



Red Hat Enterprise Linux 9

Gestion des systèmes de fichiers

Créer, modifier et administrer des systèmes de fichiers dans Red Hat Enterprise
Linux 9

Red Hat Enterprise Linux 9 Gestion des systèmes de fichiers

Créer, modifier et administrer des systèmes de fichiers dans Red Hat Enterprise Linux 9

Notice légale

Copyright © 2023 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

Résumé

Cette collection de documentation fournit des instructions sur la manière de gérer efficacement les systèmes de fichiers dans Red Hat Enterprise Linux 9.

Table des matières

RENDRE L'OPEN SOURCE PLUS INCLUSIF	8
FOURNIR UN RETOUR D'INFORMATION SUR LA DOCUMENTATION DE RED HAT	9
CHAPITRE 1. APERÇU DES SYSTÈMES DE FICHIERS DISPONIBLES	10
1.1. TYPES DE SYSTÈMES DE FICHIERS	10
1.2. SYSTÈMES DE FICHIERS LOCAUX	11
1.3. LE SYSTÈME DE FICHIERS XFS	11
1.4. LE SYSTÈME DE FICHIERS EXT4	13
1.5. COMPARAISON ENTRE XFS ET EXT4	13
1.6. CHOIX D'UN SYSTÈME DE FICHIERS LOCAL	14
1.7. SYSTÈMES DE FICHIERS EN RÉSEAU	16
1.8. SYSTÈMES DE FICHIERS À STOCKAGE PARTAGÉ	16
1.9. CHOISIR ENTRE LES SYSTÈMES DE FICHIERS EN RÉSEAU ET LES SYSTÈMES DE FICHIERS À STOCKAGE PARTAGÉ	17
1.10. SYSTÈMES DE FICHIERS GÉRANT LES VOLUMES	17
CHAPITRE 2. GESTION DU STOCKAGE LOCAL À L'AIDE DES RÔLES SYSTÈME RHEL	19
2.1. INTRODUCTION AU RÔLE DU SYSTÈME RHEL STORAGE	19
2.2. PARAMÈTRES QUI IDENTIFIENT UN PÉRIPHÉRIQUE DE STOCKAGE DANS LE RÔLE DE SYSTÈME RHEL STORAGE	20
2.3. EXEMPLE DE SCRIPT ANSIBLE POUR CRÉER UN SYSTÈME DE FICHIERS XFS SUR UN PÉRIPHÉRIQUE BLOC	20
2.4. EXEMPLE DE PLAYBOOK ANSIBLE POUR MONTER UN SYSTÈME DE FICHIERS DE MANIÈRE PERSISTANTE	21
2.5. EXEMPLE DE SCRIPT ANSIBLE POUR LA GESTION DES VOLUMES LOGIQUES	21
2.6. EXEMPLE DE SCRIPT ANSIBLE POUR ACTIVER L'ÉLIMINATION DES BLOCS EN LIGNE	22
2.7. EXEMPLE DE SCRIPT ANSIBLE POUR CRÉER ET MONTER UN SYSTÈME DE FICHIERS EXT4	23
2.8. EXEMPLE DE SCRIPT ANSIBLE POUR CRÉER ET MONTER UN SYSTÈME DE FICHIERS EXT3	23
2.9. EXEMPLE DE PLAYBOOK ANSIBLE POUR REDIMENSIONNER UN SYSTÈME DE FICHIERS EXT4 OU EXT3 EXISTANT À L'AIDE DU RÔLE DE SYSTÈME RHEL STORAGE	24
2.10. EXEMPLE DE PLAYBOOK ANSIBLE POUR REDIMENSIONNER UN SYSTÈME DE FICHIERS EXISTANT SUR LVM À L'AIDE DU RÔLE SYSTÈME STORAGE RHEL	25
2.11. EXEMPLE DE PLAYBOOK ANSIBLE POUR CRÉER UN VOLUME D'ÉCHANGE À L'AIDE DU RÔLE DE SYSTÈME RHEL STORAGE	26
2.12. CONFIGURATION D'UN VOLUME RAID À L'AIDE DU RÔLE DE SYSTÈME DE STOCKAGE	27
2.13. CONFIGURATION D'UN POOL LVM AVEC RAID À L'AIDE DU RÔLE SYSTÈME STORAGE RHEL	28
2.14. EXEMPLE DE PLAYBOOK ANSIBLE POUR COMPRESSER ET DÉDUPLIQUER UN VOLUME VDO SUR LVM À L'AIDE DU RÔLE SYSTÈME STORAGE RHEL	29
2.15. CRÉATION D'UN VOLUME CHIFFRÉ LUKS2 À L'AIDE DU RÔLE SYSTÈME STORAGE RHEL	30
2.16. EXEMPLE DE PLAYBOOK ANSIBLE POUR EXPRIMER LES TAILLES DE VOLUME DE POOL EN POURCENTAGE À L'AIDE DU RÔLE DE SYSTÈME RHEL STORAGE	32
2.17. RESSOURCES SUPPLÉMENTAIRES	33
CHAPITRE 3. MONTAGE DES PARTAGES NFS	34
3.1. INTRODUCTION À NFS	34
3.2. VERSIONS NFS PRISES EN CHARGE	34
3.3. SERVICES REQUIS PAR LES SNF	35
3.4. FORMATS DES NOMS D'HÔTES NFS	36
3.5. INSTALLATION DE NFS	37
3.6. DÉCOUVRIR LES EXPORTATIONS NFS	37
3.7. MONTAGE D'UN PARTAGE NFS AVEC MOUNT	38
3.8. OPTIONS DE MONTAGE NFS COURANTES	39

3.9. RESSOURCES SUPPLÉMENTAIRES	40
CHAPITRE 4. EXPORTATION DE PARTAGES NFS	41
4.1. INTRODUCTION À NFS	41
4.2. VERSIONS NFS PRISES EN CHARGE	41
4.3. LES PROTOCOLES TCP ET UDP DANS NFSV3 ET NFSV4	42
4.4. SERVICES REQUIS PAR LES SNF	42
4.5. FORMATS DES NOMS D'HÔTES NFS	43
4.6. CONFIGURATION DU SERVEUR NFS	44
4.7. NFS ET RPCBIND	47
4.8. INSTALLATION DE NFS	48
4.9. DÉMARRAGE DU SERVEUR NFS	48
4.10. DÉPANNAGE DE NFS ET RPCBIND	48
4.11. CONFIGURER LE SERVEUR NFS POUR QU'IL FONCTIONNE DERRIÈRE UN PARE-FEU	49
4.12. EXPORTER DES QUOTAS RPC À TRAVERS UN PARE-FEU	53
4.13. ACTIVATION DE NFS SUR RDMA (NFSORDMA)	54
4.14. RESSOURCES SUPPLÉMENTAIRES	55
CHAPITRE 5. CONFIGURATION D'UN SERVEUR NFSV4 UNIQUEMENT	56
5.1. AVANTAGES ET INCONVÉNIENTS D'UN SERVEUR NFSV4 UNIQUEMENT	56
5.2. CONFIGURER LE SERVEUR NFS POUR QU'IL NE PRENNE EN CHARGE QUE NFSV4	56
5.3. VÉRIFICATION DE LA CONFIGURATION NFSV4-ONLY	57
CHAPITRE 6. SÉCURISATION DE NFS	59
6.1. SÉCURITÉ NFS AVEC AUTH_SYS ET CONTRÔLE DES EXPORTATIONS	59
6.2. SÉCURITÉ NFS AVEC AUTH_GSS	59
6.3. CONFIGURATION D'UN SERVEUR ET D'UN CLIENT NFS POUR L'UTILISATION DE KERBEROS	60
6.4. OPTIONS DE SÉCURITÉ NFSV4	61
6.5. DROITS D'ACCÈS AUX FICHIERS SUR LES EXPORTATIONS NFS MONTÉES	61
CHAPITRE 7. ACTIVATION DES CONFIGURATIONS SCSI PNFS DANS NFS	62
7.1. LA TECHNOLOGIE PNFS	62
7.2. DISPOSITIONS SCSI PNFS	62
7.3. RECHERCHE D'UN PÉRIPHÉRIQUE SCSI COMPATIBLE AVEC PNFS	63
7.4. CONFIGURATION DE PNFS SCSI SUR LE SERVEUR	64
7.5. CONFIGURATION DE PNFS SCSI SUR LE CLIENT	64
7.6. LIBÉRATION DE LA RÉSERVATION SCSI PNFS SUR LE SERVEUR	65
CHAPITRE 8. SURVEILLANCE DE LA FONCTIONNALITÉ DES LAYOUTS SCSI PNFS	67
8.1. VÉRIFICATION DES OPÉRATIONS SCSI PNFS À PARTIR DU SERVEUR À L'AIDE DE NFSSTAT	67
8.2. VÉRIFICATION DES OPÉRATIONS SCSI PNFS À PARTIR DU CLIENT À L'AIDE DE MOUNTSTATS	67
CHAPITRE 9. PREMIERS PAS AVEC FS-CACHE	69
9.1. VUE D'ENSEMBLE DU FS-CACHE	69
9.2. GARANTIE DE PERFORMANCE	70
9.3. MISE EN PLACE D'UN CACHE	70
9.4. CONFIGURATION DES LIMITES DU CACHE	72
9.5. RÉCUPÉRATION D'INFORMATIONS STATISTIQUES DU MODULE NOYAU FSCACHE	73
9.6. RÉFÉRENCES FS-CACHE	73
CHAPITRE 10. UTILISATION DU CACHE AVEC NFS	74
10.1. CONFIGURATION DU PARTAGE DU CACHE NFS	74
10.2. LIMITATIONS DU CACHE AVEC NFS	75
CHAPITRE 11. MONTAGE D'UN PARTAGE SMB SOUS RED HAT ENTERPRISE LINUX	77

11.1. VERSIONS DU PROTOCOLE SMB PRISES EN CHARGE	77
11.2. SUPPORT DES EXTENSIONS UNIX	78
11.3. MONTAGE MANUEL D'UN PARTAGE SMB	78
11.4. MONTAGE AUTOMATIQUE D'UN PARTAGE SMB AU DÉMARRAGE DU SYSTÈME	79
11.5. AUTHENTIFICATION À UN PARTAGE SMB À L'AIDE D'UN FICHIER D'INFORMATIONS D'IDENTIFICATION	80
11.6. OPTIONS DE MONTAGE FRÉQUEMMENT UTILISÉES	80
CHAPITRE 12. EXÉCUTION D'UN MONTAGE SMB MULTI-UTILISATEURS	82
12.1. MONTAGE D'UN PARTAGE AVEC L'OPTION MULTIUTILISATEUR	82
12.2. VÉRIFIER SI UN PARTAGE SMB EST MONTÉ AVEC L'OPTION MULTIUTILISATEUR	83
12.3. ACCÈS À UN PARTAGE EN TANT QU'UTILISATEUR	83
CHAPITRE 13. VUE D'ENSEMBLE DES ATTRIBUTS DE DÉNOMINATION PERSISTANTS	84
13.1. INCONVÉNIENTS DES ATTRIBUTS DE DÉNOMINATION NON PERSISTANTS	84
13.2. SYSTÈME DE FICHIERS ET IDENTIFICATEURS DE PÉRIPHÉRIQUES	85
13.3. NOMS DES PÉRIPHÉRIQUES GÉRÉS PAR LE MÉCANISME UDEV DANS /DEV/DISK/	85
13.4. LE WORLD WIDE IDENTIFIER AVEC DM MULTIPATH	88
13.5. LIMITES DE LA CONVENTION DE DÉNOMINATION DES PÉRIPHÉRIQUES UDEV	88
13.6. LISTE DES ATTRIBUTS DE NOMS PERSISTANTS	89
13.7. MODIFICATION DES ATTRIBUTS DE DÉNOMINATION PERSISTANTS	90
CHAPITRE 14. OPÉRATIONS DE PARTITIONNEMENT AVEC PARTED	92
14.1. AFFICHAGE DE LA TABLE DE PARTITION AVEC PARTED	92
14.2. CRÉATION D'UNE TABLE DE PARTITION SUR UN DISQUE AVEC PARTED	93
14.3. CRÉATION D'UNE PARTITION AVEC PARTED	94
14.4. SUPPRESSION D'UNE PARTITION AVEC PARTED	96
14.5. REDIMENSIONNEMENT D'UNE PARTITION AVEC PARTED	97
CHAPITRE 15. STRATÉGIES DE REPARTITIONNEMENT D'UN DISQUE	99
15.1. UTILISATION DE L'ESPACE LIBRE NON PARTITIONNÉ	99
15.2. UTILISATION DE L'ESPACE D'UNE PARTITION INUTILISÉE	99
15.3. UTILISATION DE L'ESPACE LIBRE D'UNE PARTITION ACTIVE	100
CHAPITRE 16. PREMIERS PAS AVEC XFS	104
16.1. LE SYSTÈME DE FICHIERS XFS	104
16.2. COMPARAISON DES OUTILS UTILISÉS AVEC EXT4 ET XFS	105
CHAPITRE 17. CRÉATION D'UN SYSTÈME DE FICHIERS XFS	107
17.1. CRÉATION D'UN SYSTÈME DE FICHIERS XFS AVEC MKFS.XFS	107
CHAPITRE 18. SAUVEGARDE D'UN SYSTÈME DE FICHIERS XFS	108
18.1. CARACTÉRISTIQUES DE LA SAUVEGARDE XFS	108
18.2. SAUVEGARDE D'UN SYSTÈME DE FICHIERS XFS AVEC XFSDDUMP	108
18.3. RESSOURCES SUPPLÉMENTAIRES	109
CHAPITRE 19. RESTAURATION D'UN SYSTÈME DE FICHIERS XFS À PARTIR D'UNE SAUVEGARDE	110
19.1. FONCTIONNALITÉS DE LA RESTAURATION DE XFS À PARTIR D'UNE SAUVEGARDE	110
19.2. RESTAURATION D'UN SYSTÈME DE FICHIERS XFS À PARTIR D'UNE SAUVEGARDE AVEC XFSRESTORE	110
19.3. MESSAGES D'INFORMATION LORS DE LA RESTAURATION D'UNE SAUVEGARDE XFS À PARTIR D'UNE BANDE	111
19.4. RESSOURCES SUPPLÉMENTAIRES	112
CHAPITRE 20. AUGMENTER LA TAILLE D'UN SYSTÈME DE FICHIERS XFS	113
20.1. AUGMENTER LA TAILLE D'UN SYSTÈME DE FICHIERS XFS AVEC XFS_GROWFS	113

CHAPITRE 21. CONFIGURATION DU COMPORTEMENT DES ERREURS XFS	114
21.1. GESTION CONFIGURABLE DES ERREURS DANS XFS	114
21.2. FICHIERS DE CONFIGURATION POUR LES CONDITIONS D'ERREUR XFS SPÉCIFIQUES ET NON DÉFINIES	114
21.3. DÉFINITION DU COMPORTEMENT DE XFS DANS DES CONDITIONS SPÉCIFIQUES	115
21.4. DÉFINITION DU COMPORTEMENT DE XFS EN CAS DE CONDITIONS NON DÉFINIES	115
21.5. PARAMÉTRAGE DU COMPORTEMENT DE DÉMONTAGE DE XFS	116
CHAPITRE 22. VÉRIFICATION ET RÉPARATION D'UN SYSTÈME DE FICHIERS	117
22.1. SCÉNARIOS NÉCESSITANT UNE VÉRIFICATION DU SYSTÈME DE FICHIERS	117
22.2. EFFETS SECONDAIRES POTENTIELS DE L'EXÉCUTION DE FSCK	118
22.3. MÉCANISMES DE TRAITEMENT DES ERREURS DANS LE SYSTÈME XFS	118
22.4. VÉRIFICATION D'UN SYSTÈME DE FICHIERS XFS AVEC XFS_REPAIR	119
22.5. RÉPARATION D'UN SYSTÈME DE FICHIERS XFS AVEC XFS_REPAIR	120
22.6. MÉCANISMES DE GESTION DES ERREURS DANS EXT2, EXT3 ET EXT4	121
22.7. VÉRIFICATION D'UN SYSTÈME DE FICHIERS EXT2, EXT3 OU EXT4 AVEC E2FSCK	122
22.8. RÉPARATION D'UN SYSTÈME DE FICHIERS EXT2, EXT3 OU EXT4 AVEC E2FSCK	122
CHAPITRE 23. MONTAGE DES SYSTÈMES DE FICHIERS	124
23.1. LE MÉCANISME DE MONTAGE DE LINUX	124
23.2. LISTE DES SYSTÈMES DE FICHIERS ACTUELLEMENT MONTÉS	124
23.3. MONTAGE D'UN SYSTÈME DE FICHIERS AVEC MOUNT	125
23.4. DÉPLACEMENT D'UN POINT DE MONTAGE	126
23.5. DÉMONTAGE D'UN SYSTÈME DE FICHIERS AVEC UMount	126
23.6. OPTIONS DE MONTAGE COURANTES	127
CHAPITRE 24. PARTAGE D'UN MONTAGE SUR PLUSIEURS POINTS DE MONTAGE	129
24.1. TYPES DE MONTAGES PARTAGÉS	129
24.2. CRÉATION D'UN DUPLICATA DE POINT DE MONTAGE PRIVÉ	129
24.3. CRÉATION D'UN POINT DE MONTAGE PARTAGÉ EN DOUBLE	130
24.4. CRÉATION D'UN DUPLICATA DE POINT DE MONTAGE ESCLAVE	132
24.5. EMPÊCHER LA DUPLICATION D'UN POINT DE MONTAGE	133
CHAPITRE 25. MONTAGE PERSISTANT DES SYSTÈMES DE FICHIERS	135
25.1. LE FICHIER /ETC/FSTAB	135
25.2. AJOUT D'UN SYSTÈME DE FICHIERS À /ETC/FSTAB	136
CHAPITRE 26. MONTAGE DE SYSTÈMES DE FICHIERS À LA DEMANDE	137
26.1. LE SERVICE AUTOFS	137
26.2. LES FICHIERS DE CONFIGURATION D'AUTOFS	137
26.3. CONFIGURATION DES POINTS DE MONTAGE AUTOFS	139
26.4. MONTAGE AUTOMATIQUE DES RÉPERTOIRES PERSONNELS DES UTILISATEURS DU SERVEUR NFS AVEC LE SERVICE AUTOFS	140
26.5. REMPLACER OU COMPLÉTER LES FICHIERS DE CONFIGURATION DU SITE AUTOFS	140
26.6. UTILISATION DE LDAP POUR STOCKER DES CARTES DE COMPTEURS AUTOMATIQUES	142
26.7. UTILISER SYSTEMD.AUTOMOUNT POUR MONTER UN SYSTÈME DE FICHIERS À LA DEMANDE AVEC /ETC/FSTAB	143
26.8. UTILISATION DE SYSTEMD.AUTOMOUNT POUR MONTER UN SYSTÈME DE FICHIERS À LA DEMANDE AVEC UNE UNITÉ DE MONTAGE	144
CHAPITRE 27. UTILISATION DU COMPOSANT SSSD DE L'IDM POUR METTRE EN CACHE LES CARTES AUTOFS	146
27.1. CONFIGURATION MANUELLE D'AUTOFS POUR UTILISER LE SERVEUR IDM COMME SERVEUR LDAP	146
27.2. CONFIGURATION DE SSSD POUR LA MISE EN CACHE DES CARTES AUTOFS	147

CHAPITRE 28. DÉFINITION DES AUTORISATIONS DE LECTURE SEULE POUR LE SYSTÈME DE FICHIERS RACINE	149
28.1. FICHIERS ET RÉPERTOIRES QUI CONSERVENT TOUJOURS LES DROITS D'ÉCRITURE	149
28.2. CONFIGURER LE SYSTÈME DE FICHIERS RACINE POUR QU'IL SOIT MONTÉ AVEC DES AUTORISATIONS DE LECTURE SEULE AU DÉMARRAGE	150
CHAPITRE 29. LIMITER L'UTILISATION DE L'ESPACE DE STOCKAGE SUR XFS AVEC DES QUOTAS ...	152
29.1. QUOTAS DE DISQUES	152
29.2. L'OUTIL XFS_QUOTA	152
29.3. GESTION DES QUOTAS DANS LE SYSTÈME DE FICHIERS XFS	153
29.4. ACTIVATION DES QUOTAS DE DISQUE POUR XFS	153
29.5. RAPPORT SUR L'UTILISATION DE XFS	154
29.6. MODIFICATION DES LIMITES DE QUOTAS XFS	154
29.7. DÉFINITION DES LIMITES DU PROJET POUR XFS	155
CHAPITRE 30. LIMITER L'UTILISATION DE L'ESPACE DE STOCKAGE SUR EXT4 AVEC DES QUOTAS ..	157
30.1. INSTALLATION DE L'OUTIL QUOTA	157
30.2. ACTIVATION DE LA FONCTION DE QUOTA LORS DE LA CRÉATION D'UN SYSTÈME DE FICHIERS	157
30.3. ACTIVATION DE LA FONCTION DE QUOTA SUR LES SYSTÈMES DE FICHIERS EXISTANTS	157
30.4. ACTIVATION DE L'APPLICATION DES QUOTAS	158
30.5. ATTRIBUTION DE QUOTAS PAR UTILISATEUR	159
30.6. ATTRIBUTION DE QUOTAS PAR GROUPE	160
30.7. ATTRIBUTION DE QUOTAS PAR PROJET	161
30.8. DÉFINITION DU DÉLAI DE GRÂCE POUR LES LIMITES SOUPLES	162
30.9. DÉSACTIVER LES QUOTAS DE SYSTÈME DE FICHIERS	162
30.10. RAPPORT SUR LES QUOTAS DE DISQUE	163
CHAPITRE 31. MISE AU REBUT DES BLOCS INUTILISÉS	164
Requirements	164
31.1. TYPES D'OPÉRATIONS D'ANNULATION DE BLOCS	164
31.2. EXÉCUTION DE L'ÉLIMINATION DES BLOCS PAR LOTS	164
31.3. ACTIVATION DE L'ÉLIMINATION DES BLOCS EN LIGNE	165
31.4. ACTIVATION DE L'ÉLIMINATION PÉRIODIQUE DES BLOCS	165
CHAPITRE 32. CONFIGURATION DES SYSTÈMES DE FICHIERS STRATIS	167
32.1. QU'EST-CE QUE STRATIS ?	167
32.2. COMPOSANTS D'UN VOLUME STRATIS	167
32.3. DISPOSITIFS DE BLOCAGE UTILISABLES AVEC STRATIS	168
32.4. INSTALLATION DE STRATIS	169
32.5. CRÉATION D'UN POOL STRATIS NON CHIFFRÉ	169
32.6. CRÉATION D'UN POOL STRATIS CRYPTÉ	170
32.7. MISE EN PLACE D'UNE COUCHE DE PROVISIONNEMENT FIN DANS LE SYSTÈME DE FICHIERS STRATIS	172
32.8. LIER UN POOL STRATIS À L'EDNB	173
32.9. LIER UN POOL STRATIS À UNE MPT	174
32.10. DÉVERROUILLER UN POOL STRATIS CRYPTÉ AVEC LE TROUSSEAU DE CLÉS DU NOYAU	174
32.11. DÉVERROUILLER UN POOL STRATIS CRYPTÉ AVEC CLEVIS	175
32.12. DÉTACHER UN POOL STRATIS DU CHIFFREMENT SUPPLÉMENTAIRE	175
32.13. DÉMARRAGE ET ARRÊT DU POOL STRATIS	176
32.14. CRÉATION D'UN SYSTÈME DE FICHIERS STRATIS	177
32.15. MONTAGE D'UN SYSTÈME DE FICHIERS STRATIS	178
32.16. MONTAGE PERSISTANT D'UN SYSTÈME DE FICHIERS STRATIS	178
32.17. CONFIGURATION DE SYSTÈMES DE FICHIERS STRATIS NON ROOT DANS /ETC/FSTAB À L'AIDE D'UN SERVICE SYSTEMD	179

CHAPITRE 33. EXTENSION D'UN VOLUME STRATIS AVEC DES PÉRIPHÉRIQUES DE BLOC SUPPLÉMENTAIRES	181
33.1. COMPOSANTS D'UN VOLUME STRATIS	181
33.2. AJOUT DE BLOCS À UN POOL STRATIS	182
33.3. RESSOURCES SUPPLÉMENTAIRES	182
CHAPITRE 34. SURVEILLANCE DES SYSTÈMES DE FICHIERS STRATIS	183
34.1. TAILLES DES STRATIS RAPPORTÉES PAR LES DIFFÉRENTS SERVICES PUBLICS	183
34.2. AFFICHAGE D'INFORMATIONS SUR LES VOLUMES STRATIS	183
34.3. RESSOURCES SUPPLÉMENTAIRES	184
CHAPITRE 35. UTILISATION D'INSTANTANÉS SUR LES SYSTÈMES DE FICHIERS STRATIS	185
35.1. CARACTÉRISTIQUES DES INSTANTANÉS STRATIS	185
35.2. CRÉATION D'UN INSTANTANÉ STRATIS	185
35.3. ACCÉDER AU CONTENU D'UN INSTANTANÉ STRATIS	186
35.4. REVENIR À UN INSTANTANÉ PRÉCÉDENT D'UN SYSTÈME DE FICHIERS STRATIS	186
35.5. SUPPRESSION D'UN INSTANTANÉ STRATIS	187
35.6. RESSOURCES SUPPLÉMENTAIRES	187
CHAPITRE 36. SUPPRESSION DES SYSTÈMES DE FICHIERS STRATIS	188
36.1. COMPOSANTS D'UN VOLUME STRATIS	188
36.2. SUPPRESSION D'UN SYSTÈME DE FICHIERS STRATIS	189
36.3. SUPPRESSION D'UN POOL STRATIS	189
36.4. RESSOURCES SUPPLÉMENTAIRES	190
CHAPITRE 37. PREMIERS PAS AVEC UN SYSTÈME DE FICHIERS EXT4	191
37.1. CARACTÉRISTIQUES D'UN SYSTÈME DE FICHIERS EXT4	191
37.2. CRÉATION D'UN SYSTÈME DE FICHIERS EXT4	192
37.3. MONTAGE D'UN SYSTÈME DE FICHIERS EXT4	193
37.4. REDIMENSIONNEMENT D'UN SYSTÈME DE FICHIERS EXT4	194
37.5. COMPARAISON DES OUTILS UTILISÉS AVEC EXT4 ET XFS	195

RENDRE L'OPEN SOURCE PLUS INCLUSIF

Red Hat s'engage à remplacer les termes problématiques dans son code, sa documentation et ses propriétés Web. Nous commençons par ces quatre termes : master, slave, blacklist et whitelist. En raison de l'ampleur de cette entreprise, ces changements seront mis en œuvre progressivement au cours de plusieurs versions à venir. Pour plus de détails, voir le [message de notre directeur technique Chris Wright](#).

FOURNIR UN RETOUR D'INFORMATION SUR LA DOCUMENTATION DE RED HAT

Nous apprécions vos commentaires sur notre documentation. Faites-nous savoir comment nous pouvons l'améliorer.

Soumettre des commentaires sur des passages spécifiques

1. Consultez la documentation au format **Multi-page HTML** et assurez-vous que le bouton **Feedback** apparaît dans le coin supérieur droit après le chargement complet de la page.
2. Utilisez votre curseur pour mettre en évidence la partie du texte que vous souhaitez commenter.
3. Cliquez sur le bouton **Add Feedback** qui apparaît près du texte en surbrillance.
4. Ajoutez vos commentaires et cliquez sur **Submit**.

Soumettre des commentaires via Bugzilla (compte requis)

1. Connectez-vous au site Web de [Bugzilla](#).
2. Sélectionnez la version correcte dans le menu **Version**.
3. Saisissez un titre descriptif dans le champ **Summary**.
4. Saisissez votre suggestion d'amélioration dans le champ **Description**. Incluez des liens vers les parties pertinentes de la documentation.
5. Cliquez sur **Submit Bug**.

CHAPITRE 1. APERÇU DES SYSTÈMES DE FICHIERS DISPONIBLES

Le choix du système de fichiers approprié à votre application est une décision importante en raison du grand nombre d'options disponibles et des compromis qu'elles impliquent.

Les sections suivantes décrivent les systèmes de fichiers que Red Hat Enterprise Linux 9 inclut par défaut, ainsi que des recommandations sur le système de fichiers le plus approprié pour votre application.

1.1. TYPES DE SYSTÈMES DE FICHIERS

Red Hat Enterprise Linux 9 prend en charge une variété de systèmes de fichiers (FS). Les différents types de systèmes de fichiers résolvent différents types de problèmes, et leur utilisation est spécifique à l'application. Au niveau le plus général, les systèmes de fichiers disponibles peuvent être regroupés dans les principaux types suivants :

Tableau 1.1. Types de systèmes de fichiers et leurs cas d'utilisation

Type	Système de fichiers	Attributs et cas d'utilisation
Disque ou FS local	XFS	XFS est le système de fichiers par défaut de RHEL. Étant donné qu'il présente les fichiers sous forme d'extensions, il est moins vulnérable à la fragmentation que ext4. Red Hat recommande de déployer XFS en tant que système de fichiers local à moins qu'il n'y ait des raisons spécifiques de faire autrement : par exemple, la compatibilité ou des cas particuliers concernant la performance.
	ext4	ext4 a l'avantage de la longévité sous Linux. Par conséquent, il est pris en charge par presque toutes les applications Linux. Dans la plupart des cas, il rivalise avec XFS en termes de performances. ext4 est couramment utilisé pour les répertoires personnels.
FS en réseau ou client-serveur	NFS	Utiliser NFS pour partager des fichiers entre plusieurs systèmes sur le même réseau.
	PME	Utiliser SMB pour le partage de fichiers avec les systèmes Microsoft Windows.
Stockage partagé ou disque partagé FS	GFS2	GFS2 fournit un accès partagé en écriture aux membres d'une grappe de calcul. L'accent est mis sur la stabilité et la fiabilité, avec l'expérience fonctionnelle d'un système de fichiers local autant que possible. SAS Grid, Tibco MQ, IBM Websphere MQ et Red Hat Active MQ ont été déployés avec succès sur GFS2.

Type	Système de fichiers	Attributs et cas d'utilisation
FS à gestion de volume	Stratis (aperçu technologique)	Stratis est un gestionnaire de volume basé sur une combinaison de XFS et de LVM. L'objectif de Stratis est d'émuler les capacités offertes par les systèmes de fichiers de gestion de volume tels que Btrfs et ZFS. Il est possible de construire cette pile manuellement, mais Stratis réduit la complexité de la configuration, met en œuvre les meilleures pratiques et consolide les informations d'erreur.

1.2. SYSTÈMES DE FICHIERS LOCAUX

Les systèmes de fichiers locaux sont des systèmes de fichiers qui s'exécutent sur un seul serveur local et sont directement attachés au stockage.

Par exemple, un système de fichiers local est le seul choix possible pour les disques internes SATA ou SAS, et il est utilisé lorsque votre serveur dispose de contrôleurs RAID matériels internes avec des disques locaux. Les systèmes de fichiers locaux sont également les systèmes de fichiers les plus couramment utilisés sur le stockage attaché au SAN lorsque le périphérique exporté sur le SAN n'est pas partagé.

Tous les systèmes de fichiers locaux sont conformes à POSIX et sont entièrement compatibles avec toutes les versions de Red Hat Enterprise Linux prises en charge. Les systèmes de fichiers conformes à POSIX prennent en charge un ensemble bien défini d'appels système, tels que **read()**, **write()**, et **seek()**.

Du point de vue du programmeur d'applications, il y a relativement peu de différences entre les systèmes de fichiers locaux. Les différences les plus notables du point de vue de l'utilisateur sont liées à l'évolutivité et aux performances. Lorsque vous envisagez de choisir un système de fichiers, réfléchissez à sa taille, aux caractéristiques uniques qu'il doit posséder et à ses performances dans le cadre de votre charge de travail.

Systèmes de fichiers locaux disponibles

- XFS
- ext4

1.3. LE SYSTÈME DE FICHIERS XFS

XFS est un système de fichiers de journalisation 64 bits hautement évolutif, performant, robuste et mature qui prend en charge des fichiers et des systèmes de fichiers très volumineux sur un seul hôte. Il s'agit du système de fichiers par défaut de Red Hat Enterprise Linux 9. XFS a été développé au début des années 1990 par SGI et fonctionne depuis longtemps sur des serveurs et des baies de stockage de très grande taille.

Les caractéristiques de XFS sont les suivantes

Fiabilité

- La journalisation des métadonnées, qui garantit l'intégrité du système de fichiers après une

panne du système en conservant un enregistrement des opérations du système de fichiers qui peut être rejoué lorsque le système est redémarré et que le système de fichiers est remonté

- Contrôle étendu de la cohérence des métadonnées au moment de l'exécution
- Utilitaires de réparation évolutifs et rapides
- Journalisation des quotas. Cela évite d'avoir à effectuer de longs contrôles de cohérence des quotas après une panne.

Évolutivité et performance

- Taille du système de fichiers pris en charge jusqu'à 1024 TiB
- Capacité à prendre en charge un grand nombre d'opérations simultanées
- Indexation B-tree pour l'extensibilité de la gestion de l'espace libre
- Algorithmes sophistiqués de lecture anticipée des métadonnées
- Optimisations pour les charges de travail liées à la vidéo en continu

Systemes d'allocation

- Allocation basée sur l'étendue
- Politiques d'allocation tenant compte des bandes
- Attribution retardée
- Pré-affectation de l'espace
- Inodes alloués dynamiquement

Autres caractéristiques

- Copies de fichiers basées sur des liens hypertextes
- Utilitaires de sauvegarde et de restauration étroitement intégrés
- Défragmentation en ligne
- Croissance du système de fichiers en ligne
- Capacités de diagnostic complètes
- Attributs étendus (**xattr**). Cela permet au système d'associer plusieurs paires nom/valeur supplémentaires par fichier.
- Quotas de projet ou de répertoire. Cela permet de restreindre les quotas sur une arborescence de répertoires.
- Horodatage à la seconde près

Caractéristiques de performance

XFS est très performant sur les grands systèmes avec des charges de travail d'entreprise. Un système de grande taille est un système doté d'un nombre relativement élevé de CPU, de plusieurs HBA et de connexions à des baies de disques externes. XFS est également performant sur les systèmes plus petits qui ont une charge de travail d'E/S parallèle et multithread.

XFS est relativement peu performant pour les charges de travail à fil unique et à forte intensité de métadonnées : par exemple, une charge de travail qui crée ou supprime un grand nombre de petits fichiers en un seul fil.

1.4. LE SYSTÈME DE FICHIERS EXT4

Le système de fichiers ext4 est la quatrième génération de la famille des systèmes de fichiers ext. Il s'agit du système de fichiers par défaut dans Red Hat Enterprise Linux 6.

Le pilote ext4 peut lire et écrire sur les systèmes de fichiers ext2 et ext3, mais le format du système de fichiers ext4 n'est pas compatible avec les pilotes ext2 et ext3.

ext4 ajoute plusieurs fonctionnalités nouvelles et améliorées, telles que

- Taille du système de fichiers pris en charge : jusqu'à 50 TiB
- Métadonnées basées sur l'étendue
- Attribution retardée
- Somme de contrôle du journal
- Grand support de stockage

Les métadonnées basées sur l'étendue et les fonctions d'allocation différée offrent un moyen plus compact et plus efficace de suivre l'espace utilisé dans un système de fichiers. Ces fonctionnalités améliorent les performances du système de fichiers et réduisent l'espace consommé par les métadonnées. L'allocation différée permet au système de fichiers de différer la sélection de l'emplacement permanent des données utilisateur nouvellement écrites jusqu'à ce que les données soient transférées sur le disque. Cela permet d'améliorer les performances, car les allocations sont plus importantes et plus contiguës, ce qui permet au système de fichiers de prendre des décisions sur la base d'informations plus précises.

Le temps de réparation du système de fichiers à l'aide de l'utilitaire **fsck** dans ext4 est beaucoup plus rapide que dans ext2 et ext3. Certaines réparations de systèmes de fichiers ont permis de multiplier les performances par six.

1.5. COMPARAISON ENTRE XFS ET EXT4

XFS est le système de fichiers par défaut de RHEL. Cette section compare l'utilisation et les fonctionnalités de XFS et ext4.

Comportement en cas d'erreur de métadonnées

Dans ext4, vous pouvez configurer le comportement lorsque le système de fichiers rencontre des erreurs de métadonnées. Le comportement par défaut consiste à poursuivre l'opération. Lorsque XFS rencontre une erreur de métadonnées irrécupérable, il ferme le système de fichiers et renvoie l'erreur **EFSCORRUPTED**.

Quotas

Dans ext4, vous pouvez activer les quotas lors de la création du système de fichiers ou ultérieurement sur un système de fichiers existant. Vous pouvez ensuite configurer l'application des quotas à l'aide d'une option de montage.

Les quotas XFS ne sont pas une option de remontage. Vous devez activer les quotas lors du montage initial.

L'exécution de la commande **quotacheck** sur un système de fichiers XFS n'a aucun effet. La première fois que vous activez la comptabilisation des quotas, XFS vérifie automatiquement les quotas.

Redimensionnement du système de fichiers

XFS ne dispose d'aucun utilitaire permettant de réduire la taille d'un système de fichiers. Vous ne pouvez qu'augmenter la taille d'un système de fichiers XFS. En comparaison, ext4 permet à la fois d'étendre et de réduire la taille d'un système de fichiers.

Numéros d'inodes

Le système de fichiers ext4 ne prend pas en charge plus de²³² inodes.

XFS alloue dynamiquement des inodes. Un système de fichiers XFS ne peut pas manquer d'inodes tant qu'il y a de l'espace libre sur le système de fichiers.

Certaines applications ne peuvent pas gérer correctement les numéros d'inodes supérieurs à²³² sur un système de fichiers XFS. Ces applications peuvent entraîner l'échec des appels stat 32 bits avec la valeur de retour **Eoverflow**. Le numéro d'inode dépasse²³² dans les conditions suivantes :

- Le système de fichiers est supérieur à 1 TiB avec des inodes de 256 octets.
- Le système de fichiers est supérieur à 2 TiB avec des inodes de 512 octets.

Si votre application échoue avec de grands nombres d'inodes, montez le système de fichiers XFS avec l'option **-o inode32** pour imposer des nombres d'inodes inférieurs à²³². Notez que l'utilisation de l'option **inode32** n'affecte pas les inodes déjà alloués avec des numéros de 64 bits.



IMPORTANT

not N'utilisez pas l'option **inode32** à moins qu'un environnement spécifique ne l'exige. L'option **inode32** modifie le comportement de l'allocation. Par conséquent, l'erreur **ENOSPC** peut se produire si aucun espace n'est disponible pour allouer des inodes dans les blocs de disque inférieurs.

1.6. CHOIX D'UN SYSTÈME DE FICHIERS LOCAL

Pour choisir un système de fichiers qui réponde aux exigences de votre application, vous devez comprendre le système cible sur lequel vous allez déployer le système de fichiers. Les questions suivantes peuvent vous aider à prendre votre décision :

- Avez-vous un grand serveur ?
- Vous avez des besoins importants en matière de stockage ou vous disposez d'un disque SATA local et lent ?
- Quel type de charge de travail d'E/S attendez-vous de votre application ?
- Quels sont vos besoins en termes de débit et de latence ?

- Quelle est la stabilité de votre serveur et de votre matériel de stockage ?
- Quelle est la taille habituelle de vos fichiers et de vos données ?
- En cas de défaillance du système, combien de temps d'arrêt pouvez-vous subir ?

Si votre serveur et votre périphérique de stockage sont de grande taille, XFS est le meilleur choix. Même avec des baies de stockage plus petites, XFS fonctionne très bien lorsque la taille moyenne des fichiers est importante (par exemple, des centaines de mégaoctets).

Si votre charge de travail actuelle a bien fonctionné avec ext4, vous et vos applications devraient bénéficier d'un environnement très familier.

Le système de fichiers ext4 a tendance à mieux fonctionner sur les systèmes dont la capacité d'E/S est limitée. Il est plus performant sur une bande passante limitée (moins de 200 Mo/s) et jusqu'à une capacité d'environ 1 000 IOPS. Pour tout ce qui a une capacité supérieure, XFS a tendance à être plus rapide.

XFS consomme environ deux fois plus de CPU par opération de métadonnées que ext4, donc si vous avez une charge de travail liée au CPU avec peu de concurrence, ext4 sera plus rapide. En général, ext4 est meilleur si une application utilise un seul thread de lecture/écriture et des fichiers de petite taille, tandis que XFS brille lorsqu'une application utilise plusieurs threads de lecture/écriture et des fichiers plus volumineux.

Vous ne pouvez pas réduire un système de fichiers XFS. Si vous avez besoin de pouvoir réduire le système de fichiers, envisagez d'utiliser ext4, qui prend en charge la réduction hors ligne.

En général, Red Hat vous recommande d'utiliser XFS à moins que vous n'ayez un cas d'utilisation spécifique pour ext4. Vous devriez également mesurer les performances de votre application spécifique sur votre serveur cible et votre système de stockage afin de vous assurer que vous choisissiez le type de système de fichiers approprié.

Tableau 1.2. Résumé des recommandations relatives aux systèmes de fichiers locaux

Scénario	Système de fichiers recommandé
Pas de cas d'utilisation particulier	XFS
Grand serveur	XFS
Dispositifs de stockage de grande taille	XFS
Fichiers volumineux	XFS
E/S multithreadées	XFS
E/S monotâches	ext4
Capacité d'E/S limitée (moins de 1000 IOPS)	ext4
Largeur de bande limitée (moins de 200 Mo/s)	ext4
Charge de travail liée à l'unité centrale	ext4

Scénario	Système de fichiers recommandé
Prise en charge de la réduction hors ligne	ext4

1.7. SYSTÈMES DE FICHIERS EN RÉSEAU

Les systèmes de fichiers en réseau, également appelés systèmes de fichiers client/serveur, permettent aux systèmes clients d'accéder aux fichiers stockés sur un serveur partagé. Cela permet à plusieurs utilisateurs sur plusieurs systèmes de partager des fichiers et des ressources de stockage.

Ces systèmes de fichiers sont construits à partir d'un ou plusieurs serveurs qui exportent un ensemble de systèmes de fichiers vers un ou plusieurs clients. Les nœuds clients n'ont pas accès au stockage en bloc sous-jacent, mais interagissent avec le stockage à l'aide d'un protocole qui permet un meilleur contrôle d'accès.

Systèmes de fichiers réseau disponibles

- Le système de fichiers client/serveur le plus courant pour les clients RHEL est le système de fichiers NFS. RHEL fournit à la fois un composant serveur NFS pour exporter un système de fichiers local sur le réseau et un client NFS pour importer ces systèmes de fichiers.
- RHEL comprend également un client CIFS qui prend en charge les célèbres serveurs de fichiers Microsoft SMB pour l'interopérabilité avec Windows. Le serveur Samba de l'espace utilisateur fournit aux clients Windows un service Microsoft SMB à partir d'un serveur RHEL.

1.8. SYSTÈMES DE FICHIERS À STOCKAGE PARTAGÉ

Les systèmes de fichiers à stockage partagé, parfois appelés systèmes de fichiers en grappe, permettent à chaque serveur de la grappe d'accéder directement à un périphérique de bloc partagé via un réseau de stockage local (SAN).

Comparaison avec les systèmes de fichiers en réseau

Comme les systèmes de fichiers client/serveur, les systèmes de fichiers à stockage partagé fonctionnent sur un ensemble de serveurs qui sont tous membres d'une grappe. Toutefois, contrairement à NFS, aucun serveur ne fournit d'accès aux données ou aux métadonnées aux autres membres : chaque membre de la grappe a un accès direct au même périphérique de stockage (*shared storage*) et tous les nœuds de la grappe accèdent au même ensemble de fichiers.

Concurrence

La cohérence du cache est essentielle dans un système de fichiers en grappe pour garantir la cohérence et l'intégrité des données. Il doit exister une version unique de tous les fichiers d'une grappe, visible par tous les nœuds de la grappe. Le système de fichiers doit empêcher les membres de la grappe de mettre à jour le même bloc de stockage en même temps et de provoquer une corruption des données. Pour ce faire, les systèmes de fichiers à stockage partagé utilisent un mécanisme de verrouillage large de la grappe pour arbitrer l'accès au stockage en tant que mécanisme de contrôle de la concurrence. Par exemple, avant de créer un nouveau fichier ou d'écrire dans un fichier ouvert sur plusieurs serveurs, le composant du système de fichiers sur le serveur doit obtenir le bon verrou.

Les systèmes de fichiers en grappe doivent fournir un service hautement disponible tel qu'un serveur web Apache. Tout membre de la grappe aura une vue parfaitement cohérente des données stockées dans leur système de fichiers sur disque partagé, et toutes les mises à jour seront arbitrées correctement par les mécanismes de verrouillage.

Caractéristiques de performance

Les systèmes de fichiers à disque partagé ne sont pas toujours aussi performants que les systèmes de fichiers locaux fonctionnant sur le même système, en raison du coût de calcul de la surcharge de verrouillage. Les systèmes de fichiers à disque partagé fonctionnent bien avec les charges de travail où chaque nœud écrit presque exclusivement sur un ensemble particulier de fichiers qui ne sont pas partagés avec d'autres nœuds ou lorsqu'un ensemble de fichiers est partagé presque exclusivement en lecture seule sur un ensemble de nœuds. Il en résulte un minimum d'invalidation du cache entre les nœuds, ce qui permet d'optimiser les performances.

La mise en place d'un système de fichiers sur disque partagé est complexe, et le réglage d'une application pour qu'elle fonctionne bien sur un système de fichiers sur disque partagé peut s'avérer difficile.

Systèmes de fichiers de stockage partagé disponibles

- Red Hat Enterprise Linux fournit le système de fichiers GFS2. GFS2 est étroitement intégré au module complémentaire de haute disponibilité de Red Hat Enterprise Linux et au module complémentaire de stockage résilient.

Red Hat Enterprise Linux prend en charge GFS2 sur des grappes dont la taille varie de 2 à 16 nœuds.

1.9. CHOISIR ENTRE LES SYSTÈMES DE FICHIERS EN RÉSEAU ET LES SYSTÈMES DE FICHIERS À STOCKAGE PARTAGÉ

Lorsque vous choisissez entre un système de fichiers en réseau et un système de stockage partagé, tenez compte des points suivants :

- Les systèmes de fichiers en réseau basés sur NFS sont un choix extrêmement courant et populaire pour les environnements qui fournissent des serveurs NFS.
- Les systèmes de fichiers en réseau peuvent être déployés à l'aide de technologies de réseau à très hautes performances comme Infiniband ou 10 Gigabit Ethernet. Cela signifie que vous ne devriez pas vous tourner vers des systèmes de fichiers de stockage partagé uniquement pour obtenir une bande passante brute vers votre stockage. Si la vitesse d'accès est primordiale, utilisez NFS pour exporter un système de fichiers local tel que XFS.
- Les systèmes de fichiers à stockage partagé ne sont pas faciles à mettre en place ou à maintenir, c'est pourquoi vous ne devez les déployer que lorsque vous ne pouvez pas assurer la disponibilité requise avec des systèmes de fichiers locaux ou en réseau.
- Un système de fichiers de stockage partagé dans un environnement en grappe permet de réduire les temps d'arrêt en éliminant les étapes de démontage et de montage qui doivent être effectuées lors d'un scénario typique de basculement impliquant le déplacement d'un service à haute disponibilité.

