Red Hat Enterprise Linux 9

Gestion des périphériques de stockage

Déploiement et configuration du stockage à nœud unique dans Red Hat Enterprise Linux 9

Last Updated: 2024-02-15

Red Hat Enterprise Linux 9 Gestion des périphériques de stockage

Déploiement et configuration du stockage à nœud unique dans Red Hat Enterprise Linux 9

Notice légale

Résumé

Cette documentation fournit des instructions sur la manière de gérer efficacement les périphériques de stockage dans Red Hat Enterprise Linux 9.

Table des matières

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**.

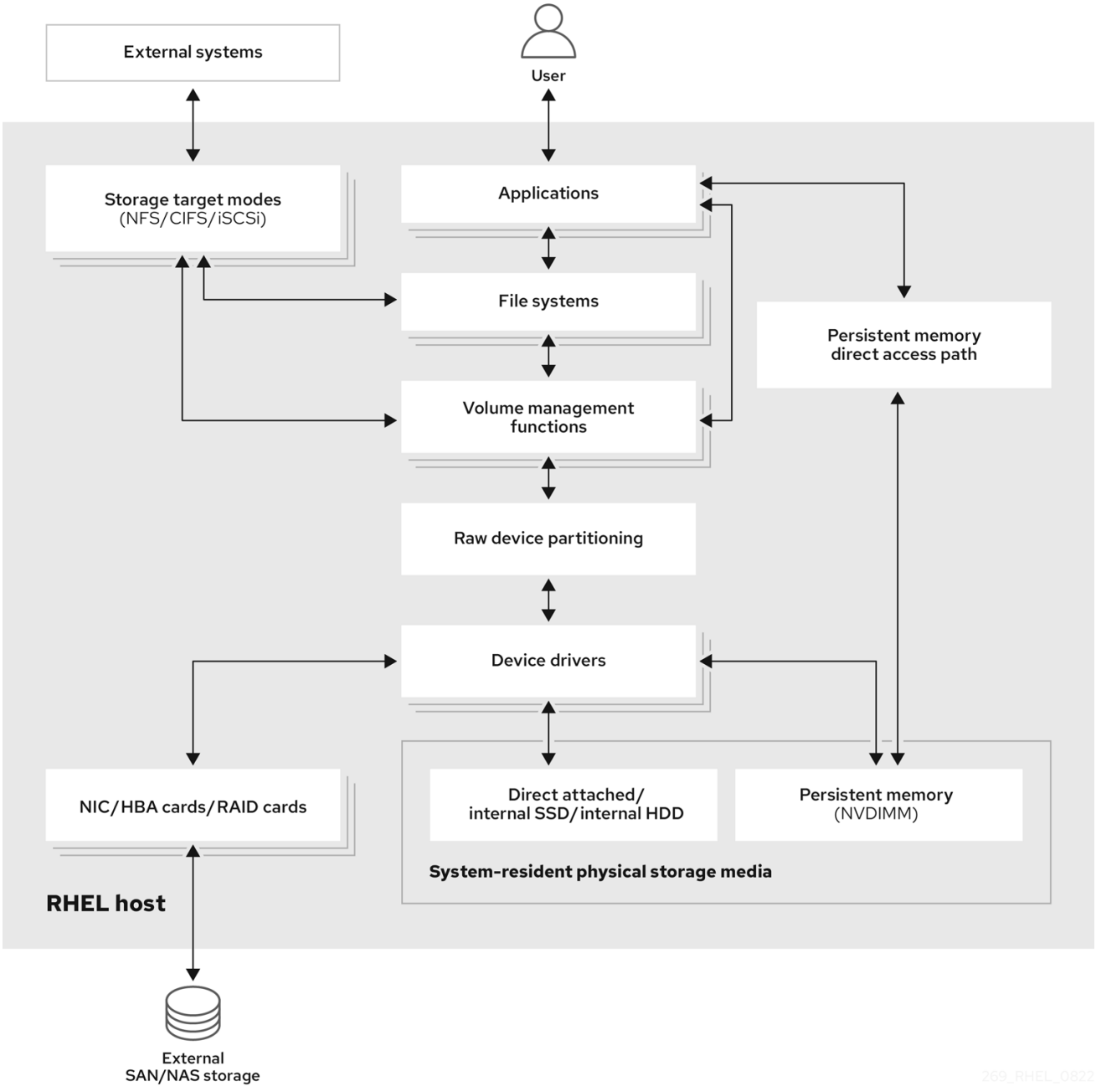
CHAPITRE 1. APERÇU DES OPTIONS DE STOCKAGE DISPONIBLES

Il existe plusieurs options de stockage local, distant et en grappe disponibles sur Red Hat Enterprise Linux 8.

Le stockage local implique que les périphériques de stockage sont soit installés sur le système, soit directement attachés au système.

Dans le cas du stockage à distance, l'accès aux périphériques se fait par le biais d'un réseau local, d'Internet ou d'un réseau Fibre Channel. Le diagramme de stockage de haut niveau suivant de Red Hat Enterprise Linux décrit les différentes options de stockage.

Figure 1.1. Diagramme de stockage de haut niveau de Red Hat Enterprise Linux



1.1. VUE D'ENSEMBLE DU STOCKAGE LOCAL

Red Hat Enterprise Linux 9 offre plusieurs options de stockage local.

Administration de base des disques

En utilisant **parted** et **fdisk**, vous pouvez créer, modifier, supprimer et visualiser des partitions de disque. Les normes d'agencement des partitions sont les suivantes :

Master Boot Record (MBR)

Il est utilisé avec les ordinateurs basés sur le BIOS. Vous pouvez créer des partitions primaires, étendues et logiques.

Table de partition GUID (GPT)

Il utilise l'identifiant unique global (GUID) et fournit un GUID unique pour le disque et la partition.

Pour chiffrer la partition, vous pouvez utiliser Linux Unified Key Setup-on-disk-format (LUKS). Pour chiffrer la partition, sélectionnez l'option pendant l'installation et l'invite s'affiche pour saisir la phrase de passe. Cette phrase de passe déverrouille la clé de chiffrement.

Options de consommation de stockage

Gestion des modules de mémoire double en ligne non volatile (NVDIMM)

Il s'agit d'une combinaison de mémoire et de stockage. Vous pouvez activer et gérer différents types de stockage sur les périphériques NVDIMM connectés à votre système.

Gestion du stockage par blocs

Les données sont stockées sous forme de blocs, chaque bloc ayant un identifiant unique.

Stockage de fichiers

Les données sont stockées au niveau des fichiers sur le système local. Il est possible d'accéder à ces données localement en utilisant XFS (par défaut) ou ext4, et sur un réseau en utilisant NFS et SMB.

Volumes logiques

Gestionnaire de volume logique (LVM)

Il crée des périphériques logiques à partir de périphériques physiques. Le volume logique (LV) est une combinaison des volumes physiques (PV) et des groupes de volumes (VG). La configuration de LVM comprend

- Création de PV à partir des disques durs.
- Création de VG à partir de PV.
- Création d'un LV à partir du VG et attribution de points de montage au LV.

Optimiseur de données virtuelles (VDO)

Il est utilisé pour la réduction des données au moyen de la déduplication, de la compression et de l'approvisionnement fin. L'utilisation de LV ci-dessous aide à :

- Extension du volume de la VDO
- Répartition du volume VDO sur plusieurs appareils

Systèmes de fichiers locaux

XFS

Le système de fichiers par défaut de RHEL.

Ext4

Un ancien système de fichiers.

Stratis

Il est disponible en tant qu'aperçu technologique. 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.

1.2. APERÇU DU STOCKAGE À DISTANCE

Vous trouverez ci-dessous les options de stockage à distance disponibles dans Red Hat Enterprise Linux 8 :

Options de connectivité de stockage

iSCSI

RHEL 9 utilise l'outil targetcli pour ajouter, supprimer, afficher et surveiller les interconnexions de stockage iSCSI.

Fibre Channel (FC)

RHEL 9 fournit les pilotes Fibre Channel natifs suivants :

- **lpfc**
- **qla2xxx**
- **Zfcp**

Mémoire non volatile Express (NVMe)

Une interface qui permet à l'utilitaire du logiciel hôte de communiquer avec les disques d'état solide. Utilisez les types de transport de tissu suivants pour configurer NVMe sur les tissus :

- NVMe sur des tissus utilisant l'accès direct à la mémoire à distance (RDMA).
- NVMe sur les réseaux utilisant Fibre Channel (FC)

Device Mapper multipathing (DM Multipath)

Permet de configurer plusieurs chemins d'E/S entre les nœuds de serveur et les baies de stockage dans un seul appareil. Ces chemins d'E/S sont des connexions SAN physiques qui peuvent inclure des câbles, des commutateurs et des contrôleurs distincts.

Système de fichiers en réseau

- NFS
- PME

1.3. VUE D'ENSEMBLE DU SYSTÈME DE FICHIERS GFS2

Le système de fichiers Red Hat Global File System 2 (GFS2) est un système de fichiers en grappe symétrique de 64 bits qui fournit un espace de noms partagé et gère la cohérence entre plusieurs nœuds partageant un périphérique de bloc commun. Un système de fichiers GFS2 est destiné à fournir un ensemble de fonctionnalités aussi proche que possible d'un système de fichiers local, tout en

appliquant une cohérence de cluster complète entre les nœuds. Pour ce faire, les nœuds utilisent un système de verrouillage à l'échelle de la grappe pour les ressources du système de fichiers. Ce schéma de verrouillage utilise des protocoles de communication tels que TCP/IP pour échanger des informations de verrouillage.

Dans certains cas, l'API du système de fichiers Linux ne permet pas de rendre totalement transparente la nature groupée de GFS2 ; par exemple, les programmes utilisant des verrous POSIX dans GFS2 doivent éviter d'utiliser la fonction **GETLK** car, dans un environnement groupé, l'ID du processus peut être celui d'un nœud différent dans le groupe. Dans la plupart des cas, cependant, la fonctionnalité d'un système de fichiers GFS2 est identique à celle d'un système de fichiers local.

Le module complémentaire de stockage résilient de Red Hat Enterprise Linux fournit GFS2 et dépend du module complémentaire de haute disponibilité de Red Hat Enterprise Linux pour fournir la gestion de cluster requise par GFS2.

Le module du noyau **gfs2.ko** implémente le système de fichiers GFS2 et est chargé sur les nœuds du cluster GFS2.

Pour obtenir les meilleures performances de GFS2, il est important de prendre en compte les considérations de performance qui découlent de la conception sous-jacente. Tout comme un système de fichiers local, GFS2 s'appuie sur le cache de pages pour améliorer les performances grâce à la mise en cache locale des données fréquemment utilisées. Afin de maintenir la cohérence entre les nœuds du cluster, le contrôle du cache est assuré par la machine d'état *glock*.

Ressources supplémentaires

- Configuration des systèmes de fichiers GFS2

1.4. VUE D'ENSEMBLE DU STOCKAGE EN GRAPPE

Red Hat Gluster Storage (RHGS) est une plateforme de stockage définie par logiciel qui peut être déployée en grappes. Elle regroupe les ressources de stockage sur disque de plusieurs serveurs en un seul espace de noms global. GlusterFS est un système de fichiers distribué open source qui convient aux solutions cloud et hybrides.

Les volumes constituent la base de GlusterFS et répondent à différentes exigences. Chaque volume est une collection de briques, qui sont des unités de base de stockage représentées par un répertoire d'exportation sur un serveur dans le pool de stockage de confiance.

Les types de volumes GlusterFS suivants sont disponibles :

- **Distributed GlusterFS volume** est le volume par défaut où chaque fichier est stocké dans une brique et où le fichier ne peut pas être partagé entre différentes briques.
- **Replicated GlusterFS volume** réplique les données de l'utilisateur, de sorte que si une brique tombe en panne, les données restent accessibles.
- **Distributed replicated GlusterFS volume** est un volume hybride qui répartit les répliques sur un grand nombre de systèmes. Il convient aux environnements où l'évolutivité et la fiabilité du stockage sont essentielles.

Ressources supplémentaires

- Guide d'administration du stockage gluster de Red Hat

1.5. APERÇU DU STOCKAGE CEPH

Red Hat Ceph Storage (RHCS) est une plateforme de stockage évolutive, ouverte et définie par logiciel qui combine la version la plus stable du système de stockage Ceph avec une plateforme de gestion Ceph, des utilitaires de déploiement et des services d'assistance.

Red Hat Ceph Storage est conçu pour l'infrastructure en nuage et le stockage d'objets à l'échelle du web. Les clusters de Red Hat Ceph Storage sont constitués des types de nœuds suivants :

Nœud d'administration Red Hat Ceph Storage Ansible

Ce type de nœud agit comme le nœud d'administration Ceph traditionnel pour les versions précédentes de Red Hat Ceph Storage. Ce type de nœud offre les fonctions suivantes :

- Gestion centralisée des grappes de stockage
- Les fichiers de configuration Ceph et les clés
- Optionnellement, des dépôts locaux pour l'installation de Ceph sur des nœuds qui ne peuvent pas accéder à l'Internet pour des raisons de sécurité

Surveiller les nœuds

Chaque nœud de surveillance exécute le démon de surveillance (**ceph-mon**), qui conserve une copie de la carte de la grappe. La carte de la grappe comprend la topologie de la grappe. Un client qui se connecte au cluster Ceph récupère la copie actuelle de la carte du cluster à partir du moniteur, ce qui lui permet de lire et d'écrire des données dans le cluster.

IMPORTANT

Ceph peut fonctionner avec un moniteur ; cependant, pour assurer une haute disponibilité dans un cluster de production, Red Hat ne prendra en charge que les déploiements avec au moins trois nœuds de moniteurs. Red Hat recommande de déployer un total de 5 moniteurs Ceph pour les clusters de stockage dépassant 750 OSD.

Nœuds OSD

Chaque nœud OSD (Object Storage Device) exécute le démon Ceph OSD (**ceph-osd**), qui interagit avec les disques logiques attachés au nœud. Ceph stocke les données sur ces nœuds OSD.
Ceph peut fonctionner avec très peu de nœuds OSD, trois par défaut, mais les clusters de production réalisent de meilleures performances à partir d'échelles modestes, par exemple 50 OSD dans un cluster de stockage. Idéalement, un cluster Ceph possède plusieurs nœuds OSD, ce qui permet d'isoler les domaines de défaillance en créant la carte CRUSH.

Nœuds MDS

Chaque nœud du serveur de métadonnées (MDS) exécute le démon MDS (**ceph-mds**), qui gère les métadonnées relatives aux fichiers stockés dans le système de fichiers Ceph (CephFS). Le démon MDS coordonne également l'accès au cluster partagé.

Nœud de passerelle d'objets

Le nœud Ceph Object Gateway exécute le démon Ceph RADOS Gateway (**ceph-radosgw**), et est une interface de stockage d'objets construite au-dessus de **librados** pour fournir aux applications une passerelle RESTful vers les clusters de stockage Ceph. La passerelle Ceph Object Gateway prend en charge deux interfaces :

S3

Fournit des fonctionnalités de stockage d'objets avec une interface compatible avec un large sous-ensemble de l'API RESTful Amazon S3.

Swift

Fournit des fonctionnalités de stockage d'objets avec une interface compatible avec un large sous-ensemble de l'API Swift d'OpenStack.

