedimento

- No cliente, monte o sistema de arquivos XFS exportado usando o NFS versão 4.1 ou superior:

```
# montagem -t nfs -o nfsvers=4,1 host:/remote/export /local/directory
```

Não montar o sistema de arquivo XFS diretamente sem NFS.

### Recursos adicionais

- Para mais informações sobre a montagem de ações da NFS, veja [Capítulo 3, Montagem de ações da NFS](#).

## 6.6. LIBERAÇÃO DA RESERVA DO PNFS SCSI NO SERVIDOR

Este procedimento libera a reserva persistente que um servidor NFS mantém em um dispositivo SCSI. Isto permite que você possa redirecionar o dispositivo SCSI quando não precisar mais exportar o SCSI pNFS.

Você deve remover a reserva do servidor. Ela não pode ser removida de um Nexus de TI diferente.

### Pré-requisitos

- Instale o pacote **sg3\_utils**:

```
# yum instalar sg3_utils
```

### Procedimento

1. Consultar uma reserva existente no servidor:

```
# sg_persist --ler-reserva path-to-scsi-device
```

#### Exemplo 6.3. Consultar uma reserva em /dev/sda

```
# sg_persist --read-reservation /dev/sda
LIO-ORG block_1 4.0
Peripheral device type: disk
PR generation=0x8, Reservation follows:
Key=0x1000000000000000
scope: LU_SCOPE, type: Exclusive Access, registrants only
```

2. Remover o registro existente no servidor:

```
# sg_persist --out \
  --release \
  --param-rk=reservation-key \
  --prout-type=6 \
  path-to-scsi-device
```

#### Exemplo 6.4. Removendo uma reserva em /dev/sda

```
# sg_persist --out \
  --release \
```

```
--param-rk=0x1000000000000000 \
--prout-type=6 \
/dev/sda
```

```
LIO-ORG block_1 4.0
Peripheral device type: disk
```

## Recursos adicionais

- A página do homem **sg\_persist(8)**

## 6.7. MONITORAMENTO DA FUNCIONALIDADE DOS LAYOUTS SCSI DO PNFS

Você pode monitorar se o cliente pNFS e o servidor trocam operações pNFS SCSI adequadas ou se elas recaem em operações NFS regulares.

### Pré-requisitos

- Um cliente e servidor pNFS SCSI são configurados.

### 6.7.1. Verificação das operações pNFS SCSI a partir do servidor usando o `nfsstat`

Este procedimento utiliza o utilitário **nfsstat** para monitorar as operações SCSI do pNFS a partir do servidor.

### Procedimento

1. Monitorar as operações atendidas a partir do servidor:

```
# watch --differences \
  "nfsstat --server | egrep --after-context=1 read\|write\|layout"

Every 2.0s: nfsstat --server | egrep --after-context=1 read\|write\|layout

putrootfh read readdir readlink remove rename
2 0% 0 0% 1 0% 0 0% 0 0% 0 0%
--
setctidconf verify write rellockowner bc_ctl bind_conn
0 0% 0 0% 0 0% 0 0% 0 0% 0 0%
--
getdevlist layoutcommit layoutget layoutreturn secinphonam sequence
0 0% 29 1% 49 1% 5 0% 0 0% 2435 86%
```

2. O cliente e o servidor utilizam operações pNFS SCSI quando:
  - Os balcões **layoutget**, **layoutreturn**, e **layoutcommit** incrementam. Isto significa que o servidor está servindo layouts.
  - O servidor **read** e **write** não incrementam os balcões. Isto significa que os clientes estão realizando solicitações de E/S diretamente para os dispositivos SCSI.

## 6.7.2. Verificação das operações pNFS SCSI por parte do cliente utilizando montarias

Este procedimento utiliza o arquivo `/proc/self/mountstats` para monitorar as operações pNFS SCSI do cliente.

### Procedimento

1. Liste os contadores por operação de montagem:

```
# cat /proc/self/mountstats \
  | awk /scsi_lun_0/,/^$/ \
  | egrep device|\READ|\WRITE|\LAYOUT

device 192.168.122.73:/exports/scsi_lun_0 mounted on /mnt/rhel7/scsi_lun_0 with fstype
nfs4 statvers=1.1
nfsv4:
bm0=0xfdffbfff,bm1=0x40f9be3e,bm2=0x803,acl=0x3,sessions,pnfs=LAYOUT_SCSI
  READ: 0 0 0 0 0 0 0
  WRITE: 0 0 0 0 0 0 0
  READLINK: 0 0 0 0 0 0 0
  READDIR: 0 0 0 0 0 0 0
  LAYOUTGET: 49 49 0 11172 9604 2 19448 19454
  LAYOUTCOMMIT: 28 28 0 7776 4808 0 24719 24722
  LAYOUTRETURN: 0 0 0 0 0 0 0
  LAYOUTSTATS: 0 0 0 0 0 0 0
```

2. Nos resultados:

- As estatísticas **LAYOUT** indicam solicitações onde o cliente e o servidor utilizam operações SCSI pNFS.
- As estatísticas **READ** e **WRITE** indicam solicitações onde o cliente e o servidor retornam às operações NFS.

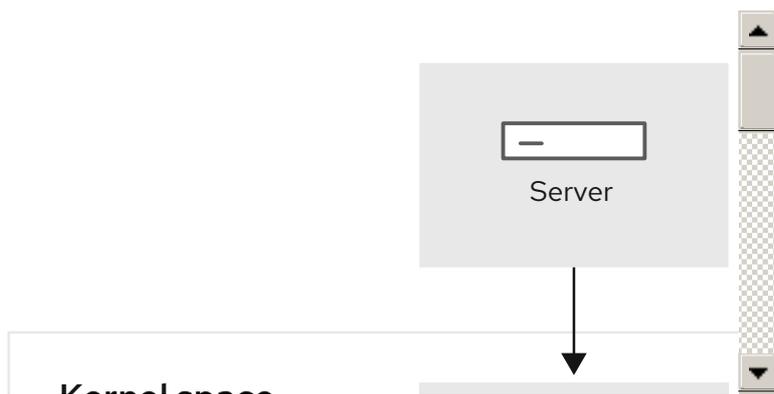
## CAPÍTULO 7. COMEÇANDO COM O FS-CACHE

O FS-Cache é um cache local persistente que os sistemas de arquivos podem usar para obter dados recuperados através da rede e armazená-los em cache em disco local. Isto ajuda a minimizar o tráfego de rede para usuários que acessam dados de um sistema de arquivo montado sobre a rede (por exemplo, NFS).

### 7.1. VISÃO GERAL DO FS-CACHE

O diagrama a seguir é uma ilustração de alto nível de como funciona o FS-Cache:

Figura 7.1. Visão geral do FS-Cache



O FS-Cache é projetado para ser o mais transparente possível para os usuários e administradores de um sistema. Ao contrário de **cachefs** no Solaris, o FS-Cache permite que um sistema de arquivo em um servidor interaja diretamente com o cache local de um cliente sem criar um sistema de arquivo super montado. Com NFS, uma opção de montagem instrui o cliente a montar o compartilhamento NFS com o FS-cache habilitado. O ponto de montagem causará o upload automático para dois módulos do kernel: **fscache** e **cachefiles**. O daemon **cachefilesd** se comunica com os módulos do kernel para implementar o cache.

O FS-Cache não altera a operação básica de um sistema de arquivo que funciona através da rede - ele apenas fornece a esse sistema de arquivo um lugar persistente no qual ele pode armazenar dados. Por exemplo, um cliente ainda pode montar um compartilhamento NFS quer o FS-Cache esteja ou não habilitado. Além disso, o NFS em cache pode lidar com arquivos que não caberão no cache (seja individual ou coletivamente), pois os arquivos podem ser parcialmente armazenados em cache e não precisam ser lidos completamente na frente. O FS-Cache também esconde todos os erros de E/S que ocorrem no cache do driver do sistema de arquivos do cliente.

Para fornecer serviços de cache, a FS-Cache precisa de um *cache back end*. Um back end de cache é um driver de armazenamento configurado para fornecer serviços de cache, que é **cachefiles**. Neste caso, o FS-Cache requer um sistema de arquivo baseado em bloco montado que suporte **bmap** e atributos estendidos (por exemplo, ext3) como seu back end de cache.

Os sistemas de arquivo que suportam as funcionalidades exigidas pelo back end de cache FS-Cache incluem as implementações do Red Hat Enterprise Linux 8 dos seguintes sistemas de arquivo:

- ext3 (com atributos estendidos ativados)
- ext4
- XFS

O FS-Cache não pode armazenar arbitrariamente qualquer sistema de arquivos, seja através da rede ou

de outra forma: o driver do sistema de arquivos compartilhado deve ser alterado para permitir interação com o FS-Cache, armazenamento/recuperação de dados e configuração e validação de metadados. O FS-Cache precisa de *indexing keys* e *coherency data* do sistema de arquivo em cache para suportar a persistência: chaves de indexação para fazer corresponder os objetos do sistema de arquivo aos objetos do cache, e dados de coerência para determinar se os objetos do cache ainda são válidos.



## NOTA

No Red Hat Enterprise Linux 8, o **cachefilesd** não é instalado por padrão e precisa ser instalado manualmente.

## 7.2. GARANTIA DE DESEMPENHO

O FS-Cache faz *not* garantir maior desempenho. O uso de um cache incorre em uma penalidade de desempenho: por exemplo, as ações NFS em cache adicionam acessos em disco para buscas em redes cruzadas. Enquanto o FS-Cache tenta ser o mais assíncrono possível, existem caminhos síncronos (por exemplo, leituras) onde isto não é possível.

Por exemplo, o uso do FS-Cache para o cache de um compartilhamento NFS entre dois computadores através de uma rede GigE sem carga, provavelmente não demonstrará nenhuma melhoria de desempenho no acesso a arquivos. Ao contrário, as solicitações NFS seriam satisfeitas mais rapidamente a partir da memória do servidor do que a partir do disco local.

O uso do FS-Cache, portanto, é um *compromise* entre vários fatores. Se o FS-Cache está sendo usado para armazenar o tráfego NFS, por exemplo, ele pode atrasar um pouco o cliente, mas reduzir maciçamente a carga da rede e do servidor, satisfazendo as solicitações de leitura localmente sem consumir a largura de banda da rede.

## 7.3. MONTANDO UM CACHE

Atualmente, o Red Hat Enterprise Linux 8 fornece apenas o back end de cache **cachefiles**. O daemon **cachefilesd** inicia e gerencia **cachefiles**. O arquivo **/etc/cachefilesd.conf** controla como **cachefiles** fornece serviços de caching.

O back end do cache funciona mantendo uma certa quantidade de espaço livre na partição que hospeda o cache. Ele cresce e diminui o cache em resposta a outros elementos do sistema que utilizam o espaço livre, tornando-o seguro para uso no sistema de arquivos raiz (por exemplo, em um laptop). O FS-Cache estabelece padrões neste comportamento, que podem ser configurados via *cache cull limits*. Para mais informações sobre a configuração dos limites de cache de eliminação, veja [Seção 7.5, "Configuração dos limites de abate de caches"](#).

Este procedimento mostra como montar um cache.

### Pré-requisitos

- O **cachefilesd** e o serviço começou com sucesso. Para ter certeza de que o serviço está funcionando, use o seguinte comando:

```
# systemctl start cachefilesd
# systemctl status cachefilesd
```

O status deve ser *active (running)*.

### Procedimento

1. Configure em um back end de cache qual diretório usar como cache, use o seguinte parâmetro:

```
$ dir /path/to/cache
```

2. Tipicamente, o diretório back end do cache é definido em **/etc/cachefilesd.conf** como **/var/cache/fscache**, como em:

```
$ dir /var/cache/fscache
```

3. Se você quiser mudar o diretório back end do cache, o contexto do selinux deve ser o mesmo que **/var/cache/fscache**:

```
# semanage fcontext -a -e /var/cache/fscache /path/to/cache
# restorecon -Rv /path/to/cache
```

4. Substitua */path/to/cache* pelo nome do diretório durante a configuração do cache.
5. Se os comandos dados para definir o contexto selinux não funcionaram, use os seguintes comandos:

```
# semanage permissive -a cachefilesd_t
# semanage permissive -a cachefiles_kernel_t
```

O FS-Cache armazenará o cache no sistema de arquivo que hospeda **/path/to/cache**. Em um laptop, é aconselhável usar o sistema de arquivos raiz (*/*) como o sistema de arquivos host, mas para uma máquina desktop seria mais prudente montar uma partição de disco especificamente para o cache.

6. O sistema de arquivo host deve suportar atributos estendidos definidos pelo usuário; o FS-Cache usa esses atributos para armazenar informações de manutenção de coerência. Para permitir atributos estendidos definidos pelo usuário para sistemas de arquivos ext3 (ou seja **device**), use:

```
# tune2fs -o user_xattr /device/device
```

7. Para ativar atributos estendidos para um sistema de arquivo no momento da montagem, como alternativa, use o seguinte comando:

```
# montar /dispositivo/dispositivo /caminho/para/cache -o user_xattr
```

8. Uma vez que o arquivo de configuração esteja pronto, inicie o serviço **cachefilesd**:

```
# systemctl start cachefilesd
```

9. Para configurar **cachefilesd** para iniciar no momento do boot, execute o seguinte comando como root:

```
# systemctl habilita os arquivos de cache
```

## 7.4. USANDO O CACHE COM NFS

A NFS não utilizará o cache a menos que seja explicitamente instruída. Este parágrafo mostra como configurar uma montagem NFS usando o FS-Cache.

### Pré-requisitos

- O **cachefilesd** pacote está instalado e funcionando. Para garantir a sua execução, use o seguinte comando:

```
# systemctl start cachefilesd
# systemctl status cachefilesd
```

O status deve ser *active (running)*.

- Monte as ações da NFS com a seguinte opção:

```
# mount nfs-share:/ /mount/point -o fsc
```

Todo acesso a arquivos em **/mount/point** passará através do cache, a menos que o arquivo seja aberto para E/S direta ou escrita. Para maiores informações, veja [Seção 7.4.2, “Limitações de cache com NFS”](#). NFS indexa o conteúdo do cache usando o handle do arquivo NFS, *not* o nome do arquivo, o que significa que os arquivos com link rígido compartilham o cache corretamente.

As versões NFS 3, 4.0, 4.1 e 4.2 suportam o caching. Entretanto, cada versão utiliza ramos diferentes para o caching.

### 7.4.1. Configuração do compartilhamento de cache NFS

Há vários problemas potenciais relacionados ao compartilhamento de cache NFS. Como o cache é persistente, os blocos de dados no cache são indexados em uma seqüência de quatro chaves:

- Nível 1: Detalhes do servidor
- Nível 2: Algumas opções de montagem; tipo de segurança; FSID; unificadores
- Nível 3: File Handle
- Nível 4: Número de página no arquivo

Para evitar problemas de gerenciamento de coerência entre superblocos, todos os superblocos NFS que requerem o cache dos dados têm chaves únicas de nível 2. Normalmente, dois NFS montados com o mesmo volume de fonte e opções compartilham um superbloco, e assim compartilham o cache, mesmo que eles montem diretórios diferentes dentro desse volume.

Este é um exemplo de como configurar o compartilhamento de cache com diferentes opções.

### Procedimento

1. Monte NFS compartilha com os seguintes comandos:

```
mount home0:/disk0/fred /home/fred -o fsc
mount home0:/disk0/jim /home/jim -o fsc
```

Aqui, **/home/fred** e **/home/jim** provavelmente compartilham o superbloco, pois têm as mesmas opções, especialmente se vierem do mesmo volume/partição no servidor NFS (**home0**).

- Para não compartilhar o superbloco, use o comando **mount** com as seguintes opções:

```
mount home0:/disk0/fred /home/fred -o fsc,rsize=8192
mount home0:/disk0/jim /home/jim -o fsc,rsize=65536
```

Neste caso, **/home/fred** e **/home/jim** não compartilharão o superbloco, pois têm parâmetros diferentes de acesso à rede, que fazem parte da chave do Nível 2.

- Para armazenar o conteúdo das duas subárvores (**/home/fred1** e **/home/fred2**) *twice* com o não compartilhamento do superbloco, use o seguinte comando:

```
mount home0:/disk0/fred /home/fred1 -o fsc,rsize=8192
mount home0:/disk0/fred /home/fred2 -o fsc,rsize=65536
```

- Outra maneira de evitar o compartilhamento de superblocos é suprimi-lo explicitamente com o parâmetro **nosharecache**. Usando o mesmo exemplo:

```
mount home0:/disk0/fred /home/fred -o nosharecache,fsc
mount home0:/disk0/jim /home/jim -o nosharecache,fsc
```

Entretanto, neste caso, apenas um dos superblocos é permitido usar o cache, já que não há nada que distinga as chaves de nível 2 de **home0:/disk0/fred** e **home0:/disk0/jim**.

- Para especificar o endereçamento ao superbloco, adicionar um endereço *unique identifier* em pelo menos uma das montagens, ou seja **fsc=unique-identifier**.

```
mount home0:/disk0/fred /home/fred -o nosharecache,fsc
mount home0:/disk0/jim /home/jim -o nosharecache,fsc=jim
```

Aqui, o identificador único **jim** é adicionado à chave de nível 2 usada no cache para **/home/jim**.



### IMPORTANTE

O usuário não pode compartilhar caches entre superblocos que têm comunicações ou parâmetros de protocolo diferentes. Por exemplo, não é possível compartilhar entre NFSv4.0 e NFSv3 ou entre NFSv4.1 e NFSv4.2 porque eles forçam superblocos diferentes. Também a definição de parâmetros, como o tamanho de leitura (*rsize*), impede o compartilhamento do cache porque, novamente, força um superbloqueio diferente.

## 7.4.2. Limitações de cache com NFS

Há algumas limitações de cache com NFS:

- A abertura de um arquivo a partir de um sistema de arquivo compartilhado para E/S direta contorna automaticamente o cache. Isto porque este tipo de acesso deve ser direto ao servidor.
- A abertura de um arquivo a partir de um sistema de arquivo compartilhado para E/S direta ou escrita descarrega a cópia em cache do arquivo. O FS-Cache não armazenará novamente o arquivo em cache até que ele não seja mais aberto para E/S direta ou para escrita.
- Além disso, este lançamento do FS-Cache somente armazena arquivos NFS regulares. O FS-Cache irá *not*. Diretórios de cache, symlinks, arquivos de dispositivos, FIFOs e soquetes.

## 7.5. CONFIGURAÇÃO DOS LIMITES DE ABATE DE CACHES

O daemon **cachefilesd** funciona através do cache de dados remotos de sistemas de arquivos compartilhados para liberar espaço no disco. Isto poderia potencialmente consumir todo o espaço livre disponível, o que poderia ser ruim se o disco também abrigasse a partição raiz. Para controlar isto, **cachefilesd** tenta manter uma certa quantidade de espaço livre descartando objetos antigos (ou seja, acessados menos recentemente) do cache. Este comportamento é conhecido como *cache culling*.

O abate de cache é feito com base na porcentagem de blocos e na porcentagem de arquivos disponíveis no sistema de arquivos subjacente. Há configurações em **/etc/cachefilesd.conf** que controlam seis limites:

### **brun N%** (porcentagem de blocos), **frun N%** (porcentagem de arquivos)

Se a quantidade de espaço livre e o número de arquivos disponíveis no cache aumentar acima desses dois limites, então o abate é desligado.

### **bcull N%** (porcentagem de blocos), **fcull N%** (porcentagem de arquivos)

Se a quantidade de espaço disponível ou o número de arquivos no cache cair abaixo de qualquer um desses limites, então o abate é iniciado.

### **bstop N%** (porcentagem de blocos), **fstop N%** (porcentagem de arquivos)

Se a quantidade de espaço disponível ou o número de arquivos disponíveis no cache cair abaixo de qualquer um desses limites, então não é permitida mais nenhuma alocação de espaço em disco ou arquivos até que o abate tenha aumentado novamente as coisas acima desses limites.

O valor padrão de **N** para cada configuração é o seguinte:

- **brun/frun** - 10%
- **bcull/fcull** - 7%
- **bstop/fstop** - 3%

Ao configurar estas configurações, o seguinte deve se manter verdadeiro:

- 0
- 0

Estas são as porcentagens de espaço disponível e arquivos disponíveis e não aparecem como 100 menos a porcentagem exibida pelo programa **df**.



### IMPORTANTE

O abate depende dos pares **bxxx** e **fxxx** simultaneamente; o usuário não pode tratá-los separadamente.

## 7.6. OBTENÇÃO DE INFORMAÇÕES ESTATÍSTICAS DO MÓDULO FSCACHE KERNEL

O FS-Cache também mantém registro de informações estatísticas gerais. Este procedimento mostra como obter estas informações.

### Procedimento

1. Para visualizar as informações estatísticas sobre o FS-Cache, use o seguinte comando:

```
# cat /proc/fs/fscache/stats
```

As estatísticas do FS-Cache incluem informações sobre pontos de decisão e contadores de objetos. Para maiores informações, veja o seguinte documento do kernel:

**[/usr/share/doc/kernel-doc-4.18.0/Documentation/filesystems/caching/fscache.txt](#)**

## 7.7. REFERÊNCIAS DO FS-CACHE

Esta seção fornece informações de referência para o FS-Cache.

1. Para mais informações sobre **cachefilesd** e como configurá-lo, veja **man cachefilesd** e **man cachefilesd.conf**. Os seguintes documentos do kernel também fornecem informações adicionais:
  - **[/usr/share/doc/cachefilesd/README](#)**
  - **[/usr/share/man/man5/cachefilesd.conf.5.gz](#)**
  - **[/usr/share/man/man8/cachefilesd.8.gz](#)**
2. Para informações gerais sobre o FS-Cache, incluindo detalhes sobre suas restrições de projeto, estatísticas disponíveis e capacidades, veja o seguinte documento do kernel:  
**[/usr/share/doc/kernel-doc-4.18.0/Documentation/filesystems/caching/fscache.txt](#)**

## CAPÍTULO 8. MONTAGEM DE UMA AÇÃO SMB NO RED HAT ENTERPRISE LINUX

O protocolo Server Message Block (SMB) implementa um protocolo de rede de camada de aplicação utilizado para acessar recursos em um servidor, tais como compartilhamento de arquivos e impressoras compartilhadas.



### NOTA

No contexto do SMB, você pode encontrar menções sobre o protocolo CIFS (Common Internet File System), que é um dialeto do SMB. Tanto o protocolo SMB quanto o CIFS são suportados, e o módulo do kernel e utilitários envolvidos na montagem de compartilhamentos SMB e CIFS usam ambos o nome **cifs**.

Esta seção descreve como montar ações a partir de um servidor SMB. Para detalhes sobre como montar um servidor SMB no Red Hat Enterprise Linux usando Samba, veja [Usando o Samba como um servidor](#).

### Pré-requisitos

No Microsoft Windows, o SMB é implementado por padrão. No Red Hat Enterprise Linux, o módulo de sistema de arquivo **cifs.ko** do kernel fornece suporte para montagem de ações SMB. Por isso, instale o pacote **cifs-utils**:

```
# yum instalar cifs-utils
```

O pacote **cifs-utils** fornece utilidades para:

- Montar ações SMB e CIFS
- Gerenciar as credenciais do NT Lan Manager (NTLM) no chaveiro do kernel
- Definir e exibir Listas de Controle de Acesso (ACL) em um descritor de segurança sobre ações SMB e CIFS

## 8.1. VERSÕES DE PROTOCOLO SMB SUPORTADAS

O módulo do kernel **cifs.ko** suporta as seguintes versões do protocolo SMB:

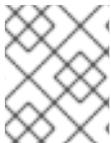
- SMB 1



### ATENÇÃO

O protocolo SMB1 é depreciado devido a questões de segurança conhecidas, e é apenas **safe to use on a private network**. A principal razão pela qual o SMB1 ainda é fornecido como uma opção suportada é que atualmente é a única versão do protocolo SMB que suporta extensões UNIX. Se você não precisa usar extensões UNIX em SMB, a Red Hat recomenda fortemente o uso do SMB2 ou posterior.

- SMB 2.0
- SMB 2.1
- SMB 3.0
- SMB 3.1.1



#### NOTA

Dependendo da versão do protocolo, nem todas as características SMB são implementadas.

## 8.2. SUPORTE A EXTENSÕES UNIX

O Samba usa o bit de capacidade **CAP\_UNIX** no protocolo SMB para fornecer o recurso de extensões UNIX. Estas extensões também são suportadas pelo módulo **cifs.ko** do kernel. No entanto, tanto o Samba quanto o módulo do kernel suportam extensões UNIX somente no protocolo SMB 1.

Para utilizar extensões UNIX:

1. Defina o parâmetro **server min protocol** na seção **[global]** no arquivo **/etc/samba/smb.conf** para **NT1**.
2. Monte a ação usando o protocolo SMB 1, fornecendo a opção **-o vers=1.0** para o comando de montagem. Por exemplo:

```
# mount -t cifs -o vers=1.0,username=user_name //server_name/share_name /mnt/
```

Por padrão, o módulo do kernel usa SMB 2 ou a versão de protocolo posterior mais alta suportada pelo servidor. Passando a opção **-o vers=1.0** para o comando **mount**, o módulo do kernel usa o protocolo SMB 1 que é necessário para o uso de extensões UNIX.

Para verificar se as extensões UNIX estão habilitadas, exibir as opções da ação montada:

```
# mount
...
//server/share on /mnt type cifs (... ,unix,...)
```

Se a entrada **unix** for exibida na lista de opções de montagem, as extensões UNIX estarão habilitadas.

## 8.3. MONTAGEM MANUAL DE UMA AÇÃO SMB

Se você precisar apenas de uma ação SMB para ser montada temporariamente, você pode montá-la manualmente usando o utilitário **mount**.



#### NOTA

As ações montadas manualmente não são montadas automaticamente novamente quando se reinicia o sistema. Para configurar que o Red Hat Enterprise Linux monta automaticamente a ação quando o sistema é reinicializado, veja [Seção 8.4, “Montagem automática de um compartilhamento SMB quando o sistema inicia”](#).

### Pré-requisitos

- O pacote **cifs-utils** está instalado.

## Procedimento

Para montar manualmente um compartilhamento SMB, use o utilitário **mount** com o parâmetro **-t cifs**:

```
# mount -t cifs -o username=user_name //server_name/share_name /mnt/
Password for user_name@//server_name/share_name: password
```

No parâmetro **-o**, você pode especificar as opções que são usadas para montar a ação. Para detalhes, consulte [Seção 8.7, "Opções de montagem frequentemente utilizadas"](#) e a seção **OPTIONS** na página de manual **mount.cifs(8)**.

### Exemplo 8.1. Montagem de uma ação utilizando uma conexão SMB 3.0 criptografada

To mount the `\\server\example\` share as the **DOMAINAdministrator** user over an encrypted SMB 3.0 connection into the `/mnt/` directory:

```
# mount -t cifs -o username=DOMAINAdministrator,seal,vers=3.0 //server/example /mnt/
Password for DOMAINAdministrator@//server_name/share_name: password
```

## 8.4. MONTAGEM AUTOMÁTICA DE UM COMPARTILHAMENTO SMB QUANDO O SISTEMA INICIA

Se o acesso a um compartilhamento SMB montado for permanentemente necessário em um servidor, monte o compartilhamento automaticamente no momento da inicialização.

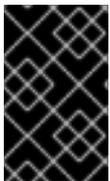
### Pré-requisitos

- O pacote **cifs-utils** está instalado.

## Procedimento

Para montar um compartilhamento SMB automaticamente quando o sistema inicia, adicione uma entrada para o compartilhamento ao arquivo `/etc/fstab`. Por exemplo:

```
//server_name/share_name /mnt cifs credentials=/root/smb.cred 0 0
```



### IMPORTANTE

Para permitir que o sistema monte uma ação automaticamente, você deve armazenar o nome de usuário, senha e nome de domínio em um arquivo de credenciais. Para detalhes, veja [Seção 8.5, "Autenticação para uma ação SMB usando um arquivo de credenciais"](#).

No quarto campo da linha no `/etc/fstab`, especificar as opções de montagem, como o caminho para o arquivo de credenciais. Para detalhes, consulte [Seção 8.7, "Opções de montagem frequentemente utilizadas"](#) e a seção **OPTIONS** na página de manual **mount.cifs(8)**.

Para verificar se a ação é montada com sucesso, entre:

```
# montar /mnt/
```

## 8.5. AUTENTICAÇÃO PARA UMA AÇÃO SMB USANDO UM ARQUIVO DE CREDENCIAIS

Em certas situações, como ao montar uma ação automaticamente no momento da inicialização, uma ação deve ser montada sem a digitação do nome de usuário e senha. Para implementar isto, crie um arquivo de credenciais.

### Pré-requisitos

- O pacote **cifs-utils** está instalado.

### Procedimento

1. Criar um arquivo, tal como **/root/smb.cred**, e especificar o nome de usuário, senha e nome de domínio desse arquivo:

```
username=user_name
password=password
domain=domain_name
```

2. Defina as permissões para permitir apenas que o proprietário tenha acesso ao arquivo:

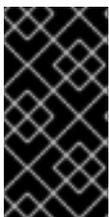
```
# chown user_name /root/smb.cred
# chmod 600 /root/smb.cred
```

Agora você pode passar o **credentials=file\_name** opção de montagem para o utilitário **mount** ou utilizá-lo no arquivo **/etc/fstab** para montar a ação sem ser solicitado o nome do usuário e a senha.

## 8.6. EXECUÇÃO DE UMA MONTAGEM SMB MULTIUSUÁRIO

As credenciais que você fornece para montar uma ação determinam as permissões de acesso no ponto de montagem, por padrão. Por exemplo, se você usar o **DOMAINexample** ao montar uma ação, todas as operações sobre a ação serão executadas como este usuário, independentemente do usuário local que realiza a operação.

Entretanto, em certas situações, o administrador quer montar uma ação automaticamente quando o sistema inicia, mas os usuários devem realizar ações sobre o conteúdo da ação usando suas próprias credenciais. As opções de montagem do **multiuser** permitem configurar este cenário.



### IMPORTANTE

Para usar a opção de montagem **multiuser**, você deve definir adicionalmente a opção de montagem **sec** para um tipo de segurança que suporte o fornecimento de credenciais de forma não interativa, como **krb5** ou a opção **ntlmssp** com um arquivo de credenciais. Para detalhes, veja [Seção 8.6.3, "Acesso a uma ação como um usuário"](#).

O usuário **root** monta a ação usando a opção **multiuser** e uma conta que tem acesso mínimo ao conteúdo da ação. Os usuários regulares podem então fornecer seu nome de usuário e senha para o chaveiro do kernel da sessão atual usando o utilitário **cifscreds**. Se o usuário acessar o conteúdo do compartilhamento montado, o kernel usa as credenciais do chaveiro do kernel ao invés do utilizado inicialmente para montar o compartilhamento.

A utilização desta característica consiste nas seguintes etapas:

- Monte uma ação com a opção **multiuser** .
- Opcionalmente, verificar se a ação foi montada com sucesso com a opção **multiuser** .
- Acesse a ação como um usuário .

### Pré-requisitos

- O pacote **cifs-utils** está instalado.