Red Hat vous recommande d'utiliser des systèmes de fichiers réseau à moins que vous ne disposiez d'un cas d'utilisation spécifique pour les systèmes de fichiers de stockage partagé. Utilisez les systèmes de fichiers de stockage partagé principalement pour les déploiements qui doivent fournir des services de haute disponibilité avec un temps d'arrêt minimal et qui ont des exigences strictes en matière de niveau de service.

1.10. SYSTÈMES DE FICHIERS GÉRANT LES VOLUMES

Les systèmes de fichiers à gestion de volume intègrent l'ensemble de la pile de stockage à des fins de simplicité et d'optimisation de la pile.

Systèmes de fichiers disponibles pour la gestion des volumes

- Red Hat Enterprise Linux 9 fournit le gestionnaire de volume Stratis en tant qu'aperçu technologique. Stratis utilise XFS comme couche de système de fichiers et l'intègre avec LVM, Device Mapper et d'autres composants.

Stratis a été publié pour la première fois dans Red Hat Enterprise Linux 8.0. Il a été conçu pour combler le vide créé par l'abandon de Btrfs par Red Hat. Stratis 1.0 est un gestionnaire de volume intuitif, basé sur une ligne de commande, qui peut effectuer des opérations de gestion de stockage importantes tout en dissimulant la complexité à l'utilisateur :

- Gestion des volumes
- Création d'une piscine
- Pools de stockage légers
- Instantanés
- Cache de lecture automatisé

Stratis offre des fonctionnalités puissantes, mais il lui manque actuellement certaines capacités d'autres offres auxquelles il pourrait être comparé, telles que Btrfs ou ZFS. En particulier, il ne prend pas en charge les CRC avec autoguérison.

CHAPITRE 2. GESTION DU STOCKAGE LOCAL À L'AIDE DES RÔLES SYSTÈME RHEL

Pour gérer LVM et les systèmes de fichiers locaux (FS) à l'aide d'Ansible, vous pouvez utiliser le rôle **storage**, qui est l'un des rôles système RHEL disponibles dans RHEL 9.

L'utilisation du rôle **storage** vous permet d'automatiser l'administration des systèmes de fichiers sur les disques et les volumes logiques sur plusieurs machines et sur toutes les versions de RHEL à partir de RHEL 7.7.

Pour plus d'informations sur les rôles système RHEL et leur application, voir [Introduction aux rôles système RHEL](#).

2.1. INTRODUCTION AU RÔLE DU SYSTÈME RHEL STORAGE

Le rôle de **storage** peut être géré :

- Systèmes de fichiers sur des disques qui n'ont pas été partitionnés
- Groupes de volumes LVM complets, y compris leurs volumes logiques et leurs systèmes de fichiers
- Volumes RAID MD et leurs systèmes de fichiers

Le rôle **storage** vous permet d'effectuer les tâches suivantes :

- Créer un système de fichiers
- Supprimer un système de fichiers
- Monter un système de fichiers
- Démonter un système de fichiers
- Créer des groupes de volumes LVM
- Supprimer les groupes de volumes LVM
- Créer des volumes logiques
- Supprimer des volumes logiques
- Créer des volumes RAID
- Supprimer des volumes RAID
- Créer des groupes de volumes LVM avec RAID
- Supprimer les groupes de volumes LVM avec RAID
- Créer des groupes de volumes LVM cryptés
- Créer des volumes logiques LVM avec RAID

2.2. PARAMÈTRES QUI IDENTIFIENT UN PÉRIPHÉRIQUE DE STOCKAGE DANS LE RÔLE DE SYSTÈME RHEL STORAGE

Votre configuration du rôle **storage** n'affecte que les systèmes de fichiers, les volumes et les pools que vous avez répertoriés dans les variables suivantes.

storage_volumes

Liste des systèmes de fichiers sur tous les disques non partitionnés à gérer.

storage_volumes peut également inclure des volumes **raid**.

Les partitions ne sont actuellement pas prises en charge.

storage_pools

Liste des pools à gérer.

Actuellement, le seul type de pool pris en charge est LVM. Avec LVM, les pools représentent des groupes de volumes (VG). Sous chaque pool se trouve une liste de volumes à gérer par le rôle. Avec LVM, chaque volume correspond à un volume logique (LV) avec un système de fichiers.

2.3. EXEMPLE DE SCRIPT ANSIBLE POUR CRÉER UN SYSTÈME DE FICHIERS XFS SUR UN PÉRIPHÉRIQUE BLOC

Cette section fournit un exemple de script Ansible. Ce playbook applique le rôle **storage** pour créer un système de fichiers XFS sur un périphérique bloc à l'aide des paramètres par défaut.



AVERTISSEMENT

Le rôle **storage** peut créer un système de fichiers uniquement sur un disque entier non partitionné ou sur un volume logique (LV). Il ne peut pas créer le système de fichiers sur une partition.

Exemple 2.1. Un playbook qui crée XFS sur /dev/sdb

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

- Le nom du volume (**barefs** dans l'exemple) est actuellement arbitraire. Le rôle **storage** identifie le volume par l'unité de disque répertoriée sous l'attribut **disks**.

- Vous pouvez omettre la ligne **fs_type: xfs** car XFS est le système de fichiers par défaut dans RHEL 9.
- Pour créer le système de fichiers sur un LV, fournissez la configuration LVM sous l'attribut **disks:**, y compris le groupe de volumes qui l'entoure. Pour plus de détails, voir [Exemple Ansible playbook to manage logical volumes](#).
Ne pas fournir le chemin d'accès au dispositif LV.

Ressources supplémentaires

- Le fichier `/usr/share/ansible/roles/rhel-system-roles.storage/README.md`.

2.4. EXEMPLE DE PLAYBOOK ANSIBLE POUR MONTER UN SYSTÈME DE FICHIERS DE MANIÈRE PERSISTANTE

Cette section fournit un exemple de plan de jeu Ansible. Ce playbook applique le rôle **storage** pour monter immédiatement et de manière persistante un système de fichiers XFS.

Exemple 2.2. Un playbook qui monte un système de fichiers sur `/dev/sdb` vers `/mnt/data`

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

- Cette procédure ajoute le système de fichiers au fichier `/etc/fstab` et monte immédiatement le système de fichiers.
- Si le système de fichiers sur le périphérique `/dev/sdb` ou le répertoire du point de montage n'existent pas, la séquence les crée.

Ressources supplémentaires

- Le fichier `/usr/share/ansible/roles/rhel-system-roles.storage/README.md`.

2.5. EXEMPLE DE SCRIPT ANSIBLE POUR LA GESTION DES VOLUMES LOGIQUES

Cette section fournit un exemple de manuel de jeu Ansible. Ce playbook applique le rôle **storage** pour créer un volume logique LVM dans un groupe de volumes.

Exemple 2.3. Un playbook qui crée un volume logique `mylv` dans le groupe de volumes `myvg`

■

```

- hosts: all
  vars:
    storage_pools:
      - name: myvg
        disks:
          - sda
          - sdb
          - sdc
        volumes:
          - name: mylv
            size: 2G
            fs_type: ext4
            mount_point: /mnt/data
  roles:
    - rhel-system-roles.storage

```

- Le groupe de volumes **myvg** se compose des disques suivants :
 - **/dev/sda**
 - **/dev/sdb**
 - **/dev/sdc**
- Si le groupe de volumes **myvg** existe déjà, la procédure ajoute le volume logique au groupe de volumes.
- Si le groupe de volumes **myvg** n'existe pas, le playbook le crée.
- La procédure crée un système de fichiers Ext4 sur le volume logique **mylv** et monte de manière persistante le système de fichiers à l'adresse **/mnt**.

Ressources supplémentaires

- Le fichier `/usr/share/ansible/roles/rhel-system-roles.storage/README.md`.

2.6. EXEMPLE DE SCRIPT ANSIBLE POUR ACTIVER L'ÉLIMINATION DES BLOCS EN LIGNE

Cette section fournit un exemple de manuel de jeu Ansible. Ce playbook applique le rôle **storage** pour monter un système de fichiers XFS avec l'option d'élimination des blocs en ligne activée.

Exemple 2.4. Un playbook qui active l'élimination des blocs en ligne sur `/mnt/data/`

```

---
- hosts: all
  vars:
    storage_volumes:
      - name: barefs
        type: disk
        disks:
          - sdb
        fs_type: xfs
        mount_point: /mnt/data

```

```

mount_options: discard
roles:
  - rhel-system-roles.storage

```

Ressources supplémentaires

- [Exemple de playbook Ansible pour monter un système de fichiers de manière persistante](#)
- Le fichier `/usr/share/ansible/roles/rhel-system-roles.storage/README.md`.

2.7. EXEMPLE DE SCRIPT ANSIBLE POUR CRÉER ET MONTER UN SYSTÈME DE FICHIERS EXT4

Cette section fournit un exemple de script Ansible. Ce playbook applique le rôle **storage** pour créer et monter un système de fichiers Ext4.

Exemple 2.5. Un playbook qui crée Ext4 sur `/dev/sdb` et le monte dans `/mnt/data`

```

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

```

- Le playbook crée le système de fichiers sur le disque `/dev/sdb`.
- Le playbook monte de manière persistante le système de fichiers dans le répertoire `/mnt/data` répertoire.
- L'étiquette du système de fichiers est ***label-name***.

Ressources supplémentaires

- Le fichier `/usr/share/ansible/roles/rhel-system-roles.storage/README.md`.

2.8. EXEMPLE DE SCRIPT ANSIBLE POUR CRÉER ET MONTER UN SYSTÈME DE FICHIERS EXT3

Cette section fournit un exemple de script Ansible. Ce playbook applique le rôle **storage** pour créer et monter un système de fichiers Ext3.

Exemple 2.6. Un playbook qui crée Ext3 sur `/dev/sdb` et le monte à l'adresse `/mnt/data`

```

-
```

```

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

```

- Le playbook crée le système de fichiers sur le disque **/dev/sdb**.
- Le playbook monte de manière persistante le système de fichiers dans le répertoire **/mnt/data** répertoire.
- L'étiquette du système de fichiers est **label-name**.

Ressources supplémentaires

- Le fichier **/usr/share/ansible/roles/rhel-system-roles.storage/README.md**.

2.9. EXEMPLE DE PLAYBOOK ANSIBLE POUR REDIMENSIONNER UN SYSTÈME DE FICHIERS EXT4 OU EXT3 EXISTANT À L'AIDE DU RÔLE DE SYSTÈME RHEL STORAGE

Cette section fournit un exemple de plan de jeu Ansible. Ce playbook applique le rôle **storage** pour redimensionner un système de fichiers Ext4 ou Ext3 existant sur un périphérique bloc.

Exemple 2.7. Un playbook qui configure un seul volume sur un disque

```

---
- name: Create a disk device mounted on /opt/barefs
- hosts: all
  vars:
    storage_volumes:
      - name: barefs
        type: disk
        disks:
          - /dev/sdb
        size: 12 GiB
        fs_type: ext4
        mount_point: /opt/barefs
  roles:
    - rhel-system-roles.storage

```

- Si le volume de l'exemple précédent existe déjà, pour redimensionner le volume, vous devez exécuter le même playbook, mais avec une valeur différente pour le paramètre **size**. Par

exemple, vous devez exécuter le même playbook, mais avec une valeur différente pour le paramètre :

Exemple 2.8. Un playbook qui redimensionne ext4 sur /dev/sdb

```
---
- name: Create a disk device mounted on /opt/barefs
- hosts: all
vars:
  storage_volumes:
    - name: barefs
      type: disk
      disks:
        - /dev/sdb
      size: 10 GiB
      fs_type: ext4
      mount_point: /opt/barefs
roles:
  - rhel-system-roles.storage
```

- Le nom du volume (barefs dans l'exemple) est actuellement arbitraire. Le rôle Stockage identifie le volume par l'unité de disque listée dans l'attribut disks .:



NOTE

L'utilisation de l'action **Resizing** dans d'autres systèmes de fichiers peut détruire les données de l'appareil sur lequel vous travaillez.

Ressources supplémentaires

- Le fichier `/usr/share/ansible/roles/rhel-system-roles.storage/README.md`.

2.10. EXEMPLE DE PLAYBOOK ANSIBLE POUR REDIMENSIONNER UN SYSTÈME DE FICHIERS EXISTANT SUR LVM À L'AIDE DU RÔLE SYSTÈME STORAGE RHEL

Cette section fournit un exemple de script Ansible. Ce playbook applique le rôle système **storage** RHEL pour redimensionner un volume logique LVM avec un système de fichiers.



AVERTISSEMENT

L'utilisation de l'action **Resizing** dans d'autres systèmes de fichiers peut détruire les données de l'appareil sur lequel vous travaillez.

Exemple 2.9. Un playbook qui redimensionne les volumes logiques mylv1 et mylv2 existants dans le groupe de volumes myvg

```

---
- hosts: all
  vars:
    storage_pools:
      - name: myvg
        disks:
          - /dev/sda
          - /dev/sdb
          - /dev/sdc
        volumes:
          - name: mylv1
            size: 10 GiB
            fs_type: ext4
            mount_point: /opt/mount1
          - name: mylv2
            size: 50 GiB
            fs_type: ext4
            mount_point: /opt/mount2

- name: Create LVM pool over three disks
  include_role:
    name: rhel-system-roles.storage

```

- Cette procédure redimensionne les systèmes de fichiers existants suivants :
 - Le système de fichiers Ext4 sur le volume **mylv1**, qui est monté sur **/opt/mount1**, est redimensionné à 10 GiB.
 - Le système de fichiers Ext4 sur le volume **mylv2**, qui est monté sur **/opt/mount2**, est redimensionné à 50 GiB.

Ressources supplémentaires

- Le fichier `/usr/share/ansible/roles/rhel-system-roles.storage/README.md`.

2.11. EXEMPLE DE PLAYBOOK ANSIBLE POUR CRÉER UN VOLUME D'ÉCHANGE À L'AIDE DU RÔLE DE SYSTÈME RHEL STORAGE

Cette section fournit un exemple de script Ansible. Ce playbook applique le rôle **storage** pour créer un volume d'échange, s'il n'existe pas, ou pour modifier le volume d'échange, s'il existe déjà, sur un périphérique de bloc en utilisant les paramètres par défaut.

Exemple 2.10. Un playbook qui crée ou modifie un XFS existant sur `/dev/sdb`

```

---
- name: Create a disk device with swap
- hosts: all
  vars:
    storage_volumes:
      - name: swap_fs
        type: disk
        disks:

```

```

- /dev/sdb
size: 15 GiB
fs_type: swap
roles:
- rhel-system-roles.storage

```

- Le nom du volume (**swap_fs** dans l'exemple) est actuellement arbitraire. Le rôle **storage** identifie le volume par l'unité de disque répertoriée sous l'attribut **disks**.

Ressources supplémentaires

- Le fichier `/usr/share/ansible/roles/rhel-system-roles.storage/README.md`.

2.12. CONFIGURATION D'UN VOLUME RAID À L'AIDE DU RÔLE DE SYSTÈME DE STOCKAGE

Avec le rôle de système **storage**, vous pouvez configurer un volume RAID sur RHEL en utilisant Red Hat Ansible Automation Platform et Ansible-Core. Créez un playbook Ansible avec les paramètres pour configurer un volume RAID en fonction de vos besoins.

Conditions préalables

- Le paquetage Ansible Core est installé sur la machine de contrôle.
- Le paquetage **rhel-system-roles** est installé sur le système à partir duquel vous souhaitez exécuter le playbook.
- Vous disposez d'un fichier d'inventaire détaillant les systèmes sur lesquels vous souhaitez déployer un volume RAID à l'aide du rôle de système **storage**.

Procédure

1. Créez un nouveau fichier `playbook.yml` avec le contenu suivant :

```

---
- name: Configure the storage
  hosts: managed-node-01.example.com
  tasks:
  - name: Create a RAID on sdd, sde, sdf, and sdg
    include_role:
      name: rhel-system-roles.storage
  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

```



AVERTISSEMENT

Les noms de périphériques peuvent changer dans certaines circonstances, par exemple lorsque vous ajoutez un nouveau disque à un système. Par conséquent, pour éviter toute perte de données, n'utilisez pas de noms de disques spécifiques dans le guide de lecture.

2. Facultatif : Vérifiez la syntaxe du playbook :

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

3. Exécutez le manuel de jeu :

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

Ressources supplémentaires

- Le fichier `/usr/share/ansible/roles/rhel-system-roles.storage/README.md`
- [Préparation d'un nœud de contrôle et de nœuds gérés à l'utilisation des rôles système RHEL](#)

2.13. CONFIGURATION D'UN POOL LVM AVEC RAID À L'AIDE DU RÔLE SYSTÈME STORAGE RHEL

Avec le rôle de système **storage**, vous pouvez configurer un pool LVM avec RAID sur RHEL à l'aide de Red Hat Ansible Automation Platform. Dans cette section, vous apprendrez à configurer un playbook Ansible avec les paramètres disponibles pour configurer un pool LVM avec RAID.

Conditions préalables

- Le paquetage Ansible Core est installé sur la machine de contrôle.
- Le paquetage **rhel-system-roles** est installé sur le système à partir duquel vous souhaitez exécuter le playbook.
- Vous disposez d'un fichier d'inventaire détaillant les systèmes sur lesquels vous souhaitez configurer un pool LVM avec RAID à l'aide du rôle de système **storage**.

Procédure

1. Créez un nouveau fichier ***playbook.yml*** avec le contenu suivant :

```
- 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

```



NOTE

Pour créer un pool LVM avec RAID, vous devez spécifier le type de RAID à l'aide du paramètre **raid_level**.

2. Facultatif. Vérifier la syntaxe du playbook.

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

3. Exécutez le playbook sur votre fichier d'inventaire :

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

Ressources supplémentaires

- Le fichier `/usr/share/ansible/roles/rhel-system-roles.storage/README.md`.

2.14. EXEMPLE DE PLAYBOOK ANSIBLE POUR COMPRESSER ET DÉDUPLIQUER UN VOLUME VDO SUR LVM À L'AIDE DU RÔLE SYSTÈME STORAGE RHEL

Cette section fournit un exemple de plan de jeu Ansible. Ce playbook applique le rôle système **storage** RHEL pour activer la compression et la déduplication des volumes logiques (LVM) à l'aide de Virtual Data Optimizer (VDO).

Exemple 2.11. Un playbook qui crée un volumemylv1 LVM VDO dans le groupe de volumesmyvg

```

---
- name: Create LVM VDO volume under volume group 'myvg'
  hosts: all
  roles:
    - rhel-system-roles.storage
  vars:
    storage_pools:
      - name: myvg
        disks:
          - /dev/sdb
    volumes:
      - name: mylv1
        compression: true
        deduplication: true

```

```
vdo_pool_size: 10 GiB
size: 30 GiB
mount_point: /mnt/app/shared
```

Dans cet exemple, les pools **compression** et **deduplication** sont réglés sur `true`, ce qui spécifie que le VDO est utilisé. Les paragraphes suivants décrivent l'utilisation de ces paramètres :

- Le site **deduplication** est utilisé pour dédupliquer les données dupliquées stockées sur le volume de stockage.
- La compression est utilisée pour comprimer les données stockées sur le volume de stockage, ce qui permet d'augmenter la capacité de stockage.
- Le paramètre `vdo_pool_size` indique la taille réelle du volume sur le périphérique. La taille virtuelle du volume VDO est définie par le paramètre **size**. NOTE : En raison de l'utilisation du rôle de stockage de LVM VDO, un seul volume par pool peut utiliser la compression et la déduplication.

2.15. CRÉATION D'UN VOLUME CHIFFRÉ LUKS2 À L'AIDE DU RÔLE SYSTÈME STORAGE RHEL

Vous pouvez utiliser le rôle **storage** pour créer et configurer un volume chiffré avec LUKS en exécutant un manuel de jeu Ansible.

Conditions préalables

- Accès et autorisations à un ou plusieurs nœuds gérés, qui sont des systèmes que vous souhaitez configurer avec le rôle de système **crypto_policies**.
- Un fichier d'inventaire, qui répertorie les nœuds gérés.
- Accès et permissions à un nœud de contrôle, qui est un système à partir duquel Red Hat Ansible Core configure d'autres systèmes. Sur le nœud de contrôle, les paquets **ansible-core** et **rhel-system-roles** sont installés.

IMPORTANT

RHEL 8.0-8.5 donne accès à un dépôt Ansible distinct qui contient Ansible Engine 2.9 pour l'automatisation basée sur Ansible. Ansible Engine contient des utilitaires de ligne de commande tels que **ansible**, **ansible-playbook**, des connecteurs tels que **docker** et **podman**, ainsi que de nombreux plugins et modules. Pour plus d'informations sur la manière d'obtenir et d'installer Ansible Engine, consultez l'article de la base de connaissances [Comment télécharger et installer Red Hat Ansible Engine](#) .

RHEL 8.6 et 9.0 ont introduit Ansible Core (fourni en tant que paquetage **ansible-core**), qui contient les utilitaires de ligne de commande Ansible, les commandes et un petit ensemble de plugins Ansible intégrés. RHEL fournit ce paquetage par l'intermédiaire du dépôt AppStream, et sa prise en charge est limitée. Pour plus d'informations, consultez l'article de la base de connaissances intitulé [Scope of support for the Ansible Core package included in the RHEL 9 and RHEL 8.6 and later AppStream repositories \(Portée de la prise en charge du package Ansible Core inclus dans les dépôts AppStream RHEL 9 et RHEL 8.6 et versions ultérieures\)](#) .

Procédure

1. Créez un nouveau fichier ***playbook.yml*** avec le contenu suivant :

```
- 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
```

Vous pouvez également ajouter les autres paramètres de cryptage tels que **encryption_key**, **encryption_cipher**, **encryption_key_size**, et **encryption_luks** version dans le fichier *playbook.yml*.

2. Facultatif : Vérifier la syntaxe du playbook :

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

3. Exécutez le playbook sur votre fichier d'inventaire :

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

Vérification

1. Visualiser l'état du cryptage :

```
# cryptsetup status sdb

/dev/mapper/sdb is active and is in use.
type: LUKS2
cipher: aes-xts-plain64
keysize: 512 bits
key location: keyring
device: /dev/sdb
[...]
```

2. Vérifiez le volume crypté LUKS créé :

```
# cryptsetup luksDump /dev/sdb

Version:      2
Epoch:       6
Metadata area: 16384 [bytes]
Keyslots area: 33521664 [bytes]
UUID:        a4c6be82-7347-4a91-a8ad-9479b72c9426
```

```
Label:      (no label)
Subsystem: (no subsystem)
Flags:     allow-discards
```

```
Data segments:
 0: crypt
offset: 33554432 [bytes]
length: (whole device)
cipher: aes-xts-plain64
sector: 4096 [bytes]
[...]
```

3. Consultez les paramètres **cryptsetup** dans le fichier **playbook.yml** que le rôle **storage** prend en charge :

```
# cat ~/playbook.yml

- hosts: all
  vars:
    storage_volumes:
      - name: foo
        type: disk
        disks:
          - nvme0n1
        fs_type: xfs
        fs_label: label-name
        mount_point: /mnt/data
        encryption: true
        #encryption_password: passwdpasswd
        encryption_key: /home/passwd_key
        encryption_cipher: aes-xts-plain64
        encryption_key_size: 512
        encryption_luks_version: luks2

  roles:
    - rhel-system-roles.storage
```

Ressources supplémentaires

- [Chiffrement des blocs de données à l'aide de LUKS](#)
- `/usr/share/ansible/roles/rhel-system-roles.storage/README.md` fichier

2.16. EXEMPLE DE PLAYBOOK ANSIBLE POUR EXPRIMER LES TAILLES DE VOLUME DE POOL EN POURCENTAGE À L'AIDE DU RÔLE DE SYSTÈME RHEL STORAGE

Cette section fournit un exemple de manuel de jeu Ansible. Ce manuel applique le rôle de système **storage** pour vous permettre d'exprimer la taille des volumes LVM (Logical Manager Volumes) en pourcentage de la taille totale du pool.

Exemple 2.12. Un playbook qui exprime la taille des volumes en pourcentage de la taille totale du pool

■

```
---
- name: Express volume sizes as a percentage of the pool's total size
  hosts: all
  roles
  - rhel-system-roles.storage
  vars:
    storage_pools:
    - name: myvg
      disks:
      - /dev/sdb
      volumes:
      - name: data
        size: 60%
        mount_point: /opt/mount/data
      - name: web
        size: 30%
        mount_point: /opt/mount/web
      - name: cache
        size: 10%
        mount_point: /opt/cache/mount
```

Cet exemple spécifie la taille des volumes LVM en pourcentage de la taille du pool, par exemple : "60%". En outre, vous pouvez également spécifier la taille des volumes LVM en pourcentage de la taille du pool dans une taille lisible par l'homme du système de fichiers, par exemple : "10g" ou "50 GiB".

2.17. RESSOURCES SUPPLÉMENTAIRES

- [/usr/share/doc/rhel-system-roles/storage/](#)
- [/usr/share/ansible/roles/rhel-system-roles.storage/](#)

CHAPITRE 3. MONTAGE DES PARTAGES NFS

En tant qu'administrateur système, vous pouvez monter des partages NFS distants sur votre système pour accéder à des données partagées.

3.1. INTRODUCTION À NFS

Cette section explique les concepts de base du service NFS.

Un système de fichiers réseau (NFS) permet aux hôtes distants de monter des systèmes de fichiers sur un réseau et d'interagir avec ces systèmes de fichiers comme s'ils étaient montés localement. Cela vous permet de consolider les ressources sur des serveurs centralisés sur le réseau.

Le serveur NFS se réfère au fichier de configuration **/etc/exports** pour déterminer si le client est autorisé à accéder aux systèmes de fichiers exportés. Une fois cette vérification effectuée, toutes les opérations sur les fichiers et les répertoires sont accessibles à l'utilisateur.

3.2. VERSIONS NFS PRISES EN CHARGE

Cette section répertorie les versions de NFS prises en charge par Red Hat Enterprise Linux et leurs fonctionnalités.

Actuellement, Red Hat Enterprise Linux 9 prend en charge les versions majeures suivantes de NFS :

- La version 3 de NFS (NFSv3) prend en charge les écritures asynchrones sûres et est plus robuste que la version précédente NFSv2 en ce qui concerne la gestion des erreurs. Elle prend également en charge les tailles de fichiers et les décalages de 64 bits, ce qui permet aux clients d'accéder à plus de 2 Go de données de fichiers.
- NFS version 4 (NFSv4) fonctionne à travers les pare-feu et sur Internet, ne nécessite plus de service **rpcbind**, prend en charge les listes de contrôle d'accès (ACL) et utilise des opérations avec état.

NFS version 2 (NFSv2) n'est plus pris en charge par Red Hat.

Version NFS par défaut

La version par défaut de NFS dans Red Hat Enterprise Linux 9 est 4.2. Les clients NFS tentent de monter en utilisant NFSv4.2 par défaut, et reviennent à NFSv4.1 lorsque le serveur ne prend pas en charge NFSv4.2. Le montage revient ensuite à NFSv4.0, puis à NFSv3.

Caractéristiques des versions mineures de NFS

Voici les caractéristiques de NFSv4.2 dans Red Hat Enterprise Linux 9 :

Copie côté serveur

Permet au client NFS de copier efficacement des données sans gaspiller les ressources du réseau à l'aide de l'appel système **copy_file_range()**.

Fichiers épars

Permet aux fichiers d'avoir un ou plusieurs *holes*, qui sont des blocs de données non alloués ou non initialisés composés uniquement de zéros. L'opération **lseek()** dans NFSv4.2 prend en charge **seek_hole()** et **seek_data()**, ce qui permet aux applications de déterminer l'emplacement des trous dans le fichier clairsemé.

Réservation d'espace

Permet aux serveurs de stockage de réserver de l'espace libre, ce qui empêche les serveurs de

manquer d'espace. NFSv4.2 prend en charge l'opération **allocate()** pour réserver de l'espace, l'opération **deallocate()** pour libérer de l'espace et l'opération **fallocate()** pour pré-allouer ou désallouer de l'espace dans un fichier.

NFS étiqueté

Renforce les droits d'accès aux données et active les étiquettes SELinux entre un client et un serveur pour des fichiers individuels sur un système de fichiers NFS.

Amélioration de la mise en page

Fournit l'opération **layoutstats()**, qui permet à certains serveurs NFS parallèles (pNFS) de collecter de meilleures statistiques de performance.

Voici les caractéristiques de NFSv4.1 :

- Améliore les performances et la sécurité du réseau, et inclut également la prise en charge côté client de pNFS.
- Ne nécessite plus de connexion TCP distincte pour les rappels, ce qui permet à un serveur NFS d'accorder des délégations même lorsqu'il ne peut pas contacter le client : par exemple, en cas d'interférence due à la NAT ou à un pare-feu.
- La sémantique "exactement une fois" (sauf pour les opérations de redémarrage) permet d'éviter un problème antérieur dans lequel certaines opérations renvoyaient parfois un résultat inexact si une réponse était perdue et que l'opération était envoyée deux fois.

3.3. SERVICES REQUIS PAR LES SNF

Cette section répertorie les services système qui sont nécessaires à l'exécution d'un serveur NFS ou au montage de partages NFS. Red Hat Enterprise Linux démarre ces services automatiquement.

Red Hat Enterprise Linux utilise une combinaison de support au niveau du noyau et de processus de service pour fournir le partage de fichiers NFS. Toutes les versions de NFS reposent sur des appels de procédure à distance (RPC) entre les clients et les serveurs. Pour partager ou monter des systèmes de fichiers NFS, les services suivants fonctionnent ensemble en fonction de la version de NFS mise en œuvre :

nfsd

Module du noyau du serveur NFS qui traite les demandes de systèmes de fichiers NFS partagés.

rpcbind

Accepte les réservations de ports des services RPC locaux. Ces ports sont ensuite mis à disposition (ou annoncés) pour que les services RPC distants correspondants puissent y accéder. Le service **rpcbind** répond aux demandes de services RPC et établit des connexions avec le service RPC demandé. Ce service n'est pas utilisé avec NFSv4.

rpc.mountd

Ce processus est utilisé par un serveur NFS pour traiter les demandes **MOUNT** des clients NFSv3. Il vérifie que le partage NFS demandé est actuellement exporté par le serveur NFS et que le client est autorisé à y accéder. Si la demande de montage est autorisée, le service **nfs-mountd** répond avec un statut "Success" et fournit au client NFS le File-Handle pour ce partage NFS.

rpc.nfsd

Ce processus permet de définir les versions et les protocoles NFS explicitement annoncés par le serveur. Il travaille avec le noyau Linux pour répondre aux demandes dynamiques des clients NFS, par exemple en fournissant des threads de serveur à chaque fois qu'un client NFS se connecte. Ce processus correspond au service **nfs-server**.

lockd

Il s'agit d'un thread du noyau qui s'exécute à la fois sur les clients et les serveurs. Il implémente le protocole Network Lock Manager (NLM), qui permet aux clients NFSv3 de verrouiller des fichiers sur le serveur. Il est démarré automatiquement à chaque fois que le serveur NFS est exécuté et qu'un système de fichiers NFS est monté.

rpc.statd

Ce processus met en œuvre le protocole RPC Network Status Monitor (NSM), qui notifie les clients NFS lorsqu'un serveur NFS est redémarré sans avoir été arrêté de manière gracieuse. Le service **rpc-statd** est démarré automatiquement par le service **nfs-server** et ne nécessite pas de configuration de la part de l'utilisateur. Il n'est pas utilisé avec NFSv4.

rpc.rquotad

Ce processus fournit des informations sur les quotas d'utilisateurs pour les utilisateurs distants. Le service **rpc-rquotad**, qui est fourni par le paquetage **quota-rpc**, doit être démarré par l'utilisateur lorsque le paquetage **nfs-server** est démarré.

rpc.idmapd

Ce processus fournit des appels ascendants aux clients et aux serveurs NFSv4, qui établissent une correspondance entre les noms NFSv4 sur le fil (chaînes de caractères sous la forme de **user@domain**) et les UID et GID locaux. Pour que **idmapd** fonctionne avec NFSv4, le fichier **/etc/idmapd.conf** doit être configuré. Il faut au moins spécifier le paramètre **Domain**, qui définit le domaine de mappage NFSv4. Si le domaine de mappage NFSv4 est identique au nom de domaine DNS, ce paramètre peut être ignoré. Le client et le serveur doivent être d'accord sur le domaine de mappage NFSv4 pour que le mappage d'ID fonctionne correctement.

Seul le serveur NFSv4 utilise **rpc.idmapd**, qui est lancé par le service **nfs-idmapd**. Le client NFSv4 utilise l'utilitaire **nfsidmap**, basé sur un trousseau de clés, qui est appelé par le noyau à la demande pour effectuer le mappage des identifiants. En cas de problème avec **nfsidmap**, le client revient à l'utilisation de **rpc.idmapd**.

Les services RPC avec NFSv4

Les protocoles de montage et de verrouillage ont été incorporés dans le protocole NFSv4. Le serveur écoute également sur le port TCP 2049 bien connu. Ainsi, NFSv4 n'a pas besoin d'interagir avec les services **rpcbind**, **lockd** et **rpc-statd**. Le service **nfs-mountd** est toujours nécessaire sur le serveur NFS pour configurer les exportations, mais il n'est impliqué dans aucune opération sur le fil.

Ressources supplémentaires

- [Configuration d'un serveur NFSv4 uniquement sans **rpcbind**](#).

3.4. FORMATS DES NOMS D'HÔTES NFS

Cette section décrit les différents formats que vous pouvez utiliser pour spécifier un hôte lors du montage ou de l'exportation d'un partage NFS.

Vous pouvez spécifier l'hôte dans les formats suivants :

Machine unique

L'un ou l'autre des éléments suivants :

- Un nom de domaine complet (qui peut être résolu par le serveur)
- Nom d'hôte (qui peut être résolu par le serveur)
- Une adresse IP.

Réseaux IP

L'un ou l'autre des formats suivants est valable :

- ***a.b.c.d/z*** où ***a.b.c.d*** est le réseau et ***z*** est le nombre de bits du masque de réseau ; par exemple **192.168.0.0/24**.
- ***a.b.c.d/netmask*** où ***a.b.c.d*** est le réseau et ***netmask*** est le masque de réseau ; par exemple, **192.168.100.8/255.255.255.0**.

Netgroups

Le format **@*group-name*** où ***group-name*** est le nom du groupe net NIS.

3.5. INSTALLATION DE NFS

Cette procédure installe tous les paquets nécessaires pour monter ou exporter des partages NFS.

Procédure

- Installez le paquetage **nfs-utils**:

```
# dnf install nfs-utils
```

3.6. DÉCOUVRIR LES EXPORTATIONS NFS

Cette procédure permet de découvrir les systèmes de fichiers exportés par un serveur NFSv3 ou NFSv4 donné.

Procédure

- Avec n'importe quel serveur prenant en charge NFSv3, utilisez l'utilitaire **showmount**:

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

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

- Avec n'importe quel serveur supportant NFSv4, montez le répertoire racine et regardez autour de vous :

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

```
exports
```

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

```
foo
bar
```

Sur les serveurs qui supportent à la fois NFSv4 et NFSv3, les deux méthodes fonctionnent et donnent les mêmes résultats.

Ressources supplémentaires

- **showmount(8)** page de manuel

3.7. MONTAGE D'UN PARTAGE NFS AVEC MOUNT

Monter un partage NFS exporté d'un serveur à l'aide de l'utilitaire **mount**.



AVERTISSEMENT

Vous pouvez rencontrer des conflits dans votre NFSv4 **clientid** et leur expiration soudaine si vos clients NFS ont le même nom d'hôte court. Pour éviter toute expiration soudaine de votre NFSv4 **clientid**, vous devez utiliser des noms d'hôte uniques pour les clients NFS ou configurer l'identifiant sur chaque conteneur, en fonction du système que vous utilisez. Pour plus d'informations, consultez l'article de la base de connaissances [NFSv4 clientid was expired suddenly due to use same hostname on several NFS clients](#).

Procédure

- Pour monter un partage NFS, utilisez la commande suivante :

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

Cette commande utilise les variables suivantes :

options

Liste d'options de montage délimitée par des virgules.

host

Le nom d'hôte, l'adresse IP ou le nom de domaine complet du serveur exportant le système de fichiers que vous souhaitez monter.

/remote/export

Le système de fichiers ou le répertoire exporté du serveur, c'est-à-dire le répertoire que vous souhaitez monter.

/local/directory

L'emplacement du client où **/remote/export** est monté.

Ressources supplémentaires

- [Options de montage NFS courantes](#).
- [Formats des noms d'hôtes NFS](#).
- [Montage d'un système de fichiers avec mount](#).
- **mount(8)** page de manuel
- **exports(5)** page de manuel

3.8. OPTIONS DE MONTAGE NFS COURANTES

Les options suivantes sont couramment utilisées lors du montage de partages NFS. Vous pouvez utiliser ces options avec les commandes manuelles **mount**, les paramètres **/etc/fstab** et **autofs**.

lookupcache=mode

Spécifie comment le noyau doit gérer son cache d'entrées de répertoire pour un point de montage donné. Les arguments valides pour *mode* sont **all**, **none**, ou **positive**.

nfsvers=version

Spécifie la version du protocole NFS à utiliser, où *version* est **3**, **4**, **4.0**, **4.1**, ou **4.2**. Ceci est utile pour les hôtes qui exécutent plusieurs serveurs NFS, ou pour désactiver les tentatives de montage avec des versions inférieures. Si aucune version n'est spécifiée, NFS utilise la version la plus élevée prise en charge par le noyau et l'utilitaire **mount**.

L'option **vers** est identique à **nfsvers** et est incluse dans cette version pour des raisons de compatibilité.

noacl

Désactive tous les traitements ACL. Cela peut s'avérer nécessaire lors de l'interface avec des versions plus anciennes de Red Hat Enterprise Linux, Red Hat Linux ou Solaris, car la technologie ACL la plus récente n'est pas compatible avec les systèmes plus anciens.

noexec

Désactive le verrouillage des fichiers. Ce paramètre est parfois nécessaire lors de la connexion à de très vieux serveurs NFS.

noexec

Empêche l'exécution des binaires sur les systèmes de fichiers montés. Ceci est utile si le système monte un système de fichiers non-Linux contenant des binaires incompatibles.

nosuid

Désactive les bits **set-user-identifiant** et **set-group-identifiant**. Cela empêche les utilisateurs distants d'obtenir des privilèges plus élevés en exécutant un programme **setuid**.

port=num

Spécifie la valeur numérique du port du serveur NFS. Si *num* est **0** (valeur par défaut), **mount** demande au service **rpcbind** de l'hôte distant le numéro de port à utiliser. Si le service NFS de l'hôte distant n'est pas enregistré auprès de son service **rpcbind**, le numéro de port NFS standard (TCP 2049) est utilisé à la place.

rsize=num et wsize=num

Ces options définissent le nombre maximal d'octets à transférer lors d'une seule opération de lecture ou d'écriture NFS.

Il n'y a pas de valeur fixe par défaut pour **rsize** et **wsize**. Par défaut, NFS utilise la plus grande valeur possible prise en charge par le serveur et le client. Dans Red Hat Enterprise Linux 9, la valeur maximale pour le client et le serveur est de 1 048 576 octets. Pour plus de détails, consultez l'article [Quelles sont les valeurs par défaut et maximales pour rsize et wsize avec les montages NFS ?](#) KBase.

sec=flavors

Saveurs de sécurité à utiliser pour accéder aux fichiers de l'exportation montée. La valeur *flavors* est une liste séparée par deux points d'une ou plusieurs saveurs de sécurité.

Par défaut, le client tente de trouver une variante de sécurité prise en charge à la fois par le client et le serveur. Si le serveur ne prend en charge aucune des saveurs sélectionnées, l'opération de montage échoue.

Saveurs disponibles :

- **sec=sys** utilise les UID et GID UNIX locaux. Ceux-ci utilisent **AUTH_SYS** pour authentifier les opérations NFS.
- **sec=krb5** utilise Kerberos V5 au lieu des UID et GID UNIX locaux pour authentifier les utilisateurs.
- **sec=krb5i** utilise Kerberos V5 pour l'authentification des utilisateurs et effectue un contrôle d'intégrité des opérations NFS à l'aide de sommes de contrôle sécurisées afin d'empêcher la falsification des données.
- **sec=krb5p** utilise Kerberos V5 pour l'authentification des utilisateurs, le contrôle d'intégrité et le chiffrement du trafic NFS afin d'empêcher le reniflage du trafic. Il s'agit du paramètre le plus sûr, mais c'est aussi celui qui entraîne le plus de surcoût en termes de performances.

tcp

Indique au montage NFS d'utiliser le protocole TCP.

Ressources supplémentaires

- **mount(8)** page de manuel
- **nfs(5)** page de manuel

3.9. RESSOURCES SUPPLÉMENTAIRES

- [Le wiki Linux NFS](#)
- [Montage persistant des partages NFS.](#)
- [Montage de partages NFS à la demande .](#)

CHAPITRE 4. EXPORTATION DE PARTAGES NFS

En tant qu'administrateur système, vous pouvez utiliser le serveur NFS pour partager un répertoire sur votre système via le réseau.

4.1. INTRODUCTION À NFS

Cette section explique les concepts de base du service NFS.

Un système de fichiers réseau (NFS) permet aux hôtes distants de monter des systèmes de fichiers sur un réseau et d'interagir avec ces systèmes de fichiers comme s'ils étaient montés localement. Cela vous permet de consolider les ressources sur des serveurs centralisés sur le réseau.

Le serveur NFS se réfère au fichier de configuration **/etc/exports** pour déterminer si le client est autorisé à accéder aux systèmes de fichiers exportés. Une fois cette vérification effectuée, toutes les opérations sur les fichiers et les répertoires sont accessibles à l'utilisateur.

4.2. VERSIONS NFS PRISES EN CHARGE

Cette section répertorie les versions de NFS prises en charge par Red Hat Enterprise Linux et leurs fonctionnalités.

Actuellement, Red Hat Enterprise Linux 9 prend en charge les versions majeures suivantes de NFS :

- La version 3 de NFS (NFSv3) prend en charge les écritures asynchrones sûres et est plus robuste que la version précédente NFSv2 en ce qui concerne la gestion des erreurs. Elle prend également en charge les tailles de fichiers et les décalages de 64 bits, ce qui permet aux clients d'accéder à plus de 2 Go de données de fichiers.
- NFS version 4 (NFSv4) fonctionne à travers les pare-feu et sur Internet, ne nécessite plus de service **rpcbind**, prend en charge les listes de contrôle d'accès (ACL) et utilise des opérations avec état.

NFS version 2 (NFSv2) n'est plus pris en charge par Red Hat.

Version NFS par défaut

La version par défaut de NFS dans Red Hat Enterprise Linux 9 est 4.2. Les clients NFS tentent de monter en utilisant NFSv4.2 par défaut, et reviennent à NFSv4.1 lorsque le serveur ne prend pas en charge NFSv4.2. Le montage revient ensuite à NFSv4.0, puis à NFSv3.

Caractéristiques des versions mineures de NFS

Voici les caractéristiques de NFSv4.2 dans Red Hat Enterprise Linux 9 :

Copie côté serveur

Permet au client NFS de copier efficacement des données sans gaspiller les ressources du réseau à l'aide de l'appel système **copy_file_range()**.

Fichiers épars

Permet aux fichiers d'avoir un ou plusieurs *holes*, qui sont des blocs de données non alloués ou non initialisés composés uniquement de zéros. L'opération **lseek()** dans NFSv4.2 prend en charge **seek_hole()** et **seek_data()**, ce qui permet aux applications de déterminer l'emplacement des trous dans le fichier clairsemé.

Réservation d'espace

Permet aux serveurs de stockage de réserver de l'espace libre, ce qui empêche les serveurs de

manquer d'espace. NFSv4.2 prend en charge l'opération **allocate()** pour réserver de l'espace, l'opération **deallocate()** pour libérer de l'espace et l'opération **fallocate()** pour pré-allouer ou désallouer de l'espace dans un fichier.

NFS étiqueté

Renforce les droits d'accès aux données et active les étiquettes SELinux entre un client et un serveur pour des fichiers individuels sur un système de fichiers NFS.

Amélioration de la mise en page

Fournit l'opération **layoutstats()**, qui permet à certains serveurs NFS parallèles (pNFS) de collecter de meilleures statistiques de performance.

Voici les caractéristiques de NFSv4.1 :

- Améliore les performances et la sécurité du réseau, et inclut également la prise en charge côté client de pNFS.
- Ne nécessite plus de connexion TCP distincte pour les rappels, ce qui permet à un serveur NFS d'accorder des délégations même lorsqu'il ne peut pas contacter le client : par exemple, en cas d'interférence due à la NAT ou à un pare-feu.
- La sémantique "exactement une fois" (sauf pour les opérations de redémarrage) permet d'éviter un problème antérieur dans lequel certaines opérations renvoyaient parfois un résultat inexact si une réponse était perdue et que l'opération était envoyée deux fois.

4.3. LES PROTOCOLES TCP ET UDP DANS NFSV3 ET NFSV4

NFSv4 nécessite l'utilisation du protocole de contrôle de transmission (TCP) sur un réseau IP.

NFSv3 pouvait également utiliser le protocole User Datagram Protocol (UDP) dans les versions antérieures de Red Hat Enterprise Linux. Dans Red Hat Enterprise Linux 9, NFS via UDP n'est plus pris en charge. Par défaut, UDP est désactivé dans le serveur NFS.

4.4. SERVICES REQUIS PAR LES SNF

Cette section répertorie les services système qui sont nécessaires à l'exécution d'un serveur NFS ou au montage de partages NFS. Red Hat Enterprise Linux démarre ces services automatiquement.