Ressources supplémentaires

- Stockage Red Hat Ceph

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 Example 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 myvl2 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 volume mylv1 LVM VDO dans le groupe de volumes myvg

```
---
- 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 :

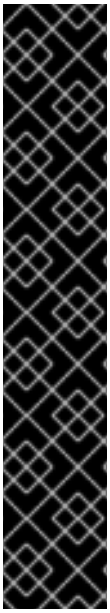
- Le site **deduplication** est utilisé pour dédupliquer les données dupliquées stockées sur le volume de stockage.
- La compression est utilisée pour comprimer les données stockées sur le volume de stockage, ce qui permet d'augmenter la capacité de stockage.
- Le paramètre vdo_pool_size indique la taille 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. PARTITIONS DE DISQUE

Pour diviser un disque en une ou plusieurs zones logiques, utilisez l'utilitaire de partitionnement du disque. Il permet de gérer séparément chaque partition.

3.1. VUE D'ENSEMBLE DES PARTITIONS

Le disque dur stocke les informations relatives à l'emplacement et à la taille de chaque partition dans la table de partition. En utilisant les informations de la table de partition, le système d'exploitation traite chaque partition comme un disque logique. Voici quelques-uns des avantages du partitionnement de disque :

- Réduire la probabilité d'oublis administratifs des volumes physiques
- Assurer une sauvegarde suffisante
- Assurer une gestion efficace des disques

Ressources supplémentaires

- Quels sont les avantages et les inconvénients de l'utilisation du partitionnement sur les LUN, soit directement, soit avec LVM entre les deux ?

3.2. CONSIDÉRATIONS À PRENDRE EN COMPTE AVANT DE MODIFIER LES PARTITIONS D'UN DISQUE

Avant de créer, de supprimer ou de redimensionner des partitions de disque, tenez compte des aspects suivants.

Sur un périphérique, le type de table de partition détermine le nombre et la taille maximum des partitions individuelles.

Nombre maximum de partitions :

- Sur un appareil formaté avec la table de partition **Master Boot Record (MBR)**, vous pouvez avoir :
  - Jusqu'à quatre partitions primaires.
  - Jusqu'à trois partitions primaires, une partition étendue
    - Plusieurs partitions logiques au sein de la partition étendue
- Sur un appareil formaté avec **GUID Partition Table (GPT)**, vous pouvez avoir :
  - Jusqu'à 128 partitions, si vous utilisez l'utilitaire **parted**.
    - Bien que la spécification GPT autorise davantage de partitions en augmentant la taille réservée de la table de partition, l'utilitaire parted limite la zone requise pour 128 partitions.

Taille maximale des partitions :

- Sur un appareil formaté avec la table de partition **Master Boot Record (MBR)**
  - Si vous utilisez des lecteurs à secteur de 512b, la taille maximale est de 2 TiB.

- Si vous utilisez des lecteurs à secteur 4k, la taille maximale est de 16 TiB.

- Sur un appareil formaté avec le logiciel **GUID Partition Table (GPT)**:
  - Lors de l'utilisation de lecteurs à secteurs de 512b, la taille maximale est de 8 ZiB.
  - Si vous utilisez des lecteurs à secteur 4k, la taille maximale est de 64 ZiB.

En utilisant l'utilitaire **parted**, vous pouvez spécifier la taille de la partition en utilisant plusieurs suffixes différents :

- **MiB, GiB, or TiB**
  - Taille exprimée en puissance de 2.
  - Le point de départ de la partition est aligné sur le secteur exact spécifié par la taille.
  - Le point final est aligné sur la taille spécifiée moins 1 secteur.
- **MB, GB, or TB**:
  - Taille exprimée en puissances de 10.
  - Les points de départ et d'arrivée sont alignés sur la moitié de l'unité spécifiée. Par exemple, ±500KB lorsque le suffixe MB est utilisé.

NOTE

Cette section ne couvre pas la table de partition DASD, qui est spécifique à l'architecture IBM Z.

Ressources supplémentaires

- Configuration d'une instance Linux sur IBM Z
- Ce qu'il faut savoir sur le DASD

3.3. COMPARAISON DES TYPES DE TABLES DE PARTITION

Pour activer les partitions sur un périphérique, formatez un périphérique bloc avec différents types de tables de partition. Le tableau suivant compare les propriétés des différents types de tables de partition que vous pouvez créer sur un périphérique de bloc.

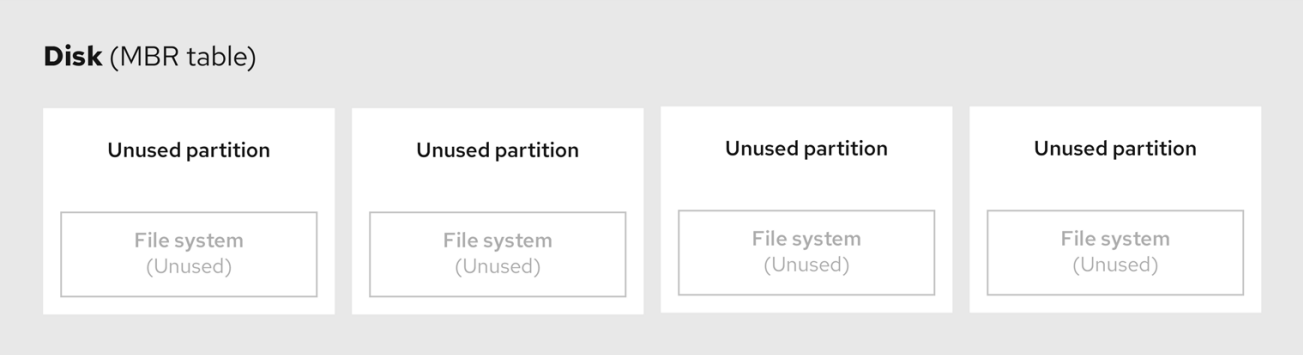
Tableau 3.1. Types de tables de partition

Table de partition	Nombre maximal de partitions	Taille maximale de la partition
Master Boot Record (MBR)	4 partitions primaires, ou 3 partitions primaires et 1 partition étendue avec 12 partitions logiques	2TiB
Table de partition GUID (GPT)	128	8ZiB

3.4. PARTITIONS DE DISQUE MBR

La table de partition est stockée au tout début du disque, avant tout système de fichiers ou toute donnée utilisateur. Pour un exemple plus clair, la table de partition est présentée comme étant séparée dans les diagrammes suivants.

Figure 3.1. Disque avec table de partition MBR



Comme le montre le schéma précédent, la table de partition est divisée en quatre sections de quatre partitions primaires inutilisées. Une partition primaire est une partition sur un disque dur qui ne contient qu'une seule unité logique (ou section). Chaque unité logique contient les informations nécessaires pour définir une seule partition, ce qui signifie que la table de partitions ne peut pas définir plus de quatre partitions primaires.

Chaque entrée de la table de partition contient des caractéristiques importantes de la partition :

- Les points du disque où la partition commence et se termine
- L'état de la partition, étant donné qu'une seule partition peut être signalée comme étant **active**
- Le type de partition

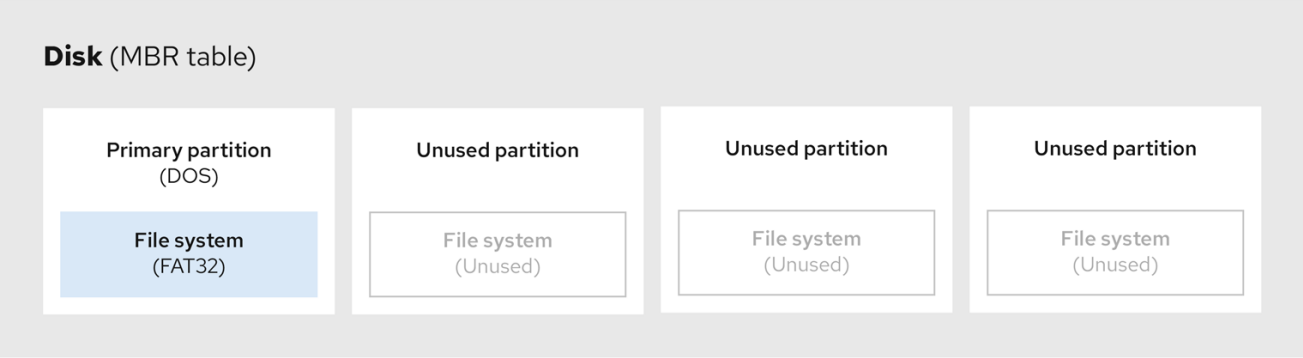
Les points de départ et d'arrivée définissent la taille et l'emplacement de la partition sur le disque. Certains chargeurs de démarrage de systèmes d'exploitation utilisent l'indicateur **active**. Cela signifie que le système d'exploitation de la partition marquée "active" est démarré.

Le type est un nombre qui identifie l'utilisation prévue d'une partition. Certains systèmes d'exploitation utilisent le type de partition pour :

- Indique un type de système de fichiers spécifique
- Marquer la partition comme étant associée à un système d'exploitation particulier
- Indiquer que la partition contient un système d'exploitation amorçable

Le diagramme suivant montre un exemple de disque avec une seule partition. Dans cet exemple, la première partition est étiquetée en tant que type de partition **DOS**:

Figure 3.2. Disque avec une seule partition



Ressources supplémentaires

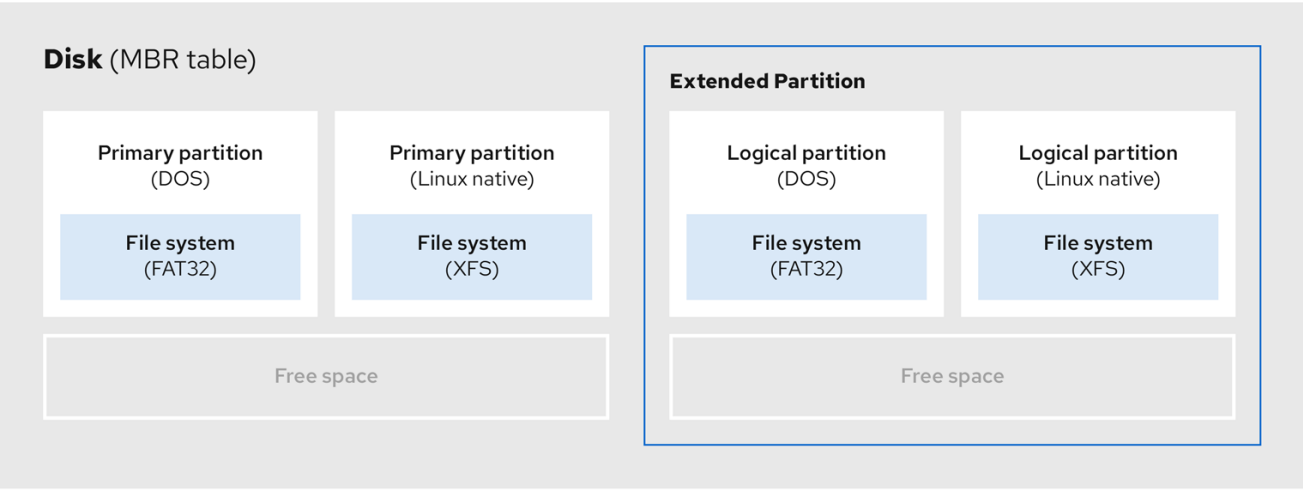
- Types de partitions MBR

3.5. PARTITIONS MBR ÉTENDUES

Pour créer des partitions supplémentaires, si nécessaire, définissez le type sur **extended**.

Une partition étendue est similaire à une unité de disque. Elle possède sa propre table de partition, qui pointe vers une ou plusieurs partitions logiques, entièrement contenues dans la partition étendue. Le diagramme suivant montre une unité de disque avec deux partitions primaires et une partition étendue contenant deux partitions logiques, ainsi qu'un espace libre non partitionné.

Figure 3.3. Disque avec deux partitions primaires et une partition étendue MBR



Vous ne pouvez avoir que quatre partitions primaires et étendues, mais il n'y a pas de limite fixe au nombre de partitions logiques. Pour limiter l'accès aux partitions dans Linux, une seule unité de disque permet un maximum de 15 partitions logiques.

3.6. TYPES DE PARTITIONS MBR

Le tableau ci-dessous présente une liste des types de partitions MBR les plus couramment utilisés et les nombres hexadécimaux qui les représentent.

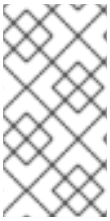
Tableau 3.2. Types de partitions MBR

MBR partition type	Value	MBR partition type	Value
Vide	00	Novell Netware 386	65
DOS 12-bit FAT	01	PIC/IX	75
Racine XENIX	O2	Ancien MINIX	80
XENIX usr	O3	Linux/MINUX	81
DOS 16-bit ⇐32M	04	Linux swap	82
Prolongé	05	Linux natif	83
DOS 16 bits >=32	06	Linux étendu	85
OS/2 HPFS	07	Amibe	93
AIX	08	Amibe BBT	94
AIX bootable	09	BSD/386	a5
Gestionnaire d'amorçage OS/2	0a	OpenBSD	a6
Win95 FAT32	0b	NEXTSTEP	a7
Win95 FAT32 (LBA)	0c	BSDI fs	b7
Win95 FAT16 (LBA)	0e	Échange BSDI	b8
Win95 étendu (LBA)	0f	Syrinx	c7
Venix 80286	40	CP/M	db
Novell	51	Accès DOS	e1
Botte PRep	41	DOS R/O	e3
GNU HURD	63	DOS secondaire	f2
Novell Netware 286	64	BBT	ff

3.7. TABLE DE PARTITION GUID

La table de partition GUID (GPT) est un schéma de partitionnement basé sur l'identifiant unique global (GUID).

GPT traite les limitations de la table de partition MBR (Mater Boot Record). La table de partition MBR ne peut pas prendre en charge un espace de stockage supérieur à 2 TiB, soit environ 2,2 To. En revanche, GPT prend en charge les disques durs de plus grande capacité. La taille maximale d'un disque adressable est de 8 ZiB, si l'on utilise des disques durs de 512b, et de 64 ZiB, si l'on utilise des disques durs de 4096b. En outre, par défaut, GPT prend en charge la création de 128 partitions primaires au maximum. Il est possible d'étendre le nombre maximal de partitions primaires en allouant plus d'espace à la table de partition.



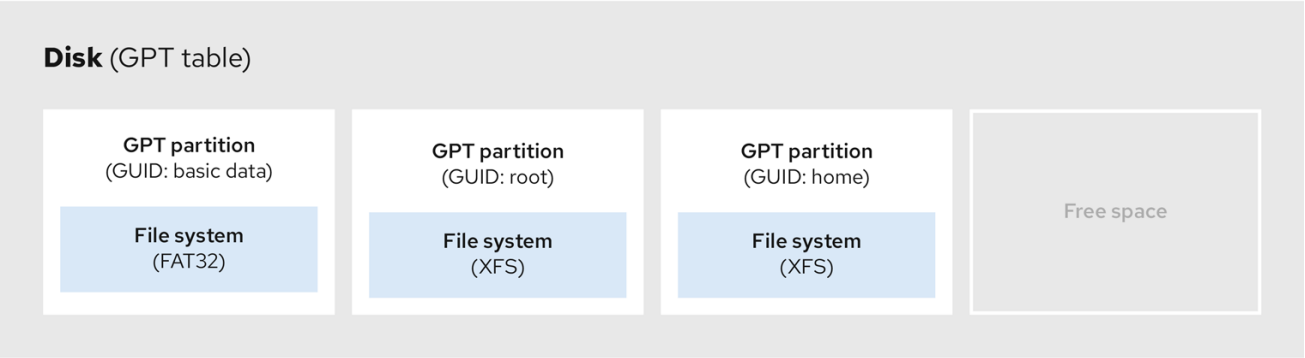
NOTE

Un GPT a des types de partitions basés sur des GUID. Certaines partitions nécessitent un GUID spécifique. Par exemple, la partition système pour les chargeurs de démarrage EFI (Extensible Firmware Interface) nécessite le GUID **C12A7328-F81F-11D2-BA4B-00A0C93EC93B**.

Les disques GPT utilisent l'adressage par blocs logiques (LBA) et une disposition de partition comme suit :

- Pour assurer la rétrocompatibilité avec les disques MBR, le système réserve le premier secteur (LBA 0) de GPT aux données MBR et applique le nom "MBR protecteur".
- GPT primaire
  - L'en-tête commence sur le deuxième bloc logique (LBA 1) du périphérique. L'en-tête contient le GUID du disque, l'emplacement de la table de partition primaire, l'emplacement de l'en-tête GPT secondaire et les sommes de contrôle CRC32 de lui-même et de la table de partition primaire. Il spécifie également le nombre d'entrées de partition dans la table.
  - Par défaut, le GPT primaire comprend 128 entrées de partition. Chaque partition a une taille d'entrée de 128 octets, un GUID de type de partition et un GUID de partition unique.
- GPT secondaire
  - Pour la récupération, elle est utile comme table de sauvegarde au cas où la table de partition primaire serait corrompue.
  - Le dernier secteur logique du disque contient l'en-tête GPT secondaire et récupère les informations GPT, au cas où l'en-tête primaire serait corrompu.
  - Il contient
    - Le GUID du disque
    - Emplacement de la table de partition secondaire et de l'en-tête GPT primaire
    - Somme de contrôle CRC32 de lui-même
    - La table de partition secondaire
    - Le nombre d'entrées de partition possibles

Figure 3.4. Disque avec table de partition GUID



IMPORTANT

Pour que l'installation du chargeur de démarrage sur un disque GPT réussisse, une partition de démarrage du BIOS doit être présente. La réutilisation n'est possible que si le disque contient déjà une partition d'amorçage du BIOS. Cela inclut les disques initialisés par le programme d'installation **Anaconda**.

3.8. TYPES DE PARTITION

Il existe plusieurs façons de gérer les types de partitions :

- L'utilitaire **fdisk** prend en charge l'ensemble des types de partition en spécifiant des codes hexadécimaux.
- L'utilitaire de génération d'unités **systemd-gpt-auto-generator** utilise le type de partition pour identifier et monter automatiquement les périphériques.
- L'utilitaire **parted** détermine le type de partition à l'aide de *flags*. L'utilitaire **parted** ne gère que les types de partitions de type Cératine, par exemple LVM, swap ou RAID.
  L'utilitaire **parted** permet de définir les drapeaux suivants :
  - **boot**
  - **root**
  - **swap**
  - **hidden**
  - **raid**
  - **lvm**
  - **lba**
  - **legacy_boot**
  - **irst**
  - **esp**

- **palo**

Sur Red Hat Enterprise Linux 9 avec **parted** 3.5, vous pouvez utiliser les drapeaux supplémentaires **chromeos_kernel** et **bls_boot**.

L'utilitaire **parted** accepte facultativement un argument de type de système de fichiers lors de la création d'une partition. Voir Créer une partition avec parted

pour une liste des conditions requises. Utilisez la valeur pour :

- Définir les drapeaux de partition sur le MBR.
- Définissez le type d'UUID de la partition sur GPT. Par exemple, les types de systèmes de fichiers **swap**, **fat** ou **hfs** définissent des GUID différents. La valeur par défaut est le GUID de données Linux.

L'argument ne modifie pas le système de fichiers sur la partition. Il ne fait que différencier les drapeaux et les GUID pris en charge.

Les types de systèmes de fichiers suivants sont pris en charge :

- **xfs**
- **ext2**
- **ext3**
- **ext4**
- **fat16**
- **fat32**
- **hfs**
- **hfs**
- **linux-swap**
- **ntfs**
- **reiserfs**

3.9. SCHÉMA DE DÉNOMINATION DES PARTITIONS

Red Hat Enterprise Linux utilise un système de dénomination basé sur les fichiers, avec des noms de fichiers sous la forme de **/dev/*xxyN***.

Les noms de périphériques et de partitions sont constitués de la structure suivante :

/dev/

Nom du répertoire qui contient tous les fichiers de l'appareil. Les disques durs contiennent des partitions ; les fichiers représentant toutes les partitions possibles se trouvent donc à l'adresse **/dev**.

xx

Les deux premières lettres du nom de la partition indiquent le type de périphérique qui contient la partition.

y

Cette lettre indique le périphérique spécifique contenant la partition. Par exemple, **/dev/sda** pour le premier disque dur et **/dev/sdb** pour le second. Vous pouvez utiliser plus de lettres dans les systèmes comportant plus de 26 disques, par exemple, **/dev/sdaa1**.

N

La dernière lettre indique le numéro de la partition. Les quatre premières partitions (primaires ou étendues) sont numérotées de **1** à **4**. Les partitions logiques commencent à **5**. Par exemple, **/dev/sda3** est la troisième partition primaire ou étendue du premier disque dur, et **/dev/sdb6** est la deuxième partition logique du deuxième disque dur. La numérotation des partitions de disque ne s'applique qu'aux tables de partition MBR. Notez que ***N*** ne signifie pas toujours partition.

NOTE

Même si Red Hat Enterprise Linux peut identifier et se référer à *all* types de partitions de disque, il peut ne pas être en mesure de lire le système de fichiers et donc d'accéder aux données stockées sur chaque type de partition. Cependant, dans de nombreux cas, il est possible d'accéder avec succès aux données d'une partition dédiée à un autre système d'exploitation.

3.10. POINTS DE MONTAGE ET PARTITIONS DE DISQUE

Dans Red Hat Enterprise Linux, chaque partition constitue une partie du stockage, nécessaire pour prendre en charge un seul ensemble de fichiers et de répertoires. Le montage d'une partition rend le stockage de cette partition disponible, en commençant par le répertoire spécifié, connu sous le nom de *mount point*.

Par exemple, si la partition **/dev/sda5** est montée sur **/usr/**, cela signifie que tous les fichiers et répertoires sous **/usr/** résident physiquement sur **/dev/sda5**. Le fichier **/usr/share/doc/FAQ/txt/Linux-FAQ** réside sur **/dev/sda5**, alors que le fichier **/etc/gdm/custom.conf** n'y réside pas.

Si l'on poursuit l'exemple, il est également possible qu'un ou plusieurs répertoires situés sous **/usr/** soient des points de montage pour d'autres partitions. Par exemple, **/usr/local/man/whatis** réside sur **/dev/sda7**, plutôt que sur **/dev/sda5**, si **/usr/local** comprend une partition **/dev/sda7** montée.

CHAPITRE 4. COMMENCER AVEC LES PARTITIONS

Le partitionnement du disque permet de diviser un disque en une ou plusieurs zones logiques, ce qui permet de travailler sur chaque partition séparément. Le disque dur stocke les informations relatives à l'emplacement et à la taille de chaque partition dans la table de partition. Grâce à cette table, chaque partition apparaît comme un disque logique au système d'exploitation. Vous pouvez alors lire et écrire sur ces disques individuels.

Pour une vue d'ensemble des avantages et des inconvénients de l'utilisation de partitions sur des périphériques en bloc, voir Quels sont les avantages et les inconvénients de l'utilisation du partitionnement sur des LUN, soit directement, soit avec LVM entre les deux ?

4.1. 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 4.1. 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.

4.2. 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.

4.3. 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-swap**, **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 4.2. 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
```

Ressources supplémentaires

- **parted(8)** man page.
- Création d'une table de partition sur un disque avec parted .
- Redimensionnement d'une partition avec parted

4.4. DÉFINIR UN TYPE DE PARTITION AVEC FDISK

Vous pouvez définir un type de partition ou un drapeau à l'aide de l'utilitaire **fdisk**.

Conditions préalables

- Une partition sur le disque.

Procédure

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

```
# fdisk block-device
```

2. Affichez la table de partition actuelle pour déterminer le numéro de la partition mineure :

```
Commande (m pour l'aide) : print
```

Vous pouvez voir le type de partition actuel dans la colonne **Type** et son ID de type correspondant dans la colonne **Id**.

3. Entrez la commande de type de partition et sélectionnez une partition à l'aide de son numéro mineur :

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

4. Facultatif : Afficher la liste en codes hexadécimaux :

```
Code hexagonal (tapez L pour obtenir la liste de tous les codes) : L
```

5. Définir le type de partition :

```
Code hexagonal (tapez L pour obtenir la liste de tous les codes) : 8e
```

6. Écrivez vos modifications et quittez le shell **fdisk**:

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

7. Vérifiez vos modifications :

```
# fdisk --list block-device
```

4.5. 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

4.6. 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

CHAPITRE 5. 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.

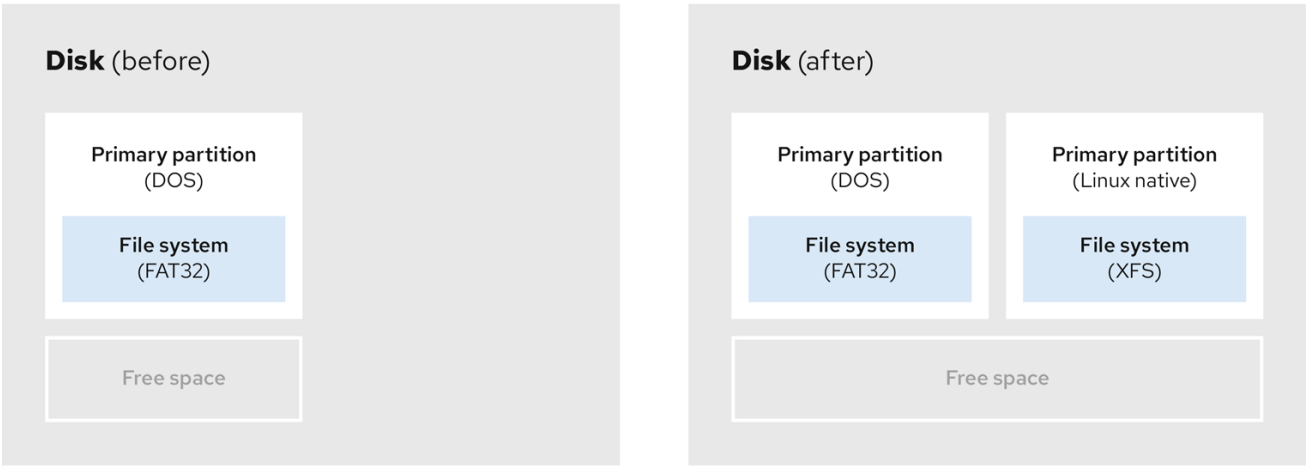
NOTE

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.

5.1. 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 5.1. Disque avec espace libre non partitionné



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.

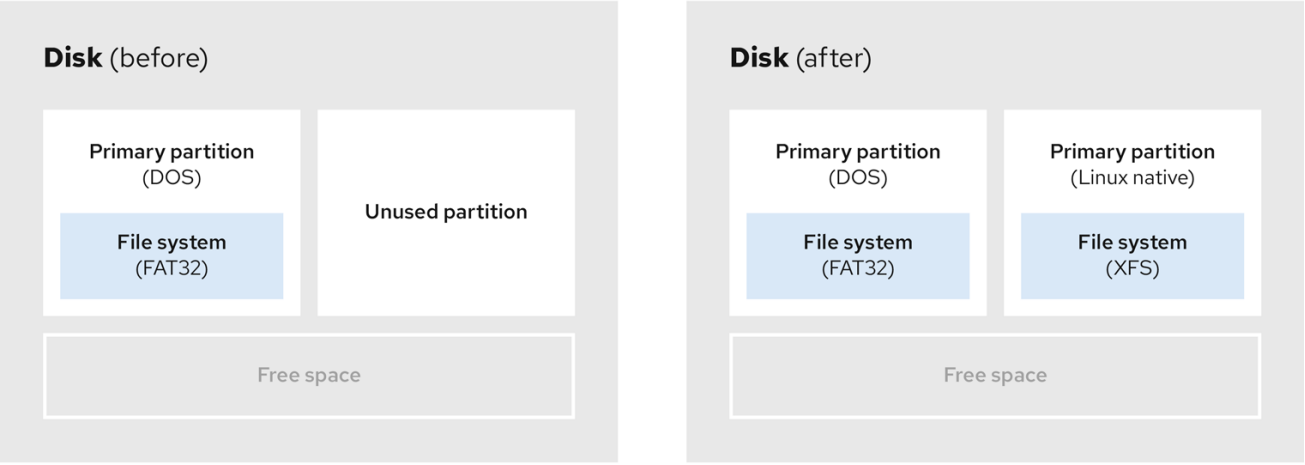
Un disque dur inutilisé entre également dans cette catégorie. La seule différence est que *all* l'espace ne fait partie d'aucune partition définie.

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.

5.2. 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 5.2. Disque avec une partition inutilisée



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.

5.3. 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.

5.3.1. 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.

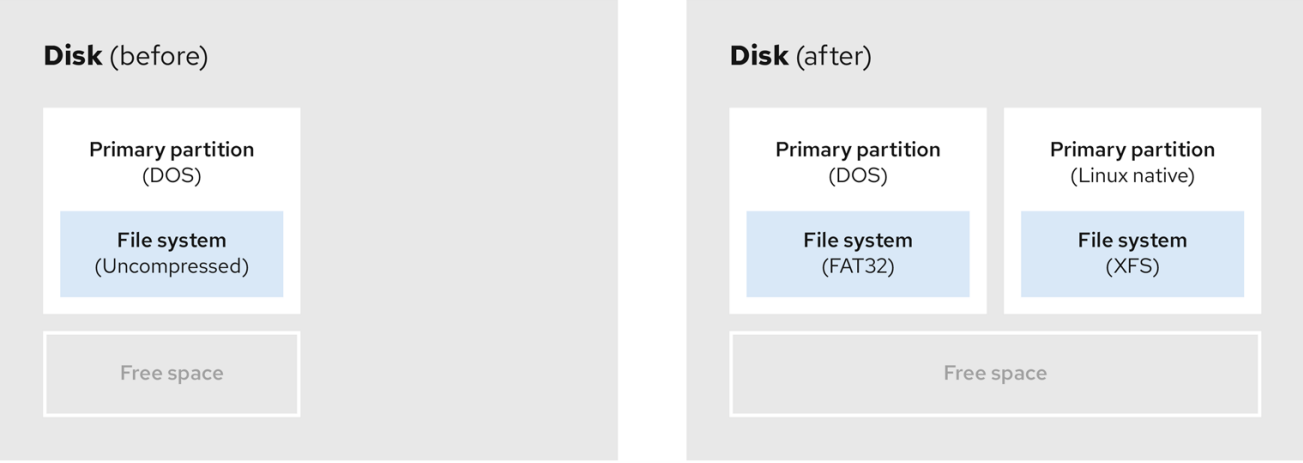
Après avoir créé une partition plus petite pour votre système d'exploitation existant, vous pouvez.. :

- 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 5.3. Action de repartitionnement destructive sur le disque



AVERTISSEMENT

Cette méthode supprime toutes les données précédemment stockées dans la partition d'origine.

5.3.2. 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.

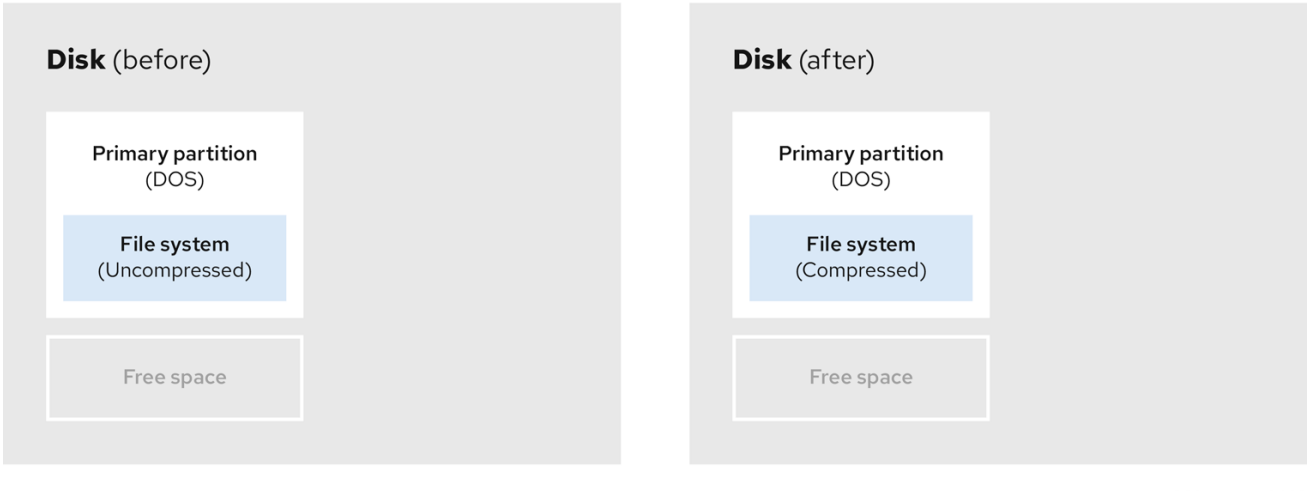
Voici une liste de méthodes qui peuvent aider à initier un repartitionnement non destructif.