### 8.6.1. Montagem de uma ação com a opção multiusuário

Antes que os usuários possam acessar o compartilhamento com suas próprias credenciais, monte o compartilhamento como o usuário **root** usando uma conta com permissões limitadas.

#### Procedimento

Para montar um compartilhamento automaticamente com a opção **multiuser** quando o sistema inicia:

1. Crie a entrada para a participação no arquivo **/etc/fstab**. Por exemplo:

```
//server_name/share_name /mnt cifs multiuser,sec=ntlmssp,credentials=/root/smb.cred
0 0
```

2. Monte a ação:

```
# montar /mnt/
```

Se você não quiser montar o compartilhamento automaticamente quando o sistema iniciar, monte-o manualmente passando **-o multiuser,sec=security\_type** para o comando **mount**. Para detalhes sobre a montagem manual de um compartilhamento SMB, veja [Seção 8.3, “Montagem manual de uma ação SMB”](#).

### 8.6.2. Verificar se uma ação SMB é montada com a opção multiusuário

Para verificar se uma ação está montada com a opção **multiuser**, exibir as opções de montagem.

#### Procedimento

```
# mount
...
//server_name/share_name on /mnt type cifs (sec=ntlmssp,multiuser,...)
```

Se a entrada **multiuser** for exibida na lista de opções de montagem, o recurso é ativado.

### 8.6.3. Acesso a uma ação como um usuário

Se um compartilhamento SMB for montado com a opção **multiuser**, os usuários podem fornecer suas credenciais para o servidor ao chaveiro do kernel:

```
# cifscreds add -u SMB_user_name server_name
Password: password
```

Quando o usuário realiza operações no diretório que contém o compartilhamento SMB montado, o servidor aplica as permissões do sistema de arquivos para este usuário, ao invés daquela utilizada inicialmente quando o compartilhamento foi montado.



## NOTA

Vários usuários podem realizar operações usando suas próprias credenciais sobre a ação montada ao mesmo tempo.

## 8.7. OPÇÕES DE MONTAGEM FREQUENTEMENTE UTILIZADAS

Quando você monta uma ação SMB, as opções de montagem são determinantes:

- Como a conexão será estabelecida com o servidor. Por exemplo, qual versão do protocolo SMB é usada quando se conecta ao servidor.
- Como a ação será montada no sistema de arquivo local. Por exemplo, se o sistema anular o arquivo remoto e as permissões de diretório para permitir que vários usuários locais acessem o conteúdo no servidor.

Para definir várias opções no quarto campo do arquivo `/etc/fstab` ou no parâmetro `-o` de um comando de montagem, separe as opções com vírgulas. Por exemplo, ver [Seção 8.6.1, "Montagem de uma ação com a opção multiusuário"](#).

A lista a seguir fornece opções de montagem frequentemente utilizadas:

Opção	Descrição
<code>credenciais=file_name</code>	Estabelece o caminho para o arquivo de credenciais. Ver <a href="#">Seção 8.5, "Autenticação para uma ação SMB usando um arquivo de credenciais"</a>
<code>dir_mode=mode</code>	Define o modo de diretório se o servidor não suportar as extensões CIFS UNIX.
<code>file_mode=mode</code>	Define o modo de arquivo se o servidor não suportar as extensões CIFS UNIX.
<code>senha=password</code>	Define a senha usada para autenticar no servidor SMB. Alternativamente, especifique um arquivo de credenciais usando a opção <b>credentials</b> .
<code>selo</code>	Permite suporte de criptografia para conexões usando SMB 3.0 ou uma versão de protocolo posterior. Portanto, use <b>seal</b> junto com a opção de montagem <b>vers</b> definida para <b>3.0</b> ou posterior. Ver <a href="#">Exemplo 8.1, "Montagem de uma ação utilizando uma conexão SMB 3.0 criptografada"</a> .

Opção	Descrição
<code>sec=security_mode</code>	<p>Define o modo de segurança, tal como <b>ntlmsspi</b>, para permitir o hashing da senha NTLMv2 e habilitar a assinatura de pacotes. Para uma lista dos valores suportados, veja a descrição da opção na página de manual <b>mount.cifs(8)</b>.</p> <p>Se o servidor não suporta o modo de segurança <b>ntlmv2</b>, use <b>sec=ntlmssp</b>, que é o padrão.</p> <p>Por razões de segurança, não utilize o inseguro modo de segurança <b>ntlm</b>.</p>
<code>username=user_name</code>	<p>Define o nome do usuário usado para autenticar no servidor SMB. Alternativamente, especifique um arquivo de credenciais usando a opção <b>credentials</b>.</p>
<code>vers=SMB_protocol_version</code>	<p>Define a versão do protocolo SMB utilizada para a comunicação com o servidor.</p>

Para obter uma lista completa, consulte a seção **OPTIONS** na página de manual **mount.cifs(8)**.

## CAPÍTULO 9. VISÃO GERAL DOS ATRIBUTOS DE NOMEAÇÃO PERSISTENTES

Como administrador de sistema, você precisa se referir aos volumes de armazenamento usando atributos de nomes persistentes para construir configurações de armazenamento que sejam confiáveis em várias boots de sistema.

### 9.1. DESVANTAGENS DOS ATRIBUTOS DE NOMENCLATURA NÃO-PERSISTENTES

O Red Hat Enterprise Linux oferece uma série de maneiras de identificar dispositivos de armazenamento. É importante usar a opção correta para identificar cada dispositivo quando usado a fim de evitar o acesso inadvertido ao dispositivo errado, particularmente ao instalar ou reformatar unidades.

Tradicionalmente, nomes não-persistentes na forma de `/dev/sd(major number)(minor number)` são usados no Linux para se referir aos dispositivos de armazenamento. A faixa de números maiores e menores e os nomes associados a **sd** são alocados para cada dispositivo quando ele é detectado. Isto significa que a associação entre o intervalo de números maior e menor e os nomes associados **sd** podem mudar se a ordem de detecção do dispositivo mudar.

Tal mudança no ordenamento pode ocorrer nas seguintes situações:

- A paralelização do processo de inicialização do sistema detecta os dispositivos de armazenamento em uma ordem diferente com cada inicialização do sistema.
- Um disco não consegue ligar ou responder ao controlador SCSI. Isto faz com que ele não seja detectado pela sonda normal do dispositivo. O disco não é acessível ao sistema e os dispositivos subsequentes terão sua faixa de maior e menor número, incluindo os nomes associados **sd** deslocados para baixo. Por exemplo, se um disco normalmente referido como **sdb** não for detectado, um disco que normalmente é referido como **sdc** aparecerá como **sdb**.
- Um controlador SCSI (host bus adapter, ou HBA) não se inicializa, fazendo com que todos os discos conectados a esse HBA não sejam detectados. A quaisquer discos conectados a HBAs subsequentemente sondados são atribuídas diferentes faixas de números maiores e menores, e diferentes nomes associados a **sd**.
- A ordem de inicialização do motorista muda se diferentes tipos de HBAs estiverem presentes no sistema. Isto faz com que os discos conectados a esses HBAs sejam detectados em uma ordem diferente. Isto também pode ocorrer se os HBAs forem movidos para diferentes slots PCI no sistema.
- Os discos conectados ao sistema com adaptadores Fibre Channel, iSCSI ou FCoE podem estar inacessíveis no momento em que os dispositivos de armazenamento são sondados, devido a uma matriz de armazenamento ou interruptor de intervenção sendo desligado, por exemplo. Isto pode ocorrer quando um sistema é reinicializado após uma falha de energia, se a matriz de armazenamento levar mais tempo para ficar on-line do que o sistema leva para arrancar. Embora alguns drivers Fibre Channel suportem um mecanismo para especificar uma identificação de alvo SCSI persistente para o mapeamento WWPN, isto não faz com que as faixas de números maiores e menores, e os nomes associados **sd** sejam reservados; ele só fornece números de identificação de alvo SCSI consistentes.

Estas razões tornam indesejável o uso da faixa de números maiores e menores ou dos nomes associados **sd** ao se referir aos dispositivos, como no arquivo `/etc/fstab`. Existe a possibilidade de que o dispositivo errado seja montado e a corrupção de dados possa resultar.

Ocasionalmente, porém, ainda é necessário consultar os nomes **sd** mesmo quando outro mecanismo é utilizado, como quando são relatados erros por um dispositivo. Isto porque o kernel Linux usa os nomes **sd** (e também os nomes SCSI host/channel/target/LUN tuples) nas mensagens do kernel referentes ao dispositivo.

## 9.2. SISTEMA DE ARQUIVOS E IDENTIFICADORES DE DISPOSITIVOS

Esta seção explica a diferença entre atributos persistentes que identificam sistemas de arquivos e dispositivos de blocos.

### Identificadores do sistema de arquivo

Os identificadores do sistema de arquivo são vinculados a um sistema de arquivo particular criado em um dispositivo de bloco. O identificador também é armazenado como parte do sistema de arquivo. Se você copiar o sistema de arquivo para um dispositivo diferente, ele ainda carrega o mesmo identificador de sistema de arquivo. Por outro lado, se você reescrever o dispositivo, por exemplo, formatando-o com o utilitário **mkfs**, o dispositivo perde o atributo.

Os identificadores do sistema de arquivo incluem:

- Identificador único (UUID)
- Etiqueta

### Identificadores de dispositivos

Os identificadores de dispositivos são vinculados a um dispositivo de bloco: por exemplo, um disco ou uma partição. Se você reescrever o dispositivo, por exemplo, formatando-o com o utilitário **mkfs**, o dispositivo mantém o atributo, pois ele não é armazenado no sistema de arquivos.

Os identificadores de dispositivos incluem:

- World Wide Identifier (WWID)
- Partição UUID
- Número de série

### Recomendações

- Alguns sistemas de arquivo, tais como volumes lógicos, abrangem vários dispositivos. A Red Hat recomenda acessar esses sistemas de arquivo usando identificadores de sistemas de arquivo em vez de identificadores de dispositivos.

## 9.3. NOMES DE DISPOSITIVOS GERENCIADOS PELO MECANISMO UDEV EM /DEV/DISCO/

Esta seção lista diferentes tipos de atributos de nomes persistentes que o serviço **udev** fornece no diretório **/dev/disk/**.

O mecanismo **udev** é usado para todos os tipos de dispositivos no Linux, não apenas para dispositivos de armazenamento. No caso de dispositivos de armazenamento, o Red Hat Enterprise Linux contém **udev** regras que criam links simbólicos no diretório **/dev/disk/**. Isto permite que você se refira aos dispositivos de armazenamento por:

- Seu conteúdo
- Um identificador único

- Seu número de série.

Embora os atributos de nomenclatura **udev** sejam persistentes, na medida em que não mudam por si mesmos através de reinicializações do sistema, alguns também são configuráveis.

### 9.3.1. Identificadores do sistema de arquivo

#### O atributo UUID em `/dev/disco/by-uuid/`

As entradas neste diretório fornecem um nome simbólico que se refere ao dispositivo de armazenamento por um **unique identifier** (UUID) no conteúdo (ou seja, os dados) armazenados no dispositivo. Por exemplo, o nome do dispositivo:

```
/dev/disco/by-uuid/3e6be9de-8139-11d1-9106-a43f08d823a6
```

Você pode usar a UUID para consultar o dispositivo no arquivo **/etc/fstab** usando a seguinte sintaxe:

```
UUID=3e6be9de-8139-11d1-9106-a43f08d823a6
```

Você pode configurar o atributo UUID ao criar um sistema de arquivo, e também pode alterá-lo mais tarde.

#### O atributo Label in `/dev/disk/by-label/`

As entradas neste diretório fornecem um nome simbólico que se refere ao dispositivo de armazenamento por um **label** no conteúdo (ou seja, os dados) armazenados no dispositivo.

Por exemplo:

```
/dev/disco/by-label/Boot
```

Você pode usar a etiqueta para se referir ao dispositivo no arquivo **/etc/fstab** usando a seguinte sintaxe:

```
LABEL=Boot
```

Você pode configurar o atributo Label ao criar um sistema de arquivo, e também pode alterá-lo posteriormente.

### 9.3.2. Identificadores de dispositivos

#### O atributo WWID em `/dev/disco/by-id/`

O World Wide Identifier (WWID) é um persistente, **system-independent identifier** que a norma SCSI exige de todos os dispositivos SCSI. O identificador WWID é garantido como único para cada dispositivo de armazenamento, e independente do caminho que é usado para acessar o dispositivo. O identificador é uma propriedade do dispositivo, mas não é armazenado no conteúdo (ou seja, os dados) dos dispositivos.

Este identificador pode ser obtido emitindo um SCSI Inquiry para recuperar os Dados Vitais de Identificação do Dispositivo (página **0x83**) ou o Número de Série da Unidade (página **0x80**).

O Red Hat Enterprise Linux mantém automaticamente o mapeamento adequado do nome do dispositivo baseado na WWID para um nome atual **/dev/sd** naquele sistema. As aplicações podem usar o nome **/dev/disk/by-id/** para referenciar os dados no disco, mesmo que o caminho para o dispositivo mude, e mesmo ao acessar o dispositivo a partir de sistemas diferentes.

#### Exemplo 9.1. Mapas da WWID

WWID symlink	Dispositivo não-persistente	Nota
<code>/dev/disk/by-id/scsi-3600508b400105e210000900000490000</code>	<code>/dev/sda</code>	Um dispositivo com uma página <b>0x83</b> identificador
<code>/dev/disk/by-id/scsi-SSEAGATE_ST373453LW_3HW1RHM6</code>	<code>/dev/sdb</code>	Um dispositivo com uma página <b>0x80</b> identificador
<code>/dev/disk/by-id/ata-SAMSUNG_MZNLN256HHMQ-000L7_S2WDX0J336519-part3</code>	<code>/dev/sdc3</code>	Uma partição de disco

Além desses nomes persistentes fornecidos pelo sistema, você também pode usar as regras do **udev** para implementar nomes persistentes próprios, mapeados para a WWID do armazenamento.

### O atributo Partição UUID em `/dev/disco/by-partuuid`

O atributo Partição UUID (PARTUUID) identifica as partições como definidas pela tabela de partição GPT.

#### Exemplo 9.2. Mapeamentos de partição UUID

Link simbólico PARTUÍDO	Dispositivo não-persistente
<code>/dev/disk/by-partuuid/4cd1448a-01</code>	<code>/dev/sda1</code>
<code>/dev/disk/by-partuuid/4cd1448a-02</code>	<code>/dev/sda2</code>
<code>/dev/disk/by-partuuid/4cd1448a-03</code>	<code>/dev/sda3</code>

### O atributo Caminho em `/dev/disco/by-path/`

Este atributo fornece um nome simbólico que se refere ao dispositivo de armazenamento pelo **hardware path** usado para acessar o dispositivo.

O atributo Path falha se qualquer parte do caminho do hardware (por exemplo, o PCI ID, porta de destino, ou número LUN) mudar. O atributo Caminho não é, portanto, confiável. Entretanto, o atributo Caminho pode ser útil em um dos seguintes cenários:

- Você precisa identificar um disco que está planejando substituir mais tarde.
- Você planeja instalar um serviço de armazenamento em um disco em um local específico.

## 9.4. O IDENTIFICADOR MUNDIAL COM DM MULTIPATH

Esta seção descreve o mapeamento entre o World Wide Identifier (WWID) e os nomes de dispositivos não persistentes em uma configuração Multipath Mapper Device Mapper.

Se houver múltiplos caminhos de um sistema para um dispositivo, a DM Multipath usa a WWID para detectar isso. A DM Multipath então apresenta um único "pseudo-dispositivo" no diretório `/dev/mapper/wwid`, tal como `/dev/mapper/3600508b400105df70000e00000ac0000`.

O comando `multipath -l` mostra o mapeamento aos identificadores não-persistentes:

- **Host:Channel:Target:LUN**
- `/dev/sd` nome
- **major:minor** número

### Exemplo 9.3. Mapas da WWID em uma configuração multipath

Um exemplo de saída do comando `multipath -l`:

```
3600508b400105df70000e00000ac0000 dm-2 vendor,product
[size=20G][features=1 queue_if_no_path][hwhandler=0][rw]
\_ round-robin 0 [prio=0][active]
\_ 5:0:1:1 sdc 8:32 [active][undef]
\_ 6:0:1:1 sdg 8:96 [active][undef]
\_ round-robin 0 [prio=0][enabled]
\_ 5:0:0:1 sdb 8:16 [active][undef]
\_ 6:0:0:1 sdf 8:80 [active][undef]
```

DM Multipath mantém automaticamente o mapeamento adequado de cada nome de dispositivo baseado na WWID para seu correspondente nome `/dev/sd` no sistema. Estes nomes são persistentes nas mudanças de caminho, e são consistentes ao acessar o dispositivo a partir de diferentes sistemas.

Quando o recurso `user_friendly_names` do DM Multipath é usado, o WWID é mapeado para um nome do formulário `/dev/mapper/mpathN`. Por padrão, este mapeamento é mantido no arquivo `/etc/multipath/bindings`. Estes `mpathN` são persistentes, desde que esse arquivo seja mantido.



#### IMPORTANTE

Se você usa `user_friendly_names`, então são necessários passos adicionais para obter nomes consistentes em um cluster.

## 9.5. LIMITAÇÕES DA CONVENÇÃO DE NOMENCLATURA DE DISPOSITIVOS UDEV

A seguir estão algumas limitações da convenção de nomenclatura `udev`:

- É possível que o dispositivo não esteja acessível no momento em que a consulta é realizada porque o mecanismo `udev` pode confiar na capacidade de consultar o dispositivo de armazenamento quando as regras do `udev` são processadas para um evento `udev`. Isto é mais provável que ocorra com dispositivos de armazenamento Fibre Channel, iSCSI ou FCoE quando o dispositivo não está localizado no chassi do servidor.
- O kernel pode enviar eventos `udev` a qualquer momento, fazendo com que as regras sejam processadas e possivelmente fazendo com que os links `/dev/disk/by-*/` sejam removidos se o dispositivo não for acessível.

- Pode haver um atraso entre quando o evento **udev** é gerado e quando ele é processado, como quando um grande número de dispositivos é detectado e o serviço **udev** de espaço do usuário leva algum tempo para processar as regras para cada um deles. Isto pode causar um atraso entre quando o kernel detecta o dispositivo e quando os nomes **/dev/disk/by-\*/** estão disponíveis.
- Programas externos como o **blkid** invocado pelas regras podem abrir o dispositivo por um breve período de tempo, tornando o dispositivo inacessível para outros usos.
- Os nomes dos dispositivos gerenciados pelo mecanismo **udev** em **/dev/disco/** podem mudar entre os principais lançamentos, exigindo a atualização dos links.

## 9.6. LISTAGEM DE ATRIBUTOS DE NOMEAÇÃO PERSISTENTES

Este procedimento descreve como descobrir os atributos persistentes de nomes de dispositivos de armazenamento não persistentes.

### Procedimento

- Para listar os atributos UUID e Label, use o utilitário **lsblk**:

```
$ lsblk --fs storage-device
```

Por exemplo:

#### Exemplo 9.4. Visualização da UUID e do rótulo de um sistema de arquivo

```
$ lsblk --fs /dev/sda1
```

NAME	FSTYPE	LABEL	UUID	MOUNTPOINT
sda1	xf	Boot	afa5d5e3-9050-48c3-acc1-bb30095f3dc4	/boot

- Para listar o atributo PARTUUID, use o utilitário **lsblk** com a opção **--output PARTUIDO**:

```
$ lsblk -- saída PARTUIDO
```

Por exemplo:

#### Exemplo 9.5. Visualização do atributo PARTUIDO de uma partição

```
$ lsblk --output +PARTUIDO /dev/sda1
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT	PARTUIDO
sda1	8:1	0	512M	0	part	/boot	4cd1448a-01

- Para listar o atributo WWID, examine os alvos dos links simbólicos no diretório **/dev/disk/by-id/**.  
Por exemplo:

#### Exemplo 9.6. Visualização da WWID de todos os dispositivos de armazenamento no sistema

```

$ file /dev/disk/by-id/*

/dev/disk/by-id/ata-QEMU_HARDDISK_QM00001
symbolic link to ../../sda
/dev/disk/by-id/ata-QEMU_HARDDISK_QM00001-part1
symbolic link to ../../sda1
/dev/disk/by-id/ata-QEMU_HARDDISK_QM00001-part2
symbolic link to ../../sda2
/dev/disk/by-id/dm-name-rhel_rhel8-root
symbolic link to ../../dm-0
/dev/disk/by-id/dm-name-rhel_rhel8-swap
symbolic link to ../../dm-1
/dev/disk/by-id/dm-uuid-LVM-
QIWtEHtXGobe5bewlIUdIVKOz5ofkgFhP0RMFsNyySVihqEI2cWWbR7MjXJoID6g
symbolic link to ../../dm-1
/dev/disk/by-id/dm-uuid-LVM-
QIWtEHtXGobe5bewlIUdIVKOz5ofkgFhXqH2M45hD2H9nAf2qfWSrIRLhzfMyOKd
symbolic link to ../../dm-0
/dev/disk/by-id/lvm-pv-uuid-atlr2Y-vuMo-ueoH-CpMG-4JuH-AhEF-wu4QQm
symbolic link to ../../sda2

```

## 9.7. MODIFICAÇÃO DE ATRIBUTOS DE NOMEAÇÃO PERSISTENTES

Este procedimento descreve como alterar o atributo de nomeação persistente UUID ou Label de um sistema de arquivo.



### NOTA

A mudança dos atributos **udev** acontece em segundo plano e pode demorar muito tempo. O comando **udevadm settle** espera até que a mudança seja totalmente registrada, o que garante que seu próximo comando será capaz de utilizar o novo atributo corretamente.

Nos seguintes comandos:

- Substitua *new-uuid* com a UUID que você deseja definir; por exemplo, **1cdfbc07-1c90-4984-b5ec-f61943f5ea50**. Você pode gerar uma UUID usando o comando **uuidgen**.
- Substitua *new-label* com um rótulo; por exemplo, **backup\_data**.

### Pré-requisitos

- Se você estiver modificando os atributos de um sistema de arquivo XFS, desmonte-o primeiro.

### Procedimento

- Para alterar os atributos UUID ou Label de um sistema de arquivos **XFS**, use o utilitário **xfs\_admin**:

```

# xfs_admin -U new-uuid -L new-label storage-device
# udevadm settle

```

- Para alterar os atributos UUID ou Label de um sistema de arquivos **ext4**, **ext3**, ou **ext2**, use o utilitário **tune2fs**:

```
# tune2fs -U new-uuid -L new-label storage-device  
# udevadm settle
```

- Para alterar os atributos UUID ou Label de um volume swap, use o utilitário **swaplabel**:

```
# swaplabel --uuid new-uuid --label new-label swap-device  
# udevadm settle
```

## CAPÍTULO 10. COMEÇANDO COM AS DIVISÓRIAS

Como administrador do sistema, você pode usar os seguintes procedimentos para criar, excluir e modificar vários tipos de partições de disco.

Para uma visão geral das vantagens e desvantagens de usar divisórias em dispositivos de blocos, veja o seguinte artigo da KBase: <https://access.redhat.com/solutions/163853>.

### 10.1. VISUALIZAÇÃO DA MESA DIVISÓRIA

Como administrador de sistema, você pode exibir a tabela de partição de um dispositivo de bloco para ver o layout da partição e detalhes sobre as partições individuais.

#### 10.1.1. Visualizando a mesa divisória com separação

Este procedimento descreve como visualizar a tabela de partição em um dispositivo de bloco usando o utilitário **parted**.

##### Procedimento

1. Inicie o shell interativo **parted**:

```
# Separado block-device
```

- Substitua *block-device* com o caminho para o dispositivo que você deseja examinar: por exemplo, **/dev/sda**.

2. Veja a tabela de partição:

```
(dividido) impressão
```

3. Opcionalmente, use o seguinte comando para mudar para outro dispositivo que você queira examinar a seguir:

```
(separados) selecione block-device
```

##### Recursos adicionais

- A página do homem **parted(8)**.

#### 10.1.2. Exemplo de saída de **parted print**

Esta seção fornece um exemplo de saída do comando **print** na shell **parted** e descreve os campos na saída.

##### Exemplo 10.1. Saída do comando **print**