Red Hat Enterprise Linux utilise une combinaison de support au niveau du noyau et de processus de service pour fournir le partage de fichiers NFS. Toutes les versions de NFS reposent sur des appels de procédure à distance (RPC) entre les clients et les serveurs. Pour partager ou monter des systèmes de fichiers NFS, les services suivants fonctionnent ensemble en fonction de la version de NFS mise en œuvre :

nfsd

Module du noyau du serveur NFS qui traite les demandes de systèmes de fichiers NFS partagés.

rpcbind

Accepte les réservations de ports des services RPC locaux. Ces ports sont ensuite mis à disposition (ou annoncés) pour que les services RPC distants correspondants puissent y accéder. Le service **rpcbind** répond aux demandes de services RPC et établit des connexions avec le service RPC demandé. Ce service n'est pas utilisé avec NFSv4.

rpc.mountd

Ce processus est utilisé par un serveur NFS pour traiter les demandes **MOUNT** des clients NFSv3. Il vérifie que le partage NFS demandé est actuellement exporté par le serveur NFS et que le client est

autorisé à y accéder. Si la demande de montage est autorisée, le service **nfs-mountd** répond avec un statut "Success" et fournit au client NFS le File-Handle pour ce partage NFS.

rpc.nfsd

Ce processus permet de définir les versions et les protocoles NFS explicitement annoncés par le serveur. Il travaille avec le noyau Linux pour répondre aux demandes dynamiques des clients NFS, par exemple en fournissant des threads de serveur à chaque fois qu'un client NFS se connecte. Ce processus correspond au service **nfs-server**.

lockd

Il s'agit d'un thread du noyau qui s'exécute à la fois sur les clients et les serveurs. Il implémente le protocole Network Lock Manager (NLM), qui permet aux clients NFSv3 de verrouiller des fichiers sur le serveur. Il est démarré automatiquement à chaque fois que le serveur NFS est exécuté et qu'un système de fichiers NFS est monté.

rpc.statd

Ce processus met en œuvre le protocole RPC Network Status Monitor (NSM), qui notifie les clients NFS lorsqu'un serveur NFS est redémarré sans avoir été arrêté de manière gracieuse. Le service **rpc-statd** est démarré automatiquement par le service **nfs-server** et ne nécessite pas de configuration de la part de l'utilisateur. Il n'est pas utilisé avec NFSv4.

rpc.rquotad

Ce processus fournit des informations sur les quotas d'utilisateurs pour les utilisateurs distants. Le service **rpc-rquotad**, qui est fourni par le paquetage **quota-rpc**, doit être démarré par l'utilisateur lorsque le paquetage **nfs-server** est démarré.

rpc.idmapd

Ce processus fournit des appels ascendants aux clients et aux serveurs NFSv4, qui établissent une correspondance entre les noms NFSv4 sur le fil (chaînes de caractères sous la forme de **user@domain**) et les UID et GID locaux. Pour que **idmapd** fonctionne avec NFSv4, le fichier **/etc/idmapd.conf** doit être configuré. Il faut au moins spécifier le paramètre **Domain**, qui définit le domaine de mappage NFSv4. Si le domaine de mappage NFSv4 est identique au nom de domaine DNS, ce paramètre peut être ignoré. Le client et le serveur doivent être d'accord sur le domaine de mappage NFSv4 pour que le mappage d'ID fonctionne correctement.

Seul le serveur NFSv4 utilise **rpc.idmapd**, qui est lancé par le service **nfs-idmapd**. Le client NFSv4 utilise l'utilitaire **nfsidmap**, basé sur un trousseau de clés, qui est appelé par le noyau à la demande pour effectuer le mappage des identifiants. En cas de problème avec **nfsidmap**, le client revient à l'utilisation de **rpc.idmapd**.

Les services RPC avec NFSv4

Les protocoles de montage et de verrouillage ont été incorporés dans le protocole NFSv4. Le serveur écoute également sur le port TCP 2049 bien connu. Ainsi, NFSv4 n'a pas besoin d'interagir avec les services **rpcbind**, **lockd** et **rpc-statd**. Le service **nfs-mountd** est toujours nécessaire sur le serveur NFS pour configurer les exportations, mais il n'est impliqué dans aucune opération sur le fil.

Ressources supplémentaires

- [Configuration d'un serveur NFSv4 uniquement sans **rpcbind**](#).

4.5. FORMATS DES NOMS D'HÔTES NFS

Cette section décrit les différents formats que vous pouvez utiliser pour spécifier un hôte lors du montage ou de l'exportation d'un partage NFS.

Vous pouvez spécifier l'hôte dans les formats suivants :

Machine unique

L'un ou l'autre des éléments suivants :

- Un nom de domaine complet (qui peut être résolu par le serveur)
- Nom d'hôte (qui peut être résolu par le serveur)
- Une adresse IP.

Réseaux IP

L'un ou l'autre des formats suivants est valable :

- ***a.b.c.d/z*** où ***a.b.c.d*** est le réseau et ***z*** est le nombre de bits du masque de réseau ; par exemple **192.168.0.0/24**.
- ***a.b.c.d/netmask*** où ***a.b.c.d*** est le réseau et ***netmask*** est le masque de réseau ; par exemple, **192.168.100.8/255.255.255.0**.

Netgroups

Le format **@*group-name*** où ***group-name*** est le nom du groupe net NIS.

4.6. CONFIGURATION DU SERVEUR NFS

Cette section décrit la syntaxe et les options de deux façons de configurer les exportations sur un serveur NFS :

- Modification manuelle du fichier de configuration **/etc/exports**
- Utilisation de l'utilitaire **exportfs** sur la ligne de commande

4.6.1. Le fichier de configuration /etc/exports

Le fichier **/etc/exports** contrôle les systèmes de fichiers qui sont exportés vers des hôtes distants et spécifie des options. Il suit les règles syntaxiques suivantes :

- Les lignes vides sont ignorées.
- Pour ajouter un commentaire, commencez une ligne par le signe dièse (**#**).
- Vous pouvez entourer les lignes longues d'une barre oblique inverse (****).
- Chaque système de fichiers exporté doit se trouver sur sa propre ligne.
- Toute liste d'hôtes autorisés placée après un système de fichiers exporté doit être séparée par des caractères d'espacement.
- Les options pour chacun des hôtes doivent être placées entre parenthèses directement après l'identifiant de l'hôte, sans espace entre l'hôte et la première parenthèse.

Entrée de l'exportation

Chaque entrée d'un système de fichiers exporté a la structure suivante :

```
export host(options)
```

Il est également possible de spécifier plusieurs hôtes, ainsi que des options spécifiques pour chacun d'entre eux. Pour ce faire, il faut les énumérer sur la même ligne sous la forme d'une liste délimitée par des espaces, chaque nom d'hôte étant suivi de ses options respectives (entre parenthèses), comme dans l'exemple suivant :

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

Dans cette structure :

export

Le répertoire exporté

host

L'hôte ou le réseau vers lequel l'exportation est partagée

options

Les options à utiliser pour l'hôte

Exemple 4.1. Un simple fichier `/etc/exports`

Dans sa forme la plus simple, le fichier `/etc/exports` spécifie uniquement le répertoire exporté et les hôtes autorisés à y accéder :

```
/exported/directory bob.example.com
```

Ici, **bob.example.com** peut monter `/exported/directory/` à partir du serveur NFS. Comme aucune option n'est spécifiée dans cet exemple, NFS utilise les options par défaut.

IMPORTANT

Le format du fichier `/etc/exports` est très précis, notamment en ce qui concerne l'utilisation du caractère espace. N'oubliez pas de toujours séparer les systèmes de fichiers exportés des hôtes et les hôtes les uns des autres par un caractère espace. Cependant, il ne doit pas y avoir d'autres caractères d'espacement dans le fichier, sauf sur les lignes de commentaires.

Par exemple, les deux lignes suivantes n'ont pas la même signification :

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

La première ligne autorise uniquement les utilisateurs de **bob.example.com** à accéder en lecture et en écriture au répertoire `/home`. La deuxième ligne permet aux utilisateurs de **bob.example.com** de monter le répertoire en lecture seule (par défaut), tandis que le reste du monde peut le monter en lecture/écriture.

Options par défaut

Les options par défaut pour une entrée d'exportation sont les suivantes :

ro

Le système de fichiers exporté est en lecture seule. Les hôtes distants ne peuvent pas modifier les données partagées sur le système de fichiers. Pour permettre aux hôtes d'apporter des modifications au système de fichiers (c'est-à-dire de lire et d'écrire), spécifiez l'option `rw`.

sync

Le serveur NFS ne répondra pas aux demandes avant que les modifications apportées par les demandes précédentes ne soient écrites sur le disque. Pour activer les écritures asynchrones, spécifiez l'option **async**.

wdelay

Le serveur NFS retardera l'écriture sur le disque s'il soupçonne qu'une autre demande d'écriture est imminente. Cela peut améliorer les performances car cela réduit le nombre de fois où le disque doit être accédé par des commandes d'écriture distinctes, réduisant ainsi la surcharge d'écriture. Pour désactiver cette fonction, spécifiez l'option **no_wdelay**, qui n'est disponible que si l'option **sync** par défaut est également spécifiée.

root_squash

Cela empêche les utilisateurs root connectés à distance (et non localement) d'avoir les privilèges root ; au lieu de cela, le serveur NFS leur attribue l'ID utilisateur **nobody**. Cela a pour effet de réduire les pouvoirs de l'utilisateur racine distant à ceux de l'utilisateur local le plus bas, empêchant ainsi toute écriture non autorisée sur le serveur distant. Pour désactiver l'écrasement de la racine, spécifiez l'option **no_root_squash**.

Pour écraser tous les utilisateurs distants (y compris root), utilisez l'option **all_squash**. Pour spécifier les ID d'utilisateur et de groupe que le serveur NFS doit attribuer aux utilisateurs distants d'un hôte particulier, utilisez les options **anonuid** et **anongid**, respectivement, comme dans l'exemple ci-dessous :

```
export host(anonuid=uidanongid=gid)
```

Ici, *uid* et *gid* sont respectivement le numéro d'identification de l'utilisateur et le numéro d'identification du groupe. Les options **anonuid** et **anongid** vous permettent de créer un compte d'utilisateur et de groupe spécial pour les utilisateurs NFS distants à partager.

Par défaut, les listes de contrôle d'accès (ACL) sont prises en charge par NFS sous Red Hat Enterprise Linux. Pour désactiver cette fonctionnalité, spécifiez l'option **no_acl** lors de l'exportation du système de fichiers.

Options par défaut et options prioritaires

Chaque valeur par défaut de chaque système de fichiers exporté doit être explicitement remplacée. Par exemple, si l'option **rw** n'est pas spécifiée, le système de fichiers exporté est partagé en lecture seule. Voici un exemple de ligne provenant de **/etc/exports** qui remplace deux options par défaut :

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

Dans cet exemple, **192.168.0.3** peut monter **/another/exported/directory/** en lecture et en écriture, et toutes les écritures sur disque sont asynchrones.

4.6.2. L'utilitaire exportfs

L'utilitaire **exportfs** permet à l'utilisateur root d'exporter ou de non-exporter sélectivement des répertoires sans redémarrer le service NFS. Lorsqu'il dispose des options appropriées, l'utilitaire **exportfs** écrit les systèmes de fichiers exportés sur **/var/lib/nfs/xtab**. Étant donné que le service **nfs-mountd** se réfère au fichier **xtab** lorsqu'il décide des privilèges d'accès à un système de fichiers, les modifications apportées à la liste des systèmes de fichiers exportés prennent effet immédiatement.

Options courantes d'exportfs

Voici une liste des options couramment utilisées pour **exportfs**:

-r

Provoque l'exportation de tous les répertoires répertoriés dans **/etc/exports** en construisant une nouvelle liste d'exportation dans **/var/lib/nfs/etab**. Cette option permet d'actualiser la liste d'exportation en fonction des modifications apportées à **/etc/exports**.

-a

Provoque l'exportation ou la non-exportation de tous les répertoires, en fonction des autres options transmises à **exportfs**. Si aucune autre option n'est spécifiée, **exportfs** exporte tous les systèmes de fichiers spécifiés dans **/etc/exports**.

-o file-systems

Spécifie les répertoires à exporter qui ne sont pas répertoriés dans **/etc/exports**. Remplacer *file-systems* par des systèmes de fichiers supplémentaires à exporter. Ces systèmes de fichiers doivent être formatés de la même manière qu'ils sont spécifiés dans **/etc/exports**. Cette option est souvent utilisée pour tester un système de fichiers exporté avant de l'ajouter définitivement à la liste des systèmes de fichiers exportés.

-i

Ignore **/etc/exports**; seules les options données sur la ligne de commande sont utilisées pour définir les systèmes de fichiers exportés.

-u

Désexporte tous les répertoires partagés. La commande **exportfs -ua** suspend le partage de fichiers NFS tout en maintenant tous les services NFS. Pour réactiver le partage NFS, utilisez la commande **exportfs -r**.

-v

Fonctionnement verbeux, où les systèmes de fichiers exportés ou non exportés sont affichés plus en détail lorsque la commande **exportfs** est exécutée.

Si aucune option n'est transmise à l'utilitaire **exportfs**, celui-ci affiche une liste des systèmes de fichiers actuellement exportés.

Ressources supplémentaires

- [Formats des noms d'hôtes NFS](#).

4.7. NFS ET RPCBIND

Le service **rpcbind** associe les services RPC (Remote Procedure Call) aux ports sur lesquels ils écoutent. Les processus RPC informent **rpcbind** lorsqu'ils démarrent, en enregistrant les ports sur lesquels ils écoutent et les numéros de programme RPC qu'ils s'attendent à servir. Le système client contacte alors **rpcbind** sur le serveur avec un numéro de programme RPC particulier. Le service **rpcbind** redirige le client vers le numéro de port approprié afin qu'il puisse communiquer avec le service demandé.

Le système de fichiers réseau version 3 (NFSv3) nécessite le service **rpcbind**.

Étant donné que les services basés sur RPC s'appuient sur **rpcbind** pour établir toutes les connexions avec les requêtes entrantes des clients, **rpcbind** doit être disponible avant que ces services ne démarrent.

Les règles de contrôle d'accès pour **rpcbind** affectent tous les services basés sur RPC. Il est également possible de spécifier des règles de contrôle d'accès pour chacun des démons RPC NFS.

Ressources supplémentaires

- **rpc.mountd(8)** page de manuel
- **rpc.statd(8)** page de manuel

4.8. INSTALLATION DE NFS

Cette procédure installe tous les paquets nécessaires pour monter ou exporter des partages NFS.

Procédure

- Installez le paquetage **nfs-utils**:

```
# dnf install nfs-utils
```

4.9. DÉMARRAGE DU SERVEUR NFS

Cette procédure décrit comment démarrer le serveur NFS, qui est nécessaire pour exporter des partages NFS.

Conditions préalables

- Pour les serveurs qui prennent en charge les connexions NFSv3, le service **rpcbind** doit être en cours d'exécution. Pour vérifier que **rpcbind** est actif, utilisez la commande suivante :

```
$ systemctl status rpcbind
```

Si le service est arrêté, démarrez-le et activez-le :

```
$ systemctl enable --now rpcbind
```

Procédure

- Pour démarrer le serveur NFS et lui permettre de démarrer automatiquement au démarrage, utilisez la commande suivante :

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

Ressources supplémentaires

- [Configuration d'un serveur NFSv4 uniquement](#) .

4.10. DÉPANNAGE DE NFS ET RPCBIND

Comme le service **rpcbind** assure la coordination entre les services RPC et les numéros de port utilisés pour communiquer avec eux, il est utile d'afficher l'état des services RPC actuels à l'aide de **rpcbind** lors du dépannage. L'utilitaire **rpcinfo** affiche chaque service RPC avec les numéros de port, un numéro de programme RPC, un numéro de version et un type de protocole IP (TCP ou UDP).

Procédure

1. Pour s'assurer que les services NFS RPC appropriés sont activés pour **rpcbind**, utilisez la commande suivante :

```
# rpcinfo -p
```

Exemple 4.2. sortie de la commande rpcinfo -p

Voici un exemple de sortie de cette commande :

```
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
```

Si l'un des services NFS ne démarre pas correctement, **rpcbind** sera incapable de faire correspondre les requêtes RPC des clients de ce service au port correct.

2. Dans de nombreux cas, si NFS n'est pas présent dans la sortie **rpcinfo**, le redémarrage de NFS permet au service de s'enregistrer correctement sur **rpcbind** et de commencer à fonctionner :

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

Ressources supplémentaires

- [Configuration d'un serveur NFSv4 uniquement](#) .

4.11. CONFIGURER LE SERVEUR NFS POUR QU'IL FONCTIONNE DERRIÈRE UN PARE-FEU

NFS nécessite le service **rpcbind**, qui attribue dynamiquement des ports pour les services RPC et peut poser des problèmes pour la configuration des règles de pare-feu. Les sections suivantes décrivent comment configurer les versions de NFS pour qu'elles fonctionnent derrière un pare-feu si vous

souhaitez les prendre en charge :

- NFSv3
 - Il s'agit de tous les serveurs qui prennent en charge NFSv3 :
 - Serveurs NFSv3 uniquement
 - Serveurs supportant à la fois NFSv3 et NFSv4
- NFSv4 uniquement

4.11.1. Configurer le serveur NFSv3 pour qu'il fonctionne derrière un pare-feu

La procédure suivante décrit comment configurer les serveurs qui prennent en charge NFSv3 pour qu'ils fonctionnent derrière un pare-feu. Il s'agit des serveurs NFSv3 uniquement et des serveurs qui prennent en charge à la fois NFSv3 et NFSv4.

Procédure

1. Pour permettre aux clients d'accéder aux partages NFS derrière un pare-feu, configurez ce dernier en exécutant les commandes suivantes sur le serveur NFS :

```
firewall-cmd --permanent --add-service mountd
firewall-cmd --permanent --add-service rpc-bind
firewall-cmd --permanent --add-service nfs
```

2. Spécifiez les ports à utiliser par le service RPC **nlockmgr** dans le fichier **/etc/nfs.conf** comme suit :

```
[lockd]
port=tcp-port-number
udp-port=udp-port-number
```

Vous pouvez également spécifier **nlm_tcpport** et **nlm_udpport** dans le fichier **/etc/modprobe.d/lockd.conf**.

3. Ouvrez les ports spécifiés dans le pare-feu en exécutant les commandes suivantes sur le serveur NFS :

```
firewall-cmd --permanent --add-port=<lockd-tcp-port>/tcp
firewall-cmd --permanent --add-port=<lockd-udp-port>/udp
```

4. Ajoutez des ports statiques pour **rpc.statd** en modifiant la section **[statd]** du fichier **/etc/nfs.conf** comme suit :

```
[statd]
port=port-number
```

5. Ouvrez les ports ajoutés dans le pare-feu en exécutant les commandes suivantes sur le serveur NFS :

```

firewall-cmd --permanent --add-port=<statd-tcp-port>/tcp
firewall-cmd --permanent --add-port=<statd-udp-port>/udp

```

6. Recharger la configuration du pare-feu :

```

firewall-cmd --reload

```

7. Redémarrez d'abord le service **rpc-statd**, puis le service **nfs-server**:

```

# systemctl restart rpc-statd.service
# systemctl restart nfs-server.service

```

Alternativement, si vous avez spécifié les ports **lockd** dans le fichier **/etc/modprobe.d/lockd.conf**:

- a. Mettre à jour les valeurs actuelles de **/proc/sys/fs/nfs/nlm_tcpport** et **/proc/sys/fs/nfs/nlm_udpport**:

```

# sysctl -w fs.nfs.nlm_tcpport=<tcp-port>
# sysctl -w fs.nfs.nlm_udpport=<udp-port>

```

- b. Redémarrez les services **rpc-statd** et **nfs-server**:

```

# systemctl restart rpc-statd.service
# systemctl restart nfs-server.service

```

4.11.2. Configurer le serveur NFSv4-only pour qu'il fonctionne derrière un pare-feu

La procédure suivante décrit comment configurer le serveur NFSv4-only pour qu'il fonctionne derrière un pare-feu.

Procédure

1. Pour permettre aux clients d'accéder aux partages NFS derrière un pare-feu, configurez ce dernier en exécutant la commande suivante sur le serveur NFS :

```

firewall-cmd --permanent --add-service nfs

```

2. Recharger la configuration du pare-feu :

```

firewall-cmd --reload

```

3. Redémarrez le serveur nfs :

```

# systemctl restart nfs-server

```

4.11.3. Configurer un client NFSv3 pour qu'il fonctionne derrière un pare-feu

La procédure de configuration d'un client NFSv3 derrière un pare-feu est similaire à la procédure de configuration d'un serveur NFSv3 derrière un pare-feu.

Si la machine que vous configurez est à la fois un client et un serveur NFS, suivez la procédure décrite dans la section [Configurer le serveur compatible NFSv3 pour qu'il fonctionne derrière un pare-feu](#) .

La procédure suivante décrit comment configurer une machine qui n'est qu'un client NFS pour qu'elle fonctionne derrière un pare-feu.

Procédure

1. Pour permettre au serveur NFS d'effectuer des rappels vers le client NFS lorsque ce dernier se trouve derrière un pare-feu, ajoutez le service **rpc-bind** au pare-feu en exécutant la commande suivante sur le client NFS :

```
firewall-cmd --permanent --add-service rpc-bind
```

2. Spécifiez les ports à utiliser par le service RPC **nlockmgr** dans le fichier **/etc/nfs.conf** comme suit :

```
[lockd]

port=port-number
udp-port=udp-port-number
```

Vous pouvez également spécifier **nlm_tcpport** et **nlm_udpport** dans le fichier **/etc/modprobe.d/lockd.conf**.

3. Ouvrez les ports spécifiés dans le pare-feu en exécutant les commandes suivantes sur le client NFS :

```
firewall-cmd --permanent --add-port=<lockd-tcp-port>/tcp
firewall-cmd --permanent --add-port=<lockd-udp-port>/udp
```

4. Ajoutez des ports statiques pour **rpc.statd** en modifiant la section **[statd]** du fichier **/etc/nfs.conf** comme suit :

```
[statd]

port=port-number
```

5. Ouvrez les ports ajoutés dans le pare-feu en exécutant les commandes suivantes sur le client NFS :

```
firewall-cmd --permanent --add-port=<statd-tcp-port>/tcp
firewall-cmd --permanent --add-port=<statd-udp-port>/udp
```

6. Recharger la configuration du pare-feu :

```
firewall-cmd --reload
```

7. Redémarrez le service **rpc-statd**:

```
# systemctl restart rpc-statd.service
```

Alternativement, si vous avez spécifié les ports **lockd** dans le fichier **/etc/modprobe.d/lockd.conf**:

- a. Mettre à jour les valeurs actuelles de **/proc/sys/fs/nfs/nlm_tcpport** et **/proc/sys/fs/nfs/nlm_udpport**:

```
# sysctl -w fs.nfs.nlm_tcpport=<tcp-port>
# sysctl -w fs.nfs.nlm_udpport=<udp-port>
```

- b. Redémarrez le service **rpc-statd**:

```
# systemctl restart rpc-statd.service
```

4.11.4. Configurer un client NFSv4 pour qu'il fonctionne derrière un pare-feu

N'effectuez cette procédure que si le client utilise NFSv4.0. Dans ce cas, il est nécessaire d'ouvrir un port pour les rappels NFSv4.0.

Cette procédure n'est pas nécessaire pour NFSv4.1 ou supérieur, car dans les versions ultérieures du protocole, le serveur effectue des rappels sur la même connexion que celle initiée par le client.

Procédure

1. Pour permettre aux rappels NFSv4.0 de traverser les pare-feux, définissez **/proc/sys/fs/nfs/nfs_callback_tcpport** et autorisez le serveur à se connecter à ce port sur le client comme suit :

```
# echo "fs.nfs.nfs_callback_tcpport = <callback-port>" >/etc/sysctl.d/90-nfs-callback-
port.conf
# sysctl -p /etc/sysctl.d/90-nfs-callback-port.conf
```

2. Ouvrez le port spécifié dans le pare-feu en exécutant la commande suivante sur le client NFS :

```
firewall-cmd --permanent --add-port=<callback-port>/tcp
```

3. Recharger la configuration du pare-feu :

```
firewall-cmd --reload
```

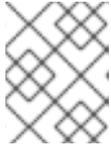
4.12. EXPORTER DES QUOTAS RPC À TRAVERS UN PARE-FEU

Si vous exportez un système de fichiers qui utilise des quotas de disque, vous pouvez utiliser le service RPC (Remote Procedure Call) quota pour fournir des données de quotas de disque aux clients NFS.

Procédure

1. Activez et démarrez le service **rpc-rquotad**:

```
# systemctl enable --now rpc-rquotad
```

**NOTE**

Le service **rpc-rquotad** est, s'il est activé, démarré automatiquement après le démarrage du service `nfs-server`.

2. Pour rendre le service quota RPC accessible derrière un pare-feu, le port TCP (ou UDP, si UDP est activé) 875 doit être ouvert. Le numéro de port par défaut est défini dans le fichier `/etc/services`.
Vous pouvez remplacer le numéro de port par défaut en ajoutant **-p port-number** à la variable **RPCRQUOTADOPTS** dans le fichier `/etc/sysconfig/rpc-rquotad`.
3. Par défaut, les hôtes distants ne peuvent que lire les quotas. Si vous souhaitez autoriser les clients à définir des quotas, ajoutez l'option **-S** à la variable **RPCRQUOTADOPTS** dans le fichier `/etc/sysconfig/rpc-rquotad`.
4. Redémarrez **rpc-rquotad** pour que les modifications apportées au fichier `/etc/sysconfig/rpc-rquotad` prennent effet :

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

4.13. ACTIVATION DE NFS SUR RDMA (NFSORDMA)

Dans Red Hat Enterprise Linux 9, le service d'accès direct à la mémoire à distance (RDMA) sur du matériel compatible RDMA fournit une prise en charge du protocole Network File System (NFS) pour le transfert de fichiers à grande vitesse sur le réseau.

Procédure

1. Installez le paquetage **rdma-core**:

```
# dnf install rdma-core
```

2. Vérifiez que les lignes contenant **xprtrdma** et **svcrdma** ne sont pas commentées dans le fichier `/etc/rdma/modules/rdma.conf`:

```
# NFS over RDMA client support
xprtrdma
# NFS over RDMA server support
svcrdma
```

3. Sur le serveur NFS, créez le répertoire `/mnt/nfsordma` et exportez-le vers `/etc/exports`:

```
# mkdir /mnt/nfsordma
# echo "/mnt/nfsordma *(fsid=0,rw,async,insecure,no_root_squash)" >> /etc/exports
```

4. Sur le client NFS, montez le partage nfs avec l'adresse IP du serveur, par exemple, **172.31.0.186**:

```
# mount -o rdma,port=20049 172.31.0.186:/mnt/nfs-share /mnt/nfs
```

5. Redémarrez le service **nfs-server**:

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

-

Ressources supplémentaires

- [La norme RFC 5667](#)

4.14. RESSOURCES SUPPLÉMENTAIRES

- [Le wiki Linux NFS](#)

CHAPITRE 5. CONFIGURATION D'UN SERVEUR NFSV4 UNIQUEMENT

En tant qu'administrateur du serveur NFS, vous pouvez configurer le serveur NFS pour qu'il ne prenne en charge que NFSv4, ce qui réduit le nombre de ports ouverts et de services en cours d'exécution sur le système.

5.1. AVANTAGES ET INCONVÉNIENTS D'UN SERVEUR NFSV4 UNIQUEMENT

Cette section explique les avantages et les inconvénients de la configuration du serveur NFS pour qu'il ne prenne en charge que NFSv4.

Par défaut, le serveur NFS prend en charge les connexions NFSv3 et NFSv4 dans Red Hat Enterprise Linux 9. Cependant, vous pouvez également configurer NFS pour qu'il ne prenne en charge que la version 4.0 et ultérieure de NFS. Cela minimise le nombre de ports ouverts et de services en cours d'exécution sur le système, car NFSv4 ne nécessite pas que le service **rpcbind** écoute sur le réseau.

Lorsque votre serveur NFS est configuré en tant que NFSv4 uniquement, les clients qui tentent de monter des partages à l'aide de NFSv3 se heurtent à une erreur du type suivant :

```
Requested NFS version or transport protocol is not supported.
```

Vous pouvez également désactiver l'écoute des appels de protocole **RPCBIND**, **MOUNT** et **NSM**, qui ne sont pas nécessaires dans le cas de NFSv4 uniquement.

Les effets de la désactivation de ces options supplémentaires sont les suivants :

- Les clients qui tentent de monter des partages sur votre serveur à l'aide de NFSv3 ne répondent plus.
- Le serveur NFS lui-même n'est pas en mesure de monter des systèmes de fichiers NFSv3.

5.2. CONFIGURER LE SERVEUR NFS POUR QU'IL NE PRENNE EN CHARGE QUE NFSV4

Cette procédure décrit comment configurer votre serveur NFS pour qu'il ne prenne en charge que la version 4.0 et les versions ultérieures de NFS.

Procédure

1. Désactivez NFSv3 en ajoutant les lignes suivantes à la section **[nfsd]** du fichier de configuration **/etc/nfs.conf**:

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

2. En option, désactivez l'écoute des appels de protocole **RPCBIND**, **MOUNT** et **NSM**, qui ne sont pas nécessaires dans le cas de NFSv4 uniquement. Désactiver les services connexes :

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

3. Redémarrez le serveur NFS :

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

Les modifications prennent effet dès que vous démarrez ou redémarrez le serveur NFS.

5.3. VÉRIFICATION DE LA CONFIGURATION NFSV4-ONLY

Cette procédure décrit comment vérifier que votre serveur NFS est configuré en mode NFSv4 uniquement à l'aide de l'utilitaire **netstat**.

Procédure

- Utilisez l'utilitaire **netstat** pour dresser la liste des services qui écoutent les protocoles TCP et UDP :

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

Exemple 5.1. Sortie sur un serveur NFSv4 uniquement

Voici un exemple de sortie de **netstat** sur un serveur NFSv4 uniquement ; l'écoute de **RPCBIND**, **MOUNT** et **NSM** est également désactivée. Ici, **nfs** est le seul service NFS en écoute :

```
# 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:*
```

Exemple 5.2. Sortie avant la configuration d'un serveur NFSv4 uniquement

En comparaison, la sortie **netstat** avant la configuration d'un serveur NFSv4 uniquement comprend les services **sunrpc** et **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 [::]:*
```

CHAPITRE 6. SÉCURISATION DE NFS

Pour minimiser les risques de sécurité NFS et protéger les données sur le serveur, tenez compte des sections suivantes lorsque vous exportez des systèmes de fichiers NFS sur un serveur ou que vous les montez sur un client.

6.1. SÉCURITÉ NFS AVEC AUTH_SYS ET CONTRÔLE DES EXPORTATIONS

NFS propose les options traditionnelles suivantes pour contrôler l'accès aux fichiers exportés :

- Le serveur limite les hôtes autorisés à monter les systèmes de fichiers par adresse IP ou par nom d'hôte.
- Le serveur applique les autorisations du système de fichiers pour les utilisateurs des clients NFS de la même manière qu'il le fait pour les utilisateurs locaux. Traditionnellement, NFS utilise pour cela le message d'appel **AUTH_SYS** (également appelé **AUTH_UNIX**), qui s'appuie sur le client pour indiquer l'UID et le GID de l'utilisateur. Il faut savoir qu'un client malveillant ou mal configuré peut facilement se tromper et permettre à un utilisateur d'accéder à des fichiers qu'il ne devrait pas.

Pour limiter les risques potentiels, les administrateurs limitent souvent l'accès à la lecture seule ou écrasent les autorisations d'un utilisateur et d'un groupe d'ID communs. Malheureusement, ces solutions empêchent d'utiliser le partage NFS de la manière prévue à l'origine.

En outre, si un pirate prend le contrôle du serveur DNS utilisé par le système exportant le système de fichiers NFS, il peut faire pointer le système associé à un nom d'hôte particulier ou à un nom de domaine pleinement qualifié vers une machine non autorisée. À ce stade, la machine non autorisée *is* est le système autorisé à monter le partage NFS, car aucun nom d'utilisateur ou mot de passe n'est échangé pour fournir une sécurité supplémentaire pour le montage NFS.

Les caractères génériques doivent être utilisés avec parcimonie lors de l'exportation de répertoires via NFS, car il est possible que le champ d'application du caractère générique englobe plus de systèmes que prévu.

Ressources supplémentaires

- Pour sécuriser NFS et **rpcbind**, utilisez, par exemple, **nftables** et **firewalld**.
- **nft(8)** page de manuel
- **firewalld-cmd(1)** page de manuel

6.2. SÉCURITÉ NFS AVEC AUTH_GSS

Toutes les versions de NFS prennent en charge RPCSEC_GSS et le mécanisme Kerberos.

Contrairement à AUTH_SYS, avec le mécanisme Kerberos RPCSEC_GSS, le serveur ne dépend pas du client pour représenter correctement l'utilisateur qui accède au fichier. La cryptographie est utilisée pour authentifier les utilisateurs auprès du serveur, ce qui empêche un client malveillant d'usurper l'identité d'un utilisateur sans disposer des informations d'identification Kerberos de ce dernier. L'utilisation du mécanisme Kerberos RPCSEC_GSS est le moyen le plus simple de sécuriser les montages car, une fois Kerberos configuré, aucune autre installation n'est nécessaire.

6.3. CONFIGURATION D'UN SERVEUR ET D'UN CLIENT NFS POUR L'UTILISATION DE KERBEROS

Kerberos est un système d'authentification réseau qui permet aux clients et aux serveurs de s'authentifier les uns les autres en utilisant un cryptage symétrique et un tiers de confiance, le KDC. Red Hat recommande d'utiliser Identity Management (IdM) pour configurer Kerberos.

Conditions préalables

- Le centre de distribution de clés Kerberos (**KDC**) est installé et configuré.

Procédure

1. • Créer le **nfs/hostname.domain@REALM** du côté du serveur NFS.
 - Créer le **host/hostname.domain@REALM** tant du côté du serveur que du côté du client.
 - Ajouter les clés correspondantes aux keytabs du client et du serveur.
2. Côté serveur, utilisez l'option **sec=** pour activer les variantes de sécurité souhaitées. Pour activer toutes les saveurs de sécurité ainsi que les montages non cryptographiques :

```
/export *(sec=sys:krb5:krb5i:krb5p)
```

Les saveurs de sécurité valables à utiliser avec l'option **sec=** sont les suivantes :

- **sys**: pas de protection cryptographique, la valeur par défaut de l'option
 - **krb5**authentification uniquement
 - **krb5i**protection de l'intégrité : protection de l'intégrité
 - utilise Kerberos V5 pour l'authentification des utilisateurs et effectue un contrôle d'intégrité des opérations NFS à l'aide de sommes de contrôle sécurisées afin d'empêcher la falsification des données.
 - **krb5p**protection de la vie privée : protection de la vie privée
 - utilise Kerberos V5 pour l'authentification des utilisateurs, le contrôle d'intégrité et le chiffrement du trafic NFS afin d'empêcher le reniflage du trafic. Il s'agit du paramètre le plus sûr, mais c'est aussi celui qui entraîne le plus de surcoût en termes de performances.
3. Côté client, ajoutez **sec=krb5** (ou **sec=krb5i**, ou **sec=krb5p**, selon la configuration) aux options de montage :

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

Ressources supplémentaires

- [Création de fichiers en tant que root sur NFS sécurisé par krb5](#) . Déconseillé.
- **exports(5)** page de manuel
- **nfs(5)** page de manuel

6.4. OPTIONS DE SÉCURITÉ NFSV4

NFSv4 inclut un support ACL basé sur le modèle Microsoft Windows NT, et non sur le modèle POSIX, en raison des fonctionnalités du modèle Microsoft Windows NT et de son large déploiement.

Une autre caractéristique de sécurité importante de NFSv4 est la suppression de l'utilisation du protocole **MOUNT** pour le montage des systèmes de fichiers. Le protocole **MOUNT** présentait un risque de sécurité en raison de la manière dont il traitait les fichiers.

6.5. DROITS D'ACCÈS AUX FICHIERS SUR LES EXPORTATIONS NFS MONTÉES

Une fois que le système de fichiers NFS est monté en lecture ou en lecture et écriture par un hôte distant, la seule protection dont bénéficie chaque fichier partagé est celle de ses autorisations. Si deux utilisateurs partageant la même valeur d'ID utilisateur montent le même système de fichiers NFS sur des systèmes clients différents, ils peuvent modifier leurs fichiers respectifs. En outre, toute personne connectée en tant que root sur le système client peut utiliser la commande **su -** pour accéder à tous les fichiers du partage NFS.

Par défaut, les listes de contrôle d'accès (ACL) sont prises en charge par NFS sous Red Hat Enterprise Linux. Red Hat recommande de garder cette fonctionnalité activée.

Par défaut, NFS utilise *root squashing* lors de l'exportation d'un système de fichiers. L'ID utilisateur de toute personne accédant au partage NFS en tant qu'utilisateur root sur sa machine locale est ainsi défini sur **nobody**. L'écrasement de la racine est contrôlé par l'option par défaut **root_squash**; pour plus d'informations sur cette option, voir [Configuration du serveur NFS](#).

Lorsque vous exportez un partage NFS en lecture seule, pensez à utiliser l'option **all_squash**. Cette option fait en sorte que chaque utilisateur accédant au système de fichiers exporté prenne l'identifiant de l'utilisateur **nobody**.

CHAPITRE 7. ACTIVATION DES CONFIGURATIONS SCSI PNFS DANS NFS

Vous pouvez configurer le serveur et le client NFS pour qu'ils utilisent la disposition pNFS SCSI pour accéder aux données. pNFS SCSI est utile dans les cas d'utilisation qui impliquent un accès à un fichier par un seul client pendant une longue période.

Conditions préalables

- Le client et le serveur doivent pouvoir envoyer des commandes SCSI au même périphérique de bloc. En d'autres termes, le périphérique de bloc doit se trouver sur un bus SCSI partagé.
- Le périphérique de bloc doit contenir un système de fichiers XFS.
- Le périphérique SCSI doit prendre en charge les réservations persistantes SCSI telles que décrites dans la spécification des commandes primaires SCSI-3.

7.1. LA TECHNOLOGIE PNFS

L'architecture pNFS améliore l'évolutivité de NFS. Lorsqu'un serveur met en œuvre pNFS, le client peut accéder aux données par l'intermédiaire de plusieurs serveurs simultanément. Cela permet d'améliorer les performances.

pNFS prend en charge les protocoles ou dispositions de stockage suivants sur RHEL :

- Dossiers
- Flexfiles
- SCSI

7.2. DISPOSITIONS SCSI PNFS

L'agencement SCSI s'appuie sur le travail des agencements de blocs pNFS. L'agencement est défini sur l'ensemble des périphériques SCSI. Elle contient une série séquentielle de blocs de taille fixe en tant qu'unités logiques (LU) qui doivent être capables de prendre en charge les réservations persistantes SCSI. Les périphériques LU sont identifiés par leur identification de périphérique SCSI.

pNFS SCSI donne de bons résultats dans les cas d'utilisation qui impliquent l'accès à un fichier par un seul client pendant une longue période. Il peut s'agir par exemple d'un serveur de messagerie ou d'une machine virtuelle hébergeant un cluster.

Opérations entre le client et le serveur

Lorsqu'un client NFS lit ou écrit dans un fichier, il effectue une opération **LAYOUTGET**. Le serveur répond en indiquant l'emplacement du fichier sur le périphérique SCSI. Il se peut que le client doive effectuer une opération supplémentaire (**GETDEVICEINFO**) pour déterminer le périphérique SCSI à utiliser. Si ces opérations fonctionnent correctement, le client peut envoyer des demandes d'E/S directement au périphérique SCSI au lieu d'envoyer les opérations **READ** et **WRITE** au serveur.

Des erreurs ou des conflits entre clients peuvent amener le serveur à rappeler des schémas ou à ne pas les transmettre aux clients. Dans ce cas, les clients se contentent d'envoyer les opérations **READ** et **WRITE** au serveur au lieu d'envoyer des demandes d'E/S directement au périphérique SCSI.

Pour surveiller les opérations, voir [Surveillance de la fonctionnalité des layouts SCSI pNFS](#).

Réservations d'appareils

le pNFS SCSI gère les clôtures par l'attribution de réservations. Avant que le serveur n'envoie des layouts aux clients, il réserve le périphérique SCSI pour s'assurer que seuls les clients enregistrés peuvent y accéder. Si un client peut envoyer des commandes à ce périphérique SCSI mais qu'il n'est pas enregistré auprès de celui-ci, de nombreuses opérations effectuées par le client sur ce périphérique échouent. Par exemple, la commande **blkid** sur le client n'affiche pas l'UUID du système de fichiers XFS si le serveur n'a pas donné au client une disposition pour ce périphérique.

Le serveur ne supprime pas sa propre réservation persistante. Cela permet de protéger les données du système de fichiers sur le périphérique en cas de redémarrage des clients et des serveurs. Pour réutiliser le périphérique SCSI, il peut être nécessaire de supprimer manuellement la réservation persistante sur le serveur NFS.

7.3. RECHERCHE D'UN PÉRIPHÉRIQUE SCSI COMPATIBLE AVEC PNFS

Cette procédure permet de vérifier si un périphérique SCSI prend en charge l'agencement SCSI pNFS.

Conditions préalables

- Installez le paquetage **sg3_utils**:

```
# dnf install sg3_utils
```

Procédure

- Sur le serveur et le client, vérifiez que les périphériques SCSI sont correctement pris en charge :

```
# sg_persist --in --report-capabilities --verbose path-to-scsi-device
```

Assurez-vous que le bit *Persist Through Power Loss Active* (**PTPL_A**) est activé.

Exemple 7.1. Un périphérique SCSI qui prend en charge pNFS SCSI

Voici un exemple de la sortie de **sg_persist** pour un périphérique SCSI qui prend en charge pNFS SCSI. Le bit **PTPL_A** indique **1**.

```
inquiry cdb: 12 00 00 00 24 00
Persistent Reservation In cmd: 5e 02 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
```

Ressources supplémentaires

- [sg_persist\(8\)](#) page de manuel

7.4. CONFIGURATION DE PNFS SCSI SUR LE SERVEUR

Cette procédure permet de configurer un serveur NFS pour qu'il exporte une disposition SCSI pNFS.

Procédure

1. Sur le serveur, montez le système de fichiers XFS créé sur le périphérique SCSI.
2. Configurez le serveur NFS pour qu'il exporte la version 4.1 ou supérieure de NFS. Définissez l'option suivante dans la section **[nfsd]** du fichier **/etc/nfs.conf**:

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

3. Configurez le serveur NFS pour exporter le système de fichiers XFS sur NFS avec l'option **pnfs**:

Exemple 7.2. Une entrée dans **/etc/exports** pour exporter pNFS SCSI

L'entrée suivante dans le fichier de configuration **/etc/exports** exporte le système de fichiers monté sur **/exported/directory/** vers le client **allowed.example.com** sous la forme d'une disposition SCSI pNFS :

```
/exported/directory allowed.example.com(pnfs)
```



NOTE

Le système de fichiers exporté doit être créé sur l'ensemble du périphérique de bloc, et pas seulement sur une partition.

Ressources supplémentaires

- [Exportation des partages NFS.](#)

7.5. CONFIGURATION DE PNFS SCSI SUR LE CLIENT

Cette procédure permet de configurer un client NFS pour monter une disposition SCSI pNFS.

Conditions préalables

- Le serveur NFS est configuré pour exporter un système de fichiers XFS via pNFS SCSI. Voir [Configuration de pNFS SCSI sur le serveur](#) .

Procédure

- Sur le client, montez le système de fichiers XFS exporté à l'aide de la version 4.1 ou supérieure de NFS :

```
# mount -t nfs -o nfsvers=4.1 host:/remote/export /local/directory
```

Ne montez pas le système de fichiers XFS directement sans NFS.

Ressources supplémentaires

- [Montage des partages NFS.](#)

7.6. LIBÉRATION DE LA RÉSERVATION SCSI PNFS SUR LE SERVEUR

Cette procédure libère la réservation persistante qu'un serveur NFS détient sur un périphérique SCSI. Cela vous permet de réutiliser le périphérique SCSI lorsque vous n'avez plus besoin d'exporter pNFS SCSI.

Vous devez supprimer la réservation du serveur. Elle ne peut pas être supprimée d'un autre Nexus informatique.

Conditions préalables

- Installez le paquetage **sg3_utils**:

```
# dnf install sg3_utils
```

Procédure

1. Interroger une réservation existante sur le serveur :

```
# sg_persist --read-reservation path-to-scsi-device
```

Exemple 7.3. Interroger une réservation sur /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. Supprimer l'enregistrement existant sur le serveur :

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

Exemple 7.4. Suppression d'une réservation sur /dev/sda

```
# sg_persist --out \  
  --release \  
  --param-rk=0x1000000000000000 \  
  --prout-type=6 \  
  /dev/sda
```

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

Ressources supplémentaires

- **sg_persist(8)** page de manuel

CHAPITRE 8. SURVEILLANCE DE LA FONCTIONNALITÉ DES LAYOUTS SCSI PNFS

Vous pouvez contrôler si le client et le serveur pNFS échangent des opérations SCSI pNFS correctes ou s'ils se rabattent sur des opérations NFS normales.

Conditions préalables

- Un client et un serveur SCSI pNFS sont configurés.

8.1. VÉRIFICATION DES OPÉRATIONS SCSI PNFS À PARTIR DU SERVEUR À L'AIDE DE NFSSTAT

Cette procédure utilise l'utilitaire **nfsstat** pour surveiller les opérations SCSI pNFS à partir du serveur.

Procédure

1. Contrôler les opérations effectuées à partir du serveur :

```
# 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%
--
getdevlist layoutcommit layoutget layoutreturn secinfonyonam sequence
0      0% 29     1% 49     1% 5      0% 0      0% 2435 86%
```

2. Le client et le serveur utilisent les opérations SCSI pNFS lorsque :

- Les compteurs **layoutget**, **layoutreturn** et **layoutcommit** s'incrémentent. Cela signifie que le serveur sert des modèles.
- Les compteurs du serveur **read** et **write** ne s'incrémentent pas. Cela signifie que les clients effectuent des requêtes d'E/S directement sur les périphériques SCSI.

8.2. VÉRIFICATION DES OPÉRATIONS SCSI PNFS À PARTIR DU CLIENT À L'AIDE DE MOUNTSTATS

Cette procédure utilise le fichier **/proc/self/mountstats** pour surveiller les opérations SCSI pNFS du client.

Procédure

1. Liste des compteurs d'opérations par montage :

```
# 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=0xfdfbfbff,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. Dans les résultats :

- Les statistiques **LAYOUT** indiquent les demandes pour lesquelles le client et le serveur utilisent des opérations SCSI pNFS.
- Les statistiques **READ** et **WRITE** indiquent les demandes pour lesquelles le client et le serveur reviennent à des opérations NFS.