- 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 repartition 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 5.4. Compression des données sur un disque



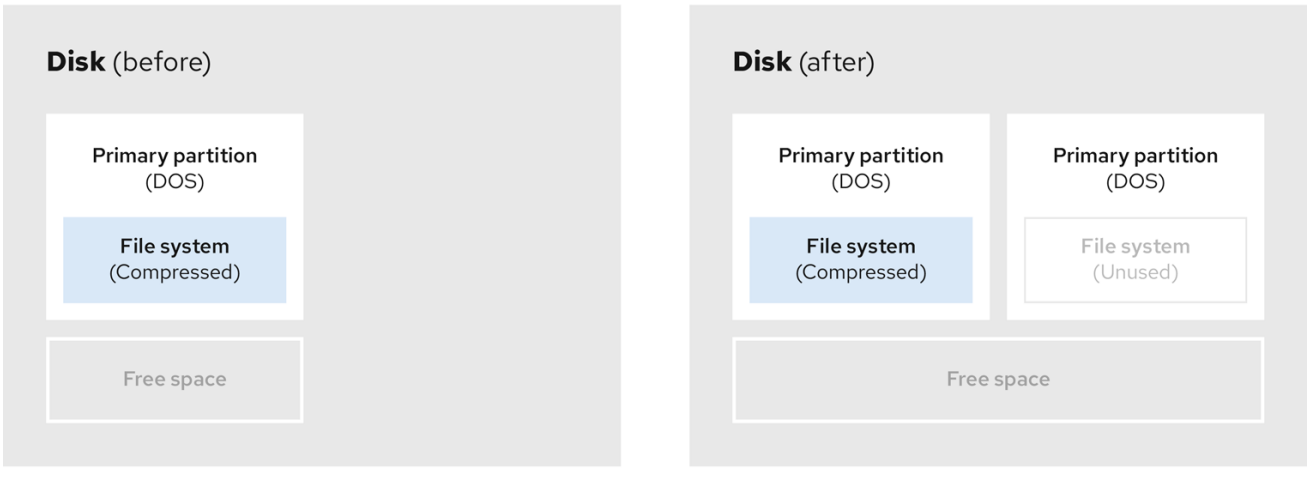
Pour éviter toute perte de données, créez une sauvegarde avant de poursuivre le processus de compression.

- 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 5.5. Redimensionnement d'une partition sur un disque

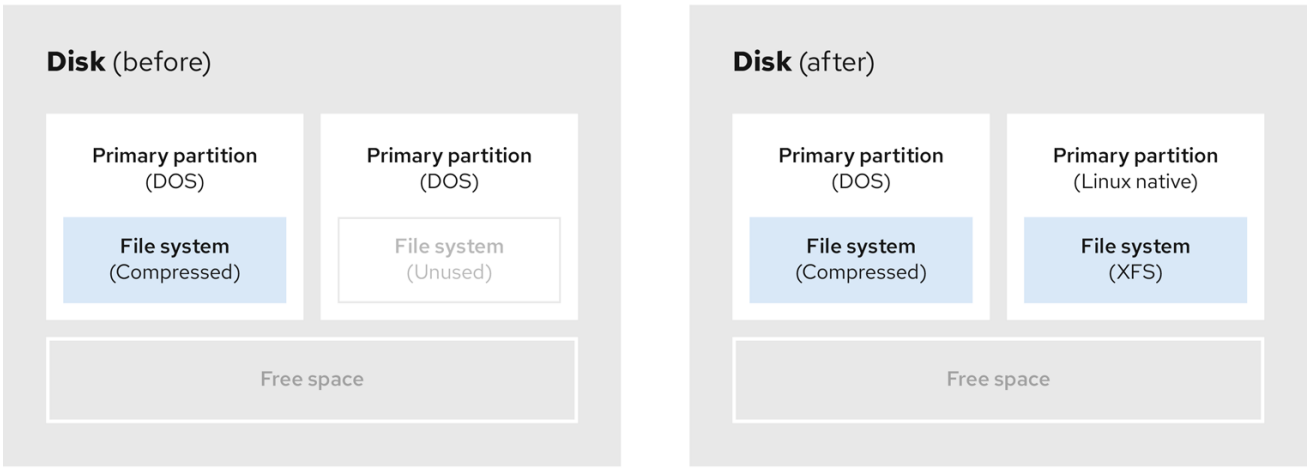


- Facultatif : Créer de nouvelles partitions

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 5.6. Disque avec configuration finale de la partition



CHAPITRE 6. CONFIGURATION D'UNE CIBLE ISCSI

Red Hat Enterprise Linux utilise l'interpréteur de commandes **targetcli** comme interface de ligne de commande pour effectuer les opérations suivantes :

- Ajouter, supprimer, visualiser et surveiller les interconnexions de stockage iSCSI pour utiliser le matériel iSCSI.
- Exporter vers des systèmes distants des ressources de stockage locales sauvegardées par des fichiers, des volumes, des périphériques SCSI locaux ou des disques RAM.

L'outil **targetcli** a une présentation arborescente et comprend une complétion de tabulation intégrée, un support d'auto-complétion et une documentation en ligne.

6.1. INSTALLATION DE TARGETCLI

Installez l'outil **targetcli** pour ajouter, surveiller et supprimer les interconnexions de stockage iSCSI.

Procédure

1. Installer l'outil **targetcli**:

```
# dnf install targetcli
```

2. Démarrer le service cible :

```
# systemctl start target
```

3. Configurer la cible pour qu'elle démarre au moment du démarrage :

```
# systemctl enable target
```

4. Ouvrez le port **3260** dans le pare-feu et rechargez la configuration du pare-feu :

```
# firewall-cmd --permanent --add-port=3260/tcp
Success

# firewall-cmd --reload
Success
```

Vérification

- Voir le site **targetcli**:

```
# targetcli
/> ls
o- /.....[...]
  o- backstores.....[...]
  | o- block.....[Storage Objects: 0]
  | o- fileio.....[Storage Objects: 0]
  | o- pscsi.....[Storage Objects: 0]
```

```
| o- ramdisk.....[Storage Objects: 0]
o- iscsi.....[Targets: 0]
o- loopback.....[Targets: 0]
```

Ressources supplémentaires

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

6.2. CRÉATION D'UNE CIBLE ISCSI

La création d'une cible iSCSI permet à l'initiateur iSCSI du client d'accéder aux périphériques de stockage du serveur. Les cibles et les initiateurs ont tous deux des noms d'identification uniques.

Conditions préalables

- Installation et exécution de **targetcli**. Pour plus d'informations, voir Installation de targetcli.

Procédure

1. Naviguez jusqu'au répertoire iSCSI :

```
/> iscsi/
```

NOTE

La commande **cd** est utilisée pour changer de répertoire ainsi que pour répertorier le chemin à emprunter.

2. Utilisez l'une des options suivantes pour créer une cible iSCSI :

   a. Création d'une cible iSCSI à l'aide d'un nom de cible par défaut :

```
/iscsi> create

Created target
iqn.2003-01.org.linux-iscsi.hostname.x8664:sn.78b473f296ff
Created TPG1
```

   b. Création d'une cible iSCSI à l'aide d'un nom spécifique :

```
/iscsi> create iqn.2006-04.com.example:444

Created target iqn.2006-04.com.example:444
Created TPG1
Here iqn.2006-04.com.example:444 is target_iqn_name
```

   Remplacer *iqn.2006-04.com.example:444* par le nom spécifique de la cible.

3. Vérifier la cible nouvellement créée :

```
/iscsi> ls

o- iscsi.....[1 Target]
```

```
o- iqn.2006-04.com.example:444.....[1 TPG]
  o- tpg1.....[enabled, auth]
    o- acls.....[0 ACL]
     o- luns.....[0 LUN]
    o- portals.....[0 Portal]
```

Ressources supplémentaires

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

6.3. BACKSTORE ISCSI

Un backstore iSCSI permet de prendre en charge différentes méthodes de stockage des données d'un LUN exporté sur la machine locale. La création d'un objet de stockage définit les ressources utilisées par le backstore.

Un administrateur peut choisir l'un des périphériques de stockage suivants pris en charge par Linux-IO (LIO) :

fileio backstore

Créez un objet de stockage **fileio** si vous utilisez des fichiers ordinaires du système de fichiers local comme images disque. Pour créer un backstore **fileio**, voir Création d'un objet de stockage fileio .

block backstore

Créez un objet de stockage **block** si vous utilisez un périphérique bloc local et un périphérique logique. Pour créer un backstore **block**, voir Création d'un objet de stockage en bloc .

pscsi backstore

Créez un objet de stockage **pscsi** si votre objet de stockage prend en charge la transmission directe des commandes SCSI. Pour créer un backstore **pscsi**, voir Création d'un objet de stockage pscsi .

ramdisk backstore

Créez un objet de stockage **ramdisk** si vous souhaitez créer un périphérique RAM temporaire. Pour créer un backstore **ramdisk**, voir Création d'un objet de stockage disque RAM Memory Copy .

Ressources supplémentaires

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

6.4. CRÉATION D'UN OBJET DE STOCKAGE FILEIO

fileio peuvent prendre en charge les opérations **write_back** ou **write_thru**. L'opération **write_back** active le cache du système de fichiers local. Elle améliore les performances mais augmente le risque de perte de données.

Il est recommandé d'utiliser **write_back=false** pour désactiver l'opération **write_back** au profit de l'opération **write_thru**.

Conditions préalables

- Installation et exécution de **targetcli**. Pour plus d'informations, voir Installation de targetcli.

Procédure

1. Naviguez vers le site **fileio**/ à partir du répertoire **backstores**/:

```
/> backstores/fileio
```

2. Créer un objet de stockage **fileio**:

```
/backstores/fileio> create file1 /tmp/disk1.img 200M write_back=false

Created fileio file1 with size 209715200
```

Vérification

- Vérifiez l'objet de stockage **fileio** créé :

```
/backstores/fileio> ls
```

Ressources supplémentaires

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

6.5. CRÉATION D'UN OBJET DE STOCKAGE EN BLOC

Le pilote de bloc permet d'utiliser avec Linux-IO (LIO) n'importe quel périphérique de bloc apparaissant dans le répertoire **/sys/block/**. Cela inclut les périphériques physiques tels que les disques durs, les disques SSD, les CD et les DVD, et les périphériques logiques tels que les volumes RAID logiciels ou matériels, ou les volumes LVM.

Conditions préalables

- Installation et exécution de **targetcli**. Pour plus d'informations, voir Installation de targetcli.

Procédure

1. Naviguez vers le site **block/** à partir du répertoire **backstores/**:

```
/> backstores/block/
```

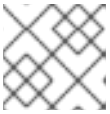
2. Créer un backstore **block**:

```
/backstores/block> create name=block_backend dev=/dev/sdb

Generating a wwn serial.
Created block storage object block_backend using /dev/vdb.
```

Vérification

- Vérifiez l'objet de stockage **block** créé :

```
/backstores/block> ls
```



NOTE

Vous pouvez également créer un backstore **block** sur un volume logique.

Ressources supplémentaires

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

6.6. CRÉATION D'UN OBJET DE STOCKAGE PSCSI

Vous pouvez configurer, en tant que backstore, tout objet de stockage qui prend en charge la transmission directe des commandes SCSI sans émulation SCSI, et avec un périphérique SCSI sous-jacent qui apparaît avec **lsscsi** dans le site **/proc/scsi/scsi**, tel qu'un disque dur SAS. Ce sous-système prend en charge les systèmes SCSI-3 et supérieurs.

AVERTISSEMENT

pscsi ne doivent être utilisées que par des utilisateurs expérimentés. Les commandes SCSI avancées telles que l'affectation d'unités logiques asymétriques (ALUA) ou les réservations persistantes (par exemple, celles utilisées par VMware ESX et vSphere) ne sont généralement pas implémentées dans le micrologiciel du périphérique et peuvent provoquer des dysfonctionnements ou des pannes. En cas de doute, utilisez plutôt **block** backstore pour les configurations de production.

Conditions préalables

- Installation et exécution de **targetcli**. Pour plus d'informations, voir Installation de targetcli.

Procédure

1. Naviguez vers le site **pscsi/** à partir du répertoire **backstores/**:

```
/> backstores/pscsi/
```

2. Créez un backstore **pscsi** pour un périphérique SCSI physique, un périphérique TYPE_ROM utilisant **/dev/sr0** dans cet exemple :

```
/backstores/pscsi> create name=pscsi_backend dev=/dev/sr0

Generating a wwn serial.
Created pscsi storage object pscsi_backend using /dev/sr0
```

Vérification

- Vérifiez l'objet de stockage **pscsi** créé :

```
/backstores/pscsi> ls
```

Ressources supplémentaires

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

6.7. CRÉATION D'UN OBJET DE STOCKAGE SUR DISQUE RAM DE TYPE MEMORY COPY

Les disques RAM à copie de mémoire (**ramdisk**) fournissent des disques RAM avec une émulation SCSI complète et des mappages de mémoire séparés utilisant la copie de mémoire pour les initiateurs. Cela permet des sessions multiples et est particulièrement utile pour un stockage de masse rapide et volatile à des fins de production.

Conditions préalables

- Installation et exécution de **targetcli**. Pour plus d'informations, voir Installation de targetcli.

Procédure

1. Naviguez vers le site **ramdisk**/ à partir du répertoire **backstores**/:

```
/> backstores/ramdisk/
```

2. Créez un disque RAM backstore de 1 Go :

```
/backstores/ramdisk> create name=rd_backend size=1GB

Generating a wwn serial.
Created rd_mcp ramdisk rd_backend with size 1GB.
```

Vérification

- Vérifiez l'objet de stockage **ramdisk** créé :

```
/backstores/ramdisk> ls
```

Ressources supplémentaires

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

6.8. CRÉATION D'UN PORTAIL ISCSI

La création d'un portail iSCSI ajoute une adresse IP et un port à la cible qui reste activée.

Conditions préalables

- Installation et exécution de **targetcli**. Pour plus d'informations, voir Installation de targetcli.
- Une cible iSCSI associée à un groupe de portail cible (TPG). Pour plus d'informations, voir Création d'une cible iSCSI.

Procédure

1. Naviguez jusqu'au répertoire TPG :

```
/iscsi> iqn.2006-04.example:444/tpg1/
```

2. Utilisez l'une des options suivantes pour créer un portail iSCSI :

   a. La création d'un portail par défaut utilise le port iSCSI par défaut **3260** et permet à la cible d'écouter toutes les adresses IP sur ce port :

```
/iscsi/iqn.20...mple:444/tpg1> portals/ create

Using default IP port 3260
Binding to INADDR_Any (0.0.0.0)
Created network portal 0.0.0.0:3260
```

NOTE

Lorsqu'une cible iSCSI est créée, un portail par défaut est également créé. Ce portail est configuré pour écouter toutes les adresses IP avec le numéro de port par défaut, à savoir : **0.0.0.0:3260**.

Pour supprimer le portail par défaut, utilisez la commande suivante :

```
/iscsi/iqn-name/tpg1/portals delete ip_address=0.0.0.0 ip_port=3260
```

   b. Création d'un portail utilisant une adresse IP spécifique :

```
/iscsi/iqn.20...mple:444/tpg1> portals/ create 192.168.122.137

Using default IP port 3260
Created network portal 192.168.122.137:3260
```

Vérification

- Vérifiez le portail nouvellement créé :

```
/iscsi/iqn.20...mple:444/tpg1> ls

o- tpg..... [enambled, auth]
   o- acls .....[0 ACL]
   o- luns .....[0 LUN]
   o- portals .....[1 Portal]
     o- 192.168.122.137:3260.....[OK]
```

Ressources supplémentaires

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

6.9. CRÉATION D'UN LUN ISCSI

Le numéro d'unité logique (LUN) est un périphérique physique qui est sauvegardé par le backstore iSCSI. Chaque LUN a un numéro unique.

Conditions préalables

- Installation et exécution de **targetcli**. Pour plus d'informations, voir Installation de targetcli.

- Une cible iSCSI associée à un groupe de portail cible (TPG). Pour plus d'informations, voir Création d'une cible iSCSI.
- Objets de stockage créés. Pour plus d'informations, voir iSCSI Backstore.

Procédure

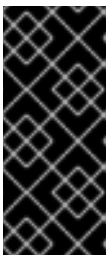
1. Créer des LUN d'objets de stockage déjà créés :

```
/iscsi/iqn.20...mple:444/tpg1> luns/ create /backstores/ramdisk/rd_backend
Created LUN 0.

/iscsi/iqn.20...mple:444/tpg1> luns/ create /backstores/block/block_backend
Created LUN 1.

/iscsi/iqn.20...mple:444/tpg1> luns/ create /backstores/fileio/file1
Created LUN 2.
```

2. Vérifiez les LUN créés :

```
/iscsi/iqn.20...mple:444/tpg1> ls

o- tpg..... [enambled, auth]
   o- acls .....[0 ACL]
   o- luns .....[3 LUNs]
   |  o- lun0.....[ramdisk/ramdisk1]
   |  o- lun1.....[block/block1 (/dev/vdb1)]
   |  o- lun2.....[fileio/file1 (/foo.img)]
   o- portals .....[1 Portal]
      o- 192.168.122.137:3260.....[OK]
```

   Le nom du LUN par défaut commence par **0**.



IMPORTANT

Par défaut, les LUN sont créés avec des permissions de lecture-écriture. Si un nouveau LUN est ajouté après la création des ACL, le LUN est automatiquement mappé à tous les ACL disponibles, ce qui peut entraîner un risque de sécurité. Pour créer une LUN avec des autorisations en lecture seule, voir Création d'une LUN iSCSI en lecture seule.

3. Configurez les ACL. Pour plus d'informations, voir Création d'une ACL iSCSI.

Ressources supplémentaires

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

6.10. CRÉATION D'UN LUN ISCSI EN LECTURE SEULE

Par défaut, les LUN sont créés avec des permissions de lecture-écriture. Cette procédure décrit comment créer une LUN en lecture seule.

Conditions préalables

- Installation et exécution de **targetcli**. Pour plus d'informations, voir Installation de targetcli.
- Une cible iSCSI associée à un groupe de portail cible (TPG). Pour plus d'informations, voir Création d'une cible iSCSI.
- Objets de stockage créés. Pour plus d'informations, voir iSCSI Backstore.

Procédure

1. Définir les autorisations de lecture seule :

```
/> set global auto_add_mapped_luns=false

Parameter auto_add_mapped_luns is now 'false'.
```

   Cela empêche le mappage automatique des LUN aux ACL existants, ce qui permet le mappage manuel des LUN.

2. Naviguez jusqu'au répertoire *initiator_iqn_name*:

```
/> iscsi/target_iqn_name/tpg1/acls/initiator_iqn_name/
```

3. Créer le LUN :

```
/iscsi/nom_de_la_cible/tpg1/acls/initiator_iqn_name> create
mapped_lun=next_sequential_LUN_number tpg_lun_ou_backstore=backstore
write_protect=1
```

   Exemple :

```
/iscsi/target_iqn_name/tpg1/acls/2006-04.com.example:888> create mapped_lun=1
tpg_lun_or_backstore=/backstores/block/block2 write_protect=1

Created LUN 1.
Created Mapped LUN 1.
```

4. Vérifiez le LUN créé :

```
/iscsi/target_iqn_name/tpg1/acls/2006-04.com.example:888> ls
 o- 2006-04.com.example:888 .. [Mapped LUNs: 2]
 | o- mapped_lun0 ..... [lun0 block/disk1 (rw)]
 | o- mapped_lun1 ..... [lun1 block/disk2 (ro)]
```

   La ligne mapped_lun1 comporte désormais (**ro**) à la fin (contrairement à la ligne mapped_lun0 (**rw**)), indiquant qu'elle est en lecture seule.

5. Configurez les ACL. Pour plus d'informations, voir Création d'une ACL iSCSI.

Ressources supplémentaires

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

6.11. CRÉATION D'UNE ACL ISCSI

Le service **targetcli** utilise des listes de contrôle d'accès (ACL) pour définir des règles d'accès et accorder à chaque initiateur l'accès à un numéro d'unité logique (LUN).

Les cibles et les initiateurs ont des noms d'identification uniques. Vous devez connaître le nom unique de l'initiateur pour configurer les ACL. Le fichier **/etc/iscsi/initiatorname.iscsi**, fourni par le paquetage **iscsi-initiator-utils**, contient les noms des initiateurs iSCSI.

Conditions préalables

- Le service **targetcli** est installé et fonctionne.
- Une cible iSCSI associée à un groupe de portail cible (TPG).

Procédure

1. Facultatif : Pour désactiver le mappage automatique des LUN aux ACL, voir Création d'un LUN iSCSI en lecture seule.
2. Naviguez jusqu'au répertoire acls :

```
/> iscsi/target_iqn_name/tpg_name/acls/
```

3. Utilisez l'une des options suivantes pour créer une liste de contrôle d'accès :
   - Utilisez le *initiator_iqn_name* du fichier **/etc/iscsi/initiatorname.iscsi** sur l'initiateur :

```
iscsi/target_iqn_name/tpg_name/acls> create initiator_iqn_name

Created Node ACL for initiator_iqn_name
Created mapped LUN 2.
Created mapped LUN 1.
Created mapped LUN 0.
```

   - Utilisez une adresse *custom_name* et mettez à jour l'initiateur pour qu'il corresponde à cette adresse :

```
iscsi/target_iqn_name/tpg_name/acls> create custom_name

Created Node ACL for custom_name
Created mapped LUN 2.
Created mapped LUN 1.
Created mapped LUN 0.
```

     Pour plus d'informations sur la mise à jour du nom de l'initiateur, voir Création d'un initiateur iSCSI.

Vérification

- Vérifiez l'ACL créé :

```
iscsi/target_iqn_name/tpg_name/acls> ls

o- acls .....[1 ACL]
  o- target_iqn_name ....[3 Mapped LUNs, auth]
```

```
o- mapped_lun0 .....[lun0 ramdisk/ramdisk1 (rw)]
o- mapped_lun1 .....[lun1 block/block1 (rw)]
o- mapped_lun2 .....[lun2 fileio/file1 (rw)]
```

Ressources supplémentaires

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

6.12. CONFIGURATION DU PROTOCOLE D'AUTHENTIFICATION CHALLENGE-HANDSHAKE POUR LA CIBLE

En utilisant **Challenge-Handshake Authentication Protocol (CHAP)**, les utilisateurs peuvent protéger la cible par un mot de passe. L'initiateur doit connaître ce mot de passe pour pouvoir se connecter à la cible.

Conditions préalables

- Création d'un ACL iSCSI. Pour plus d'informations, voir Création d'une ACL iSCSI .

Procédure

1. Définir l'authentification des attributs :

```
/iscsi/iqn.20...mple:444/tpg1> set attribute authentication=1

Parameter authentication is now '1'.
```

2. Set **userid** et **password**:

```
/tpg1> set auth userid=redhat
Parameter userid is now 'redhat'.

/iscsi/iqn.20...689dcbb3/tpg1> set auth password=redhat_passwd
Parameter password is now 'redhat_passwd'.
```

Ressources supplémentaires

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

6.13. SUPPRESSION D'UN OBJET ISCSI À L'AIDE DE L'OUTIL TARGETCLI

Cette procédure décrit comment supprimer les objets iSCSI à l'aide de l'outil **targetcli**.

Procédure

1. Se déconnecter de la cible :

```
# iscsiadm -m node -T iqn.2006-04.example:444 -u
```

Pour plus d'informations sur la façon de se connecter à la cible, voir Création d'un initiateur iSCSI.

2. Supprimez l'ensemble de la cible, y compris les ACL, les LUN et les portails :

```
/> iscsi/ delete iqn.2006-04.com.example:444
```

Remplacez *iqn.2006-04.com.example:444* par le nom de l'utilisateur cible.

- Pour supprimer un backstore iSCSI :

```
/> backstores/backstore-type/ delete block_backend
```

  - Remplacez *backstore-type* par **fileio**, **block**, **pscsi**, ou **ramdisk**.
  - Remplacez *block_backend* par le *backstore-name* que vous souhaitez supprimer.