```
Model: ATA SAMSUNG MZNLN256 (scsi)
Disk /dev/sda: 256GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:
```

Number	Start	End	Size	Type	File system	Flags
1	1049kB	269MB	268MB	primary	xf	boot
2	269MB	34.6GB	34.4GB	primary		
3	34.6GB	45.4GB	10.7GB	primary		
4	45.4GB	256GB	211GB	extended		
5	45.4GB	256GB	211GB	logical		

A seguir, uma descrição dos campos:

#### **Model: ATA SAMSUNG MZNLN256 (scsi)**

O tipo de disco, fabricante, número do modelo e interface.

#### **Disk /dev/sda: 256GB**

O caminho do arquivo para o dispositivo do bloco e a capacidade de armazenamento.

#### **Partition Table: msdos**

O tipo de etiqueta do disco.

#### **Number**

O número da divisória. Por exemplo, a partição com número menor 1 corresponde a **/dev/sda1**.

#### **Start e End**

A localização no dispositivo onde começa e termina a divisória.

#### **Type**

Os tipos válidos são metadados, livres, primários, estendidos ou lógicos.

#### **File system**

O tipo de sistema de arquivo. Se o campo **File system** de um dispositivo não mostrar nenhum valor, isto significa que seu tipo de sistema de arquivo é desconhecido. O utilitário **parted** não consegue reconhecer o sistema de arquivo em dispositivos criptografados.

#### **Flags**

Relaciona as bandeiras definidas para a divisória. As bandeiras disponíveis são **boot**, **root**, **swap**, **hidden**, **raid**, **lvm**, ou **lba**.

## 10.2. CRIAÇÃO DE UMA TABELA DE PARTIÇÃO EM UM DISCO

Como administrador de sistema, você pode formatar um dispositivo de bloco com diferentes tipos de tabelas de partição para permitir o uso de partições no dispositivo.



### **ATENÇÃO**

A formatação de um dispositivo de bloco com uma tabela de partição elimina todos os dados armazenados no dispositivo.

### 10.2.1. Considerações antes de modificar as partições em um disco

Esta seção lista os pontos-chave a serem considerados antes de criar, remover ou redimensionar as divisórias.



## NOTA

Esta seção não cobre a tabela de partição DASD, que é específica para a arquitetura IBM Z. Para informações sobre o DASD, veja:

- [Configuração de uma instância Linux na IBM Z](#)
- O [que você deve saber sobre](#) o artigo do [DASD](#) no Centro de Conhecimento IBM

### O número máximo de divisórias

O número de divisórias em um dispositivo é limitado pelo tipo da tabela de divisórias:

- Em um dispositivo formatado com a tabela de partição **Master Boot Record (MBR)**, você pode ter qualquer um dos dois:
  - Até quatro partições primárias, ou
  - Até três divisórias primárias, uma divisória estendida, e múltiplas divisórias lógicas dentro da estendida.
- Em um dispositivo formatado com o **GUID Partition Table (GPT)** o número máximo de divisórias é de 128. Enquanto a especificação GPT permite mais partições aumentando a área reservada para a tabela de partição, a prática comum usada pelo utilitário **parted** é limitá-la a uma área suficiente para 128 partições.



## NOTA

A Red Hat recomenda que, a menos que você tenha uma razão para fazer o contrário, você deve *at least* criar as seguintes partições: **swap**, **/boot/**, e **/** (raiz).

### O tamanho máximo de uma divisória

O tamanho de uma divisória em um dispositivo é limitado pelo tipo da mesa divisória:

- Em um dispositivo formatado com a tabela de partição **Master Boot Record (MBR)**, o tamanho máximo é 2TiB.
- Em um dispositivo formatado com o **GUID Partition Table (GPT)** o tamanho máximo é 8ZiB.

Se você quiser criar uma partição maior que 2TiB, o disco deve ser formatado com GPT.

### Alinhamento de tamanhos

O utilitário **parted** permite especificar o tamanho da partição usando vários sufixos diferentes:

#### MiB, GiB, ou TiB

Tamanho expresso em poderes de 2.

- O ponto de partida da divisória é alinhado ao setor exato especificado por tamanho.
- O ponto final é alinhado com o tamanho especificado menos 1 setor.

#### MB, GB, ou TB

Tamanho expresso em poderes de 10.

O ponto inicial e final é alinhado dentro de uma metade da unidade especificada: por exemplo,  $\pm 500\text{KB}$  ao utilizar o sufixo MB.

## 10.2.2. Comparação dos tipos de mesas divisórias

Esta seção compara as propriedades de diferentes tipos de tabelas de partição que você pode criar em um dispositivo de bloco.

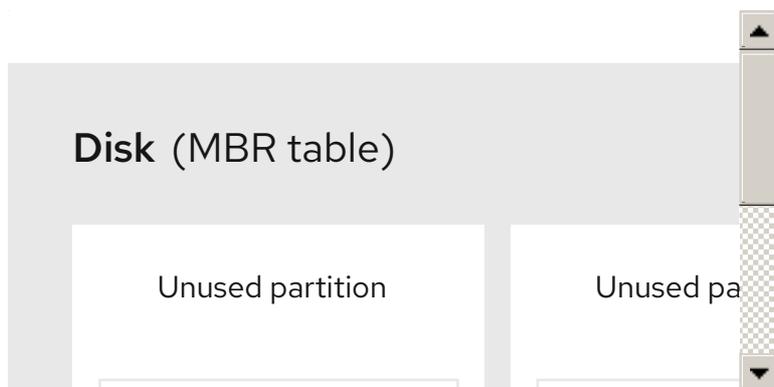
Tabela 10.1. Tipos de mesas divisórias

Mesa divisória	Número máximo de divisórias	Tamanho máximo da divisória
Registro de Bota Principal (MBR)	4 primários, ou 3 primários e 12 lógicos dentro de uma partição estendida	2TiB
Tabela de partição do GUID (GPT)	128	8ZiB

## 10.2.3. Partições de disco MBR

Os diagramas deste capítulo mostram a tabela de partição como sendo separada do disco real. No entanto, isto não é totalmente exato. Na realidade, a tabela de partição é armazenada logo no início do disco, antes de qualquer sistema de arquivos ou dados do usuário, mas, para maior clareza, eles estão separados nos diagramas a seguir.

Figura 10.1. Disco com mesa divisória MBR



Como mostra o diagrama anterior, a tabela de partições está dividida em quatro seções de quatro partições primárias. Uma partição primária é uma partição em um disco rígido que pode conter apenas um disco lógico (ou seção). Cada seção pode conter as informações necessárias para definir uma única partição, o que significa que a tabela de partição não pode definir mais do que quatro partições.

Cada entrada da tabela de partição contém várias características importantes da partição:

- Os pontos no disco onde começa e termina a partição.
- Se a divisória é **active**. Apenas uma partição pode ser sinalizada como **active**.
- O tipo da divisória.

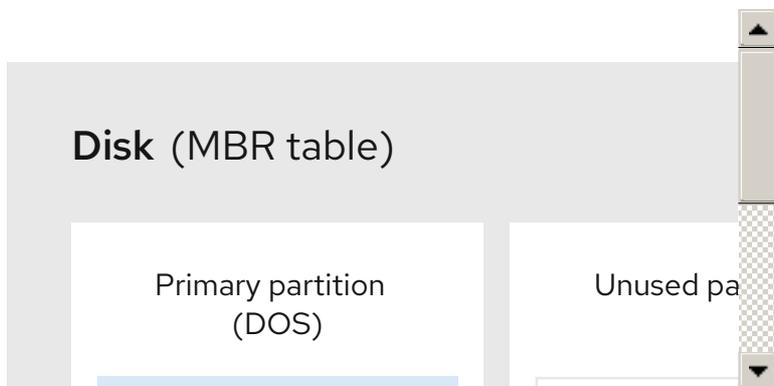
Os pontos inicial e final definem o tamanho e a localização da partição no disco. A bandeira "ativa" é usada por alguns sistemas operacionais carregadores de inicialização. Em outras palavras, o sistema operacional na partição que está marcado como "ativo" é inicializado, neste caso.

O tipo é um número que identifica o uso antecipado da divisória. Alguns sistemas operacionais usam o

tipo de partição para denotar um tipo específico de sistema de arquivo, para marcar a partição como sendo associada a um sistema operacional específico, para indicar que a partição contém um sistema operacional inicializável, ou alguma combinação dos três.

O diagrama a seguir mostra um exemplo de uma unidade com uma única partição:

Figura 10.2. Disco com uma única divisória



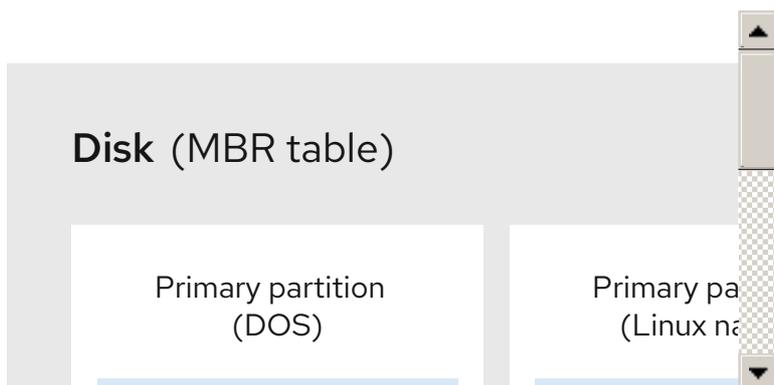
A partição única neste exemplo está etiquetada como **DOS**. Esta etiqueta mostra o tipo de partição, sendo **DOS** uma das mais comuns.

#### 10.2.4. Partições MBR estendidas

Caso quatro divisórias sejam insuficientes para suas necessidades, você pode usar divisórias estendidas para criar divisórias adicionais. Você pode fazer isso definindo o tipo de partição como "Estendida".

Uma partição estendida é como uma unidade de disco em seu próprio direito – ela tem sua própria tabela de partições, que aponta para uma ou mais partições (agora chamadas partições lógicas, ao contrário das quatro partições primárias), contidas inteiramente dentro da própria partição estendida. O diagrama a seguir mostra um drive de disco com uma partição primária e uma partição estendida contendo duas partições lógicas (juntamente com algum espaço livre não particionado):

Figura 10.3. Disco com uma partição MBR primária e uma MBR estendida



Como este número implica, há uma diferença entre as partições primárias e lógicas – só pode haver até quatro partições primárias e estendidas, mas não há um limite fixo para o número de partições lógicas que podem existir. Entretanto, devido à forma como as partições são acessadas no Linux, não podem ser definidas mais de 15 partições lógicas em um único drive de disco.

#### 10.2.5. Tipos de divisórias MBR

A tabela abaixo mostra uma lista de alguns dos tipos de partição MBR comumente usados e números hexadecimais usados para representá-los.

Tabela 10.2. Tipos de divisórias MBR

MBR partition type	Value	MBR partition type	Value
Vazio	00	Novell Netware 386	65
DOS 12-bit FAT	01	PIC/IX	75
Raiz XENIX	02	Velho MINIX	80
XENIX usr	03	Linux/MINUX	81
DOS 16 bits $\leq$ 32M	04	Troca de Linux	82
Estendido	05	Linux nativo	83
DOS 16-bit $\geq$ 32	06	Linux estendido	85
OS/2 HPFS	07	Amoeba	93
AIX	08	Amoeba BBT	94
AIX inicializável	09	BSD/386	a5
Gerenciador de Boot OS/2	0a	OpenBSD	a6
Win95 FAT32	0b	NEXTSTEP	a7
Win95 FAT32 (LBA)	0c	BSDI fs	b7
Win95 FAT16 (LBA)	0e	Troca de BSDI	b8
Win95 Estendido (LBA)	0f	Syrinx	c7
Venix 80286	40	CP/M	db
Novell	51	Acesso DOS	e1
Bota PRep	41	DOS R/O	e3
GNU HURD	63	DOS secundário	f2
Novell Netware 286	64	BBT	ff

### 10.2.6. Tabela de partição do GUID

A Tabela de Partição GUID (GPT) é um esquema de partição baseado no uso de um Identificador Único

Global (GUID). O GPT foi desenvolvido para lidar com as limitações da tabela de partição MBR, especialmente com o espaço máximo de armazenamento endereçável limitado de um disco. Ao contrário do MBR, que é incapaz de endereçar armazenamento maior que 2 TiB (equivalente a aproximadamente 2,2 TB), o GPT é usado com discos rígidos maiores que este; o tamanho máximo do disco endereçável é 2,2 ZiB. Além disso, o GPT, por padrão, suporta a criação de até 128 partições primárias. Este número poderia ser ampliado alocando mais espaço para a tabela de partições.



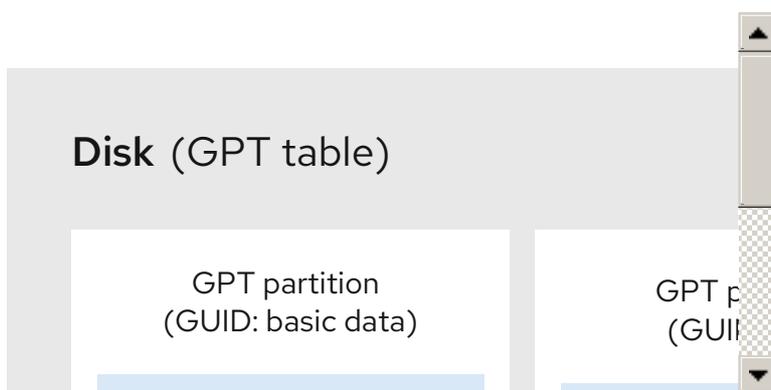
## NOTA

Um GPT tem tipos de divisórias baseadas em GUIDs. Note que certas partições requerem um GUID específico. Por exemplo, a partição do sistema para carregadores de inicialização EFI requer o GUID **C12A7328-F81F-11D2-BA4B-00A0C93EC93B**.

Os discos GPT usam endereçamento lógico de blocos (LBA) e o layout da partição é o seguinte:

- Para preservar a compatibilidade com os discos MBR, o primeiro setor (LBA 0) do GPT é reservado para os dados MBR e é chamado de "MBR de proteção".
- O cabeçalho primário GPT começa no segundo bloco lógico (LBA 1) do dispositivo. O cabeçalho contém o GUID do disco, a localização da tabela de partição primária, a localização do cabeçalho GPT secundário, e os checksums CRC32 de si mesmo, e a tabela de partição primária. Ele também especifica o número de entradas de partição na tabela.
- O GPT primário inclui, por padrão, 128 entradas de partição, cada uma com um tamanho de entrada de 128 bytes, seu tipo de partição GUID e GUID de partição única.
- O GPT secundário é idêntico ao GPT primário. Ele é usado principalmente como uma tabela de reserva para recuperação no caso da tabela de partição primária estar corrompida.
- O cabeçalho secundário GPT está localizado no último setor lógico do disco e pode ser usado para recuperar informações GPT no caso do cabeçalho primário estar corrompido. Ele contém o GUID do disco, a localização da tabela de partição secundária e do cabeçalho GPT primário, os checksums CRC32 de si mesmo e da tabela de partição secundária, e o número de possíveis entradas de partição.

Figura 10.4. Disco com uma tabela de partição GUID



## IMPORTANTE

Deve haver uma partição de inicialização BIOS para que o carregador de inicialização seja instalado com sucesso em um disco que contenha uma GPT (GUID Partition Table). Isto inclui discos inicializados por **Anaconda**. Se o disco já contém uma partição de inicialização da BIOS, ela pode ser reutilizada.

### 10.2.7. Criação de uma tabela de partição em um disco com partição

Este procedimento descreve como formatar um dispositivo de bloco com uma tabela de partição usando o utilitário **parted**.

#### Procedimento

1. Inicie o shell interativo **parted**:

```
# Separado block-device
```

- Substitua *block-device* com o caminho para o dispositivo onde você deseja criar uma tabela de partição: por exemplo, **/dev/sda**.

2. Determinar se já existe uma tabela de partição no dispositivo:

```
(dividido) impressão
```

Se o dispositivo já contém partições, elas serão apagadas nas próximas etapas.

3. Criar a nova mesa divisória:

```
(separado) mklabel table-type
```

- Substitua *table-type* com o tipo de mesa divisória prevista:
  - **msdos** para MBR
  - **gpt** para GPT

#### Exemplo 10.2. Criação de uma tabela GPT

Por exemplo, para criar uma tabela GPT sobre o disco, use:

```
(separado) mklabel gpt
```

As mudanças começam a acontecer assim que você entra neste comando, portanto, revise-o antes de executá-lo.

4. Veja a tabela de partição para confirmar que a tabela de partição existe:

```
(dividido) impressão
```

5. Saia da casca **parted**:

```
(separado) desistir
```

#### Recursos adicionais

- A página do homem **parted(8)**.

#### Próximos passos

- Criar divisórias no dispositivo. Veja [Seção 10.3, “Criando uma divisória”](#) para detalhes.

## 10.3. CRIANDO UMA DIVISÓRIA

Como administrador do sistema, você pode criar novas partições em um disco.

### 10.3.1. Considerações antes de modificar as partições em um disco

Esta seção lista os pontos-chave a serem considerados antes de criar, remover ou redimensionar as divisórias.



#### NOTA

Esta seção não cobre a tabela de partição DASD, que é específica para a arquitetura IBM Z. Para informações sobre o DASD, veja:

- [Configuração de uma instância Linux na IBM Z](#)
- [O que você deve saber sobre](#) o artigo do [DASD](#) no Centro de Conhecimento IBM

#### O número máximo de divisórias

O número de divisórias em um dispositivo é limitado pelo tipo da tabela de divisórias:

- Em um dispositivo formatado com a tabela de partição **Master Boot Record (MBR)**, você pode ter qualquer um dos dois:
  - Até quatro partições primárias, ou
  - Até três divisórias primárias, uma divisória estendida, e múltiplas divisórias lógicas dentro da estendida.
- Em um dispositivo formatado com o **GUID Partition Table (GPT)** o número máximo de divisórias é de 128. Enquanto a especificação GPT permite mais partições aumentando a área reservada para a tabela de partição, a prática comum usada pelo utilitário **parted** é limitá-la a uma área suficiente para 128 partições.



#### NOTA

A Red Hat recomenda que, a menos que você tenha uma razão para fazer o contrário, você deve *at least* criar as seguintes partições: **swap**, **/boot/**, e **/** (raiz).

#### O tamanho máximo de uma divisória

O tamanho de uma divisória em um dispositivo é limitado pelo tipo da mesa divisória:

- Em um dispositivo formatado com a tabela de partição **Master Boot Record (MBR)** o tamanho máximo é 2TiB.
- Em um dispositivo formatado com o **GUID Partition Table (GPT)** o tamanho máximo é 8ZiB.

Se você quiser criar uma partição maior que 2TiB, o disco deve ser formatado com GPT.

#### Alinhamento de tamanhos

O utilitário **parted** permite especificar o tamanho da partição usando vários sufixos diferentes:

#### MiB, GiB, ou TiB

Tamanho expresso em poderes de 2.

- O ponto de partida da divisória é alinhado ao setor exato especificado por tamanho.
- O ponto final é alinhado com o tamanho especificado menos 1 setor.

### MB, GB, ou TB

Tamanho expresso em poderes de 10.

O ponto inicial e final é alinhado dentro de uma metade da unidade especificada: por exemplo,  $\pm 500\text{KB}$  ao utilizar o sufixo MB.

## 10.3.2. Tipos de partição

Esta seção descreve diferentes atributos que especificam o tipo de uma partição.

### Tipos de partição ou bandeiras

O tipo de divisória, ou bandeira, é usado por um sistema em funcionamento apenas raramente.

Entretanto, o tipo de partição é importante para os geradores em funcionamento, tais como **systemd-gpt-auto-generator**, que utilizam o tipo de partição para, por exemplo, identificar e montar automaticamente os dispositivos.

- O utilitário **parted** fornece algum controle dos tipos de partição através do mapeamento do tipo de partição para *flags*. O utilitário **parted** pode controlar apenas certos tipos de partição: por exemplo LVM, swap, ou RAID.
- O utilitário **fdisk** suporta toda a gama de tipos de divisórias especificando códigos hexadecimais.

### Tipo de sistema de arquivo de partição

O utilitário **parted** aceita opcionalmente um argumento de tipo de sistema de arquivo ao criar uma partição. O valor é usado para:

- Coloque as bandeiras de partição no MBR, ou
- Defina a partição tipo UUID no GPT. Por exemplo, os tipos de sistema de arquivo **swap**, **fat**, ou **hfs** definem diferentes GUIDs. O valor padrão é o GUID de dados Linux.

O argumento não modifica de forma alguma o sistema de arquivo na partição. Ele apenas diferencia entre as bandeiras ou GUIDs suportadas.

Os seguintes tipos de sistemas de arquivo são suportados:

- **xf**s
- **ext2**
- **ext3**
- **ext4**
- **fat16**
- **fat32**
- **hfs**

- **hfs**
- **linux-swaps**
- **ntfs**
- **reiserfs**

### 10.3.3. Esquema de nomenclatura das partições

O Red Hat Enterprise Linux usa um esquema de nomes baseado em arquivos, com nomes de arquivos na forma de **/dev/xyN**.

Os nomes dos dispositivos e das divisórias consistem na seguinte estrutura:

**/dev/**

Este é o nome do diretório no qual estão localizados todos os arquivos do dispositivo. Como as partições são colocadas em discos rígidos, e os discos rígidos são dispositivos, os arquivos que representam todas as partições possíveis estão localizados em **/dev**.

**xx**

As duas primeiras letras do nome das divisórias indicam o tipo de dispositivo em que se encontra a divisória, geralmente **sd**.

**y**

Esta letra indica em qual dispositivo a divisória está ligada. Por exemplo, **/dev/sda** para o primeiro disco rígido, **/dev/sdb** para o segundo, e assim por diante. Em sistemas com mais de 26 drives, é possível usar mais letras. Por exemplo, **/dev/sdaa1**.

**N**

A letra final indica o número que representa a partição. As quatro primeiras partições (primárias ou ampliadas) são numeradas **1** até **4**. As partições lógicas começam em **5**. Por exemplo, **/dev/sda3** é a terceira partição primária ou estendida no primeiro disco rígido, e **/dev/sdb6** é a segunda partição lógica no segundo disco rígido. A numeração das partições do disco se aplica somente às tabelas de partições MBR. Observe que **N** nem sempre significa partição.



#### NOTA

Mesmo que o Red Hat Enterprise Linux possa identificar e consultar os tipos de partições de disco *all*, ele pode não ser capaz de ler o sistema de arquivo e, portanto, acessar os dados armazenados em cada tipo de partição. Entretanto, em muitos casos, é possível acessar com sucesso dados em uma partição dedicada a outro sistema operacional.

### 10.3.4. Pontos de montagem e partições de disco

No Red Hat Enterprise Linux, cada partição é usada para formar parte do armazenamento necessário para suportar um único conjunto de arquivos e diretórios. Isto é feito usando o processo conhecido como *mounting*, que associa uma partição a um diretório. A montagem de uma partição torna seu armazenamento disponível a partir do diretório especificado, conhecido como *mount point*.

Por exemplo, se a partição **/dev/sda5** estiver montada em **/usr/**, isso significaria que todos os arquivos e diretórios sob **/usr/** residem fisicamente em **/dev/sda5**. Assim, o arquivo **/usr/share/doc/FAQ/txt/Linux-FAQ** seria armazenado em **/dev/sda5**, enquanto o arquivo **/etc/gdm/custom.conf** não o seria.

Continuando o exemplo, também é possível que um ou mais diretórios abaixo de **/usr/** sejam pontos de montagem para outras partições. Por exemplo, uma partição **/dev/sda7** poderia ser montada em **/usr/local**, o que significa que **/usr/local/man/whatis** residiria então em **/dev/sda7** em vez de **/dev/sda5**.

### 10.3.5. Criação de uma divisória com separação

Este procedimento descreve como criar uma nova partição em um dispositivo de bloco usando o utilitário **parted**.

#### Pré-requisitos

- Há uma tabela de partição no disco. Para detalhes sobre como formatar o disco, veja [Seção 10.2, “Criação de uma tabela de partição em um disco”](#).
- Se a partição que você deseja criar for maior que 2TiB, o disco deve ser formatado com a Tabela de Partição GUID (GPT).

#### Procedimento

1. Inicie o shell interativo **parted**:

```
# Separado block-device
```

- Substitua *block-device* com o caminho para o dispositivo onde você quer criar uma partição: por exemplo, **/dev/sda**.

2. Veja a tabela de partição atual para determinar se há espaço livre suficiente:

```
(dividido) impressão
```

- Se não houver espaço livre suficiente, você pode redimensionar uma divisória existente. Para mais informações, consulte [Seção 10.5, “Redimensionamento de uma divisória”](#).
- A partir da tabela de partição, determine:
  - Os pontos de início e fim da nova divisória
  - No MBR, que tipo de partição deve ser.

3. Criar a nova divisória:

```
(separado) mkpart part-type name fs-type start end
```

- Substitua *part-type* com **primary**, **logical**, ou **extended** com base no que você decidiu a partir da tabela de partição. Isto se aplica somente à tabela de partição MBR.
- Substitua *name* com um nome de partição arbitrária. Isto é necessário para as tabelas de partição GPT.
- Substitua *fs-type* com qualquer um de **xfs**, **ext2**, **ext3**, **ext4**, **fat16**, **fat32**, **hfs**, **hfs**, **linux-swamp**, **ntfs**, ou **reiserfs**. O *fs-type* parâmetro é opcional. Note que **parted** não cria o sistema de arquivo na partição.

- Substitua *start* e *end* com os tamanhos que determinam os pontos inicial e final da partição, contando desde o início do disco. Pode-se usar sufixos de tamanho, como **512MiB**, **20GiB**, ou **1.5TiB**. Os megabytes de tamanho padrão.

### Exemplo 10.3. Criação de uma pequena partição primária

Por exemplo, para criar uma partição primária de 1024MiB até 2048MiB em uma tabela MBR, use:

```
(dividido) mkpart primário 1024MiB 2048MiB
```

As mudanças começam a acontecer assim que você entra neste comando, portanto, revise-o antes de executá-lo.

4. Veja a tabela de partição para confirmar que a partição criada está na tabela de partição com o tipo, tipo de sistema de arquivo e tamanho corretos:

```
(dividido) impressão
```

5. Saia da casca **parted**:

```
(separado) desistir
```

6. Use o seguinte comando para esperar que o sistema registre o novo nó de dispositivo:

```
# udevadm assentar
```

7. Verificar se o núcleo reconhece a nova partição:

```
# gato /proc/partições
```

### Recursos adicionais

- A página do homem **parted(8)**.

### 10.3.6. Definição de um tipo de divisória com fdisk

Este procedimento descreve como definir um tipo de partição, ou bandeira, usando o utilitário **fdisk**.

#### Pré-requisitos

- Há uma partição no disco.

#### Procedimento

1. Inicie o shell interativo **fdisk**:

```
# fdisk block-device
```

- Substitua *block-device* com o caminho para o dispositivo onde você deseja definir um tipo de partição: por exemplo, **/dev/sda**.

2. Veja a tabela de partição atual para determinar o número da partição menor:

```
Comando (m de ajuda) print
```

Você pode ver o tipo de partição atual na coluna **Type** e sua identificação de tipo correspondente na coluna **Id**.

3. Digite o comando tipo de partição e selecione uma partição usando seu número menor:

```
Command (m for help): type
Partition number (1,2,3 default 3): 2
```

4. Opcionalmente, liste os códigos hexadecimais disponíveis:

```
Código hexadecimal (tipo L para listar todos os códigos) L
```

5. Defina o tipo de divisória:

```
Código hexadecimal (tipo L para listar todos os códigos) 8e
```

6. Escreva suas mudanças e saia da casca **fdisk**:

```
Command (m for help): write
The partition table has been altered.
Syncing disks.
```

7. Verifique suas mudanças:

```
# fdisk --lista block-device
```

## 10.4. REMOÇÃO DE UMA DIVISÓRIA

Como administrador do sistema, você pode remover uma partição de disco que não é mais utilizada para liberar espaço em disco.



### ATENÇÃO

A remoção de uma partição apaga todos os dados armazenados na partição.

### 10.4.1. Considerações antes de modificar as partições em um disco

Esta seção lista os pontos-chave a serem considerados antes de criar, remover ou redimensionar as divisórias.



## NOTA

Esta seção não cobre a tabela de partição DASD, que é específica para a arquitetura IBM Z. Para informações sobre o DASD, veja:

- [Configuração de uma instância Linux na IBM Z](#)
- [O que você deve saber sobre](#) o artigo do [DASD](#) no Centro de Conhecimento IBM

### O número máximo de divisórias

O número de divisórias em um dispositivo é limitado pelo tipo da tabela de divisórias:

- Em um dispositivo formatado com a tabela de partição **Master Boot Record (MBR)**, você pode ter qualquer um dos dois:
  - Até quatro partições primárias, ou
  - Até três divisórias primárias, uma divisória estendida, e múltiplas divisórias lógicas dentro da estendida.
- Em um dispositivo formatado com o **GUID Partition Table (GPT)** o número máximo de divisórias é de 128. Enquanto a especificação GPT permite mais partições aumentando a área reservada para a tabela de partição, a prática comum usada pelo utilitário **parted** é limitá-la a uma área suficiente para 128 partições.



## NOTA

A Red Hat recomenda que, a menos que você tenha uma razão para fazer o contrário, você deve *at least* criar as seguintes partições: **swap**, **/boot/**, e **/** (raiz).

### O tamanho máximo de uma divisória

O tamanho de uma divisória em um dispositivo é limitado pelo tipo da mesa divisória:

- Em um dispositivo formatado com a tabela de partição **Master Boot Record (MBR)**, o tamanho máximo é 2TiB.
- Em um dispositivo formatado com o **GUID Partition Table (GPT)** o tamanho máximo é 8ZiB.

Se você quiser criar uma partição maior que 2TiB, o disco deve ser formatado com GPT.

### Alinhamento de tamanhos

O utilitário **parted** permite especificar o tamanho da partição usando vários sufixos diferentes:

#### MiB, GiB, ou TiB

Tamanho expresso em poderes de 2.

- O ponto de partida da divisória é alinhado ao setor exato especificado por tamanho.
- O ponto final é alinhado com o tamanho especificado menos 1 setor.

#### MB, GB, ou TB

Tamanho expresso em poderes de 10.

O ponto inicial e final é alinhado dentro de uma metade da unidade especificada: por exemplo,  $\pm 500\text{KB}$  ao utilizar o sufixo MB.

## 10.4.2. Remoção de uma divisória com separação

Este procedimento descreve como remover uma partição de disco usando o utilitário **parted**.

### Procedimento

1. Inicie o shell interativo **parted**:

```
# Separado block-device
```

- Substitua *block-device* com o caminho para o dispositivo onde você quer remover uma partição: por exemplo, **/dev/sda**.

2. Veja a tabela de partição atual para determinar o número menor da partição a ser removida:

```
(dividido) impressão
```

3. Remover a divisória:

```
(separados) rm minor-number
```

- Substitua *minor-number* com o número menor da partição que você deseja remover: por exemplo, **3**.

As mudanças começam a acontecer assim que você entra neste comando, portanto, revise-o antes de executá-lo.

4. Confirmar que a divisória é removida da mesa divisória:

```
(dividido) impressão
```

5. Saia da casca **parted**:

```
(separado) desistir
```

6. Verificar se o núcleo sabe que a partição foi removida:

```
# gato /proc/partições
```

7. Remova a partição do arquivo **/etc/fstab** se ela estiver presente. Encontre a linha que declara a partição removida, e remova-a do arquivo.

8. Regenere as unidades de montagem para que seu sistema registre a nova configuração **/etc/fstab**:

```
# systemctl daemon-reload
```

9. Se você tiver excluído uma partição swap ou removido pedaços de LVM, remova todas as referências à partição da linha de comando do kernel no arquivo **/etc/default/grub** e regenere a configuração do GRUB:

- Em um sistema baseado na BIOS:

```
# grub2-mkconfig --output=/etc/grub2.cfg
```

- Em um sistema baseado na UEFI:

```
# grub2-mkconfig --output=/etc/grub2-efi.cfg
```

10. Para registrar as mudanças no sistema de inicialização inicial, reconstrua o sistema de arquivos **initramfs**:

```
# dracut --force --verbose
```

### Recursos adicionais

- A página do homem **parted(8)**

## 10.5. REDIMENSIONAMENTO DE UMA DIVISÓRIA

Como administrador de sistema, você pode estender uma partição para utilizar o espaço não utilizado em disco, ou encolher uma partição para utilizar sua capacidade para diferentes propósitos.

### 10.5.1. Considerações antes de modificar as partições em um disco

Esta seção lista os pontos-chave a serem considerados antes de criar, remover ou redimensionar as divisórias.



#### NOTA

Esta seção não cobre a tabela de partição DASD, que é específica para a arquitetura IBM Z. Para informações sobre o DASD, veja:

- [Configuração de uma instância Linux na IBM Z](#)
- [O que você deve saber sobre](#) o artigo do [DASD](#) no Centro de Conhecimento IBM

#### O número máximo de divisórias

O número de divisórias em um dispositivo é limitado pelo tipo da tabela de divisórias:

- Em um dispositivo formatado com a tabela de partição **Master Boot Record (MBR)**, você pode ter qualquer um dos dois:
  - Até quatro partições primárias, ou
  - Até três divisórias primárias, uma divisória estendida, e múltiplas divisórias lógicas dentro da estendida.
- Em um dispositivo formatado com o **GUID Partition Table (GPT)** o número máximo de divisórias é de 128. Enquanto a especificação GPT permite mais partições aumentando a área reservada para a tabela de partição, a prática comum usada pelo utilitário **parted** é limitá-la a uma área suficiente para 128 partições.



#### NOTA

A Red Hat recomenda que, a menos que você tenha uma razão para fazer o contrário, você deve *at least* criar as seguintes partições: **swap**, **/boot/**, e **/** (raiz).

### O tamanho máximo de uma divisória

O tamanho de uma divisória em um dispositivo é limitado pelo tipo da mesa divisória:

- Em um dispositivo formatado com a tabela de partição **Master Boot Record (MBR)**, o tamanho máximo é 2TiB.
- Em um dispositivo formatado com o **GUID Partition Table (GPT)** o tamanho máximo é 8ZiB.

Se você quiser criar uma partição maior que 2TiB, o disco deve ser formatado com GPT.

### Alinhamento de tamanhos

O utilitário **parted** permite especificar o tamanho da partição usando vários sufixos diferentes:

#### MiB, GiB, ou TiB

Tamanho expresso em poderes de 2.

- O ponto de partida da divisória é alinhado ao setor exato especificado por tamanho.
- O ponto final é alinhado com o tamanho especificado menos 1 setor.

#### MB, GB, ou TB

Tamanho expresso em poderes de 10.

O ponto inicial e final é alinhado dentro de uma metade da unidade especificada: por exemplo,  $\pm 500\text{KB}$  ao utilizar o sufixo MB.

## 10.5.2. Redimensionamento de uma divisória com separação

Este procedimento redimensiona uma partição de disco usando o utilitário **parted**.

### Pré-requisitos

- Se você quiser encolher uma partição, faça backup dos dados que estão armazenados nela.



#### ATENÇÃO

A retração de uma partição pode resultar em perda de dados na partição.

- Se você quiser redimensionar uma partição para ser maior que 2TiB, o disco deve ser formatado com a Tabela de Partição GUID (GPT). Para obter detalhes sobre como formatar o disco, consulte [Seção 10.2, "Criação de uma tabela de partição em um disco"](#).

### Procedimento

1. Se você quiser diminuir a partição, encolha primeiro o sistema de arquivo sobre ela para que não seja maior do que a partição redimensionada. Note que o XFS não suporta o encolhimento.
2. Inicie o shell interativo **parted**:

```
# Separado block-device
```

- Substitua *block-device* com o caminho para o dispositivo onde você deseja redimensionar uma partição: por exemplo, **/dev/sda**.

3. Veja a tabela de partição atual:

```
(dividido) impressão
```

A partir da tabela de partição, determine:

- O número menor da divisória
- A localização da divisória existente e seu novo ponto final após o redimensionamento

4. Redimensionar a divisória:

```
(dividido) reizepart minor-number new-end
```

- Substitua *minor-number* com o número menor da partição que você está redimensionando: por exemplo, **3**.
- Substitua *new-end* com o tamanho que determina o novo ponto final da partição redimensionada, contando desde o início do disco. Pode-se usar sufixos de tamanho, como **512MiB**, **20GiB**, ou **1.5TiB**. O tamanho padrão de megabytes.

#### Exemplo 10.4. Ampliação de uma divisória

Por exemplo, para estender uma partição localizada no início do disco para ser 2GiB em tamanho, use:

```
(dividido) resizepart 1 2GiB
```

As mudanças começam a acontecer assim que você entra neste comando, portanto, revise-o antes de executá-lo.

5. Veja a tabela de partição para confirmar que a partição redimensionada está na tabela de partição com o tamanho correto:

```
(dividido) impressão
```

6. Saia da casca **parted**:

```
(separado) desistir
```

7. Verificar se o núcleo reconhece a nova partição:

```
# gato /proc/partições
```

8. Se você estender a partição, estenda o sistema de arquivo também sobre ela. Veja (referência) para detalhes.

## Recursos adicionais

- A página do homem **parted(8)**.

## 10.6. ESTRATÉGIAS PARA REPARTICIONAR UM DISCO

Há várias maneiras diferentes de repartir um disco. Esta seção discute as seguintes abordagens possíveis:

- Espaço livre não repartido está disponível
- Uma divisória não utilizada está disponível
- Espaço livre em uma divisória usada ativamente está disponível

Observe que esta seção discute os conceitos mencionados anteriormente apenas teoricamente e não inclui nenhuma etapa processual sobre como realizar o reparticionamento de discos passo a passo.



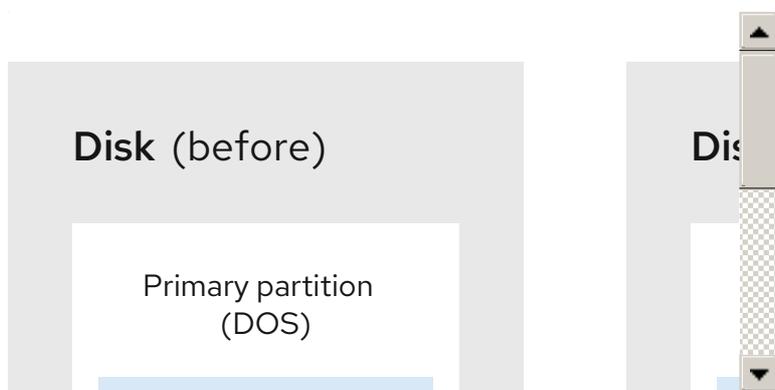
### NOTA

As ilustrações a seguir são simplificadas no interesse da clareza e não refletem o layout exato da partição que você encontra ao instalar efetivamente o Red Hat Enterprise Linux.

### 10.6.1. Utilização de espaço livre não particionado

Nesta situação, as partições já definidas não abrangem todo o disco rígido, deixando espaço não alocado que não faz parte de nenhuma partição definida. O diagrama a seguir mostra o que isto pode parecer:

Figura 10.5. Disco com espaço livre não particionado



No exemplo anterior, o primeiro diagrama representa um disco com uma partição primária e uma partição indefinida com espaço não alocado, e o segundo diagrama representa um disco com duas partições definidas com espaço alocado.

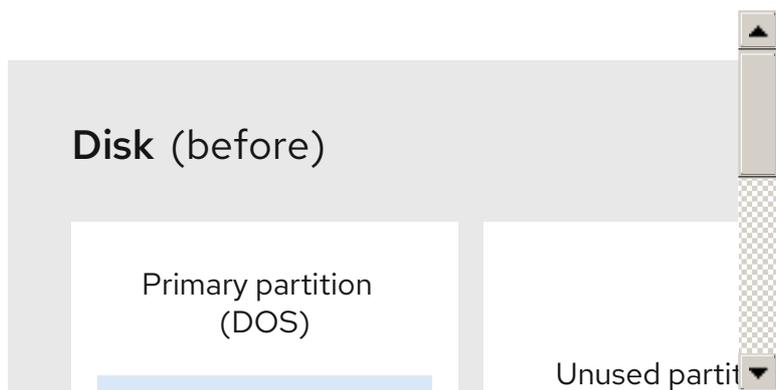
Um disco rígido não utilizado também se enquadra nesta categoria. A única diferença é que *all* o espaço não é parte de nenhuma partição definida.

Em qualquer caso, você pode criar as partições necessárias a partir do espaço não utilizado. Este cenário é mais provável para um novo disco. A maioria dos sistemas operacionais pré-instalados são configurados para ocupar todo o espaço disponível em um drive de disco.

### 10.6.2. Usando o espaço de uma divisória não utilizada

Neste caso, você pode ter uma ou mais divisórias que você não usa mais. O diagrama a seguir ilustra tal situação.

**Figura 10.6. Disco com uma divisória não utilizada**



No exemplo anterior, o primeiro diagrama representa um disco com uma partição não utilizada, e o segundo diagrama representa a realocação de uma partição não utilizada para o Linux.

Nesta situação, você pode utilizar o espaço alocado para a divisória não utilizada. Você deve apagar a partição e depois criar a(s) partição(ões) Linux apropriada(s) em seu lugar. Você pode apagar a partição não utilizada e criar manualmente novas partições durante o processo de instalação.

### 10.6.3. Usando o espaço livre de uma divisória ativa

Esta é a situação mais comum. É também a mais difícil de lidar, porque mesmo que você tenha espaço livre suficiente, ele está atualmente alocado a uma divisória que já está em uso. Se você adquiriu um computador com software pré-instalado, o disco rígido muito provavelmente tem uma partição maciça que contém o sistema operacional e os dados.

Além de adicionar um novo disco rígido ao seu sistema, você pode escolher entre repartições destrutivas e não-destrutivas.

#### 10.6.3.1. Repartição destrutiva

Isto elimina a divisória e cria várias menores em seu lugar. Você deve fazer um backup completo porque qualquer dado na partição original é destruído. Crie dois backups, use verificação (se disponível em seu software de backup), e tente ler os dados do backup *before* apagando a partição.

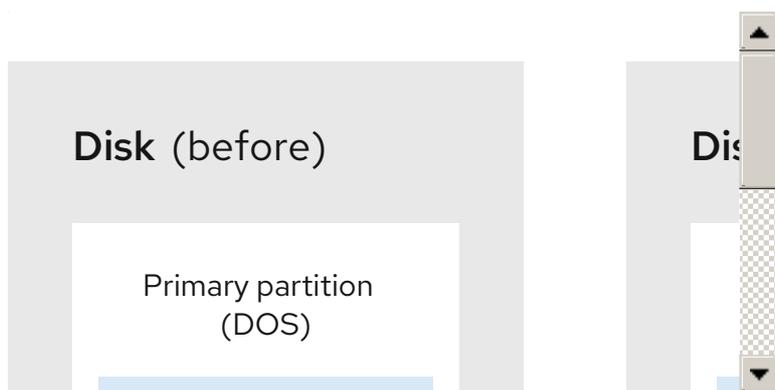


#### ATENÇÃO

Se um sistema operacional foi instalado naquela partição, ele deve ser reinstalado se você quiser usar esse sistema também. Esteja ciente de que alguns computadores vendidos com sistemas operacionais pré-instalados podem não incluir a mídia de instalação para reinstalar o sistema operacional original. Você deve verificar se isto se aplica a seu sistema *before* você destrói sua partição original e sua instalação do sistema operacional.

Após criar uma partição menor para seu sistema operacional existente, você pode reinstalar o software, restaurar seus dados e iniciar sua instalação do Red Hat Enterprise Linux.

Figura 10.7. Ação reparticionadora destrutiva em disco

**ATENÇÃO**

Qualquer dado anteriormente presente na partição original é perdido.

**10.6.3.2. Repartição não-destrutiva**

Com o reparticionamento não destrutivo você executa um programa que torna uma grande partição menor sem perder nenhum dos arquivos armazenados naquela partição. Este método é geralmente confiável, mas pode ser muito demorado em grandes unidades.

O processo de reparticionamento não destrutivo é simples e consiste em três etapas:

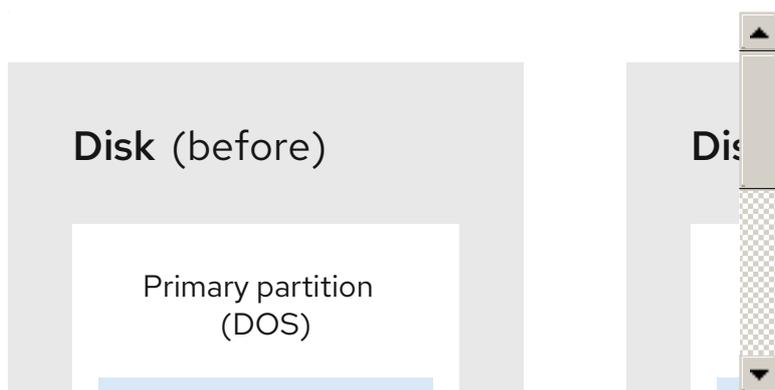
1. Comprimir e fazer backup dos dados existentes
2. Redimensionar a divisória existente
3. Criar nova(s) divisória(s)

Cada passo é descrito com mais detalhes.

**10.6.3.2.1. Compressão de dados existentes**

O primeiro passo é comprimir os dados em sua partição existente. A razão para fazer isto é reorganizar os dados para maximizar o espaço livre disponível no "fim" da partição.

Figura 10.8. Compressão em disco



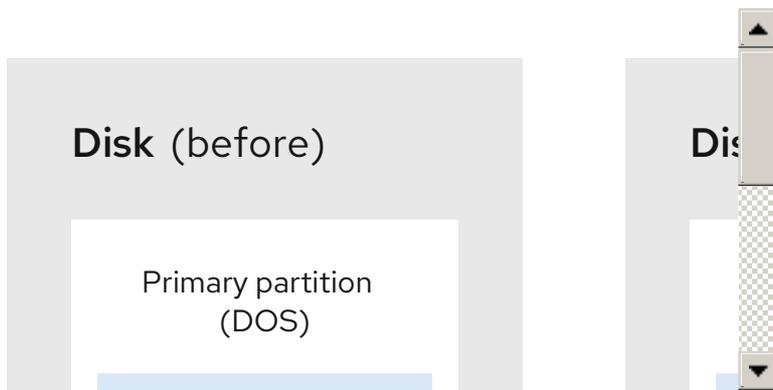
No exemplo anterior, o primeiro diagrama representa o disco antes da compressão, e o segundo diagrama após a compressão.

Este passo é crucial. Sem ela, a localização dos dados poderia impedir que a partição fosse redimensionada na medida desejada. Note que alguns dados não podem ser movidos. Neste caso, isso restringe severamente o tamanho de suas novas partições, e você poderá ser forçado a reparticionar seu disco de forma destrutiva.

#### 10.6.3.2.2. Redimensionamento da divisória existente

A figura a seguir mostra o processo real de redimensionamento. Enquanto o resultado real da operação de redimensionamento varia, dependendo do software utilizado, na maioria dos casos o espaço recém-liberado é utilizado para criar uma partição sem formatação do mesmo tipo da partição original.

Figura 10.9. Redimensionamento da partição em disco



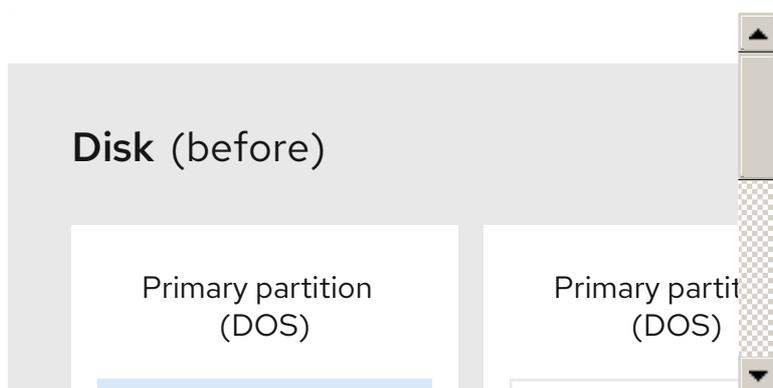
No exemplo anterior, o primeiro diagrama representa a partição antes do redimensionamento, e o segundo diagrama após o redimensionamento.

É importante entender o que o software de redimensionamento que você usa faz com o espaço recém-liberado, para que você possa tomar as medidas apropriadas. No caso aqui ilustrado, seria melhor apagar a nova partição DOS e criar a partição ou partições Linux apropriadas.

#### 10.6.3.2.3. Criação de novas divisórias

Como mencionado no exemplo [Redimensionando a partição existente](#), pode ou não ser necessário criar novas partições. Entretanto, a menos que seu software de redimensionamento suporte sistemas com Linux instalado, é provável que você tenha que apagar a partição que foi criada durante o processo de redimensionamento.

Figura 10.10. Disco com configuração final da divisória



No exemplo anterior, o primeiro diagrama representa o disco antes da configuração, e o segundo diagrama após a configuração.

## CAPÍTULO 11. COMEÇANDO COM XFS

Esta é uma visão geral de como criar e manter sistemas de arquivos XFS.

### 11.1. O SISTEMA DE ARQUIVO XFS

O XFS é um sistema de arquivo de diário de 64 bits altamente escalável, de alto desempenho, robusto e maduro que suporta arquivos muito grandes e sistemas de arquivo em um único host. É o sistema de arquivo padrão no Red Hat Enterprise Linux 8. O XFS foi originalmente desenvolvido no início dos anos 90 pela SGI e tem um longo histórico de rodar em servidores e arrays de armazenamento extremamente grandes.

As características do XFS incluem:

#### Confiabilidade

- Diário de Metadados, que garante a integridade do sistema de arquivo após uma falha do sistema, mantendo um registro das operações do sistema de arquivo que pode ser reproduzido quando o sistema é reiniciado e o sistema de arquivo recontado
- Verificação extensiva da consistência dos metadados em tempo de execução
- Utilitários escaláveis e de reparo rápido
- Diário de cotas. Isto evita a necessidade de longas verificações de consistência de cotas após uma queda.

#### Escalabilidade e desempenho

- Tamanho do sistema de arquivo suportado até 1024 TiB
- Capacidade de suportar um grande número de operações simultâneas
- Indexação B-tree para a escalabilidade da gestão do espaço livre
- Sofisticados algoritmos de leitura de metadados
- Otimizações para a carga de trabalho de streaming de vídeo

#### Esquemas de alocação

- Alocação baseada na extensão
- Políticas de alocação de listras
- Atraso na alocação
- Pré-alocação de espaço
- Inódios alocados dinamicamente

#### Outras características

- Cópias de arquivos com base no Refflink (novo no Red Hat Enterprise Linux 8)
- Utilitários de backup e restauração bem integrados

- Desfragmentação on-line
- Sistema de arquivo on-line crescendo
- Recursos abrangentes de diagnóstico
- Atributos estendidos (**xattr**). Isto permite que o sistema associe vários pares de nome/valor adicionais por arquivo.
- Cotas de projetos ou diretórios. Isto permite restrições de cotas sobre uma árvore de diretórios.
- Carimbos de tempo subseqüentes

### Características de desempenho

O XFS tem um alto desempenho em grandes sistemas com cargas de trabalho empresariais. Um sistema grande é aquele com um número relativamente alto de CPUs, múltiplos HBAs e conexões com matrizes de disco externas. O XFS também tem um bom desempenho em sistemas menores que têm uma carga de trabalho de E/S paralela e multi-tarefa.

O XFS tem um desempenho relativamente baixo para cargas de trabalho com um único threaded e metadata intensivo: por exemplo, uma carga de trabalho que cria ou elimina um grande número de pequenos arquivos em um único thread.

## 11.2. CRIAÇÃO DE UM SISTEMA DE ARQUIVOS XFS

Como administrador de sistema, você pode criar um sistema de arquivos XFS em um dispositivo de bloco para permitir que ele armazene arquivos e diretórios.

### 11.2.1. Criando um sistema de arquivo XFS com `mkfs.xfs`

Este procedimento descreve como criar um sistema de arquivo XFS em um dispositivo de bloco.

#### Procedimento

1. Para criar o sistema de arquivo:

- Se o dispositivo for uma partição normal, um volume LVM, um volume MD, um disco, ou um dispositivo similar, use o seguinte comando:

```
# mkfs.xfs block-device
```

- Substitua *block-device* com o caminho para o dispositivo do bloco. Por exemplo, `/dev/sdb1`, `/dev/disk/by-uuid/05e99ec8-def1-4a5e-8a9d-5945339ceb2a`, ou `/dev/myvolgroup/my-lv`.
- Em geral, as opções padrão são ótimas para uso comum.
- Ao utilizar **mkfs.xfs** em um dispositivo de bloco contendo um sistema de arquivo existente, adicione a opção **-f** para sobrescrever esse sistema de arquivo.
- Para criar o sistema de arquivo em um dispositivo RAID de hardware, verifique se o sistema detecta corretamente a geometria da faixa do dispositivo:

- Se as informações sobre a geometria das faixas estiverem corretas, não são necessárias opções adicionais. Crie o sistema de arquivo:

```
# mkfs.xfs block-device
```

- Se as informações estiverem incorretas, especifique manualmente a geometria das faixas com os parâmetros **su** e **sw** da opção **-d**. O parâmetro **su** especifica o tamanho do trecho RAID, e o parâmetro **sw** especifica o número de discos de dados no dispositivo RAID.

Por exemplo:

```
# mkfs.xfs -d su=64k,sw=4 /dev/sda3
```

- Use o seguinte comando para esperar que o sistema registre o novo nó de dispositivo:

```
# udevadm settle
```

### Recursos adicionais

- A página do homem **mkfs.xfs(8)**.

## 11.2.2. Criação de um sistema de arquivo XFS em um dispositivo de bloco usando funções do sistema RHEL

Esta seção descreve como criar um sistema de arquivos XFS em um dispositivo de bloco em múltiplas máquinas de destino usando a função **storage**.

### Pré-requisitos

- Existe um livro de brincadeiras possível que utiliza o papel **storage**. Para informações sobre como aplicar tal playbook, consulte [Aplicando um papel](#).

#### 11.2.2.1. 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 11.1. 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`.

#### 11.2.2.2. Recursos adicionais

- Para mais informações sobre a função **storage**, ver [Seção 2.1, "Introdução à função de armazenamento"](#).

## 11.3. CÓPIA DE SEGURANÇA DE UM SISTEMA DE ARQUIVO XFS

Como administrador de sistema, você pode usar o **xfsdump** para fazer backup de um sistema de arquivo XFS em um arquivo ou em uma fita. Isto fornece um mecanismo de backup simples.

### 11.3.1. Características do backup do XFS

Esta seção descreve os principais conceitos e características de backup de um sistema de arquivos XFS com o utilitário **xfsdump**.

Você pode usar o utilitário **xfsdump** para:

- Realizar backups para imagens de arquivos regulares. Apenas um backup pode ser escrito em um arquivo normal.
- Realizar backups para unidades de fita adesiva. O utilitário **xfsdump** também permite que você escreva vários backups na mesma fita. Uma cópia de segurança pode abranger várias fitas.

Para fazer backup de vários sistemas de arquivos em um único dispositivo de fita, basta escrever o backup em uma fita que já contenha um backup XFS. Isto acrescenta o novo backup ao anterior. Por padrão, **xfsdump** nunca sobrescreve backups existentes.

- Criar backups incrementais.  
O utilitário **xfsdump** usa níveis de despejo para determinar um backup de base ao qual outros backups são relativos. Números de 0 a 9 referem-se ao aumento dos níveis de despejo. Um backup incremental só faz backup de arquivos que mudaram desde o último depósito de lixo de um nível inferior:
  - Para realizar um backup completo, realize uma descarga de nível 0 no sistema de arquivo.
  - Uma descarga de nível 1 é o primeiro backup incremental após um backup completo. O próximo backup incremental seria o nível 2, que só faz backup de arquivos que mudaram desde o último depósito de lixo de nível 1; e assim por diante, a um máximo de nível 9.
- Excluir arquivos de um backup usando bandeiras de tamanho, sub-árvore ou inode para filtrá-los.

### Recursos adicionais

- A página do homem **xfsdump(8)**.

### 11.3.2. Cópia de segurança de um sistema de arquivo XFS com xfsdump

Este procedimento descreve como fazer o backup do conteúdo de um sistema de arquivo XFS em um arquivo ou fita.

#### Pré-requisitos

- Um sistema de arquivo XFS que você pode fazer backup.
- Outro sistema de arquivo ou um drive de fita onde você pode armazenar o backup.

#### Procedimento

- Use o seguinte comando para fazer o backup de um sistema de arquivos XFS:

```
# xfsdump -l level [-L label] \  
-f backup-destination path-to-xfs-filesystem
```

- Substitua *level* com o nível de despejo de seu backup. Use **0** para realizar um backup completo ou **1** para **9** para realizar os backups incrementais seqüentes.
- Substitua *backup-destination* pelo caminho onde você quer armazenar seu backup. O destino pode ser um arquivo normal, um drive de fita ou um dispositivo remoto de fita. Por exemplo, **/backup-files/Data.xfsdump** para um arquivo ou **/dev/st0** para uma unidade de fita adesiva.
- Substitua *path-to-xfs-filesystem* pelo ponto de montagem do sistema de arquivos XFS que você deseja fazer backup. Por exemplo, **/mnt/data/**. O sistema de arquivo deve ser montado.
- Ao fazer backup de vários sistemas de arquivos e salvá-los em um único dispositivo de fita, adicione uma etiqueta de sessão a cada backup usando o **-L *label*** para que seja mais fácil identificá-las ao restaurá-las. Substitua *label* por qualquer nome para seu backup: por exemplo, **backup\_data**.

#### Exemplo 11.2. Cópia de segurança de múltiplos sistemas de arquivo XFS

- Para fazer backup do conteúdo dos sistemas de arquivos XFS montados nos diretórios **/boot/** e **/data/** e salvá-los como arquivos no diretório **/backup-files/**:

```
# xfsdump -l 0 -f /backup-files/boot.xfsdump /boot
# xfsdump -l 0 -f /backup-files/data.xfsdump /data
```

- Para fazer o backup de vários sistemas de arquivo em um único dispositivo de fita, adicione uma etiqueta de sessão a cada backup usando o **-L label** opção:

```
# xfsdump -l 0 -L "backup_boot" -f /dev/st0 /boot
# xfsdump -l 0 -L "backup_data" -f /dev/st0 /data
```

### Recursos adicionais

- A página do homem **xfsdump(8)**.

### 11.3.3. Recursos adicionais

- A página do homem **xfsdump(8)**.

## 11.4. RESTAURANDO UM SISTEMA DE ARQUIVO XFS A PARTIR DE BACKUP

Como administrador do sistema, você pode usar o utilitário **xfsrestore** para restaurar o backup XFS criado com o utilitário **xfsdump** e armazenado em um arquivo ou em uma fita.

### 11.4.1. Características da restauração do XFS a partir do backup

Esta seção descreve os principais conceitos e características da restauração de um sistema de arquivos XFS a partir de backup com o utilitário **xfsrestore**.

O utilitário **xfsrestore** restaura os sistemas de arquivos a partir de backups produzidos por **xfsdump**. O utilitário **xfsrestore** tem dois modos:

- O modo **simple** permite aos usuários restaurar um sistema de arquivo inteiro a partir de um depósito de nível 0. Este é o modo padrão.
- O modo **cumulative** permite a restauração do sistema de arquivos a partir de um backup incremental: ou seja, do nível 1 ao nível 9.

Um exclusivo *session ID* ou *session label* identifica cada backup. A restauração de um backup a partir de uma fita contendo múltiplos backups requer sua identificação de sessão ou etiqueta correspondente.

Para extrair, adicionar ou excluir arquivos específicos de um backup, entre no modo interativo **xfsrestore**. O modo interativo fornece um conjunto de comandos para manipular os arquivos de backup.

### Recursos adicionais

- A página do homem **xfsrestore(8)**.

### 11.4.2. Restaurando um sistema de arquivo XFS a partir de backup com xfsrestore

Este procedimento descreve como restaurar o conteúdo de um sistema de arquivo XFS a partir de um arquivo ou fita de backup.

### Pré-requisitos

- Um arquivo ou fita de backup dos sistemas de arquivo XFS, como descrito em [Seção 11.3, “Cópia de segurança de um sistema de arquivo XFS”](#).
- Um dispositivo de armazenamento onde você pode restaurar o backup.

### Procedimento

- O comando para restaurar o backup varia dependendo se você está restaurando a partir de um backup completo ou incremental, ou se está restaurando múltiplos backups a partir de um único dispositivo de fita:

```
# xfsrestore [-r] [-S session-id] [-L session-label] [-i]
-f backup-location restoration-path
```

- Substitua *backup-location* com a localização do backup. Pode ser um arquivo normal, um drive de fita ou um dispositivo remoto de fita. Por exemplo, **/backup-files/Data.xfsdump** para um arquivo ou **/dev/st0** para uma unidade de fita adesiva.
- Substitua *restoration-path* com o caminho para o diretório onde se deseja restaurar o sistema de arquivos. Por exemplo, **/mnt/data/**.
- Para restaurar um sistema de arquivo de um backup incremental (nível 1 para nível 9), adicione a opção **-r**.
- Para restaurar um backup a partir de um dispositivo de fita que contém múltiplos backups, especifique o backup usando as opções **-S** ou **-L**. A opção **-S** permite que você escolha um backup por seu ID de sessão, enquanto a opção **-L** permite que você escolha pelo rótulo da sessão. Para obter a identificação da sessão e a etiqueta da sessão, use o comando **xfsrestore -i**.

Substitua *session-id* com a identificação da sessão de backup. Por exemplo, **b74a3586-e52e-4a4a-8775-c3334fa8ea2c**. Substituir *session-label* com a etiqueta da sessão de backup. Por exemplo, **my\_backup\_session\_label**.

- Para usar **xfsrestore** interativamente, use a opção **-i**. O diálogo interativo começa após **xfsrestore** terminar a leitura do dispositivo especificado. Os comandos disponíveis no shell interativo **xfsrestore** incluem **cd**, **ls**, **add**, **delete**, e **extract**; para uma lista completa de comandos, use o comando **help**.

#### Exemplo 11.3. Restaurando múltiplos sistemas de arquivos XFS

- Para restaurar os arquivos de backup do XFS e salvar seu conteúdo em diretórios sob **/mnt/**:

```
# xfsrestore -f /backup-files/boot.xfsdump /mnt/boot/
# xfsrestore -f /backup-files/data.xfsdump /mnt/data/
```

- Para restaurar a partir de um dispositivo de fita contendo múltiplos backups, especifique cada backup por sua etiqueta de sessão ou ID de sessão:

```
# xfsrestore -L "backup_boot" -f /dev/st0 /mnt/boot/
# xfsrestore -S "45e9af35-efd2-4244-87bc-4762e476cbab" \
-f /dev/st0 /mnt/data/
```

### Recursos adicionais

- A página do homem **xfsrestore(8)**.

### 11.4.3. Mensagens informativas na restauração de um backup XFS a partir de uma fita

Ao restaurar um backup a partir de uma fita com backups de vários sistemas de arquivos, o utilitário **xfsrestore** pode emitir mensagens. As mensagens informam se foi encontrada uma correspondência do backup solicitado quando **xfsrestore** examina cada backup na fita em ordem seqüencial. Por exemplo:

```
xfsrestore: preparing drive
xfsrestore: examining media file 0
xfsrestore: inventory session uuid (8590224e-3c93-469c-a311-fc8f23029b2a) does not match the
media header's session uuid (7eda9f86-f1e9-4dfd-b1d4-c50467912408)
xfsrestore: examining media file 1
xfsrestore: inventory session uuid (8590224e-3c93-469c-a311-fc8f23029b2a) does not match the
media header's session uuid (7eda9f86-f1e9-4dfd-b1d4-c50467912408)
[...]
```

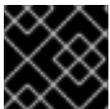
As mensagens informativas continuam aparecendo até que o backup correspondente seja encontrado.

### 11.4.4. Recursos adicionais

- A página do homem **xfsrestore(8)**.

## 11.5. AUMENTANDO O TAMANHO DE UM SISTEMA DE ARQUIVO XFS

Como administrador de sistema, você pode aumentar o tamanho de um sistema de arquivos XFS para utilizar uma maior capacidade de armazenamento.



### IMPORTANTE

Atualmente não é possível diminuir o tamanho dos sistemas de arquivo XFS.

### 11.5.1. Aumentando o tamanho de um sistema de arquivo XFS com **xfs\_growfs**

Este procedimento descreve como cultivar um sistema de arquivos XFS usando o utilitário **xfs\_growfs**.

#### Pré-requisitos

- Certifique-se de que o dispositivo de bloco subjacente seja de tamanho apropriado para segurar o sistema de arquivo redimensionado posteriormente. Usar os métodos apropriados de redimensionamento para o dispositivo de bloco afetado.
- Montar o sistema de arquivos XFS.

## Procedimento

- Enquanto o sistema de arquivo XFS é montado, use o utilitário **xfs\_growfs** para aumentar seu tamanho:

```
# xfs_growfs file-system -D new-size
```

- Substitua *file-system* com o ponto de montagem do sistema de arquivo XFS.
- Com a opção **-D**, substitua *new-size* com o novo tamanho desejado do sistema de arquivo especificado no número de blocos do sistema de arquivo.  
Para descobrir o tamanho do bloco em kB de um determinado sistema de arquivos XFS, use o utilitário **xfs\_info**:

```
# xfs_info block-device
...
data    =          bsize=4096
...
```

- Sem a opção **-D**, **xfs\_growfs** aumenta o sistema de arquivo até o tamanho máximo suportado pelo dispositivo subjacente.

## Recursos adicionais

- A página do homem **xfs\_growfs(8)**.

## 11.6. COMPARAÇÃO DAS FERRAMENTAS UTILIZADAS COM EXT4 E XFS

Esta seção compara quais ferramentas usar para realizar tarefas comuns nos sistemas de arquivos ext4 e XFS.

Tarefa	ext4	XFS
Criar um sistema de arquivo	<b>mkfs.ext4</b>	<b>mkfs.xfs</b>
Verificação do sistema de arquivo	<b>e2fsck</b>	<b>xfs_repair</b>
Redimensionar um sistema de arquivo	<b>resize2fs</b>	<b>xfs_growfs</b>
Salvar uma imagem de um sistema de arquivo	<b>e2image</b>	<b>xfs_metadump</b> e <b>xfs_mdrestore</b>
Etiquetar ou afinar um sistema de arquivo	<b>tune2fs</b>	<b>xfs_admin</b>
Faça o backup de um sistema de arquivo	<b>dump</b> e <b>restore</b>	<b>xfsdump</b> e <b>xfsrestore</b>

Tarefa	ext4	XFS
Gestão de cotas	<b>quota</b>	<b>xfs_quota</b>
Mapeamento de arquivos	<b>filefrag</b>	<b>xfs_bmap</b>

## CAPÍTULO 12. CONFIGURAÇÃO DO COMPORTAMENTO DE ERRO DO XFS

Você pode configurar como um sistema de arquivo XFS se comporta quando encontra diferentes erros de E/S.

### 12.1. MANUSEIO DE ERROS CONFIGURÁVEL EM XFS

O sistema de arquivo XFS responde de uma das seguintes maneiras quando ocorre um erro durante uma operação de E/S:

- O XFS repetidamente repete a operação de E/S até que a operação tenha sucesso ou o XFS atinja um limite estabelecido.  
O limite é baseado ou em um número máximo de tentativas ou em um tempo máximo de tentativas.
- O XFS considera o erro permanente e pára a operação no sistema de arquivo.

Você pode configurar como o XFS reage às seguintes condições de erro:

#### **EIO**

Erro ao ler ou escrever

#### **ENOSPC**

Não há mais espaço no dispositivo

#### **ENODEV**

O dispositivo não pode ser encontrado

Você pode definir o número máximo de tentativas e o tempo máximo em segundos até que a XFS considere um erro permanente. O XFS deixa de tentar novamente a operação quando atinge qualquer um dos limites.

Você também pode configurar o XFS para que, ao desmontar um sistema de arquivo, o XFS cancele imediatamente as novas tentativas, independentemente de qualquer outra configuração. Esta configuração permite que a operação de desmontagem seja bem sucedida, apesar dos erros persistentes.

#### **Comportamento padrão**

O comportamento padrão para cada condição de erro XFS depende do contexto de erro. Alguns erros XFS, como **ENODEV**, são considerados fatais e irrecuperáveis, independentemente da contagem da repetição. Seu limite padrão de reentrada é 0.

### 12.2. ARQUIVOS DE CONFIGURAÇÃO PARA CONDIÇÕES DE ERRO XFS ESPECÍFICAS E INDEFINIDAS

Os seguintes diretórios armazenam arquivos de configuração que controlam o comportamento de erro do XFS para diferentes condições de erro:

**/sys/fs/xfs/device/error/metadata/EIO/**

Para a condição de erro **EIO**

**/sys/fs/xfs/device/error/metadata/ENODEV/**

Para a condição de erro **ENODEV**

**/sys/fs/xfs/device/error/metadata/ENOSPC/**

Para a condição de erro **ENOSPC**

**/sys/fs/xfs/device/error/default/**

Configuração comum para todas as outras condições de erro indefinido

Cada diretório contém os seguintes arquivos de configuração para configurar os limites de nova tentativa:

**max\_retries**

Controla o número máximo de vezes que o XFS re-testa a operação.

**retry\_timeout\_seconds**

Especifica o limite de tempo em segundos após o qual o XFS deixa de tentar novamente a operação.

## 12.3. DEFINIÇÃO DO COMPORTAMENTO XFS PARA CONDIÇÕES ESPECÍFICAS

Este procedimento configura como o XFS reage a condições de erro específicas.

### Procedimento

- Defina o número máximo de tentativas, o limite de tempo de tentativas, ou ambos:
  - Para definir o número máximo de tentativas, escreva o número desejado no arquivo **max\_retries**:

```
# echo value > /sys/fs/xfs/device/error/metadata/condition/max_retries
```

- Para definir o limite de tempo, escreva o número de segundos desejado no arquivo **retry\_timeout\_seconds**:

```
# echo value > /sys/fs/xfs/device/error/metadata/condition/retry_timeout_second
```

*value* é um número entre -1 e o valor máximo possível do tipo C inteiro assinado. Este é 2147483647 em Linux de 64 bits.

Em ambos os limites, o valor **-1** é usado para tentativas contínuas e **0** para parar imediatamente.

*device* é o nome do dispositivo, como encontrado no diretório **/dev/**; por exemplo, **sda**.

## 12.4. DEFININDO O COMPORTAMENTO DO XFS PARA CONDIÇÕES INDEFINIDAS

Este procedimento configura como o XFS reage a todas as condições de erro indefinidas, que compartilham uma configuração comum.

### Procedimento

- Defina o número máximo de tentativas, o limite de tempo de tentativas, ou ambos:
  - Para definir o número máximo de tentativas, escreva o número desejado no arquivo **max\_retries**:

■

```
# echo value > /sys/fs/xfs/device/error/metadata/default/max_retries
```

- Para definir o limite de tempo, escreva o número de segundos desejado no arquivo **retry\_timeout\_seconds**:

```
# echo value > /sys/fs/xfs/device/error/metadata/default/retry_timeout_seconds
```

*value* é um número entre -1 e o valor máximo possível do tipo C inteiro assinado. Este é 2147483647 em Linux de 64 bits.

Em ambos os limites, o valor **-1** é usado para tentativas contínuas e **0** para parar imediatamente.

*device* é o nome do dispositivo, como encontrado no diretório **/dev/**; por exemplo, **sda**.

## 12.5. COMPORTAMENTO DE DESMONTAGEM DO XFS

Este procedimento configura como o XFS reage às condições de erro ao desmontar o sistema de arquivo.

Se você definir a opção **fail\_at\_unmount** no sistema de arquivo, ela substitui todas as outras configurações de erro durante a desmontagem, e imediatamente desmonta o sistema de arquivo sem tentar novamente a operação de E/S. Isto permite que a operação de desmontagem seja bem sucedida mesmo no caso de erros persistentes.



### ATENÇÃO

Não é possível alterar o valor **fail\_at\_unmount** após o início do processo de desmontagem, porque o processo de desmontagem remove os arquivos de configuração da interface **sysfs** para o respectivo sistema de arquivos. Você deve configurar o comportamento de desmontagem antes de o sistema de arquivos começar a desmontar.

### Procedimento

- Habilite ou desabilite a opção **fail\_at\_unmount**:
  - Para cancelar a repetição de todas as operações quando o sistema de arquivos for desmontado, habilite a opção:

```
# echo 1 > /sys/fs/xfs/device/error/fail_at_unmount
```

- Para respeitar os limites **max\_retries** e **retry\_timeout\_seconds**, quando o sistema de arquivo se desmonta, desabilite a opção:

```
# echo 0 > /sys/fs/xfs/device/error/fail_at_unmount
```

*device* é o nome do dispositivo, como encontrado no diretório **/dev/**; por exemplo, **sda**.

## CAPÍTULO 13. VERIFICAÇÃO E REPARO DE UM SISTEMA DE ARQUIVO

A RHEL fornece utilitários de administração de sistemas de arquivos que são capazes de verificar e reparar sistemas de arquivos. Essas ferramentas são frequentemente referidas como ferramentas **fsck**, onde **fsck** é uma versão resumida de *file system check*. Na maioria dos casos, estes utilitários são executados automaticamente durante a inicialização do sistema, se necessário, mas também podem ser invocados manualmente, se necessário.



### IMPORTANTE

Os verificadores do sistema de arquivos garantem apenas a consistência dos metadados em todo o sistema de arquivos. Eles não têm conhecimento dos dados reais contidos no sistema de arquivos e não são ferramentas de recuperação de dados.

### 13.1. CENÁRIOS QUE REQUEREM UMA VERIFICAÇÃO DO SISTEMA DE ARQUIVO

As ferramentas relevantes **fsck** podem ser usadas para verificar seu sistema se alguma das seguintes ocorrências ocorrer:

- O sistema não inicia
- Os arquivos em um disco específico se tornam corruptos
- O sistema de arquivo se desliga ou muda para somente leitura devido a inconsistências
- Um arquivo no sistema de arquivo é inacessível

As inconsistências do sistema de arquivos podem ocorrer por vários motivos, incluindo, mas não se limitando a, erros de hardware, erros de administração de armazenamento e bugs de software.



### IMPORTANTE

As ferramentas de verificação do sistema de arquivos não podem reparar problemas de hardware. Um sistema de arquivo deve ser totalmente legível e gravável para que o reparo possa funcionar com sucesso. Se um sistema de arquivo foi corrompido devido a um erro de hardware, o sistema de arquivo deve primeiro ser movido para um bom disco, por exemplo, com o utilitário **dd(8)**.

Para sistemas de arquivo de diário, tudo o que normalmente é necessário no momento da inicialização é reproduzir o diário, se necessário, e isto geralmente é uma operação muito curta.

Entretanto, se ocorrer uma inconsistência ou corrupção do sistema de arquivo, mesmo para sistemas de arquivo de diário, então o verificador do sistema de arquivo deve ser usado para reparar o sistema de arquivo.



### IMPORTANTE

É possível desativar a verificação do sistema de arquivos na inicialização, definindo o sexto campo em **/etc/fstab** para **0**. Entretanto, a Red Hat não recomenda fazer isso a menos que você tenha problemas com **fsck** no momento da inicialização, por exemplo, com sistemas de arquivo extremamente grandes ou remotos.

## Recursos adicionais

- A página do homem **fstab(5)**.
- A página do homem **fsck(8)**.
- A página do homem **dd(8)**.

## 13.2. POTENCIAIS EFEITOS COLATERAIS DO FSCK EM FUNCIONAMENTO

Geralmente, a execução da ferramenta de verificação e reparo do sistema de arquivos pode ser esperada para reparar automaticamente pelo menos algumas das inconsistências encontradas. Em alguns casos, podem surgir as seguintes questões:

- Os inodos ou diretórios gravemente danificados podem ser descartados se não puderem ser reparados.
- Podem ocorrer mudanças significativas no sistema de arquivos.

Para garantir que mudanças inesperadas ou indesejáveis não sejam feitas permanentemente, certifique-se de seguir quaisquer passos de precaução delineados no procedimento.

## 13.3. MECANISMOS DE TRATAMENTO DE ERROS EM XFS

Esta seção descreve como o XFS lida com vários tipos de erros no sistema de arquivo.

### Montagens não limpas

Journalling mantém um registro transacional das mudanças de metadados que acontecem no sistema de arquivos.

No caso de uma falha no sistema, falha de energia ou outra montagem não limpa, o XFS usa o diário (também chamado log) para recuperar o sistema de arquivo. O kernel realiza a recuperação do diário ao montar o sistema de arquivos XFS.

### Corrupção

Neste contexto, *corruption* significa erros no sistema de arquivos causados, por exemplo, por:

- Falhas de hardware
- Bugs no firmware de armazenamento, drivers de dispositivos, a pilha de software, ou o próprio sistema de arquivos
- Problemas que fazem com que partes do sistema de arquivo sejam sobregravadas por algo fora do sistema de arquivo

Quando o XFS detecta corrupção no sistema de arquivos ou nos metadados do sistema de arquivos, ele pode desligar o sistema de arquivos e relatar o incidente no registro do sistema. Note que se a corrupção ocorreu no sistema de arquivo que hospeda o diretório **/var**, estes logs não estarão disponíveis após uma reinicialização.

#### Exemplo 13.1. Entrada de registro no sistema relatando uma corrupção no XFS

```
# dmesg --notime | tail -15
```

```

XFS (loop0): Mounting V5 Filesystem
XFS (loop0): Metadata CRC error detected at xfs_agi_read_verify+0xcb/0xf0 [xfs], xfs_agi block
0x2
XFS (loop0): Unmount and run xfs_repair
XFS (loop0): First 128 bytes of corrupted metadata buffer:
00000000027b3b56: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000000005f9abc7a: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000000005b0aef35: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000000da9d2ded: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000000001e265b07: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000000006a40df69: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000000b272907: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000000e484aac5: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
XFS (loop0): metadata I/O error in "xfs_trans_read_buf_map" at daddr 0x2 len 1 error 74
XFS (loop0): xfs_imap_lookup: xfs_ialloc_read_agi() returned error -117, agno 0
XFS (loop0): Failed to read root inode 0x80, error 11

```

Os utilitários de espaço do usuário geralmente relatam a mensagem *Input/output error* quando tentam acessar um sistema de arquivo XFS corrompido. A montagem de um sistema de arquivo XFS com um log corrompido resulta em uma montagem falhada e a seguinte mensagem de erro:

```

montar /mount-point: a chamada ao sistema de montagem(2) falhou: A estrutura precisa ser limpa.

```

Você deve usar manualmente o utilitário **xfs\_repair** para reparar a corrupção.

#### Recursos adicionais

- A página de manual **xfs\_repair(8)** fornece uma lista detalhada de verificações de corrupção XFS.

## 13.4. VERIFICAÇÃO DE UM SISTEMA DE ARQUIVO XFS COM **XFS\_REPAIR**

Este procedimento realiza uma verificação somente leitura de um sistema de arquivos XFS usando o utilitário **xfs\_repair**. Você deve usar manualmente o utilitário **xfs\_repair** para reparar qualquer corrupção. Ao contrário de outros utilitários de reparo do sistema de arquivos, **xfs\_repair** não funciona no momento da inicialização, mesmo quando um sistema de arquivos XFS não foi limpo e desmontado. No caso de uma desmontagem não limpa, o XFS simplesmente reproduz o registro no momento da montagem, garantindo um sistema de arquivo consistente; **xfs\_repair** não pode reparar um sistema de arquivo XFS com um registro sujo sem montá-lo novamente primeiro.



### NOTA

Embora um binário **fsck.xfs** esteja presente no pacote **xfsprogs**, ele está presente apenas para satisfazer **initscripts** que procuram um binário do sistema **fsck.file** no momento da inicialização. **fsck.xfs** sai imediatamente com um código de saída 0.

#### Procedimento

1. Reproduzir o registro, montando e desmontando o sistema de arquivo:

```

# mount file-system
# umount file-system

```



## NOTA

Se a montagem falhar com uma estrutura precisa de erro de limpeza, o registro é corrompido e não pode ser reproduzido. A corrida a seco deve descobrir e relatar mais corrupção no disco como resultado.

- Use o utilitário **xfs\_repair** para realizar uma corrida a seco para verificar o sistema de arquivo. Quaisquer erros são impressos e uma indicação das ações que seriam tomadas, sem modificar o sistema de arquivo.

```
# xfs_repair -n block-device
```

- Montar o sistema de arquivo:

```
# montar file-system
```

## Recursos adicionais

- A página do homem **xfs\_repair(8)**.
- A página do homem **xfs\_metadump(8)**.

## 13.5. CONSERTO DE UM SISTEMA DE ARQUIVO XFS COM XFS\_REPAIR

Este procedimento repara um sistema de arquivo XFS corrompido usando o utilitário **xfs\_repair**.

### Procedimento

- Criar uma imagem de metadados antes do reparo para fins de diagnóstico ou teste usando o utilitário **xfs\_metadump**. Uma imagem de metadados de um sistema de arquivo de pré-reserto pode ser útil para investigações de suporte se a corrupção for devida a um bug de software. Padrões de corrupção presentes na imagem de pré-reparo podem ajudar na análise da causa-raiz.
  - Use a ferramenta de depuração **xfs\_metadump** para copiar os metadados de um sistema de arquivo XFS para um arquivo. O arquivo **metadump** resultante pode ser comprimido usando utilitários de compressão padrão para reduzir o tamanho do arquivo se grandes arquivos **metadump** precisarem ser enviados para suporte.

```
# xfs_metadump block-device metadump-file
```

- Reproduzir o log reconfigurando o sistema de arquivo:

```
# mount file-system
# umount file-system
```

- Use o utilitário **xfs\_repair** para reparar o sistema de arquivo não montado:

- Se a montagem for bem sucedida, não são necessárias opções adicionais:

```
# xfs_repair block-device
```

- Se a montagem falhou com o erro *Structure needs cleaning*, o registro está corrompido e não pode ser reproduzido. Use a opção **-L** (*force log zeroing*) para limpar o log:



### ATENÇÃO

Este comando faz com que todas as atualizações de metadados em andamento no momento da queda sejam perdidas, o que pode causar danos significativos ao sistema de arquivos e perda de dados. Isto só deve ser usado como último recurso se o registro não puder ser reproduzido.

```
# xfs_repair -L block-device
```

4. Montar o sistema de arquivo:

```
# montar file-system
```

#### Recursos adicionais

- A página do homem **xfs\_repair(8)**.

## 13.6. MECANISMOS DE TRATAMENTO DE ERROS EM EXT2, EXT3, E EXT4

Os sistemas de arquivo ext2, ext3 e ext4 utilizam o utilitário **e2fsck** para realizar verificações e reparos no sistema de arquivo. Os nomes dos arquivos **fsck.ext2**, **fsck.ext3**, e **fsck.ext4** são links para o utilitário **e2fsck**. Estes binários são executados automaticamente no momento da inicialização e seu comportamento difere com base no sistema de arquivo que está sendo verificado e no estado do sistema de arquivo.

Uma verificação completa do sistema de arquivo e reparo é invocada para ext2, que não é um sistema de arquivo de metadados de diário, e para sistemas de arquivo ext4 sem um diário.

Para os sistemas de arquivos ext3 e ext4 com diário de metadados, o diário é reproduzido no espaço do usuário e as saídas de utilidades. Esta é a ação padrão porque o journal replay garante um sistema de arquivo consistente após uma falha.

Se estes sistemas de arquivo encontrarem inconsistências de metadados durante a montagem, eles registram este fato no superbloqueio do sistema de arquivo. Se **e2fsck** descobrir que um sistema de arquivo está marcado com tal erro, **e2fsck** executa uma verificação completa após a reprodução do diário (se presente).

#### Recursos adicionais

- A página do homem **fsck(8)**.
- A página do homem **e2fsck(8)**.

## 13.7. VERIFICAÇÃO DE UM SISTEMA DE ARQUIVO EXT2, EXT3, OU EXT4 COM E2FSCK

Este procedimento verifica um sistema de arquivo ext2, ext3, ou ext4 usando o utilitário **e2fsck**.

### Procedimento

1. Reproduzir o log reconfigurando o sistema de arquivo:

```
# mount file-system
# umount file-system
```

2. Realizar uma corrida a seco para verificar o sistema de arquivo.

```
# e2fsck -n block-device
```



### NOTA

Quaisquer erros são impressos e uma indicação das ações que seriam tomadas, sem modificar o sistema de arquivo. Fases posteriores de verificação de consistência podem imprimir erros extras, pois descobre inconsistências que teriam sido corrigidas em fases iniciais se estivesse funcionando em modo de reparo.

### Recursos adicionais

- A página do homem **e2image(8)**.
- A página do homem **e2fsck(8)**.

## 13.8. CONSERTO DE UM SISTEMA DE ARQUIVO EXT2, EXT3, OU EXT4 COM E2FSCK

Este procedimento repara um sistema de arquivo ext2, ext3 ou ext4 corrompido usando o utilitário **e2fsck**.

### Procedimento

1. Salvar uma imagem do sistema de arquivos para investigações de suporte. Uma imagem de metadados de sistema de arquivo pré-resparação pode ser útil para investigações de suporte se a corrupção for devida a um bug de software. Padrões de corrupção presentes na imagem de pré-resparo podem ajudar na análise da causa raiz.



### NOTA

Sistemas de arquivos gravemente danificados podem causar problemas com a criação de imagens de metadados.

- Se você estiver criando a imagem para fins de teste, use a opção **-r** para criar um arquivo esparsado do mesmo tamanho que o próprio sistema de arquivo. **e2fsck** pode então operar diretamente no arquivo resultante.

```
# e2image -r block-device image-file
```

- Se você estiver criando a imagem a ser arquivada ou fornecida para diagnóstico, use a opção **-Q**, que cria um formato de arquivo mais compacto, adequado para transferência.

```
# e2image -Q block-device image-file
```

2. Reproduzir o log reconfigurando o sistema de arquivo:

```
# mount file-system  
# umount file-system
```

3. Consertar automaticamente o sistema de arquivo. Se for necessária a intervenção do usuário, **e2fsck** indica o problema não corrigido em sua saída e reflete este status no código de saída.

```
# e2fsck -p block-device
```

### Recursos adicionais

- A página do homem **e2image(8)**.
- A página do homem **e2fsck(8)**.

## CAPÍTULO 14. MONTAGEM DE SISTEMAS DE ARQUIVO

Como administrador de sistema, você pode montar sistemas de arquivos em seu sistema para acessar dados sobre eles.

### 14.1. O MECANISMO DE MONTAGEM LINUX

Esta seção explica os conceitos básicos de montagem de sistemas de arquivos no Linux.

Em sistemas operacionais Linux, UNIX e similares, sistemas de arquivos em diferentes partições e dispositivos removíveis (CDs, DVDs ou unidades flash USB, por exemplo) podem ser anexados a um determinado ponto (o ponto de montagem) na árvore de diretórios, e depois destacados novamente. Enquanto um sistema de arquivo é montado em um diretório, o conteúdo original do diretório não é acessível.

Note que o Linux não impede que você monte um sistema de arquivos em um diretório com um sistema de arquivos já anexado a ele.

Ao montar, você pode identificar o dispositivo por:

- um identificador universalmente único (UUID): por exemplo, **UUID=34795a28-ca6d-4fd8-a347-73671d0c19cb**
- uma etiqueta de volume: por exemplo, **LABEL=home**
- um caminho completo para um dispositivo de bloqueio não-persistente: por exemplo, **/dev/sda3**

Quando você monta um sistema de arquivo usando o comando **mount** sem todas as informações necessárias, ou seja, sem o nome do dispositivo, o diretório de destino ou o tipo de sistema de arquivo, o utilitário **mount** lê o conteúdo do arquivo **/etc/fstab** para verificar se o sistema de arquivo em questão está listado lá. O arquivo **/etc/fstab** contém uma lista de nomes de dispositivos e os diretórios nos quais os sistemas de arquivo selecionados estão definidos para serem montados, assim como o tipo de sistema de arquivo e opções de montagem. Portanto, ao montar um sistema de arquivo que é especificado em **/etc/fstab**, a seguinte sintaxe de comando é suficiente:

- Montagem pelo ponto de montagem:

```
# montar directory
```

- Montagem pelo dispositivo de bloco:

```
# montar device
```

#### Recursos adicionais

- A página do homem **mount(8)**.
- Para informações sobre como listar atributos de nomeação persistentes, como a UUID, consulte [Seção 9.6, "Listagem de atributos de nomeação persistentes"](#).

### 14.2. LISTAGEM DOS SISTEMAS DE ARQUIVO ATUALMENTE MONTADOS

Este procedimento descreve como listar todos os sistemas de arquivo atualmente montados na linha de comando.

### Procedimento

- Para listar todos os sistemas de arquivos montados, use o utilitário **findmnt**:

```
$ findmnt
```

- Para limitar os sistemas de arquivo listados apenas a um determinado tipo de sistema de arquivo, adicione a opção **--types**:

```
$ findmnt -- tipos fs-type
```

Por exemplo:

#### Exemplo 14.1. Listagem apenas de sistemas de arquivos XFS

```
$ findmnt --types xfs
```

TARGET	SOURCE	FSTYPE	OPTIONS
/	/dev/mapper/luks-5564ed00-6aac-4406-bfb4-c59bf5de48b5	xfs	rw,relatime
	/boot /dev/sda1	xfs	rw,relatime
	/home /dev/mapper/luks-9d185660-7537-414d-b727-d92ea036051e	xfs	rw,relatime

### Recursos adicionais

- A página do homem **findmnt(8)**.

## 14.3. MONTAGEM DE UM SISTEMA DE ARQUIVO COM MONTAGEM

Este procedimento descreve como montar um sistema de arquivo usando o utilitário **mount**.

### Pré-requisitos

- Certifique-se de que nenhum sistema de arquivo já esteja montado em seu ponto de montagem escolhido:

```
$ findmnt mount-point
```

### Procedimento

1. Para anexar um determinado sistema de arquivo, use o utilitário **mount**:

```
# montar device mount-point
```

#### Exemplo 14.2. Montagem de um sistema de arquivo XFS

Por exemplo, para montar um sistema de arquivo XFS local identificado pela UUID:

```
# montagem UUID=ea74bbec-536d-490c-b8d9-5b40bbd7545b /mnt/dados
```

2. Se **mount** não conseguir reconhecer o tipo de sistema de arquivo automaticamente, especifique-o usando a opção **--types**:

```
# montagem -- tipos type device mount-point
```

#### Exemplo 14.3. Montagem de um sistema de arquivo NFS

Por exemplo, para montar um sistema de arquivo NFS remoto:

```
# montar -- tipos nfs4 host:/remote-export /mnt/nfs
```

#### Recursos adicionais

- A página do homem **mount(8)**.

## 14.4. MOVENDO UM PONTO DE MONTAGEM

Este procedimento descreve como mudar o ponto de montagem de um sistema de arquivo montado para um diretório diferente.

#### Procedimento

1. Para mudar o diretório no qual um sistema de arquivo é montado:

```
# montar --move old-directory new-directory
```

#### Exemplo 14.4. Movendo um sistema de arquivo doméstico

Por exemplo, para mover o sistema de arquivo montado no diretório **/mnt/userdirs/** para o ponto de montagem **/home/**:

```
# montar --move /mnt/userdirs /home
```

2. Verificar se o sistema de arquivo foi movido como esperado:

```
$ findmnt  
$ ls old-directory  
$ ls new-directory
```

#### Recursos adicionais

- A página do homem **mount(8)**.

## 14.5. DESMONTANDO UM SISTEMA DE ARQUIVO COM UMOUNT

Este procedimento descreve como desmontar um sistema de arquivo usando o utilitário **umount**.

## Procedimento

1. Tente desmontar o sistema de arquivos usando um dos seguintes comandos:

- Por ponto de montagem:

```
# umount mount-point
```

- Por dispositivo:

```
# umount device
```

Se o comando falhar com um erro semelhante ao seguinte, significa que o sistema de arquivo está em uso por causa de um processo que está usando recursos nele:

```
umount /run/media/user/FlashDrive: o alvo está ocupado.
```

2. Se o sistema de arquivo estiver em uso, use o utilitário **fuser** para determinar quais processos estão acessando o sistema. Por exemplo:

```
$ fuser --mount /run/media/user/FlashDrive
```

```
/run/media/user/FlashDrive: 18351
```

Em seguida, encerrar os processos usando o sistema de arquivo e tentar desmontá-lo novamente.

## 14.6. OPÇÕES COMUNS DE MONTAGEM

Esta seção lista algumas opções comumente usadas do utilitário **mount**.

Você pode usar estas opções na seguinte sintaxe:

```
# montagem --opções option1,option2,option3 device mount-point
```

Tabela 14.1. Opções comuns de montagem

Opção	Descrição
<b>async</b>	Permite operações assíncronas de entrada e saída no sistema de arquivo.
<b>auto</b>	Permite que o sistema de arquivo seja montado automaticamente usando o comando <b>mount -a</b> .
<b>defaults</b>	Fornecer um pseudônimo para as opções do site <b>async,auto,dev,exec,nouser,rw,suid</b> .
<b>exec</b>	Permite a execução de arquivos binários no sistema de arquivos em particular.
<b>loop</b>	Monta uma imagem como um dispositivo de loop.

Opção	Descrição
<b>noauto</b>	O comportamento padrão desabilita a montagem automática do sistema de arquivo usando o comando <b>mount -a</b> .
<b>noexec</b>	Não permite a execução de arquivos binários no sistema de arquivos em particular.
<b>nouser</b>	Não permite que um usuário comum (ou seja, que não seja root) monte e desmonte o sistema de arquivos.
<b>remount</b>	Remonta o sistema de arquivo caso ele já esteja montado.
<b>ro</b>	Monta o sistema de arquivo apenas para leitura.
<b>rw</b>	Monta o sistema de arquivo tanto para leitura quanto para escrita.
<b>user</b>	Permite que um usuário comum (ou seja, que não seja root) monte e desmonte o sistema de arquivos.

## 14.7. COMPARTILHAR UMA MONTAGEM EM VÁRIOS PONTOS DE MONTAGEM

Como administrador de sistemas, você pode duplicar os pontos de montagem para tornar os sistemas de arquivos acessíveis a partir de múltiplos diretórios.

### 14.7.1. Tipos de montagens compartilhadas

Há vários tipos de suportes compartilhados que você pode usar. A diferença entre eles é o que acontece quando você monta outro sistema de arquivo sob um dos pontos de montagem compartilhados. As montagens compartilhadas são implementadas usando a funcionalidade *shared subtrees*.

Os tipos são:

#### Montagem privada

Este tipo não recebe nem encaminha nenhum evento de propagação.

Quando você monta outro sistema de arquivo sob o ponto de montagem duplicado ou original, ele não é refletido no outro.

#### Montagem compartilhada

Este tipo cria uma réplica exata de um determinado ponto de montagem.

Quando um ponto de montagem é marcado como uma montagem compartilhada, qualquer montagem dentro do ponto de montagem original é refletida nele, e vice versa.

Este é o tipo padrão de montagem do sistema de arquivo raiz.

#### Montagem em escravos

Este tipo cria uma duplicata limitada de um determinado ponto de montagem.

Quando um ponto de montagem é marcado como uma montagem escrava, qualquer montagem dentro do ponto de montagem original é refletida nele, mas nenhuma montagem dentro de uma montagem escrava é refletida em seu original.

### Montagem sem fixação

Este tipo evita que o ponto de montagem em questão seja duplicado.

## 14.7.2. Criação de um duplicado de ponto de montagem privado

Este procedimento duplica um ponto de montagem como uma montagem privada. Os sistemas de arquivo que você depois monta sob o duplicado ou o ponto de montagem original não são refletidos no outro.

### Procedimento

1. Criar um nó de sistema de arquivo virtual (VFS) a partir do ponto de montagem original:

```
# montar --fixar original-dir original-dir
```

2. Marcar o ponto de montagem original como privado:

```
# montar -- fazer-privado original-dir
```

Alternativamente, para mudar o tipo de montagem para o ponto de montagem selecionado e todos os pontos de montagem abaixo dele, use a opção **--make-rprivate** ao invés de **--make-private**.

3. Criar o duplicado:

```
# montar --fixar original-dir duplicate-dir
```

### Exemplo 14.5. Duplicação /media em /mnt como um ponto de montagem privado

1. Criar um nó VFS a partir do diretório **/media**:

```
# montagem --bind /media /media
```

2. Marque o diretório **/media** como privado:

```
# montar -- fazer-private /media
```

3. Crie sua duplicata em **/mnt**:

```
# montagem --bind /media /mnt
```

4. Agora é possível verificar que **/media** e **/mnt** compartilham conteúdo, mas nenhuma das montagens dentro de **/media** aparece em **/mnt**. Por exemplo, se a unidade de CD-ROM contiver mídia não vazia e o diretório **/media/cdrom/** existir, use:

```
# mount /dev/cdrom /media/cdrom
# ls /media/cdrom
```

```
EFI GPL isolinux LiveOS
# ls /mnt/cdrom
#
```

5. Também é possível verificar se os sistemas de arquivo montados no diretório **/mnt** não estão refletidos em **/media**. Por exemplo, se uma unidade flash USB não vazia que usa o dispositivo **/dev/sdc1** estiver conectada e o diretório **/mnt/flashdisk/** estiver presente, use:

```
# mount /dev/sdc1 /mnt/flashdisk
# ls /media/flashdisk
# ls /mnt/flashdisk
en-US publican.cfg
```

## Recursos adicionais

- A página do homem **mount(8)**.

### 14.7.3. Criação de um duplicado de ponto de montagem compartilhado

Este procedimento duplica um ponto de montagem como uma montagem compartilhada. Os sistemas de arquivo que você posteriormente monta sob o diretório original ou a duplicata são sempre refletidos no outro.

#### Procedimento

1. Criar um nó de sistema de arquivo virtual (VFS) a partir do ponto de montagem original:

```
# montar --fixar original-dir original-dir
```

2. Marcar o ponto de montagem original como compartilhado:

```
# montar -- fazer compartilhado original-dir
```

Alternativamente, para mudar o tipo de montagem para o ponto de montagem selecionado e todos os pontos de montagem abaixo dele, use a opção **--make-rshared** ao invés de **--make-shared**.

3. Criar o duplicado:

```
# montar --fixar original-dir duplicate-dir
```

#### Exemplo 14.6. Duplicação **/media** em **/mnt** como um ponto de montagem compartilhado

Para que os diretórios **/media** e **/mnt** compartilhem o mesmo conteúdo:

1. Criar um nó VFS a partir do diretório **/media**:

```
# montagem --bind /media /media
```

2. Marque o diretório **/media** como compartilhado:

```
# montagem --make-shared /media
```

- 3. Crie sua duplicata em **/mnt**:

```
# montagem --bind /media /mnt
```

- 4. Agora é possível verificar que uma montagem dentro de **/media** também aparece em **/mnt**. Por exemplo, se a unidade de CD-ROM contiver mídia não vazia e o diretório **/media/cdrom/** existir, use:

```
# mount /dev/cdrom /media/cdrom
# ls /media/cdrom
EFI GPL isolinux LiveOS
# ls /mnt/cdrom
EFI GPL isolinux LiveOS
```

- 5. Da mesma forma, é possível verificar que qualquer sistema de arquivo montado no diretório **/mnt** é refletido em **/media**. Por exemplo, se uma unidade flash USB não vazia que usa o dispositivo **/dev/sdc1** estiver conectada e o diretório **/mnt/flashdisk/** estiver presente, use:

```
# mount /dev/sdc1 /mnt/flashdisk
# ls /media/flashdisk
en-US publican.cfg
# ls /mnt/flashdisk
en-US publican.cfg
```

### Recursos adicionais

- A página do homem **mount(8)**.

#### 14.7.4. Criação de um duplicado de ponto de montagem de escravos

Este procedimento duplica um ponto de montagem como uma montagem de escravos. Os sistemas de arquivo que você mais tarde monta sob o ponto de montagem original são refletidos na duplicata, mas não o contrário.

#### Procedimento

1. Criar um nó de sistema de arquivo virtual (VFS) a partir do ponto de montagem original:

```
# montar --fixar original-dir original-dir
```

2. Marcar o ponto de montagem original como compartilhado:

```
# montar -- fazer compartilhado original-dir
```

Alternativamente, para mudar o tipo de montagem para o ponto de montagem selecionado e todos os pontos de montagem abaixo dele, use a opção **--make-rshared** ao invés de **--make-shared**.

3. Criar o duplicado e marcá-lo como escravo:

```
# mount --bind original-dir duplicate-dir
# mount --make-slave duplicate-dir
```

### Exemplo 14.7. Duplicação /media em /mnt como um ponto de montagem de escravos

Este exemplo mostra como obter o conteúdo do diretório **/media** para aparecer também em **/mnt**, mas sem nenhuma montagem no diretório **/mnt** para ser refletido em **/media**.

1. Criar um nó VFS a partir do diretório **/media**:

```
# montagem --bind /media /media
```

2. Marque o diretório **/media** como compartilhado:

```
# montagem --make-shared /media
```

3. Crie seu duplicado em **/mnt** e marque-o como escravo:

```
# mount --bind /media /mnt
# mount --make-slave /mnt
```

4. Verifique se uma montagem dentro de **/media** também aparece em **/mnt**. Por exemplo, se a unidade de CD-ROM contiver mídia não vazia e o diretório **/media/cdrom/** existir, use:

```
# mount /dev/cdrom /media/cdrom
# ls /media/cdrom
EFI GPL isolinux LiveOS
# ls /mnt/cdrom
EFI GPL isolinux LiveOS
```

5. Verifique também se os sistemas de arquivo montados no diretório **/mnt** não estão refletidos em **/media**. Por exemplo, se uma unidade flash USB não vazia que usa o dispositivo **/dev/sdc1** estiver conectada e o diretório **/mnt/flashdisk/** estiver presente, use:

```
# mount /dev/sdc1 /mnt/flashdisk
# ls /media/flashdisk
# ls /mnt/flashdisk
en-US publican.cfg
```

#### Recursos adicionais

- A página do homem **mount(8)**.

### 14.7.5. Impedindo que um ponto de montagem seja duplicado

Este procedimento marca um ponto de montagem como não obrigatório, de modo que não é possível duplicá-lo em outro ponto de montagem.

#### Procedimento

- Para mudar o tipo de um ponto de montagem para uma montagem não vinculável, use:

```
# mount --bind mount-point mount-point
# mount --make-unbindable mount-point
```

Alternativamente, para mudar o tipo de montagem para o ponto de montagem selecionado e todos os pontos de montagem abaixo dele, use a opção **--make-runbindable** ao invés de **--make-unbindable**.

Qualquer tentativa subsequente de fazer um duplicado desta montagem falha com o seguinte erro:

```
# mount --bind mount-point duplicate-dir

mount: wrong fs type, bad option, bad superblock on mount-point,
missing codepage or helper program, or other error
In some cases useful info is found in syslog - try
dmesg | tail or so
```

### Exemplo 14.8. Impedindo a duplicação de mídia

- Para evitar que o diretório **/media** seja compartilhado, use:

```
# mount --bind /media /media
# mount --make-unbindable /media
```

#### Recursos adicionais

- A página do homem **mount(8)**.

#### 14.7.6. Informações relacionadas

- O artigo *Shared subtrees* no Linux Weekly News: <https://lwn.net/Articles/159077/>.

## 14.8. MONTAGEM PERSISTENTE DE SISTEMAS DE ARQUIVO

Como administrador de sistemas, você pode montar persistentemente sistemas de arquivos para configurar o armazenamento não removível.

### 14.8.1. O arquivo **/etc/fstab**

Esta seção descreve o arquivo de configuração **/etc/fstab**, que controla os pontos de montagem persistentes dos sistemas de arquivo. Usar **/etc/fstab** é a maneira recomendada para montar sistemas de arquivo de forma persistente.

Cada linha no arquivo **/etc/fstab** define um ponto de montagem de um sistema de arquivo. Ele inclui seis campos separados por espaço em branco:

1. O dispositivo de bloco identificado por um atributo persistente ou por um caminho, é o diretório **/dev**.
2. O diretório onde o dispositivo será montado.
3. O sistema de arquivo no dispositivo.

4. Opções de montagem para o sistema de arquivo. A opção **defaults** significa que a partição é montada no momento da inicialização com opções padrão. Esta seção também reconhece **systemd** opções de unidade de montagem no arquivo **x-systemd.option** formato.
5. Opção de backup para a utilidade **dump**.
6. Verifique o pedido para a utilidade **fsck**.

#### Exemplo 14.9. O sistema de arquivo/boot em /etc/fstab

Dispositivo de bloqueio	Ponto de montagem	Sistema de arquivo	Opções	Cópia de segurança	Verifique
UUID=ea74b bec-536d- 490c-b8d9- 5b40bbd754 5b	/boot	xfs	defaults	0	0

O serviço **systemd** gera automaticamente unidades de montagem a partir das entradas em **/etc/fstab**.

#### Recursos adicionais

- A página do homem **fstab(5)**.
- A seção *fstab* da página de manual **systemd.mount(5)**.

### 14.8.2. Adicionando um sistema de arquivo ao /etc/fstab

Este procedimento descreve como configurar o ponto de montagem persistente para um sistema de arquivo no arquivo de configuração **/etc/fstab**.

#### Procedimento

1. Descubra o atributo UUID do sistema de arquivo:

```
$ lsblk --fs storage-device
```

Por exemplo:

#### Exemplo 14.10. Visualizando a UUID de uma divisória

```
$ lsblk --fs /dev/sda1
```

```
NAME FSTYPE LABEL UUID MOUNTPOINT
sda1 xfs Boot ea74bbec-536d-490c-b8d9-5b40bbd7545b /boot
```

2. Se o diretório de pontos de montagem não existir, crie-o:

```
# mkdir -p /mount-point
```

- 
- Como root, editar o arquivo `/etc/fstab` e adicionar uma linha para o sistema de arquivo, identificado pela UUID.

Por exemplo:

#### Exemplo 14.11. O ponto de montagem `/boot` em `/etc/fstab`

```
UUID=ea74bbec-536d-490c-b8d9-5b40bbd7545b /boot xfs defaults 0 0
```

- Regenere unidades de montagem para que seu sistema registre a nova configuração:

```
# systemctl daemon-reload
```

- Tente montar o sistema de arquivo para verificar se a configuração funciona:

```
# montar mount-point
```

#### Recursos adicionais

- Outros atributos persistentes que você pode usar para identificar o sistema de arquivo [Seção 9.3, "Nomes de dispositivos gerenciados pelo mecanismo udev em `/dev/disco/`"](#)

### 14.8.3. Montagem persistente de um sistema de arquivo usando os papéis do sistema RHEL

Esta seção descreve como montar persistentemente um sistema de arquivo usando a função **storage**.

#### Pré-requisitos

- Existe um livro de brincadeiras possível que utiliza o papel **storage**. Para informações sobre como aplicar tal playbook, consulte [Aplicando um papel](#).

#### 14.8.3.1. 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 14.12. 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**.

#### 14.8.3.2. Recursos adicionais

- Para mais informações sobre a função **storage**, ver [Seção 2.1, “Introdução à função de armazenamento”](#).

## 14.9. MONTAGEM DE SISTEMAS DE ARQUIVOS SOB DEMANDA

Como administrador de sistemas, você pode configurar sistemas de arquivo, como o NFS, para montar automaticamente sob demanda.

### 14.9.1. O serviço autofs

Esta seção explica os benefícios e conceitos básicos do serviço **autofs**, utilizado para montar sistemas de arquivos sob demanda.

Uma desvantagem da montagem permanente usando a configuração **/etc/fstab** é que, independentemente de quão raramente um usuário acessa o sistema de arquivo montado, o sistema deve dedicar recursos para manter o sistema de arquivo montado no lugar. Isto pode afetar o desempenho do sistema quando, por exemplo, o sistema está mantendo a montagem do NFS em muitos sistemas ao mesmo tempo.

Uma alternativa ao **/etc/fstab** é usar o serviço **autofs** baseado no kernel. Ele consiste nos seguintes componentes:

- Um módulo de kernel que implementa um sistema de arquivo, e
- Um serviço de espaço do usuário que desempenha todas as outras funções.

O serviço **autofs** pode montar e desmontar sistemas de arquivos automaticamente (on-demand), economizando assim recursos do sistema. Pode ser usado para montar sistemas de arquivo como NFS, AFS, SMBFS, CIFS, e sistemas de arquivo locais.

### Recursos adicionais

- A página do homem **autofs(8)**.

#### 14.9.2. Os arquivos de configuração autofs

Esta seção descreve o uso e a sintaxe dos arquivos de configuração utilizados pelo serviço **autofs**.

### O arquivo do mapa principal

O serviço **autofs** usa **/etc/auto.master** (mapa mestre) como seu arquivo primário de configuração padrão. Isto pode ser alterado para usar outra fonte e nome de rede suportados usando a configuração **autofs** no arquivo de configuração **/etc/autofs.conf**, em conjunto com o mecanismo Name Service Switch (NSS).

Todos os pontos de montagem sob demanda devem ser configurados no mapa mestre. Pontos de montagem, nome do host, diretório exportado e opções podem ser todos especificados em um conjunto de arquivos (ou outras fontes de rede suportadas) em vez de configurá-los manualmente para cada host.

O arquivo do mapa mestre lista pontos de montagem controlados por **autofs**, e seus arquivos de configuração correspondentes ou fontes de rede conhecidas como mapas de montagem automática. O formato do mapa mestre é o seguinte:

```
mount-point map-name options
```

As variáveis utilizadas neste formato são:

#### **mount-point**

O ponto de montagem **autofs**; por exemplo, **/mnt/data/**.

#### **map-file**

O arquivo fonte do mapa, que contém uma lista de pontos de montagem e a localização do sistema de arquivo a partir do qual esses pontos de montagem devem ser montados.

#### **options**

Se fornecidas, estas se aplicam a todas as entradas no mapa dado, se elas próprias não tiverem opções especificadas.

### Exemplo 14.13. O arquivo **/etc/auto.master**

A seguir, uma linha de exemplo do arquivo **/etc/auto.master**:

```
/mnt/data /etc/auto.data
```

### Arquivos de mapas

Os arquivos de mapas configuram as propriedades de pontos de montagem individuais sob demanda.

O montador cria os diretórios se eles não existirem. Se os diretórios existirem antes do início do montador, o montador não os removerá ao sair. Se for especificado um timeout, o diretório será automaticamente desmontado se o diretório não for acessado durante o período de timeout.

O formato geral dos mapas é semelhante ao do mapa mestre. Entretanto, o campo de opções aparece entre o ponto de montagem e o local em vez de no final da entrada, como no mapa mestre:

```
mount-point options location
```

As variáveis utilizadas neste formato são:

#### **mount-point**

Isto se refere ao ponto de montagem **autofs**. Pode ser um único nome de diretório para uma

montagem indireta ou o caminho completo do ponto de montagem para montagens diretas. Cada chave de entrada de mapa direta e indireta (*mount-point*) pode ser seguido por uma lista separada por espaço de diretórios offset (nomes de subdiretórios cada um começando com /) tornando-os o que é conhecido como uma entrada multi-montante.

### options

Quando fornecidas, estas são as opções de montagem para as entradas do mapa que não especificam suas próprias opções. Este campo é opcional.

### location

Isto se refere à localização do sistema de arquivo, como um caminho do sistema de arquivo local (precedido pelo caractere de escape em formato de mapa Sun : para nomes de mapas começando com /), um sistema de arquivo NFS ou outra localização válida do sistema de arquivo.

## Exemplo 14.14. Um arquivo de mapa

O seguinte é uma amostra de um arquivo de mapa; por exemplo, **/etc/auto.misc**:

```
payroll -fstype=nfs4 personnel:/dev/disk/by-uuid/52b94495-e106-4f29-b868-fe6f6c2789b1
sales -fstype=xfss /dev/disk/by-uuid/5564ed00-6aac-4406-bfb4-c59bf5de48b5
```

A primeira coluna no arquivo do mapa indica o ponto de montagem **autofs**: **sales** e **payroll** do servidor chamado **personnel**. A segunda coluna indica as opções para o ponto de montagem **autofs**. A terceira coluna indica a origem da montagem.

Seguindo a configuração dada, os pontos de montagem **autofs** serão **/home/payroll** e **/home/sales**. A opção **-fstype=** é freqüentemente omitida e geralmente não é necessária para o funcionamento correto.

Usando a configuração dada, se um processo exigir acesso a um diretório **autofs** não montado como **/home/payroll/2006/July.sxc**, o serviço **autofs** monta automaticamente o diretório.

### O formato amd map

O serviço **autofs** reconhece a configuração do mapa também no formato **amd**. Isto é útil se você quiser reutilizar a configuração de montadora existente escrita para o serviço **am-utils**, que foi removida do Red Hat Enterprise Linux.

Entretanto, a Red Hat recomenda o uso do formato **autofs** mais simples descrito nas seções anteriores.

### Recursos adicionais

- As páginas de manual **autofs(5)**, **autofs.conf(5)**, e **auto.master(5)**.
- Para detalhes sobre o formato do mapa **amd**, consulte o arquivo **/usr/share/doc/autofs/README.amd-maps**, que é fornecido pelo pacote **autofs**.

## 14.9.3. Configuração de pontos de montagem autofs

Este procedimento descreve como configurar pontos de montagem sob demanda usando o serviço **autofs**.

### Pré-requisitos

- Instale o pacote **autofs**:

■

```
# yum instalar autofs
```

- Inicie e habilite o serviço **autofs**:

```
# systemctl habilita --agora autofs
```

### Procedimento

1. Criar um arquivo de mapa para o ponto de montagem sob demanda, localizado em **/etc/auto.identified**. Substituir *identified* com um nome que identifica o ponto de montagem.
2. No arquivo do mapa, preencha os campos de ponto de montagem, opções e localização, conforme descrito em [Seção 14.9.2, "Os arquivos de configuração autofs"](#).
3. Registre o arquivo de mapa no arquivo de mapa principal, como descrito em [Seção 14.9.2, "Os arquivos de configuração autofs"](#).
4. Tente acessar conteúdo no diretório on-demand:

```
$ ls automounted-directory
```

### 14.9.4. Montagem automática dos diretórios domésticos dos usuários do servidor NFS com o serviço autofs

Este procedimento descreve como configurar o serviço autofs para montar automaticamente os diretórios residenciais dos usuários.

#### Pré-requisitos

- O **autofs** pacote é instalado.
- O **autofs** serviço está habilitado e em funcionamento.

### Procedimento

1. Especifique o ponto de montagem e a localização do arquivo de mapa editando o arquivo **/etc/auto.master** em um servidor no qual você precisa montar os diretórios home do usuário. Para fazer isso, adicione a seguinte linha no arquivo **/etc/auto.master**:

```
/home /etc/auto.home
```

2. Crie um arquivo de mapa com o nome de **/etc/auto.home** em um servidor no qual você precisa montar diretórios home do usuário, e edite o arquivo com os seguintes parâmetros

```
* -fstype=nfs,rw,sync host.example.com:/home/&i
```

Você pode pular **fstype** parâmetro, como é **nfs** por padrão. Para mais informações, consulte a página de manual **autofs(5)**.

3. Recarregue o serviço **autofs**:

```
# systemctl reload autofs
```

### 14.9.5. Substituição ou aumento dos arquivos de configuração do site Autofs

Às vezes é útil sobrepor-se aos padrões do local para um ponto de montagem específico em um sistema cliente.

#### Exemplo 14.15. Condições iniciais

Por exemplo, considere as seguintes condições:

- Os mapas de montadoras são armazenados no NIS e o arquivo **/etc/nsswitch.conf** tem a seguinte diretiva:

```
automount: files nis
```

- O arquivo **auto.master** contém:

```
auto.master
```

- O arquivo de mapa do NIS **auto.master** contém:

```
/home auto.home
```

- O mapa do NIS **auto.home** contém:

```
beth fileserver.example.com:/export/home/beth
joe fileserver.example.com:/export/home/joe
* fileserver.example.com:/export/home/&
```

- O mapa do arquivo **/etc/auto.home** não existe.

#### Exemplo 14.16. Montagem de diretórios domésticos a partir de um servidor diferente

Dadas as condições anteriores, vamos supor que o sistema do cliente precisa anular o mapa NIS **auto.home** e montar diretórios domésticos a partir de um servidor diferente.

- Neste caso, o cliente precisa usar o seguinte mapa **/etc/auto.master**:

```
/home /etc/auto.home
+auto.master
```

- O mapa **/etc/auto.home** contém a entrada:

```
* host.example.com:/export/home/&
```

Como o automontador só processa a primeira ocorrência de um ponto de montagem, o diretório **/home** contém o conteúdo de **/etc/auto.home** em vez do mapa NIS **auto.home**.

#### Exemplo 14.17. Aumentar o auto.home com apenas entradas selecionadas

Alternativamente, para aumentar o mapa do site **auto.home** com apenas algumas entradas:

1. Criar um mapa de arquivo **/etc/auto.home**, e nele colocar as novas entradas. No final, inclua o mapa **auto.home** do NIS. Em seguida, o mapa de arquivo **/etc/auto.home** parece semelhante a:

```
mydir someserver:/export/mydir
+auto.home
```

2. Com estas condições do mapa NIS **auto.home**, listando o conteúdo das saídas do diretório **/home**:

```
$ ls /home

beth joe mydir
```

Este último exemplo funciona como esperado porque **autofs** não inclui o conteúdo de um mapa de arquivo com o mesmo nome que o que está lendo. Como tal, **autofs** passa para a próxima fonte do mapa na configuração **nsswitch**.

### 14.9.6. Usando o LDAP para armazenar mapas de montadoras

Este procedimento configura **autofs** para armazenar mapas de montadores automáticos na configuração LDAP em vez de em **autofs** arquivos de mapas.

#### Pré-requisitos

- As bibliotecas de clientes LDAP devem ser instaladas em todos os sistemas configurados para recuperar os mapas de montagem automática do LDAP. No Red Hat Enterprise Linux, o pacote **openldap** deve ser instalado automaticamente como uma dependência do pacote **autofs**.

#### Procedimento

1. Para configurar o acesso ao LDAP, modifique o arquivo **/etc/openldap/ldap.conf**. Certifique-se de que as opções **BASE**, **URI**, e **schema** estejam definidas apropriadamente para seu site.
2. O esquema mais recentemente estabelecido para o armazenamento de mapas automotivos no LDAP é descrito pelo rascunho **rfc2307bis**. Para usar este esquema, defina-o no arquivo de configuração **/etc/autofs.conf** removendo os caracteres de comentário da definição do esquema. Por exemplo:

#### Exemplo 14.18. Configuração de autofs

```
DEFAULT_MAP_OBJECT_CLASS="automountMap"
DEFAULT_ENTRY_OBJECT_CLASS="automount"
DEFAULT_MAP_ATTRIBUTE="automountMapName"
DEFAULT_ENTRY_ATTRIBUTE="automountKey"
DEFAULT_VALUE_ATTRIBUTE="automountInformation"
```

3. Certifique-se de que todas as outras entradas do esquema sejam comentadas na configuração. O atributo **automountKey** substitui o atributo **cn** no esquema **rfc2307bis**. A seguir, um exemplo de uma configuração LDAP Data Interchange Format (LDIF):

#### Exemplo 14.19. Configuração LDF

```
# extended LDIF
#
# LDAPv3
# base <> with scope subtree
# filter: (&(objectclass=automountMap)(automountMapName=auto.master))
# requesting: ALL
#

# auto.master, example.com
dn: automountMapName=auto.master,dc=example,dc=com
objectClass: top
objectClass: automountMap
automountMapName: auto.master

# extended LDIF
#
# LDAPv3
# base <automountMapName=auto.master,dc=example,dc=com> with scope subtree
# filter: (objectclass=automount)
# requesting: ALL
#

# /home, auto.master, example.com
dn: automountMapName=auto.master,dc=example,dc=com
objectClass: automount
cn: /home

automountKey: /home
automountInformation: auto.home

# extended LDIF
#
# LDAPv3
# base <> with scope subtree
# filter: (&(objectclass=automountMap)(automountMapName=auto.home))
# requesting: ALL
#

# auto.home, example.com
dn: automountMapName=auto.home,dc=example,dc=com
objectClass: automountMap
automountMapName: auto.home

# extended LDIF
#
# LDAPv3
# base <automountMapName=auto.home,dc=example,dc=com> with scope subtree
# filter: (objectclass=automount)
# requesting: ALL
#

# foo, auto.home, example.com
dn: automountKey=foo,automountMapName=auto.home,dc=example,dc=com
objectClass: automount
automountKey: foo
automountInformation: filer.example.com:/export/foo
```

```
# /, auto.home, example.com
dn: automountKey=/,automountMapName=auto.home,dc=example,dc=com
objectClass: automount
automountKey: /
automountInformation: filer.example.com:/export/&
```

## Recursos adicionais

- O rascunho **rfc2307bis**: <https://tools.ietf.org/html/draft-howard-rfc2307bis>.

## 14.10. DEFINIÇÃO DE PERMISSÕES SOMENTE LEITURA PARA O SISTEMA DE ARQUIVOS RAIZ

Às vezes, é necessário montar o sistema de arquivos raiz (/) com permissões somente de leitura. Exemplos de casos de uso incluem o aumento da segurança ou a garantia da integridade dos dados após um desligamento inesperado do sistema.

### 14.10.1. Arquivos e diretórios que sempre retêm permissões de escrita

Para que o sistema funcione corretamente, alguns arquivos e diretórios precisam reter as permissões de escrita. Quando o sistema de arquivos raiz é montado em modo somente leitura, estes arquivos são montados na RAM usando o sistema de arquivos temporários **tmpfs**.

O conjunto padrão de tais arquivos e diretórios é lido do arquivo **/etc/rwtab**, que contém:

```
dirs /var/cache/man
dirs /var/gdm
<content truncated>

empty /tmp
empty /var/cache/foomatic
<content truncated>

files /etc/adjtime
files /etc/ntp.conf
<content truncated>
```

As entradas no arquivo **/etc/rwtab** seguem este formato:

```
copy-method path
```

Nesta sintaxe:

- Substitua *copy-method* com uma das palavras-chave especificando como o arquivo ou diretório é copiado para tmpfs.
- Substitua *path* com o caminho para o arquivo ou diretório.

O arquivo **/etc/rwtab** reconhece as seguintes formas nas quais um arquivo ou diretório pode ser copiado para **tmpfs**:

#### **empty**

Um caminho vazio é copiado para **tmpfs**. Por exemplo:

```
vazio /tmp
```

### dirs

Uma árvore de diretórios é copiada para **tmpfs**, vazia. Por exemplo, uma árvore de diretório é copiada para , vazia:

```
dirs /var/run
```

### files

Um arquivo ou uma árvore de diretório é copiado para **tmpfs** intacto. Por exemplo:

```
arquivos /etc/resolv.conf
```

O mesmo formato se aplica ao adicionar caminhos personalizados para **/etc/rwtab.d/**.

## 14.10.2. Configuração do sistema de arquivo raiz para montagem com permissões somente leitura na inicialização

Com este procedimento, o sistema de arquivo raiz é montado somente para leitura em todas as botas seguintes.

### Procedimento

1. No arquivo **/etc/sysconfig/readonly-root**, defina a opção **READONLY** para **yes**:

```
# Set to 'yes' to mount the file systems as read-only.  
READONLY=yes
```

2. Adicione a opção **ro** na entrada raiz (/) no arquivo **/etc/fstab**:

```
/dev/mapper/luks-c376919e... / xfs x-systemd.device-timeout=0,ro 1 1
```

3. Adicione a opção **ro** à diretiva **GRUB\_CMDLINE\_LINUX** no arquivo **/etc/default/grub** e certifique-se de que a diretiva não contenha **rw**:

```
GRUB_CMDLINE_LINUX=="rhgb quiet... ro"
```

4. Recriar o arquivo de configuração do GRUB2:

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

5. Se você precisar adicionar arquivos e diretórios para serem montados com permissões de escrita no sistema de arquivos **tmpfs**, crie um arquivo de texto no diretório **/etc/rwtab.d/** e coloque a configuração lá.

Por exemplo, para montar o arquivo **/etc/example/file** com permissões de escrita, adicione esta linha ao arquivo **/etc/rwtab.d/example**:

```
arquivos /etc/exemplo/arquivo
```



## IMPORTANTE

As mudanças feitas nos arquivos e diretórios em **tmpfs** não persistem através das botas.

6. Reinicie o sistema para aplicar as mudanças.

### Solução de problemas

- Se você montar o sistema de arquivo raiz com permissões somente de leitura por engano, você pode montá-lo novamente com permissões de leitura e escrita usando o seguinte comando:

```
# montar -o remount,rw /
```

## CAPÍTULO 15. LIMITANDO O USO DO ESPAÇO DE ARMAZENAMENTO COM COTAS

Você pode restringir a quantidade de espaço em disco disponível para usuários ou grupos, implementando cotas em disco. Você também pode definir um nível de alerta no qual os administradores do sistema são informados antes que um usuário consuma muito espaço em disco ou que uma partição fique cheia.

### 15.1. COTAS DE DISCOS

Na maioria dos ambientes computacionais, o espaço em disco não é infinito. O subsistema de quotas fornece um mecanismo para controlar o uso do espaço em disco.

Você pode configurar cotas de disco para usuários individuais, bem como grupos de usuários nos sistemas de arquivos locais. Isto torna possível gerenciar o espaço alocado para arquivos específicos do usuário (como e-mail) separadamente do espaço alocado para os projetos nos quais o usuário trabalha. O subsistema de cotas avisa os usuários quando eles excedem seu limite alocado, mas permite algum espaço extra para o trabalho atual (limite duro/limite suave).

Se as cotas forem implementadas, você precisa verificar se as cotas foram excedidas e certificar-se de que as cotas sejam precisas. Se os usuários excederem repetidamente suas cotas ou atingirem consistentemente seus limites de soft, um administrador de sistema pode ajudar o usuário a determinar como utilizar menos espaço em disco ou aumentar a cota de disco do usuário.

Você pode definir cotas a serem controladas:

- O número de blocos de disco consumidos.
- O número de inodes, que são estruturas de dados que contêm informações sobre arquivos em sistemas de arquivos UNIX. Como os inodes armazenam informações relacionadas a arquivos, isto permite o controle sobre o número de arquivos que podem ser criados.

#### 15.1.1. A ferramenta `xfs_quota`

Você pode usar a ferramenta `xfs_quota` para gerenciar cotas em sistemas de arquivos XFS. Além disso, você pode usar sistemas de arquivos XFS com aplicação de limites desligada como um sistema eficaz de contabilidade de uso de disco.

O sistema de quotas XFS difere de outros sistemas de arquivos de várias maneiras. Mais importante ainda, o XFS considera as informações de cota como metadados do sistema de arquivos e usa o jornalismo para fornecer um nível mais alto de garantia de consistência.

#### Recursos adicionais

- A página do homem `xfs_quota(8)`.

### 15.2. GERENCIANDO QUOTAS DE DISCOS XFS

Você pode usar a ferramenta `xfs_quota` para gerenciar quotas em XFS e para configurar limites para diretórios controlados por projeto.

Ferramentas genéricas de configuração de cotas (`quota`, `repquota`, e `edquota` por exemplo) também podem ser usadas para manipular as cotas XFS. Entretanto, estas ferramentas não podem ser usadas com cotas de projetos XFS.



## IMPORTANTE

A Red Hat recomenda o uso do site **xfs\_quota** sobre todas as outras ferramentas disponíveis.

### 15.2.1. Gestão de quotas do sistema de arquivos em XFS

O subsistema de quotas XFS gerencia os limites de espaço em disco (blocos) e uso de arquivos (inode). O controle de cotas XFS ou relatório sobre o uso desses itens em nível de usuário, grupo ou diretório ou projeto. As cotas de grupo e projeto só se excluem mutuamente em formatos de disco XFS mais antigos, que não sejam por defeito.

Ao gerenciar em uma base por diretório ou por projeto, o XFS gerencia o uso do disco de hierarquias de diretórios associadas a um projeto específico.

### 15.2.2. Habilitação de quotas de disco para XFS

Este procedimento permite quotas de disco para usuários, grupos e projetos em um sistema de arquivos XFS. Uma vez habilitadas as cotas, a ferramenta **xfs\_quota** pode ser usada para definir limites e relatórios sobre o uso do disco.

#### Procedimento

1. Habilitar cotas para os usuários:

```
# mount -o uquota /dev/xvdb1 /xfs
```

Substituir **uquota** por **uqnoenforce** para permitir relatórios de uso sem impor quaisquer limites.

2. Habilitar cotas para grupos:

```
# montagem -o gquota /dev/xvdb1 /xfs
```

Substituir **gquota** por **gqnoenforce** para permitir relatórios de uso sem impor quaisquer limites.

3. Habilitar cotas para projetos:

```
# montar -o pquota /dev/xvdb1 /xfs
```

Substituir **pquota** por **pqnoenforce** para permitir relatórios de uso sem impor quaisquer limites.

4. Alternativamente, inclua as opções de montagem de cotas no arquivo **/etc/fstab**. O exemplo a seguir mostra as entradas no arquivo **/etc/fstab** para permitir cotas para usuários, grupos e projetos, respectivamente, em um sistema de arquivo XFS. Estes exemplos também montam o sistema de arquivo com permissões de leitura/gravação:

```
# vim /etc/fstab
/dev/xvdb1 /xfs xfs rw,quota 0 0
/dev/xvdb1 /xfs xfs rw,gquota 0 0
/dev/xvdb1 /xfs xfs rw,prjquota 0 0
```

#### Recursos adicionais

- A página do homem **mount(8)**.

- A página do homem **xfs\_quota(8)**.

### 15.2.3. Relatório de uso de XFS

Você pode usar a ferramenta **xfs\_quota** para estabelecer limites e informar sobre o uso do disco. Por padrão, **xfs\_quota** é executado interativamente, e em modo básico. Os subcomandos do modo básico simplesmente reportam o uso, e estão disponíveis para todos os usuários.

#### Pré-requisitos

- As cotas foram habilitadas para o sistema de arquivos XFS. Veja [Habilitação de cotas de disco para XFS](#).

#### Procedimento

1. Inicie a casca **xfs\_quota**:

```
# xfs_quota
```

2. Mostrar uso e limites para o usuário em questão:

```
# xfs_quota> quota username
```

3. Mostrar contagens livres e usadas para blocos e inodes:

```
# xfs_quota> df
```

4. Execute o comando de ajuda para exibir os comandos básicos disponíveis com **xfs\_quota**.

```
# xfs_quota> ajuda
```

5. Especifique **q** para sair **xfs\_quota**.