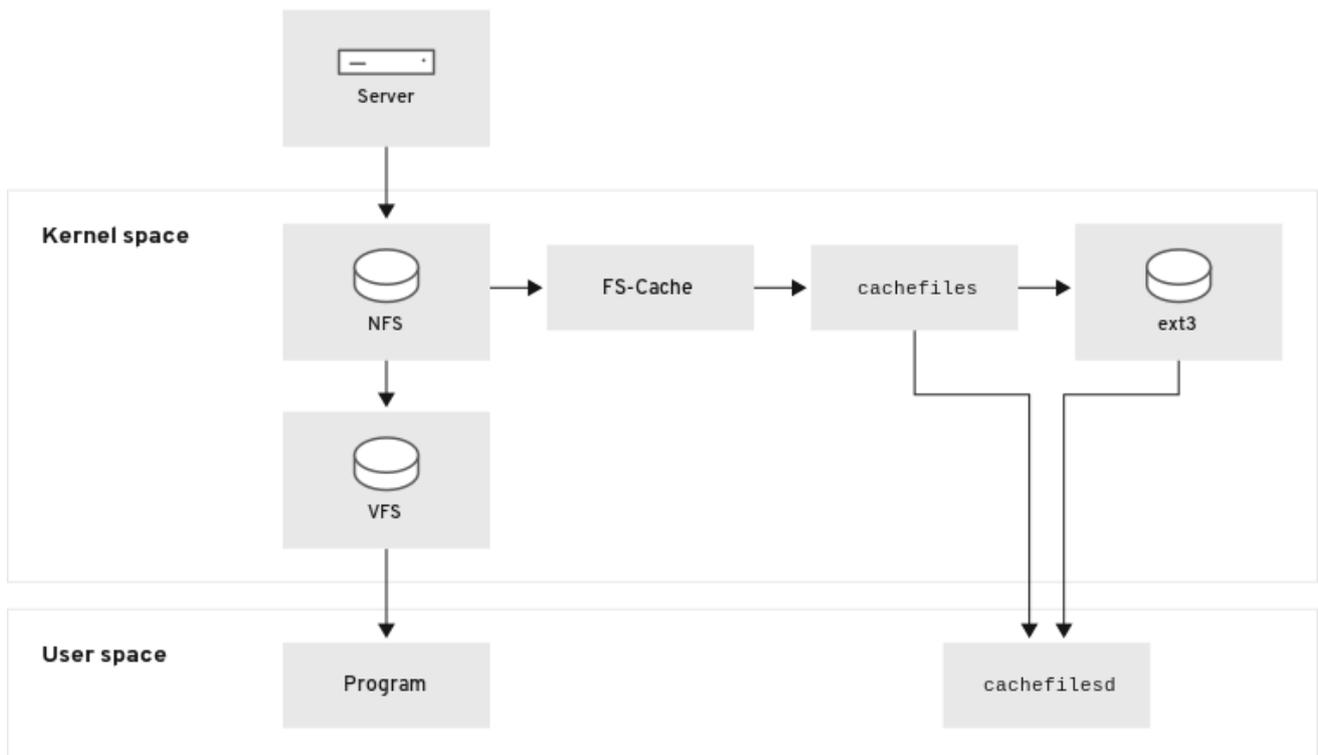
CHAPITRE 9. PREMIERS PAS AVEC FS-CACHE

FS-Cache est un cache local persistant que les systèmes de fichiers peuvent utiliser pour récupérer des données sur le réseau et les mettre en cache sur le disque local. Cela permet de réduire le trafic réseau pour les utilisateurs qui accèdent aux données d'un système de fichiers monté sur le réseau (par exemple, NFS).

9.1. VUE D'ENSEMBLE DU FS-CACHE

Le diagramme suivant est une illustration de haut niveau du fonctionnement de FS-Cache :

Figure 9.1. Aperçu de FS-Cache



94_RHEL_0720

FS-Cache est conçu pour être aussi transparent que possible pour les utilisateurs et les administrateurs d'un système. Contrairement à **cachefs** sous Solaris, FS-Cache permet à un système de fichiers sur un serveur d'interagir directement avec le cache local d'un client sans créer de système de fichiers surmonté. Avec NFS, une option de montage indique au client de monter le partage NFS avec FS-cache activé. Le point de montage entraînera le téléchargement automatique de deux modules du noyau : **fscache** et **cachefiles**. Le démon **cachefilesd** communique avec les modules du noyau pour mettre en œuvre le cache.

FS-Cache ne modifie pas le fonctionnement de base d'un système de fichiers qui fonctionne sur le réseau - il fournit simplement à ce système de fichiers un emplacement persistant dans lequel il peut mettre des données en cache. Par exemple, un client peut toujours monter un partage NFS, que FS-Cache soit activé ou non. En outre, le système NFS mis en cache peut gérer des fichiers qui ne tiennent pas dans le cache (individuellement ou collectivement), car les fichiers peuvent être partiellement mis en cache et n'ont pas besoin d'être lus complètement au départ. FS-Cache cache également toutes les erreurs d'E/S qui se produisent dans le cache au pilote du système de fichiers client.

Pour fournir des services de mise en cache, FS-Cache a besoin d'un *cache back end*. Un back-end de cache est un pilote de stockage configuré pour fournir des services de mise en cache, qui est **cachefiles**.

Dans ce cas, FS-Cache a besoin d'un système de fichiers en mode bloc monté qui prend en charge **bmap** et des attributs étendus (par exemple ext3) en tant que back-end de cache.

Les systèmes de fichiers qui prennent en charge les fonctionnalités requises par le back-end du cache FS-Cache comprennent les implémentations Red Hat Enterprise Linux 9 des systèmes de fichiers suivants :

- ext3 (avec les attributs étendus activés)
- ext4
- XFS

FS-Cache ne peut pas mettre en cache arbitrairement n'importe quel système de fichiers, que ce soit via le réseau ou autrement : le pilote du système de fichiers partagé doit être modifié pour permettre l'interaction avec FS-Cache, le stockage et l'extraction des données, ainsi que la configuration et la validation des métadonnées. FS-Cache a besoin de *indexing keys* et *coherency data* du système de fichiers mis en cache pour assurer la persistance : des clés d'indexation pour faire correspondre les objets du système de fichiers aux objets du cache, et des données de cohérence pour déterminer si les objets du cache sont toujours valides.



NOTE

Dans Red Hat Enterprise Linux 9, le paquetage **cachefilesd** n'est pas installé par défaut et doit être installé manuellement.

9.2. GARANTIE DE PERFORMANCE

FS-Cache ne *not* garantit pas de meilleures performances. L'utilisation d'un cache entraîne une pénalité en termes de performances : par exemple, les partages NFS mis en cache ajoutent des accès au disque aux recherches inter-réseaux. Bien que FS-Cache s'efforce d'être aussi asynchrone que possible, il existe des chemins synchrones (par exemple, les lectures) où cela n'est pas possible.

Par exemple, l'utilisation de FS-Cache pour mettre en cache un partage NFS entre deux ordinateurs sur un réseau GigE par ailleurs peu chargé ne permettra probablement pas d'améliorer les performances en matière d'accès aux fichiers. Les requêtes NFS seront plutôt satisfaites plus rapidement à partir de la mémoire du serveur qu'à partir du disque local.

L'utilisation de FS-Cache est donc une question de *compromise* entre plusieurs facteurs. Si FS-Cache est utilisé pour mettre en cache le trafic NFS, par exemple, il peut ralentir légèrement le client, mais réduire massivement la charge du réseau et du serveur en satisfaisant les demandes de lecture localement sans consommer la bande passante du réseau.

9.3. MISE EN PLACE D'UN CACHE

Actuellement, Red Hat Enterprise Linux 9 ne fournit que le back-end de mise en cache **cachefiles**. Le démon **cachefilesd** lance et gère **cachefiles**. Le fichier **/etc/cachefilesd.conf** contrôle la manière dont **cachefiles** fournit les services de mise en cache.

Le back-end du cache fonctionne en maintenant une certaine quantité d'espace libre sur la partition hébergeant le cache. Il agrandit et réduit le cache en fonction de l'espace libre utilisé par d'autres éléments du système, ce qui permet de l'utiliser en toute sécurité sur le système de fichiers racine (par exemple, sur un ordinateur portable). FS-Cache définit des valeurs par défaut pour ce comportement, qui peuvent être configurées via *cache cull limits*. Pour plus d'informations sur la configuration des limites du cache, voir [Configuration des limites du cache](#).

Cette procédure montre comment mettre en place un cache.

Conditions préalables

- Le paquet **cachefilesd** est installé et le service a démarré avec succès. Pour s'assurer que le service est en cours d'exécution, utilisez la commande suivante :

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

Le statut doit être *active (running)*.

Procédure

1. Configurez dans un back-end de cache le répertoire à utiliser comme cache, en utilisant le paramètre suivant :

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

2. En règle générale, le répertoire du cache est défini dans **/etc/cachefilesd.conf** comme **/var/cache/fscache**, comme dans :

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

3. Si vous voulez changer le répertoire du cache, le contexte selinux doit être le même que **/var/cache/fscache**:

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

4. Remplacez */path/to/cache* par le nom du répertoire lors de la mise en place du cache.
5. Si les commandes données pour définir le contexte selinux n'ont pas fonctionné, utilisez les commandes suivantes :

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

FS-Cache stocke le cache dans le système de fichiers qui héberge le **/path/to/cache**. Sur un ordinateur portable, il est conseillé d'utiliser le système de fichiers racine (*/*) comme système de fichiers hôte, mais sur un ordinateur de bureau, il serait plus prudent de monter une partition de disque spécifiquement pour le cache.

6. Le système de fichiers hôte doit prendre en charge les attributs étendus définis par l'utilisateur ; FS-Cache utilise ces attributs pour stocker les informations relatives au maintien de la cohérence. Pour activer les attributs étendus définis par l'utilisateur pour les systèmes de fichiers ext3 (c'est-à-dire **device**), utilisez :

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

7. Pour activer les attributs étendus d'un système de fichiers au moment du montage, vous pouvez utiliser la commande suivante :

```
# mount /dev/device /path/to/cache -o user_xattr
```

- Une fois le fichier de configuration en place, démarrez le service **cached**:

```
# systemctl start cached
```

- Pour configurer **cached** afin qu'il démarre au moment du démarrage, exécutez la commande suivante en tant que root :

```
# systemctl enable cached
```

9.4. CONFIGURATION DES LIMITES DU CACHE

Le démon **cached** fonctionne en mettant en cache les données distantes des systèmes de fichiers partagés dans l'espace libre du disque. Cela peut potentiellement consommer tout l'espace libre disponible, ce qui peut être mauvais si le disque héberge également la partition racine. Pour contrôler ce phénomène, **cached** tente de maintenir un certain espace libre en supprimant du cache les anciens objets (c'est-à-dire ceux qui ont été accédés moins récemment). Ce comportement est connu sous le nom de *cache culling*.

L'élimination du cache se fait sur la base du pourcentage de blocs et du pourcentage de fichiers disponibles dans le système de fichiers sous-jacent. Il existe des paramètres dans **/etc/cached.conf** qui contrôlent six limites :

brun *N*% (pourcentage de blocs), **frun** *N*% (pourcentage de fichiers)

Si l'espace libre et le nombre de fichiers disponibles dans le cache dépassent ces deux limites, l'élimination est désactivée.

bcull *N*% (pourcentage de blocs), **fcull** *N*% (pourcentage de fichiers)

Si l'espace disponible ou le nombre de fichiers dans le cache est inférieur à l'une ou l'autre de ces limites, l'élimination est lancée.

bstop *N*% (pourcentage de blocs), **fstop** *N*% (pourcentage de fichiers)

Si l'espace disponible ou le nombre de fichiers disponibles dans le cache tombe en dessous de l'une ou l'autre de ces limites, aucune autre allocation d'espace disque ou de fichiers n'est autorisée jusqu'à ce que l'élimination ait permis de repasser au-dessus de ces limites.

La valeur par défaut de **N** pour chaque paramètre est la suivante :

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

Lors de la configuration de ces paramètres, les points suivants doivent être respectés :

- $0 \leq \mathbf{bstop} < \mathbf{bcull} < \mathbf{brun} < 100$
- $0 \leq \mathbf{fstop} < \mathbf{fcull} < \mathbf{frun} < 100$

Il s'agit des pourcentages d'espace disponible et de fichiers disponibles, qui ne correspondent pas à 100 moins le pourcentage affiché par le programme **df**.



IMPORTANT

L'élimination dépend simultanément des paires `bxxx` et `fxxx`; l'utilisateur ne peut pas les traiter séparément.

9.5. RÉCUPÉRATION D'INFORMATIONS STATISTIQUES DU MODULE NOYAU FSCACHE

FS-Cache conserve également des informations statistiques générales. Cette procédure montre comment obtenir ces informations.

Procédure

1. Pour afficher les informations statistiques sur FS-Cache, utilisez la commande suivante :

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

Les statistiques FS-Cache comprennent des informations sur les points de décision et les compteurs d'objets. Pour plus d'informations, voir le document suivant sur le noyau :

</usr/share/doc/kernel-doc-4.18.0/Documentation/filesystems/caching/fscache.txt>

9.6. RÉFÉRENCES FS-CACHE

Cette section fournit des informations de référence sur FS-Cache.

1. Pour plus d'informations sur **cachefilesd** et la manière de le configurer, voir **man cachefilesd** et **man cachefilesd.conf**. Les documents suivants sur le noyau fournissent également des informations supplémentaires :
 - </usr/share/doc/cachefilesd/README>
 - </usr/share/man/man5/cachefilesd.conf.5.gz>
 - </usr/share/man/man8/cachefilesd.8.gz>
2. Pour des informations générales sur FS-Cache, y compris des détails sur ses contraintes de conception, les statistiques disponibles et ses capacités, voir le document suivant sur le noyau :
</usr/share/doc/kernel-doc-4.18.0/Documentation/filesystems/caching/fscache.txt>

CHAPITRE 10. UTILISATION DU CACHE AVEC NFS

NFS n'utilisera pas le cache à moins d'en avoir reçu l'instruction explicite. Ce paragraphe montre comment configurer un montage NFS en utilisant FS-Cache.

Conditions préalables

- Le paquet **cachefilesd** est installé et fonctionne. Pour vous en assurer, utilisez la commande suivante :

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

Le statut doit être *active (running)*.

- Montez les partages NFS avec l'option suivante :

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

Tous les accès aux fichiers sous **/mount/point** passent par le cache, sauf si le fichier est ouvert pour des entrées/sorties ou des écritures directes. Pour plus d'informations, voir [Limitations du cache avec NFS](#).

NFS indexe le contenu du cache à l'aide de l'identifiant du fichier NFS, *not* le nom du fichier, ce qui signifie que les fichiers liés en dur partagent correctement le cache.

Les versions 3, 4.0, 4.1 et 4.2 de NFS prennent en charge la mise en cache. Cependant, chaque version utilise des branches différentes pour la mise en cache.

10.1. CONFIGURATION DU PARTAGE DU CACHE NFS

Le partage du cache NFS peut poser plusieurs problèmes. Le cache étant persistant, les blocs de données qu'il contient sont indexés sur une séquence de quatre clés :

- Niveau 1 : Détails du serveur
- Niveau 2 : Quelques options de montage ; type de sécurité ; FSID ; identifiant unique
- Niveau 3 : Poignée de dossier
- Niveau 4 : Numéro de page dans le fichier

Pour éviter les problèmes de gestion de la cohérence entre les superblocs, tous les superblocs NFS qui doivent mettre les données en cache ont des clés de niveau 2 uniques. Normalement, deux montages NFS avec le même volume source et les mêmes options partagent un superbloc, et donc la mise en cache, même s'ils montent des répertoires différents à l'intérieur de ce volume.

Voici un exemple de configuration du partage de cache avec différentes options.

Procédure

- Montez les partages NFS à l'aide des commandes suivantes :

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

-

Ici, **/home/fred** et **/home/jim** partagent probablement le superbloc puisqu'ils ont les mêmes options, surtout s'ils proviennent du même volume/partition sur le serveur NFS (**home0**).

2. Pour ne pas partager le superbloc, utilisez la commande **mount** avec les options suivantes :

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

Dans ce cas, **/home/fred** et **/home/jim** ne partageront pas le superbloc car ils ont des paramètres d'accès au réseau différents, qui font partie de la clé de niveau 2.

3. Pour mettre en cache le contenu des deux sous-arbres (**/home/fred1** et **/home/fred2**) *twice* qui ne partagent pas le superbloc, utilisez la commande suivante :

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

4. Une autre façon d'éviter le partage de superblocs est de le supprimer explicitement avec le paramètre **nosharecache**. Reprenons le même exemple :

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

Toutefois, dans ce cas, un seul des superblocs est autorisé à utiliser la mémoire cache, car rien ne permet de distinguer les clés de niveau 2 de **home0:/disk0/fred** et **home0:/disk0/jim**.

5. Pour spécifier l'adressage au superbloc, ajoutez un *unique identifier* sur au moins un des montages, c'est-à-dire **fsc=unique-identifier**.

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

Ici, l'identifiant unique **jim** est ajouté à la clé de niveau 2 utilisée dans le cache pour **/home/jim**.



IMPORTANT

L'utilisateur ne peut pas partager des caches entre des superblocs qui ont des communications ou des paramètres de protocole différents. Par exemple, il n'est pas possible de partager entre NFSv4.0 et NFSv3 ou entre NFSv4.1 et NFSv4.2 parce qu'ils imposent des superblocs différents. La définition de paramètres tels que la taille de lecture (*rsize*) empêche également le partage du cache car, là encore, elle impose un superbloc différent.

10.2. LIMITATIONS DU CACHE AVEC NFS

NFS présente certaines limitations au niveau de la mémoire cache :

- L'ouverture d'un fichier à partir d'un système de fichiers partagé pour une entrée/sortie directe contourne automatiquement le cache. En effet, ce type d'accès doit être direct au serveur.
- L'ouverture d'un fichier à partir d'un système de fichiers partagés pour des E/S directes ou en écriture vide la copie en cache du fichier. FS-Cache ne remettra pas le fichier en cache jusqu'à ce qu'il ne soit plus ouvert pour des entrées/sorties directes ou pour l'écriture.

- En outre, cette version de FS-Cache ne met en cache que les fichiers NFS ordinaires. FS-Cache mettra en cache *not* les répertoires, les liens symboliques, les fichiers de périphériques, les FIFO et les sockets.

CHAPITRE 11. MONTAGE D'UN PARTAGE SMB SOUS RED HAT ENTERPRISE LINUX

Le protocole Server Message Block (SMB) met en œuvre un protocole réseau de couche application utilisé pour accéder aux ressources d'un serveur, telles que les partages de fichiers et les imprimantes partagées.



NOTE

Dans le contexte de SMB, vous pouvez trouver des mentions du protocole Common Internet File System (CIFS), qui est un dialecte de SMB. Les protocoles SMB et CIFS sont tous deux pris en charge, et le module du noyau et les utilitaires impliqués dans le montage des partages SMB et CIFS utilisent tous deux le nom **cifs**.

Cette section décrit comment monter des partages à partir d'un serveur SMB.

Conditions préalables

Sous Microsoft Windows, SMB est implémenté par défaut. Sur Red Hat Enterprise Linux, le module de système de fichiers **cifs.ko** du noyau prend en charge le montage des partages SMB. Par conséquent, installez le paquetage **cifs-utils**:

```
# dnf install cifs-utils
```

Le paquetage **cifs-utils** fournit des utilitaires pour :

- Monter des partages SMB et CIFS
- Gérer les informations d'identification NT Lan Manager (NTLM) dans le trousseau de clés du noyau
- Définir et afficher des listes de contrôle d'accès (ACL) dans un descripteur de sécurité sur des partages SMB et CIFS

11.1. VERSIONS DU PROTOCOLE SMB PRISES EN CHARGE

Le module du noyau **cifs.ko** prend en charge les versions suivantes du protocole SMB :

- PME 1



AVERTISSEMENT

Le protocole SMB1 est déprécié en raison de problèmes de sécurité connus et n'est disponible qu'à l'adresse **safe to use on a private network**. La principale raison pour laquelle le protocole SMB1 est encore fourni comme option prise en charge est qu'il s'agit actuellement de la seule version du protocole SMB qui prend en charge les extensions UNIX. Si vous n'avez pas besoin d'utiliser les extensions UNIX sur SMB, Red Hat recommande fortement d'utiliser SMB2 ou une version ultérieure.

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



NOTE

Selon la version du protocole, toutes les fonctionnalités du SMB ne sont pas implémentées.

11.2. SUPPORT DES EXTENSIONS UNIX

Samba utilise le bit de capacité **CAP_UNIX** dans le protocole SMB pour fournir la fonction d'extension UNIX. Ces extensions sont également prises en charge par le module du noyau **cifs.ko**. Cependant, Samba et le module du noyau ne prennent en charge les extensions UNIX que dans le protocole SMB 1.

Pour utiliser les extensions UNIX :

1. Dans la section **[global]** du fichier `/etc/samba/smb.conf`, le paramètre **server min protocol** doit être réglé sur **NT1**.
2. Montez le partage en utilisant le protocole SMB 1 en fournissant l'option **-o vers=1.0** à la commande `mount`. Par exemple :

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

Par défaut, le module du noyau utilise SMB 2 ou la version la plus récente du protocole prise en charge par le serveur. L'ajout de l'option **-o vers=1.0** à la commande `mount` oblige le module du noyau à utiliser le protocole SMB 1 requis pour l'utilisation des extensions UNIX.

Pour vérifier si les extensions UNIX sont activées, affichez les options du partage monté :

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

Si l'entrée **unix** est affichée dans la liste des options de montage, les extensions UNIX sont activées.

11.3. MONTAGE MANUEL D'UN PARTAGE SMB

Si un partage SMB ne doit être monté que temporairement, vous pouvez le monter manuellement à l'aide de l'utilitaire **mount**.



NOTE

Les partages montés manuellement ne sont pas remontés automatiquement lorsque vous redémarrez le système. Pour configurer Red Hat Enterprise Linux de manière à ce qu'il monte automatiquement le partage au démarrage du système, reportez-vous à [Montage automatique d'un partage SMB au démarrage du système](#) .

Conditions préalables

- Le paquet **cifs-utils** est installé.

Procédure

Pour monter manuellement un partage SMB, utilisez l'utilitaire **mount** avec le paramètre **-t cifs**:

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

Dans le paramètre **-o**, vous pouvez spécifier les options utilisées pour monter le partage. Pour plus de détails, voir la section **OPTIONS** dans la page de manuel **mount.cifs(8)** et les [options de montage fréquemment utilisées](#).

Exemple 11.1. Montage d'un partage à l'aide d'une connexion SMB 3.0 cryptée

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
```

11.4. MONTAGE AUTOMATIQUE D'UN PARTAGE SMB AU DÉMARRAGE DU SYSTÈME

Si l'accès à un partage SMB monté est nécessaire en permanence sur un serveur, montez le partage automatiquement au démarrage.

Conditions préalables

- Le paquet **cifs-utils** est installé.

Procédure

Pour monter un partage SMB automatiquement lorsque le système démarre, ajoutez une entrée pour le partage au fichier `/etc/fstab`. Par exemple :

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



IMPORTANT

Pour permettre au système de monter un partage automatiquement, vous devez stocker le nom d'utilisateur, le mot de passe et le nom de domaine dans un fichier d'informations d'identification. Pour plus d'informations, voir [Authentification à un partage SMB à l'aide d'un fichier d'informations d'identification](#).

Dans le quatrième champ de la ligne du site `/etc/fstab`, indiquez les options de montage, telles que le chemin d'accès au fichier d'informations d'identification. Pour plus de détails, voir la section **OPTIONS** de la page de manuel **mount.cifs(8)** et les [options de montage fréquemment utilisées](#).

Pour vérifier que le partage est monté avec succès, entrez :

```
# mount /mnt/
```

11.5. AUTHENTIFICATION À UN PARTAGE SMB À L'AIDE D'UN FICHER D'INFORMATIONS D'IDENTIFICATION

Dans certaines situations, par exemple lors du montage automatique d'un partage au démarrage, un partage doit être monté sans saisir le nom d'utilisateur et le mot de passe. Pour ce faire, créez un fichier d'informations d'identification.

Conditions préalables

- Le paquet **cifs-utils** est installé.

Procédure

1. Créez un fichier, tel que **/root/smb.cred**, et indiquez le nom d'utilisateur, le mot de passe et le nom de domaine de ce fichier :

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

2. Définissez les autorisations de manière à ce que seul le propriétaire puisse accéder au fichier :

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

Vous pouvez maintenant transmettre l'option **credentials=*file_name*** à l'utilitaire **mount** ou l'utiliser dans le fichier **/etc/fstab** pour monter le partage sans être invité à fournir le nom d'utilisateur et le mot de passe.

11.6. OPTIONS DE MONTAGE FRÉQUEMMENT UTILISÉES

Lorsque vous montez un partage SMB, les options de montage déterminent :

- Comment la connexion sera établie avec le serveur. Par exemple, quelle version du protocole SMB est utilisée lors de la connexion au serveur.
- Comment le partage sera monté dans le système de fichiers local. Par exemple, si le système remplace les autorisations de fichiers et de répertoires distants pour permettre à plusieurs utilisateurs locaux d'accéder au contenu du serveur.

Pour définir plusieurs options dans le quatrième champ du fichier **/etc/fstab** ou dans le paramètre **-o** d'une commande de montage, séparez-les par des virgules. Par exemple, voir [Montage d'un partage avec l'option multiutilisateur](#).

La liste suivante présente les options de montage les plus fréquemment utilisées :

Option	Description
<code>références=<i>file_name</i></code>	Définit le chemin d'accès au fichier d'informations d'identification. Voir Authentification à un partage SMB à l'aide d'un fichier d'informations d'identification .

Option	Description
<code>dir_mode=mode</code>	Définit le mode répertoire si le serveur ne prend pas en charge les extensions CIFS UNIX.
<code>file_mode=mode</code>	Définit le mode de fichier si le serveur ne prend pas en charge les extensions UNIX CIFS.
<code>mot de passe=password</code>	Définit le mot de passe utilisé pour s'authentifier auprès du serveur SMB. Il est également possible de spécifier un fichier d'informations d'identification à l'aide de l'option credentials .
<code>sceller</code>	Active la prise en charge du cryptage pour les connexions utilisant SMB 3.0 ou une version ultérieure du protocole. Par conséquent, utilisez seal avec l'option de montage vers réglée sur 3.0 ou une version ultérieure. Voir l'exemple de montage manuel d'un partage SMB .
<code>sec=security_mode</code>	Définit le mode de sécurité, tel que ntlmsspi , pour activer le hachage de mot de passe NTLMv2 et la signature de paquets activée. Pour une liste des valeurs prises en charge, voir la description de l'option dans la page de manuel mount.cifs(8) . Si le serveur ne prend pas en charge le mode de sécurité ntlmv2 , utilisez sec=ntlmssp , qui est le mode par défaut. Pour des raisons de sécurité, n'utilisez pas le mode de sécurité non sécurisé ntlm .
<code>nom d'utilisateur=user_name</code>	Définit le nom d'utilisateur utilisé pour s'authentifier auprès du serveur SMB. Il est également possible de spécifier un fichier d'informations d'identification à l'aide de l'option credentials .
<code>vers=SMB_protocol_version</code>	Définit la version du protocole SMB utilisée pour la communication avec le serveur.

Pour une liste complète, voir la section **OPTIONS** dans la page de manuel **mount.cifs(8)**.

CHAPITRE 12. EXÉCUTION D'UN MONTAGE SMB MULTI-UTILISATEURS

Les informations d'identification que vous fournissez pour monter un partage déterminent par défaut les autorisations d'accès au point de montage. Par exemple, si vous utilisez l'utilisateur **DOMAINexample** lorsque vous montez un partage, toutes les opérations sur le partage seront exécutées sous cet utilisateur, quel que soit l'utilisateur local qui effectue l'opération.

Toutefois, dans certaines situations, l'administrateur souhaite monter un partage automatiquement au démarrage du système, mais les utilisateurs doivent effectuer des actions sur le contenu du partage à l'aide de leurs propres informations d'identification. Les options de montage de **multiuser** vous permettent de configurer ce scénario.



IMPORTANT

Pour utiliser l'option de montage **multiuser**, vous devez également définir l'option de montage **sec** sur un type de sécurité qui prend en charge la fourniture d'informations d'identification de manière non interactive, comme **krb5** ou l'option **ntlmssp** avec un fichier d'informations d'identification. Pour plus d'informations, voir [Accès à un partage en tant qu'utilisateur](#).

L'utilisateur **root** monte le partage en utilisant l'option **multiuser** et un compte ayant un accès minimal au contenu du partage. Les utilisateurs ordinaires peuvent ensuite fournir leur nom d'utilisateur et leur mot de passe au trousseau de clés du noyau de la session en cours à l'aide de l'utilitaire **cifscreds**. Si l'utilisateur accède au contenu du partage monté, le noyau utilise les informations d'identification du trousseau du noyau au lieu de celles utilisées initialement pour monter le partage.

L'utilisation de cette fonction comprend les étapes suivantes :

- [Monter un partage avec l'option **multiuser**](#) .
- [Optionnellement, vérifiez si le partage a été monté avec succès avec l'option **multiuser**](#) .
- [Accédez au partage en tant qu'utilisateur](#) .

Conditions préalables

- Le paquet **cifs-utils** est installé.

12.1. MONTAGE D'UN PARTAGE AVEC L'OPTION MULTIUTILISATEUR

Avant que les utilisateurs puissent accéder au partage avec leurs propres informations d'identification, montez le partage en tant qu'utilisateur **root** à l'aide d'un compte aux autorisations limitées.

Procédure

Pour monter un partage automatiquement avec l'option **multiuser** lorsque le système démarre :

1. Créez l'entrée pour le partage dans le fichier **/etc/fstab**. Par exemple :

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

2. Monter le partage :

```
# mount /mnt/
```

Si vous ne souhaitez pas monter le partage automatiquement au démarrage du système, montez-le manuellement en passant **-o multiuser,sec=security_type** à la commande **mount**. Pour plus d'informations sur le montage manuel d'un partage SMB, voir [Montage manuel d'un partage SMB](#).

12.2. VÉRIFIER SI UN PARTAGE SMB EST MONTÉ AVEC L'OPTION MULTIUTILISATEUR

Pour vérifier si un partage est monté avec l'option **multiuser**, affichez les options de montage.

Procédure

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

Si l'entrée **multiuser** est affichée dans la liste des options de montage, la fonctionnalité est activée.

12.3. ACCÈS À UN PARTAGE EN TANT QU'UTILISATEUR

Si un partage SMB est monté avec l'option **multiuser**, les utilisateurs peuvent fournir leurs informations d'identification pour le serveur au trousseau de clés du noyau :

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

Lorsque l'utilisateur effectue des opérations dans le répertoire qui contient le partage SMB monté, le serveur applique les autorisations du système de fichiers pour cet utilisateur, au lieu de celles utilisées initialement lorsque le partage a été monté.



NOTE

Plusieurs utilisateurs peuvent effectuer simultanément des opérations sur le partage monté à l'aide de leurs propres informations d'identification.

CHAPITRE 13. VUE D'ENSEMBLE DES ATTRIBUTS DE DÉNOMINATION PERSISTANTS

En tant qu'administrateur système, vous devez faire référence aux volumes de stockage à l'aide d'attributs de dénomination persistants afin de créer des configurations de stockage fiables après plusieurs démarrages du système.

13.1. INCONVÉNIENTS DES ATTRIBUTS DE DÉNOMINATION NON PERSISTANTS

Red Hat Enterprise Linux propose un certain nombre de méthodes pour identifier les périphériques de stockage. Il est important d'utiliser la bonne option pour identifier chaque périphérique lorsqu'il est utilisé afin d'éviter d'accéder par inadvertance au mauvais périphérique, en particulier lors de l'installation ou du reformatage des disques.

Traditionnellement, les noms non persistants sous la forme de `/dev/sd(major number)(minor number)` sont utilisés sous Linux pour désigner les périphériques de stockage. La plage de numéros majeurs et mineurs et les noms **sd** associés sont attribués à chaque périphérique lorsqu'il est détecté. Cela signifie que l'association entre la plage de numéros majeurs et mineurs et les noms **sd** associés peut changer si l'ordre de détection des périphériques change.

Une telle modification de l'ordre peut se produire dans les situations suivantes :

- La parallélisation du processus de démarrage du système détecte les périphériques de stockage dans un ordre différent à chaque démarrage du système.
- Un disque ne s'allume pas ou ne répond pas au contrôleur SCSI. Il n'est donc pas détecté par la sonde normale des périphériques. Le disque n'est pas accessible au système et les périphériques suivants verront leur plage de numéros majeurs et mineurs, y compris les noms **sd** associés, décalés vers le bas. Par exemple, si un disque normalement appelé **sdb** n'est pas détecté, un disque normalement appelé **sdc** apparaîtra à la place comme **sdb**.
- Un contrôleur SCSI (adaptateur de bus hôte, ou HBA) ne parvient pas à s'initialiser, ce qui entraîne la non-détection de tous les disques connectés à ce HBA. Tous les disques connectés aux adaptateurs de bus hôte détectés par la suite se voient attribuer des plages de numéros majeurs et mineurs différentes, ainsi que des noms **sd** différents.
- L'ordre d'initialisation du pilote change si différents types de HBA sont présents dans le système. Les disques connectés à ces HBA sont alors détectés dans un ordre différent. Cela peut également se produire si les HBA sont déplacés vers différents emplacements PCI sur le système.
- Les disques connectés au système avec des adaptateurs Fibre Channel, iSCSI ou FCoE peuvent être inaccessibles au moment où les périphériques de stockage sont sondés, en raison de la mise hors tension d'une baie de stockage ou d'un commutateur intermédiaire, par exemple. Cela peut se produire lors du redémarrage d'un système après une panne de courant, si la baie de stockage met plus de temps à se connecter que le système à démarrer. Bien que certains pilotes Fibre Channel prennent en charge un mécanisme permettant de spécifier une correspondance persistante entre l'ID de cible SCSI et le WWPN, cela n'entraîne pas la réservation des plages de numéros majeurs et mineurs, ni des noms **sd** associés ; cela permet uniquement d'obtenir des numéros d'ID de cible SCSI cohérents.

Pour ces raisons, il n'est pas souhaitable d'utiliser la série de numéros majeurs et mineurs ou les noms **sd** associés lorsqu'on se réfère à des périphériques, comme dans le fichier `/etc/fstab`. Il est possible que le mauvais périphérique soit monté et que des données soient corrompues.

Cependant, il est parfois nécessaire de se référer aux noms **sd** même lorsqu'un autre mécanisme est utilisé, par exemple lorsque des erreurs sont signalées par un périphérique. En effet, le noyau Linux utilise les noms **sd** (ainsi que les tuples SCSI host/channel/target/LUN) dans les messages du noyau concernant le périphérique.

13.2. SYSTÈME DE FICHIERS ET IDENTIFICATEURS DE PÉRIPHÉRIQUES

Cette section explique la différence entre les attributs persistants qui identifient les systèmes de fichiers et les périphériques en mode bloc.

Identifiants du système de fichiers

Les identifiants de système de fichiers sont liés à un système de fichiers particulier créé sur un périphérique en mode bloc. L'identifiant est également stocké dans le système de fichiers. Si vous copiez le système de fichiers sur un autre périphérique, il porte toujours le même identifiant de système de fichiers. En revanche, si vous réécrivez le périphérique, par exemple en le formatant à l'aide de l'utilitaire **mkfs**, le périphérique perd l'attribut.

Les identifiants du système de fichiers sont les suivants

- Identifiant unique (UUID)
- Étiquette

Identifiants des appareils

Les identificateurs de périphérique sont liés à un périphérique en bloc : par exemple, un disque ou une partition. Si vous réécrivez le périphérique, par exemple en le formatant à l'aide de l'utilitaire **mkfs**, le périphérique conserve l'attribut, car il n'est pas stocké dans le système de fichiers.

Les identifiants des appareils sont les suivants

- Identifiant mondial (WWID)
- UUID de la partition
- Numéro de série

Recommandations

- Certains systèmes de fichiers, tels que les volumes logiques, couvrent plusieurs périphériques. Red Hat recommande d'accéder à ces systèmes de fichiers en utilisant des identifiants de systèmes de fichiers plutôt que des identifiants de périphériques.

13.3. NOMS DES PÉRIPHÉRIQUES GÉRÉS PAR LE MÉCANISME UDEV DANS /DEV/DISK/

Le mécanisme **udev** est utilisé pour tous les types de périphériques sous Linux, et n'est pas limité aux seuls périphériques de stockage. Il fournit différents types d'attributs de noms persistants dans le répertoire **/dev/disk/**. Dans le cas des périphériques de stockage, Red Hat Enterprise Linux contient des règles **udev** qui créent des liens symboliques dans le répertoire **/dev/disk/**. Cela vous permet de faire référence aux périphériques de stockage par :

- Leur contenu
- Un identifiant unique

- Leur numéro de série.

Bien que les attributs de dénomination de **udev** soient persistants, c'est-à-dire qu'ils ne changent pas d'eux-mêmes lors des redémarrages du système, certains d'entre eux sont également configurables.

13.3.1. Identifiants du système de fichiers

L'attribut UUID dans `/dev/disk/by-uuid/`

Les entrées de ce répertoire fournissent un nom symbolique qui fait référence au périphérique de stockage par un **unique identifiant** (UUID) dans le contenu (c'est-à-dire les données) stocké sur le périphérique. Par exemple :

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

Vous pouvez utiliser l'UUID pour faire référence à l'appareil dans le fichier `/etc/fstab` en utilisant la syntaxe suivante :

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

Vous pouvez configurer l'attribut UUID lors de la création d'un système de fichiers et vous pouvez également le modifier ultérieurement.

L'attribut Label dans `/dev/disk/by-label/`

Les entrées de ce répertoire fournissent un nom symbolique qui renvoie à l'unité de stockage par une adresse **label** dans le contenu (c'est-à-dire les données) stocké sur l'unité.

Par exemple :

```
/dev/disk/by-label/Boot
```

Vous pouvez utiliser l'étiquette pour faire référence à l'appareil dans le fichier `/etc/fstab` en utilisant la syntaxe suivante :

```
LABEL=Boot
```

Vous pouvez configurer l'attribut Label lors de la création d'un système de fichiers et vous pouvez également le modifier ultérieurement.

13.3.2. Identifiants des appareils

L'attribut WWID dans `/dev/disk/by-id/`

Le World Wide Identifier (WWID) est un identifiant persistant, **system-independent identifiant**, que la norme SCSI exige de tous les périphériques SCSI. L'identifiant WWID est garanti unique pour chaque périphérique de stockage et indépendant du chemin utilisé pour accéder au périphérique. L'identifiant est une propriété du périphérique mais n'est pas stocké dans le contenu (c'est-à-dire les données) des périphériques.

Cet identifiant peut être obtenu en lançant une requête SCSI pour récupérer les données vitales d'identification du produit (page **0x83**) ou le numéro de série de l'unité (page **0x80**).

Red Hat Enterprise Linux maintient automatiquement le mappage approprié du nom de périphérique basé sur le WWID à un nom `/dev/sd` actuel sur ce système. Les applications peuvent utiliser le nom `/dev/disk/by-id/` pour référencer les données sur le disque, même si le chemin d'accès au périphérique change, et même en accédant au périphérique à partir de différents systèmes.



NOTE

Si vous utilisez un périphérique NVMe, vous pourriez être confronté à un changement de nom du disque by-id pour certains fournisseurs, si le numéro de série de votre périphérique comporte des espaces blancs.

Exemple 13.1. Correspondances WWID

Lien symbolique WWID	Dispositif non persistant	Note
/dev/disk/by-id/scsi-3600508b400105e210000900000490000	/dev/sda	Un appareil avec un identifiant de page 0x83
/dev/disk/by-id/scsi-SSEAGATE_ST373453LW_3HW1RHM6	/dev/sdb	Un appareil avec un identifiant de page 0x80
/dev/disk/by-id/ata-SAMSUNG_MZNLN256MHQ-000L7_S2WDNX0J336519-part3	/dev/sdc3	Une partition de disque

Outre ces noms persistants fournis par le système, vous pouvez également utiliser les règles **udev** pour mettre en œuvre vos propres noms persistants, associés au WWID du stockage.

L'attribut UUID de la partition dans /dev/disk/by-partuuid

L'attribut Partition UUID (PARTUUID) identifie les partitions telles que définies par la table de partition GPT.

Exemple 13.2. Mappages d'UUID de partition

Lien symbolique PARTUUID	Dispositif non persistant
/dev/disk/by-partuuid/4cd1448a-01	/dev/sda1
/dev/disk/by-partuuid/4cd1448a-02	/dev/sda2
/dev/disk/by-partuuid/4cd1448a-03	/dev/sda3

L'attribut Path dans le fichier /dev/disk/by-path/

Cet attribut fournit un nom symbolique qui fait référence à l'unité de stockage par l'adresse **hardware path** utilisée pour accéder à l'unité.

L'attribut Path échoue si une partie du chemin matériel (par exemple, l'ID PCI, le port cible ou le numéro LUN) change. L'attribut Path n'est donc pas fiable. Cependant, l'attribut Path peut être utile dans l'un des scénarios suivants :

- Vous devez identifier un disque que vous prévoyez de remplacer ultérieurement.
- Vous prévoyez d'installer un service de stockage sur un disque à un emplacement spécifique.

13.4. LE WORLD WIDE IDENTIFIER AVEC DM MULTIPATH

Vous pouvez configurer Device Mapper (DM) Multipath pour établir une correspondance entre le World Wide Identifier (WWID) et les noms de périphériques non persistants.

S'il existe plusieurs chemins entre un système et un appareil, DM Multipath utilise le WWID pour le détecter. DM Multipath présente alors un seul "pseudo-dispositif" dans le répertoire `/dev/mapper/wwid`, par exemple `/dev/mapper/3600508b400105df70000e00000ac0000`.

La commande `multipath -l` montre la correspondance avec les identifiants non persistants :

- **Host:Channel:Target:LUN**
- `/dev/sd` nom
- **major:minor** nombre

Exemple 13.3. Mappages WWID dans une configuration à chemins multiples

Exemple de sortie de la commande `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 maintient automatiquement la correspondance entre chaque nom de périphérique basé sur le WWID et le nom `/dev/sd` correspondant sur le système. Ces noms persistent lors des changements de chemin d'accès et sont cohérents lorsque l'on accède à l'appareil à partir de différents systèmes.

Lorsque la fonction `user_friendly_names` de DM Multipath est utilisée, le WWID est associé à un nom de la forme suivante `/dev/mapper/mpathN`. Par défaut, cette correspondance est conservée dans le fichier `/etc/multipath/bindings`. Ces noms `mpathN` ces noms sont persistants tant que ce fichier est maintenu.



IMPORTANT

Si vous utilisez `user_friendly_names`, des étapes supplémentaires sont nécessaires pour obtenir des noms cohérents dans un cluster.

13.5. LIMITES DE LA CONVENTION DE DÉNOMINATION DES PÉRIPHÉRIQUES UDEV

Voici quelques limitations de la convention d'appellation `udev`:

- Il est possible que le dispositif ne soit pas accessible au moment où la requête est effectuée car le mécanisme **udev** peut s'appuyer sur la capacité d'interroger le dispositif de stockage lorsque les règles **udev** sont traitées pour un événement **udev**. Cette situation est plus susceptible de se produire avec les dispositifs de stockage Fibre Channel, iSCSI ou FCoE lorsque le dispositif n'est pas situé dans le châssis du serveur.
- Le noyau peut envoyer des événements **udev** à tout moment, ce qui entraîne le traitement des règles et éventuellement la suppression des liens **/dev/disk/by-*/** si le périphérique n'est pas accessible.
- Il peut y avoir un délai entre le moment où l'événement **udev** est généré et le moment où il est traité, par exemple lorsqu'un grand nombre de périphériques sont détectés et que le service **udev** en espace utilisateur prend un certain temps pour traiter les règles pour chacun d'entre eux. Cela peut entraîner un délai entre le moment où le noyau détecte le périphérique et le moment où les noms **/dev/disk/by-*/** sont disponibles.
- Les programmes externes tels que **blkid** invoqués par les règles peuvent ouvrir le dispositif pendant une brève période de temps, rendant le dispositif inaccessible pour d'autres utilisations.
- Les noms de périphériques gérés par le mécanisme **udev** dans **/dev/disk/** peuvent changer entre les versions majeures, ce qui vous oblige à mettre à jour les liens.

13.6. LISTE DES ATTRIBUTS DE NOMS PERSISTANTS

Cette procédure décrit comment trouver les attributs de dénomination persistants des périphériques de stockage non persistants.

Procédure

- Pour dresser la liste des attributs UUID et Label, utilisez l'utilitaire **lsblk**:

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

Par exemple :

Exemple 13.4. Affichage de l'UUID et de l'étiquette d'un système de fichiers

```
$ lsblk --fs /dev/sda1
NAME FSTYPE LABEL UUID                                MOUNTPOINT
sda1 xfs   Boot afa5d5e3-9050-48c3-acc1-bb30095f3dc4 /boot
```

- Pour répertorier l'attribut PARTUUID, utilisez l'utilitaire **lsblk** avec l'option **--output PARTUUID**:

```
$ lsblk --output +PARTUUID
```

Par exemple :

Exemple 13.5. Affichage de l'attribut PARTUUID d'une partition

```
$ lsblk --output +PARTUUID /dev/sda1
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT PARTUUID
```

```
sda1 8:1 0 512M 0 part /boot 4cd1448a-01
```

- Pour répertorier l'attribut WWID, examinez les cibles des liens symboliques dans le répertoire `/dev/disk/by-id/`. Par exemple :

Exemple 13.6. Affichage du WWID de tous les périphériques de stockage du système

```
$ 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-
QIWtEHtXGobe5bewIIUDivKOz5ofkgFhP0RMFsNyySVihqEI2cWWbR7MjXJoID6g
symbolic link to ../../dm-1
/dev/disk/by-id/dm-uuid-LVM-
QIWtEHtXGobe5bewIIUDivKOz5ofkgFhXqH2M45hD2H9nAf2qfWSrIRLhzfMyOKd
symbolic link to ../../dm-0
/dev/disk/by-id/lvm-pv-uuid-atlr2Y-vuMo-ueoH-CpMG-4JuH-AhEF-wu4QQm
symbolic link to ../../sda2
```

13.7. MODIFICATION DES ATTRIBUTS DE DÉNOMINATION PERSISTANTS

Cette procédure décrit comment modifier l'UUID ou l'attribut de dénomination permanent Label d'un système de fichiers.



NOTE

La modification des attributs de **udev** s'effectue en arrière-plan et peut prendre beaucoup de temps. La commande **udevadm settle** attend que la modification soit entièrement enregistrée, ce qui garantit que votre prochaine commande pourra utiliser le nouvel attribut correctement.

Dans les commandes suivantes :

- Remplacez *new-uuid* par l'UUID que vous souhaitez définir ; par exemple, **1cdfbc07-1c90-4984-b5ec-f61943f5ea50**. Vous pouvez générer un UUID à l'aide de la commande **uuidgen**.
- Remplacer *new-label* par une étiquette ; par exemple, **backup_data**.

Conditions préalables

- Si vous modifiez les attributs d'un système de fichiers XFS, démontez-le d'abord.

Procédure

- Pour modifier les attributs UUID ou Label d'un système de fichiers **XFS**, utilisez l'utilitaire **xfs_admin**:

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

- Pour modifier les attributs UUID ou Label d'un système de fichiers **ext4**, **ext3** ou **ext2**, utilisez l'utilitaire **tune2fs**:

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

- Pour modifier les attributs UUID ou Label d'un volume d'échange, utilisez l'utilitaire **swaplabel**:

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

CHAPITRE 14. OPÉRATIONS DE PARTITIONNEMENT AVEC PARTED

parted est un programme qui permet de manipuler les partitions des disques. Il prend en charge plusieurs formats de table de partition, y compris MS-DOS et GPT. Il est utile pour créer de l'espace pour de nouveaux systèmes d'exploitation, réorganiser l'utilisation du disque et copier des données sur de nouveaux disques durs.

14.1. AFFICHAGE DE LA TABLE DE PARTITION AVEC PARTED

Affichez la table de partition d'un périphérique de bloc pour voir la disposition des partitions et des détails sur les partitions individuelles. Vous pouvez afficher la table de partition d'un périphérique de bloc à l'aide de l'utilitaire **parted**.

Procédure

1. Lancez l'utilitaire **parted**. Par exemple, la sortie suivante répertorie le périphérique **/dev/sda**:

```
# parted /dev/sda
```

2. Affichez la table de partition :

```
# (parted) 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 xfs          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
```

3. Facultatif : Passez à l'appareil que vous souhaitez examiner ensuite :

```
# (parted) select block-device
```

Pour une description détaillée de la sortie de la commande d'impression, voir ce qui suit :

Model: ATA SAMSUNG MZNLN256 (scsi)

Le type de disque, le fabricant, le numéro de modèle et l'interface.

Disk /dev/sda: 256GB

Le chemin d'accès au périphérique de bloc et la capacité de stockage.

Partition Table: msdos

Le type d'étiquette du disque.

Number

Le numéro de la partition. Par exemple, la partition portant le numéro de mineur 1 correspond à **/dev/sda1**.

Start et End

L'emplacement sur l'appareil où la partition commence et se termine.

Type

Les types valides sont les suivants : métadonnées, libre, primaire, étendu ou logique.

File system

Le type de système de fichiers. Si le champ **File system** d'un périphérique n'affiche aucune valeur, cela signifie que son type de système de fichiers est inconnu. L'utilitaire **parted** ne peut pas reconnaître le système de fichiers sur les périphériques cryptés.

Flags

Liste les drapeaux définis pour la partition. Les drapeaux disponibles sont **boot**, **root**, **swap**, **hidden**, **raid**, **lvm** ou **lba**.

Ressources supplémentaires

- **parted(8)** man page.

14.2. CRÉATION D'UNE TABLE DE PARTITION SUR UN DISQUE AVEC PARTED

Utilisez l'utilitaire **parted** pour formater plus facilement un périphérique en mode bloc avec une table de partition.



AVERTISSEMENT

Le formatage d'un périphérique bloc avec une table de partition supprime toutes les données stockées sur le périphérique.

Procédure

1. Démarrer l'interpréteur de commandes interactif **parted**:

```
# parted block-device
```

2. Déterminez s'il existe déjà une table de partition sur le périphérique :

```
# (parted) print
```

Si l'appareil contient déjà des partitions, celles-ci seront supprimées au cours des étapes suivantes.

3. Créez la nouvelle table de partition :

```
# (parted) mklabel table-type
```

- Remplacez *table-type* par le type de table de partition prévu :
 - **msdos** pour MBR
 - **gpt** pour GPT

Exemple 14.1. Création d'une table de partition GUID (GPT)

Pour créer une table GPT sur le disque, utilisez la commande suivante

```
# (parted) mklabel gpt
```

Les modifications commencent à s'appliquer après l'entrée de cette commande.

4. Affichez la table de partition pour confirmer qu'elle est créée :

```
# (parted) print
```

5. Quitter le shell **parted**:

```
# (parted) quit
```

Ressources supplémentaires

- **parted(8)** man page.

14.3. CRÉATION D'UNE PARTITION AVEC PARTED

En tant qu'administrateur système, vous pouvez créer de nouvelles partitions sur un disque à l'aide de l'utilitaire **parted**.



NOTE

Les partitions requises sont **swap**, **/boot/**, et **/ (root)**.

Conditions préalables

- Une table de partition sur le disque.
- Si la partition que vous souhaitez créer est supérieure à 2 To, formatez le disque à l'aide de la commande **GUID Partition Table (GPT)**

Procédure

1. Lancez l'utilitaire **parted**:

```
# parted block-device
```

2. Affichez la table de partition actuelle pour déterminer s'il y a suffisamment d'espace libre :

```
# (parted) print
```

- Redimensionnez la partition si l'espace libre est insuffisant.
- À partir de la table de partition, déterminez
 - Les points de départ et d'arrivée de la nouvelle partition.
 - Dans le cas du MBR, quel type de partition doit être utilisé.

3. Créez la nouvelle partition :

```
# (parted) mkpart part-type name fs-type start end
```

- Remplacez *part-type* par **primary**, **logical** ou **extended**. Cela ne s'applique qu'à la table de partition MBR.
- Remplacez *name* par un nom de partition arbitraire. Cela est nécessaire pour les tables de partition GPT.
- Remplacez *fs-type* par **xfs**, **ext2**, **ext3**, **ext4**, **fat16**, **fat32**, **hfs**, **hfs**, **linux-swaps**, **ntfs**, ou **reiserfs**. Le paramètre *fs-type* est facultatif. Notez que l'utilitaire **parted** ne crée pas le système de fichiers sur la partition.
- Remplacez *start* et *end* par les tailles qui déterminent les points de départ et d'arrivée de la partition, en comptant à partir du début du disque. Vous pouvez utiliser des suffixes de taille, tels que **512MiB**, **20GiB**, ou **1.5TiB**. La taille par défaut est en mégaoctets.

Exemple 14.2. Création d'une petite partition primaire

Pour créer une partition primaire de 1024MiB à 2048MiB sur une table MBR, utilisez :

```
# (parted) mkpart primary 1024MiB 2048MiB
```

Les modifications commencent à s'appliquer après la saisie de la commande.

4. Affichez la table de partition pour confirmer que la partition créée se trouve dans la table de partition avec le type de partition, le type de système de fichiers et la taille corrects :

```
# (parted) print
```

5. Quitter le shell **parted**:

```
# (parted) quit
```

6. Enregistrer le nouveau nœud de l'appareil :

```
# udevadm settle
```

7. Vérifiez que le noyau reconnaît la nouvelle partition :

```
# cat /proc/partitions
```

- **parted(8)** man page.
- [Création d'une table de partition sur un disque avec parted](#) .
- [Redimensionnement d'une partition avec parted](#)

14.4. SUPPRESSION D'UNE PARTITION AVEC PARTED

À l'aide de l'utilitaire **parted**, vous pouvez supprimer une partition de disque pour libérer de l'espace disque.



AVERTISSEMENT

La suppression d'une partition efface toutes les données qui y sont stockées.

Procédure

1. Démarrer l'interpréteur de commandes interactif **parted**:

```
# parted block-device
```

- Remplacez *block-device* par le chemin d'accès au périphérique sur lequel vous souhaitez supprimer une partition : par exemple, **/dev/sda**.

2. Affichez la table de partition actuelle pour déterminer le numéro mineur de la partition à supprimer :

```
(parted) print
```

3. Retirer la partition :

```
(paré) rm minor-number
```

- Remplacez *minor-number* par le numéro mineur de la partition que vous souhaitez supprimer.

Les modifications commencent à s'appliquer dès que vous entrez cette commande.

4. Vérifiez que vous avez supprimé la partition de la table de partition :

```
(parted) print
```

5. Quitter le shell **parted**:

```
(parted) quit
```

6. Vérifiez que le noyau enregistre la suppression de la partition :

```
# cat /proc/partitions
```

7. Supprimez la partition du fichier **/etc/fstab**, si elle est présente. Trouvez la ligne qui déclare la partition supprimée et supprimez-la du fichier.
8. Régénérez les unités de montage pour que votre système enregistre la nouvelle configuration **/etc/fstab**:

```
# systemctl daemon-reload
```

9. Si vous avez supprimé une partition d'échange ou des éléments de LVM, supprimez toutes les références à la partition dans la ligne de commande du noyau :
 - a. Liste les options actives du noyau et vérifie si l'une d'entre elles fait référence à la partition supprimée :

```
# grubby --info=ALL
```

- b. Supprime les options du noyau qui font référence à la partition supprimée :

```
# grubby --update-kernel=ALL --remove-args="option"
```

10. Pour enregistrer les changements dans le système de démarrage anticipé, reconstruisez le système de fichiers **initramfs**:

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

Ressources supplémentaires

- **parted(8)** page de manuel

14.5. REDIMENSIONNEMENT D'UNE PARTITION AVEC PARTED

À l'aide de l'utilitaire **parted**, étendez une partition pour utiliser l'espace disque inutilisé ou réduisez une partition pour utiliser sa capacité à d'autres fins.

Conditions préalables

- Sauvegardez les données avant de réduire une partition.
- Si la partition que vous souhaitez créer est supérieure à 2 To, formatez le disque à l'aide de la commande **GUID Partition Table (GPT)**
- Si vous souhaitez réduire la partition, réduisez d'abord le système de fichiers de manière à ce qu'il ne soit pas plus grand que la partition redimensionnée.



NOTE

XFS ne prend pas en charge le rétrécissement.

Procédure

1. Lancez l'utilitaire **parted**:

```
# parted block-device
```

2. Afficher la table de partition actuelle :

```
# (parted) print
```

À partir de la table de partition, déterminez

- Le numéro mineur de la partition.
- L'emplacement de la partition existante et son nouveau point d'arrivée après le redimensionnement.

3. Redimensionner la partition :

```
# (parted) resizepart 1 2GiB
```

- Remplacez *1* par le numéro mineur de la partition que vous redimensionnez.
- Remplacez *2* par la taille qui détermine le nouveau point final de la partition redimensionnée, en comptant à partir du début du disque. Vous pouvez utiliser des suffixes de taille, tels que **512MiB**, **20GiB**, ou **1.5TiB**. La taille par défaut est en mégaoctets.

4. Affichez la table de partition pour confirmer que la partition redimensionnée se trouve dans la table de partition avec la taille correcte :

```
# (parted) print
```

5. Quitter le shell **parted**:

```
# (parted) quit
```

6. Vérifiez que le noyau enregistre la nouvelle partition :

```
# cat /proc/partitions
```

7. Facultatif : si vous avez étendu la partition, étendez également le système de fichiers qu'elle contient.

Ressources supplémentaires

- **parted(8)** man page.
- [Création d'une table de partition sur un disque avec parted](#)
- [Redimensionnement d'un système de fichiers ext4](#)
- [Augmenter la taille d'un système de fichiers XFS](#)

CHAPITRE 15. STRATÉGIES DE REPARTITIONNEMENT D'UN DISQUE

Il existe différentes approches pour repartitionner un disque. Parmi celles-ci, citons

- De l'espace libre non partitionné est disponible.
- Une partition inutilisée est disponible.
- Il y a de l'espace libre dans une partition activement utilisée.



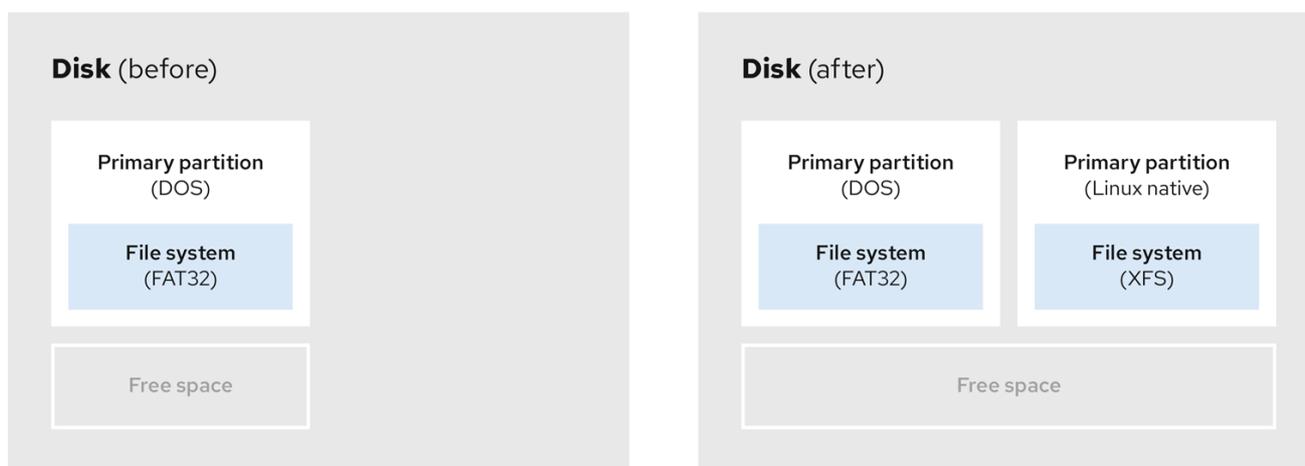
NOTE

Les exemples suivants sont simplifiés pour plus de clarté et ne reflètent pas la disposition exacte des partitions lors de l'installation de Red Hat Enterprise Linux.

15.1. UTILISATION DE L'ESPACE LIBRE NON PARTITIONNÉ

Les partitions déjà définies qui ne couvrent pas la totalité du disque dur laissent un espace non alloué qui ne fait partie d'aucune partition définie. Le diagramme suivant montre à quoi cela peut ressembler.

Figure 15.1. Disque avec espace libre non partitionné



269_RHEL_0822

Le premier diagramme représente un disque avec une partition primaire et une partition non définie avec de l'espace non alloué. Le deuxième diagramme représente un disque avec deux partitions définies avec de l'espace alloué.

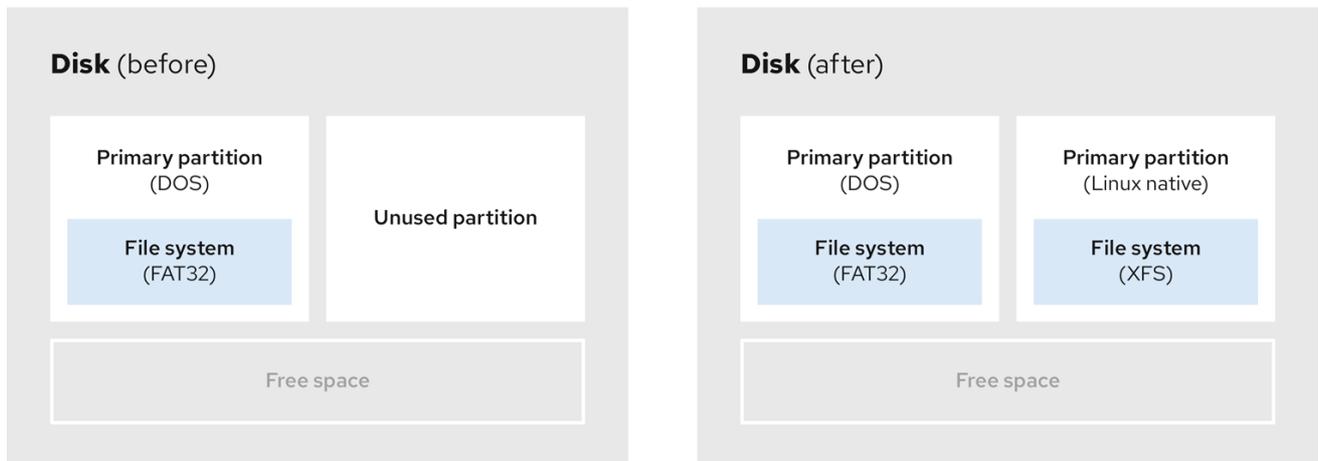
Un disque dur inutilisé entre également dans cette catégorie. La seule différence est que *all* l'espace ne fait partie d'aucune partition définie.

Sur un nouveau disque, vous pouvez créer les partitions nécessaires à partir de l'espace inutilisé. La plupart des systèmes d'exploitation préinstallés sont configurés pour occuper tout l'espace disponible sur un disque.

15.2. UTILISATION DE L'ESPACE D'UNE PARTITION INUTILISÉE

Dans l'exemple suivant, le premier diagramme représente un disque avec une partition inutilisée. Le deuxième diagramme représente la réallocation d'une partition inutilisée pour Linux.

Figure 15.2. Disque avec une partition inutilisée



269_RHEL_0822

Pour utiliser l'espace alloué à la partition inutilisée, supprimez la partition et créez la partition Linux appropriée à la place. Vous pouvez également, au cours du processus d'installation, supprimer la partition inutilisée et créer manuellement de nouvelles partitions.

15.3. UTILISATION DE L'ESPACE LIBRE D'UNE PARTITION ACTIVE

Ce processus peut être difficile à gérer car une partition active, qui est déjà utilisée, contient l'espace libre nécessaire. Dans la plupart des cas, les disques durs des ordinateurs équipés de logiciels préinstallés contiennent une partition plus grande contenant le système d'exploitation et les données.



AVERTISSEMENT

Si vous souhaitez utiliser un système d'exploitation (SE) sur une partition active, vous devez réinstaller le SE. Sachez que certains ordinateurs, qui incluent des logiciels préinstallés, ne comprennent pas de support d'installation pour réinstaller le système d'exploitation d'origine. Vérifiez si cela s'applique à votre système d'exploitation avant de détruire une partition d'origine et l'installation du système d'exploitation.

Pour optimiser l'utilisation de l'espace libre disponible, vous pouvez utiliser les méthodes de repartitionnement destructif ou non destructif.

15.3.1. Repartitionnement destructeur

Le repartitionnement destructif détruit la partition de votre disque dur et crée plusieurs partitions plus petites à la place. Sauvegardez toutes les données nécessaires de la partition d'origine, car cette méthode supprime l'intégralité du contenu.

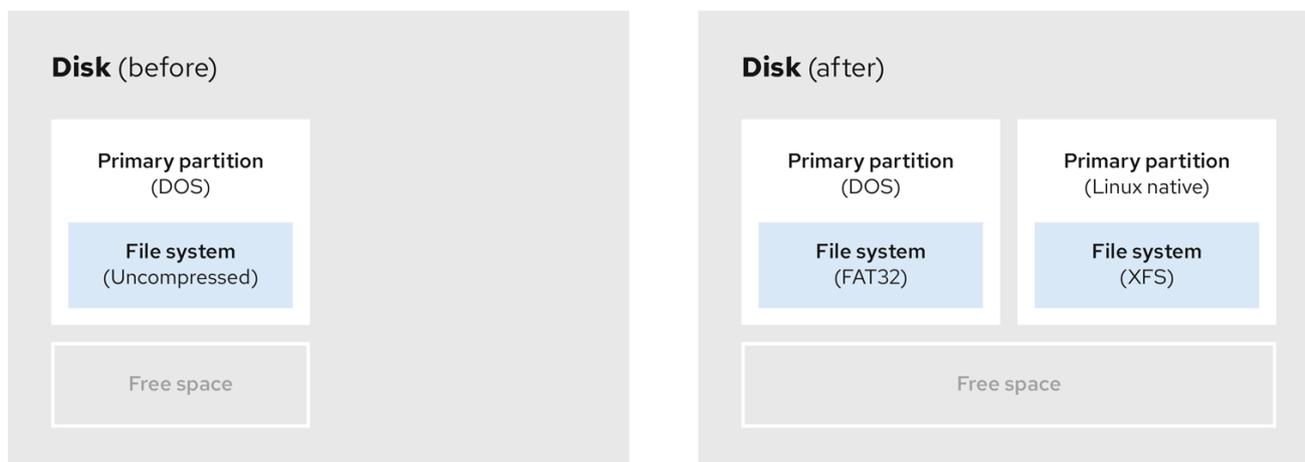
Après avoir créé une partition plus petite pour votre système d'exploitation existant, vous pouvez.. :

- Réinstaller le logiciel.

- Restaurez vos données.
- Démarrez l'installation de Red Hat Enterprise Linux.

Le diagramme suivant est une représentation simplifiée de l'utilisation de la méthode de répartition destructive.

Figure 15.3. Action de repartitionnement destructive sur le disque



269_RHEL_0822



AVERTISSEMENT

Cette méthode supprime toutes les données précédemment stockées dans la partition d'origine.

15.3.2. Repartitionnement non destructif

Le repartitionnement non destructif redimensionne les partitions, sans perte de données. Cette méthode est fiable, mais le temps de traitement est plus long pour les disques de grande taille.

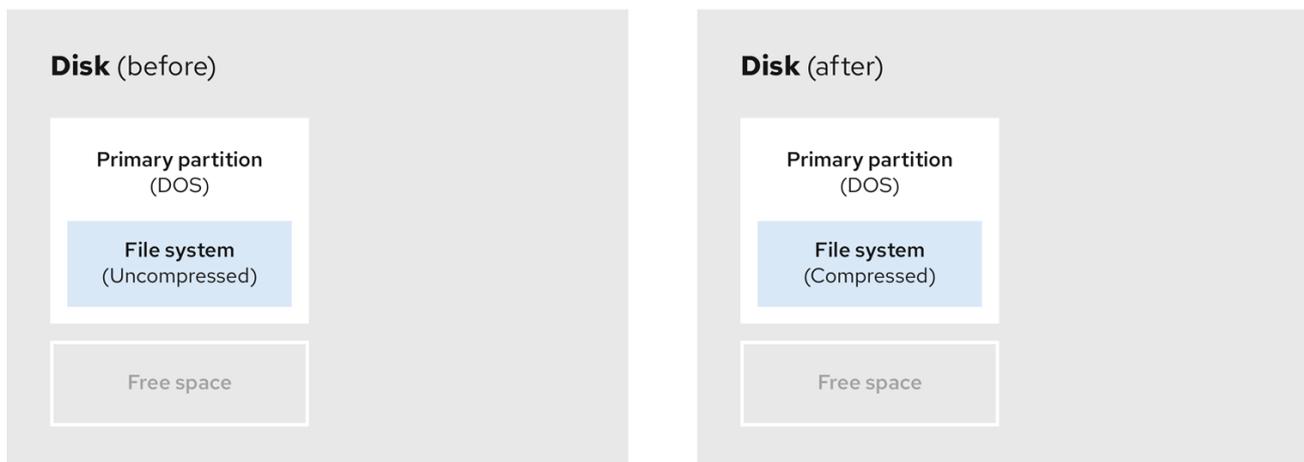
Voici une liste de méthodes qui peuvent aider à initier un repartitionnement non destructif.

- Compression des données existantes

L'emplacement de stockage de certaines données ne peut être modifié. Cela peut empêcher le redimensionnement d'une partition à la taille requise et, en fin de compte, conduire à un processus de repartitionnement destructeur. La compression des données dans une partition existante peut vous aider à redimensionner vos partitions en fonction des besoins. Elle peut également vous aider à maximiser l'espace libre disponible.

Le diagramme suivant est une représentation simplifiée de ce processus.

Figure 15.4. Compression des données sur un disque



269_RHEL_0822

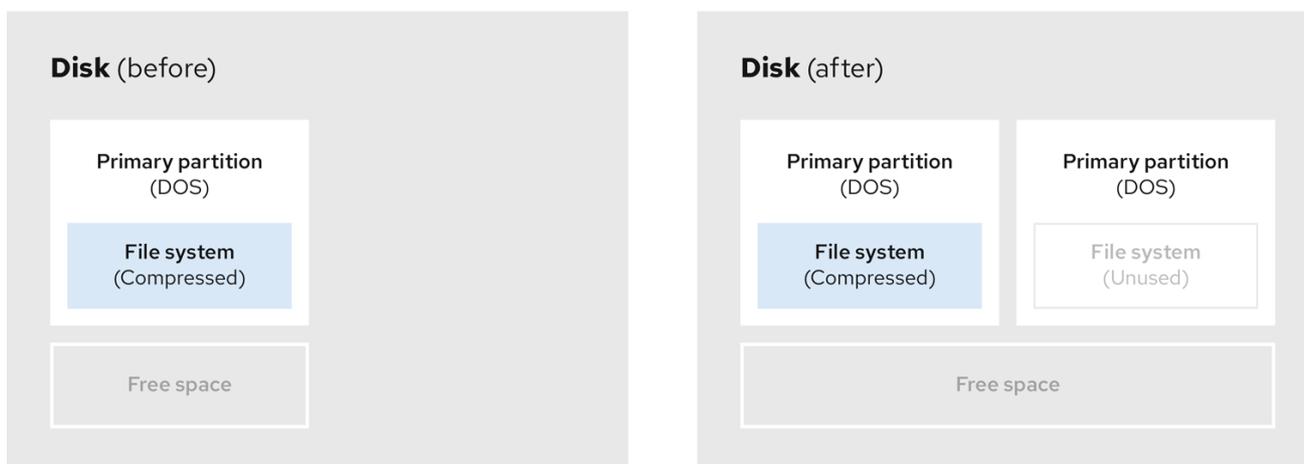
Pour éviter toute perte de données, créez une sauvegarde avant de poursuivre le processus de compression.

- Redimensionner la partition existante

En redimensionnant une partition existante, vous pouvez libérer de l'espace. Les résultats peuvent varier en fonction du logiciel de redimensionnement utilisé. Dans la majorité des cas, vous pouvez créer une nouvelle partition non formatée du même type que la partition d'origine.

Les étapes à suivre après le redimensionnement peuvent dépendre du logiciel que vous utilisez. Dans l'exemple suivant, la meilleure pratique consiste à supprimer la nouvelle partition DOS (Disk Operating System) et à créer une partition Linux à la place. Vérifiez ce qui convient le mieux à votre disque avant de lancer le processus de redimensionnement.

Figure 15.5. Redimensionnement d'une partition sur un disque



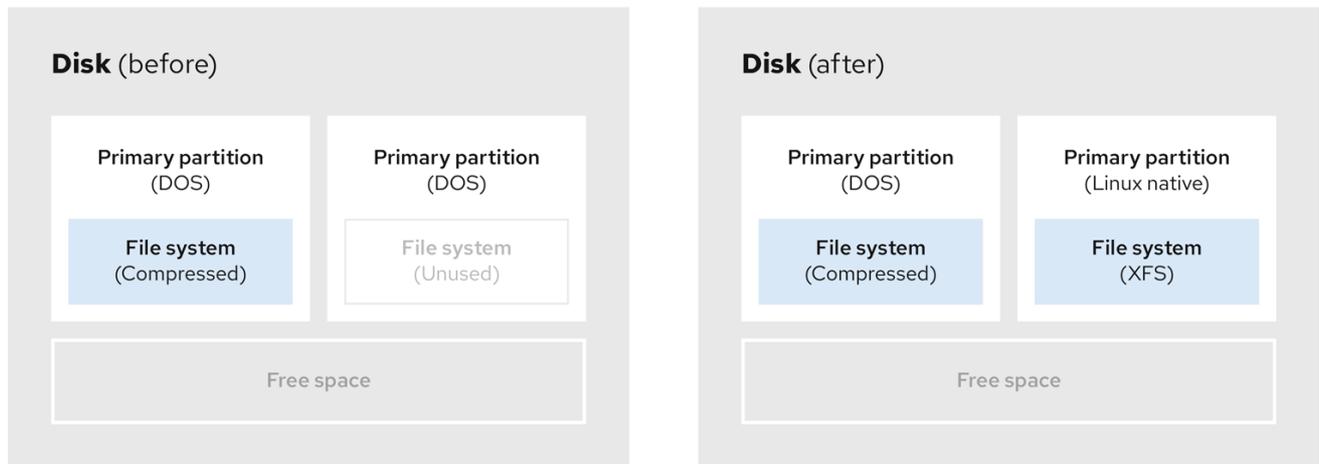
269_RHEL_0822

- Facultatif : Créer de nouvelles partitions

Certains logiciels de redimensionnement prennent en charge les systèmes basés sur Linux. Dans ce cas, il n'est pas nécessaire de supprimer la partition nouvellement créée après le redimensionnement. La création d'une nouvelle partition après le redimensionnement dépend du logiciel utilisé.

Le diagramme suivant représente l'état du disque avant et après la création d'une nouvelle partition.

Figure 15.6. Disque avec configuration finale de la partition



269_RHEL_0822

CHAPITRE 16. PREMIERS PAS AVEC XFS

Voici une vue d'ensemble de la création et de la maintenance des systèmes de fichiers XFS.

16.1. LE SYSTÈME DE FICHIERS XFS

XFS est un système de fichiers de journalisation 64 bits hautement évolutif, performant, robuste et mature qui prend en charge des fichiers et des systèmes de fichiers très volumineux sur un seul hôte. Il s'agit du système de fichiers par défaut de Red Hat Enterprise Linux 9. XFS a été développé au début des années 1990 par SGI et fonctionne depuis longtemps sur des serveurs et des baies de stockage de très grande taille.

Les caractéristiques de XFS sont les suivantes

Fiabilité

- La journalisation des métadonnées, qui garantit l'intégrité du système de fichiers après une panne du système en conservant un enregistrement des opérations du système de fichiers qui peut être rejoué lorsque le système est redémarré et que le système de fichiers est remonté
- Contrôle étendu de la cohérence des métadonnées au moment de l'exécution
- Utilitaires de réparation évolutifs et rapides
- Journalisation des quotas. Cela évite d'avoir à effectuer de longs contrôles de cohérence des quotas après une panne.

Évolutivité et performance

- Taille du système de fichiers pris en charge jusqu'à 1024 TiB
- Capacité à prendre en charge un grand nombre d'opérations simultanées
- Indexation B-tree pour l'extensibilité de la gestion de l'espace libre
- Algorithmes sophistiqués de lecture anticipée des métadonnées
- Optimisations pour les charges de travail liées à la vidéo en continu

Systèmes d'allocation

- Allocation basée sur l'étendue
- Politiques d'allocation tenant compte des bandes
- Attribution retardée
- Pré-affectation de l'espace
- Inodes alloués dynamiquement

Autres caractéristiques

- Copies de fichiers basées sur des liens hypertextes

- Utilitaires de sauvegarde et de restauration étroitement intégrés
- Défragmentation en ligne
- Croissance du système de fichiers en ligne
- Capacités de diagnostic complètes
- Attributs étendus (**xattr**). Cela permet au système d'associer plusieurs paires nom/valeur supplémentaires par fichier.
- Quotas de projet ou de répertoire. Cela permet de restreindre les quotas sur une arborescence de répertoires.
- Horodatage à la seconde près

Caractéristiques de performance

XFS est très performant sur les grands systèmes avec des charges de travail d'entreprise. Un système de grande taille est un système doté d'un nombre relativement élevé de CPU, de plusieurs HBA et de connexions à des baies de disques externes. XFS est également performant sur les systèmes plus petits qui ont une charge de travail d'E/S parallèle et multithread.

XFS est relativement peu performant pour les charges de travail à fil unique et à forte intensité de métadonnées : par exemple, une charge de travail qui crée ou supprime un grand nombre de petits fichiers en un seul fil.

16.2. COMPARAISON DES OUTILS UTILISÉS AVEC EXT4 ET XFS

Cette section compare les outils à utiliser pour accomplir des tâches courantes sur les systèmes de fichiers ext4 et XFS.

Tâche	ext4	XFS
Créer un système de fichiers	mkfs.ext4	mkfs.xfs
Vérification du système de fichiers	e2fsck	xfs_repair
Redimensionner un système de fichiers	resize2fs	xfs_growfs
Enregistrer une image d'un système de fichiers	e2image	xfs_metadump et xfs_mdrestore
Étiqueter ou régler un système de fichiers	tune2fs	xfs_admin
Sauvegarder un système de fichiers	tar et rsync	xfsdump et xfsrestore
Gestion des quotas	quota	xfs_quota

Tâche	ext4	XFS
Cartographie des fichiers	filefrag	xfs_bmap

**NOTE**

Si vous souhaitez une solution client-serveur complète pour les sauvegardes sur le réseau, vous pouvez utiliser l'utilitaire de sauvegarde **bacula** disponible dans RHEL 9. Pour plus d'informations sur Bacula, voir la [solution de sauvegarde Bacula](#).

CHAPITRE 17. CRÉATION D'UN SYSTÈME DE FICHIERS XFS

En tant qu'administrateur système, vous pouvez créer un système de fichiers XFS sur un périphérique bloc pour lui permettre de stocker des fichiers et des répertoires.

17.1. CRÉATION D'UN SYSTÈME DE FICHIERS XFS AVEC MKFS.XFS

Cette procédure décrit comment créer un système de fichiers XFS sur un périphérique en mode bloc.

Procédure

1. Pour créer le système de fichiers :

- Si le périphérique est une partition normale, un volume LVM, un volume MD, un disque ou un périphérique similaire, utilisez la commande suivante :

```
# mkfs.xfs block-device
```

- Remplacer *block-device* par le chemin d'accès au dispositif de blocage. Par exemple, `/dev/sdb1`, `/dev/disk/by-uuid/05e99ec8-def1-4a5e-8a9d-5945339ceb2a`, ou `/dev/myvolgroup/my-lv`.
- En général, les options par défaut sont optimales pour une utilisation courante.
- Lorsque vous utilisez **mkfs.xfs** sur un périphérique en mode bloc contenant un système de fichiers existant, ajoutez l'option **-f** pour écraser ce système de fichiers.
- Pour créer le système de fichiers sur un périphérique RAID matériel, vérifiez que le système détecte correctement la géométrie des bandes du périphérique :
 - Si les informations relatives à la géométrie des bandes sont correctes, aucune option supplémentaire n'est nécessaire. Créez le système de fichiers :

```
# mkfs.xfs block-device
```

- Si les informations sont incorrectes, spécifiez manuellement la géométrie de la bande avec les paramètres **su** et **sw** de l'option **-d**. Le paramètre **su** spécifie la taille du bloc RAID et le paramètre **sw** spécifie le nombre de disques de données dans le périphérique RAID.

Par exemple :

```
# mkfs.xfs -d su=64k,sw=4 /dev/sda3
```

2. Utilisez la commande suivante pour attendre que le système enregistre le nouveau nœud de dispositif :

```
# udevadm settle
```

Ressources supplémentaires

- **mkfs.xfs(8)** page de manuel.

CHAPITRE 18. SAUVEGARDE D'UN SYSTÈME DE FICHIERS XFS

En tant qu'administrateur système, vous pouvez utiliser le site **xfsdump** pour sauvegarder un système de fichiers XFS dans un fichier ou sur une bande. Il s'agit d'un mécanisme de sauvegarde simple.

18.1. CARACTÉRISTIQUES DE LA SAUVEGARDE XFS

Cette section décrit les concepts et les fonctionnalités clés de la sauvegarde d'un système de fichiers XFS à l'aide de l'utilitaire **xfsdump**.

Vous pouvez utiliser l'utilitaire **xfsdump** pour

- Effectuer des sauvegardes sur des images de fichiers régulières.
Une seule sauvegarde peut être écrite dans un fichier normal.
- Effectuer des sauvegardes sur des lecteurs de bandes.
L'utilitaire **xfsdump** vous permet également d'écrire plusieurs sauvegardes sur la même bande. Une sauvegarde peut s'étendre sur plusieurs bandes.

Pour sauvegarder plusieurs systèmes de fichiers sur une seule unité de bande, il suffit d'écrire la sauvegarde sur une bande qui contient déjà une sauvegarde XFS. La nouvelle sauvegarde est ajoutée à la précédente. Par défaut, **xfsdump** n'écrase jamais les sauvegardes existantes.

- Créer des sauvegardes incrémentielles.
L'utilitaire **xfsdump** utilise les niveaux de vidage pour déterminer une sauvegarde de base à laquelle les autres sauvegardes sont relatives. Les chiffres de 0 à 9 correspondent à des niveaux de vidage croissants. Une sauvegarde incrémentielle ne sauvegarde que les fichiers qui ont été modifiés depuis la dernière sauvegarde d'un niveau inférieur :
 - Pour effectuer une sauvegarde complète, procédez à une vidange de niveau 0 du système de fichiers.
 - Une sauvegarde de niveau 1 est la première sauvegarde incrémentielle après une sauvegarde complète. La sauvegarde incrémentielle suivante est le niveau 2, qui ne sauvegarde que les fichiers modifiés depuis la dernière sauvegarde de niveau 1, et ainsi de suite jusqu'au niveau 9.
- Exclure des fichiers d'une sauvegarde en utilisant des indicateurs de taille, de sous-arbre ou d'inode pour les filtrer.

Ressources supplémentaires

- **xfsdump(8)** page de manuel.

18.2. SAUVEGARDE D'UN SYSTÈME DE FICHIERS XFS AVEC XFSDUMP

Cette procédure décrit comment sauvegarder le contenu d'un système de fichiers XFS dans un fichier ou sur une bande.

Conditions préalables

- Un système de fichiers XFS que vous pouvez sauvegarder.

- Un autre système de fichiers ou un lecteur de bandes où vous pouvez stocker la sauvegarde.

Procédure

- Utilisez la commande suivante pour sauvegarder un système de fichiers XFS :

```
# xfsdump -l level [-L label] \  
-f backup-destination path-to-xfs-filesystem
```

- Remplacez *level* par le niveau de vidage de votre sauvegarde. Utilisez **0** pour effectuer une sauvegarde complète ou **1** à **9** pour effectuer des sauvegardes incrémentielles conséquentes.
- Remplacez *backup-destination* par le chemin d'accès où vous souhaitez stocker votre sauvegarde. La destination peut être un fichier ordinaire, un lecteur de bande ou un périphérique de bande distant. Par exemple, **/backup-files/Data.xfsdump** pour un fichier ou **/dev/st0** pour un lecteur de bande.
- Remplacez *path-to-xfs-filesystem* par le point de montage du système de fichiers XFS que vous souhaitez sauvegarder. Par exemple, **/mnt/data/**. Le système de fichiers doit être monté.
- Lorsque vous sauvegardez plusieurs systèmes de fichiers et que vous les enregistrez sur une seule unité de bande, ajoutez une étiquette de session à chaque sauvegarde à l'aide de l'option **-L *label*** afin qu'il soit plus facile de les identifier lors de la restauration. Remplacez *label* par le nom de votre sauvegarde : par exemple, **backup_data**.

Exemple 18.1. Sauvegarde de plusieurs systèmes de fichiers XFS

- Sauvegarder le contenu des systèmes de fichiers XFS montés sur les répertoires **/boot/** et **/data/** et les enregistrer sous forme de fichiers dans le répertoire **/backup-files/**:

```
# xfsdump -l 0 -f /backup-files/boot.xfsdump /boot  
# xfsdump -l 0 -f /backup-files/data.xfsdump /data
```

- Pour sauvegarder plusieurs systèmes de fichiers sur une seule unité de bande, ajoutez une étiquette de session à chaque sauvegarde à l'aide de l'option **-L *label*** à l'aide de l'option

```
# xfsdump -l 0 -L "backup_boot" -f /dev/st0 /boot  
# xfsdump -l 0 -L "backup_data" -f /dev/st0 /data
```

Ressources supplémentaires

- **xfsdump(8)** page de manuel.

18.3. RESSOURCES SUPPLÉMENTAIRES

- **xfsdump(8)** page de manuel

CHAPITRE 19. RESTAURATION D'UN SYSTÈME DE FICHIERS XFS À PARTIR D'UNE SAUVEGARDE

En tant qu'administrateur système, vous pouvez utiliser l'utilitaire **xfsrestore** pour restaurer une sauvegarde XFS créée avec l'utilitaire **xfsdump** et stockée dans un fichier ou sur une bande.

19.1. FONCTIONNALITÉS DE LA RESTAURATION DE XFS À PARTIR D'UNE SAUVEGARDE

L'utilitaire **xfsrestore** restaure les systèmes de fichiers à partir des sauvegardes produites par **xfsdump**. L'utilitaire **xfsrestore** a deux modes :

- Le mode **simple** permet aux utilisateurs de restaurer un système de fichiers entier à partir d'un dump de niveau 0. Il s'agit du mode par défaut.
- Le mode **cumulative** permet de restaurer un système de fichiers à partir d'une sauvegarde incrémentale, c'est-à-dire du niveau 1 au niveau 9.

Une adresse unique *session ID* ou *session label* identifie chaque sauvegarde. La restauration d'une sauvegarde à partir d'une bande contenant plusieurs sauvegardes nécessite l'ID de session ou l'étiquette correspondante.

Pour extraire, ajouter ou supprimer des fichiers spécifiques d'une sauvegarde, entrez dans le mode interactif de **xfsrestore**. Le mode interactif fournit un ensemble de commandes permettant de manipuler les fichiers de sauvegarde.

Ressources supplémentaires

- **xfsrestore(8)** page de manuel.

19.2. RESTAURATION D'UN SYSTÈME DE FICHIERS XFS À PARTIR D'UNE SAUVEGARDE AVEC XFSRESTORE

Cette procédure décrit comment restaurer le contenu d'un système de fichiers XFS à partir d'un fichier ou d'une bande de sauvegarde.

Conditions préalables

- Sauvegarde de fichiers ou de bandes de systèmes de fichiers XFS, comme décrit dans [Sauvegarde d'un système de fichiers XFS](#).
- Périphérique de stockage où vous pouvez restaurer la sauvegarde.

Procédure

- La commande de restauration de la sauvegarde varie selon que vous restaurez une sauvegarde complète ou incrémentielle, ou que vous restaurez plusieurs sauvegardes à partir d'une seule unité de bande :

```
# xfsrestore [-r] [-S session-id] [-L session-label] [-i]
             -f backup-location restoration-path
```

- Remplacer *backup-location* par l'emplacement de la sauvegarde. Il peut s'agir d'un fichier ordinaire, d'un lecteur de bande ou d'un périphérique de bande distant. Par exemple, **/backup-files/Data.xfsdump** pour un fichier ou **/dev/st0** pour un lecteur de bande.
- Remplacez *restoration-path* par le chemin d'accès au répertoire dans lequel vous souhaitez restaurer le système de fichiers. Par exemple, **/mnt/data/**.
- Pour restaurer un système de fichiers à partir d'une sauvegarde incrémentale (niveau 1 à niveau 9), ajoutez l'option **-r**.
- Pour restaurer une sauvegarde à partir d'une unité de bande qui contient plusieurs sauvegardes, spécifiez la sauvegarde à l'aide des options **-S** ou **-L**.
L'option **-S** vous permet de choisir une sauvegarde en fonction de son identifiant de session, tandis que l'option **-L** vous permet de choisir en fonction du libellé de la session. Pour obtenir l'ID de session et les étiquettes de session, utilisez la commande **xfsrestore -i**.

Remplacer *session-id* par l'identifiant de session de la sauvegarde. Par exemple, **b74a3586-e52e-4a4a-8775-c3334fa8ea2c**. Remplacer *session-label* par l'étiquette de session de la sauvegarde. Par exemple, **my_backup_session_label**.

- Pour utiliser **xfsrestore** de manière interactive, utilisez l'option **-i**.
Le dialogue interactif commence après que **xfsrestore** a fini de lire le périphérique spécifié. Les commandes disponibles dans le shell interactif **xfsrestore** sont **cd**, **ls**, **add**, **delete**, et **extract**; pour une liste complète des commandes, utilisez la commande **help**.

Exemple 19.1. Restauration de plusieurs systèmes de fichiers XFS

- Pour restaurer les fichiers de sauvegarde XFS et enregistrer leur contenu dans des répertoires sous **/mnt/**:

```
# xfsrestore -f /backup-files/boot.xfsdump /mnt/boot/
# xfsrestore -f /backup-files/data.xfsdump /mnt/data/
```

- Pour restaurer à partir d'une unité de bande contenant plusieurs sauvegardes, spécifiez chaque sauvegarde par son étiquette ou son ID de session :

```
# xfsrestore -L "backup_boot" -f /dev/st0 /mnt/boot/
# xfsrestore -S "45e9af35-efd2-4244-87bc-4762e476cbab" \
-f /dev/st0 /mnt/data/
```

Ressources supplémentaires

- **xfsrestore(8)** page de manuel.

19.3. MESSAGES D'INFORMATION LORS DE LA RESTAURATION D'UNE SAUVEGARDE XFS À PARTIR D'UNE BANDE

Lors de la restauration d'une sauvegarde à partir d'une bande contenant des sauvegardes de plusieurs systèmes de fichiers, l'utilitaire **xfsrestore** peut émettre des messages. Ces messages vous informent si une correspondance de la sauvegarde demandée a été trouvée lorsque **xfsrestore** examine chaque sauvegarde sur la bande dans l'ordre séquentiel. Par exemple :

```
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)
[...]
```

Les messages d'information continuent à s'afficher jusqu'à ce que la sauvegarde correspondante soit trouvée.

19.4. RESSOURCES SUPPLÉMENTAIRES

- **xfsrestore(8)** page de manuel

CHAPITRE 20. AUGMENTER LA TAILLE D'UN SYSTÈME DE FICHIERS XFS

En tant qu'administrateur système, vous pouvez augmenter la taille d'un système de fichiers XFS afin d'utiliser pleinement une plus grande capacité de stockage.



IMPORTANT

Il n'est actuellement pas possible de réduire la taille des systèmes de fichiers XFS.

20.1. AUGMENTER LA TAILLE D'UN SYSTÈME DE FICHIERS XFS AVEC XFS_GROWFS

Cette procédure décrit comment développer un système de fichiers XFS à l'aide de l'utilitaire **xfs_growfs**.

Conditions préalables

- Assurez-vous que le périphérique de bloc sous-jacent est d'une taille appropriée pour contenir le système de fichiers redimensionné ultérieurement. Utilisez les méthodes de redimensionnement appropriées pour le périphérique de bloc concerné.
- Monter le système de fichiers XFS.

Procédure

- Lorsque le système de fichiers XFS est monté, utilisez l'utilitaire **xfs_growfs** pour augmenter sa taille :

```
# xfs_growfs file-system -D new-size
```

- Remplacer *file-system* par le point de montage du système de fichiers XFS.
- Avec l'option **-D**, remplacez *new-size* par la nouvelle taille souhaitée du système de fichiers, spécifiée en nombre de blocs du système de fichiers.
Pour connaître la taille des blocs en Ko d'un système de fichiers XFS donné, utilisez l'utilitaire **xfs_info**:

```
# xfs_info block-device
...
data =          bsize=4096
...
```

- Sans l'option **-D**, **xfs_growfs** agrandit le système de fichiers jusqu'à la taille maximale supportée par le périphérique sous-jacent.

Ressources supplémentaires

- **xfs_growfs(8)** page de manuel.

CHAPITRE 21. CONFIGURATION DU COMPORTEMENT DES ERREURS XFS

Vous pouvez configurer le comportement d'un système de fichiers XFS lorsqu'il rencontre différentes erreurs d'E/S.

21.1. GESTION CONFIGURABLE DES ERREURS DANS XFS

Le système de fichiers XFS réagit de l'une des manières suivantes lorsqu'une erreur se produit au cours d'une opération d'E/S :

- XFS tente à plusieurs reprises l'opération d'E/S jusqu'à ce que l'opération réussisse ou que XFS atteigne une limite fixée.
La limite est basée soit sur un nombre maximum de tentatives, soit sur un délai maximum pour les tentatives.
- XFS considère l'erreur comme permanente et arrête l'opération sur le système de fichiers.

Vous pouvez configurer la façon dont XFS réagit aux conditions d'erreur suivantes :

EIO

Erreur lors de la lecture ou de l'écriture

ENOSPC

Il n'y a plus d'espace disponible sur l'appareil

ENODEV

Dispositif introuvable

Vous pouvez définir le nombre maximal de tentatives et le délai maximal en secondes avant que XFS ne considère une erreur comme permanente. XFS arrête de réessayer l'opération lorsqu'il atteint l'une de ces limites.

Vous pouvez également configurer XFS de manière à ce que, lors du démontage d'un système de fichiers, XFS annule immédiatement les tentatives, quelle que soit la configuration. Cette configuration permet à l'opération de démontage d'aboutir malgré des erreurs persistantes.

Comportement par défaut

Le comportement par défaut pour chaque condition d'erreur XFS dépend du contexte de l'erreur. Certaines erreurs XFS, telles que **ENODEV**, sont considérées comme fatales et irrécupérables, quel que soit le nombre de tentatives. Leur limite de tentatives par défaut est de 0.

21.2. FICHIERS DE CONFIGURATION POUR LES CONDITIONS D'ERREUR XFS SPÉCIFIQUES ET NON DÉFINIES

Les répertoires suivants contiennent des fichiers de configuration qui contrôlent le comportement du système XFS en fonction des différentes conditions d'erreur :

/sys/fs/xfs/device/error/metadata/EIO/

Pour la condition d'erreur **EIO**

/sys/fs/xfs/device/error/metadata/ENODEV/

Pour la condition d'erreur **ENODEV**

/sys/fs/xfs/device/error/metadata/ENOSPC/

Pour la condition d'erreur **ENOSPC**

/sys/fs/xfs/device/error/default/

Configuration commune pour toutes les autres conditions d'erreur non définies

Chaque répertoire contient les fichiers de configuration suivants pour configurer les limites de tentatives :

max_retries

Contrôle le nombre maximal de tentatives de l'opération par XFS.

retry_timeout_seconds

Spécifie la limite de temps en secondes après laquelle XFS arrête de réessayer l'opération.

21.3. DÉFINITION DU COMPORTEMENT DE XFS DANS DES CONDITIONS SPÉCIFIQUES

Cette procédure permet de configurer la manière dont XFS réagit à des conditions d'erreur spécifiques.

Procédure

- Définir le nombre maximal de tentatives, le délai de tentative ou les deux :
 - Pour définir le nombre maximum de tentatives, écrivez le nombre souhaité dans le fichier **max_retries**:

```
# echo value > /sys/fs/xfs/device/error/metadata/condition/max_retries
```

- Pour définir la limite de temps, écrivez le nombre de secondes souhaité dans le fichier **retry_timeout_seconds**:

```
# echo value > /sys/fs/xfs/device/error/metadata/condition/retry_timeout_second
```

value est un nombre compris entre -1 et la valeur maximale possible du type entier signé en C. Cette valeur est de 2147483647 sous Linux 64 bits.

Dans les deux limites, la valeur **-1** est utilisée pour des tentatives continues et **0** pour un arrêt immédiat.

device est le nom de l'appareil, tel qu'il figure dans le répertoire **/dev/**; par exemple, **sda**.

21.4. DÉFINITION DU COMPORTEMENT DE XFS EN CAS DE CONDITIONS NON DÉFINIES

Cette procédure configure la manière dont XFS réagit à toutes les conditions d'erreur non définies, qui partagent une configuration commune.

Procédure

- Définir le nombre maximal de tentatives, le délai de tentative ou les deux :
 - Pour définir le nombre maximum de tentatives, écrivez le nombre souhaité dans le fichier **max_retries**:

```
# echo value > /sys/fs/xfs/device/error/metadata/default/max_retries
```

- Pour définir la limite de temps, écrivez le nombre de secondes souhaité dans le fichier **retry_timeout_seconds**:

```
# echo value > /sys/fs/xfs/device/error/metadata/default/retry_timeout_seconds
```

value est un nombre compris entre -1 et la valeur maximale possible du type entier signé en C. Cette valeur est de 2147483647 sous Linux 64 bits.

Dans les deux limites, la valeur **-1** est utilisée pour des tentatives continues et **0** pour un arrêt immédiat.

device est le nom de l'appareil, tel qu'il figure dans le répertoire **/dev/**; par exemple, **sda**.

21.5. PARAMÉTRAGE DU COMPORTEMENT DE DÉMONTAGE DE XFS

Cette procédure permet de configurer la manière dont XFS réagit aux conditions d'erreur lors du démontage du système de fichiers.

Si vous définissez l'option **fail_at_unmount** dans le système de fichiers, elle remplace toutes les autres configurations d'erreur lors du démontage et démonte immédiatement le système de fichiers sans réessayer l'opération d'E/S. Cela permet de réussir l'opération de démontage même en cas d'erreurs persistantes. Cela permet à l'opération de démontage d'aboutir même en cas d'erreurs persistantes.



AVERTISSEMENT

Vous ne pouvez pas modifier la valeur de **fail_at_unmount** après le démarrage du processus de démontage, car celui-ci supprime les fichiers de configuration de l'interface **sysfs** pour le système de fichiers concerné. Vous devez configurer le comportement de démontage avant que le système de fichiers ne commence à être démonté.

Procédure

- Active ou désactive l'option **fail_at_unmount**:
 - Pour annuler la répétition de toutes les opérations lorsque le système de fichiers se démonte, activez l'option :

```
# echo 1 > /sys/fs/xfs/device/error/fail_at_unmount
```

- Pour respecter les limites de tentatives **max_retries** et **retry_timeout_seconds** lors du démontage du système de fichiers, désactivez l'option :

```
# echo 0 > /sys/fs/xfs/device/error/fail_at_unmount
```

device est le nom de l'appareil, tel qu'il figure dans le répertoire **/dev/**; par exemple, **sda**.

CHAPITRE 22. VÉRIFICATION ET RÉPARATION D'UN SYSTÈME DE FICHIERS

RHEL fournit des utilitaires d'administration des systèmes de fichiers capables de vérifier et de réparer les systèmes de fichiers. Ces outils sont souvent appelés outils **fsck**, où **fsck** est une version abrégée de *file system check*. Dans la plupart des cas, ces utilitaires sont exécutés automatiquement lors du démarrage du système, si nécessaire, mais peuvent également être invoqués manuellement si nécessaire.



IMPORTANT

Les vérificateurs de systèmes de fichiers ne garantissent que la cohérence des métadonnées dans le système de fichiers. Ils n'ont aucune connaissance des données réelles contenues dans le système de fichiers et ne sont pas des outils de récupération de données.

22.1. SCÉNARIOS NÉCESSITANT UNE VÉRIFICATION DU SYSTÈME DE FICHIERS

Les outils **fsck** appropriés peuvent être utilisés pour vérifier votre système si l'une des situations suivantes se produit :

- Le système ne démarre pas
- Les fichiers d'un disque spécifique sont corrompus
- Le système de fichiers s'arrête ou passe en lecture seule en raison d'incohérences
- Un fichier du système de fichiers est inaccessible

Les incohérences du système de fichiers peuvent survenir pour diverses raisons, notamment des erreurs matérielles, des erreurs d'administration du stockage et des bogues logiciels.



IMPORTANT

Les outils de vérification du système de fichiers ne peuvent pas réparer les problèmes matériels. Un système de fichiers doit être entièrement lisible et inscriptible pour pouvoir être réparé avec succès. Si un système de fichiers a été corrompu à la suite d'une erreur matérielle, il doit d'abord être déplacé sur un disque en bon état, par exemple à l'aide de l'utilitaire **dd(8)**.

Pour les systèmes de fichiers avec journalisation, tout ce qui est normalement requis au démarrage est de rejouer le journal si nécessaire, ce qui est généralement une opération très courte.

Toutefois, en cas d'incohérence ou de corruption du système de fichiers, même pour les systèmes de fichiers avec journalisation, le vérificateur de système de fichiers doit être utilisé pour réparer le système de fichiers.



IMPORTANT

Il est possible de désactiver la vérification du système de fichiers au démarrage en définissant le sixième champ de **/etc/fstab** sur **0**. Cependant, Red Hat ne recommande pas de le faire à moins que vous ne rencontriez des problèmes avec **fsck** au moment du démarrage, par exemple avec des systèmes de fichiers extrêmement volumineux ou distants.

Ressources supplémentaires

- **fstab(5)** page de manuel.
- **fsck(8)** page de manuel.
- **dd(8)** page de manuel.

22.2. EFFETS SECONDAIRES POTENTIELS DE L'EXÉCUTION DE FSCK

En règle générale, l'exécution de l'outil de vérification et de réparation du système de fichiers devrait permettre de réparer automatiquement au moins certaines des incohérences détectées. Dans certains cas, les problèmes suivants peuvent survenir :

- Les inodes ou répertoires gravement endommagés peuvent être éliminés s'ils ne peuvent pas être réparés.
- Des modifications importantes peuvent être apportées au système de fichiers.

Pour éviter que des modifications inattendues ou indésirables ne soient apportées de manière permanente, veillez à suivre toutes les mesures de précaution décrites dans la procédure.

22.3. MÉCANISMES DE TRAITEMENT DES ERREURS DANS LE SYSTÈME XFS

Cette section décrit la manière dont XFS gère différents types d'erreurs dans le système de fichiers.

Démontages sauvages

La journalisation maintient un enregistrement transactionnel des changements de métadonnées qui se produisent sur le système de fichiers.

En cas de panne du système, de coupure de courant ou de tout autre démontage non conforme, XFS utilise le journal (également appelé log) pour récupérer le système de fichiers. Le noyau effectue la récupération du journal lors du montage du système de fichiers XFS.

Corruption

Dans ce contexte, *corruption* désigne les erreurs sur le système de fichiers causées, par exemple, par les éléments suivants

- Défauts matériels
- Bogues dans le micrologiciel de stockage, les pilotes de périphériques, la pile logicielle ou le système de fichiers lui-même
- Problèmes entraînant l'écrasement de certaines parties du système de fichiers par un élément extérieur au système de fichiers

Lorsque XFS détecte une corruption dans le système de fichiers ou dans les métadonnées du système de fichiers, il peut arrêter le système de fichiers et signaler l'incident dans le journal du système. Notez que si la corruption s'est produite sur le système de fichiers hébergeant le répertoire **/var**, ces journaux ne seront pas disponibles après un redémarrage.

Exemple 22.1. Entrée du journal système signalant une corruption 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:
0000000027b3b56: 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
```

Les utilitaires de l'espace utilisateur affichent généralement le message *Input/output error* lorsqu'ils tentent d'accéder à un système de fichiers XFS corrompu. Le montage d'un système de fichiers XFS dont le journal est corrompu entraîne l'échec du montage et le message d'erreur suivant :

```
mount : /mount-point: l'appel système mount(2) a échoué : La structure doit être nettoyée.
```

Vous devez utiliser manuellement l'utilitaire **xfs_repair** pour réparer la corruption.

Ressources supplémentaires

- **xfs_repair(8)** page de manuel.

22.4. VÉRIFICATION D'UN SYSTÈME DE FICHIERS XFS AVEC **XFS_REPAIR**

Cette procédure permet d'effectuer une vérification en lecture seule d'un système de fichiers XFS à l'aide de l'utilitaire **xfs_repair**. Vous devez utiliser manuellement l'utilitaire **xfs_repair** pour réparer toute corruption. Contrairement à d'autres utilitaires de réparation de systèmes de fichiers, **xfs_repair** ne s'exécute pas au démarrage, même si un système de fichiers XFS n'a pas été démonté proprement. Dans le cas d'un démontage incorrect, XFS rejoue simplement le journal au moment du montage, garantissant ainsi un système de fichiers cohérent ; **xfs_repair** ne peut pas réparer un système de fichiers XFS dont le journal est sale sans le remonter au préalable.

**NOTE**

Bien qu'un binaire **fsck.xfs** soit présent dans le paquetage **xfsprogs**, il n'est présent que pour satisfaire **initscripts** qui recherche un binaire système **fsck.file** au moment du démarrage. **fsck.xfs** se termine immédiatement avec un code de sortie de 0.

Procédure

1. Reprendre le journal en montant et démontant le système de fichiers :

```
# mount file-system
# umount file-system
```

**NOTE**

Si le montage échoue avec une erreur de structure à nettoyer, le journal est corrompu et ne peut pas être rejoué. L'exécution à blanc devrait découvrir et signaler davantage de corruption sur le disque.

2. Utilisez l'utilitaire **xfs_repair** pour effectuer une vérification à blanc du système de fichiers. Les erreurs éventuelles sont imprimées, ainsi qu'une indication des mesures à prendre, sans modifier le système de fichiers.

```
# xfs_repair -n block-device
```

3. Monter le système de fichiers :

```
# mount file-system
```

Ressources supplémentaires

- **xfs_repair(8)** page de manuel.
- **xfs_metadump(8)** page de manuel.

22.5. RÉPARATION D'UN SYSTÈME DE FICHIERS XFS AVEC XFS_REPAIR

Cette procédure permet de réparer un système de fichiers XFS corrompu à l'aide de l'utilitaire **xfs_repair**.

Procédure

1. Créez une image de métadonnées avant la réparation à des fins de diagnostic ou de test à l'aide de l'utilitaire **xfs_metadump**. Une image de métadonnées du système de fichiers avant réparation peut être utile pour les enquêtes de support si la corruption est due à un bogue logiciel. Les schémas de corruption présents dans l'image avant réparation peuvent faciliter l'analyse des causes profondes.
 - Utilisez l'outil de débogage **xfs_metadump** pour copier les métadonnées d'un système de fichiers XFS dans un fichier. Le fichier **metadump** résultant peut être compressé à l'aide d'utilitaires de compression standard afin de réduire la taille du fichier si des fichiers **metadump** volumineux doivent être envoyés à l'assistance.

```
# xfs_metadump block-device metadump-file
```

2. Reproduire le journal en remontant le système de fichiers :

```
# mount file-system
# umount file-system
```

3. Utilisez l'utilitaire **xfs_repair** pour réparer le système de fichiers non monté :

- Si le montage a réussi, aucune autre option n'est nécessaire :

```
# xfs_repair block-device
```

- Si le montage a échoué avec l'erreur *Structure needs cleaning*, le journal est corrompu et ne peut pas être rejoué. Utilisez l'option **-L** (*force log zeroing*) pour effacer le journal :



AVERTISSEMENT

Cette commande entraîne la perte de toutes les mises à jour de métadonnées en cours au moment du crash, ce qui risque d'endommager considérablement le système de fichiers et de provoquer des pertes de données. Cette commande ne doit être utilisée qu'en dernier recours, si le journal ne peut pas être rejoué.

```
# xfs_repair -L block-device
```

4. Monter le système de fichiers :

```
# mount file-system
```

Ressources supplémentaires

- **xfs_repair(8)** page de manuel.

22.6. MÉCANISMES DE GESTION DES ERREURS DANS EXT2, EXT3 ET EXT4

Les systèmes de fichiers ext2, ext3 et ext4 utilisent l'utilitaire **e2fsck** pour effectuer des vérifications et des réparations du système de fichiers. Les noms de fichiers **fsck.ext2**, **fsck.ext3**, et **fsck.ext4** sont des liens en dur vers l'utilitaire **e2fsck**. Ces binaires sont exécutés automatiquement au démarrage et leur comportement diffère en fonction du système de fichiers contrôlé et de l'état du système de fichiers.

Une vérification et une réparation complètes du système de fichiers sont demandées pour ext2, qui n'est pas un système de fichiers avec journalisation des métadonnées, et pour les systèmes de fichiers ext4 sans journal.

Pour les systèmes de fichiers ext3 et ext4 avec journalisation des métadonnées, le journal est rejoué dans l'espace utilisateur et l'utilitaire se termine. Il s'agit de l'action par défaut, car la relecture du journal garantit la cohérence du système de fichiers après un crash.

Si ces systèmes de fichiers rencontrent des incohérences dans les métadonnées lorsqu'ils sont montés, ils enregistrent ce fait dans le superbloc du système de fichiers. Si **e2fsck** constate qu'un système de fichiers est marqué par une telle erreur, **e2fsck** effectue une vérification complète après avoir rejoué le journal (le cas échéant).

Ressources supplémentaires

- **fsck(8)** page de manuel.
- **e2fsck(8)** page de manuel.

22.7. VÉRIFICATION D'UN SYSTÈME DE FICHIERS EXT2, EXT3 OU EXT4 AVEC E2FSCK

Cette procédure vérifie un système de fichiers ext2, ext3 ou ext4 à l'aide de l'utilitaire **e2fsck**.

Procédure

1. Reproduire le journal en remontant le système de fichiers :

```
# mount file-system
# umount file-system
```

2. Effectuez un essai à blanc pour vérifier le système de fichiers.

```
# e2fsck -n block-device
```



NOTE

Toutes les erreurs sont affichées, ainsi qu'une indication des mesures à prendre, sans modifier le système de fichiers. Les phases ultérieures du contrôle de cohérence peuvent afficher des erreurs supplémentaires lorsqu'elles découvrent des incohérences qui auraient été corrigées au cours des premières phases si elles avaient été exécutées en mode réparation.

Ressources supplémentaires

- **e2image(8)** page de manuel.
- **e2fsck(8)** page de manuel.

22.8. RÉPARATION D'UN SYSTÈME DE FICHIERS EXT2, EXT3 OU EXT4 AVEC E2FSCK

Cette procédure permet de réparer un système de fichiers ext2, ext3 ou ext4 endommagé à l'aide de l'utilitaire **e2fsck**.

Procédure

1. Sauvegarder une image du système de fichiers pour les enquêtes de support. Une image des métadonnées du système de fichiers avant réparation peut être utile pour les enquêtes de support si la corruption est due à un bogue logiciel. Les schémas de corruption présents dans l'image de pré-réparation peuvent faciliter l'analyse des causes profondes.



NOTE

Les systèmes de fichiers gravement endommagés peuvent poser des problèmes lors de la création d'images de métadonnées.

- Si vous créez l'image à des fins de test, utilisez l'option **-r** pour créer un fichier clair de la même taille que le système de fichiers lui-même. **e2fsck** peut alors opérer directement sur le fichier résultant.

```
# e2image -r block-device image-file
```

- Si vous créez l'image pour l'archiver ou la fournir à des fins de diagnostic, utilisez l'option **-Q**, qui crée un format de fichier plus compact adapté au transfert.

```
# e2image -Q block-device image-file
```

2. Reproduire le journal en remontant le système de fichiers :

```
# mount file-system  
# umount file-system
```

3. Réparation automatique du système de fichiers. Si l'intervention de l'utilisateur est nécessaire, **e2fsck** indique le problème non résolu dans sa sortie et reflète cet état dans le code de sortie.

```
# e2fsck -p block-device
```

Ressources supplémentaires

- **e2image(8)** page de manuel.
- **e2fsck(8)** page de manuel.

CHAPITRE 23. MONTAGE DES SYSTÈMES DE FICHIERS

En tant qu'administrateur système, vous pouvez monter des systèmes de fichiers sur votre système pour accéder aux données qu'ils contiennent.

23.1. LE MÉCANISME DE MONTAGE DE LINUX

Cette section explique les concepts de base du montage de systèmes de fichiers sous Linux.

Sous Linux, UNIX et d'autres systèmes d'exploitation similaires, les systèmes de fichiers sur différentes partitions et périphériques amovibles (CD, DVD ou clés USB par exemple) peuvent être attachés à un certain point (le point de montage) dans l'arborescence des répertoires, puis détachés à nouveau. Lorsqu'un système de fichiers est monté sur un répertoire, le contenu original du répertoire n'est pas accessible.

Notez que Linux ne vous empêche pas de monter un système de fichiers dans un répertoire auquel un système de fichiers est déjà attaché.

Lors du montage, vous pouvez identifier l'appareil par :

- un identifiant universel unique (UUID) : par exemple, **UUID=34795a28-ca6d-4fd8-a347-73671d0c19cb**
- une étiquette de volume : par exemple, **LABEL=home**
- un chemin d'accès complet à un périphérique de bloc non persistant : par exemple, **/dev/sda3**

Lorsque vous montez un système de fichiers à l'aide de la commande **mount** sans toutes les informations requises, c'est-à-dire sans le nom du périphérique, le répertoire cible ou le type de système de fichiers, l'utilitaire **mount** lit le contenu du fichier **/etc/fstab** pour vérifier si le système de fichiers donné y est répertorié. Le fichier **/etc/fstab** contient une liste de noms de périphériques et de répertoires dans lesquels les systèmes de fichiers sélectionnés doivent être montés, ainsi que le type de système de fichiers et les options de montage. Par conséquent, lors du montage d'un système de fichiers spécifié dans le fichier **/etc/fstab**, la syntaxe de commande suivante est suffisante :

- Montage par le point de montage :

```
# mount directory
```

- Montage par le dispositif de blocage :

```
# mount device
```

Ressources supplémentaires

- **mount(8)** page de manuel
- [Comment dresser la liste des attributs de dénomination persistants tels que l'UUID](#) .

23.2. LISTE DES SYSTÈMES DE FICHIERS ACTUELLEMENT MONTÉS

Cette procédure décrit comment dresser la liste de tous les systèmes de fichiers actuellement montés sur la ligne de commande.

Procédure

- Pour dresser la liste de tous les systèmes de fichiers montés, utilisez l'utilitaire **findmnt**:

```
$ findmnt
```

- Pour limiter la liste des systèmes de fichiers à un certain type de système de fichiers, ajoutez l'option **--types**:

```
$ findmnt --types fs-type
```

Par exemple :

Exemple 23.1. Liste des systèmes de fichiers XFS uniquement

```
$ 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

Ressources supplémentaires

- **findmnt(8)** page de manuel

23.3. MONTAGE D'UN SYSTÈME DE FICHIERS AVEC MOUNT

Cette procédure décrit comment monter un système de fichiers à l'aide de l'utilitaire **mount**.

Conditions préalables

- Assurez-vous qu'aucun système de fichiers n'est déjà monté sur le point de montage choisi :

```
$ findmnt mount-point
```

Procédure

1. Pour attacher un certain système de fichiers, utilisez l'utilitaire **mount**:

```
# mount device mount-point
```

Exemple 23.2. Montage d'un système de fichiers XFS

Par exemple, pour monter un système de fichiers XFS local identifié par UUID :

```
# mount UUID=ea74bbec-536d-490c-b8d9-5b40bbd7545b /mnt/data
```

2. Si **mount** ne peut pas reconnaître automatiquement le type de système de fichiers, spécifiez-le à l'aide de l'option **--types**:

```
# mount --types type device mount-point
```

Exemple 23.3. Montage d'un système de fichiers NFS

Par exemple, pour monter un système de fichiers NFS distant :

```
# mount --types nfs4 host:/remote-export /mnt/nfs
```

Ressources supplémentaires

- **mount(8)** page de manuel

23.4. DÉPLACEMENT D'UN POINT DE MONTAGE

Cette procédure décrit comment changer le point de montage d'un système de fichiers monté pour un autre répertoire.

Procédure

1. Pour modifier le répertoire dans lequel un système de fichiers est monté :

```
# mount --move old-directory new-directory
```

Exemple 23.4. Déplacement d'un système de fichiers domestique

Par exemple, pour déplacer le système de fichiers monté dans le répertoire **/mnt/userdirs/** vers le point de montage **/home/**:

```
# mount --move /mnt/userdirs /home
```

2. Vérifiez que le système de fichiers a été déplacé comme prévu :

```
$ findmnt  
$ ls old-directory  
$ ls new-directory
```

Ressources supplémentaires

- **mount(8)** page de manuel

23.5. DÉMONTAGE D'UN SYSTÈME DE FICHIERS AVEC UMOUNT

Cette procédure décrit comment démonter un système de fichiers à l'aide de l'utilitaire **umount**.

Procédure

1. Essayez de démonter le système de fichiers à l'aide de l'une des commandes suivantes :
 - Par point de montage :

```
# umount mount-point
```

- Par appareil :

```
# umount device
```

Si la commande échoue avec une erreur similaire à la suivante, cela signifie que le système de fichiers est en cours d'utilisation parce qu'un processus y utilise des ressources :

```
umount : /run/media/user/FlashDrive: la cible est occupée.
```

2. Si le système de fichiers est utilisé, utilisez l'utilitaire **fuser** pour déterminer les processus qui y accèdent. Par exemple, l'utilitaire

```
$ fuser --mount /run/media/user/FlashDrive
```

```
/run/media/user/FlashDrive: 18351
```

Ensuite, mettez fin aux processus utilisant le système de fichiers et essayez à nouveau de le démonter.

23.6. OPTIONS DE MONTAGE COURANTES

Le tableau suivant répertorie les options les plus courantes de l'utilitaire **mount**. Vous pouvez appliquer ces options de montage en utilisant la syntaxe suivante :

```
# mount --options option1,option2,option3 device mount-point
```

Tableau 23.1. Options de montage courantes

Option	Description
async	Active les opérations d'entrée et de sortie asynchrones sur le système de fichiers.
auto	Permet de monter automatiquement le système de fichiers à l'aide de la commande mount -a .
defaults	Fournit un alias pour les options async,auto,dev,exec,nouser,rw,suid .
exec	Permet l'exécution de fichiers binaires sur le système de fichiers concerné.
loop	Monte une image en tant que périphérique en boucle.
noauto	Le comportement par défaut désactive le montage automatique du système de fichiers à l'aide de la commande mount -a .
noexec	Interdit l'exécution de fichiers binaires sur le système de fichiers concerné.
nouser	Interdit à un utilisateur ordinaire (c'est-à-dire autre que root) de monter et démonter le système de fichiers.

Option	Description
remount	Remonte le système de fichiers s'il est déjà monté.
ro	Monte le système de fichiers en lecture seule.
rw	Monte le système de fichiers pour la lecture et l'écriture.
user	Permet à un utilisateur ordinaire (c'est-à-dire autre que root) de monter et démonter le système de fichiers.

CHAPITRE 24. PARTAGE D'UN MONTAGE SUR PLUSIEURS POINTS DE MONTAGE

En tant qu'administrateur système, vous pouvez dupliquer les points de montage pour rendre les systèmes de fichiers accessibles à partir de plusieurs répertoires.

24.1. TYPES DE MONTAGES PARTAGÉS

Il existe plusieurs types de montages partagés que vous pouvez utiliser. La différence entre eux est ce qui se passe lorsque vous montez un autre système de fichiers sous l'un des points de montage partagés. Les montages partagés sont mis en œuvre à l'aide de la fonctionnalité *shared subtrees*.

Les types de montage suivants sont disponibles :

private

Ce type ne reçoit ni ne transmet aucun événement de propagation.

Lorsque vous montez un autre système de fichiers sous le point de montage dupliqué ou le point de montage original, cela ne se reflète pas dans l'autre.

shared

Ce type de fichier crée une réplique exacte d'un point de montage donné.

Lorsqu'un point de montage est marqué comme étant un montage **shared**, tout montage à l'intérieur du point de montage d'origine est reflété dans celui-ci, et vice versa.

Il s'agit du type de montage par défaut du système de fichiers racine.

slave

Ce type de fichier crée une copie limitée d'un point de montage donné.

Lorsqu'un point de montage est marqué comme étant un montage **slave**, tout montage dans le point de montage d'origine est reflété dans celui-ci, mais aucun montage dans un montage **slave** n'est reflété dans son point d'origine.

unbindable

Ce type empêche la duplication du point de montage donné.

Ressources supplémentaires

- [L'article *Shared subtrees* sur Linux Weekly News](#).

24.2. CRÉATION D'UN DUPLICATA DE POINT DE MONTAGE PRIVÉ

Cette procédure permet de dupliquer un point de montage en tant que montage privé. Les systèmes de fichiers que vous montez ultérieurement sous le point de montage dupliqué ou le point de montage original ne sont pas reflétés dans l'autre.

Procédure

1. Créez un nœud de système de fichiers virtuels (VFS) à partir du point de montage d'origine :

```
# mount --bind original-dir original-dir
```

2. Marquer le point de montage d'origine comme privé :

```
# mount --make-private original-dir
```

Pour modifier le type de montage du point de montage sélectionné et de tous les points de montage qui lui sont associés, vous pouvez également utiliser l'option **--make-rprivate** au lieu de **--make-private**.

3. Créer le duplicata :

```
# mount --bind original-dir duplicate-dir
```

Exemple 24.1. Duplication de /media dans /mnt comme point de montage privé

1. Créez un nœud VFS à partir du répertoire **/media**:

```
# mount --bind /media /media
```

2. Marquez le répertoire **/media** comme privé :

```
# mount --make-private /media
```

3. Créer un duplicata dans **/mnt**:

```
# mount --bind /media /mnt
```

4. Il est maintenant possible de vérifier que **/media** et **/mnt** partagent le même contenu, mais qu'aucun des montages de **/media** n'apparaît dans **/mnt**. Par exemple, si le lecteur de CD-ROM contient des supports non vides et que le répertoire **/media/cdrom/** existe, utilisez :

```
# mount /dev/cdrom /media/cdrom
# ls /media/cdrom
EFI GPL isolinux LiveOS
# ls /mnt/cdrom
#
```

5. Il est également possible de vérifier que les systèmes de fichiers montés dans le répertoire **/mnt** ne sont pas reflétés dans **/media**. Par exemple, si une clé USB non vide utilisant le périphérique **/dev/sdc1** est branchée et que le répertoire **/mnt/flashdisk/** est présent, utilisez :

```
# mount /dev/sdc1 /mnt/flashdisk
# ls /media/flashdisk
# ls /mnt/flashdisk
en-US publican.cfg
```

Ressources supplémentaires

- **mount(8)** page de manuel

24.3. CRÉATION D'UN POINT DE MONTAGE PARTAGÉ EN DOUBLE

Cette procédure permet de dupliquer un point de montage en tant que montage partagé. Les systèmes de fichiers que vous montez par la suite dans le répertoire d'origine ou dans le duplicata sont toujours reflétés dans l'autre.

Procédure

1. Créez un nœud de système de fichiers virtuels (VFS) à partir du point de montage d'origine :

```
# mount --bind original-dir original-dir
```

2. Marquer le point de montage d'origine comme partagé :

```
# mount --make-shared original-dir
```

Pour modifier le type de montage du point de montage sélectionné et de tous les points de montage qui lui sont associés, vous pouvez également utiliser l'option **--make-rshared** au lieu de **--make-shared**.

3. Créer le duplicata :

```
# mount --bind original-dir duplicate-dir
```

Exemple 24.2. Duplication de /media dans /mnt comme point de montage partagé

Pour que les répertoires **/media** et **/mnt** partagent le même contenu :

1. Créez un nœud VFS à partir du répertoire **/media**:

```
# mount --bind /media /media
```

2. Marquez le répertoire **/media** comme étant partagé :

```
# mount --make-shared /media
```

3. Créer un duplicata dans **/mnt**:

```
# mount --bind /media /mnt
```

4. Il est maintenant possible de vérifier qu'un montage dans **/media** apparaît également dans **/mnt**. Par exemple, si le lecteur de CD-ROM contient des supports non vides et que le répertoire **/media/cdrom/** existe, utilisez :

```
# mount /dev/cdrom /media/cdrom
# ls /media/cdrom
EFI GPL isolinux LiveOS
# ls /mnt/cdrom
EFI GPL isolinux LiveOS
```

5. De même, il est possible de vérifier que tout système de fichiers monté dans le répertoire **/mnt** est reflété dans **/media**. Par exemple, si une clé USB non vide utilisant le périphérique **/dev/sdc1** est branchée et que le répertoire **/mnt/flashdisk/** est présent, utilisez :

```
# mount /dev/sdc1 /mnt/flashdisk
```

```
# ls /media/flashdisk
en-US publican.cfg
# ls /mnt/flashdisk
en-US publican.cfg
```

Ressources supplémentaires

- **mount(8)** page de manuel

24.4. CRÉATION D'UN DUPLICATA DE POINT DE MONTAGE ESCLAVE

Cette procédure permet de dupliquer un point de montage en tant que type de montage **slave**. Les systèmes de fichiers que vous montez ultérieurement sous le point de montage d'origine sont pris en compte dans le duplicata, mais pas l'inverse.

Procédure

1. Créez un nœud de système de fichiers virtuels (VFS) à partir du point de montage d'origine :

```
# mount --bind original-dir original-dir
```

2. Marquer le point de montage d'origine comme partagé :

```
# mount --make-shared original-dir
```

Pour modifier le type de montage du point de montage sélectionné et de tous les points de montage qui lui sont associés, vous pouvez également utiliser l'option **--make-rshared** au lieu de **--make-shared**.

3. Créez le duplicata et marquez-le comme étant le type **slave**:

```
# mount --bind original-dir duplicate-dir
# mount --make-slave duplicate-dir
```

Exemple 24.3. Duplication de /media dans /mnt comme point de montage esclave

Cet exemple montre comment faire apparaître le contenu du répertoire **/media** dans **/mnt**, mais sans qu'aucun des montages du répertoire **/mnt** ne soit reflété dans **/media**.

1. Créez un nœud VFS à partir du répertoire **/media**:

```
# mount --bind /media /media
```

2. Marquez le répertoire **/media** comme étant partagé :

```
# mount --make-shared /media
```

3. Créez son double dans **/mnt** et marquez-le comme **slave**:

```
# mount --bind /media /mnt
# mount --make-slave /mnt
```

4. Vérifiez qu'un montage dans **/media** apparaît également dans **/mnt**. Par exemple, si le lecteur de CD-ROM contient des supports non vides et que le répertoire **/media/cdrom/** existe, utilisez :

```
# mount /dev/cdrom /media/cdrom
# ls /media/cdrom
EFI GPL isolinux LiveOS
# ls /mnt/cdrom
EFI GPL isolinux LiveOS
```

5. Vérifiez également que les systèmes de fichiers montés dans le répertoire **/mnt** ne sont pas reflétés dans **/media**. Par exemple, si une clé USB non vide utilisant le périphérique **/dev/sdc1** est branchée et que le répertoire **/mnt/flashdisk/** est présent, utilisez :

```
# mount /dev/sdc1 /mnt/flashdisk
# ls /media/flashdisk
# ls /mnt/flashdisk
en-US publican.cfg
```

Ressources supplémentaires

- **mount(8)** page de manuel

24.5. EMPÊCHER LA DUPLICATION D'UN POINT DE MONTAGE

Cette procédure marque un point de montage comme non liant, de sorte qu'il n'est pas possible de le dupliquer dans un autre point de montage.

Procédure

- Pour changer le type d'un point de montage en un point de montage inséparable, utilisez la commande

```
# mount --bind mount-point mount-point
# mount --make-unbindable mount-point
```

Pour modifier le type de montage du point de montage sélectionné et de tous les points de montage qui lui sont associés, vous pouvez également utiliser l'option **--make-runbindable** au lieu de **--make-unbindable**.

Toute tentative ultérieure de duplication de ce montage échoue avec l'erreur suivante :

```
# 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
```

Exemple 24.4. Empêcher la duplication de **/media**

- Pour éviter que le répertoire **/media** ne soit partagé, utilisez :

```
# mount --bind /media /media  
# mount --make-unbindable /media
```

Ressources supplémentaires

- **mount(8)** page de manuel

CHAPITRE 25. MONTAGE PERSISTANT DES SYSTÈMES DE FICHIERS

En tant qu'administrateur système, vous pouvez monter de manière persistante des systèmes de fichiers pour configurer un stockage inamovible.

25.1. LE FICHER /ETC/FSTAB

Utilisez le fichier de configuration **/etc/fstab** pour contrôler les points de montage persistants des systèmes de fichiers. Chaque ligne du fichier **/etc/fstab** définit un point de montage d'un système de fichiers.

Il comprend six champs séparés par des espaces blancs :

1. Le périphérique bloc identifié par un attribut persistant ou un chemin d'accès au répertoire **/dev**.
2. Le répertoire dans lequel le périphérique sera monté.
3. Le système de fichiers de l'appareil.
4. Options de montage pour le système de fichiers, y compris l'option **defaults** pour monter la partition au démarrage avec les options par défaut. Le champ de l'option de montage reconnaît également les options de l'unité de montage **systemd** au format **x-systemd.option** au format
5. Option de sauvegarde pour l'utilitaire **dump**.
6. Ordre de chèque pour l'utilitaire **fsck**.



NOTE

Le site **systemd-fstab-generator** convertit dynamiquement les entrées du fichier **/etc/fstab** en unités **systemd-mount**. L'unité **systemd** monte automatiquement les volumes LVM à partir de **/etc/fstab** lors de l'activation manuelle, sauf si l'unité **systemd-mount** est masquée.



NOTE

L'utilitaire **dump** utilisé pour la sauvegarde des systèmes de fichiers a été supprimé dans RHEL 9, et est disponible dans le dépôt EPEL 9.

Exemple 25.1. Le système de fichiers/**boot** dans **/etc/fstab**

Dispositif de blocage	Point de montage	Système de fichiers	Options	Sauvegarde	Vérifier
UUID=ea74bbec-536d-490c-b8d9-5b40bbd7545b	/boot	xfs	defaults	0	0

Le service **systemd** génère automatiquement des unités de montage à partir des entrées de **/etc/fstab**.

Ressources supplémentaires

- **fstab(5)** et **systemd.mount(5)** pages de manuel

25.2. AJOUT D'UN SYSTÈME DE FICHIERS À /ETC/FSTAB

Cette procédure décrit comment configurer le point de montage persistant pour un système de fichiers dans le fichier de configuration **/etc/fstab**.

Procédure

1. Trouver l'attribut UUID du système de fichiers :

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

Par exemple :

Exemple 25.2. Visualisation de l'UUID d'une partition

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

NAME	FSTYPE	LABEL	UUID	MOUNTPOINT
sda1	xf	Boot	ea74bbec-536d-490c-b8d9-5b40bbd7545b	/boot

2. Si le répertoire du point de montage n'existe pas, créez-le :

```
# mkdir --parents mount-point
```

3. En tant que root, éditez le fichier **/etc/fstab** et ajoutez une ligne pour le système de fichiers, identifié par l'UUID.

Par exemple :

Exemple 25.3. Le point de montage /boot dans /etc/fstab

```
UUID=ea74bbec-536d-490c-b8d9-5b40bbd7545b /boot xfs defaults 0 0
```

4. Régénérez les unités de montage pour que votre système enregistre la nouvelle configuration :

```
# systemctl daemon-reload
```

5. Essayez de monter le système de fichiers pour vérifier que la configuration fonctionne :

```
# mount mount-point
```

Ressources supplémentaires

- [Vue d'ensemble des attributs de noms persistants](#) .

CHAPITRE 26. MONTAGE DE SYSTÈMES DE FICHIERS À LA DEMANDE

En tant qu'administrateur système, vous pouvez configurer les systèmes de fichiers, tels que NFS, pour qu'ils se montent automatiquement à la demande.

26.1. LE SERVICE AUTOFS

Cette section explique les avantages et les concepts de base du service **autofs**, utilisé pour monter des systèmes de fichiers à la demande.

L'un des inconvénients du montage permanent à l'aide de la configuration **/etc/fstab** est que, quelle que soit la fréquence d'accès d'un utilisateur au système de fichiers monté, le système doit consacrer des ressources pour maintenir le système de fichiers monté en place. Cela peut affecter les performances du système lorsque, par exemple, le système maintient des montages NFS sur de nombreux systèmes à la fois.

Une alternative à **/etc/fstab** est d'utiliser le service **autofs** basé sur le noyau. Il se compose des éléments suivants :

- Un module du noyau qui implémente un système de fichiers, et
- Un service de l'espace utilisateur qui exécute toutes les autres fonctions.

Le service **autofs** peut monter et démonter des systèmes de fichiers automatiquement (à la demande), ce qui permet d'économiser des ressources système. Il peut être utilisé pour monter des systèmes de fichiers tels que NFS, AFS, SMBFS, CIFS et des systèmes de fichiers locaux.

Ressources supplémentaires

- La page de manuel **autofs(8)**.

26.2. LES FICHIERS DE CONFIGURATION D'AUTOFS

Cette section décrit l'utilisation et la syntaxe des fichiers de configuration utilisés par le service **autofs**.

Le fichier de la carte maîtresse

Le service **autofs** utilise **/etc/auto.master** (carte maîtresse) comme fichier de configuration primaire par défaut. Il peut être modifié pour utiliser une autre source de réseau et un autre nom en utilisant la configuration **autofs** dans le fichier de configuration **/etc/autofs.conf** en conjonction avec le mécanisme Name Service Switch (NSS).

Tous les points de montage à la demande doivent être configurés dans la carte maîtresse. Le point de montage, le nom d'hôte, le répertoire exporté et les options peuvent tous être spécifiés dans un ensemble de fichiers (ou d'autres sources réseau prises en charge) plutôt que de les configurer manuellement pour chaque hôte.

Le fichier principal répertorie les points de montage contrôlés par **autofs**, ainsi que les fichiers de configuration correspondants ou les sources réseau connues sous le nom de cartes automount. Le format du fichier principal est le suivant :

```
mount-point map-name options
```

Les variables utilisées dans ce format sont les suivantes

mount-point

Le point de montage **autofs**; par exemple, **/mnt/data**.

map-file

Le fichier source de la carte, qui contient une liste de points de montage et l'emplacement du système de fichiers à partir duquel ces points de montage doivent être montés.

options

Si elles sont fournies, elles s'appliquent à toutes les entrées de la carte donnée, si elles n'ont pas elles-mêmes d'options spécifiées.

Exemple 26.1. Le fichier /etc/auto.master

Voici un exemple de ligne tirée du fichier **/etc/auto.master**:

```
/mnt/data /etc/auto.data
```

Fichiers cartographiques

Les fichiers de carte configurent les propriétés des différents points de montage à la demande.

Le compteur automatique crée les répertoires s'ils n'existent pas. Si les répertoires existent avant le démarrage du compteur automatique, ce dernier ne les supprimera pas lorsqu'il quittera le système. Si un délai est spécifié, le répertoire est automatiquement démonté s'il n'est pas consulté pendant la durée du délai.

Le format général des cartes est similaire à celui de la carte maîtresse. Toutefois, le champ des options apparaît entre le point de montage et l'emplacement, et non à la fin de l'entrée comme dans la carte maîtresse :

```
mount-point options location
```

Les variables utilisées dans ce format sont les suivantes

mount-point

Il s'agit du point de montage **autofs**. Il peut s'agir d'un simple nom de répertoire pour un montage indirect ou du chemin complet du point de montage pour les montages directs. Chaque clé d'entrée de carte directe et indirecte (*mount-point*) peut être suivie d'une liste de répertoires de décalage séparés par des espaces (noms de sous-répertoires commençant chacun par /), ce qui en fait ce que l'on appelle une entrée de montage multiple.

options

Lorsqu'elles sont fournies, ces options sont ajoutées aux options d'entrée de la carte maîtresse, le cas échéant, ou utilisées à la place des options de la carte maîtresse si l'entrée de configuration **append_options** est définie sur **no**.

location

Il s'agit de l'emplacement du système de fichiers, tel qu'un chemin d'accès au système de fichiers local (précédé du caractère d'échappement du format de carte Sun : pour les noms de cartes commençant par /), un système de fichiers NFS ou tout autre emplacement de système de fichiers valide.

Exemple 26.2. Un fichier de cartes

Voici un exemple de fichier de carte ; par exemple, `/etc/auto.misc`:

```
payroll -fstype=nfs4 personnel:/exports/payroll
sales -fstype=xfstools /dev/hda4
```

La première colonne du fichier map indique le point de montage **autofs**: **sales** et **payroll** à partir du serveur appelé **personnel**. La deuxième colonne indique les options pour le montage de **autofs**. La troisième colonne indique la source du montage.

Selon la configuration donnée, les points de montage **autofs** seront **/home/payroll** et **/home/sales**. L'option **-fstype=** est souvent omise et n'est pas nécessaire si le système de fichiers est NFS, y compris les montages pour NFSv4 si la valeur par défaut du système est NFSv4 pour les montages NFS.

En utilisant la configuration donnée, si un processus a besoin d'accéder à un répertoire non monté **autofs** tel que **/home/payroll/2006/July.sxc**, le service **autofs** monte automatiquement le répertoire.

Le format des cartes amd

Le service **autofs** reconnaît également la configuration des cartes au format **amd**. Ceci est utile si vous souhaitez réutiliser la configuration existante des compteurs automatiques écrite pour le service **am-utils**, qui a été supprimé de Red Hat Enterprise Linux.

Cependant, Red Hat recommande d'utiliser le format plus simple **autofs** décrit dans les sections précédentes.

Ressources supplémentaires

- **autofs(5)** page de manuel
- **autofs.conf(5)** page de manuel
- **auto.master(5)** page de manuel
- `/usr/share/doc/autofs/README.amd-maps` fichier

26.3. CONFIGURATION DES POINTS DE MONTAGE AUTOFS

Cette procédure décrit comment configurer des points de montage à la demande à l'aide du service **autofs**.

Conditions préalables

- Installez le paquetage **autofs**:

```
# dnf install autofs
```
- Démarrez et activez le service **autofs**:

```
# systemctl enable --now autofs
```

Procédure

1. Créez un fichier de carte pour le point de montage à la demande, situé à l'adresse suivante **/etc/auto.identified**. Remplacer *identified* par un nom qui identifie le point de montage.
2. Dans le fichier map, remplissez les champs point de montage, options et emplacement comme décrit dans la section [Fichiers de configuration autofs](#).
3. Enregistrez le fichier de cartographie dans le fichier de cartographie principal, comme décrit dans la section [Fichiers de configuration autofs](#).
4. Autoriser le service à relire la configuration, afin qu'il puisse gérer le montage **autofs** nouvellement configuré :

```
# systemctl reload autofs.service
```

5. Essayez d'accéder au contenu du répertoire à la demande :

```
# ls automounted-directory
```

26.4. MONTAGE AUTOMATIQUE DES RÉPERTOIRES PERSONNELS DES UTILISATEURS DU SERVEUR NFS AVEC LE SERVICE AUTOFS

Cette procédure décrit comment configurer le service **autofs** pour qu'il monte automatiquement les répertoires personnels des utilisateurs.

Conditions préalables

- Le paquet **autofs** est installé.
- Le service **autofs** le service est activé et fonctionne.

Procédure

1. Spécifiez le point de montage et l'emplacement du fichier map en modifiant le fichier **/etc/auto.master** sur un serveur sur lequel vous devez monter les répertoires personnels des utilisateurs. Pour ce faire, ajoutez la ligne suivante au fichier **/etc/auto.master**:

```
/home /etc/auto.home
```

2. Créez un fichier map du nom de **/etc/auto.home** sur un serveur sur lequel vous devez monter les répertoires personnels des utilisateurs, et modifiez le fichier avec les paramètres suivants :

```
* -fstype=nfs,rw,sync host.example.com:/home/&
```

Vous pouvez ignorer **fstype** car il s'agit du paramètre **nfs** par défaut. Pour plus d'informations, voir la page de manuel **autofs(5)**.

3. Rechargez le service **autofs**:

```
# systemctl reload autofs
```

26.5. REMPLACER OU COMPLÉTER LES FICHIERS DE CONFIGURATION DU SITE AUTOFS

Il est parfois utile d'ignorer les valeurs par défaut du site pour un point de montage spécifique sur un système client.

Exemple 26.3. Conditions initiales

Par exemple, considérons les conditions suivantes :

- Les cartes Automounter sont stockées dans NIS et le fichier **/etc/nsswitch.conf** contient la directive suivante :

```
automount: files nis
```

- Le fichier **auto.master** contient

```
+auto.master
```

- Le fichier NIS **auto.master** map contient

```
/home auto.home
```

- La carte NIS **auto.home** contient

```
beth fileserver.example.com:/export/home/beth
joe fileserver.example.com:/export/home/joe
* fileserver.example.com:/export/home/&
```

- L'option de configuration **autofs BROWSE_MODE** est réglée sur **yes**:

```
BROWSE_MODE="yes"
```

- Le fichier map **/etc/auto.home** n'existe pas.

Procédure

Cette section décrit les exemples de montage de répertoires personnels à partir d'un autre serveur et d'ajout d'entrées sélectionnées sur le site **auto.home**.

Exemple 26.4. Montage des répertoires personnels à partir d'un autre serveur

Compte tenu des conditions précédentes, supposons que le système client doit remplacer la carte NIS **auto.home** et monter des répertoires personnels à partir d'un autre serveur.

- Dans ce cas, le client doit utiliser la carte **/etc/auto.master** suivante :

```
/home /etc/auto.home
+auto.master
```

- La carte **/etc/auto.home** contient l'entrée :

```
* host.example.com:/export/home/&
```

Comme le compteur automatique ne traite que la première occurrence d'un point de montage, le répertoire **/home** contient le contenu de **/etc/auto.home** au lieu de la carte NIS **auto.home**.

Exemple 26.5. Augmenter auto.home avec seulement des entrées sélectionnées

Il est également possible d'ajouter quelques entrées à la carte du site **auto.home**:

1. Créez un plan de fichier **/etc/auto.home** et mettez-y les nouvelles entrées. À la fin, incluez la carte NIS **auto.home**. Le plan du fichier **/etc/auto.home** ressemble alors à ce qui suit :

```
mydir someserver:/export/mydir
+auto.home
```

2. Avec ces conditions de la carte NIS **auto.home**, la liste du contenu du répertoire **/home** s'affiche :

```
$ ls /home
beth joe mydir
```

Ce dernier exemple fonctionne comme prévu car **autofs** n'inclut pas le contenu d'un fichier map du même nom que celui qu'il est en train de lire. Ainsi, **autofs** passe à la source de carte suivante dans la configuration **nsswitch**.

26.6. UTILISATION DE LDAP POUR STOCKER DES CARTES DE COMPTEURS AUTOMATIQUES

Cette procédure configure **autofs** pour qu'il stocke les cartes de compteurs automatiques dans la configuration LDAP plutôt que dans les fichiers de cartes **autofs**.

Conditions préalables

- Les bibliothèques client LDAP doivent être installées sur tous les systèmes configurés pour récupérer des cartes de compteurs automatiques à partir de LDAP. Sur Red Hat Enterprise Linux, le paquetage **openldap** devrait être installé automatiquement en tant que dépendance du paquetage **autofs**.

Procédure

1. Pour configurer l'accès LDAP, modifiez le fichier **/etc/openldap/ldap.conf**. Assurez-vous que les options **BASE**, **URI**, et **schema** sont définies de manière appropriée pour votre site.
2. Le schéma le plus récent pour le stockage des cartes de montages automatiques dans LDAP est décrit dans le projet **rfc2307bis**. Pour utiliser ce schéma, définissez-le dans le fichier de configuration **/etc/autofs.conf** en supprimant les caractères de commentaire de la définition du schéma. Par exemple :

Exemple 26.6. Configuration d'autofs

```
DEFAULT_MAP_OBJECT_CLASS="automountMap"
DEFAULT_ENTRY_OBJECT_CLASS="automount"
DEFAULT_MAP_ATTRIBUTE="automountMapName"
```

```

DEFAULT_ENTRY_ATTRIBUTE="automountKey"
DEFAULT_VALUE_ATTRIBUTE="automountInformation"

```

3. Veillez à ce que toutes les autres entrées de schéma soient commentées dans la configuration. L'attribut **automountKey** du schéma **rfc2307bis** remplace l'attribut **cn** du schéma **rfc2307**. Voici un exemple de configuration du format d'échange de données LDAP (LDIF) :

Exemple 26.7. Configuration LDIF

```

# auto.master, example.com
dn: automountMapName=auto.master,dc=example,dc=com
objectClass: top
objectClass: automountMap
automountMapName: auto.master

# /home, auto.master, example.com
dn: automountMapName=auto.master,dc=example,dc=com
objectClass: automount
automountKey: /home
automountInformation: auto.home

# auto.home, example.com
dn: automountMapName=auto.home,dc=example,dc=com
objectClass: automountMap
automountMapName: auto.home

# 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/&

```

Ressources supplémentaires

- [Le projet rfc2307bis](#)

26.7. UTILISER SYSTEMD.AUTOMOUNT POUR MONTER UN SYSTÈME DE FICHIERS À LA DEMANDE AVEC /ETC/FSTAB

Cette procédure montre comment monter un système de fichiers à la demande en utilisant les unités automount de systemd lorsque le point de montage est défini dans **/etc/fstab**. Vous devez ajouter une unité automount pour chaque montage et l'activer.

Procédure

1. Ajoutez l'entrée fstab souhaitée comme indiqué au [chapitre 30. Montage persistant de systèmes de fichiers](#). Par exemple :

```
/dev/disk/by-id/da875760-edb9-4b82-99dc-5f4b1ff2e5f4 /mount/point xfs defaults 0 0
```

2. Ajoutez **x-systemd.automount** au champ d'entrée des options créé à l'étape précédente.
3. Chargez les unités nouvellement créées pour que votre système enregistre la nouvelle configuration :

```
# systemctl daemon-reload
```

4. Démarrer l'unité automount :

```
# systemctl start mount-point.automount
```

Vérification

1. Vérifiez que **mount-point.automount** fonctionne :

```
# systemctl status mount-point.automount
```

2. Vérifier que le répertoire monté automatiquement a le contenu souhaité :

```
# ls /mount/point
```

Ressources supplémentaires

- **systemd.automount(5)** page de manuel.
- **systemd.mount(5)** page de manuel.
- [Introduction à systemd](#).

26.8. UTILISATION DE SYSTEMD.AUTOMOUNT POUR MONTER UN SYSTÈME DE FICHIERS À LA DEMANDE AVEC UNE UNITÉ DE MONTAGE

Cette procédure montre comment monter un système de fichiers à la demande en utilisant les unités automount de systemd lorsque le point de montage est défini par une unité de montage. Vous devez ajouter une unité automount pour chaque montage et l'activer.

Procédure

1. Créez une unité de montage. Par exemple :

```
mount-point.mount
[Mount]
What=/dev/disk/by-uuid/f5755511-a714-44c1-a123-cfde0e4ac688
Where=/mount/point
Type=xfs
```

2. Créer un fichier d'unité avec le même nom que l'unité de montage, mais avec l'extension **.automount**.
3. Ouvrez le fichier et créez une section **[Automount]**. Définissez l'option **Where=** avec le chemin de montage :

```
[Automount]
Where=/mount/point
[Install]
WantedBy=multi-user.target
```

4. Chargez les unités nouvellement créées pour que votre système enregistre la nouvelle configuration :

```
# systemctl daemon-reload
```

5. Activez et démarrez l'unité automount à la place :

```
# systemctl enable --now mount-point.automount
```

Vérification

1. Vérifiez que **mount-point.automount** fonctionne :

```
# systemctl status mount-point.automount
```

2. Vérifier que le répertoire monté automatiquement a le contenu souhaité :

```
# ls /mount/point
```

Ressources supplémentaires

- **systemd.automount(5)** page de manuel.
- **systemd.mount(5)** page de manuel.
- [Introduction à systemd](#).

CHAPITRE 27. UTILISATION DU COMPOSANT SSSD DE L'IDM POUR METTRE EN CACHE LES CARTES AUTOFS

Le System Security Services Daemon (SSSD) est un service système qui permet d'accéder aux répertoires de services distants et aux mécanismes d'authentification. La mise en cache des données est utile en cas de connexion réseau lente. Pour configurer le service SSSD afin qu'il mette en cache la carte autofs, suivez les procédures ci-dessous dans cette section.

27.1. CONFIGURATION MANUELLE D'AUTOFS POUR UTILISER LE SERVEUR IDM COMME SERVEUR LDAP

Cette procédure montre comment configurer **autofs** pour utiliser le serveur IdM comme serveur LDAP.

Procédure

1. Modifiez le fichier **/etc/autofs.conf** pour spécifier les attributs du schéma que **autofs** doit rechercher :

```
#
# Other common LDAP naming
#
map_object_class = "automountMap"
entry_object_class = "automount"
map_attribute = "automountMapName"
entry_attribute = "automountKey"
value_attribute = "automountInformation"
```



NOTE

L'utilisateur peut écrire les attributs en minuscules et en majuscules dans le fichier **/etc/autofs.conf**.

2. En option, spécifiez la configuration LDAP. Il y a deux façons de le faire. La plus simple consiste à laisser le service automount découvrir lui-même le serveur et les emplacements LDAP :

```
ldap_uri = "ldap:///dc=example,dc=com"
```

This option requires DNS to contain SRV records for the discoverable servers.

Il est également possible de définir explicitement le serveur LDAP à utiliser et le DN de base pour les recherches LDAP :

```
ldap_uri = "ldap://ipa.example.com"
search_base = "cn=location,cn=automount,dc=example,dc=com"
```

3. Modifiez le fichier **/etc/autofs_ldap_auth.conf** pour que autofs autorise l'authentification du client avec le serveur LDAP IdM.
 - Remplacer **authrequired** par oui.

- Définissez le principal comme étant le principal de l'hôte Kerberos pour le serveur LDAP IdM, *host/fqdn@REALM*. Le nom du principal est utilisé pour se connecter à l'annuaire IdM dans le cadre de l'authentification du client GSS.

```
<autofs_ldap_sasl_conf
  usetls="no"
  tlsrequired="no"
  authrequired="yes"
  authtype="GSSAPI"
  clientprinc="host/server.example.com@EXAMPLE.COM"
/>
```

Pour plus d'informations sur l'hôte principal, voir [Utilisation de noms d'hôtes DNS canonisés dans IdM](#).

Si nécessaire, exécutez **klist -k** pour obtenir les informations exactes sur le principal de l'hôte.

27.2. CONFIGURATION DE SSSD POUR LA MISE EN CACHE DES CARTES AUTOFS

Le service SSSD peut être utilisé pour mettre en cache les cartes **autofs** stockées sur un serveur IdM sans avoir à configurer **autofs** pour utiliser le serveur IdM.

Conditions préalables

- Le paquet **sssd** est installé.

Procédure

1. Ouvrez le fichier de configuration SSSD :

```
# vim /etc/sss/sss.conf
```

2. Ajoutez le service **autofs** à la liste des services gérés par SSSD.

```
[sss]
domains = ldap
services = nss,pam,autofs
```

3. Créez une nouvelle section **[autofs]**. Vous pouvez laisser ce champ vide, car les paramètres par défaut d'un service **autofs** fonctionnent avec la plupart des infrastructures.

```
[nss]

[pam]

[sudo]

[autofs]

[ssh]

[pac]
```

-
- Pour plus d'informations, voir la page de manuel **sssd.conf**.
- 4. Optionnellement, définir une base de recherche pour les entrées **autofs**. Par défaut, il s'agit de la base de recherche LDAP, mais un sous-arbre peut être spécifié dans le paramètre **ldap_autofs_search_base**.

```
[domain/EXAMPLE]
```

```
ldap_search_base = "dc=example,dc=com"
```

```
ldap_autofs_search_base = "ou=automount,dc=example,dc=com"
```

- 5. Redémarrer le service SSSD :

```
# systemctl restart sssd.service
```

- 6. Vérifiez le fichier **/etc/nsswitch.conf**, afin que SSSD soit listé comme source pour la configuration de l'automount :

```
automount : sss files
```

- 7. Redémarrer le service **autofs**:

```
# systemctl restart autofs.service
```

- 8. Testez la configuration en listant le répertoire **/home** d'un utilisateur, en supposant qu'il existe une entrée de carte maîtresse pour **/home**:

```
# ls /home/userName
```

Si le système de fichiers distant n'est pas monté, vérifiez que le fichier **/var/log/messages** ne contient pas d'erreurs. Si nécessaire, augmentez le niveau de débogage dans le fichier **/etc/sysconfig/autofs** en définissant le paramètre **logging** sur **debug**.

CHAPITRE 28. DÉFINITION DES AUTORISATIONS DE LECTURE SEULE POUR LE SYSTÈME DE FICHIERS RACINE

Il est parfois nécessaire de monter le système de fichiers racine (/) avec des autorisations de lecture seule. Il s'agit par exemple de renforcer la sécurité ou de garantir l'intégrité des données après une mise hors tension inattendue du système.

28.1. FICHIERS ET RÉPERTOIRES QUI CONSERVENT TOUJOURS LES DROITS D'ÉCRITURE

Pour que le système fonctionne correctement, certains fichiers et répertoires doivent conserver des droits d'écriture. Lorsque le système de fichiers racine est monté en mode lecture seule, ces fichiers sont montés en RAM à l'aide du système de fichiers temporaires **tmpfs**.

L'ensemble par défaut de ces fichiers et répertoires est lu à partir du fichier **/etc/rwtab**. Notez que le paquetage **readonly-root** est nécessaire pour que ce fichier soit présent dans votre système.

```
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>
```

Les entrées du fichier **/etc/rwtab** suivent ce format :

```
copy-method path
```

Dans cette syntaxe :

- Remplacer *copy-method* par l'un des mots-clés spécifiant comment le fichier ou le répertoire est copié dans tmpfs.
- Remplacer *path* par le chemin d'accès au fichier ou au répertoire.

Le fichier **/etc/rwtab** reconnaît les façons suivantes de copier un fichier ou un répertoire sur **tmpfs**:

empty

Un chemin vide est copié sur **tmpfs**. Par exemple :

```
empty /tmp
```

dirs

Une arborescence de répertoires est copiée sur **tmpfs**, vide. Par exemple :

```
dirs /var/run
```

files

Un fichier ou une arborescence de répertoires est copié intact sur **tmpfs**. Par exemple, un fichier ou une arborescence est copié(e) sur intact(e) :

```
files /etc/resolv.conf
```

Le même format s'applique lors de l'ajout de chemins d'accès personnalisés à **/etc/rwtab.d/**.

28.2. CONFIGURER LE SYSTÈME DE FICHIERS RACINE POUR QU'IL SOIT MONTÉ AVEC DES AUTORISATIONS DE LECTURE SEULE AU DÉMARRAGE

Avec cette procédure, le système de fichiers racine est monté en lecture seule lors de tous les démarrages suivants.

Procédure

1. Dans le fichier **/etc/sysconfig/readonly-root**, l'option **READONLY** est remplacée par **yes**:

```
# Set to 'yes' to mount the file systems as read-only.
READONLY=yes
```

2. Ajouter l'option **ro** dans l'entrée racine (**/**) du fichier **/etc/fstab**:

```
/dev/mapper/luks-c376919e... / xfs x-systemd.device-timeout=0,ro 1 1
```

3. Activer l'option **ro** kernel :

```
# grubby --update-kernel=ALL --args="ro"
```

4. Assurez-vous que l'option **rw** kernel est désactivée :

```
# grubby --update-kernel=ALL --remove-args="rw"
```

5. Si vous devez ajouter des fichiers et des répertoires à monter avec des droits d'écriture dans le système de fichiers **tmpfs**, créez un fichier texte dans le répertoire **/etc/rwtab.d/** et mettez-y la configuration.

Par exemple, pour monter le fichier **/etc/example/file** avec des droits d'écriture, ajoutez cette ligne au fichier **/etc/rwtab.d/example**:

```
files /etc/example/file
```



IMPORTANT

Les modifications apportées aux fichiers et aux répertoires sur **tmpfs** ne persistent pas d'un démarrage à l'autre.

6. Redémarrez le système pour appliquer les modifications.

Résolution de problèmes

- Si vous montez par erreur le système de fichiers racine avec des autorisations de lecture seule, vous pouvez le remonter avec des autorisations de lecture et d'écriture à l'aide de la commande suivante :

```
# mount -o remount,rw /
```

CHAPITRE 29. LIMITER L'UTILISATION DE L'ESPACE DE STOCKAGE SUR XFS AVEC DES QUOTAS

Vous pouvez limiter l'espace disque disponible pour les utilisateurs ou les groupes en mettant en place des quotas de disque. Vous pouvez également définir un niveau d'alerte à partir duquel les administrateurs système sont informés avant qu'un utilisateur ne consomme trop d'espace disque ou qu'une partition ne soit saturée.

Le sous-système de quotas XFS gère les limites d'utilisation de l'espace disque (blocs) et des fichiers (inodes). Les quotas XFS contrôlent l'utilisation de ces éléments au niveau d'un utilisateur, d'un groupe, d'un répertoire ou d'un projet, ou en rendent compte. Les quotas de groupe et de projet ne s'excluent mutuellement que sur les anciens formats de disque XFS non définis par défaut.

Lors de la gestion par répertoire ou par projet, XFS gère l'utilisation du disque des hiérarchies de répertoires associées à un projet spécifique.

29.1. QUOTAS DE DISQUES

Dans la plupart des environnements informatiques, l'espace disque n'est pas infini. Le sous-système des quotas fournit un mécanisme de contrôle de l'utilisation de l'espace disque.

Vous pouvez configurer des quotas de disque pour les utilisateurs individuels ainsi que pour les groupes d'utilisateurs sur les systèmes de fichiers locaux. Cela permet de gérer l'espace alloué aux fichiers spécifiques à l'utilisateur (tels que le courrier électronique) séparément de l'espace alloué aux projets sur lesquels un utilisateur travaille. Le sous-système de quotas avertit les utilisateurs lorsqu'ils dépassent la limite qui leur a été attribuée, mais leur laisse un peu d'espace supplémentaire pour le travail en cours (limite dure/limite souple).

Si des quotas sont mis en place, vous devez vérifier si les quotas sont dépassés et vous assurer que les quotas sont exacts. Si les utilisateurs dépassent leurs quotas de façon répétée ou atteignent régulièrement leurs limites, l'administrateur système peut soit aider l'utilisateur à déterminer comment utiliser moins d'espace disque, soit augmenter le quota de disque de l'utilisateur.

Vous pouvez fixer des quotas pour contrôler :

- Nombre de blocs de disque consommés.
- Le nombre d'inodes, qui sont des structures de données contenant des informations sur les fichiers dans les systèmes de fichiers UNIX. Comme les inodes stockent des informations relatives aux fichiers, cela permet de contrôler le nombre de fichiers pouvant être créés.

29.2. L'OUTIL `xfs_quota`

Vous pouvez utiliser l'outil `xfs_quota` pour gérer les quotas sur les systèmes de fichiers XFS. En outre, vous pouvez utiliser les systèmes de fichiers XFS dont l'application des limites est désactivée comme système efficace de comptabilisation de l'utilisation du disque.

Le système de quotas XFS diffère des autres systèmes de fichiers à plusieurs égards. Le plus important est que XFS considère les informations sur les quotas comme des métadonnées du système de fichiers et utilise la journalisation pour fournir une garantie de cohérence de plus haut niveau.

Ressources supplémentaires

- `xfs_quota(8)` page de manuel.

29.3. GESTION DES QUOTAS DANS LE SYSTÈME DE FICHIERS XFS

Le sous-système de quotas XFS gère les limites d'utilisation de l'espace disque (blocs) et des fichiers (inodes). Les quotas XFS contrôlent l'utilisation de ces éléments au niveau d'un utilisateur, d'un groupe, d'un répertoire ou d'un projet, ou en rendent compte. Les quotas de groupe et de projet ne s'excluent mutuellement que sur les anciens formats de disque XFS non définis par défaut.

Lors de la gestion par répertoire ou par projet, XFS gère l'utilisation du disque des hiérarchies de répertoires associées à un projet spécifique.

29.4. ACTIVATION DES QUOTAS DE DISQUE POUR XFS

Cette procédure permet d'activer les quotas de disque pour les utilisateurs, les groupes et les projets sur un système de fichiers XFS. Une fois les quotas activés, l'outil **xfs_quota** peut être utilisé pour définir des limites et établir des rapports sur l'utilisation des disques.

Procédure

1. Activer les quotas pour les utilisateurs :

```
# mount -o uquota /dev/xvdb1 /xfs
```

Remplacer **uquota** par **uqnoenforce** pour permettre la déclaration d'utilisation sans imposer de limites.

2. Activer les quotas pour les groupes :

```
# mount -o gquota /dev/xvdb1 /xfs
```

Remplacer **gquota** par **gqnoenforce** pour permettre la déclaration d'utilisation sans imposer de limites.

3. Activer les quotas pour les projets :

```
# mount -o pquota /dev/xvdb1 /xfs
```

Remplacer **pquota** par **pqnoenforce** pour permettre la déclaration d'utilisation sans imposer de limites.

4. Vous pouvez également inclure les options de montage des quotas dans le fichier **/etc/fstab**. L'exemple suivant montre des entrées dans le fichier **/etc/fstab** pour activer les quotas pour les utilisateurs, les groupes et les projets, respectivement, sur un système de fichiers XFS. Ces exemples montent également le système de fichiers avec des autorisations de lecture/écriture :

```
# 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
```

Ressources supplémentaires

- **mount(8)** page de manuel.
- **xfs_quota(8)** page de manuel.

29.5. RAPPORT SUR L'UTILISATION DE XFS

Vous pouvez utiliser l'outil **xfs_quota** pour fixer des limites et établir des rapports sur l'utilisation du disque. Par défaut, **xfs_quota** est exécuté de manière interactive et en mode basique. Les sous-commandes du mode de base se contentent de signaler l'utilisation et sont accessibles à tous les utilisateurs.

Conditions préalables

- Les quotas ont été activés pour le système de fichiers XFS. Voir [Activation des quotas de disque pour XFS](#).

Procédure

1. Démarrer le shell **xfs_quota**:

```
# xfs_quota
```

2. Affiche l'utilisation et les limites pour l'utilisateur donné :

```
# xfs_quota> quota username
```

3. Affiche le nombre de blocs et d'inodes libres et utilisés :

```
# xfs_quota> df
```

4. Exécutez la commande help pour afficher les commandes de base disponibles sur **xfs_quota**.

```
# xfs_quota> help
```

5. Spécifiez **q** pour quitter **xfs_quota**.

```
# xfs_quota> q
```

Ressources supplémentaires

- **xfs_quota(8)** page de manuel.

29.6. MODIFICATION DES LIMITES DE QUOTAS XFS

Lancez l'outil **xfs_quota** avec l'option **-x** pour activer le mode expert et exécuter les commandes de l'administrateur, qui permettent de modifier le système de quotas. Les sous-commandes de ce mode permettent de configurer réellement les limites et ne sont accessibles qu'aux utilisateurs disposant de privilèges élevés.

Conditions préalables

- Les quotas ont été activés pour le système de fichiers XFS. Voir [Activation des quotas de disque pour XFS](#).

Procédure

1. Lancez le shell **xfs_quota** avec l'option **-x** pour activer le mode expert :

```
# xfs_quota -x
```

2. Rapporter les informations sur les quotas pour un système de fichiers spécifique :

```
# xfs_quota> report /path
```

Par exemple, pour afficher un exemple de rapport de quotas pour **/home** (sur **/dev/blockdevice**), utilisez la commande **report -h /home**. Cette commande affiche une sortie similaire à la suivante :

```
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. Modifier les limites de quotas :

```
# xfs_quota> limit isoft=500m ihard=700m user /path
```

Par exemple, pour définir une limite douce et dure de 500 et 700 inodes respectivement pour l'utilisateur **john**, dont le répertoire personnel est **/home/john**, utilisez la commande suivante :

```
# xfs_quota -x -c 'limit isoft=500 ihard=700 john' /home/
```

Dans ce cas, passez **mount_point** qui est le système de fichiers xfs monté.

4. Exécutez la commande **help** pour afficher les commandes expertes disponibles sur **xfs_quota -x**:

```
# xfs_quota> help
```

Ressources supplémentaires

- **xfs_quota(8)** page de manuel.

29.7. DÉFINITION DES LIMITES DU PROJET POUR XFS

Cette procédure permet de configurer les limites des répertoires contrôlés par le projet.

Procédure

1. Ajoutez les répertoires contrôlés par le projet à **/etc/projects**. Par exemple, la procédure suivante ajoute le chemin **/var/log** avec un identifiant unique de 11 à **/etc/projects**. Votre identifiant de projet peut être n'importe quelle valeur numérique associée à votre projet.

```
# echo 11:/var/log >> /etc/projects
```

2. Ajoutez des noms de projet à **/etc/projid** pour associer des ID de projet à des noms de projet. Par exemple, l'exemple suivant associe un projet appelé **logfiles** à l'identifiant de projet 11 tel que défini à l'étape précédente.

```
# echo logfile:11 >> /etc/projid
```

3. Initialiser le répertoire du projet. Par exemple, la procédure suivante initialise le répertoire du projet **/var**:

```
# xfs_quota -x -c 'project -s logfile' /var
```

4. Configurer les quotas pour les projets avec des répertoires initialisés :

```
# xfs_quota -x -c 'limit -p bhard=1g logfile' /var
```

Ressources supplémentaires

- **xfs_quota(8)** page de manuel.
- **projid(5)** page de manuel.
- **projects(5)** page de manuel.

CHAPITRE 30. LIMITER L'UTILISATION DE L'ESPACE DE STOCKAGE SUR EXT4 AVEC DES QUOTAS

Vous devez activer les quotas de disque sur votre système avant de pouvoir les attribuer. Vous pouvez attribuer des quotas de disque par utilisateur, par groupe ou par projet. Toutefois, si une limite souple est définie, vous pouvez dépasser ces quotas pendant une période configurable, appelée délai de grâce.

30.1. INSTALLATION DE L'OUTIL QUOTA

Vous devez installer le paquetage RPM **quota** pour mettre en œuvre les quotas de disque.

Procédure

- Installez le paquetage **quota**:

```
# dnf install quota
```

30.2. ACTIVATION DE LA FONCTION DE QUOTA LORS DE LA CRÉATION D'UN SYSTÈME DE FICHIERS

Cette procédure décrit comment activer les quotas lors de la création d'un système de fichiers.

Procédure

1. Activer les quotas lors de la création d'un système de fichiers :

```
# mkfs.ext4 -O quota /dev/sda
```



NOTE

Seuls les quotas d'utilisateurs et de groupes sont activés et initialisés par défaut.

2. Modifier les valeurs par défaut lors de la création d'un système de fichiers :

```
# mkfs.ext4 -O quota -E quotatype=usrquota:grpquota:prjquota /dev/sda
```

3. Monter le système de fichiers :

```
# mount /dev/sda
```

Ressources supplémentaires

- **ext4(5)** page de manuel.

30.3. ACTIVATION DE LA FONCTION DE QUOTA SUR LES SYSTÈMES DE FICHIERS EXISTANTS

Cette procédure décrit comment activer la fonction de quota sur un système de fichiers existant à l'aide de la commande **tune2fs**.

Procédure

1. Démonter le système de fichiers :

```
# umount /dev/sda
```

2. Activer les quotas sur le système de fichiers existant :

```
# tune2fs -O quota /dev/sda
```



NOTE

Seuls les quotas d'utilisateurs et de groupes sont initialisés par défaut.

3. Modifier les valeurs par défaut :

```
# tune2fs -Q usrquota,grpquota,prjquota /dev/sda
```

4. Monter le système de fichiers :

```
# mount /dev/sda
```

Ressources supplémentaires

- **ext4(5)** page de manuel.

30.4. ACTIVATION DE L'APPLICATION DES QUOTAS

La comptabilisation des quotas est activée par défaut après le montage du système de fichiers sans aucune option supplémentaire, mais l'application des quotas ne l'est pas.

Conditions préalables

- La fonction de quota est activée et les quotas par défaut sont initialisés.

Procédure

- Activer l'application des quotas par **quotaon** pour le quota de l'utilisateur :

```
# mount /dev/sda /mnt
```

```
# quotaon /mnt
```



NOTE

L'application des quotas peut être activée au moment du montage à l'aide des options de montage **usrquota**, **grpquota**, ou **prjquota**.

```
# mount -o usrquota,grpquota,prjquota /dev/sda /mnt
```

- Activez les quotas d'utilisateurs, de groupes et de projets pour tous les systèmes de fichiers :

```
# quotaon -vaugP
```

- Si aucune des options **-u**, **-g** ou **-P** n'est spécifiée, seuls les quotas d'utilisateurs sont activés.
- Si seule l'option **-g** est spécifiée, seuls les quotas de groupe sont activés.
- Si l'option **-P** est spécifiée, seuls les quotas de projet sont activés.

- Activer les quotas pour un système de fichiers spécifique, tel que **/home**:

```
# quotaon -vugP /home
```

Ressources supplémentaires

- **quotaon(8)** page de manuel.

30.5. ATTRIBUTION DE QUOTAS PAR UTILISATEUR

Les quotas de disque sont attribués aux utilisateurs à l'aide de la commande **edquota**.



NOTE

L'éditeur de texte défini par la variable d'environnement **EDITOR** est utilisé par **edquota**. Pour changer d'éditeur, définissez la variable d'environnement **EDITOR** dans votre fichier **~/.bash_profile** avec le chemin complet de l'éditeur de votre choix.

Conditions préalables

- L'utilisateur doit exister avant la définition du quota d'utilisateurs.

Procédure

1. Attribuer un quota à un utilisateur :

```
# edquota username
```

Remplacez *username* par l'utilisateur auquel vous souhaitez attribuer les quotas.

Par exemple, si vous activez un quota pour la partition **/dev/sda** et que vous exécutez la commande **edquota testuser**, le texte suivant s'affiche dans l'éditeur par défaut configuré sur le système :

```
Disk quotas for user testuser (uid 501):
Filesystem blocks soft hard inodes soft hard
/dev/sda 44043 0 0 37418 0 0
```

2. Modifier les limites souhaitées.
Si l'une des valeurs est fixée à 0, la limite n'est pas fixée. Modifiez-les dans l'éditeur de texte.

Par exemple, l'exemple suivant montre que les limites de bloc souple et de bloc dur pour l'utilisateur test ont été fixées à 50000 et 55000 respectivement.

```
Disk quotas for user testuser (uid 501):
Filesystem blocks soft hard inodes soft hard
/dev/sda 44043 50000 55000 37418 0 0
```

- La première colonne est le nom du système de fichiers pour lequel un quota est activé.
- La deuxième colonne indique le nombre de blocs que l'utilisateur utilise actuellement.
- Les deux colonnes suivantes servent à définir des limites de blocs souples et rigides pour l'utilisateur sur le système de fichiers.
- La colonne **inodes** indique le nombre d'inodes que l'utilisateur utilise actuellement.
- Les deux dernières colonnes sont utilisées pour définir les limites d'inodes soft et hard pour l'utilisateur sur le système de fichiers.
 - La limite des blocs durs est la quantité maximale absolue d'espace disque qu'un utilisateur ou un groupe peut utiliser. Une fois cette limite atteinte, aucun espace disque supplémentaire ne peut être utilisé.
 - La limite souple de blocs définit la quantité maximale d'espace disque pouvant être utilisée. Toutefois, contrairement à la limite stricte, la limite souple peut être dépassée pendant un certain temps. Cette période est appelée *grace period*. Le délai de grâce peut être exprimé en secondes, minutes, heures, jours, semaines ou mois.

Verification steps

- Vérifiez que le quota de l'utilisateur a été défini :

```
# 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
```

30.6. ATTRIBUTION DE QUOTAS PAR GROUPE

Vous pouvez attribuer des quotas par groupe.

Conditions préalables

- Le groupe doit exister avant la définition du quota de groupe.

Procédure

1. Définir un quota de groupe :

```
# edquota -g groupname
```

Par exemple, pour définir un quota de groupe pour le groupe **devel**:

```
# edquota -g devel
```

Cette commande affiche le quota existant pour le groupe dans l'éditeur de texte :

```
Disk quotas for group devel (gid 505):
Filesystem blocks soft hard inodes soft hard
/dev/sda 440400 0 0 37418 0 0
```

2. Modifiez les limites et enregistrez le fichier.

Verification steps

- Vérifiez que le quota de groupe est défini :

```
# quota -vg groupname
```

30.7. ATTRIBUTION DE QUOTAS PAR PROJET

Cette procédure permet d'attribuer des quotas par projet.

Conditions préalables

- Le quota de projet est activé sur votre système de fichiers.

Procédure

1. Ajoutez les répertoires contrôlés par le projet à **/etc/projects**. Par exemple, la procédure suivante ajoute le chemin **/var/log** avec un identifiant unique de 11 à **/etc/projects**. Votre identifiant de projet peut être n'importe quelle valeur numérique associée à votre projet.

```
# echo 11:/var/log >> /etc/projects
```

2. Ajoutez des noms de projet à **/etc/projid** pour associer des ID de projet à des noms de projet. Par exemple, l'exemple suivant associe un projet appelé **Logs** à l'identifiant de projet 11 tel que défini à l'étape précédente.

```
# echo Logs:11 >> /etc/projid
```

3. Fixer les limites souhaitées :

```
# edquota -P 11
```



NOTE

Vous pouvez choisir le projet soit par son ID (**11** dans ce cas), soit par son nom (**Logs** dans ce cas).

4. À l'aide de **quotaon**, activez l'application des quotas :
Voir [Activation de l'application des quotas](#) .

Verification steps

- Vérifiez que le quota du projet est défini :

```
# quota -vP 11
```

**NOTE**

Vous pouvez vérifier soit par l'ID du projet, soit par le nom du projet.

Ressources supplémentaires

- **edquota(8)** page de manuel.
- **projid(5)** page de manuel.
- **projects(5)** page de manuel.

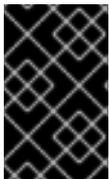
30.8. DÉFINITION DU DÉLAI DE GRÂCE POUR LES LIMITES SOUPLES

Si un quota donné comporte des limites souples, vous pouvez modifier le délai de grâce, c'est-à-dire la durée pendant laquelle une limite souple peut être dépassée. Vous pouvez définir le délai de grâce pour les utilisateurs, les groupes ou les projets.

Procédure

- Modifier le délai de grâce :

```
# edquota -t
```

**IMPORTANT**

Alors que les autres commandes **edquota** agissent sur les quotas d'un utilisateur, d'un groupe ou d'un projet particulier, l'option **-t** agit sur tous les systèmes de fichiers pour lesquels les quotas sont activés.

Ressources supplémentaires

- **edquota(8)** page de manuel.

30.9. DÉSACTIVER LES QUOTAS DE SYSTÈME DE FICHIERS

Utilisez **quotaoff** pour désactiver l'application des quotas de disque sur les systèmes de fichiers spécifiés. La comptabilisation des quotas reste activée après l'exécution de cette commande.

Procédure

- Pour désactiver tous les quotas d'utilisateurs et de groupes :

```
# quotaoff -vaugP
```

- Si aucune des options **-u**, **-g** ou **-P** n'est spécifiée, seuls les quotas d'utilisateurs sont désactivés.
- Si l'option **-g** est spécifiée, seuls les quotas de groupe sont désactivés.
- Si l'option **-P** est spécifiée, seuls les quotas de projet sont désactivés.

- L'option **-v** permet d'afficher des informations d'état détaillées lors de l'exécution de la commande.

Ressources supplémentaires

- **quotaoff(8)** page de manuel.

30.10. RAPPORT SUR LES QUOTAS DE DISQUE

Vous pouvez créer un rapport sur les quotas de disque à l'aide de l'utilitaire **repquota**.

Procédure

1. Exécutez la commande **repquota**:

```
# repquota
```

Par exemple, la commande **repquota /dev/sda** produit le résultat suivant :

```
*** 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. Affichez le rapport d'utilisation du disque pour tous les systèmes de fichiers activés par quota :

```
# repquota -augP
```

Le symbole **--** affiché après chaque utilisateur détermine si les limites de blocs ou d'inodes ont été dépassées. Si l'une ou l'autre de ces limites est dépassée, un caractère **-** apparaît à la place du caractère **-** correspondant. Le premier caractère **-** représente la limite de blocs et le second la limite d'inodes.

Les colonnes **grace** sont normalement vides. Si une limite souple a été dépassée, la colonne contient une spécification temporelle égale à la durée restante du délai de grâce. Si le délai de grâce a expiré, **none** apparaît à sa place.

Ressources supplémentaires

Pour plus d'informations, consultez la page de manuel **repquota(8)**.

CHAPITRE 31. MISE AU REBUT DES BLOCS INUTILISÉS

Vous pouvez effectuer ou planifier des opérations d'annulation sur les périphériques de bloc qui les prennent en charge. L'opération de suppression de blocs communique au stockage sous-jacent les blocs du système de fichiers qui ne sont plus utilisés par le système de fichiers monté. Les opérations d'élimination de blocs permettent aux disques SSD d'optimiser les routines de collecte des déchets et peuvent informer le stockage à provisionnement fin de la réaffectation des blocs physiques inutilisés.

Requirements

- Le périphérique de bloc sous-jacent au système de fichiers doit prendre en charge les opérations d'élimination physique.
Les opérations de rejet physique sont prises en charge si la valeur du fichier `/sys/block/<device>/queue/discard_max_bytes` est différente de zéro.

31.1. TYPES D'OPÉRATIONS D'ANNULATION DE BLOCS

Vous pouvez exécuter des opérations de rejet en utilisant différentes méthodes :

Mise au rebut par lot

Est déclenché explicitement par l'utilisateur et élimine tous les blocs inutilisés dans les systèmes de fichiers sélectionnés.

Mise au rebut en ligne

Est spécifié au moment du montage et se déclenche en temps réel sans intervention de l'utilisateur. Les opérations d'élimination en ligne n'éliminent que les blocs qui passent de l'état **used** à l'état **free**.

Rejet périodique

Il s'agit d'opérations par lots exécutées régulièrement par un service **systemd**.

Tous les types sont pris en charge par les systèmes de fichiers XFS et ext4.

Recommandations

Red Hat vous recommande d'utiliser l'élimination par lots ou périodique.

N'utilisez la fonction de rejet en ligne que si

- la charge de travail du système est telle que l'élimination par lots n'est pas réalisable, ou
- les opérations d'élimination en ligne sont nécessaires pour maintenir les performances.

31.2. EXÉCUTION DE L'ÉLIMINATION DES BLOCS PAR LOTS

Vous pouvez effectuer une opération de suppression de blocs par lots pour supprimer les blocs inutilisés d'un système de fichiers monté.

Conditions préalables

- Le système de fichiers est monté.
- Le périphérique de bloc qui sous-tend le système de fichiers prend en charge les opérations d'élimination physique.

Procédure

- Utilisez l'utilitaire **fstrim**:
 - Pour effectuer un rejet uniquement sur un système de fichiers sélectionné, utilisez :

```
# fstrim mount-point
```

- Pour effectuer un rejet sur tous les systèmes de fichiers montés, utilisez :

```
# fstrim --all
```

Si vous exécutez la commande **fstrim** sur :

- un appareil qui ne prend pas en charge les opérations de mise au rebut, ou
- un dispositif logique (LVM ou MD) composé de plusieurs dispositifs, dont l'un d'entre eux ne prend pas en charge les opérations de mise au rebut,

le message suivant s'affiche :

```
# fstrim /mnt/non_discard
```

```
fstrim: /mnt/non_discard: the discard operation is not supported
```

Ressources supplémentaires

- **fstrim(8)** page de manuel.

31.3. ACTIVATION DE L'ÉLIMINATION DES BLOCS EN LIGNE

Vous pouvez effectuer des opérations de suppression de blocs en ligne pour supprimer automatiquement les blocs inutilisés sur tous les systèmes de fichiers pris en charge.

Procédure

- Activer le rejet en ligne au moment du montage :
 - Lors du montage manuel d'un système de fichiers, ajoutez l'option **-o discard** mount :

```
# mount -o discard device mount-point
```

- Pour monter un système de fichiers de manière persistante, ajoutez l'option **discard** à l'entrée mount dans le fichier **/etc/fstab**.

Ressources supplémentaires

- **mount(8)** page de manuel.
- **fstab(5)** page de manuel.

31.4. ACTIVATION DE L'ÉLIMINATION PÉRIODIQUE DES BLOCS

Vous pouvez activer une minuterie **systemd** pour éliminer régulièrement les blocs inutilisés sur tous les systèmes de fichiers pris en charge.

Procédure

- Active et démarre la minuterie **systemd**:

```
# systemctl enable --now fstrim.timer  
Created symlink /etc/systemd/system/timers.target.wants/fstrim.timer →  
/usr/lib/systemd/system/fstrim.timer.
```

Vérification

- Vérifier l'état de la minuterie :

```
# systemctl status fstrim.timer  
fstrim.timer - Discard unused blocks once a week  
Loaded: loaded (/usr/lib/systemd/system/fstrim.timer; enabled; vendor preset: disabled)  
Active: active (waiting) since Wed 2023-05-17 13:24:41 CEST; 3min 15s ago  
Trigger: Mon 2023-05-22 01:20:46 CEST; 4 days left  
Docs: man:fstrim  
  
May 17 13:24:41 localhost.localdomain systemd[1]: Started Discard unused blocks once a  
week.
```

CHAPITRE 32. CONFIGURATION DES SYSTÈMES DE FICHIERS STRATIS

Stratis fonctionne en tant que service pour gérer des pools de périphériques de stockage physique, simplifiant la gestion du stockage local avec une grande facilité d'utilisation tout en vous aidant à mettre en place et à gérer des configurations de stockage complexes.



IMPORTANT

Stratis est une fonctionnalité d'aperçu technologique uniquement. Les fonctionnalités de l'aperçu technologique ne sont pas prises en charge par les accords de niveau de service (SLA) de production de Red Hat et peuvent ne pas être complètes sur le plan fonctionnel. Red Hat ne recommande pas de les utiliser en production. Ces fonctionnalités offrent un accès anticipé aux fonctionnalités des produits à venir, ce qui permet aux clients de tester les fonctionnalités et de fournir un retour d'information pendant le processus de développement. Pour plus d'informations sur l'étendue de l'assistance des fonctionnalités Red Hat Technology Preview, consultez <https://access.redhat.com/support/offerings/techpreview>.

32.1. QU'EST-CE QUE STRATIS ?

Stratis est une solution de gestion du stockage local pour Linux. Elle est axée sur la simplicité et la facilité d'utilisation, et vous donne accès à des fonctions de stockage avancées.

Stratis facilite les activités suivantes :

- Configuration initiale du stockage
- Modifications ultérieures
- Utilisation des fonctions de stockage avancées

Stratis est un système hybride de gestion du stockage local par l'utilisateur et le noyau qui prend en charge des fonctions de stockage avancées. Le concept central de Stratis est un pool de stockage *pool*. Ce pool est créé à partir d'un ou plusieurs disques locaux ou partitions, et les volumes sont créés à partir du pool.

La piscine offre de nombreuses fonctionnalités utiles, telles que

- Instantanés de systèmes de fichiers
- Provisionnement fin
- Tiercé

Ressources supplémentaires

- [Site web de Stratis](#)

32.2. COMPOSANTS D'UN VOLUME STRATIS

Découvrez les composants d'un volume Stratis.

En externe, Stratis présente les composants de volume suivants dans l'interface de ligne de commande et l'API :

blockdev

Périphériques en bloc, tels qu'un disque ou une partition de disque.

pool

Composé d'un ou plusieurs dispositifs de blocage.

Un pool a une taille totale fixe, égale à la taille des blocs.

Le pool contient la plupart des couches Stratis, telles que le cache de données non volatiles utilisant la cible **dm-cache**.

Stratis crée un répertoire **/dev/stratis/my-pool/** pour chaque pool. Ce répertoire contient des liens vers les périphériques qui représentent les systèmes de fichiers Stratis dans le pool.

filesystem

Chaque pool peut contenir un ou plusieurs systèmes de fichiers, qui stockent des fichiers.

Les systèmes de fichiers sont finement provisionnés et n'ont pas une taille totale fixe. La taille réelle d'un système de fichiers augmente avec les données qui y sont stockées. Si la taille des données approche la taille virtuelle du système de fichiers, Stratis augmente automatiquement le volume fin et le système de fichiers.

Les systèmes de fichiers sont formatés avec XFS.



IMPORTANT

Stratis suit des informations sur les systèmes de fichiers créés à l'aide de Stratis que XFS ne connaît pas, et les modifications apportées à l'aide de XFS ne créent pas automatiquement de mises à jour dans Stratis. Les utilisateurs ne doivent pas reformater ou reconfigurer les systèmes de fichiers XFS qui sont gérés par Stratis.

Stratis crée des liens vers les systèmes de fichiers au niveau du **/dev/stratis/my-pool/my-fs** chemin.



NOTE

Stratis utilise de nombreux dispositifs Device Mapper, qui apparaissent dans les listes **dmsetup** et le fichier **/proc/partitions**. De même, la sortie de la commande **lsblk** reflète le fonctionnement interne et les couches de Stratis.

32.3. DISPOSITIFS DE BLOCAGE UTILISABLES AVEC STRATIS

Périphériques de stockage pouvant être utilisés avec Stratis.

Dispositifs pris en charge

Les piscines Stratis ont été testées pour fonctionner sur ces types de blocs :

- LUKS
- Volumes logiques LVM
- MD RAID

- DM Multipath
- iSCSI
- Disques durs et disques SSD
- NVMe devices

Dispositifs non pris en charge

Étant donné que Stratis contient une couche de provisionnement fin, Red Hat ne recommande pas de placer un pool Stratis sur des périphériques de bloc qui sont déjà provisionnés de manière fine.

32.4. INSTALLATION DE STRATIS

Installez les paquets requis pour Stratis.

Procédure

1. Installer les paquets qui fournissent le service Stratis et les utilitaires de ligne de commande :

```
# dnf install stratisd stratis-cli
```

2. Vérifiez que le service **stratisd** est activé :

```
# systemctl enable --now stratisd
```

32.5. CRÉATION D'UN POOL STRATIS NON CHIFFRÉ

Vous pouvez créer un pool Stratis non chiffré à partir d'un ou de plusieurs périphériques de bloc.

Conditions préalables

- Stratis est installé. Pour plus d'informations, voir [Installation de Stratis](#).
- Le service **stratisd** est en cours d'exécution.
- Les périphériques de bloc sur lesquels vous créez un pool Stratis ne sont pas utilisés et ne sont pas montés.
- Chaque unité de bloc sur laquelle vous créez un pool Stratis a une taille d'au moins 1 Go.
- Sur l'architecture IBM Z, les périphériques de bloc **/dev/dasd*** doivent être partitionnés. Utilisez la partition dans le pool Stratis.

Pour plus d'informations sur le partitionnement des périphériques DASD, voir [Configuration d'une instance Linux sur IBM Z](#).



NOTE

Vous ne pouvez pas crypter un pool Stratis non crypté.

Procédure

1. Effacez tout système de fichiers, table de partition ou signature RAID existant sur chaque périphérique bloc que vous souhaitez utiliser dans le pool Stratis :

```
# wipefs --all block-device
```

où ***block-device*** est le chemin d'accès au dispositif de blocage ; par exemple, ***/dev/sdb***.

2. Créez le nouveau pool Stratis non chiffré sur le périphérique de bloc sélectionné :

```
# stratis pool create my-pool block-device
```

où ***block-device*** est le chemin d'accès à un bloc vide ou effacé.



NOTE

Spécifier plusieurs dispositifs de blocage sur une seule ligne :

```
# stratis pool create my-pool block-device-1 block-device-2
```

3. Vérifiez que le nouveau pool Stratis a été créé :

```
# stratis pool list
```

32.6. CRÉATION D'UN POOL STRATIS CRYPTÉ

Pour sécuriser vos données, vous pouvez créer un pool Stratis crypté à partir d'un ou de plusieurs périphériques en mode bloc.

Lorsque vous créez un pool Stratis crypté, le trousseau de clés du noyau est utilisé comme mécanisme de cryptage principal. Après les redémarrages ultérieurs du système, ce trousseau de clés du noyau est utilisé pour déverrouiller le pool Stratis crypté.

Lors de la création d'un pool Stratis crypté à partir d'un ou de plusieurs périphériques de bloc, il convient de tenir compte des points suivants :

- Chaque bloc est crypté à l'aide de la bibliothèque **cryptsetup** et met en œuvre le format **LUKS2**.
- Chaque pool Stratis peut avoir une clé unique ou partager la même clé avec d'autres pools. Ces clés sont stockées dans le trousseau de clés du noyau.
- Les blocs qui composent un pool Stratis doivent être soit tous chiffrés, soit tous non chiffrés. Il n'est pas possible d'avoir à la fois des blocs chiffrés et non chiffrés dans le même pool Stratis.
- Les périphériques de bloc ajoutés au niveau de données d'un pool Stratis crypté sont automatiquement cryptés.

Conditions préalables

- Stratis v2.1.0 ou une version ultérieure est installée. Pour plus d'informations, voir [Installation de Stratis](#).
- Le service **stratisd** est en cours d'exécution.

- Les périphériques de bloc sur lesquels vous créez un pool Stratis ne sont pas utilisés et ne sont pas montés.
- Les périphériques de bloc sur lesquels vous créez un pool Stratis ont une taille d'au moins 1 Go chacun.
- Sur l'architecture IBM Z, les périphériques de bloc **/dev/dasd*** doivent être partitionnés. Utilisez la partition dans le pool Stratis.

Pour plus d'informations sur le partitionnement des périphériques DASD, voir [Configuration d'une instance Linux sur IBM Z](#).

Procédure

1. Effacez tout système de fichiers, table de partition ou signature RAID existant sur chaque périphérique bloc que vous souhaitez utiliser dans le pool Stratis :

```
# wipefs --all block-device
```

où ***block-device*** est le chemin d'accès au dispositif de blocage ; par exemple, ***/dev/sdb***.

2. Si vous n'avez pas encore créé de jeu de clés, exécutez la commande suivante et suivez les invites pour créer un jeu de clés à utiliser pour le cryptage.