- Pour supprimer des parties d'une cible iSCSI, comme une ACL :

```
/> /iscsi/nom-iqn/tpg/acls/ delete iqn.2006-04.com.example:444
```

Vérification

- Voir les changements :

```
/> iscsi/ ls
```

Ressources supplémentaires

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

CHAPITRE 7. CONFIGURATION D'UN INITIATEUR ISCSI

Un initiateur iSCSI forme une session pour se connecter à la cible iSCSI. Par défaut, un service iSCSI est démarré paresseusement et le service démarre après l'exécution de la commande **iscsiadm**. Si root n'est pas sur un périphérique iSCSI ou s'il n'y a pas de nœuds marqués par **node.startup = automatic**, le service iSCSI ne démarrera pas avant l'exécution d'une commande **iscsiadm** qui nécessite le démarrage de **iscsid** ou des modules du noyau **iscsi**.

Exécutez la commande **systemctl start iscsid.service** en tant que root pour forcer le démon **iscsid** à s'exécuter et les modules du noyau iSCSI à se charger.

7.1. CRÉATION D'UN INITIATEUR ISCSI

Créez un initiateur iSCSI pour vous connecter à la cible iSCSI afin d'accéder aux périphériques de stockage sur le serveur.

Conditions préalables

- Vous disposez du nom d'hôte et de l'adresse IP d'une cible iSCSI :
  - Si vous vous connectez à une cible de stockage créée par le logiciel externe, recherchez le nom d'hôte et l'adresse IP de la cible auprès de l'administrateur du stockage.
  - Si vous créez une cible iSCSI, voir Création d'une cible iSCSI .

Procédure

1. Installer **iscsi-initiator-utils** sur la machine du client :

```
# dnf install iscsi-initiator-utils
```

2. Vérifier le nom de l'initiateur :

```
# cat /etc/iscsi/initiatorname.iscsi

InitiatorName=iqn.2006-04.com.example:888
```

3. Si l'ACL a reçu un nom personnalisé dans Création d'une ACL iSCI, mettez à jour le nom de l'initiateur pour qu'il corresponde à l'ACL :
   a. Ouvrez le fichier **/etc/iscsi/initiatorname.iscsi** et modifiez le nom de l'initiateur :

```
# vi /etc/iscsi/initiatorname.iscsi

InitiatorName=custom-name
```

   b. Redémarrez le service **iscsid**:

```
# systemctl restart iscsid
```

4. Découvrez la cible et connectez-vous à la cible avec l'IQN affiché :

```
# iscsiadm -m discovery -t st -p 10.64.24.179
```

```
    10.64.24.179:3260,1 iqn.2006-04.example:444

# iscsiadm -m node -T iqn.2006-04.example:444 -l
   Logging in to [iface: default, target: iqn.2006-04.example:444, portal: 10.64.24.179,3260]
(multiple)
   Login to [iface: default, target: iqn.2006-04.example:444, portal: 10.64.24.179,3260]
successful.
```

Remplacez *10.64.24.179* par l'adresse IP cible.

Vous pouvez utiliser cette procédure pour un nombre quelconque d'initiateurs connectés à la même cible si leurs noms d'initiateurs respectifs sont ajoutés à l'ACL comme décrit dans la section Création d'une ACL iSCSI .

5. Recherchez le nom du disque iSCSI et créez un système de fichiers sur ce disque iSCSI :

```
# grep "Attached SCSI" /var/log/messages

# mkfs.ext4 /dev/disk_name
```

Remplacez *disk_name* par le nom du disque iSCSI affiché dans le fichier **/var/log/messages**.

6. Monter le système de fichiers :

```
# mkdir /mount/point

# mount /dev/disk_name /mount/point
```

Remplacez */mount/point* par le point de montage de la partition.

7. Modifiez le fichier **/etc/fstab** pour monter le système de fichiers automatiquement au démarrage du système :

```
# vi /etc/fstab

/dev/disk_name /mount/point ext4 _netdev 0 0
```

Remplacez *disk_name* par le nom du disque iSCSI et */mount/point* par le point de montage de la partition.

Ressources supplémentaires

- **targetcli(8)** et **iscsiadm(8)** pages de manuel

7.2. MISE EN PLACE DU PROTOCOLE D'AUTHENTIFICATION CHALLENGE-HANDSHAKE POUR L'INITIATEUR

En utilisant **Challenge-Handshake Authentication Protocol (CHAP)**, les utilisateurs peuvent protéger la cible par un mot de passe. L'initiateur doit connaître ce mot de passe pour pouvoir se connecter à la cible.

Conditions préalables

- Initiateur iSCSI créé. Pour plus d'informations, voir Création d'un initiateur iSCSI .

- Définissez l'adresse **CHAP** pour la cible. Pour plus d'informations, voir Configuration du protocole d'authentification Challenge-Handshake pour la cible.

Procédure

1. Activer l'authentification CHAP dans le fichier **iscsid.conf**:

```
# vi /etc/iscsi/iscsid.conf

node.session.auth.authmethod = CHAP
```

   Par défaut, le site **node.session.auth.authmethod** est réglé sur **None**

2. Ajouter les cibles **username** et **password** dans le fichier **iscsid.conf**:

```
node.session.auth.username = redhat
node.session.auth.password = redhat_passwd
```

3. Démarrer le démon **iscsid**:

```
# systemctl start iscsid.service
```

Ressources supplémentaires

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

7.3. SURVEILLANCE D'UNE SESSION ISCSI À L'AIDE DE L'UTILITAIRE ISCSIADM

Cette procédure décrit comment surveiller la session iscsi à l'aide de l'utilitaire **iscsiadm**.

Par défaut, un service iSCSI est **lazily** démarré et le service démarre après l'exécution de la commande **iscsiadm**. Si root n'est pas sur un périphérique iSCSI ou s'il n'y a pas de nœuds marqués par **node.startup = automatic**, le service iSCSI ne démarrera pas avant l'exécution d'une commande **iscsiadm** qui nécessite le démarrage de **iscsid** ou des modules du noyau **iscsi**.

Exécutez la commande **systemctl start iscsid.service** en tant que root pour forcer le démon **iscsid** à s'exécuter et les modules du noyau iSCSI à se charger.

Procédure

1. Installer le site **iscsi-initiator-utils** sur la machine du client :

```
# dnf install iscsi-initiator-utils
```

2. Trouvez des informations sur les sessions de course à pied :

```
# iscsiadm -m session -P 3
```

   Cette commande affiche l'état de la session ou du périphérique, l'ID de session (sid), certains paramètres négociés et les périphériques SCSI accessibles via la session.

- Pour obtenir un résultat plus court, par exemple pour afficher uniquement le mappage **sid-to-node**, exécutez la commande :

```
# iscsiadm -m session -P 0
      or
# iscsiadm -m session

tcp [2] 10.15.84.19:3260,2 iqn.1992-08.com.netapp:sn.33615311
tcp [3] 10.15.85.19:3260,3 iqn.1992-08.com.netapp:sn.33615311
```

  Ces commandes impriment la liste des sessions en cours dans le format suivant : **driver [sid] target_ip:port,target_portal_group_tag proper_target_name**.

Ressources supplémentaires

- **/usr/share/doc/iscsi-initiator-utils-version/README** fichier
- **iscsiadm(8)** page de manuel

7.4. DM MULTIPATH OVERRIDES OF THE DEVICE TIMEOUT (DÉPASSEMENT DU DÉLAI D'ATTENTE DE L'APPAREIL)

L'option **recovery_tmo sysfs** contrôle le délai d'attente pour un périphérique iSCSI particulier. Les options suivantes remplacent globalement les valeurs de **recovery_tmo**:

- L'option de configuration **replacement_timeout** remplace globalement la valeur **recovery_tmo** pour tous les périphériques iSCSI.
- Pour tous les dispositifs iSCSI gérés par DM Multipath, l'option **fast_io_fail_tmo** de DM Multipath remplace globalement la valeur **recovery_tmo**.
  L'option **fast_io_fail_tmo** de DM Multipath remplace également l'option **fast_io_fail_tmo** des périphériques Fibre Channel.

L'option DM Multipath **fast_io_fail_tmo** est prioritaire sur **replacement_timeout**. Red Hat ne recommande pas l'utilisation de **replacement_timeout** pour remplacer **recovery_tmo** dans les périphériques gérés par DM Multipath car DM Multipath réinitialise toujours **recovery_tmo**, lorsque le service **multipathd** est rechargé.

CHAPITRE 8. UTILISATION DE PÉRIPHÉRIQUES FIBRE CHANNEL

Red Hat Enterprise Linux 9 fournit les pilotes Fibre Channel natifs suivants :

- **lpfc**
- **qla2xxx**
- **zfcp**

8.1. REDIMENSIONNEMENT DES UNITÉS LOGIQUES FIBRE CHANNEL

En tant qu'administrateur système, vous pouvez redimensionner les unités logiques Fibre Channel.

Procédure

1. Déterminer quels périphériques sont des chemins d'accès pour une unité logique **multipath**:

```
multipath -ll
```

2. Re-scanner les unités logiques Fibre Channel sur un système qui utilise le multipathing :

```
$ echo 1 > /sys/block/sdX/device/rescan
```

Ressources supplémentaires

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

8.2. DÉTERMINATION DU COMPORTEMENT DE PERTE DE LIEN D'UN APPAREIL UTILISANT FIBRE CHANNEL

Si un pilote implémente le callback Transport **dev_loss_tmo**, les tentatives d'accès à un périphérique via un lien seront bloquées lorsqu'un problème de transport est détecté.

Procédure

- Déterminer l'état d'un port distant :

```
$ cat /sys/class/fc_remote_port/rport-host:bus:remote-port/port_state
```

  Cette commande renvoie l'un des résultats suivants :

  - **Blocked** lorsque le port distant et les dispositifs auxquels il permet d'accéder sont bloqués.
  - **Online** si le port distant fonctionne normalement
    Si le problème n'est pas résolu dans les **dev_loss_tmo** secondes, le site **rport** et les appareils seront débloqués. Toutes les E/S en cours d'exécution sur ce périphérique ainsi que toutes les nouvelles E/S envoyées à ce périphérique échoueront.

Lorsqu'une perte de lien dépasse **dev_loss_tmo**, les périphériques **scsi_device** et **sd_N_** sont supprimés. Généralement, la classe Fibre Channel laisse le périphérique en l'état, c'est-à-dire qu'il

/dev/*sdx* reste **/dev/*sdx***. En effet, la liaison cible est sauvegardée par le pilote Fibre Channel et lorsque le port cible revient, les adresses SCSI sont recréées fidèlement. Cependant, cela ne peut pas être garanti, le ***sdx*** ne sera restauré que si aucune modification supplémentaire n'est apportée à la configuration des LUN dans la boîte de stockage.

Ressources supplémentaires

- **multipath.conf(5)** page de manuel
- Réglage recommandé au niveau du scsi, du multipath et de la couche application lors de la configuration d'un cluster Oracle RAC Article de la base de connaissances

8.3. FICHIERS DE CONFIGURATION FIBRE CHANNEL

Voici la liste des fichiers de configuration du répertoire **/sys/class/** qui fournissent l'API de l'espace utilisateur à Fibre Channel.

Les items utilisent les variables suivantes :

H
: Numéro d'hôte

B
: Numéro de bus

T
: Cible

L
: Unité logique (LUN)

R
: Numéro de port distant

IMPORTANT

Consultez votre fournisseur de matériel avant de modifier l'une des valeurs décrites dans cette section, si votre système utilise un logiciel multipath.

Configuration du transport en /sys/class/fc_transport/target*H*:*B*:*T*/

port_id
: iD/adresse du port 24 bits

node_name
: nom du nœud 64 bits

port_name
: nom du port 64 bits

Configuration des ports à distance dans/sys/class/fc_remote_ports/rport-*H*:*B*-*R*/

- **port_id**
- **node_name**

- **port_name**

- **dev_loss_tmo**
  Contrôle le moment où le périphérique scsi est retiré du système. Après le déclenchement de **dev_loss_tmo**, le périphérique scsi est supprimé. Dans le fichier **multipath.conf**, vous pouvez définir **dev_loss_tmo** comme **infinity**.

  Dans Red Hat Enterprise Linux 9, si vous ne définissez pas l'option **fast_io_fail_tmo**, **dev_loss_tmo** est plafonné à **600** secondes. Par défaut, **fast_io_fail_tmo** est défini sur **5** secondes dans Red Hat Enterprise Linux 9 si le service **multipathd** est en cours d'exécution ; sinon, il est défini sur **off**.

- **fast_io_fail_tmo**
  Spécifie le nombre de secondes d'attente avant qu'un lien ne soit marqué comme "mauvais". Une fois qu'un lien est marqué comme mauvais, les E/S en cours d'exécution ou toute nouvelle E/S sur le chemin correspondant échouent.

  Si une E/S se trouve dans une file d'attente bloquée, elle n'échouera pas tant que **dev_loss_tmo** n'aura pas expiré et que la file d'attente n'aura pas été débloquée.

  Si **fast_io_fail_tmo** est réglé sur n'importe quelle valeur sauf off, **dev_loss_tmo** n'est pas plafonné. Si **fast_io_fail_tmo** est réglé sur off, aucune E/S n'échoue jusqu'à ce que le périphérique soit retiré du système. Si **fast_io_fail_tmo** a pour valeur un nombre, les E/S échouent immédiatement lorsque le délai d'attente de **fast_io_fail_tmo** se déclenche.

Configuration de l'hôte en/sys/class/fc_host/host*H*/

- **port_id**

- **node_name**

- **port_name**

- **issue_lip**
  Demande au pilote de redécouvrir les ports distants.

8.4. DM MULTIPATH OVERRIDES OF THE DEVICE TIMEOUT (DÉPASSEMENT DU DÉLAI D'ATTENTE DE L'APPAREIL)

L'option **recovery_tmo sysfs** contrôle le délai d'attente pour un périphérique iSCSI particulier. Les options suivantes remplacent globalement les valeurs de **recovery_tmo**:

- L'option de configuration **replacement_timeout** remplace globalement la valeur **recovery_tmo** pour tous les périphériques iSCSI.

- Pour tous les dispositifs iSCSI gérés par DM Multipath, l'option **fast_io_fail_tmo** de DM Multipath remplace globalement la valeur **recovery_tmo**.
  L'option **fast_io_fail_tmo** de DM Multipath remplace également l'option **fast_io_fail_tmo** des périphériques Fibre Channel.

L'option DM Multipath **fast_io_fail_tmo** est prioritaire sur **replacement_timeout**. Red Hat ne recommande pas l'utilisation de **replacement_timeout** pour remplacer **recovery_tmo** dans les périphériques gérés par DM Multipath car DM Multipath réinitialise toujours **recovery_tmo**, lorsque le service **multipathd** est rechargé.

CHAPITRE 9. GÉRER LES MISES À NIVEAU DU SYSTÈME À L'AIDE D'INSTANTANÉS

Effectuez des mises à niveau de systèmes Red Hat Enterprise Linux capables de revenir à une version antérieure du système d'exploitation. Vous pouvez utiliser **Boom Boot Manager** et le cadre de modernisation du système d'exploitation Leapp.

Avant de procéder à la mise à niveau du système d'exploitation, il convient de tenir compte des aspects suivants :

- Les mises à niveau du système avec des instantanés ne fonctionnent pas sur plusieurs systèmes de fichiers dans l'arborescence des systèmes, par exemple, une partition /**var** ou /**usr** distincte.
- Les mises à niveau de systèmes à l'aide d'instantanés ne fonctionnent pas pour les systèmes de l'infrastructure de mise à jour de Red Hat (RHUI). Au lieu d'utiliser l'utilitaire Boom, envisagez de créer des instantanés de vos machines virtuelles (VM).

9.1. APERÇU DU PROCESSUS DE BOOM

Créez des entrées de démarrage à l'aide du gestionnaire de démarrage Boom afin de pouvoir sélectionner et accéder à ces entrées à partir du menu du chargeur de démarrage GRUB. La création d'entrées de démarrage simplifie le processus de préparation d'une mise à niveau capable de revenir en arrière.

Les entrées d'amorçage suivantes font partie des processus de mise à niveau et de retour en arrière :

- **Upgrade boot entry**
  Démarrage de l'environnement de mise à niveau de Leapp. Utilisez l'utilitaire **leapp** pour créer et gérer cette entrée de démarrage. Le processus de mise à niveau **leapp** supprime automatiquement cette entrée.
- **Red Hat Enterprise Linux 9 boot entry**
  Démarrage de l'environnement du système de mise à niveau. Utilisez l'utilitaire **leapp** pour créer cette entrée de démarrage après une mise à niveau réussie.
- **Snapshot boot entry**
  Boots l'instantané du système d'origine. Utilisez-la pour revoir et tester l'état précédent du système d'exploitation, après une tentative de mise à niveau réussie ou non. Avant de mettre à niveau le système d'exploitation, utilisez la commande **boom** pour créer cette entrée de démarrage.
- **Rollback boot entry**
  Permet de démarrer l'environnement système d'origine et de revenir à l'état précédent du système en cas de mise à niveau. Utilisez la commande **boom** pour créer cette entrée de démarrage lors du lancement d'une procédure de retour en arrière de la mise à niveau.

Ressources supplémentaires

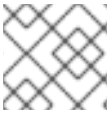
- **boom(1)** page de manuel

9.2. MISE À NIVEAU VERS UNE AUTRE VERSION À L'AIDE DE BOOM BOOT MANAGER

Effectuez une mise à niveau de votre système d'exploitation Red Hat Enterprise Linux à l'aide du gestionnaire de démarrage Boom.

Conditions préalables

- Vous utilisez une version actuelle de Red Hat Enterprise Linux.
- Vous avez installé la version actuelle du paquetage **boom-boot** (version **boom-1.3-3.el9**, idéalement **boom-1.4-4.el9** ou ultérieure).
- Vous disposez d'un espace suffisant pour l'instantané. Faites une estimation de la taille en vous basant sur la taille de l'installation d'origine. Dressez la liste de tous les volumes logiques montés.
- Vous avez installé le paquetage **leapp**.
- Vous avez activé les dépôts de logiciels.
- Vous avez activé le volume d'instantanés. S'il n'est pas actif, la commande **boom** échoue.



NOTE

D'autres entrées d'amorçage peuvent inclure **/usr** ou **/var**.

Procédure

1. Créez un instantané de votre volume logique *root*:
   - Si votre système de fichiers racine utilise le provisionnement fin, créez un instantané fin :

     ```
     # lvcreate -s rhel/root -kn -n root_snapshot_before_changes
     ```

     Ici :

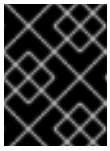
     - **-s** crée l'instantané.
     - **rhel/root** copie le système de fichiers sur le volume logique.
     - **-n** *root_snapshot_before_changes* indique le nom de l'instantané.
       Lors de la création d'un instantané fin, ne définissez pas la taille de l'instantané. L'instantané est alloué à partir du thin pool.
   - Si votre système de fichiers racine utilise le provisionnement épais, créez un instantané épais :

     ```
     # lvcreate -s  rhel/root -n root_snapshot_before_changes -L 25g
     ```

     Ici :

     - **-s** crée l'instantané.
     - **rhel/root** copie le système de fichiers sur le volume logique.
     - **-n** *root_snapshot_before_changes* indique le nom de l'instantané.
     - **-L** *25g* est la taille de l'instantané. Faites une estimation de la taille en vous basant sur la taille de l'installation d'origine.

Lors de la création d'un instantané épais, définissez la taille de l'instantané qui peut contenir toutes les modifications pendant la mise à niveau.



IMPORTANT

L'instantané créé n'inclut aucune modification supplémentaire du système.

2. Créer le profil :

```
# boom profile create --from-host --uname-pattern el9
```

3. Créer une entrée de démarrage instantanée du système d'origine à l'aide de copies de sauvegarde des images de démarrage d'origine :

```
# boom create --backup --title "Root LV snapshot before changes" --rootlv
rhel/root_snapshot_before_changes
```

Ici :

- **--title***Root LV snapshot before changes* est le nom de l'entrée d'amorçage qui apparaît dans la liste des entrées d'amorçage lors du démarrage du système.
- **--rootlv** est le volume logique racine qui correspond à la nouvelle entrée de démarrage.
- Après avoir effectué l'étape précédente, vous disposez d'une entrée de démarrage qui permet d'accéder au système d'origine, avant la mise à niveau.

4. Mettez à niveau vers Red Hat Enterprise Linux 9 à l'aide de l'utilitaire Leapp :

```
# leapp upgrade
```

- Examiner et résoudre les éventuels blocages indiqués dans le rapport de commandement **leapp upgrade**.

5. Redémarrez avec l'entrée de démarrage mise à jour :

```
# leapp upgrade --reboot
```

- Sélectionnez l'entrée **Red Hat Enterprise Linux Upgrade Initramfs** dans l'écran de démarrage de GRUB.
- L'utilitaire **leapp** crée l'entrée de démarrage de la mise à niveau. Exécutez la commande mentionnée ci-dessus pour redémarrer dans l'entrée de démarrage de mise à niveau, et procédez à l'exécution de la mise à niveau en place vers Red Hat Enterprise Linux 9. Après le processus de mise à niveau, l'argument reboot (redémarrer) initie un redémarrage automatique du système. L'écran GRUB s'affiche pendant le redémarrage.

NOTE

Le sous-menu Snapshots de l'écran de démarrage GRUB n'est pas disponible dans Red Hat Enterprise Linux 9.

Verification steps

- Poursuivez la mise à niveau et installez les nouveaux paquetages RPM de Red Hat Enterprise Linux 9. Une fois la mise à niveau terminée, le système redémarre automatiquement. L'écran GRUB affiche la version mise à niveau et l'ancienne version du système d'exploitation disponible. La version mise à niveau du système est la sélection par défaut.
- Vérifiez si l'entrée **Root LV snapshot before changes** boot se trouve dans le menu GRUB. Si elle est présente, elle permet d'accéder instantanément à l'état du système d'exploitation avant la mise à niveau.

Ressources supplémentaires

- **boom(1)** page de manuel
- Qu'est-ce que BOOM et comment l'installer ?
- Comment créer une entrée de démarrage BOOM

9.3. PASSER D'UNE VERSION DE RED HAT ENTERPRISE LINUX À UNE AUTRE

Accédez simultanément aux versions actuelles et précédentes de Red Hat Enterprise Linux sur votre machine. L'utilisation du site **Boom Boot Manager** pour accéder à différentes versions du système d'exploitation réduit le risque associé à la mise à niveau d'un système d'exploitation et contribue également à réduire les temps d'arrêt du matériel. Grâce à cette possibilité de passer d'un environnement à l'autre, vous pouvez :

- comparer rapidement les deux environnements côte à côte.
- passer d'un environnement à l'autre avec un minimum de frais généraux.
- récupérer le contenu plus ancien du système de fichiers.
- continuer à accéder à l'ancien système, alors que l'hôte mis à niveau est en cours d'exécution.
- d'interrompre et d'inverser le processus de mise à jour à tout moment, même lorsque la mise à jour elle-même est en cours.

Conditions préalables

- Vous utilisez une version actuelle de Red Hat Enterprise Linux.

Procédure

1. Redémarrer le système :

```
# reboot
```

2. Sélectionnez l'entrée de démarrage requise dans l'écran du chargeur de démarrage GRUB.

Verification steps

- Vérifiez que le volume de démarrage sélectionné est affiché :

```
# cat /proc/cmdline
```

```
root=/dev/rhel/root_snapshot_before_changes ro
rd.lvm.lv=rhel/root_snapshot_before_changes rd.lvm.lv=vg_root/swap rhgb quiet
```

Ressources supplémentaires

- **boom(1)** page de manuel
- Mise à niveau vers une autre version à l'aide de Boom Boot Manager

9.4. SUPPRESSION DE L'INSTANTANÉ DU VOLUME LOGIQUE

La création d'un instantané de votre système d'exploitation actuel vous permet d'accéder à l'état précédent du système d'exploitation, de le réviser et de le tester. Une fois que vous avez fini de travailler avec l'instantané du système d'exploitation, vous pouvez le supprimer pour libérer de l'espace de stockage.

IMPORTANT

Vous ne pouvez pas effectuer d'autres opérations avec l'instantané de volume logique (LV) après l'avoir supprimé.