```
# xfs_quota> q
```

#### Recursos adicionais

- A página do homem **xfs\_quota(8)**.

### 15.2.4. Modificando os limites de quotas XFS

Inicie a ferramenta **xfs\_quota** com a opção **-x** para habilitar o modo especialista e executar os comandos do administrador, que permitem modificações no sistema de cotas. Os subcomandos deste modo permitem a configuração real dos limites, e estão disponíveis apenas para usuários com privilégios elevados.

#### Pré-requisitos

- As cotas foram habilitadas para o sistema de arquivos XFS. Veja [Habilitação de cotas de disco para XFS](#).

#### Procedimento

1. Inicie o shell **xfs\_quota** com a opção **-x** para habilitar o modo especialista:

```
■
```

```
# xfs_quota -x
```

2. Relatar informações de cota para um sistema de arquivo específico:

```
# xfs_quota> relatório /path
```

Por exemplo, para exibir um exemplo de relatório de cota para **/home** (em **/dev/blockdevice**), use o comando **report -h /home**. Isto exibe resultados semelhantes aos seguintes:

```
User quota on /home (/dev/blockdevice)
Blocks
User ID    Used  Soft  Hard Warn/Grace
-----
root       0    0    0 00 [-----]
testuser 103.4G  0    0 00 [-----]
```

3. Modificar os limites das cotas:

```
# xfs_quota> limite isoft=500m ihard=700m user /path
```

Por exemplo, para definir um limite de contagem de 500 e 700 inode macio e duro respectivamente para o usuário **john**, cujo diretório home é **/home/john**, use o seguinte comando:

```
# xfs_quota -x -c 'limite isoft=500 ihard=700 john' /home/
```

Neste caso, passe **mount\_point** que é o sistema de arquivo xfs montado.

4. Execute o comando de ajuda para exibir os comandos especializados disponíveis com **xfs\_quota -x**:

```
# xfs_quota> ajuda
```

#### Recursos adicionais

- A página do homem **xfs\_quota(8)**.

### 15.2.5. Estabelecendo os limites do projeto para XFS

Este procedimento configura limites para diretórios controlados pelo projeto.

#### Procedimento

1. Adicione os diretórios controlados pelo projeto a **/etc/projects**. Por exemplo, o seguinte acrescenta o caminho **/var/log** com uma identificação única de 11 para **/etc/projects**. Seu ID de projeto pode ser qualquer valor numérico mapeado para seu projeto.

```
# echo 11:/var/log >> /etc/projects
```

2. Adicionar nomes de projetos a **/etc/projid** para mapear os IDs dos projetos a nomes de projetos. Por exemplo, os seguintes associam um projeto chamado **Logs** com o ID do projeto 11, conforme definido na etapa anterior.

```
# echo Logs:11 >> /etc/projid
```

- 
- 3. Inicializar o diretório do projeto. Por exemplo, o seguinte inicializa o diretório do projeto **/var**:

```
# xfs_quota -x -c 'projeto -s logfiles' /var
```

- 4. Configurar cotas para projetos com diretórios inicializados:

```
# xfs_quota -x -c 'limite -p bhard=lg logfiles' /var
```

#### Recursos adicionais

- A página do homem **xfs\_quota(8)**.
- A página do homem **projid(5)**.
- A página do homem **projects(5)**.

## 15.3. GERENCIANDO QUOTAS DE DISCOS EXT3 E EXT4

Você tem que ativar as cotas de disco em seu sistema antes de poder atribuí-las. Você pode atribuir cotas de disco por usuário, por grupo ou por projeto. Entretanto, se houver um limite suave definido, você pode exceder essas cotas por um período de tempo configurável, conhecido como o período de carência.

### 15.3.1. Instalando a ferramenta de cota

Você deve instalar o pacote **quota** RPM para implementar quotas de disco.

#### Procedimento

- Instale o pacote **quota**:

```
# yum quota de instalação
```

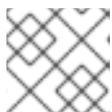
### 15.3.2. Possibilitando o recurso de cota na criação do sistema de arquivos

Este procedimento descreve como habilitar as cotas na criação de sistemas de arquivo.

#### Procedimento

1. Habilitar cotas na criação de sistemas de arquivos:

```
# mkfs.ext4 -O quota /dev/sda
```



#### NOTA

Somente quotas de usuários e grupos são ativadas e inicializadas por padrão.

2. Alterar os padrões na criação do sistema de arquivo:

```
# mkfs.ext4 -O quota -E quotatype=usrquota:grpquota:prjquota /dev/sda
```

3. Montar o sistema de arquivo:

```
# montar /dev/sda
```

### Recursos adicionais

Consulte a página **man** para **ext4** para obter informações adicionais.

### 15.3.3. Habilitação do recurso de cota nos sistemas de arquivo existentes

Este procedimento descreve como habilitar o recurso de cota no sistema de arquivos existente usando o comando **tune2fs**.

#### Procedimento

1. Desmontar o sistema de arquivo:

```
# umount /dev/sda
```

2. Habilitar cotas no sistema de arquivos existente:

```
# tune2fs -O quota /dev/sda
```



#### NOTA

Somente quotas de usuários e grupos são inicializadas por padrão.

3. Alterar as inadimplências:

```
# tune2fs -Q usrquota,grpquota,prjquota /dev/sda
```

4. Montar o sistema de arquivo:

```
# montar /dev/sda
```

### Recursos adicionais

Consulte a página **man** para **ext4** para obter informações adicionais.

### 15.3.4. Possibilitando a aplicação de cotas

A contabilização de cotas é ativada por padrão após a montagem do sistema de arquivos sem nenhuma opção adicional, mas a aplicação de cotas não o é.

#### Pré-requisitos

- O recurso de cotas é ativado e as cotas padrão são inicializadas.

#### Procedimento

- Habilitar a aplicação de cota por **quotaon** para a cota de usuários:

```
# montar /dev/sda /mnt
```

```
# cota /mnt
```



### NOTA

A aplicação de cotas pode ser ativada no momento da montagem usando **usrquota**, **grpquota**, ou **prjquota** opções de montagem.

```
# montar -o usrquota,grpquota,prjquota /dev/sda /mnt
```

- Permitir cotas de usuários, grupos e projetos para todos os sistemas de arquivos:

```
# quotaon -vaugP
```

- Se nenhuma das opções **-u**, **-g**, ou **-P** for especificada, somente as cotas de usuários serão habilitadas.
  - Se apenas a opção **-g** for especificada, apenas as quotas de grupo serão ativadas.
  - Se apenas a opção **-P** for especificada, apenas cotas de projetos serão ativadas.
- Habilitar quotas para um sistema de arquivo específico, como o **/home**:

```
# quotaon -vugP /home
```

### Recursos adicionais

- Veja a página de manual **quotaon(8)**.

### 15.3.5. Atribuição de cotas por usuário

As cotas em disco são atribuídas aos usuários com o comando **edquota**.



### NOTA

O editor de texto definido pela variável de ambiente **EDITOR** é utilizado por **edquota**. Para mudar o editor, defina a variável de ambiente **EDITOR** em seu arquivo **~/.bash\_profile** para o caminho completo do editor de sua escolha.

### Pré-requisitos

- O usuário deve existir antes de definir a cota de usuário.

### Procedimento

1. Atribuir a cota a um usuário:

```
# edquota username
```

Substitua *username* pelo usuário ao qual você deseja atribuir as cotas.

Por exemplo, se você ativar uma cota para a partição **/dev/sda** e executar o comando **edquota testuser**, o seguinte é exibido no editor padrão configurado no sistema:

```
Disk quotas for user testuser (uid 501):
Filesystem blocks soft hard inodes soft hard
/dev/sda 44043 0 0 37418 0 0
```

## 2. Alterar os limites desejados.

Se qualquer um dos valores for definido como 0, o limite não é definido. Modifique-os no editor de texto.

Por exemplo, o seguinte mostra os limites de blocos macios e rígidos para o usuário de teste foram definidos para 50000 e 55000, respectivamente.

```
Disk quotas for user testuser (uid 501):
Filesystem blocks soft hard inodes soft hard
/dev/sda 44043 50000 55000 37418 0 0
```

- A primeira coluna é o nome do sistema de arquivo que tem uma cota habilitada para ele.
- A segunda coluna mostra quantos blocos o usuário está utilizando atualmente.
- As duas colunas seguintes são usadas para definir limites de blocos macios e rígidos para o usuário no sistema de arquivo.
- A coluna **inodes** mostra quantos inodes o usuário está usando atualmente.
- As duas últimas colunas são usadas para definir os limites de inode macio e duro para o usuário no sistema de arquivo.
  - O limite do bloco rígido é a quantidade máxima absoluta de espaço em disco que um usuário ou grupo pode utilizar. Uma vez atingido este limite, nenhum outro espaço em disco pode ser utilizado.
  - O limite de blocos macios define a quantidade máxima de espaço em disco que pode ser utilizada. Entretanto, ao contrário do limite rígido, o limite macio pode ser excedido por um determinado período de tempo. Esse tempo é conhecido como o *grace period*. O período de carência pode ser expresso em segundos, minutos, horas, dias, semanas, ou meses.

## Etapas de verificação

- Verificar se a cota para o usuário foi definida:

```
# quota -v testuser
Disk quotas for user testuser:
Filesystem blocks quota limit grace files quota limit grace
/dev/sda 1000* 1000 1000 0 0 0
```

### 15.3.6. Atribuição de cotas por grupo

Você pode atribuir cotas por grupo.

## Pré-requisitos

- O grupo deve existir antes de estabelecer a cota do grupo.

### Procedimento

1. Estabelecer uma cota de grupo:

```
# edquota -g groupname
```

Por exemplo, para estabelecer uma cota de grupo para o grupo **devel**:

```
# edquota -g devel
```

Este comando exibe a cota existente para o grupo no editor de texto:

```
Disk quotas for group devel (gid 505):
Filesystem blocks soft hard inodes soft hard
/dev/sda 440400 0 0 37418 0 0
```

2. Modificar os limites e salvar o arquivo.

### Etapas de verificação

- Verificar se a cota do grupo está definida:

```
# quota -vg groupname
```

## 15.3.7. Atribuição de cotas por projeto

Este procedimento atribui cotas por projeto.

### Pré-requisitos

- A cota do projeto é habilitada em seu sistema de arquivos.

### Procedimento

1. Adicione os diretórios controlados pelo projeto a **/etc/projects**. Por exemplo, o seguinte acrescenta o caminho **/var/log** com uma identificação única de 11 para **/etc/projects**. Seu ID de projeto pode ser qualquer valor numérico mapeado para seu projeto.

```
# echo 11:/var/log >> /etc/projects
```

2. Adicionar nomes de projetos a **/etc/projid** para mapear os IDs dos projetos a nomes de projetos. Por exemplo, os seguintes associam um projeto chamado **Logs** com o ID do projeto 11, conforme definido na etapa anterior.

```
# echo Logs:11 >> /etc/projid
```

3. Estabelecer os limites desejados:

```
# edquota -P 11
```

**NOTA**

Você pode escolher o projeto por seu ID de projeto (**11** neste caso), ou por seu nome (**Logs** neste caso).

4. Usando **quotaon**, permitir a aplicação de cotas:  
Ver [Permitindo a aplicação de cotas](#).

**Etapas de verificação**

- Verificar se a cota do projeto está definida:

```
# quota -vP 11
```

**NOTA**

Você pode verificar tanto pelo ID do projeto, quanto pelo nome do projeto.

**Recursos adicionais**

- A página do homem **edquota(8)**.
- A página do homem **projid(5)**.
- A página do homem **projects(5)**.

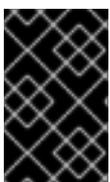
**15.3.8. Estabelecendo o período de carência para limites suaves**

Se uma determinada cota tem limites suaves, você pode editar o período de carência, que é a quantidade de tempo durante o qual um limite suave pode ser excedido. Você pode definir o período de carência para usuários, grupos ou projetos.

**Procedimento**

- Editar o período de carência:

```
# edquota -t
```

**IMPORTANTE**

Enquanto outros comandos **edquota** operam em cotas para um determinado usuário, grupo ou projeto, a opção **-t** opera em todos os sistemas de arquivos com cotas habilitadas.

**Recursos adicionais**

- A página do homem **edquota(8)**.

**15.3.9. Desligamento das quotas do sistema de arquivos**

Use **quotaoff** para desligar a aplicação de cota de disco nos sistemas de arquivo especificados. A contabilização das cotas permanece ativada após a execução deste comando.

## Procedimento

- Para desligar todas as quotas de usuários e grupos:

```
# quotaoff -vaugP
```

- Se nenhuma das opções **-u**, **-g**, ou **-P** for especificada, somente as cotas de usuários serão desativadas.
- Se apenas a opção **-g** for especificada, apenas as quotas de grupo serão desativadas.
- Se apenas a opção **-P** for especificada, apenas as cotas do projeto serão desativadas.
- A chave **-v** faz com que as informações de status verboso sejam exibidas conforme o comando executa.

## Recursos adicionais

- Veja a página de manual [quotaoff\(8\)](#).

### 15.3.10. Relatórios sobre quotas de discos

Você pode criar um relatório de cota de disco usando o utilitário **repquota**.

## Procedimento

1. Execute o comando **repquota**:

```
# repquota
```

Por exemplo, o comando **repquota /dev/sda** produz esta saída:

```
*** Report for user quotas on device /dev/sda
Block grace time: 7days; Inode grace time: 7days
Block limits  File limits
User  used soft hard grace used soft hard grace
-----
root  --   36   0   0         4   0   0
kristin --  540   0   0        125   0   0
testuser -- 440400 500000 550000    37418   0   0
```

2. Veja o relatório de utilização do disco para todos os sistemas de arquivos habilitados para cotas:

```
# repquota -augP
```

O símbolo **--** exibido após cada usuário determina se os limites do bloco ou inodo foram excedidos. Se um ou outro limite suave for excedido, um caractere  aparece no lugar do correspondente caracter **-**. O primeiro caracter **-** representa o limite do bloco, e o segundo representa o limite do inodo.

As colunas **grace** normalmente estão em branco. Se um limite suave foi excedido, a coluna contém uma especificação de tempo igual à quantidade de tempo restante no período de carência. Se o período de carência tiver expirado, **none** aparece em seu lugar.

## Recursos adicionais

A página de manual **repquota(8)** para mais informações.

## CAPÍTULO 16. DESCARTANDO BLOCOS NÃO UTILIZADOS

Você pode realizar ou programar operações de descarte em dispositivos de bloco que os suportem.

### 16.1. OPERAÇÕES DE DESCARTE EM BLOCO

As operações de descarte de blocos descartam blocos que não estão mais em uso por um sistema de arquivo montado. Eles são úteis:

- Acionamentos em estado sólido (SSDs)
- Armazenagem de provisão fina

#### Requisitos

O dispositivo de bloco subjacente ao sistema de arquivo deve suportar operações de descarte físico.

As operações de descarte físico são suportadas se o valor no `/sys/block/device/queue/discard_max_bytes` arquivo não é zero.

### 16.2. TIPOS DE OPERAÇÕES DE DESCARTE EM BLOCO

Você pode executar operações de descarte usando métodos diferentes:

#### Lote de descarte

São administrados explicitamente pelo usuário. Eles descartam todos os blocos não utilizados nos sistemas de arquivo selecionados.

#### Descarte on-line

São especificados no momento da montagem. Funcionam em tempo real, sem a intervenção do usuário. As operações de descarte on-line descartam apenas os blocos que estão passando de usados para livres.

#### Descarte periódico

São operações em lote que são administradas regularmente por um serviço **systemd**.

Todos os tipos são suportados pelos sistemas de arquivo XFS e ext4 e pela VDO.

#### Recomendações

A Red Hat recomenda que você utilize descarte em lote ou periódico.

Usar o descarte on-line somente se:

- a carga de trabalho do sistema é tal que o descarte de lotes não é viável, ou
- as operações de descarte on-line são necessárias para manter o desempenho.

### 16.3. EXECUÇÃO DE DESCARTE DE BLOCOS EM LOTE

Este procedimento realiza uma operação de descarte de blocos em lote para descartar blocos não utilizados em um sistema de arquivo montado.

#### Pré-requisitos

- O sistema de arquivo é montado.

- O dispositivo de bloco subjacente ao sistema de arquivo suporta operações de descarte físico.

### Procedimento

- Use o utilitário **fstrim**:
  - Para realizar descarte somente em um sistema de arquivo selecionado, use:

```
# fstrim mount-point
```

- Para realizar descarte em todos os sistemas de arquivos montados, use:

```
# fstrim -- tudo
```

Se você executar o comando **fstrim** em:

- um dispositivo que não suporta operações de descarte, ou
- um dispositivo lógico (LVM ou MD) composto de múltiplos dispositivos, onde qualquer um dos dispositivos não suporta operações de descarte,

a seguinte mensagem é exibida:

```
# fstrim /mnt/non_discard
```

```
fstrim: /mnt/non_discard: the discard operation is not supported
```

### Recursos adicionais

- A página do homem **fstrim(8)**

## 16.4. PERMITINDO O DESCARTE EM BLOCO ONLINE

Este procedimento permite operações de descarte de blocos on-line que descartam automaticamente blocos não utilizados em todos os sistemas de arquivo suportados.

### Procedimento

- Habilitar o descarte on-line no momento da montagem:
  - Ao montar um sistema de arquivo manualmente, adicione a opção de montagem **-o discard**:
 

```
# montar -o discard device mount-point
```
  - Ao montar um sistema de arquivo de forma persistente, adicione a opção **discard** à entrada de montagem no arquivo **/etc/fstab**.

### Recursos adicionais

- A página do homem **mount(8)**
- A página do homem **fstab(5)**

## 16.5. PERMITINDO O DESCARTE EM BLOCO ONLINE USANDO AS FUNÇÕES DO SISTEMA RHEL

Esta seção descreve como habilitar o descarte em bloco online usando a função **storage**.

### Pré-requisitos

- Existe um livro de brincadeiras possível, incluindo o papel **storage**.

Para informações sobre como aplicar tal playbook, consulte [Aplicando um papel](#).

### 16.5.1. 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 16.1. 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 [Seção 2.4, "Exemplo Livro de reprodução 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`.

### 16.5.2. Recursos adicionais

- Para mais informações sobre a função **storage**, ver [Seção 2.1, "Introdução à função de armazenamento"](#).

## 16.6. POSSIBILITANDO O DESCARTE PERIÓDICO EM BLOCO

Este procedimento permite um temporizador **systemd** que descarta regularmente blocos não utilizados em todos os sistemas de arquivo suportados.

## Procedimento

- Habilite e inicie o temporizador **systemd**:

```
# systemctl habilita --now fstrim.timer
```

## CAPÍTULO 17. GERENCIAMENTO DE ARMAZENAMENTO LOCAL EM CAMADAS COM O STRATIS

Você pode facilmente configurar e gerenciar configurações complexas de armazenamento integradas pelo sistema de alto nível Stratis.



### IMPORTANTE

O Stratis está disponível como uma Pré-visualização Tecnológica. Para informações sobre o escopo de suporte das características de Technology Preview da Red Hat, consulte o documento [Technology Preview Features Support Scope](#).

Os clientes que implantam Stratis são encorajados a fornecer feedback à Red Hat.

### 17.1. MONTAGEM DE SISTEMAS DE ARQUIVO STRATIS

Como administrador do sistema, você pode ativar e configurar o sistema de arquivo de gerenciamento de volume Stratis em seu sistema para gerenciar facilmente o armazenamento em camadas.

#### 17.1.1. O propósito e as características de Stratis

Stratis é uma solução de gerenciamento de armazenamento local para Linux. Ele é focado na simplicidade e facilidade de uso, e dá acesso a recursos avançados de armazenamento.

Stratis facilita as seguintes atividades:

- Configuração inicial de armazenamento
- Fazendo mudanças mais tarde
- Utilização de recursos avançados de armazenamento

O Stratis é um sistema híbrido de gerenciamento de armazenamento local de usuário e núcleo que suporta recursos avançados de armazenamento. O conceito central do Stratis é um sistema de armazenamento *pool*. Este *pool* é criado a partir de um ou mais discos ou partições locais, e os volumes são criados a partir do *pool*.

O *pool* permite muitas características úteis, como por exemplo:

- Instantâneos do sistema de arquivo
- Provisão fina
- Classificação

#### 17.1.2. Componentes de um volume Stratis

Externamente, Stratis apresenta os seguintes componentes de volume na interface de linha de comando e no API:

##### **blockdev**

Dispositivos de bloqueio, tais como uma partição de disco ou uma partição de disco.

##### **pool**

Composto por um ou mais dispositivos de bloco.

Uma piscina tem um tamanho total fixo, igual ao tamanho dos dispositivos do bloco.

O pool contém a maioria das camadas do Stratis, como o cache de dados não volátil, usando o alvo **dm-cache**.

Stratis cria um **/stratis/my-pool/** para cada pool. Este diretório contém links para dispositivos que representam os sistemas de arquivos Stratis no pool.

## filesystem

Cada pool pode conter um ou mais sistemas de arquivos, que armazenam arquivos.

Os sistemas de arquivo são pouco provisionados e não têm um tamanho total fixo. O tamanho real de um sistema de arquivo cresce com os dados nele armazenados. Se o tamanho dos dados se aproximar do tamanho virtual do sistema de arquivo, o Stratis aumenta o volume fino e o sistema de arquivo automaticamente.

Os sistemas de arquivo são formatados com XFS.



### IMPORTANTE

Stratis rastreia informações sobre sistemas de arquivo criados usando Stratis que o XFS não conhece, e as mudanças feitas usando o XFS não criam automaticamente atualizações no Stratis. Os usuários não devem reformatar ou reconfigurar sistemas de arquivos XFS que são gerenciados pelo Stratis.

Stratis cria links para sistemas de arquivo no **/stratis/my-pool/my-fs** caminho.



### NOTA

Stratis usa muitos dispositivos Device Mapper, que aparecem na lista **dmsetup** e no arquivo **/proc/partitions**. Da mesma forma, a saída do comando **lsblk** reflete o funcionamento interno e as camadas do Stratis.

## 17.1.3. Dispositivos de bloqueio utilizáveis com Stratis

Esta seção lista os dispositivos de armazenamento que você pode usar para o Stratis.

### Dispositivos com suporte

As piscinas Stratis foram testadas para trabalhar com estes tipos de dispositivos de blocos:

- LUKS
- LVM volumes lógicos
- MD RAID
- DM Multipath
- iSCSI
- HDDs e SSDs
- Dispositivos NVMe



## ATENÇÃO

Na versão atual, o Stratis não lida com falhas em discos rígidos ou outro hardware. Se você criar um pool Stratis sobre múltiplos dispositivos de hardware, você aumenta o risco de perda de dados porque múltiplos dispositivos devem estar operacionais para acessar os dados.

### Dispositivos sem suporte

Como o Stratis contém uma camada de provisão fina, a Red Hat não recomenda a colocação de um pool Stratis em dispositivos de blocos que já são fornecidos de forma fina.

### Recursos adicionais

- Para iSCSI e outros dispositivos de bloco que requerem rede, consulte a página de manual **systemd.mount(5)** para obter informações sobre a opção de montagem **\_netdev**.

## 17.1.4. Instalando o Stratis

Este procedimento instala todos os pacotes necessários para o uso do Stratis.

### Procedimento

1. Instalar pacotes que fornecem os serviços e utilitários de linha de comando do Stratis:

```
# yum instalar stratisd stratis-cli
```

2. Certifique-se de que o serviço **stratisd** esteja habilitado:

```
# systemctl enable --now stratisd
```

## 17.1.5. Criação de um pool Stratis

Este procedimento descreve como criar um pool de Stratis criptografados ou não criptografados a partir de um ou mais dispositivos de bloco.

As seguintes notas se aplicam aos pools de Stratis criptografados:

- Cada dispositivo de bloco é codificado usando a biblioteca **cryptsetup** e implementa o formato **LUKS2**.
- Cada piscina Stratis pode ter uma chave única ou pode compartilhar a mesma chave com outras piscinas. Estas chaves são armazenadas no chaveiro do kernel.
- Todos os dispositivos de bloco que compõem um pool Stratis são criptografados ou não criptografados. Não é possível ter tanto dispositivos de bloco criptografados quanto não criptografados no mesmo pool do Stratis.
- Os dispositivos de bloqueio adicionados ao nível de dados de um pool de Stratis criptografados são automaticamente criptografados.

## Pré-requisitos

- Stratis v2.2.1 está instalado em seu sistema. Veja [Seção 17.1.4, “Instalando o Stratis”](#).
- O serviço **stratisd** está funcionando.
- Os dispositivos de blocos nos quais você está criando um pool Stratis não estão em uso e não estão montados.
- Os dispositivos de blocos sobre os quais você está criando um pool Stratis têm pelo menos 1 GiB de tamanho cada um.
- Na arquitetura IBM Z, os dispositivos de bloco **/dev/dasd\*** devem ser particionados. Use a partição no pool do Stratis. Para informações sobre como particionar dispositivos DASD, veja [Configurando uma instância Linux na IBM Z](#).

## Procedimento

1. Se o dispositivo de bloco selecionado contiver sistema de arquivos, tabela de partição ou assinaturas RAID, apague-os usando o seguinte comando:

```
# limpa-tudo block-device
```

onde ***block-device*** é o caminho para o dispositivo do bloco; por exemplo, ***/dev/sdb***.

2. Criar o novo pool Stratis no(s) dispositivo(s) bloco(s) selecionado(s):



### NOTA

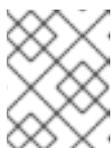
Especifique dispositivos de blocos múltiplos em uma única linha, separados por um espaço:

```
# criar um pool stratis my-pool block-device-1 block-device-2
```

- Para criar um pool Stratis não criptografado, use o seguinte comando e vá para o passo 3:

```
# criar um pool stratis my-pool block-device
```

onde ***block-device*** é o caminho para um dispositivo de blocos vazios ou limpos.



### NOTA

Você não pode criptografar um pool Stratis não criptografado depois de criá-lo.

- Para criar um pool criptografado de Stratis, complete os seguintes passos:
  - a. Se você ainda não tiver criado um conjunto de chaves, execute o seguinte comando e siga as instruções para criar um conjunto de chaves a ser usado para a criptografia:

```
# conjunto chave stratis --capture-key key-description
```

onde ***key-description*** é a descrição ou o nome do conjunto de chaves.

- b. Criar o pool de Stratis criptografados e especificar a descrição da chave a ser usada para a criptografia. Você também pode especificar o caminho da chave usando o parâmetro **--keyfile-path**.

```
# stratis pool criar --key-desc key-description my-pool block-device
```

onde

#### **key-description**

Especifica a descrição ou nome do arquivo chave a ser usado para a criptografia.

#### **my-pool**

Especifica o nome da nova piscina do Stratis.

#### **block-device**

Especifica o caminho para um dispositivo de blocos vazios ou limpos.

3. Verificar se o novo pool Stratis foi criado:

```
# lista stratis pool
```

## Solução de problemas

Após um reinício do sistema, às vezes você pode não ver seu pool de Stratis criptografado ou os dispositivos de bloco que o compõem. Se você encontrar este problema, você deve desbloquear o pool do Stratis para torná-lo visível.

Para desbloquear a piscina do Stratis, complete os seguintes passos:

1. Recriar o conjunto de chaves usando a mesma descrição chave que foi usada anteriormente:

```
# conjunto chave stratis --capture-key key-description
```

2. Desbloquear a piscina do Stratis e o(s) dispositivo(s) de bloqueio:

```
# desbloqueio do pool stratis
```

3. Verificar se o pool do Stratis está visível:

```
# lista stratis pool
```

## Recursos adicionais

- A página do homem **stratis(8)**.

## Próximos passos

- Criar um sistema de arquivo Stratis no pool. Para maiores informações, ver [Seção 17.1.6, "Criação de um sistema de arquivo Stratis"](#).

### 17.1.6. Criação de um sistema de arquivo Stratis

Este procedimento cria um sistema de arquivos Stratis em um pool Stratis existente.

### Pré-requisitos

- O Stratis está instalado. Ver [Seção 17.1.4, “Instalando o Stratis”](#).
- O serviço **stratisd** está funcionando.
- Você criou uma piscina Stratis. Veja [Seção 17.1.5, “Criação de um pool Stratis”](#).

### Procedimento

1. Para criar um sistema de arquivos Stratis em um pool, use:

```
# stratis fs criam my-pool my-fs
```

- Substitua *my-pool* com o nome da sua piscina Stratis existente.
- Substitua *my-fs* com um nome arbitrário para o sistema de arquivo.

2. Para verificar, liste os sistemas de arquivos dentro do pool:

```
# lista stratis fs my-pool
```

### Recursos adicionais

- A página do homem **stratis(8)**

### Próximos passos

- Montar o sistema de arquivos Stratis. Ver [Seção 17.1.7, “Montagem de um sistema de arquivo Stratis”](#).

## 17.1.7. Montagem de um sistema de arquivo Stratis

Este procedimento monta um sistema de arquivo Stratis existente para acessar o conteúdo.

### Pré-requisitos

- O Stratis está instalado. Ver [Seção 17.1.4, “Instalando o Stratis”](#).
- O serviço **stratisd** está funcionando.
- Você criou um sistema de arquivo Stratis. Veja [Seção 17.1.6, “Criação de um sistema de arquivo Stratis”](#).

### Procedimento

- Para montar o sistema de arquivo, use as entradas que Stratis mantém no diretório **/stratis/**:

```
# montagem /stratis/my-pool/my-fs mount-point
```

O sistema de arquivo está agora montado no *mount-point* e pronto para uso.

### Recursos adicionais

- A página do homem **mount(8)**

### 17.1.8. Montagem persistente de um sistema de arquivo Stratis

Este procedimento monta persistentemente um sistema de arquivo Stratis para que ele esteja disponível automaticamente após a inicialização do sistema.

#### Pré-requisitos

- O Stratis está instalado. Ver [Seção 17.1.4, “Instalando o Stratis”](#).
- O serviço **stratisd** está funcionando.
- Você criou um sistema de arquivo Stratis. Veja [Seção 17.1.6, “Criação de um sistema de arquivo Stratis”](#).

#### Procedimento

1. Determinar o atributo UUID do sistema de arquivo:

```
$ lsblk --output=UUID /stratis/my-pool/my-fs
```

Por exemplo:

#### Exemplo 17.1. Visualizando a UUID do sistema de arquivos Stratis

```
$ lsblk --output=UUID /stratis/my-pool/fs1
UUID
a1f0b64a-4ebb-4d4e-9543-b1d79f600283
```

2. Se o diretório de pontos de montagem não existir, crie-o:

```
# mkdir -p mount-point
```

3. Como root, editar o arquivo **/etc/fstab** e adicionar uma linha para o sistema de arquivo, identificado pela UUID. Use **xf**s como o tipo de sistema de arquivo e adicione a opção **x-systemd.requires=stratisd.service**.

Por exemplo:

#### Exemplo 17.2. O ponto de montagem /fs1 em /etc/fstab

```
UUID=a1f0b64a-4ebb-4d4e-9543-b1d79f600283 /fs1 xfs default,x-
systemd.requires=stratisd.service 0 0
```

4. Regenere unidades de montagem para que seu sistema registre a nova configuração:

```
# systemctl daemon-reload
```

5. Tente montar o sistema de arquivo para verificar se a configuração funciona:

# montar *mount-point*

## Recursos adicionais

- [Seção 14.8, “Montagem persistente de sistemas de arquivo”](#)

### 17.1.9. Informações relacionadas

- O site *Stratis Storage*: <https://stratis-storage.github.io/>

## 17.2. AMPLIAÇÃO DE UM VOLUME STRATIS COM DISPOSITIVOS DE BLOQUEIO ADICIONAIS

Você pode anexar dispositivos de blocos adicionais a um pool Stratis para fornecer mais capacidade de armazenamento para os sistemas de arquivos Stratis.

### 17.2.1. Componentes de um volume Stratis

Externamente, Stratis apresenta os seguintes componentes de volume na interface de linha de comando e no API:

#### **blockdev**

Dispositivos de bloqueio, tais como uma partição de disco ou uma partição de disco.

#### **pool**

Composto por um ou mais dispositivos de bloco.

Uma piscina tem um tamanho total fixo, igual ao tamanho dos dispositivos do bloco.

O pool contém a maioria das camadas do Stratis, como o cache de dados não volátil, usando o alvo **dm-cache**.

Stratis cria um **/stratis/my-pool/** para cada pool. Este diretório contém links para dispositivos que representam os sistemas de arquivos Stratis no pool.

#### **filesystem**

Cada pool pode conter um ou mais sistemas de arquivos, que armazenam arquivos.

Os sistemas de arquivo são pouco provisionados e não têm um tamanho total fixo. O tamanho real de um sistema de arquivo cresce com os dados nele armazenados. Se o tamanho dos dados se aproximar do tamanho virtual do sistema de arquivo, o Stratis aumenta o volume fino e o sistema de arquivo automaticamente.

Os sistemas de arquivo são formatados com XFS.



#### **IMPORTANTE**

Stratis rastreia informações sobre sistemas de arquivo criados usando Stratis que o XFS não conhece, e as mudanças feitas usando o XFS não criam automaticamente atualizações no Stratis. Os usuários não devem reformatar ou reconfigurar sistemas de arquivos XFS que são gerenciados pelo Stratis.

Stratis cria links para sistemas de arquivo no **/stratis/my-pool/my-fs** caminho.



## NOTA

Stratis usa muitos dispositivos Device Mapper, que aparecem na lista **dmsetup** e no arquivo **/proc/partitions**. Da mesma forma, a saída do comando **lsblk** reflete o funcionamento interno e as camadas do Stratis.

### 17.2.2. Adicionando dispositivos de blocos a uma piscina Stratis

Este procedimento acrescenta um ou mais dispositivos de bloco a um pool Stratis para ser utilizado pelos sistemas de arquivo Stratis.

#### Pré-requisitos

- O Stratis está instalado. Ver [Seção 17.1.4, "Instalando o Stratis"](#).
- O serviço **stratisd** está funcionando.
- Os dispositivos de bloco que você está adicionando ao pool Stratis não estão em uso e não estão montados.
- Os dispositivos de bloco que você está adicionando ao pool do Stratis têm pelo menos 1 GiB de tamanho cada um.

#### Procedimento

- Para adicionar um ou mais dispositivos de bloqueio ao pool, use:

```
# dados adicionais do pool stratis my-pool device-1 device-2 device-n
```

#### Recursos adicionais

- A página do homem **stratis(8)**

### 17.2.3. Informações relacionadas

- O site *Stratis Storage*: <https://stratis-storage.github.io/>

## 17.3. MONITORAMENTO DOS SISTEMAS DE ARQUIVO STRATIS

Como um usuário Stratis, você pode visualizar informações sobre volumes Stratis em seu sistema para monitorar seu estado e espaço livre.

### 17.3.1. Tamanhos Stratis relatados por diferentes utilidades

Esta seção explica a diferença entre os tamanhos do Stratis relatados por utilitários padrão, tais como **df** e o utilitário **stratis**.

Utilitários Linux padrão como o **df** informam o tamanho da camada do sistema de arquivos XFS no Stratis, que é 1 TiB. Esta informação não é útil, porque o uso real de armazenamento do Stratis é menor devido ao provisionamento fino, e também porque o Stratis aumenta automaticamente o sistema de arquivo quando a camada XFS está perto de estar cheia.



## IMPORTANTE

Monitore regularmente a quantidade de dados escritos em seus sistemas de arquivos Stratis, que é relatada como o valor *Total Physical Used*. Certifique-se de que não exceda o valor *Total Physical Size*.

### Recursos adicionais

- A página do homem **stratis(8)**

### 17.3.2. Exibindo informações sobre os volumes do Stratis

Este procedimento lista estatísticas sobre seus volumes Stratis, tais como o total, usado e tamanho livre ou sistemas de arquivos e dispositivos de blocos pertencentes a um pool.

#### Pré-requisitos

- O Stratis está instalado. Ver [Seção 17.1.4, "Instalando o Stratis"](#).
- O serviço **stratisd** está funcionando.

#### Procedimento

- Para exibir informações sobre todos os **block devices** utilizados para o Stratis em seu sistema:

```
# stratis blockdev

Pool Name Device Node Physical Size State Tier
my-pool /dev/sdb 9.10 TiB In-use Data
```

- Para exibir informações sobre todos os Stratis **pools** em seu sistema:

```
# stratis pool

Name Total Physical Size Total Physical Used
my-pool 9.10 TiB 598 MiB
```

- Para exibir informações sobre todos os Stratis **file systems** em seu sistema:

```
# stratis filesystem

Pool Name Name Used Created Device
my-pool my-fs 546 MiB Nov 08 2018 08:03 /stratis/my-pool/my-fs
```

### Recursos adicionais

- A página do homem **stratis(8)**

### 17.3.3. Informações relacionadas

- O site *Stratis Storage*: <https://stratis-storage.github.io/>

## 17.4. USANDO SNAPSHOTS EM SISTEMAS DE ARQUIVO STRATIS

Você pode usar instantâneos nos sistemas de arquivo Stratis para capturar o estado do sistema de arquivo em momentos arbitrários e restaurá-lo no futuro.

### 17.4.1. Características dos snapshots de Stratis

Esta seção descreve as propriedades e limitações dos instantâneos do sistema de arquivo no Stratis.

No Stratis, um instantâneo é um sistema de arquivo Stratis regular criado como uma cópia de outro sistema de arquivo Stratis. O snapshot inicialmente contém o mesmo conteúdo de arquivo do sistema de arquivo original, mas pode mudar à medida que o snapshot é modificado. Quaisquer mudanças que você fizer no instantâneo não serão refletidas no sistema de arquivo original.

A implementação atual no Stratis é caracterizada pelo seguinte:

- Um instantâneo de um sistema de arquivo é outro sistema de arquivo.
- Um instantâneo e sua origem não estão ligados durante toda a vida. Um sistema de arquivo com instantâneo pode viver mais tempo do que o sistema de arquivo a partir do qual foi criado.
- Um sistema de arquivo não precisa ser montado para criar um instantâneo a partir dele.
- Cada snapshot usa cerca de meio gigabyte de armazenamento real, que é necessário para o log XFS.

### 17.4.2. Criando um instantâneo do Stratis

Este procedimento cria um sistema de arquivo Stratis como um instantâneo de um sistema de arquivo Stratis existente.

#### Pré-requisitos

- O Stratis está instalado. Ver [Seção 17.1.4, "Instalando o Stratis"](#).
- O serviço **stratisd** está funcionando.
- Você criou um sistema de arquivo Stratis. Veja [Seção 17.1.6, "Criação de um sistema de arquivo Stratis"](#).

#### Procedimento

- Para criar um instantâneo do Stratis, use:

```
# foto de stratis fs my-pool my-fs my-fs-snapshot
```

#### Recursos adicionais

- A página do homem **stratis(8)**

### 17.4.3. Acesso ao conteúdo de uma foto de Stratis

Este procedimento monta um instantâneo de um sistema de arquivo Stratis para torná-lo acessível para operações de leitura e escrita.

#### Pré-requisitos

- O Stratis está instalado. Ver [Seção 17.1.4, “Instalando o Stratis”](#).
- O serviço **stratisd** está funcionando.
- Você criou um instantâneo do Stratis. Veja [Seção 17.4.2, “Criando um instantâneo do Stratis”](#).

### Procedimento

- Para acessar o instantâneo, monte-o como um sistema de arquivo regular a partir do **/stratis/my-pool/** diretório:

```
# montagem /stratis/my-pool/my-fs-snapshot mount-point
```

### Recursos adicionais

- [Seção 17.1.7, “Montagem de um sistema de arquivo Stratis”](#)
- A página do homem **mount(8)**

## 17.4.4. Revertendo um sistema de arquivo Stratis para um instantâneo anterior

Este procedimento reverte o conteúdo de um sistema de arquivo Stratis para o estado capturado em um instantâneo Stratis.

### Pré-requisitos

- O Stratis está instalado. Ver [Seção 17.1.4, “Instalando o Stratis”](#).
- O serviço **stratisd** está funcionando.
- Você criou um instantâneo do Stratis. Veja [Seção 17.4.2, “Criando um instantâneo do Stratis”](#).

### Procedimento

1. Opcionalmente, faça um backup do estado atual do sistema de arquivos para poder acessá-lo mais tarde:

```
# foto do sistema de arquivos stratis my-pool my-fs my-fs-backup
```

2. Desmontar e remover o sistema de arquivo original:

```
# umount /stratis/my-pool/my-fs
# stratis filesystem destroy my-pool my-fs
```

3. Criar uma cópia do instantâneo sob o nome do sistema de arquivo original:

```
# foto do sistema de arquivos stratis my-pool my-fs-snapshot my-fs
```

4. Monte o instantâneo, que agora está acessível com o mesmo nome do sistema de arquivo original:

```
# montagem /stratis/my-pool/my-fs mount-point
```

O conteúdo do sistema de arquivo chamado *my-fs* é agora idêntico ao snapshot *my-fs-snapshot*.

### Recursos adicionais

- A página do homem **stratis(8)**

## 17.4.5. Removendo uma foto do Stratis

Este procedimento remove um instantâneo de Stratis de uma piscina. Os dados do instantâneo são perdidos.

### Pré-requisitos

- O Stratis está instalado. Ver [Seção 17.1.4, “Instalando o Stratis”](#).
- O serviço **stratisd** está funcionando.
- Você criou um instantâneo do Stratis. Veja [Seção 17.4.2, “Criando um instantâneo do Stratis”](#).

### Procedimento

1. Desmontar o instantâneo:

```
# umount /stratis/my-pool/my-fs-snapshot
```

2. Destrua o instantâneo:

```
# sistema de arquivos stratis destruir my-pool my-fs-snapshot
```

### Recursos adicionais

- A página do homem **stratis(8)**

## 17.4.6. Informações relacionadas

- O site *Stratis Storage*: <https://stratis-storage.github.io/>

# 17.5. REMOÇÃO DOS SISTEMAS DE ARQUIVO STRATIS

Você pode remover um sistema de arquivos Stratis existente ou um pool Stratis, destruindo os dados sobre eles.

## 17.5.1. Componentes de um volume Stratis

Externamente, Stratis apresenta os seguintes componentes de volume na interface de linha de comando e no API:

### **blockdev**

Dispositivos de bloqueio, tais como uma partição de disco ou uma partição de disco.

### **pool**

Composto por um ou mais dispositivos de bloco.

Uma piscina tem um tamanho total fixo, igual ao tamanho dos dispositivos do bloco.

O pool contém a maioria das camadas do Stratis, como o cache de dados não volátil, usando o alvo **dm-cache**.

Stratis cria um **/stratis/my-pool/** para cada pool. Este diretório contém links para dispositivos que representam os sistemas de arquivos Stratis no pool.

## filesystem

Cada pool pode conter um ou mais sistemas de arquivos, que armazenam arquivos.

Os sistemas de arquivo são pouco provisionados e não têm um tamanho total fixo. O tamanho real de um sistema de arquivo cresce com os dados nele armazenados. Se o tamanho dos dados se aproximar do tamanho virtual do sistema de arquivo, o Stratis aumenta o volume fino e o sistema de arquivo automaticamente.

Os sistemas de arquivo são formatados com XFS.



### IMPORTANTE

Stratis rastreia informações sobre sistemas de arquivo criados usando Stratis que o XFS não conhece, e as mudanças feitas usando o XFS não criam automaticamente atualizações no Stratis. Os usuários não devem reformatar ou reconfigurar sistemas de arquivos XFS que são gerenciados pelo Stratis.

Stratis cria links para sistemas de arquivo no **/stratis/my-pool/my-fs** caminho.



### NOTA

Stratis usa muitos dispositivos Device Mapper, que aparecem na lista **dmsetup** e no arquivo **/proc/partitions**. Da mesma forma, a saída do comando **lsblk** reflete o funcionamento interno e as camadas do Stratis.

## 17.5.2. Remoção de um sistema de arquivo Stratis

Este procedimento remove um sistema de arquivos Stratis existente. Os dados armazenados nele são perdidos.

### Pré-requisitos

- O Stratis está instalado. Ver [Seção 17.1.4, "Instalando o Stratis"](#).
- O serviço **stratisd** está funcionando.
- Você criou um sistema de arquivo Stratis. Veja [Seção 17.1.6, "Criação de um sistema de arquivo Stratis"](#).

### Procedimento

1. Desmontar o sistema de arquivo:

```
# umount /stratis/my-pool/my-fs
```

2. Destruir o sistema de arquivos:

```
# sistema de arquivos stratis destruir my-pool my-fs
```

3. Verificar se o sistema de arquivo não existe mais:

```
# lista do sistema de arquivos stratis my-pool
```

### Recursos adicionais

- A página do homem **stratis(8)**

### 17.5.3. Remoção de uma piscina Stratis

Este procedimento remove uma piscina existente do Stratis. Os dados armazenados nele são perdidos.

#### Pré-requisitos

- O Stratis está instalado. Ver [Seção 17.1.4, “Instalando o Stratis”](#).
- O serviço **stratisd** está funcionando.
- Você criou uma piscina Stratis. Veja [Seção 17.1.5, “Criação de um pool Stratis”](#).

#### Procedimento

1. Listar os sistemas de arquivos no pool:

```
# lista do sistema de arquivos stratis my-pool
```

2. Desmontar todos os sistemas de arquivos no pool:

```
# umount /stratis/my-pool/my-fs-1 \  
/stratis/my-pool/my-fs-2 \  
/stratis/my-pool/my-fs-n
```

3. Destruir os sistemas de arquivo:

```
# sistema de arquivos stratis destruir my-pool my-fs-1 my-fs-2
```

4. Destruir a piscina:

```
# stratis pool destruir my-pool
```

5. Verificar se o pool não existe mais:

```
# lista stratis pool
```

### Recursos adicionais

- A página do homem **stratis(8)**

### 17.5.4. Informações relacionadas

- O site *Stratis Storage*: <https://stratis-storage.github.io/>

# CAPÍTULO 18. COMEÇANDO COM UM SISTEMA DE ARQUIVO EXT3

Como administrador de sistema, você pode criar, montar, redimensionar, fazer backup e restaurar um sistema de arquivos ext3. O sistema de arquivos ext3 é essencialmente uma versão aprimorada do sistema de arquivos ext2.

## 18.1. CARACTERÍSTICAS DE UM SISTEMA DE ARQUIVO EXT3

A seguir estão as características de um sistema de arquivo ext3:

- Disponibilidade: Após uma falha inesperada de energia ou falha do sistema, a verificação do sistema de arquivo não é necessária devido ao diário fornecido. O tamanho padrão do diário leva cerca de um segundo para ser recuperado, dependendo da velocidade do hardware



### NOTA

O único modo de jornalismo suportado no ext3 é **data=ordered** (padrão). Para mais informações, consulte [a opção EXT journaling "data=writeback" suportada na RHEL?](#) Artigo da Base de Conhecimento.

- Integridade dos dados: O sistema de arquivo ext3 evita a perda da integridade dos dados durante uma falha inesperada de energia ou uma falha do sistema.
- Velocidade: Apesar de escrever alguns dados mais de uma vez, o ext3 tem um rendimento maior na maioria dos casos do que o ext2 porque o diário do ext3 otimiza o movimento da cabeça do disco rígido.
- Transição fácil: É fácil migrar de ext2 para ext3 e obter os benefícios de um sistema robusto de arquivo de periódicos sem reformatar.

### Recursos adicionais

- A página do homem **ext3**.

## 18.2. CRIAÇÃO DE UM SISTEMA DE ARQUIVO EXT3

Como administrador de sistema, você pode criar um sistema de arquivo ext3 em um dispositivo de bloco usando o comando **mkfs.ext3**.

### Pré-requisitos

- Uma partição em seu disco. Para informações sobre a criação de partições MBR ou GPT, veja [Seção 10.2, "Criação de uma tabela de partição em um disco"](#). Alternativamente, use um volume LVM ou MD.

### Procedimento

1. Para criar um sistema de arquivo ext3:
  - Para um dispositivo de partição regular, um volume LVM, um volume MD, ou um dispositivo similar, use o seguinte comando:

```
# mkfs.ext3 /dev/block_device
```

Substituir `/dev/block_device` pelo caminho para um dispositivo de bloco.

Por exemplo, `/dev/sdb1`, `/dev/disk/by-uuid/05e99ec8-def1-4a5e-8a9d-5945339ceb2a`, ou `/dev/my-volgroup/my-lv`. Em geral, as opções padrão são ótimas para a maioria dos cenários de uso.



#### NOTA

- Para especificar uma UUID ao criar um sistema de arquivo:

```
# mkfs.ext3 -U UUID /dev/block_device
```

Substitua `UUID` pela UUID que você deseja definir: por exemplo, **7cd65de3-e0be-41d9-b66d-96d749c02da7**.

Substitua `/dev/block_device` pelo caminho para um sistema de arquivo ext3 para ter o UUID adicionado a ele: por exemplo, `/dev/sda8`.

- Para especificar uma etiqueta ao criar um sistema de arquivo:

```
# mkfs.ext3 -L label-name /dev/block_device
```

2. Para visualizar o sistema de arquivo ext3 criado:

```
# blkid
```

#### Recursos adicionais

- A página do homem **ext3**.
- A página do homem **mkfs.ext3**.

## 18.3. MONTAGEM DE UM SISTEMA DE ARQUIVO EXT3

Como administrador do sistema, você pode montar um sistema de arquivos ext3 usando o utilitário **mount**.

#### Pré-requisitos

- Um sistema de arquivo ext3. Para informações sobre como criar um sistema de arquivo ext3, veja [Seção 18.2, "Criação de um sistema de arquivo ext3"](#).

#### Procedimento

1. Para criar um ponto de montagem para montar o sistema de arquivo:

```
# mkdir /mount/point
```

Substituir `/mount/point` pelo nome do diretório onde o ponto de montagem da partição deve ser criado.

2. Para montar um sistema de arquivo ext3:

- Para montar um sistema de arquivo ext3 sem opções extras:

```
# montar /dev/block_device /mount/point
```

- Para montar o sistema de arquivo de forma persistente, veja [Seção 14.8, "Montagem persistente de sistemas de arquivo"](#).

3. Para visualizar o sistema de arquivo montado:

```
# df -h
```

### Recursos adicionais

- A página do homem **mount**.
- A página do homem **ext3**.
- A página do homem **fstab**.
- [Capítulo 14, Montagem de sistemas de arquivo](#)

## 18.4. REDIMENSIONAMENTO DE UM SISTEMA DE ARQUIVO EXT3

Como administrador de sistema, você pode redimensionar um sistema de arquivos ext3 usando o utilitário **resize2fs**. O utilitário **resize2fs** lê o tamanho em unidades de tamanho de bloco do sistema de arquivos, a menos que um sufixo indicando uma unidade específica seja usado. Os sufixos a seguir indicam unidades específicas:

- s (setores) - **512** byte sectors
- K (kilobytes) - **1,024** bytes
- M (megabytes) - **1,048,576** bytes
- G (gigabytes) - **1,073,741,824** bytes
- T (terabytes) - **1,099,511,627,776** bytes

### Pré-requisitos

- Um sistema de arquivo ext3. Para informações sobre como criar um sistema de arquivo ext3, veja [Seção 18.2, "Criação de um sistema de arquivo ext3"](#).
- Um dispositivo de bloco subjacente de tamanho apropriado para segurar o sistema de arquivo após o redimensionamento.

### Procedimento

1. Para redimensionar um sistema de arquivo ext3, tome as seguintes medidas:

- Reduzir e aumentar o tamanho de um sistema de arquivo ext3 não montado:

```
# umount /dev/block_device
# e2fsck -f /dev/block_device
# resize2fs /dev/block_device size
```

Substituir `/dev/block_device` pelo caminho para o dispositivo do bloco, por exemplo `/dev/sdb1`.

Substituir `size` pelo valor de redimensionamento necessário usando **s**, **K**, **M**, **G**, e **T** sufixos.

- Um sistema de arquivo ext3 pode ser desenvolvido enquanto montado usando o comando **resize2fs**:

```
# redimensionar2fs /mount/device size
```



#### NOTA

O parâmetro de tamanho é opcional (e muitas vezes redundante) quando se expande. O **resize2fs** se expande automaticamente para preencher o espaço disponível do container, geralmente um volume lógico ou partição.

2. Para visualizar o sistema de arquivo redimensionado:

```
# df -h
```

#### Recursos adicionais

- A página do homem **resize2fs**.
- A página do homem **e2fsck**.
- A página do homem **ext3**.

## 18.5. CRIAÇÃO E MONTAGEM DE SISTEMAS DE ARQUIVO EXT3 USANDO FUNÇÕES DO SISTEMA RHEL

Esta seção descreve como criar um sistema de arquivo ext3 com uma determinada etiqueta em um disco, e montar persistentemente o sistema de arquivo usando a função **storage**.

#### Pré-requisitos

- Existe um livro de brincadeiras possível, incluindo o papel **storage**.

Para informações sobre como aplicar tal playbook, consulte [Aplicando um papel](#).

### 18.5.1. 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 18.1. 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**.

#### 18.5.2. Recursos adicionais

- Para mais informações sobre a função **storage**, ver [Seção 2.1, "Introdução à função de armazenamento"](#).

# CAPÍTULO 19. COMEÇANDO COM UM SISTEMA DE ARQUIVO EXT4

Como administrador de sistema, você pode criar, montar, redimensionar, fazer backup e restaurar um sistema de arquivos ext4. O sistema de arquivo ext4 é uma extensão escalável do sistema de arquivo ext3. Com o Red Hat Enterprise Linux 8, ele pode suportar um tamanho máximo de arquivo individual de **16** terabytes, e sistema de arquivo até um máximo de **50** terabytes.

## 19.1. CARACTERÍSTICAS DE UM SISTEMA DE ARQUIVO EXT4

A seguir estão as características de um sistema de arquivo ext4:

- Usando extensões: O sistema de arquivos ext4 usa extensões, o que melhora o desempenho quando se usam arquivos grandes e reduz a sobrecarga de metadados para arquivos grandes.
- Ext4 etiquetas de blocos não alocados e seções de tabela inode de acordo, o que permite que os grupos de blocos e seções de tabela sejam pulados durante uma verificação do sistema de arquivo. Isto leva a uma rápida verificação do sistema de arquivo, que se torna mais benéfica à medida que o sistema de arquivo cresce em tamanho.
- Metadata checksum: Por default, este recurso é habilitado no Red Hat Enterprise Linux 8.
- Características de alocação de um sistema de arquivo ext4:
  - Pré-alocação persistente
  - Atraso na alocação
  - Alocação multiblocos
  - Alocação de listras
- Atributos estendidos (**xattr**): Isto permite que o sistema associe vários pares de nomes e valores adicionais por arquivo.
- Diário de cotas: Isto evita a necessidade de longas verificações de consistência de cotas após uma queda.



### NOTA

O único modo de jornalismo suportado no ext4 é **data=ordered** (padrão). Para mais informações, consulte [a opção EXT journaling "data=writeback" suportada na RHEL?](#) Artigo da Base de Conhecimento.

- Carimbos temporais do sub-segundo - Isto dá carimbos temporais para o sub-segundo.

### Recursos adicionais

- A página do homem **ext4**.

## 19.2. CRIAÇÃO DE UM SISTEMA DE ARQUIVO EXT4

Como administrador de sistema, você pode criar um sistema de arquivo ext4 em um dispositivo de bloco usando o comando **mkfs.ext4**.

## Pré-requisitos

- Uma partição em seu disco. Para informações sobre a criação de partições MBR ou GPT, veja [Seção 10.2, “Criação de uma tabela de partição em um disco”](#) . Alternativamente, use um volume LVM ou MD.

## Procedimento

1. Para criar um sistema de arquivo ext4:

- Para um dispositivo de partição regular, um volume LVM, um volume MD, ou um dispositivo similar, use o seguinte comando:

```
# mkfs.ext4 /dev/block_device
```

Substituir `/dev/block_device` pelo caminho para um dispositivo de bloco.

Por exemplo, `/dev/sdb1`, `/dev/disk/by-uuid/05e99ec8-def1-4a5e-8a9d-5945339ceb2a`, ou `/dev/my-volgroup/my-lv`. Em geral, as opções padrão são ótimas para a maioria dos cenários de uso.

- Para dispositivos de blocos listrados (por exemplo, matrizes RAID5), a geometria da banda pode ser especificada no momento da criação do sistema de arquivo. O uso de geometria de faixas adequada aumenta o desempenho de um sistema de arquivo ext4. Por exemplo, para criar um sistema de arquivo com um passo de 64k (ou seja, 16 x 4096) em um sistema de arquivo de 4k-block, use o seguinte comando:

```
# mkfs.ext4 -E stride=16,stripe-width=64 /dev/block_device
```

No exemplo dado:

- `stride=value`: Especifica o tamanho do pedaço RAID
- `largura de faixa=valor`: Especifica o número de discos de dados em um dispositivo RAID, ou o número de unidades de listras na listra.



### NOTA

- Para especificar uma UUID ao criar um sistema de arquivo:

```
# mkfs.ext4 -U UUID /dev/block_device
```

Substitua `UUID` pela UUID que você deseja definir: por exemplo, **7cd65de3-e0be-41d9-b66d-96d749c02da7**.

Substitua `/dev/block_device` pelo caminho para um sistema de arquivo ext4 para ter o UUID adicionado a ele: por exemplo, `/dev/sda8`.

- Para especificar uma etiqueta ao criar um sistema de arquivo:

```
# mkfs.ext4 -L label-name /dev/block_device
```

2. Para visualizar o sistema de arquivo ext4 criado:

```
# blkid
```

### Recursos adicionais

- A página do homem **ext4**.
- A página do homem **mkfs.ext4**.

## 19.3. MONTAGEM DE UM SISTEMA DE ARQUIVO EXT4

Como administrador do sistema, você pode montar um sistema de arquivos ext4 usando o utilitário **mount**.

### Pré-requisitos

- Um sistema de arquivo ext4. Para informações sobre como criar um sistema de arquivo ext4, veja [Seção 19.2, "Criação de um sistema de arquivo ext4"](#).

### Procedimento

1. Para criar um ponto de montagem para montar o sistema de arquivo:

```
# mkdir /mount/point
```

Substituir */mount/point* pelo nome do diretório onde o ponto de montagem da partição deve ser criado.

2. Para montar um sistema de arquivo ext4:

- Para montar um sistema de arquivo ext4 sem opções extras:

```
# montar /dev/block_device /mount/point
```

- Para montar o sistema de arquivo de forma persistente, veja [Seção 14.8, "Montagem persistente de sistemas de arquivo"](#).

3. Para visualizar o sistema de arquivo montado:

```
# df -h
```

### Recursos adicionais

- A página do homem **mount**.
- A página do homem **ext4**.
- A página do homem **fstab**.
- [Capítulo 14, Montagem de sistemas de arquivo](#)

## 19.4. REDIMENSIONAMENTO DE UM SISTEMA DE ARQUIVO EXT4

Como administrador de sistema, você pode redimensionar um sistema de arquivos ext4 usando o

utilitário **resize2fs**. O utilitário **resize2fs** lê o tamanho em unidades de tamanho de bloco do sistema de arquivos, a menos que um sufixo indicando uma unidade específica seja usado. Os sufixos a seguir indicam unidades específicas:

- s (setores) - **512** byte sectors
- K (kilobytes) - **1,024** bytes
- M (megabytes) - **1,048,576** bytes
- G (gigabytes) - **1,073,741,824** bytes
- T (terabytes) - **1,099,511,627,776** bytes

### Pré-requisitos

- Um sistema de arquivo ext4. Para informações sobre como criar um sistema de arquivo ext4, veja [Seção 19.2, "Criação de um sistema de arquivo ext4"](#).
- Um dispositivo de bloco subjacente de tamanho apropriado para segurar o sistema de arquivo após o redimensionamento.

### Procedimento

1. Para redimensionar um sistema de arquivo ext4, tome as seguintes medidas:

- Reduzir e aumentar o tamanho de um sistema de arquivo ext4 não montado:

```
# umount /dev/block_device  
# e2fsck -f /dev/block_device  
# resize2fs /dev/block_device size
```

Substituir `/dev/block_device` pelo caminho para o dispositivo do bloco, por exemplo `/dev/sdb1`.

Substituir `size` pelo valor de redimensionamento necessário usando **s**, **K**, **M**, **G**, e **T** sufixos.

- Um sistema de arquivo ext4 pode ser desenvolvido enquanto montado usando o comando **resize2fs**:

```
# redimensionar2fs /mount/device size
```



#### NOTA

O parâmetro de tamanho é opcional (e muitas vezes redundante) quando se expande. O **resize2fs** se expande automaticamente para preencher o espaço disponível do container, geralmente um volume lógico ou partição.

2. Para visualizar o sistema de arquivo redimensionado:

```
# df -h
```

### Recursos adicionais

- A página do homem **resize2fs**.
- A página do homem **e2fsck**.
- A página do homem **ext4**.

## 19.5. CRIAÇÃO E MONTAGEM DE SISTEMAS DE ARQUIVO EXT4 USANDO FUNÇÕES DO SISTEMA RHEL

Esta seção descreve como criar um sistema de arquivo ext4 com uma determinada etiqueta em um disco, e montar persistentemente o sistema de arquivo usando a função **storage**.

### Pré-requisitos

- Existe um livro de brincadeiras possível, incluindo o papel **storage**.

Para informações sobre como aplicar tal playbook, consulte [Aplicando um papel](#).

### 19.5.1. 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 19.1. 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**.

### Recursos adicionais

- Para mais informações sobre a função **storage**, ver [Seção 2.1, “Introdução à função de armazenamento”](#).

## 19.6. COMPARAÇÃO DAS FERRAMENTAS UTILIZADAS COM EXT4 E XFS

Esta seção compara quais ferramentas usar para realizar tarefas comuns nos sistemas de arquivos ext4 e XFS.

Tarefa	ext4	XFS
Criar um sistema de arquivo	<b>mkfs.ext4</b>	<b>mkfs.xfs</b>
Verificação do sistema de arquivo	<b>e2fsck</b>	<b>xfs_repair</b>
Redimensionar um sistema de arquivo	<b>resize2fs</b>	<b>xfs_growfs</b>
Salvar uma imagem de um sistema de arquivo	<b>e2image</b>	<b>xfs_metadump</b> e <b>xfs_mdrestore</b>
Etiquetar ou afinar um sistema de arquivo	<b>tune2fs</b>	<b>xfs_admin</b>
Faça o backup de um sistema de arquivo	<b>dump</b> e <b>restore</b>	<b>xfsdump</b> e <b>xfsrestore</b>
Gestão de cotas	<b>quota</b>	<b>xfs_quota</b>
Mapeamento de arquivos	<b>filefrag</b>	<b>xfs_bmap</b>