```
# stratis key set --capture-key key-description
```

où ***key-description*** est une référence à la clé créée dans le trousseau du noyau.

3. Créez le pool Stratis crypté et indiquez la description de la clé à utiliser pour le cryptage. Vous pouvez également spécifier le chemin d'accès à la clé en utilisant l'option **--keyfile-path** au lieu de l'option ***key-description*** au lieu d'utiliser l'option

```
# stratis pool create --key-desc key-description my-pool block-device
```

où

key-description

Fait référence à la clé qui existe dans le trousseau de clés du noyau, que vous avez créé à l'étape précédente.

my-pool

Spécifie le nom du nouveau pool Stratis.

block-device

Spécifie le chemin d'accès à un bloc vide ou effacé.



NOTE

Spécifier plusieurs dispositifs de blocage sur une seule ligne :

```
# stratis pool create --key-desc key-description my-pool block-device-1  
block-device-2
```

- Vérifiez que le nouveau pool Stratis a été créé :

```
# stratis pool list
```

32.7. MISE EN PLACE D'UNE COUCHE DE PROVISIONNEMENT FIN DANS LE SYSTÈME DE FICHIERS STRATIS

Une pile de stockage peut atteindre un état de surprovisionnement. Si la taille du système de fichiers devient supérieure à celle du pool qui le soutient, le pool devient plein. Pour éviter cela, désactivez l'overprovisioning, qui garantit que la taille de tous les systèmes de fichiers sur le pool ne dépasse pas le stockage physique disponible fourni par le pool. Si vous utilisez Stratis pour des applications critiques ou le système de fichiers racine, ce mode permet d'éviter certains cas de défaillance.

Si vous activez le surprovisionnement, un signal API vous avertit lorsque votre espace de stockage a été entièrement alloué. La notification sert d'avertissement à l'utilisateur pour l'informer que lorsque l'espace de stockage restant est plein, Stratis n'a plus d'espace à étendre.

Conditions préalables

- Stratis est installé. Pour plus d'informations, voir [Installation de Stratis](#).

Procédure

Pour installer correctement la piscine, deux possibilités s'offrent à vous :

- Créer un pool à partir d'un ou plusieurs blocs :

```
# stratis pool create --no-overprovision pool-name /dev/sdb
```

- En utilisant l'option **--no-overprovision**, le pool ne peut pas allouer plus d'espace logique que l'espace physique réellement disponible.

- Définir le mode de surprovisionnement dans le pool existant :

```
# stratis pool overprovision pool-name <yes|no>
```

- Si la valeur est "oui", vous activez l'overprovisioning pour le pool. Cela signifie que la somme des tailles logiques des systèmes de fichiers Stratis, pris en charge par le pool, peut dépasser la quantité d'espace de données disponible.

Vérification

- Cliquez sur le lien suivant pour obtenir la liste complète des piscines Stratis :

```
# stratis pool list
```

```
Name          Total Physical          Properties  UUID                      Alerts
pool-name     1.42 TiB / 23.96 MiB / 1.42 TiB  ~Ca,~Cr,~Op  cb7cb4d8-9322-4ac4-a6fd-
eb7ae9e1e540
```

- Vérifiez s'il y a une indication du drapeau du mode d'overprovisionnement du pool dans la sortie de **stratis pool list**. Le " ~ " est un symbole mathématique pour " NOT", donc **~Op** signifie qu'il n'y a pas d'overprovisioning.

- Facultatif : Exécutez l'opération suivante pour vérifier le surprovisionnement d'un pool spécifique :

```
# stratis pool overprovision pool-name yes

# stratis pool list

Name          Total Physical          Properties  UUID                      Alerts
pool-name    1.42 TiB / 23.96 MiB / 1.42 TiB  ~Ca,~Cr,~Op  cb7cb4d8-9322-4ac4-a6fd-
eb7ae9e1e540
```

Ressources supplémentaires

- [La page web *Stratis Storage*](#) .

32.8. LIER UN POOL STRATIS À L'EDNB

Lier un pool Stratis crypté à Network Bound Disk Encryption (NBDE) nécessite un serveur Tang. Lorsqu'un système contenant le pool Stratis redémarre, il se connecte au serveur Tang pour déverrouiller automatiquement le pool crypté sans que vous ayez à fournir la description du trousseau de clés du noyau.



NOTE

Le fait de lier un pool Stratis à un mécanisme de chiffrement Clevis supplémentaire ne supprime pas le chiffrement du trousseau de clés du noyau principal.

Conditions préalables

- Stratis v2.3.0 ou une version ultérieure est installée. Pour plus d'informations, voir [Installation de Stratis](#).
- Le service **stratisd** est en cours d'exécution.
- Vous avez créé un pool Stratis chiffré et vous disposez de la description de la clé utilisée pour le chiffrement. Pour plus d'informations, voir [Création d'un pool Stratis crypté](#) .
- Vous pouvez vous connecter au serveur Tang. Pour plus d'informations, voir [Déploiement d'un serveur Tang avec SELinux en mode d'exécution](#)

Procédure

- Lier un pool Stratis crypté à l'EDNB :

```
# stratis pool bind nbde --trust-url my-pool tang-server
```

où

my-pool

Spécifie le nom du pool Stratis crypté.

tang-server

Spécifie l'adresse IP ou l'URL du serveur Tang.

Ressources supplémentaires

- [Configuration du déverrouillage automatique des volumes chiffrés à l'aide du déchiffrement basé sur une stratégie](#)

32.9. LIER UN POOL STRATIS À UNE MPT

Lorsque vous liez un pool Stratis chiffré au Trusted Platform Module (TPM) 2.0, lorsque le système contenant le pool redémarre, le pool est automatiquement déverrouillé sans que vous ayez à fournir la description du trousseau de clés du noyau.

Conditions préalables

- Stratis v2.3.0 ou une version ultérieure est installée. Pour plus d'informations, voir [Installation de Stratis](#).
- Le service **stratisd** est en cours d'exécution.
- Vous avez créé un pool Stratis crypté. Pour plus d'informations, voir [Création d'un pool Stratis crypté](#).

Procédure

- Lier un pool Stratis crypté au TPM :

```
# stratis pool bind tpm my-pool key-description
```

où

my-pool

Spécifie le nom du pool Stratis crypté.

key-description

Fait référence à la clé qui existe dans le trousseau de clés du noyau et qui a été générée lors de la création du pool Stratis crypté.

32.10. DÉVERROUILLER UN POOL STRATIS CRYPTÉ AVEC LE TROUSSEAU DE CLÉS DU NOYAU

Après un redémarrage du système, votre pool Stratis crypté ou les périphériques de bloc qui le composent peuvent ne pas être visibles. Vous pouvez déverrouiller le pool à l'aide du trousseau de clés du noyau qui a été utilisé pour crypter le pool.

Conditions préalables

- Stratis v2.1.0 est installé. Pour plus d'informations, voir [Installation de Stratis](#).
- Le service **stratisd** est en cours d'exécution.
- Vous avez créé un pool Stratis crypté. Pour plus d'informations, voir [Création d'un pool Stratis crypté](#).

Procédure

1. Recréer le jeu de clés en utilisant la même description de clé que celle utilisée précédemment :

```
# stratis key set --capture-key key-description
```

où *key-description* fait référence à la clé qui existe dans le trousseau de clés du noyau et qui a été générée lors de la création du pool Stratis crypté.

2. Déverrouiller le pool Stratis et les blocs qui le composent :

```
# stratis pool unlock keyring
```

3. Vérifiez que le pool Stratis est visible :

```
# stratis pool list
```

32.11. DÉVERROUILLER UN POOL STRATIS CRYPTÉ AVEC CLEVIS

Après un redémarrage du système, votre pool Stratis chiffré ou les périphériques de bloc qui le composent peuvent ne pas être visibles. Vous pouvez déverrouiller un pool Stratis chiffré à l'aide du mécanisme de chiffrement supplémentaire auquel le pool est lié.

Conditions préalables

- Stratis v2.3.0 ou une version ultérieure est installée. Pour plus d'informations, voir [Installation de Stratis](#).
- Le service **stratisd** est en cours d'exécution.
- Vous avez créé un pool Stratis crypté. Pour plus d'informations, voir [Création d'un pool Stratis crypté](#).
- Le pool chiffré de Stratis est lié à un mécanisme de chiffrement supplémentaire pris en charge. Pour plus d'informations, voir [Lier un pool Stratis crypté à NBDE](#)

ou [Lier un pool Stratis crypté au TPM](#) .

Procédure

1. Déverrouiller le pool Stratis et les blocs qui le composent :

```
# stratis pool unlock clevis
```

2. Vérifiez que le pool Stratis est visible :

```
# stratis pool list
```

32.12. DÉTACHER UN POOL STRATIS DU CHIFFREMENT SUPPLÉMENTAIRE

Lorsque vous libérez un pool Stratis chiffré d'un mécanisme de chiffrement supplémentaire pris en charge, le chiffrement du trousseau de clés du noyau principal reste en place.

Conditions préalables

Conditions préalables

- Stratis v2.3.0 ou une version ultérieure est installée sur votre système. Pour plus d'informations, voir [Installation de Stratis](#).
- Vous avez créé un pool Stratis crypté. Pour plus d'informations, voir [Création d'un pool Stratis crypté](#).
- Le pool chiffré de Stratis est lié à un mécanisme de chiffrement supplémentaire pris en charge.

Procédure

- Dissocier un pool Stratis crypté d'un mécanisme de cryptage supplémentaire :

```
# stratis pool unbind clevis my-pool
```

où

my-pool spécifie le nom du pool Stratis que vous souhaitez délier.

Ressources supplémentaires

- [Lier un pool Stratis crypté à l'EDNB](#)
- [Liaison d'un pool Stratis crypté à un TPM](#)

32.13. DÉMARRAGE ET ARRÊT DU POOL STRATIS

Vous pouvez démarrer et arrêter les pools Stratis. Vous avez ainsi la possibilité de démanteler ou d'arrêter tous les objets utilisés pour construire le pool, tels que les systèmes de fichiers, les périphériques de cache, le thin pool et les périphériques cryptés. Notez que si le pool utilise activement un périphérique ou un système de fichiers, il peut émettre un avertissement et ne pas pouvoir s'arrêter.

Les pools arrêtés enregistrent leur état d'arrêt dans leurs métadonnées. Ces pools ne démarrent pas au démarrage suivant, jusqu'à ce que le pool reçoive une commande de démarrage.

S'ils ne sont pas cryptés, les pools précédemment démarrés démarrent automatiquement au démarrage. Les pools cryptés ont toujours besoin d'une commande **pool start** au démarrage, car **pool unlock** est remplacé par **pool start** dans cette version de Stratis.

Conditions préalables

- Stratis est installé. Pour plus d'informations, voir [Installation de Stratis](#).
- Le service **stratisd** est en cours d'exécution.
- Vous avez créé un pool Stratis non chiffré ou chiffré. Voir [Création d'un pool Stratis non chiffré](#)

ou [Créer un pool Stratis crypté](#) .

Procédure

- Utilisez la commande suivante pour démarrer le pool Stratis. L'option **--unlock-method** spécifie la méthode de déverrouillage du pool s'il est crypté :

```
# stratis pool start pool-uuid --unlock-method <keyring|clevis>
```

- Vous pouvez également utiliser la commande suivante pour arrêter le pool Stratis. Cela détruit la pile de stockage mais laisse toutes les métadonnées intactes :

```
# stratis pool stop pool-name
```

Verification steps

- Utilisez la commande suivante pour répertorier tous les pools du système :

```
# stratis pool list
```

- Utilisez la commande suivante pour dresser la liste de tous les pools qui n'ont pas encore été démarrés. Si l'UUID est spécifié, la commande affiche des informations détaillées sur le pool correspondant à l'UUID :

```
# stratis pool list --stopped --uuid UUID
```

32.14. CRÉATION D'UN SYSTÈME DE FICHIERS STRATIS

Créer un système de fichiers Stratis sur un pool Stratis existant.

Conditions préalables

- Stratis est installé. Pour plus d'informations, voir [Installation de Stratis](#).
- Le service **stratisd** est en cours d'exécution.
- Vous avez créé un pool Stratis. Voir [Création d'un pool Stratis non crypté](#)

ou [Créer un pool Stratis crypté](#) .

Procédure

1. Pour créer un système de fichiers Stratis sur un pool, utilisez la commande suivante

```
# stratis filesystem create --size number-and-unit my-pool my-fs
```

où

number-and-unit

Spécifie la taille d'un système de fichiers. Le format de spécification doit suivre le format de spécification de taille standard pour l'entrée, c'est-à-dire B, KiB, MiB, GiB, TiB ou PiB.

my-pool

Spécifie le nom du pool Stratis.

my-fs

Spécifie un nom arbitraire pour le système de fichiers.

Par exemple :

```
Exemple 32.1. Création d'un système de fichiers Stratis
```

```
# stratis filesystem create --size 10GiB pool1 filesystem1
```

Verification steps

- Liste les systèmes de fichiers du pool pour vérifier si le système de fichiers Stratis est créé :

```
# stratis fs list my-pool
```

Ressources supplémentaires

- [Montage d'un système de fichiers Stratis](#) .

32.15. MONTAGE D'UN SYSTÈME DE FICHIERS STRATIS

Monter un système de fichiers Stratis existant pour accéder au contenu.

Conditions préalables

- Stratis est installé. Pour plus d'informations, voir [Installation de Stratis](#).
- Le service **stratisd** est en cours d'exécution.
- Vous avez créé un système de fichiers Stratis. Pour plus d'informations, voir [Création d'un système de fichiers Stratis](#).

Procédure

- Pour monter le système de fichiers, utilisez les entrées que Stratis maintient dans le répertoire **/dev/stratis/**:

```
# mount /dev/stratis/my-pool/my-fs mount-point
```

Le système de fichiers est maintenant monté dans le répertoire *mount-point* et prêt à être utilisé.

Ressources supplémentaires

- [Création d'un système de fichiers Stratis](#) .

32.16. MONTAGE PERSISTANT D'UN SYSTÈME DE FICHIERS STRATIS

Cette procédure permet de monter de manière persistante un système de fichiers Stratis afin qu'il soit disponible automatiquement après le démarrage du système.

Conditions préalables

- Stratis est installé. Voir [Installation de Stratis](#).
- Le service **stratisd** est en cours d'exécution.
- Vous avez créé un système de fichiers Stratis. Voir [Création d'un système de fichiers Stratis](#) .

Procédure

1. Déterminez l'attribut UUID du système de fichiers :

```
$ lsblk --output=UUID /dev/stratis/my-pool/my-fs
```

Par exemple :

Exemple 32.2. Affichage de l'UUID du système de fichiers Stratis

```
$ lsblk --output=UUID /dev/stratis/my-pool/fs1
UUID
a1f0b64a-4ebb-4d4e-9543-b1d79f600283
```

2. Si le répertoire du point de montage n'existe pas, créez-le :

```
# mkdir --parents mount-point
```

3. En tant que root, éditez le fichier **/etc/fstab** et ajoutez une ligne pour le système de fichiers, identifié par l'UUID. Utilisez **xfs** comme type de système de fichiers et ajoutez l'option **x-systemd.requires=stratisd.service**.

Par exemple :

Exemple 32.3. Le point de montage /fs1 dans /etc/fstab

```
UUID=a1f0b64a-4ebb-4d4e-9543-b1d79f600283 /fs1 xfs defaults,x-
systemd.requires=stratisd.service 0 0
```

4. Régénérez les unités de montage pour que votre système enregistre la nouvelle configuration :

```
# systemctl daemon-reload
```

5. Essayez de monter le système de fichiers pour vérifier que la configuration fonctionne :

```
# mount mount-point
```

Ressources supplémentaires

- [Montage persistant des systèmes de fichiers](#) .

32.17. CONFIGURATION DE SYSTÈMES DE FICHIERS STRATIS NON ROOT DANS /ETC/FSTAB À L'AIDE D'UN SERVICE SYSTEMD

Vous pouvez gérer la configuration des systèmes de fichiers non racine dans le fichier `/etc/fstab` à l'aide d'un service `systemd`.

Conditions préalables

- Stratis est installé. Voir [Installation de Stratis](#).
- Le service **stratisd** est en cours d'exécution.
- Vous avez créé un système de fichiers Stratis. Voir [Création d'un système de fichiers Stratis](#) .

Procédure

- Pour tous les systèmes de fichiers Stratis non racine, utilisez :

```
# /dev/stratis/[STRATIS_SYMLINK] [MOUNT_POINT] xfs defaults, x-  
systemd.requires=stratis-fstab-setup@[POOL_UUID].service,x-systemd.after=stratis-stab-  
setup@[POOL_UUID].service <dump_value> <fsck_value>
```

Ressources supplémentaires

- [Montage persistant des systèmes de fichiers](#) .

CHAPITRE 33. EXTENSION D'UN VOLUME STRATIS AVEC DES PÉRIPHÉRIQUES DE BLOC SUPPLÉMENTAIRES

Vous pouvez attacher des périphériques de bloc supplémentaires à un pool Stratis afin de fournir une plus grande capacité de stockage pour les systèmes de fichiers Stratis.



IMPORTANT

Stratis est une fonctionnalité d'aperçu technologique uniquement. Les fonctionnalités de l'aperçu technologique ne sont pas prises en charge par les accords de niveau de service (SLA) de production de Red Hat et peuvent ne pas être complètes sur le plan fonctionnel. Red Hat ne recommande pas de les utiliser en production. Ces fonctionnalités offrent un accès anticipé aux fonctionnalités des produits à venir, ce qui permet aux clients de tester les fonctionnalités et de fournir un retour d'information pendant le processus de développement. Pour plus d'informations sur l'étendue de l'assistance des fonctionnalités Red Hat Technology Preview, consultez <https://access.redhat.com/support/offerings/techpreview>.

33.1. COMPOSANTS D'UN VOLUME STRATIS

Découvrez les composants d'un volume Stratis.

En externe, Stratis présente les composants de volume suivants dans l'interface de ligne de commande et l'API :

blockdev

Périphériques en bloc, tels qu'un disque ou une partition de disque.

pool

Composé d'un ou plusieurs dispositifs de blocage.

Un pool a une taille totale fixe, égale à la taille des blocs.

Le pool contient la plupart des couches Stratis, telles que le cache de données non volatiles utilisant la cible **dm-cache**.

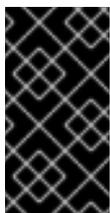
Stratis crée un répertoire **/dev/stratis/my-pool/** pour chaque pool. Ce répertoire contient des liens vers les périphériques qui représentent les systèmes de fichiers Stratis dans le pool.

filesystem

Chaque pool peut contenir un ou plusieurs systèmes de fichiers, qui stockent des fichiers.

Les systèmes de fichiers sont finement provisionnés et n'ont pas une taille totale fixe. La taille réelle d'un système de fichiers augmente avec les données qui y sont stockées. Si la taille des données approche la taille virtuelle du système de fichiers, Stratis augmente automatiquement le volume fin et le système de fichiers.

Les systèmes de fichiers sont formatés avec XFS.



IMPORTANT

Stratis suit des informations sur les systèmes de fichiers créés à l'aide de Stratis que XFS ne connaît pas, et les modifications apportées à l'aide de XFS ne créent pas automatiquement de mises à jour dans Stratis. Les utilisateurs ne doivent pas reformater ou reconfigurer les systèmes de fichiers XFS qui sont gérés par Stratis.

Stratis crée des liens vers les systèmes de fichiers au niveau du `/dev/stratis/my-pool/my-fs` chemin.



NOTE

Stratis utilise de nombreux dispositifs Device Mapper, qui apparaissent dans les listes `dmsetup` et le fichier `/proc/partitions`. De même, la sortie de la commande `lsblk` reflète le fonctionnement interne et les couches de Stratis.

33.2. AJOUT DE BLOCS À UN POOL STRATIS

Cette procédure ajoute un ou plusieurs périphériques de bloc à un pool Stratis pour qu'ils soient utilisables par les systèmes de fichiers Stratis.

Conditions préalables

- Stratis est installé. Voir [Installation de Stratis](#).
- Le service `stratisd` est en cours d'exécution.
- Les périphériques de bloc que vous ajoutez au pool Stratis ne sont pas utilisés et ne sont pas montés.
- Les périphériques de bloc que vous ajoutez au pool Stratis ont une taille d'au moins 1 gigaoctet chacun.

Procédure

- Pour ajouter un ou plusieurs périphériques de bloc au pool, utilisez la procédure suivante :

```
# stratis pool add-data my-pool device-1 device-2 device-n
```

Ressources supplémentaires

- `stratis(8)` page de manuel

33.3. RESSOURCES SUPPLÉMENTAIRES

- [Le site web Stratis Storage](#)

CHAPITRE 34. SURVEILLANCE DES SYSTÈMES DE FICHIERS STRATIS

En tant qu'utilisateur Stratis, vous pouvez afficher des informations sur les volumes Stratis de votre système afin de surveiller leur état et l'espace libre.



IMPORTANT

Stratis est une fonctionnalité d'aperçu technologique uniquement. Les fonctionnalités de l'aperçu technologique ne sont pas prises en charge par les accords de niveau de service (SLA) de production de Red Hat et peuvent ne pas être complètes sur le plan fonctionnel. Red Hat ne recommande pas de les utiliser en production. Ces fonctionnalités offrent un accès anticipé aux fonctionnalités des produits à venir, ce qui permet aux clients de tester les fonctionnalités et de fournir un retour d'information pendant le processus de développement. Pour plus d'informations sur l'étendue de l'assistance des fonctionnalités Red Hat Technology Preview, consultez <https://access.redhat.com/support/offerings/techpreview>.

34.1. TAILLES DES STRATIS RAPPORTÉES PAR LES DIFFÉRENTS SERVICES PUBLICS

Cette section explique la différence entre les tailles de Stratis indiquées par les utilitaires standard tels que **df** et l'utilitaire **stratis**.

Les utilitaires Linux standard tels que **df** indiquent la taille de la couche du système de fichiers XFS sur Stratis, qui est de 1 TiB. Cette information n'est pas utile, car l'utilisation réelle du stockage de Stratis est moindre en raison de l'approvisionnement fin, et aussi parce que Stratis agrandit automatiquement le système de fichiers lorsque la couche XFS est presque pleine.



IMPORTANT

Surveillez régulièrement la quantité de données écrites sur vos systèmes de fichiers Stratis, qui est indiquée par la valeur *Total Physical Used*. Assurez-vous qu'elle ne dépasse pas la valeur *Total Physical Size*.

Ressources supplémentaires

- **stratis(8)** page de manuel.

34.2. AFFICHAGE D'INFORMATIONS SUR LES VOLUMES STRATIS

Cette procédure répertorie les statistiques relatives à vos volumes Stratis, telles que la taille totale, utilisée et libre ou les systèmes de fichiers et les périphériques de bloc appartenant à un pool.

Conditions préalables

- Stratis est installé. Voir [Installation de Stratis](#).
- Le service **stratisd** est en cours d'exécution.

Procédure

- Pour afficher des informations sur tous les sites **block devices** utilisés pour Stratis sur votre système :

```
# stratis blockdev

Pool Name Device Node Physical Size State Tier
my-pool /dev/sdb 9.10 TiB In-use Data
```

- Pour afficher des informations sur tous les sites Stratis **pools** de votre système :

```
# stratis pool

Name Total Physical Size Total Physical Used
my-pool 9.10 TiB 598 MiB
```

- Pour afficher des informations sur tous les sites Stratis **file systems** de votre système :

```
# stratis filesystem

Pool Name Name Used Created Device
my-pool my-fs 546 MiB Nov 08 2018 08:03 /dev/stratis/my-pool/my-fs
```

Ressources supplémentaires

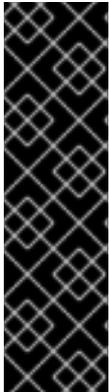
- **stratis(8)** page de manuel.

34.3. RESSOURCES SUPPLÉMENTAIRES

- [Le site web Stratis Storage](#)

CHAPITRE 35. UTILISATION D'INSTANTANÉS SUR LES SYSTÈMES DE FICHIERS STRATIS

Vous pouvez utiliser des instantanés sur les systèmes de fichiers Stratis pour capturer l'état du système de fichiers à des moments arbitraires et le restaurer ultérieurement.



IMPORTANT

Stratis est une fonctionnalité d'aperçu technologique uniquement. Les fonctionnalités de l'aperçu technologique ne sont pas prises en charge par les accords de niveau de service (SLA) de production de Red Hat et peuvent ne pas être complètes sur le plan fonctionnel. Red Hat ne recommande pas de les utiliser en production. Ces fonctionnalités offrent un accès anticipé aux fonctionnalités des produits à venir, ce qui permet aux clients de tester les fonctionnalités et de fournir un retour d'information pendant le processus de développement. Pour plus d'informations sur l'étendue de l'assistance des fonctionnalités Red Hat Technology Preview, consultez <https://access.redhat.com/support/offerings/techpreview>.

35.1. CARACTÉRISTIQUES DES INSTANTANÉS STRATIS

Dans Stratis, un instantané est un système de fichiers Stratis normal créé en tant que copie d'un autre système de fichiers Stratis. L'instantané contient initialement le même contenu de fichier que le système de fichiers d'origine, mais il peut changer au fur et à mesure que l'instantané est modifié. Les modifications apportées à l'instantané ne seront pas répercutées dans le système de fichiers d'origine.

La mise en œuvre actuelle de l'instantané dans Stratis se caractérise par les éléments suivants :

- Un instantané d'un système de fichiers est un autre système de fichiers.
- La durée de vie d'un instantané et de son origine n'est pas liée. Un système de fichiers instantané peut vivre plus longtemps que le système de fichiers à partir duquel il a été créé.
- Il n'est pas nécessaire qu'un système de fichiers soit monté pour créer un instantané à partir de celui-ci.
- Chaque instantané utilise environ un demi gigaoctet de mémoire de sauvegarde, qui est nécessaire pour le journal XFS.

35.2. CRÉATION D'UN INSTANTANÉ STRATIS

Cette procédure crée un système de fichiers Stratis sous la forme d'un instantané d'un système de fichiers Stratis existant.

Conditions préalables

- Stratis est installé. Voir [Installation de Stratis](#).
- Le service **stratisd** est en cours d'exécution.
- Vous avez créé un système de fichiers Stratis. Voir [Création d'un système de fichiers Stratis](#).

Procédure

- Pour créer un instantané Stratis, utilisez :

```
# stratis fs snapshot my-pool my-fs my-fs-snapshot
```

Ressources supplémentaires

- **stratis(8)** page de manuel.

35.3. ACCÉDER AU CONTENU D'UN INSTANTANÉ STRATIS

Cette procédure permet de monter un instantané d'un système de fichiers Stratis afin de le rendre accessible pour les opérations de lecture et d'écriture.

Conditions préalables

- Stratis est installé. Voir [Installation de Stratis](#).
- Le service **stratisd** est en cours d'exécution.
- Vous avez créé un instantané Stratis. Voir [Création d'un système de fichiers Stratis](#).

Procédure

- Pour accéder à l'instantané, montez-le comme un système de fichiers normal à partir du répertoire **/dev/stratis/my-pool/** dans le répertoire

```
# mount /dev/stratis/my-pool/my-fs-snapshot mount-point
```

Ressources supplémentaires

- [Montage d'un système de fichiers Stratis](#).
- **mount(8)** page de manuel.

35.4. REVENIR À UN INSTANTANÉ PRÉCÉDENT D'UN SYSTÈME DE FICHIERS STRATIS

Cette procédure rétablit le contenu d'un système de fichiers Stratis à l'état capturé dans un instantané Stratis.

Conditions préalables

- Stratis est installé. Voir [Installation de Stratis](#).
- Le service **stratisd** est en cours d'exécution.
- Vous avez créé un instantané Stratis. Voir [Création d'un instantané Stratis](#).

Procédure

1. Il est possible de sauvegarder l'état actuel du système de fichiers afin de pouvoir y accéder ultérieurement :

```
# stratis filesystem snapshot my-pool my-fs my-fs-backup
```

- Démonter et supprimer le système de fichiers d'origine :

```
# umount /dev/stratis/my-pool/my-fs
# stratis filesystem destroy my-pool my-fs
```

- Créer une copie de l'instantané sous le nom du système de fichiers d'origine :

```
# stratis filesystem snapshot my-pool my-fs-snapshot my-fs
```

- Montez l'instantané, qui est désormais accessible sous le même nom que le système de fichiers d'origine :

```
# mount /dev/stratis/my-pool/my-fs mount-point
```

Le contenu du système de fichiers nommé *my-fs* est maintenant identique à l'instantané *my-fs-snapshot*.

Ressources supplémentaires

- **stratis(8)** page de manuel.

35.5. SUPPRESSION D'UN INSTANTANÉ STRATIS

Cette procédure permet de supprimer un instantané Stratis d'un pool. Les données de l'instantané sont perdues.

Conditions préalables

- Stratis est installé. Voir [Installation de Stratis](#).
- Le service **stratisd** est en cours d'exécution.
- Vous avez créé un instantané Stratis. Voir [Création d'un instantané Stratis](#).

Procédure

- Démonter l'instantané :

```
# umount /dev/stratis/my-pool/my-fs-snapshot
```

- Détruire l'instantané :

```
# stratis filesystem destroy my-pool my-fs-snapshot
```

Ressources supplémentaires

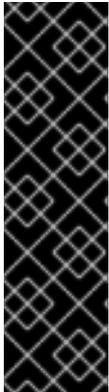
- **stratis(8)** page de manuel.

35.6. RESSOURCES SUPPLÉMENTAIRES

- [Le site web Stratis Storage](#)

CHAPITRE 36. SUPPRESSION DES SYSTÈMES DE FICHIERS STRATIS

Vous pouvez supprimer un système de fichiers Stratis existant ou un pool Stratis en détruisant les données qu'ils contiennent.



IMPORTANT

Stratis est une fonctionnalité d'aperçu technologique uniquement. Les fonctionnalités de l'aperçu technologique ne sont pas prises en charge par les accords de niveau de service (SLA) de production de Red Hat et peuvent ne pas être complètes sur le plan fonctionnel. Red Hat ne recommande pas de les utiliser en production. Ces fonctionnalités offrent un accès anticipé aux fonctionnalités des produits à venir, ce qui permet aux clients de tester les fonctionnalités et de fournir un retour d'information pendant le processus de développement. Pour plus d'informations sur l'étendue de l'assistance des fonctionnalités Red Hat Technology Preview, consultez <https://access.redhat.com/support/offerings/techpreview>.

36.1. COMPOSANTS D'UN VOLUME STRATIS

Découvrez les composants d'un volume Stratis.

En externe, Stratis présente les composants de volume suivants dans l'interface de ligne de commande et l'API :

blockdev

Périphériques en bloc, tels qu'un disque ou une partition de disque.

pool

Composé d'un ou plusieurs dispositifs de blocage.

Un pool a une taille totale fixe, égale à la taille des blocs.

Le pool contient la plupart des couches Stratis, telles que le cache de données non volatiles utilisant la cible **dm-cache**.

Stratis crée un répertoire **/dev/stratis/my-pool/** pour chaque pool. Ce répertoire contient des liens vers les périphériques qui représentent les systèmes de fichiers Stratis dans le pool.

filesystem

Chaque pool peut contenir un ou plusieurs systèmes de fichiers, qui stockent des fichiers.

Les systèmes de fichiers sont finement provisionnés et n'ont pas une taille totale fixe. La taille réelle d'un système de fichiers augmente avec les données qui y sont stockées. Si la taille des données approche la taille virtuelle du système de fichiers, Stratis augmente automatiquement le volume fin et le système de fichiers.

Les systèmes de fichiers sont formatés avec XFS.



IMPORTANT

Stratis suit des informations sur les systèmes de fichiers créés à l'aide de Stratis que XFS ne connaît pas, et les modifications apportées à l'aide de XFS ne créent pas automatiquement de mises à jour dans Stratis. Les utilisateurs ne doivent pas reformater ou reconfigurer les systèmes de fichiers XFS qui sont gérés par Stratis.

Stratis crée des liens vers les systèmes de fichiers au niveau du `/dev/stratis/my-pool/my-fs` chemin.



NOTE

Stratis utilise de nombreux dispositifs Device Mapper, qui apparaissent dans les listes `dmsetup` et le fichier `/proc/partitions`. De même, la sortie de la commande `lsblk` reflète le fonctionnement interne et les couches de Stratis.

36.2. SUPPRESSION D'UN SYSTÈME DE FICHIERS STRATIS

Cette procédure permet de supprimer un système de fichiers Stratis existant. Les données qui y sont stockées sont perdues.

Conditions préalables

- Stratis est installé. Voir [Installation de Stratis](#).
- Le service `stratisd` est en cours d'exécution.
- Vous avez créé un système de fichiers Stratis. Voir [Création d'un système de fichiers Stratis](#).

Procédure

1. Démonter le système de fichiers :

```
# umount /dev/stratis/my-pool/my-fs
```

2. Détruire le système de fichiers :

```
# stratis filesystem destroy my-pool my-fs
```

3. Vérifiez que le système de fichiers n'existe plus :

```
# stratis filesystem list my-pool
```

Ressources supplémentaires

- `stratis(8)` page de manuel.

36.3. SUPPRESSION D'UN POOL STRATIS

Cette procédure permet de supprimer un pool Stratis existant. Les données qui y sont stockées sont perdues.

Conditions préalables

- Stratis est installé. Voir [Installation de Stratis](#).
- Le service `stratisd` est en cours d'exécution.
- Vous avez créé un pool Stratis :

- Pour créer un pool non chiffré, voir [Création d'un pool Stratis non chiffré](#)
- Pour créer un pool crypté, voir [Création d'un pool crypté Stratis](#) .

Procédure

1. Liste des systèmes de fichiers du pool :

```
# stratis filesystem list my-pool
```

2. Démonter tous les systèmes de fichiers du pool :

```
# umount /dev/stratis/my-pool/my-fs-1 \  
/dev/stratis/my-pool/my-fs-2 \  
/dev/stratis/my-pool/my-fs-n
```

3. Détruire les systèmes de fichiers :

```
# stratis filesystem destroy my-pool my-fs-1 my-fs-2
```

4. Détruire la piscine :

```
# stratis pool destroy my-pool
```

5. Vérifiez que le pool n'existe plus :

```
# stratis pool list
```

Ressources supplémentaires

- **stratis(8)** page de manuel.

36.4. RESSOURCES SUPPLÉMENTAIRES

- [Le site web Stratis Storage](#)

CHAPITRE 37. PREMIERS PAS AVEC UN SYSTÈME DE FICHIERS EXT4

En tant qu'administrateur système, vous pouvez créer, monter, redimensionner, sauvegarder et restaurer un système de fichiers ext4. Le système de fichiers ext4 est une extension évolutive du système de fichiers ext3. Avec Red Hat Enterprise Linux 9, il peut prendre en charge une taille maximale de fichier individuel de **16** téraoctets, et un système de fichiers d'une taille maximale de **50** téraoctets.

37.1. CARACTÉRISTIQUES D'UN SYSTÈME DE FICHIERS EXT4

Voici les caractéristiques d'un système de fichiers ext4 :

- Utilisation d'extents : Le système de fichiers ext4 utilise les extents, ce qui améliore les performances lors de l'utilisation de fichiers volumineux et réduit la surcharge de métadonnées pour les fichiers volumineux.
- Ext4 étiquette les groupes de blocs non alloués et les sections de table d'inodes en conséquence, ce qui permet aux groupes de blocs et aux sections de table d'être ignorés lors d'une vérification du système de fichiers. Cela permet une vérification rapide du système de fichiers, qui devient de plus en plus bénéfique au fur et à mesure que la taille du système de fichiers augmente.
- Somme de contrôle des métadonnées : Par défaut, cette fonctionnalité est activée dans Red Hat Enterprise Linux 9.
- Caractéristiques d'allocation d'un système de fichiers ext4 :
 - Pré-allocation permanente
 - Attribution retardée
 - Allocation multi-blocs
 - Allocation en fonction de la taille de la bande
- Attributs étendus (**xattr**) : Cela permet au système d'associer plusieurs paires de noms et de valeurs supplémentaires par fichier.
- Journalisation des quotas : Cela évite d'avoir à effectuer de longs contrôles de cohérence des quotas après une panne.



NOTE

Le seul mode de journalisation pris en charge dans ext4 est **data=ordered** (par défaut). Pour plus d'informations, voir [L'option de journalisation EXT "data=writeback" est-elle prise en charge dans RHEL ?](#) Article de la base de connaissances.

- Horodatage à la sous-seconde - Cette fonction permet d'obtenir un horodatage à la sous-seconde.

Ressources supplémentaires

- **ext4** page de manuel.

37.2. CRÉATION D'UN SYSTÈME DE FICHIERS EXT4

En tant qu'administrateur système, vous pouvez créer un système de fichiers ext4 sur un périphérique bloc à l'aide de la commande **mkfs.ext4**.

Conditions préalables

- Une partition sur votre disque. Pour plus d'informations sur la création de partitions MBR ou GPT, voir [Création d'une table de partitions sur un disque avec parted](#) .
- Vous pouvez également utiliser un volume LVM ou MD.

Procédure

1. Pour créer un système de fichiers ext4 :

- Pour un périphérique à partition normale, un volume LVM, un volume MD ou un périphérique similaire, utilisez la commande suivante :

```
# mkfs.ext4 /dev/block_device
```

Remplacez `/dev/block_device` par le chemin d'accès à un périphérique bloc.

Par exemple, `/dev/sdb1`, `/dev/disk/by-uuid/05e99ec8-def1-4a5e-8a9d-5945339ceb2a`, ou `/dev/my-volgroup/my-lv`. En général, les options par défaut sont optimales pour la plupart des scénarios d'utilisation.

- Pour les périphériques à bandes de blocs (par exemple, les matrices RAID5), la géométrie des bandes peut être spécifiée au moment de la création du système de fichiers. L'utilisation d'une géométrie de bandes appropriée améliore les performances d'un système de fichiers ext4. Par exemple, pour créer un système de fichiers avec une bande de 64k (c'est-à-dire 16 x 4096) sur un système de fichiers de 4k-blocs, utilisez la commande suivante :

```
# mkfs.ext4 -E stride=16,stripe-width=64 /dev/block_device
```

Dans l'exemple donné :

- `stride=valeur` : Spécifie la taille du bloc RAID
- `stripe-width=valeur` : Spécifie le nombre de disques de données dans un périphérique RAID ou le nombre d'unités de bande dans la bande.



NOTE

- Pour spécifier un UUID lors de la création d'un système de fichiers :

```
# mkfs.ext4 -U UUID /dev/block_device
```

Remplacez *UUID* par l'UUID que vous souhaitez définir : par exemple, **7cd65de3-e0be-41d9-b66d-96d749c02da7**.

Remplacez */dev/block_device* par le chemin d'accès à un système de fichiers ext4 auquel l'UUID doit être ajouté : par exemple, **/dev/sda8**.

- Pour spécifier une étiquette lors de la création d'un système de fichiers :

```
# mkfs.ext4 -L label-name /dev/block_device
```

2. Pour visualiser le système de fichiers ext4 créé :

```
# blkid
```

Ressources supplémentaires

- **ext4** page de manuel.
- **mkfs.ext4** page de manuel.

37.3. MONTAGE D'UN SYSTÈME DE FICHIERS EXT4

En tant qu'administrateur système, vous pouvez monter un système de fichiers ext4 à l'aide de l'utilitaire **mount**.

Conditions préalables

- Un système de fichiers ext4. Pour plus d'informations sur la création d'un système de fichiers ext4, voir [Création d'un système de fichiers ext4](#) .

Procédure

1. Pour créer un point de montage afin de monter le système de fichiers :

```
# mkdir /mount/point
```

Remplacez */mount/point* par le nom du répertoire où le point de montage de la partition doit être créé.

2. Pour monter un système de fichiers ext4 :

- Pour monter un système de fichiers ext4 sans options supplémentaires :

```
# mount /dev/block_device /mount/point
```

- Pour monter le système de fichiers de manière persistante, voir [Montage persistant des systèmes de fichiers](#).

3. Pour visualiser le système de fichiers monté :

```
# df -h
```

Ressources supplémentaires

- **mount** page de manuel.
- **ext4** page de manuel.
- **fstab** page de manuel.
- [Montage des systèmes de fichiers](#).

37.4. REDIMENSIONNEMENT D'UN SYSTÈME DE FICHIERS EXT4

En tant qu'administrateur système, vous pouvez redimensionner un système de fichiers ext4 à l'aide de l'utilitaire **resize2fs**. L'utilitaire **resize2fs** lit la taille en unités de taille de bloc du système de fichiers, sauf si un suffixe indiquant une unité spécifique est utilisé. Les suffixes suivants indiquent des unités spécifiques :

- s (secteurs) - **512** secteurs d'octets
- K (kilo-octets) - **1,024** octets
- M (mégaoctets) - **1,048,576** octets
- G (gigaoctets) - **1,073,741,824** octets
- T (téraoctets) - **1,099,511,627,776** octets

Conditions préalables

- Un système de fichiers ext4. Pour plus d'informations sur la création d'un système de fichiers ext4, voir [Création d'un système de fichiers ext4](#).
- Un périphérique de bloc sous-jacent d'une taille appropriée pour contenir le système de fichiers après le redimensionnement.

Procédure

1. Pour redimensionner un système de fichiers ext4, procédez comme suit :

- Pour réduire et augmenter la taille d'un système de fichiers ext4 non monté :

```
# umount /dev/block_device  
# e2fsck -f /dev/block_device  
# resize2fs /dev/block_device size
```

Remplacez `/dev/block_device` par le chemin d'accès au dispositif de blocage, par exemple `/dev/sdb1`.

Remplacez `size` par la valeur de redimensionnement requise en utilisant les suffixes **s**, **K**, **M**, **G**, et **T**.

- Un système de fichiers ext4 peut être agrandi lorsqu'il est monté à l'aide de la commande **resize2fs**:

```
# resize2fs /mount/device size
```



NOTE

Le paramètre de taille est facultatif (et souvent redondant) lors de l'extension. Le site **resize2fs** s'étend automatiquement pour remplir l'espace disponible du conteneur, généralement un volume logique ou une partition.

2. Pour visualiser le système de fichiers redimensionné :

```
# df -h
```

Ressources supplémentaires

- **resize2fs** page de manuel.
- **e2fsck** page de manuel.
- **ext4** page de manuel.

37.5. COMPARAISON DES OUTILS UTILISÉS AVEC EXT4 ET XFS

Cette section compare les outils à utiliser pour accomplir des tâches courantes sur les systèmes de fichiers ext4 et XFS.

Tâche	ext4	XFS
Créer un système de fichiers	mkfs.ext4	mkfs.xfs
Vérification du système de fichiers	e2fsck	xfs_repair
Redimensionner un système de fichiers	resize2fs	xfs_growfs
Enregistrer une image d'un système de fichiers	e2image	xfs_metadump et xfs_mdrestore
Étiqueter ou régler un système de fichiers	tune2fs	xfs_admin
Sauvegarder un système de fichiers	tar et rsync	xfsdump et xfsrestore
Gestion des quotas	quota	xfs_quota
Cartographie des fichiers	filefrag	xfs_bmap



NOTE

Si vous souhaitez une solution client-serveur complète pour les sauvegardes sur le réseau, vous pouvez utiliser l'utilitaire de sauvegarde **bacula** disponible dans RHEL 9. Pour plus d'informations sur Bacula, voir la [solution de sauvegarde Bacula](#).