Conditions préalables

- Vous utilisez une version actuelle de Red Hat Enterprise Linux.

Procédure

1. Démarrez Red Hat Enterprise Linux 9 à partir de l'entrée GRUB. La sortie suivante confirme que le nouvel instantané est sélectionné :

```
# boom list
BootID    Version                 Name                          RootDevice
6d2ec72 3.10.0-957.21.3.el8.x86_64 Red Hat Enterprise Linux Server
/dev/rhel/root_snapshot_before_changes
```

2. Supprimer l'entrée de l'instantané à l'aide de la valeur **BootID**:

```
# boom delete --boot-id 6d2ec72
```

   - Cette opération supprime l'entrée de démarrage du menu GRUB.

3. Supprimer l'instantané LV :

```
# lvremove rhel/root_snapshot_before_changes
Do you really want to remove active logical volume rhel/root_snapshot_before_changes?
[y/n]: y
    Logical volume "root_snapshot_before_changes" successfully removed
```

Ressources supplémentaires

- **boom(1)** page de manuel

- Mise à niveau vers une autre version à l'aide de Boom Boot Manager

9.5. CRÉATION D'UNE ENTRÉE D'AMORÇAGE DE RETOUR EN ARRIÈRE

Utilisez l'entrée de démarrage "rollback" pour accéder à l'environnement du système d'exploitation tel qu'il était avant la mise à niveau. En outre, vous pouvez annuler toute mise à niveau du système d'exploitation.

Préparez l'entrée de démarrage du rollback soit à partir du système mis à niveau, soit à partir de l'environnement snapshot.

Conditions préalables

- Vous utilisez une version actuelle de Red Hat Enterprise Linux.

Procédure

1. Fusionner l'instantané avec le volume d'origine (le point d'origine) :

```
# lvconvert --merge rhel/root_snapshot_before_changes
```

AVERTISSEMENT

Après avoir fusionné l'instantané, vous devez poursuivre toutes les étapes restantes de cette procédure afin d'éviter toute perte de données.

2. Créer une entrée de démarrage de retour en arrière pour l'instantané fusionné :
   - Pour **boom-0.9**:

```
boom create --title "RHEL Rollback" --rootlv rhel/root
```

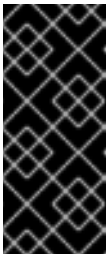
   - Pour **boom-1.2**, ou les versions ultérieures :

```
boom create --backup --title "RHEL Rollback" --rootlv rhel/root
```

3. Facultatif : Redémarrez votre machine pour rétablir l'état du système d'exploitation :

```
# reboot
```

   - Une fois le système redémarré, sélectionnez l'entrée de démarrage Red Hat Enterprise Linux Rollback dans l'écran GRUB.
   - Une fois que le volume logique **root** est actif, le système lance automatiquement l'opération de fusion des instantanés.



IMPORTANT

Une fois que l'opération de fusion commence, le volume snapshot n'est plus disponible. Après avoir démarré avec succès l'entrée de démarrage de Red Hat Enterprise Linux Rollback, le site **Root LV snapshot boot entry**ne fonctionne plus. La fusion du volume logique snapshot détruit le snapshot Root LV et restaure l'état antérieur du volume original.

4. Facultatif : Une fois l'opération de fusion terminée, supprimez les entrées inutilisées et restaurez l'entrée de démarrage d'origine :

   a. Supprimez les entrées de démarrage inutilisées de Red Hat Enterprise Linux 9 du système de fichiers **/boot** et reconstruisez le fichier **grub.cfg** pour que les modifications soient prises en compte :

   ```
   # grub2-mkconfig -o /boot/grub2/grub.cfg
   ```

   b. Restaurez l'entrée de démarrage originale de Red Hat Enterprise Linux :

   ```
   # new-kernel-pkg --update $(uname -r)
   ```

5. Après un retour au système réussi, supprimez l'entrée de démarrage **boom**:

   ```
   # boom list
   # boom delete boot-id
   ```

Ressources supplémentaires

- **boom(1)** page de manuel
- Mise à niveau vers une autre version à l'aide de Boom Boot Manager

CHAPITRE 10. CONFIGURATION DE NVME OVER FABRICS À L'AIDE DE NVME/RDMA

Dans une configuration Non-volatile Memory Express™ (NVMe™) over RDMA (NVMe™/RDMA), vous configurez un contrôleur NVMe et un initiateur NVMe.

En tant qu'administrateur système, effectuez les tâches suivantes pour déployer la configuration NVMe/RDMA :

- Configuration d'un contrôleur NVMe/RDMA à l'aide de configfs
- Configuration du contrôleur NVMe/RDMA à l'aide de nvmetcli
- Configuration d'un hôte NVMe/RDMA

10.1. APERÇU DES DISPOSITIFS NVME OVER FABRIC

Non-volatile Memory Express™ (NVMe™) est une interface qui permet à l'utilitaire du logiciel hôte de communiquer avec les disques d'état solide.

Utilisez les types de transport de tissu suivants pour configurer NVMe sur des périphériques de tissu :

NVMe sur accès direct à la mémoire à distance (NVMe/RDMA)

Pour plus d'informations sur la configuration de NVMe™/RDMA, voir Configuration de NVMe over fabrics à l'aide de NVMe/RDMA.

NVMe sur Fibre Channel (NVMe/FC)

Pour plus d'informations sur la configuration de NVMe™/FC, voir Configuration de NVMe over fabrics à l'aide de NVMe/FC.

NVMe sur TCP (NVMe/TCP)

Pour plus d'informations sur la configuration de NVMe/FC, voir Configuration de NVMe over fabrics à l'aide de NVMe/TCP.

Lors de l'utilisation de NVMe over fabrics, le lecteur à semi-conducteurs n'a pas besoin d'être local dans votre système ; il peut être configuré à distance par le biais d'un périphérique NVMe over fabrics.

10.2. CONFIGURATION D'UN CONTRÔLEUR NVME/RDMA À L'AIDE DE CONFIGFS

Utilisez cette procédure pour configurer un contrôleur Non-volatile Memory Express™ (NVMe™) over RDMA (NVMe™/RDMA) à l'aide de **configfs**.

Conditions préalables

- Vérifiez que vous disposez d'un périphérique de bloc à affecter au sous-système **nvmet**.

Procédure

1. Créer le sous-système **nvmet-rdma**:

```
# modprobe nvmet-rdma
```

```
# mkdir /sys/kernel/config/nvmet/subsystems/testnqn

# cd /sys/kernel/config/nvmet/subsystems/testnqn
```

Remplacez *testnqn* par le nom du sous-système.

2. Permet à tout hôte de se connecter à ce contrôleur :

```
# echo 1 > attr_allow_any_host
```

3. Configurer un espace de noms :

```
# mkdir namespaces/10

# cd namespaces/10
```

Remplacer *10* par le numéro de l'espace de noms

4. Définir un chemin d'accès au périphérique NVMe :

```
# echo -n /dev/nvme0n1 > device_path
```

5. Activer l'espace de noms :

```
# echo 1 > enable
```

6. Créer un répertoire avec un port NVMe :

```
# mkdir /sys/kernel/config/nvmet/ports/1

# cd /sys/kernel/config/nvmet/ports/1
```

7. Affichez l'adresse IP de *mlx5_ib0*:

```
# ip addr show mlx5_ib0

8: mlx5_ib0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 4092 qdisc mq state UP
group default qlen 256
    link/infiniband 00:00:06:2f:fe:80:00:00:00:00:00:00:e4:1d:2d:03:00:e7:0f:f6 brd
00:ff:ff:ff:ff:12:40:1b:ff:ff:00:00:00:00:00:00:ff:ff:ff:ff
    inet 172.31.0.202/24 brd 172.31.0.255 scope global noprefixroute mlx5_ib0
       valid_lft forever preferred_lft forever
    inet6 fe80::e61d:2d03:e7:ff6/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

8. Définir l'adresse de transport du contrôleur :

```
# echo -n 172.31.0.202 > addr_traddr
```

9. Définir RDMA comme type de transport :

```
# echo rdma > addr_trtype

# echo 4420 > addr_trsvcid
```

10. Définir la famille d'adresses pour le port :

```
# echo ipv4 > addr_adrfam
```

11. Créer un lien souple :

```
# ln -s /sys/kernel/config/nvmet/subsystems/testnqn
/sys/kernel/config/nvmet/ports/1/subsystems/testnqn
```

Vérification

- Vérifier que le contrôleur NVMe est à l'écoute sur le port donné et prêt à recevoir des demandes de connexion :

```
# dmesg | grep "enabling port"
[ 1091.413648] nvmet_rdma: enabling port 1 (172.31.0.202:4420)
```

Ressources supplémentaires

- **nvme(1)** page de manuel

10.3. CONFIGURATION DU CONTRÔLEUR NVME/RDMA À L'AIDE DE NVMETCLI

Utilisez l'utilitaire **nvmetcli** pour modifier, afficher et démarrer un contrôleur Non-volatile Memory Express™ (NVMe™). L'utilitaire **nvmetcli** fournit une ligne de commande et une option shell interactive. Utilisez cette procédure pour configurer le contrôleur NVMe™/RDMA en vous rendant sur le site **nvmetcli**.

Conditions préalables

- Vérifiez que vous disposez d'un périphérique de bloc à affecter au sous-système **nvmet**.
- Exécutez les opérations **nvmetcli** suivantes en tant qu'utilisateur root.

Procédure

1. Installez le paquetage **nvmetcli**:

```
# dnf install nvmetcli
```

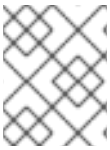
2. Téléchargez le fichier **rdma.json**:

```
# wget
http://git.infradead.org/users/hch/nvmetcli.git/blob_plain/0a6b088db2dc2e5de11e6f23f1e890e4
b54fee64:/rdma.json
```

3. Modifiez le fichier **rdma.json** et remplacez la valeur **traddr** par **172.31.0.202**.

4. Configurez le contrôleur en chargeant le fichier de configuration du contrôleur NVMe :

```
# nvmetcli restore rdma.json
```



NOTE

Si le nom du fichier de configuration du contrôleur NVMe n'est pas spécifié, le site **nvmetcli** utilise le fichier **/etc/nvmet/config.json**.

Vérification

- Vérifier que le contrôleur NVMe est à l'écoute sur le port donné et prêt à recevoir des demandes de connexion :

```
# dmesg | tail -1
[ 4797.132647] nvmet_rdma: enabling port 2 (172.31.0.202:4420)
```

- Facultatif : Effacer le contrôleur NVMe actuel :

```
# nvmetcli clear
```

Ressources supplémentaires

- **nvmetcli** et **nvme(1)** pages de manuel

10.4. CONFIGURATION D'UN HÔTE NVME/RDMA

Utilisez cette procédure pour configurer un hôte Non-volatile Memory Express™ (NVMe™) over RDMA (NVMe™/RDMA) à l'aide de l'outil NVMe management command line interface (**nvme-cli**).

Procédure

1. Installer l'outil **nvme-cli**:

```
# dnf install nvme-cli
```

2. Charger le module **nvme-rdma** s'il n'est pas chargé :

```
# modprobe nvme-rdma
```

3. Découvrez les sous-systèmes disponibles sur le contrôleur NVMe :

```
# nvme discover -t rdma -a 172.31.0.202 -s 4420

Discovery Log Number of Records 1, Generation counter 2
=====Discovery Log Entry 0======
trtype:  rdma
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified, sq flow control disable supported
portid:  1
trsvcid: 4420
```

```
subnqn:  testnqn
traddr:  172.31.0.202
rdma_prtype: not specified
rdma_qptype: connected
rdma_cms:    rdma-cm
rdma_pkey: 0x0000
```

4. Se connecter aux sous-systèmes découverts :

```
# nvme connect -t rdma -n testnqn -a 172.31.0.202 -s 4420

# lsblk
NAME                      MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda                         8:0    0 465.8G  0 disk
├─sda1                      8:1    0     1G  0 part /boot
└─sda2                      8:2    0 464.8G  0 part
  ├─rhel_rdma--virt--03-root 253:0    0    50G  0 lvm  /
  ├─rhel_rdma--virt--03-swap 253:1    0     4G  0 lvm  [SWAP]
  └─rhel_rdma--virt--03-home 253:2    0 410.8G  0 lvm  /home
nvme0n1

# cat /sys/class/nvme/nvme0/transport
rdma
```

Remplacez *testnqn* par le nom du sous-système NVMe.

Remplacez *172.31.0.202* par l'adresse IP du contrôleur.

Remplacez *4420* par le numéro de port.

Vérification

- Liste les périphériques NVMe actuellement connectés :

```
# nvme list
```

- En option : Déconnectez le contrôleur :

```
# nvme disconnect -n testnqn
NQN:testnqn disconnected 1 controller(s)

# lsblk
NAME                      MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda                         8:0    0 465.8G  0 disk
├─sda1                      8:1    0     1G  0 part /boot
└─sda2                      8:2    0 464.8G  0 part
  ├─rhel_rdma--virt--03-root 253:0    0    50G  0 lvm  /
  ├─rhel_rdma--virt--03-swap 253:1    0     4G  0 lvm  [SWAP]
  └─rhel_rdma--virt--03-home 253:2    0 410.8G  0 lvm  /home
```

Ressources supplémentaires

- **nvme(1)** page de manuel

- Dépôt Github de Nvme-cli

10.5. PROCHAINES ÉTAPES

- Activation du multipathing sur les périphériques NVMe .

CHAPITRE 11. CONFIGURATION DE NVME SUR DES TISSUS À L'AIDE DE NVME/FC

Le transport Non-volatile Memory Express™ (NVMe™) sur Fibre Channel (NVMe™/FC) est entièrement pris en charge en mode hôte lorsqu'il est utilisé avec certains adaptateurs Fibre Channel Broadcom Emulex et Marvell Qlogic. En tant qu'administrateur système, effectuez les tâches des sections suivantes pour déployer la configuration NVMe/FC :

- Configuration de l'hôte NVMe pour les adaptateurs Broadcom
- Configuration de l'hôte NVMe pour les adaptateurs QLogic

11.1. APERÇU DES DISPOSITIFS NVME OVER FABRIC

Non-volatile Memory Express™ (NVMe™) est une interface qui permet à l'utilitaire du logiciel hôte de communiquer avec les disques d'état solide.

Utilisez les types de transport de tissu suivants pour configurer NVMe sur des périphériques de tissu :

NVMe sur accès direct à la mémoire à distance (NVMe/RDMA)

Pour plus d'informations sur la configuration de NVMe™/RDMA, voir Configuration de NVMe over fabrics à l'aide de NVMe/RDMA.

NVMe sur Fibre Channel (NVMe/FC)

Pour plus d'informations sur la configuration de NVMe™/FC, voir Configuration de NVMe over fabrics à l'aide de NVMe/FC.

NVMe sur TCP (NVMe/TCP)

Pour plus d'informations sur la configuration de NVMe/FC, voir Configuration de NVMe over fabrics à l'aide de NVMe/TCP.

Lors de l'utilisation de NVMe over fabrics, le lecteur à semi-conducteurs n'a pas besoin d'être local dans votre système ; il peut être configuré à distance par le biais d'un périphérique NVMe over fabrics.

11.2. CONFIGURATION DE L'HÔTE NVME POUR LES ADAPTATEURS BROADCOM

Utilisez cette procédure pour configurer l'hôte Non-volatile Memory Express™ (NVMe™) pour le client des adaptateurs Broadcom à l'aide de l'outil d'interface de ligne de commande de gestion NVMe (**nvme-cli**).

Procédure

1. Installer l'outil **nvme-cli**:

```
# dnf install nvme-cli
```

Cette opération crée le fichier **hostnqn** dans le répertoire **/etc/nvme/**. Le fichier **hostnqn** identifie l'hôte NVMe.

2. Trouvez les identifiants WWNN et WWPN des ports locaux et distants et utilisez les résultats pour trouver le NQN du sous-système :

```
# cat /sys/class/scsi_host/host*/nvme_info

NVME Host Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x10000090fae0b5f5 WWNN x20000090fae0b5f5 DID x010f00
ONLINE
NVME RPORT       WWPN x204700a098cbcac6 WWNN x204600a098cbcac6 DID x01050e
TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 000000000e Cmpl 000000000e Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000000000008ea Issue 00000000000008ec OutIO 0000000000000002
    abort 00000000 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr 00000000 err
00000000
FCP CMPL: xb 00000000 Err 00000000
```

```
# nvme discover --transport fc \
              --traddr nn-0x204600a098cbcac6:pn-0x204700a098cbcac6 \
              --host-traddr nn-0x20000090fae0b5f5:pn-0x10000090fae0b5f5

Discovery Log Number of Records 2, Generation counter 49530
=====Discovery Log Entry 0======
trtype:  fc
adrfam:  fibre-channel
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: none
subnqn:  nqn.1992-
08.com.netapp:sn.e18bfca87d5e11e98c0800a098cbcac6:subsystem.st14_nvme_ss_1_1
traddr:  nn-0x204600a098cbcac6:pn-0x204700a098cbcac6
```

Remplacer *nn-0x204600a098cbcac6:pn-0x204700a098cbcac6* par **traddr**.

Remplacer *nn-0x20000090fae0b5f5:pn-0x10000090fae0b5f5* par **host-traddr**.

3. Connectez-vous au contrôleur NVMe à l'aide du site **nvme-cli**:

```
# nvme connect --transport fc \
              --traddr nn-0x204600a098cbcac6:pn-0x204700a098cbcac6 \
              --host-traddr nn-0x20000090fae0b5f5:pn-0x10000090fae0b5f5 \
              -n nqn.1992-
08.com.netapp:sn.e18bfca87d5e11e98c0800a098cbcac6:subsystem.st14_nvme_ss_1_1
```

Remplacer *nn-0x204600a098cbcac6:pn-0x204700a098cbcac6* par **traddr**.

Remplacer *nn-0x20000090fae0b5f5:pn-0x10000090fae0b5f5* par **host-traddr**.

Remplacer *nqn.1992-08.com.netapp:sn.e18bfca87d5e11e98c0800a098cbcac6:subsystem.st14_nvme_ss_1_1* par le ***subnqn***.

Vérification

- Liste les périphériques NVMe actuellement connectés :

```
# nvme list
Node          SN                Model                                  Namespace Usage
Format        FW Rev
----------------- -------------------- ---------------------------------------- --------- --------------------------
- ---------------- --------
/dev/nvme0n1     80BgLFM7xMJbAAAAAAAC NetApp ONTAP Controller                 1
107.37  GB / 107.37  GB      4 KiB +  0 B   FFFFFFFF
```

```
# lsblk |grep nvme
nvme0n1                 259:0    0   100G  0 disk
```

Ressources supplémentaires

- **nvme(1)** page de manuel
- Dépôt Github de Nvme-cli

11.3. CONFIGURATION DE L'HÔTE NVME POUR LES ADAPTATEURS QLOGIC

Utilisez cette procédure pour configurer l'hôte Non-volatile Memory Express™ (NVMe™) pour le client des adaptateurs Qlogic à l'aide de l'outil d'interface de ligne de commande de gestion NVMe (**nvme-cli**).

Procédure

1. Installer l'outil **nvme-cli**:

```
# dnf install nvme-cli
```

Cette opération crée le fichier **hostnqn** dans le répertoire **/etc/nvme/**. Le fichier **hostnqn** identifie l'hôte NVMe.

2. Rechargez le module **qla2xxx**:

```
# rmmod qla2xxx
# modprobe qla2xxx
```

3. Trouver les identifiants WWNN et WWPN des ports locaux et distants :

```
# dmesg |grep traddr

[    6.139862] qla2xxx [0000:04:00.0]-ffff:0: register_localport: host-traddr=nn-
0x20000024ff19bb62:pn-0x21000024ff19bb62 on portID:10700
[    6.241762] qla2xxx [0000:04:00.0]-2102:0: qla_nvme_register_remote: traddr=nn-
0x203b00a098cbcac6:pn-0x203d00a098cbcac6 PortID:01050d
```

En utilisant ces valeurs **host-traddr** et **traddr**, trouvez le sous-système NQN :

```
# nvme discover --transport fc \
                --traddr nn-0x203b00a098cbcac6:pn-0x203d00a098cbcac6 \
                --host-traddr nn-0x20000024ff19bb62:pn-0x21000024ff19bb62
```

```
Discovery Log Number of Records 2, Generation counter 49530
=====Discovery Log Entry 0======
trtype:  fc
adrfam:  fibre-channel
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: none
subnqn:  nqn.1992-
08.com.netapp:sn.c9ecc9187b1111e98c0800a098cbcac6:subsystem.vs_nvme_multipath_1_su
bsystem_468
traddr:  nn-0x203b00a098cbcac6:pn-0x203d00a098cbcac6
```

Remplacer *nn-0x203b00a098cbcac6:pn-0x203d00a098cbcac6* par **traddr**.

Remplacer *nn-0x20000024ff19bb62:pn-0x21000024ff19bb62* par **host-traddr**.

4. Connectez-vous au contrôleur NVMe à l'aide de l'outil **nvme-cli**:

```
# nvme connect  --transport fc \
            --traddr nn-0x203b00a098cbcac6:pn-0x203d00a098cbcac6 \
            --host-traddr nn-0x20000024ff19bb62:pn-0x21000024ff19bb62 \
            -n nqn.1992-
08.com.netapp:sn.c9ecc9187b1111e98c0800a098cbcac6:subsystem.vs_nvme_multipath_1_su
bsystem_468
```

Remplacer *nn-0x203b00a098cbcac6:pn-0x203d00a098cbcac6* par **traddr**.

Remplacer *nn-0x20000024ff19bb62:pn-0x21000024ff19bb62* par **host-traddr**.

Remplacer *nqn.1992-*
08.com.netapp:sn.c9ecc9187b1111e98c0800a098cbcac6:subsystem.vs_nvme_multipath_1_subsyste
par le ***subnqn***.

Vérification

- Liste les périphériques NVMe actuellement connectés :

```
# nvme list
Node          SN                   Model                                    Namespace Usage
Format           FW Rev
----------------- -------------------- ---------------------------------------- --------- ------------------------
- ---------------- --------
/dev/nvme0n1     80BgLFM7xMJbAAAAAAAC NetApp ONTAP Controller                   1
107.37  GB / 107.37  GB      4 KiB +  0 B   FFFFFFFF

# lsblk |grep nvme
nvme0n1                 259:0    0   100G  0 disk
```

Ressources supplémentaires

- **nvme(1)** page de manuel
- Dépôt Github de Nvme-cli

11.4. PROCHAINES ÉTAPES

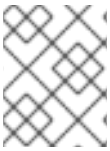
- Activation du multipathing sur les périphériques NVMe .

CHAPITRE 12. CONFIGURATION DE NVME SUR LES TISSUS À L'AIDE DE NVME/TCP

Dans une configuration Non-volatile Memory Express™ (NVMe™) over TCP (NVMe/TCP), le mode hôte est entièrement pris en charge et la configuration du contrôleur n'est pas prise en charge.

En tant qu'administrateur système, effectuez les tâches décrites dans les sections suivantes pour déployer la configuration NVMe/TCP :

- Configuration d'un hôte NVMe/TCP
- Connexion de l'hôte NVMe/TCP au contrôleur NVMe/TCP



NOTE

Dans Red Hat Enterprise Linux 9, le multipathing NVMe natif est activé par défaut. L'activation du multipathing DM n'est pas prise en charge avec NVMe/TCP.

12.1. APERÇU DES DISPOSITIFS NVME OVER FABRIC

Non-volatile Memory Express™ (NVMe™) est une interface qui permet à l'utilitaire du logiciel hôte de communiquer avec les disques d'état solide.

Utilisez les types de transport de tissu suivants pour configurer NVMe sur des périphériques de tissu :

NVMe sur accès direct à la mémoire à distance (NVMe/RDMA)

Pour plus d'informations sur la configuration de NVMe™/RDMA, voir Configuration de NVMe over fabrics à l'aide de NVMe/RDMA.

NVMe sur Fibre Channel (NVMe/FC)

Pour plus d'informations sur la configuration de NVMe™/FC, voir Configuration de NVMe over fabrics à l'aide de NVMe/FC.

NVMe sur TCP (NVMe/TCP)

Pour plus d'informations sur la configuration de NVMe/FC, voir Configuration de NVMe over fabrics à l'aide de NVMe/TCP.

Lors de l'utilisation de NVMe over fabrics, le lecteur à semi-conducteurs n'a pas besoin d'être local dans votre système ; il peut être configuré à distance par le biais d'un périphérique NVMe over fabrics.

12.2. CONFIGURATION D'UN HÔTE NVME/TCP

Utilisez l'outil d'interface de ligne de commande de gestion Non-volatile Memory Express™ (NVMe™) (nvme-cli) pour configurer un hôte NVMe/TCP.

Procédure

1. Installer l'outil **nvme-cli**:

```
# dnf install nvme-cli
```

Cet outil crée le fichier **hostnqn** dans le répertoire **/etc/nvme/**, qui identifie l'hôte NVMe.

2. Trouver le nvme **hostid** et **hostnqn**:

```
# cat /etc/nvme/hostnqn
nqn.2014-08.org.nvmexpress:uuid:8ae2b12c-3d28-4458-83e3-658e571ed4b8

# cat /etc/nvme/hostid
09e2ce17-ccc9-412d-8dcf-2b0a1d581ee3
```

   Utilisez les valeurs **hostid** et **hostnqn** pour configurer le contrôleur NVMe/TCP.

3. Vérifier l'état du contrôleur :

```
# nmcli device show ens6
GENERAL.DEVICE:                         ens6
GENERAL.TYPE:                           ethernet
GENERAL.HWADDR:                         52:57:02:12:02:02
GENERAL.MTU:                            1500
GENERAL.STATE:                          30 (disconnected)
GENERAL.CONNECTION:                     --
GENERAL.CON-PATH:                       --
WIRED-PROPERTIES.CARRIER:               on
```

4. Configurer le réseau hôte pour un contrôleur Ethernet nouvellement installé avec une adresse IP statique :

```
# nmcli connection add con-name ens6 ifname ens6 type ethernet ip4 192.168.101.154/24
gw4 192.168.101.1
```

   Remplacez ici *192.168.101.154* par l'adresse IP de l'hôte.

```
# nmcli connection mod ens6 ipv4.method manual
# nmcli connection up ens6
```

   Étant donné qu'un nouveau réseau est créé pour connecter l'hôte NVMe/TCP au contrôleur NVMe/TCP, exécutez également cette étape sur le contrôleur.

Vérification

- Vérifiez que le réseau hôte nouvellement créé fonctionne correctement :

```
# nmcli device show ens6
GENERAL.DEVICE:                         ens6
GENERAL.TYPE:                           ethernet
GENERAL.HWADDR:                         52:57:02:12:02:02
GENERAL.MTU:                            1500
GENERAL.STATE:                          100 (connected)
GENERAL.CONNECTION:                     ens6
GENERAL.CON-PATH:                       /org/freedesktop/NetworkManager/ActiveConnection/5
WIRED-PROPERTIES.CARRIER:               on
IP4.ADDRESS[1]:                         192.168.101.154/24
IP4.GATEWAY:                            192.168.101.1
IP4.ROUTE[1]:                           dst = 192.168.101.0/24, nh = 0.0.0.0, mt = 101
IP4.ROUTE[2]:                           dst = 192.168.1.1/32, nh = 0.0.0.0, mt = 101
IP4.ROUTE[3]:                           dst = 0.0.0.0/0, nh = 192.168.1.1, mt = 101
```

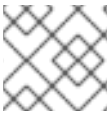
```
IP6.ADDRESS[1]:                         fe80::27ce:dde1:620:996c/64
IP6.GATEWAY:                            --
IP6.ROUTE[1]:                           dst = fe80::/64, nh = ::, mt = 101
```

Ressources supplémentaires

- La page de manuel **nvme(1)**

12.3. CONNEXION DE L'HÔTE NVME/TCP AU CONTRÔLEUR NVME/TCP

Connectez l'hôte NVMe™ over TCP (NVMe/TCP) au système contrôleur NVMe/TCP pour vérifier que l'hôte NVMe/TCP peut désormais accéder à l'espace de noms.



NOTE

Le module de contrôle NVMe/TCP (**nvmet_tcp**) n'est pas pris en charge.

Conditions préalables

- Vous avez configuré un hôte NVMe/TCP. Pour plus d'informations, voir Configuration d'un hôte NVMe/TCP.
- Vous avez configuré un contrôleur NVMe/TCP à l'aide d'un logiciel de stockage externe et le réseau est configuré sur le contrôleur. Dans cette procédure, *192.168.101.55* est l'adresse IP du contrôleur NVMe/TCP.

Procédure

1. Charger le module **nvme_tcp** si ce n'est pas déjà fait :

```
# modprobe nvme_tcp
```

2. Découvrez les sous-systèmes disponibles sur le contrôleur NVMe :

```
# nvme discover --transport=tcp --traddr=192.168.101.55 --host-traddr=192.168.101.154 --
trsvcid=8009

Discovery Log Number of Records 2, Generation counter 7
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq: not specified, sq flow control disable supported
portid:  2
trsvcid: 8009
subnqn:  nqn.2014-08.org.nvmexpress.discovery
traddr:  192.168.101.55
eflags:  not specified
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
```

```
subtype: nvme subsystem
treq: not specified, sq flow control disable supported
portid:  2
trsvcid: 8009
subnqn:  nqn.2014-08.org.nvmexpress:uuid:0c468c4d-a385-47e0-8299-6e95051277db
traddr:  192.168.101.55
eflags:  not specified
sectype: none
```

Ici, *192.168.101.55* est l'adresse IP du contrôleur NVMe/TCP et *192.168.101.154* est l'adresse IP de l'hôte NVMe/TCP.

3. Configurer le fichier **/etc/nvme/discovery.conf** pour ajouter les paramètres utilisés dans la commande **nvme discover**:

```
# echo "--transport=tcp --traddr=192.168.101.55 --host-traddr=192.168.101.154 --
trsvcid=8009" >> /etc/nvme/discovery.conf
```

4. Connectez l'hôte NVMe/TCP au système contrôleur :

```
# nvme connect-all
```

Vérification

- Vérifiez que l'hôte NVMe/TCP peut accéder à l'espace de noms :

```
# nvme list-subsys

nvme-subsys3 - NQN=nqn.2014-08.org.nvmexpress:uuid:0c468c4d-a385-47e0-8299-
6e95051277db
\
 +- nvme3 tcp traddr=192.168.101.55,trsvcid=8009,host_traddr=192.168.101.154 live
optimized

# nvme list
Node             Generic          SN               Model                                    Namespace Usage
Format       FW Rev
---------------------- ---------------------- -------------------- ---------------------------------------- --------
- -------------------------- ---------------- --------
/dev/nvme3n1     /dev/ng3n1       d93a63d394d043ab4b74 Linux                                    1
21.47  GB /  21.47  GB    512   B +  0 B   5.18.5-2
```

Ressources supplémentaires

- La page de manuel **nvme(1)**

CHAPITRE 13. ACTIVATION DU MULTIPATHING SUR LES PÉRIPHÉRIQUES NVME

Vous pouvez multipather les périphériques Non-volatile Memory Express™ (NVMe™) qui sont connectés à votre système sur un transport fabric, tel que Fibre Channel (FC). Vous pouvez choisir entre plusieurs solutions de multipathing.

13.1. MULTIPATHING NVME NATIF ET DM MULTIPATH

Les périphériques Non-volatile Memory Express™ (NVMe™) prennent en charge une fonctionnalité de multipathing native. Lors de la configuration du multipathing sur NVMe, vous pouvez choisir entre le cadre DM Multipath standard et le multipathing NVMe natif.

DM Multipath et le multipathing NVMe natif prennent tous deux en charge le schéma de multipathing ANA (Asymmetric Namespace Access) des périphériques NVMe. ANA identifie des chemins optimisés entre le contrôleur et l'hôte et améliore les performances.

Lorsque le multipathing NVMe natif est activé, il s'applique globalement à tous les périphériques NVMe. Il peut offrir de meilleures performances, mais ne contient pas toutes les fonctionnalités offertes par DM Multipath. Par exemple, le multipathing NVMe natif ne prend en charge que les méthodes de sélection de chemin **numa** et **round-robin**.

Par défaut, le multipathing NVMe est activé dans Red Hat Enterprise Linux 9 et constitue la solution de multipathing recommandée.

13.2. ACTIVATION DE DM MULTIPATH SUR LES PÉRIPHÉRIQUES NVME

Le paramètre par défaut du noyau pour l'option **nvme_core.multipath** est défini sur **Y**, ce qui signifie que le multipathing Non-volatile Memory Express™ (NVMe™) natif est activé. Vous pouvez activer DM Multipath sur les périphériques NVMe connectés en désactivant le multipathing NVMe natif.

Conditions préalables

- Les périphériques NVMe sont connectés à votre système. Pour plus d'informations, voir Vue d'ensemble des périphériques NVMe over fabric.

Procédure

1. Vérifier si le multipathing NVMe natif est activé :

   ```
   # cat /sys/module/nvme_core/parameters/multipath
   ```

   La commande affiche l'un des éléments suivants :

   N
   : Le multipathing NVMe natif est désactivé.

   Y
   : Le multipathing NVMe natif est activé.

2. Si le multipathing NVMe natif est activé, désactivez-le en utilisant l'une des méthodes suivantes :

- Utilisation d'une option du noyau :

  a. Ajoutez l'option **nvme_core.multipath=N** à la ligne de commande :

  ```
  # grubby --update-kernel=ALL --args="nvme_core.multipath=N"
  ```

  b. Sur l'architecture IBM Z 64 bits, mettez à jour le menu de démarrage :

  ```
  # zipl
  ```

  c. Redémarrer le système.

- Utilisation d'un fichier de configuration du module du noyau :

  a. Créez le fichier de configuration **/etc/modprobe.d/nvme_core.conf** avec le contenu suivant :

  ```
  options nvme_core multipath=N
  ```

  b. Sauvegarder le fichier **initramfs**:

  ```
  # cp /boot/initramfs-$(uname -r).img /boot/initramfs-$(uname r).bak.$(date %m\r%H%M%S).img
  ```

  c. Reconstruire le site **initramfs**:

  ```
  # cp /boot/initramfs-$(uname -r).img /boot/initramfs-$(uname -r).bak.$(date +%m-%d-%H%M%S).img
  # dracut --force --verbose
  ```

  d. Redémarrer le système.

3. Activer DM Multipath :

```
# systemctl enable --now multipathd.service
```

4. Distribuer les E/S sur tous les chemins disponibles. Ajoutez le contenu suivant dans le fichier **/etc/multipath.conf**:

```
devices {
        device {
                vendor "NVME"
                product ".*"
                path_grouping_policy group_by_prio
        }
}
```

NOTE

Le fichier de configuration **/sys/class/nvme-subsystem/nvme-subsys0/iopolicy** n'a aucun effet sur la distribution des E/S lorsque DM Multipath gère les périphériques NVMe.

5. Rechargez le service **multipathd** pour appliquer les changements de configuration :

```
# multipath -r
```

Vérification

- Vérifiez si le multipathing NVMe natif est désactivé :

```
# cat /sys/module/nvme_core/parameters/multipath
N
```

- Vérifiez que DM multipath reconnaît les périphériques nvme :

```
# multipath -l

eui.00007a8962ab241100a0980000d851c8 dm-6 NVME,NetApp E-Series
size=20G features='0' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=0 status=active
  |- 0:10:2:2 nvme0n2 259:3 active undef running
`-+- policy='service-time 0' prio=0 status=enabled
  |- 4:11:2:2 nvme4n2 259:28 active undef running
`-+- policy='service-time 0' prio=0 status=enabled
  |- 5:32778:2:2 nvme5n2 259:38 active undef running
`-+- policy='service-time 0' prio=0 status=enabled
  |- 6:32779:2:2 nvme6n2 259:44 active undef running
```

Ressources supplémentaires

- Configuring kernel command-line parameters
- Configuration de DM Multipath

13.3. ACTIVATION DU MULTIPATHING NVME NATIF

Si le multipathing NVMe natif est désactivé, vous pouvez l'activer à l'aide de la solution suivante.

Conditions préalables

- Les périphériques NVMe sont connectés à votre système. Pour plus d'informations, voir Vue d'ensemble des périphériques NVMe over fabric.

Procédure

1. Vérifier si le multipathing NVMe natif est activé dans le noyau :

```
# cat /sys/module/nvme_core/parameters/multipath
```

La commande affiche l'un des éléments suivants :

N

Le multipathing NVMe natif est désactivé.

Y

Le multipathing NVMe natif est activé.

2. Si le multipathing NVMe natif est désactivé, activez-le en utilisant l'une des méthodes suivantes :

   - Utilisation d'une option du noyau :

     a. Supprime l'option **nvme_core.multipath=N** de la ligne de commande du noyau :

     ```
     # grubby --update-kernel=ALL --remove-args="nvme_core.multipath=N"
     ```

     b. Sur l'architecture IBM Z 64 bits, mettez à jour le menu de démarrage :

     ```
     # zipl
     ```

     c. Redémarrer le système.

   - Utilisation d'un fichier de configuration du module du noyau :

     a. Supprimez le fichier de configuration **/etc/modprobe.d/nvme_core.conf**:

     ```
     # rm /etc/modprobe.d/nvme_core.conf
     ```

     b. Sauvegarder le fichier **initramfs**:

     ```
     # cp /boot/initramfs-$(uname -r).img /boot/initramfs-$(uname -r).bak.$(date %m-\r%H%M%S).img
     ```

     c. Reconstruire le site **initramfs**:

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

     d. Redémarrer le système.

3. Facultatif : Sur le système en cours d'exécution, modifiez la stratégie d'E/S sur les périphériques NVMe afin de répartir les E/S sur tous les chemins disponibles :

   ```
   # echo "round-robin" > /sys/class/nvme-subsystem/nvme-subsys0/iopolicy
   ```

4. Facultatif : Définissez la politique d'E/S de manière persistante à l'aide des règles **udev**. Créez le fichier **/etc/udev/rules.d/71-nvme-io-policy.rules** avec le contenu suivant :

   ```
   ACTION=="add|change", SUBSYSTEM=="nvme-subsystem", ATTR{iopolicy}="round-robin"
   ```

Vérification

1. Vérifiez que votre système reconnaît les périphériques NVMe. L'exemple suivant suppose que vous avez un sous-système de stockage NVMe over fabrics connecté avec deux espaces de noms NVMe :

   ```
   # nvme list

   Node         SN                   Model                                    Namespace Usage
   ```

```
Format           FW Rev
---------------- ---------------------- ---------------------------------------- --------- ------------------------
- ---------------- --------
/dev/nvme0n1     a34c4f3a0d6f5cec     Linux                                    1         250.06  GB /
250.06  GB    512   B +  0 B   4.18.0-2
/dev/nvme0n2     a34c4f3a0d6f5cec     Linux                                    2         250.06  GB /
250.06  GB    512   B +  0 B   4.18.0-2
```

2. Liste de tous les sous-systèmes NVMe connectés :

```
# nvme list-subsys

nvme-subsys0 - NQN=testnqn
\
 +- nvme0 fc traddr=nn-0x20000090fadd597a:pn-0x10000090fadd597a host_traddr=nn-
0x20000090fac7e1dd:pn-0x10000090fac7e1dd live
 +- nvme1 fc traddr=nn-0x20000090fadd5979:pn-0x10000090fadd5979 host_traddr=nn-
0x20000090fac7e1dd:pn-0x10000090fac7e1dd live
 +- nvme2 fc traddr=nn-0x20000090fadd5979:pn-0x10000090fadd5979 host_traddr=nn-
0x20000090fac7e1de:pn-0x10000090fac7e1de live
 +- nvme3 fc traddr=nn-0x20000090fadd597a:pn-0x10000090fadd597a host_traddr=nn-
0x20000090fac7e1de:pn-0x10000090fac7e1de live
```

Vérifiez le type de transport actif. Par exemple, **nvme0 fc** indique que l'appareil est connecté via le transport Fibre Channel, et **nvme tcp** indique que l'appareil est connecté via TCP.

3. Si vous avez modifié les options du noyau, vérifiez si le multipathing NVMe natif est activé sur la ligne de commande du noyau :

```
# cat /proc/cmdline

BOOT_IMAGE=[...] nvme_core.multipath=Y
```

4. Si vous avez modifié la stratégie d'E/S, vérifiez que **round-robin** est la stratégie d'E/S active sur les périphériques NVMe :

```
# cat /sys/class/nvme-subsystem/nvme-subsys0/iopolicy

round-robin
```

Ressources supplémentaires

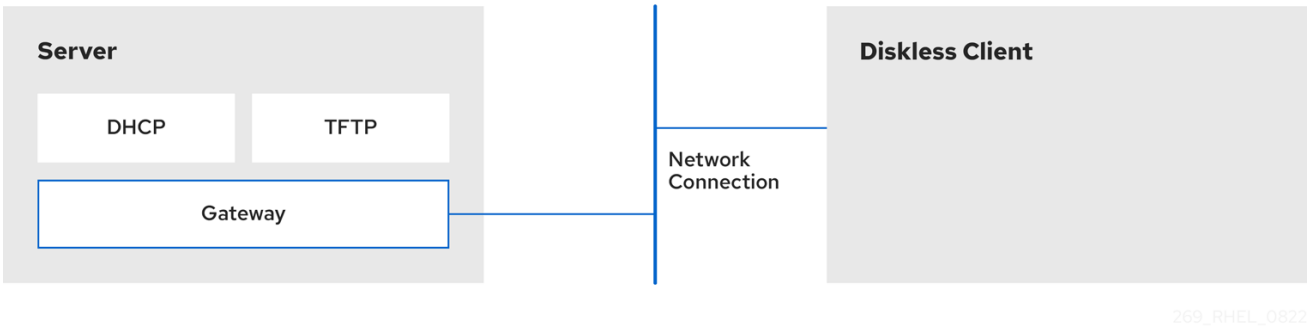
- Configuring kernel command-line parameters

CHAPITRE 14. CONFIGURATION D'UN SYSTÈME SANS DISQUE À DISTANCE

Dans un environnement réseau, vous pouvez installer plusieurs clients avec la même configuration en déployant un système distant sans disque. En utilisant la version actuelle du serveur Red Hat Enterprise Linux, vous pouvez économiser le coût des disques durs pour ces clients et configurer la passerelle sur un serveur séparé.

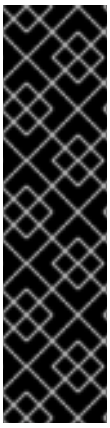
Le diagramme suivant décrit la connexion d'un client sans disque avec le serveur via les services DHCP (Dynamic Host Configuration Protocol) et TFTP (Trivial File Transfer Protocol).

Figure 14.1. Diagramme de configuration du système sans disque à distance



14.1. PRÉPARATION DES ENVIRONNEMENTS POUR LE SYSTÈME SANS DISQUE DISTANT

Préparez votre environnement pour pouvoir poursuivre la mise en œuvre du système sans disque à distance. Le démarrage du système sans disque à distance nécessite un service TFTP (Trivial File Transfer Protocol) (fourni par **tftp-server**) et un service DHCP (Dynamic Host Configuration Protocol) (fourni par **dhcp**). Le système utilise le service **tftp** pour récupérer l'image du noyau et le disque RAM initial, **initrd**, sur le réseau, par l'intermédiaire du chargeur PXE (Preboot Execution Environment).



IMPORTANT

Pour garantir le bon fonctionnement du système sans disque distant dans votre environnement, configurez les services dans l'ordre suivant :

1. **tftp** service pour les clients sans disque
2. le serveur DHCP
3. le système de fichiers en réseau (NFS)
4. le système de fichiers exporté.

Conditions préalables

- Vous avez configuré votre connexion réseau.

Procédure

1. Installez le paquetage **dracut-network**:
   -

```
# dnf install dracut-network
```

2. Ajoutez la ligne suivante au fichier **/etc/dracut.conf.d/network.conf**:

```
add_dracutmodules =" nfs "
```

14.2. CONFIGURATION D'UN SERVICE TFTP POUR LES CLIENTS SANS DISQUE

Pour que le système distant sans disque fonctionne correctement dans votre environnement, vous devez d'abord configurer un service TFTP (Trivial File Transfer Protocol) pour les clients sans disque.

NOTE

Cette configuration ne permet pas de démarrer via l'interface UEFI (Unified Extensible Firmware Interface). Pour une installation basée sur l'UEFI, voir Configuration d'un serveur TFTP pour les clients basés sur l'UEFI.

Conditions préalables

- Vous avez installé les paquets suivants :
  - **tftp-server**
  - **syslinux**

Procédure

1. Activer le service **tftp**:

```
# systemctl enable --now tftp
```

2. Créez un répertoire **pxelinux** à l'intérieur du répertoire racine **tftp**:

```
# mkdir -p /var/lib/tftpboot/pxelinux/
```

3. Copiez le fichier **/usr/share/syslinux/pxelinux.0** dans le répertoire **/var/lib/tftpboot/pxelinux/**:

```
# cp /usr/share/syslinux/pxelinux.0 /var/lib/tftpboot/pxelinux/
```

   - Vous pouvez trouver le répertoire racine tftp (**chroot**) dans le répertoire **/var/lib/tftpboot**.

4. Copier **/usr/share/syslinux/ldlinux.c32** sur **/var/lib/tftpboot/pxelinux/**:

```
# cp /usr/share/syslinux/ldlinux.c32 /var/lib/tftpboot/pxelinux/
```

5. Créez un répertoire **pxelinux.cfg** à l'intérieur du répertoire racine **tftp**:

```
# mkdir -p /var/lib/tftpboot/pxelinux/pxelinux.cfg/
```

Cette configuration ne permet pas de démarrer via l'interface UEFI (Unified Extensible Firmware Interface). Pour effectuer l'installation pour l'UEFI, suivez la procédure décrite dans la section Configuration d'un serveur TFTP pour les clients basés sur l'UEFI .

Vérification

- Vérifier l'état du service **tftp**:

```
# systemctl status tftp
...
Active: active (running)
...
```

14.3. CONFIGURATION D'UN SERVEUR DHCP POUR LES CLIENTS SANS DISQUE

Le système distant sans disque nécessite plusieurs services préinstallés pour fonctionner correctement. Tout d'abord, vous devez installer le service TFTP (Trivial File Transfer Protocol), puis configurer le serveur DHCP (Dynamic Host Configuration Protocol).

Conditions préalables

- Vous avez installé le paquet suivant :
  - **dhcp-server**
- Vous avez configuré le service **tftp** pour les clients sans disque. Voir la section Configuration d'un service TFTP pour les clients sans disque.

Procédure

1. Ajoutez la configuration au fichier **/etc/dhcp/dhcpd.conf** pour mettre en place un serveur DHCP et activer le Preboot Execution Environment (PXE) pour le démarrage :

```
option space pxelinux;
option pxelinux.magic code 208 = string;
option pxelinux.configfile code 209 = text;
option pxelinux.pathprefix code 210 = text;
option pxelinux.reboottime code 211 = unsigned integer 32;
option architecture-type code 93 = unsigned integer 16;

subnet 192.168.205.0 netmask 255.255.255.0 {
  option routers 192.168.205.1;
  range 192.168.205.10 192.168.205.25;

  class "pxeclients" {
    match if substring (option vendor-class-identifier, 0, 9) = "PXEClient";
    next-server 192.168.205.1;

    if option architecture-type = 00:07 {
      filename "BOOTX64.efi";
      } else {
      filename "pxelinux/pxelinux.0";
```

```
    }
  }
}
```

- Votre configuration DHCP peut être différente en fonction de votre environnement, comme la définition de la durée du bail ou de l'adresse fixe. Pour plus de détails, voir Fournir des services DHCP.

NOTE

Lors de l'utilisation de la machine virtuelle **libvirt** en tant que client sans disque, le démon **libvirt** fournit le service DHCP et le serveur DHCP autonome n'est pas utilisé. Dans cette situation, le démarrage en réseau doit être activé avec l'option **bootp file=<filename>** dans la configuration du réseau **libvirt**, **virsh net-edit**.

2. Activer **dhcpd.service**:

```
# systemctl enable --now dhcpd.service
```

Vérification

- Vérifier l'état du service **dhcpd.service**:

```
# systemctl status dhcpd.service
...
Active: active (running)
...
```

14.4. CONFIGURATION D'UN SYSTÈME DE FICHIERS EXPORTÉ POUR LES CLIENTS SANS DISQUE

Dans le cadre de la configuration d'un système sans disque distant dans votre environnement, vous devez configurer un système de fichiers exporté pour les clients sans disque.

Conditions préalables

- Vous avez configuré le service **tftp** pour les clients sans disque. Voir la section Configuration d'un service TFTP pour les clients sans disque.
- Vous avez configuré le serveur DHCP (Dynamic Host Configuration Protocol). Voir la section Configuration d'un serveur DHCP pour les clients sans disque .

Procédure

1. Configurez le serveur NFS (Network File System) pour exporter le répertoire racine en l'ajoutant au répertoire **/etc/exports**. Pour obtenir l'ensemble des instructions, voir Configuration du serveur NFS.
2. Installez une version complète de Red Hat Enterprise Linux dans le répertoire racine afin d'accommoder les clients sans disque. Pour ce faire, vous pouvez soit installer un nouveau système de base, soit cloner une installation existante.
   - Installez Red Hat Enterprise Linux à l'emplacement exporté en remplaçant exported-root-

- Installez Red Hat Enterprise Linux à l'emplacement exporté en remplaçant exported-root-directory par le chemin d'accès au système de fichiers exporté :

```
# dnf install @Base kernel dracut-network nfs-utils \ --
installroot=pass:quotes[exported-root-directory] --releasever=/
```

- Utilisez l'utilitaire **rsync** pour vous synchroniser avec un système en cours d'exécution :

```
# rsync -a -e ssh --exclude='/proc/' --exclude='/sys/' \ pass:quotes example.com:/
pass:quotes exported-root-directory
```

  - Remplacez *example.com* par le nom d'hôte du système en cours d'exécution avec lequel vous souhaitez vous synchroniser via l'utilitaire **rsync**.
  - Remplacez *exported-root-directory* par le chemin d'accès au système de fichiers exporté.
    Notez que pour cette option, vous devez disposer d'un système existant distinct, que vous clonerez sur le serveur à l'aide de la commande ci-dessus.

Vous devez configurer entièrement le système de fichiers, qui est prêt à être exporté, avant de pouvoir l'utiliser avec des clients sans disque. Suivez la procédure ci-dessous pour terminer la configuration.

Configuration d'un système de fichiers

1. Copiez le noyau supporté par le client sans disque (**vmlinuz-_kernel-version_pass:attributes**) dans le répertoire de démarrage **tftp**:

```
# cp /exported-root-directory/boot/vmlinuz-kernel-version /var/lib/tftpboot/pxelinux/
```

2. Créer le fichier **initramfs-*kernel-version*.img** localement et le déplacer vers le répertoire racine exporté avec le support de NFS :

```
# dracut --add nfs initramfs-kernel-version.img kernel-version
```

   Par exemple :

```
# dracut --add nfs /exports/root/boot/initramfs-5.14.0-202.el9.x86_64.img 5.14.0-
202.el9.x86_64
```

   Exemple de création d'un initrd, en utilisant la version actuelle du noyau et en écrasant l'image existante :

```
# dracut -f --add nfs \N-"boot/initramfs-$(uname -r).img\N" \N-"$(uname -r)\N"
```

3. Modifiez les permissions du fichier **initrd** en **0644**:

```
# chmod 0644 /exported-root-directory/boot/initramfs-kernel-version.img
```

AVERTISSEMENT

Si vous ne modifiez pas les permissions du fichier **initrd**, le chargeur de démarrage **pxelinux.0** échoue avec une erreur "file not found" (fichier introuvable).

4. Copiez le fichier résultant **initramfs-*kernel-version*.img** dans le répertoire de démarrage de **tftp**:

```
# cp /exported-root-directory/boot/initramfs-kernel-version.img
/var/lib/tftpboot/pxelinux/
```

5. Ajoutez la configuration suivante dans le fichier **/var/lib/tftpboot/pxelinux/pxelinux.cfg/default** pour modifier la configuration de démarrage par défaut pour l'utilisation de **initrd** et du noyau :

```
default rhel9

label rhel9
  kernel vmlinuz-kernel-version
  append initrd=initramfs-kernel-version.img root=nfs:_server-ip_:/exported-root-directory rw
```

   - Cette configuration indique à la racine du client sans disque de monter le système de fichiers exporté (**/exported-root-directory**) en lecture/écriture.

6. Facultatif : Montez le système de fichiers au format *read-only* en modifiant le fichier **/var/lib/tftpboot/pxelinux/pxelinux.cfg/default** avec la configuration suivante :

```
default rhel9

label rhel9
  kernel vmlinuz-kernel-version
  append initrd=initramfs-kernel-version.img root=nfs:server-ip:/exported-root-directory ro
```

7. Redémarrez le serveur NFS :

```
# systemctl restart nfs-server.service
```

Vous pouvez maintenant exporter le partage NFS vers des clients sans disque. Ces clients peuvent démarrer sur le réseau via l'environnement d'exécution pré-amorçage (PXE).

Ressources supplémentaires

- Configuration du serveur NFS.

14.5. RECONFIGURATION D'UN SYSTÈME DISTANT SANS DISQUE

Si vous souhaitez installer des mises à jour de paquets, redémarrer le service ou déboguer les problèmes, vous pouvez reconfigurer le système. Les étapes ci-dessous montrent comment changer le mot de passe d'un utilisateur, comment installer un logiciel sur un système, et décrivent comment diviser un

système en un site **/usr** en mode lecture seule et un site **/var** en mode lecture-écriture.

Conditions préalables

- Vous avez activé l'option **no_root_squash** dans le système de fichiers exporté.

Procédure

1. Pour modifier le mot de passe de l'utilisateur, procédez comme suit :
   - Modifiez la ligne de commande en */exported/root/directory*:

     ```
     # chroot /exported/root/directory /bin/bash
     ```

   - Modifiez le mot de passe de l'utilisateur souhaité :

     ```
     # passwd <username>
     ```

     Remplacez le *<username>* par un utilisateur réel pour lequel vous souhaitez modifier le mot de passe.

   - Quitter la ligne de commande.
2. Installer un logiciel sur un système distant sans disque :

   ```
   # dnf install <package> --installroot=/exported/root/directory --releasever=/ --config
   /etc/dnf/dnf.conf --setopt=reposdir=/etc/yum.repos.d/
   ```

   - Remplacez *<package>* par le paquet que vous souhaitez installer.
3. Configurez deux exportations distinctes pour diviser un système sans disque distant en un système **/usr** et un système **/var**. Voir la configuration du serveur NFS pour plus d'informations.

14.6. RÉSOLUTION DES PROBLÈMES COURANTS LIÉS AU CHARGEMENT D'UN SYSTÈME SANS DISQUE DISTANT

Sur la base de la configuration précédente, certains problèmes peuvent survenir lors du chargement du système sans disque distant. Voici quelques exemples des problèmes les plus courants et des moyens de les résoudre sur un serveur Red Hat Enterprise Linux.

Exemple 14.1. Le client ne reçoit pas d'adresse IP

- Vérifiez si le service DHCP (Dynamic Host Configuration Protocol) est activé sur le serveur.
  - Vérifiez si le site **dhcp.service** fonctionne :

    ```
    # systemctl status dhcpd.service
    ```

  - Si le site **dhcp.service** est inactif, vous devez l'activer et le démarrer :

    ```
    # systemctl enable dhcpd.service
    # systemctl start dhcpd.service
    ```

- Redémarrer le client sans disque.
- Vérifiez le fichier de configuration DHCP **/etc/dhcp/dhcpd.conf**. Pour plus de détails, voir Configuration d'un serveur DHCP pour les clients sans disque .

- Vérifier si les ports du pare-feu sont ouverts.
  - Vérifiez si l'adresse **dhcp.service** est répertoriée dans les services actifs :

```
# firewall-cmd --get-active-zones
# firewall-cmd --info-zone=public
```

  - Si le site **dhcp.service** n'est pas répertorié dans les services actifs, ajoutez-le à la liste :

```
# firewall-cmd --add-service=dhcp --permanent
```

  - Vérifiez si l'adresse **nfs.service** est répertoriée dans les services actifs :

```
# firewall-cmd --get-active-zones
# firewall-cmd --info-zone=public
```

  - Si le site **nfs.service** n'est pas répertorié dans les services actifs, ajoutez-le à la liste :

```
# firewall-cmd --add-service=nfs --permanent
```

Exemple 14.2. Le fichier n'est pas disponible lors du démarrage d'un système distant sans disque

1. Vérifier si le fichier se trouve dans le répertoire **/var/lib/tftpboot/**.
2. Si le fichier se trouve dans le répertoire, vérifiez les autorisations :

```
# chmod 644 pxelinux.0
```

3. Vérifier si les ports du pare-feu sont ouverts.

Exemple 14.3. Échec du démarrage du système après le chargement dekernel/initrd

1. Vérifier si le service NFS est activé sur un serveur.
   a. Vérifier si **nfs.service** fonctionne :

```
# systemctl status nfs.service
```

   b. Si le site **nfs.service** est inactif, vous devez le démarrer et l'activer :

```
# systemctl start nfs.service
# systemctl enable nfs.service
```

2. Vérifiez que les paramètres sont corrects dans le répertoire **/var/lib/tftpboot/pxelinux.cfg/**. Pour plus d'informations, voir Configuration d'un système de fichiers exporté pour les clients sans disque.

3. Vérifier si les ports du pare-feu sont ouverts.

CHAPITRE 15. DÉMARRER AVEC LE SWAP

Utilisez l'espace de pagination pour fournir un stockage temporaire aux processus et aux données inactifs, et pour éviter les erreurs de mémoire lorsque la mémoire physique est pleine. L'espace de pagination agit comme une extension de la mémoire physique et permet au système de continuer à fonctionner correctement même lorsque la mémoire physique est épuisée. Notez que l'utilisation de l'espace de pagination peut ralentir les performances du système. Il est donc préférable d'optimiser l'utilisation de la mémoire physique avant de s'appuyer sur l'espace de pagination.

15.1. VUE D'ENSEMBLE DE L'ESPACE D'ÉCHANGE

Swap space sous Linux est utilisé lorsque la quantité de mémoire physique (RAM) est pleine. Si le système a besoin de plus de ressources mémoire et que la RAM est pleine, les pages inactives de la mémoire sont déplacées vers l'espace d'échange. Bien que l'espace d'échange puisse aider les machines disposant d'une petite quantité de mémoire vive, il ne doit pas être considéré comme un substitut à une plus grande quantité de mémoire vive.

L'espace de pagination est situé sur les disques durs, qui ont un temps d'accès plus lent que la mémoire physique. L'espace de pagination peut être une partition de pagination dédiée (recommandé), un fichier de pagination ou une combinaison de partitions et de fichiers de pagination.

Par le passé, la quantité d'espace de pagination recommandée augmentait linéairement avec la quantité de mémoire vive du système. Cependant, les systèmes modernes comprennent souvent des centaines de gigaoctets de mémoire vive. Par conséquent, l'espace de pagination recommandé est considéré comme une fonction de la charge de travail de la mémoire du système, et non de la mémoire du système.

Ajouter de l'espace de pagination

Voici les différentes manières d'ajouter un espace de pagination :

- Extension de l'espace de pagination sur un volume logique LVM2
- Création d'un volume logique LVM2 pour le swap
- Création d'un fichier d'échange
  Par exemple, vous pouvez augmenter la quantité de mémoire vive de votre système de 1 Go à 2 Go, mais il n'y a que 2 Go d'espace de pagination. Il peut être avantageux d'augmenter l'espace de pagination à 4 Go si vous effectuez des opérations gourmandes en mémoire ou si vous exécutez des applications nécessitant une grande quantité de mémoire.

Suppression de l'espace de pagination

Voici les différentes façons de supprimer un espace de pagination :

- Réduire l'espace de pagination sur un volume logique LVM2
- Suppression d'un volume logique LVM2 pour l'échange
- Suppression d'un fichier d'échange
  Par exemple, vous avez réduit la quantité de RAM de votre système de 1 Go à 512 Mo, mais il reste 2 Go d'espace de pagination. Il peut être avantageux de réduire la quantité d'espace de pagination à 1 Go, car les 2 Go restants risquent de gaspiller de l'espace disque.

15.2. ESPACE DE PAGINATION RECOMMANDÉ

Cette section décrit la taille recommandée d'une partition d'échange en fonction de la quantité de mémoire vive de votre système et si vous souhaitez disposer de suffisamment de mémoire pour que votre système puisse hiberner. La taille recommandée de la partition d'échange est établie automatiquement lors de l'installation. Toutefois, pour permettre la mise en veille prolongée, vous devez modifier l'espace de pagination lors de l'étape de partitionnement personnalisé.

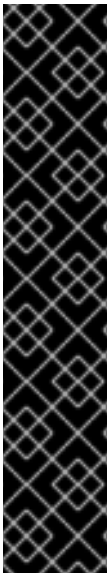
Les recommandations suivantes sont particulièrement importantes pour les systèmes dotés d'une faible mémoire (1 Go ou moins). Le fait de ne pas allouer suffisamment d'espace de pagination sur ces systèmes peut entraîner des problèmes tels que l'instabilité ou même rendre le système installé non amorçable.

Tableau 15.1. Espace de pagination recommandé

Quantité de RAM dans le système	Espace de pagination recommandé	Espace d'échange recommandé en cas d'hibernation
⩽ 2 GO	2 fois plus de RAM	3 fois plus de RAM
> 2 GB - 8 GB	Égal à la quantité de RAM	2 fois plus de RAM
> 8 GB - 64 GB	Au moins 4 Go	1.5 fois plus de RAM
> 64 GB	Au moins 4 Go	L'hibernation n'est pas recommandée

À la limite de chaque plage répertoriée dans ce tableau, par exemple un système avec 2 Go, 8 Go ou 64 Go de RAM, il est possible de choisir l'espace d'échange et la prise en charge de l'hibernation. Si les ressources de votre système le permettent, l'augmentation de l'espace de pagination peut améliorer les performances.

Notez que la répartition de l'espace de pagination sur plusieurs périphériques de stockage améliore également les performances de l'espace de pagination, en particulier sur les systèmes dotés de disques, de contrôleurs et d'interfaces rapides.



IMPORTANT

Les systèmes de fichiers et les volumes LVM2 affectés à l'espace de pagination *should not* sont en cours d'utilisation lorsqu'ils sont modifiés. Toute tentative de modification de l'espace de pagination échoue si un processus système ou le noyau utilise l'espace de pagination. Utilisez les commandes **free** et **cat /proc/swaps** pour vérifier la quantité et l'emplacement de l'espace de pagination utilisé.

Le redimensionnement de l'espace de pagination nécessite de retirer temporairement l'espace de pagination du système. Cette opération peut s'avérer problématique si les applications en cours d'exécution dépendent de l'espace de pagination supplémentaire et risquent de se retrouver dans des situations de mémoire faible. Il est préférable d'effectuer le redimensionnement de l'espace de pagination à partir du mode de secours, voir les options de démarrage de débogage dans le fichier *Performing an advanced RHEL 9 installation*. Lorsque vous êtes invité à monter le système de fichiers, sélectionnez **Ignorer**.

15.3. EXTENSION DE L'ESPACE DE PAGINATION SUR UN VOLUME LOGIQUE LVM2

Cette procédure décrit comment étendre l'espace de pagination sur un volume logique LVM2 existant. En supposant que */dev/VolGroup00/LogVol01* est le volume que vous souhaitez étendre par *2 GB*.

Conditions préalables

- Vous disposez d'un espace disque suffisant.

Procédure

1. Désactiver l'échange pour le volume logique associé :

```
# swapoff -v /dev/VolGroup00/LogVol01
```

2. Redimensionner le volume logique LVM2 par *2 GB*:

```
# lvresize /dev/VolGroup00/LogVol01 -L  2G
```

3. Formatez le nouvel espace de pagination :

```
# mkswap /dev/VolGroup00/LogVol01
```

4. Activer le volume logique étendu :

```
# swapon -v /dev/VolGroup00/LogVol01
```

Vérification

- Pour vérifier si le volume logique de permutation a été étendu et activé avec succès, inspectez l'espace de permutation actif à l'aide de la commande suivante :

```
$ cat /proc/swaps
$ free -h
```

15.4. CRÉATION D'UN VOLUME LOGIQUE LVM2 POUR LE SWAP

Cette procédure décrit comment créer un volume logique LVM2 pour la permutation. En supposant que */dev/VolGroup00/LogVol02* est le volume d'échange que vous souhaitez ajouter.

Conditions préalables

- Vous disposez de suffisamment d'espace disque.

Procédure

1. Créer le volume logique LVM2 de taille *2 GB*:

```
# lvcreate  VolGroup00 -n  LogVol02 -L  2G
```

2. Formatez le nouvel espace de pagination :

```
# mkswap /dev/VolGroup00/LogVol02
```

3. Ajoutez l'entrée suivante au fichier **/etc/fstab**:

```
/dev/VolGroup00/LogVol02 none swap defaults 0 0
```

4. Régénérez les unités de montage pour que votre système enregistre la nouvelle configuration :

```
# systemctl daemon-reload
```

5. Activer le swap sur le volume logique :

```
# swapon -v /dev/VolGroup00/LogVol02
```

Vérification

- Pour vérifier que le volume logique de permutation a été créé et activé avec succès, inspectez l'espace de permutation actif à l'aide de la commande suivante :

```
$ cat /proc/swaps
$ free -h
```

15.5. CRÉATION D'UN FICHIER D'ÉCHANGE

Cette procédure décrit comment créer un fichier d'échange.

Conditions préalables

- Vous disposez de suffisamment d'espace disque.

Procédure

1. Déterminez la taille du nouveau fichier d'échange en mégaoctets et multipliez-la par 1024 pour déterminer le nombre de blocs. Par exemple, la taille des blocs d'un fichier d'échange de 64 Mo est de 65536.

2. Créer un fichier vide :

```
# dd if=/dev/zero of=/swapfile bs=1024 count=65536
```

   Remplacez *65536* par la valeur correspondant à la taille de bloc souhaitée.

3. Configurez le fichier d'échange avec la commande :

```
# mkswap /swapfile
```

4. Modifier la sécurité du fichier d'échange pour qu'il ne soit pas lisible par le monde entier.

```
# chmod 0600 /swapfile
```

5. Modifiez le fichier **/etc/fstab** avec les entrées suivantes pour activer le fichier d'échange au moment du démarrage :

```
/swapfile none swap defaults 0 0
```

   La prochaine fois que le système démarre, il active le nouveau fichier d'échange.

6. Régénérez les unités de montage pour que votre système enregistre la nouvelle configuration **/etc/fstab**:

```
# systemctl daemon-reload
```

7. Activer immédiatement le fichier d'échange :

```
# swapon /swapfile
```

Vérification

- Pour vérifier que le nouveau fichier d'échange a été créé et activé avec succès, inspectez l'espace d'échange actif à l'aide de la commande suivante :

```
$ cat /proc/swaps
$ free -h
```

15.6. RÉDUIRE L'ESPACE DE PAGINATION SUR UN VOLUME LOGIQUE LVM2

Cette procédure décrit comment réduire l'espace de pagination sur un volume logique LVM2. En supposant que */dev/VolGroup00/LogVol01* est le volume que vous souhaitez réduire.

Procédure

1. Désactiver l'échange pour le volume logique associé :

```
# swapoff -v /dev/VolGroup00/LogVol01
```

2. Réduisez le volume logique LVM2 de 512 Mo :

```
# lvreduce /dev/VolGroup00/LogVol01 -L -512M
```

3. Formatez le nouvel espace de pagination :

```
# mkswap /dev/VolGroup00/LogVol01
```

4. Activer le swap sur le volume logique :

```
# swapon -v /dev/VolGroup00/LogVol01
```

Vérification

- Pour vérifier si le volume logique d'échange a été réduit avec succès, inspectez l'espace d'échange actif à l'aide de la commande suivante :

```
$ cat /proc/swaps
$ free -h
```

15.7. SUPPRESSION D'UN VOLUME LOGIQUE LVM2 POUR L'ÉCHANGE

Cette procédure décrit comment supprimer un volume logique LVM2 pour la permutation. En supposant que */dev/VolGroup00/LogVol02* est le volume de swap que vous souhaitez supprimer.

Procédure

1. Désactiver l'échange pour le volume logique associé :

```
# swapoff -v /dev/VolGroup00/LogVol02
```

2. Supprimer le volume logique LVM2 :

```
# lvremove /dev/VolGroup00/LogVol02
```

3. Supprimer l'entrée associée suivante du fichier **/etc/fstab**:

```
/dev/VolGroup00/LogVol02 none swap defaults 0 0
```

4. Régénérer les unités de montage pour enregistrer la nouvelle configuration :

```
# systemctl daemon-reload
```

Vérification

- Testez si le volume logique a été supprimé avec succès, inspectez l'espace de pagination actif à l'aide de la commande suivante :

```
$ cat /proc/swaps
$ free -h
```

15.8. SUPPRESSION D'UN FICHIER D'ÉCHANGE

Cette procédure décrit comment supprimer un fichier d'échange.

Procédure

1. À l'invite d'un shell, exécutez la commande suivante pour désactiver le fichier d'échange, où **/swapfile** est le fichier d'échange :

```
# swapoff -v /swapfile
```

2. Supprimez son entrée du fichier **/etc/fstab** en conséquence.

3. Régénérez les unités de montage pour que votre système enregistre la nouvelle configuration :

```
# systemctl daemon-reload
```

4. Supprimer le fichier actuel :

```
# rm /swapfile
```

CHAPITRE 16. CONFIGURATION DE FIBRE CHANNEL SUR ETHERNET

Basé sur la norme IEEE T11 FC-BB-5, Fibre Channel over Ethernet (FCoE) est un protocole permettant de transmettre des trames Fibre Channel sur des réseaux Ethernet. En règle générale, les centres de données disposent d'un réseau local dédié et d'un réseau de stockage (SAN) qui sont séparés l'un de l'autre avec leur propre configuration spécifique. FCoE combine ces réseaux en une structure unique et convergente. Les avantages de FCoE sont, par exemple, la réduction des coûts de matériel et d'énergie.

16.1. UTILISATION DE HBA FCOE MATÉRIELS DANS RHEL

Dans RHEL, vous pouvez utiliser un adaptateur de bus hôte (HBA) matériel Fibre Channel over Ethernet (FCoE), qui est pris en charge par les pilotes suivants :

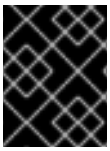
- **qedf**
- **bnx2fc**
- **fnic**

Si vous utilisez un tel adaptateur, vous devez configurer les paramètres FCoE dans la configuration de l'adaptateur. Pour plus d'informations, voir la documentation de l'adaptateur.

Une fois le HBA configuré, les numéros d'unité logique (LUN) exportés du réseau de stockage (SAN) sont automatiquement mis à la disposition de RHEL en tant que périphériques /**dev**/**sd***. Vous pouvez utiliser ces périphériques comme des périphériques de stockage locaux.

16.2. CONFIGURATION D'UN DISPOSITIF FCOE LOGICIEL

Utilisez le logiciel FCoE pour accéder aux numéros d'unité logique (LUN) via FCoE, qui utilise un adaptateur Ethernet prenant partiellement en charge le délestage FCoE.



IMPORTANT

RHEL ne prend pas en charge les périphériques FCoE logiciels qui nécessitent le module de noyau **fcoe.ko**.

Une fois cette procédure terminée, les LUN exportées du réseau de stockage (SAN) sont automatiquement mises à la disposition de RHEL en tant que périphériques /**dev**/**sd***. Vous pouvez utiliser ces périphériques de la même manière que les périphériques de stockage locaux.

Conditions préalables

- Vous avez configuré le commutateur réseau pour qu'il prenne en charge le VLAN.
- Le SAN utilise un VLAN pour séparer le trafic de stockage du trafic Ethernet normal.
- Vous avez configuré le HBA du serveur dans son BIOS.
- Le HBA est connecté au réseau et la liaison est établie. Pour plus d'informations, consultez la documentation de votre HBA.

Procédure

1. Installez le paquetage **fcoe-utils**:

```
# dnf install fcoe-utils
```

2. Copiez le fichier modèle **/etc/fcoe/cfg-ethx** dans **/etc/fcoe/cfg-*interface_name***. Par exemple, si vous voulez configurer l'interface **enp1s0** pour utiliser FCoE, entrez la commande suivante :

```
# cp /etc/fcoe/cfg-ethx /etc/fcoe/cfg-enp1s0
```

3. Activez et démarrez le service **fcoe**:

```
# systemctl enable --now fcoe
```

4. Découvrez le VLAN FCoE sur l'interface **enp1s0**, créez un périphérique réseau pour le VLAN découvert et démarrez l'initiateur :

```
# fipvlan -s -c enp1s0
Created VLAN device enp1s0.200
Starting FCoE on interface enp1s0.200
Fibre Channel Forwarders Discovered
interface       | VLAN | FCF MAC
------------------------------------------
enp1s0          | 200  | 00:53:00:a7:e7:1b
```

5. Facultatif : Affichez des détails sur les cibles découvertes, les LUN et les périphériques associés aux LUN :

```
# fcoeadm -t
Interface:       enp1s0.200
Roles:           FCP Target
Node Name:       0x500a0980824acd15
Port Name:       0x500a0982824acd15
Target ID:       0
MaxFrameSize:    2048 bytes
OS Device Name:  rport-11:0-1
FC-ID (Port ID): 0xba00a0
State:           Online

LUN ID  Device Name   Capacity   Block Size  Description
------  -----------  ----------  ----------  ----------------------
    0  sdb          28.38 GiB      512     NETAPP LUN (rev 820a)
    ...
```

Cet exemple montre que le LUN 0 du SAN a été attaché à l'hôte en tant que périphérique **/dev/sdb**.

Vérification

- Affiche des informations sur toutes les interfaces FCoE actives :

```
# fcoeadm -i
Description:     BCM57840 NetXtreme II 10 Gigabit Ethernet
Revision:        11
Manufacturer:    Broadcom Inc. and subsidiaries
```

```
Serial Number:    000AG703A9B7

Driver:           bnx2x Unknown
Number of Ports:  1

    Symbolic Name:     bnx2fc (QLogic BCM57840) v2.12.13 over enp1s0.200
    OS Device Name:    host11
    Node Name:         0x2000000af70ae935
    Port Name:         0x2001000af70ae935
    Fabric Name:       0x20c8002a6aa7e701
    Speed:             10 Gbit
    Supported Speed:   1 Gbit, 10 Gbit
    MaxFrameSize:      2048 bytes
    FC-ID (Port ID):   0xba02c0
    State:             Online
```

Ressources supplémentaires

- **fcoeadm(8)** page de manuel
- **/usr/share/doc/fcoe-utils/README**
- Utilisation de périphériques Fibre Channel

CHAPITRE 17. GESTION DES PÉRIPHÉRIQUES DE BANDE

Un périphérique à bande est une bande magnétique sur laquelle les données sont stockées et accessibles de manière séquentielle. Les données sont écrites sur ce périphérique à l'aide d'un lecteur de bande. Il n'est pas nécessaire de créer un système de fichiers pour stocker des données sur un périphérique à bande. Les lecteurs de bandes peuvent être connectés à un ordinateur hôte avec différentes interfaces telles que SCSI, FC, USB, SATA, et d'autres interfaces.

17.1. TYPES D'APPAREILS À BANDE

Voici une liste des différents types de périphériques de bande :

- **/dev/st0** est un dispositif de rembobinage de la bande.
- **/dev/nst0** est une unité de bande non rembobinée. Utilisez des périphériques sans rembobinage pour les sauvegardes quotidiennes.

L'utilisation de dispositifs à bande présente plusieurs avantages. Ils sont rentables et stables. Ils résistent également à la corruption des données et conviennent à la conservation des données.

17.2. INSTALLATION DE L'OUTIL DE GESTION DES LECTEURS DE BANDE

Utilisez la commande **mt** pour enrouler les données dans les deux sens. L'utilitaire **mt** contrôle les opérations du lecteur de bande magnétique et l'utilitaire **st** est utilisé pour le pilote de bande SCSI. Cette procédure décrit comment installer le paquetage **mt-st** pour les opérations de lecteur de bande.

Procédure

- Installez le paquetage **mt-st**:

```
# dnf install mt-st
```

Ressources supplémentaires

- **mt(1)** et **st(4)** pages de manuel

17.3. ÉCRITURE SUR DES DISPOSITIFS DE REMBOBINAGE DE BANDE

Un dispositif de rembobinage rembobine la bande après chaque opération. Pour sauvegarder des données, vous pouvez utiliser la commande **tar**. Par défaut, dans les périphériques à bande, la valeur de **block size** est de 10 Ko (**bs=10k**). Vous pouvez définir la variable d'environnement **TAPE** à l'aide de l'attribut **export TAPE=*/dev/st0*** pour définir la variable d'environnement. Utilisez plutôt l'option **-f** device pour spécifier le fichier du périphérique de bande. Cette option est utile lorsque vous utilisez plus d'un périphérique de bande.

Conditions préalables

1. Vous avez installé le paquetage **mt-st**. Pour plus d'informations, voir Installation de l'outil de gestion des lecteurs de bande.
2. Charger le lecteur de bande :

```
# mt -f /dev/st0 load
```

Procédure

1. Vérifier la tête de lecture :

```
# mt -f /dev/st0 status

SCSI 2 tape drive:
File number=-1, block number=-1, partition=0.
Tape block size 0 bytes. Density code 0x0 (default).
Soft error count since last status=0
General status bits on (50000):
 DR_OPEN IM_REP_EN
```

   Ici :

   - la valeur actuelle de **file number** est -1.
   - le site **block number** définit la tête de bande. Par défaut, il est fixé à -1.
   - le **block size** 0 indique que l'unité de bande n'a pas de taille de bloc fixe.
   - le site **Soft error count** indique le nombre d'erreurs rencontrées après l'exécution de la commande mt status.
   - le site **General status bits** explique les caractéristiques de l'appareil à bandes.
   - **DR_OPEN** indique que la porte est ouverte et que le dispositif de bande est vide. **IM_REP_EN** est le mode de rapport immédiat.

2. Si le périphérique de bande n'est pas vide, écrasez-le :

```
# tar -czf /dev/st0 _/source/directory
```

   Cette commande écrase les données d'un périphérique de bande avec le contenu de ***/source/directory***.

3. Sauvegarder le ***/source/directory*** sur l'appareil à bandes :

```
# tar -czf /dev/st0 _/source/directory
tar: Removing leading `/' from member names
/source/directory
/source/directory/man_db.conf
/source/directory/DIR_COLORS
/source/directory/rsyslog.conf
[...]
```

4. Visualiser l'état de l'unité de bande :

```
# mt -f /dev/st0  status
```

Verification steps

- Affiche la liste de tous les fichiers présents sur l'unité de bande :

```
# tar -tzf /dev/st0
/source/directory/
/source/directory/man_db.conf
/source/directory/DIR_COLORS
/source/directory/rsyslog.conf
[...]
```

Ressources supplémentaires

- **mt(1)**, **st(4)**, et **tar(1)** pages de manuel
- Support de lecteur de bande détecté comme étant protégé en écriture Article de Red Hat Knowlegebase
- Comment vérifier si les lecteurs de bandes sont détectés dans le système Article de Red Hat Knowlegebase

17.4. ÉCRITURE SUR DES DISPOSITIFS À BANDE NON REMBOBINÉE

Un périphérique à bande non réenroulable laisse la bande dans son état actuel, après l'exécution d'une certaine commande. Par exemple, après une sauvegarde, vous pouvez ajouter des données supplémentaires sur un périphérique à bande sans rembobinage. Vous pouvez également l'utiliser pour éviter tout rembobinage inattendu.

Conditions préalables

1. Vous avez installé le paquetage **mt-st**. Pour plus d'informations, voir Installation de l'outil de gestion des lecteurs de bande.
2. Charger le lecteur de bande :

```
# mt -f /dev/nst0 load
```

Procédure

1. Vérifiez la tête de bande de l'appareil à bande non rembobinée **/dev/nst0**:

```
# mt -f /dev/nst0 status
```

2. Spécifier le pointeur à la tête ou à la fin de la bande :

```
# mt -f /dev/nst0 rewind
```

3. Ajouter les données sur le périphérique de bande :

```
# mt -f /dev/nst0 eod
# tar -czf /dev/nst0 /source/directory/
```

4. Sauvegarder le ***/source/directory/*** sur l'appareil à bandes :

```
# tar -czf /dev/nst0 /source/directory/
```

```
tar: Removing leading `/' from member names
/source/directory/
/source/directory/man_db.conf
/source/directory/DIR_COLORS
/source/directory/rsyslog.conf
[...]
```

5. Visualiser l'état de l'unité de bande :

```
# mt -f /dev/nst0  status
```

Verification steps

- Affiche la liste de tous les fichiers présents sur l'unité de bande :

```
# tar -tzf /dev/nst0
/source/directory/
/source/directory/man_db.conf
/source/directory/DIR_COLORS
/source/directory/rsyslog.conf
[...]
```

Ressources supplémentaires

- **mt(1)**, **st(4)**, et **tar(1)** pages de manuel
- Support de lecteur de bande détecté comme étant protégé en écriture Article de Red Hat Knowlegebase
- Comment vérifier si les lecteurs de bandes sont détectés dans le système Article de Red Hat Knowlegebase

17.5. CHANGEMENT DE TÊTE DE BANDE DANS LES APPAREILS À BANDE

Utilisez la procédure suivante pour changer la tête de bande dans le dispositif de bande.

Conditions préalables

1. Vous avez installé le paquetage **mt-st**. Pour plus d'informations, voir Installation de l'outil de gestion des lecteurs de bande.
2. Les données sont écrites sur le périphérique à bande. Pour plus d'informations, voir Écriture sur des périphériques à bande rembobinée ou Écriture sur des périphériques à bande non rembobinée.

Procédure

- Pour visualiser la position actuelle du pointeur de bande :

```
# mt -f /dev/nst0 tell
```

- Pour commuter la tête de la bande, tout en ajoutant les données aux périphériques de la bande :

```
# mt -f /dev/nst0 eod
```

- Pour aller à l'enregistrement précédent :

```
# mt -f /dev/nst0 bsfm 1
```

- Pour passer à l'enregistrement suivant :

```
# mt -f /dev/nst0 fsf 1
```

Ressources supplémentaires

- **mt(1)** page de manuel

17.6. RESTAURATION DE DONNÉES À PARTIR DE PÉRIPHÉRIQUES À BANDES

Pour restaurer les données d'un périphérique à bandes, utilisez la commande **tar**.

Conditions préalables

1. Vous avez installé le paquetage **mt-st**. Pour plus d'informations, voir Installation de l'outil de gestion des lecteurs de bande.
2. Les données sont écrites sur le périphérique à bande. Pour plus d'informations, voir Écriture sur des périphériques à bande rembobinée ou Écriture sur des périphériques à bande non rembobinée.

Procédure

- Pour le rembobinage des appareils à bande **/dev/st0**:
  - Rétablir le ***/source/directory/***:

```
# tar -xzf /dev/st0 /source/directory /
```

- Pour les dispositifs à bande non enroulable **/dev/nst0**:
  - Rembobine l'appareil à bande :

```
# mt -f /dev/nst0 rewind
```

  - Restaurer le répertoire **etc**:

```
# tar -xzf /dev/nst0 /source/directory /
```

Ressources supplémentaires

- **mt(1)** et **tar(1)** pages de manuel

17.7. EFFACEMENT DES DONNÉES DES PÉRIPHÉRIQUES À BANDES

Pour effacer les données d'un périphérique à bande, utilisez l'option **erase**.

Conditions préalables

1. Vous avez installé le paquetage **mt-st**. Pour plus d'informations, voir Installation de l'outil de gestion des lecteurs de bande.
2. Les données sont écrites sur le périphérique à bande. Pour plus d'informations, voir Écriture sur des périphériques à bande rembobinée ou Écriture sur des périphériques à bande non rembobinée.

Procédure

1. Efface les données de l'unité de bande :

```
# mt -f /dev/st0 erase
```

2. Décharger l'appareil à bande :

```
mt -f /dev/st0 hors ligne
```

Ressources supplémentaires

- **mt(1)** page de manuel

17.8. COMMANDES DE BANDES

Voici les commandes les plus courantes sur **mt**:

Tableau 17.1. commandes mt

Commandement	Description
mt -f /dev/*st0* status	Affiche l'état de l'unité de bande.
mt -f /dev/*st0* erase	Efface la totalité de la bande.
mt -f /dev/*nst0* rewind	Rembobine l'appareil à bande.
mt -f /dev/*nst0* fsf *n*	Commute la tête de lecture de la bande sur l'enregistrement avant. Ici, *n* est un nombre de fichiers optionnel. Si un nombre de fichiers est spécifié, la tête de lecture saute les enregistrements *n*.
mt -f /dev/*nst0* bsfm *n*	Fait passer la tête de lecture de la bande à l'enregistrement précédent.
mt -f /dev/*nst0* eod	Commute la tête de la bande à la fin des données.

CHAPITRE 18. GESTION DU RAID

Vous pouvez utiliser une matrice redondante de disques indépendants (RAID) pour stocker des données sur plusieurs disques. Cela permet d'éviter les pertes de données en cas de défaillance d'un disque.

18.1. VUE D'ENSEMBLE DU RAID

Dans un RAID, plusieurs périphériques, tels que des disques durs, des disques SSD ou des NVMe, sont combinés dans une matrice pour atteindre des objectifs de performance ou de redondance impossibles à réaliser avec un seul disque de grande taille et coûteux. Cette matrice de périphériques apparaît à l'ordinateur comme une seule unité de stockage logique ou lecteur.

Le RAID prend en charge différentes configurations, notamment les niveaux 0, 1, 4, 5, 6, 10 et linéaire. Le RAID utilise des techniques telles que le striping (RAID niveau 0), le mirroring (RAID niveau 1) et le striping avec parité (RAID niveaux 4, 5 et 6) pour assurer la redondance, réduire la latence, augmenter la bande passante et maximiser la capacité de récupération en cas de défaillance du disque dur.

Le système RAID répartit les données sur chaque périphérique de la matrice en les divisant en morceaux de taille constante, généralement 256 Ko ou 512 Ko, bien que d'autres valeurs soient acceptables. Il écrit ces morceaux sur un disque dur de la matrice RAID en fonction du niveau RAID utilisé. Lors de la lecture des données, le processus est inversé, ce qui donne l'illusion que les multiples périphériques de la matrice constituent en fait un seul grand disque.

La technologie RAID est bénéfique pour ceux qui gèrent de grandes quantités de données. Voici les principales raisons de déployer le RAID :

- Il améliore la vitesse
- Il augmente la capacité de stockage en utilisant un seul disque virtuel
- Il minimise la perte de données en cas de défaillance du disque
- La configuration RAID et le niveau de conversion en ligne

18.2. TYPES DE RAID

Les types de RAID possibles sont les suivants :

Firmware RAID

Le RAID micrologiciel, également connu sous le nom d'ATARAID, est un type de RAID logiciel dans lequel les jeux RAID peuvent être configurés à l'aide d'un menu basé sur un micrologiciel. Le micrologiciel utilisé par ce type de RAID s'accroche également au BIOS, ce qui vous permet de démarrer à partir de ses jeux RAID. Différents fournisseurs utilisent différents formats de métadonnées sur disque pour marquer les membres du jeu RAID. L'Intel Matrix RAID est un exemple de système RAID à microprogrammation.

RAID matériel

Une matrice matérielle gère le sous-système RAID indépendamment de l'hôte. Elle peut présenter à l'hôte plusieurs périphériques par matrice RAID.
Les dispositifs RAID matériels peuvent être internes ou externes au système. Les périphériques internes consistent généralement en une carte contrôleur spécialisée qui gère les tâches RAID de manière transparente pour le système d'exploitation. Les périphériques externes se connectent généralement au système via SCSI, Fibre Channel, iSCSI, InfiniBand ou toute autre interconnexion réseau à grande vitesse et présentent au système des volumes tels que des unités logiques.

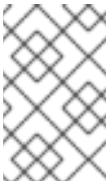
Les cartes contrôleur RAID fonctionnent comme un contrôleur SCSI pour le système d'exploitation et gèrent toutes les communications avec les disques. Vous pouvez brancher les disques dans le contrôleur RAID comme dans un contrôleur SCSI normal, puis les ajouter à la configuration du contrôleur RAID. Le système d'exploitation ne pourra pas faire la différence.

RAID logiciel

Un RAID logiciel met en œuvre les différents niveaux de RAID dans le code de périphérique de bloc du noyau. Il s'agit de la solution la moins chère car elle ne nécessite pas de cartes de contrôleur de disque onéreuses ni de châssis remplaçables à chaud. Avec les châssis remplaçables à chaud, vous pouvez retirer un disque dur sans éteindre votre système. Le RAID logiciel fonctionne également avec tous les blocs de stockage pris en charge par le noyau Linux, tels que SATA, SCSI et NVMe. Avec les processeurs plus rapides d'aujourd'hui, le RAID logiciel est généralement plus performant que le RAID matériel, à moins que vous n'utilisiez des périphériques de stockage haut de gamme. Comme le noyau Linux contient un pilote de périphériques multiples (MD), la solution RAID devient complètement indépendante du matériel. Les performances d'une matrice logicielle dépendent des performances et de la charge du processeur du serveur.

Voici les principales caractéristiques de la pile RAID logicielle Linux :

- Conception multithread
- Portabilité des tableaux entre machines Linux sans reconstruction
- Reconstruction d'un réseau en arrière-plan en utilisant les ressources inactives du système
- Prise en charge des lecteurs à chaud
- Détection automatique de l'unité centrale pour tirer parti de certaines caractéristiques de l'unité centrale, telles que la prise en charge de l'instruction unique et des données multiples (SIMD).
- Correction automatique des secteurs défectueux sur les disques d'une matrice.
- Contrôles de cohérence réguliers des données RAID pour garantir la santé de la matrice.
- Surveillance proactive des réseaux avec envoi d'alertes par courrier électronique à une adresse désignée en cas d'événements importants.
- Les bitmaps d'intention d'écriture, qui augmentent considérablement la vitesse des événements de resynchronisation en permettant au noyau de savoir précisément quelles parties d'un disque doivent être resynchronisées au lieu d'avoir à resynchroniser l'ensemble de la matrice après un crash du système.



NOTE

La resynchronisation est un processus qui permet de synchroniser les données sur les périphériques du RAID existant afin d'obtenir une redondance.

- Point de contrôle de la resynchronisation de sorte que si vous redémarrez votre ordinateur pendant une resynchronisation, la resynchronisation reprend au démarrage là où elle s'est arrêtée et ne recommence pas à zéro.
- La possibilité de modifier les paramètres de la matrice après l'installation, ce que l'on appelle le remodelage. Par exemple, vous pouvez transformer une matrice RAID5 à 4 disques en une

matrice RAID5 à 5 disques lorsque vous avez un nouveau périphérique à ajouter. Cette opération de croissance est effectuée en direct et ne nécessite pas de réinstallation sur la nouvelle matrice.

- Le remodelage permet de modifier le nombre de périphériques, l'algorithme RAID ou la taille du type de matrice RAID, tel que RAID4, RAID5, RAID6 ou RAID10.
- Takeover prend en charge la conversion des niveaux RAID, par exemple de RAID0 à RAID6.
- Cluster MD, qui est une solution de stockage pour une grappe, fournit la redondance du miroir RAID1 à la grappe. Actuellement, seul le RAID1 est pris en charge.

18.3. RAID LEVELS AND LINEAR SUPPORT

The following are the supported configurations by RAID, including levels 0, 1, 4, 5, 6, 10, and linear:

Niveau 0

RAID level 0, often called striping, is a performance-oriented striped data mapping technique. This means the data being written to the array is broken down into stripes and written across the member disks of the array, allowing high I/O performance at low inherent cost but provides no redundancy. RAID level 0 implementations only stripe the data across the member devices up to the size of the smallest device in the array. This means that if you have multiple devices with slightly different sizes, each device gets treated as though it was the same size as the smallest drive. Therefore, the common storage capacity of a level 0 array is the total capacity of all disks. If the member disks have a different size, then the RAID0 uses all the space of those disks using the available zones.

Niveau 1

RAID level 1, or mirroring, provides redundancy by writing identical data to each member disk of the array, leaving a mirrored copy on each disk. Mirroring remains popular due to its simplicity and high level of data availability. Level 1 operates with two or more disks, and provides very good data reliability and improves performance for read-intensive applications but at relatively high costs. RAID level 1 is costly because you write the same information to all of the disks in the array, which provides data reliability, but in a much less space-efficient manner than parity based RAID levels such as level 5. However, this space inefficiency comes with a performance benefit, which is parity-based RAID levels that consume considerably more CPU power in order to generate the parity while RAID level 1 simply writes the same data more than once to the multiple RAID members with very little CPU overhead. As such, RAID level 1 can outperform the parity-based RAID levels on machines where software RAID is employed and CPU resources on the machine are consistently taxed with operations other than RAID activities.

The storage capacity of the level 1 array is equal to the capacity of the smallest mirrored hard disk in a hardware RAID or the smallest mirrored partition in a software RAID. Level 1 redundancy is the highest possible among all RAID types, with the array being able to operate with only a single disk present.

Niveau 4

Level 4 uses parity concentrated on a single disk drive to protect data. Parity information is calculated based on the content of the rest of the member disks in the array. This information can then be used to reconstruct data when one disk in the array fails. The reconstructed data can then be used to satisfy I/O requests to the failed disk before it is replaced and to repopulate the failed disk after it has been replaced.
Since the dedicated parity disk represents an inherent bottleneck on all write transactions to the RAID array, level 4 is seldom used without accompanying technologies such as write-back caching.

Or it is used in specific circumstances where the system administrator is intentionally designing the software RAID device with this bottleneck in mind such as an array that has little to no write transactions once the array is populated with data. RAID level 4 is so rarely used that it is not available as an option in Anaconda. However, it could be created manually by the user if needed.

The storage capacity of hardware RAID level 4 is equal to the capacity of the smallest member partition multiplied by the number of partitions minus one. The performance of a RAID level 4 array is always asymmetrical, which means reads outperform writes. This is because write operations consume extra CPU resources and main memory bandwidth when generating parity, and then also consume extra bus bandwidth when writing the actual data to disks because you are not only writing the data, but also the parity. Read operations need only read the data and not the parity unless the array is in a degraded state. As a result, read operations generate less traffic to the drives and across the buses of the computer for the same amount of data transfer under normal operating conditions.

Niveau 5

This is the most common type of RAID. By distributing parity across all the member disk drives of an array, RAID level 5 eliminates the write bottleneck inherent in level 4. The only performance bottleneck is the parity calculation process itself. Modern CPUs can calculate parity very fast. However, if you have a large number of disks in a RAID 5 array such that the combined aggregate data transfer speed across all devices is high enough, parity calculation can be a bottleneck.
Level 5 has asymmetrical performance, and reads substantially outperforming writes. The storage capacity of RAID level 5 is calculated the same way as with level 4.

Level 6

This is a common level of RAID when data redundancy and preservation, and not performance, are the paramount concerns, but where the space inefficiency of level 1 is not acceptable. Level 6 uses a complex parity scheme to be able to recover from the loss of any two drives in the array. This complex parity scheme creates a significantly higher CPU burden on software RAID devices and also imposes an increased burden during write transactions. As such, level 6 is considerably more asymmetrical in performance than levels 4 and 5.
The total capacity of a RAID level 6 array is calculated similarly to RAID level 5 and 4, except that you must subtract two devices instead of one from the device count for the extra parity storage space.

Level 10

This RAID level attempts to combine the performance advantages of level 0 with the redundancy of level 1. It also reduces some of the space wasted in level 1 arrays with more than two devices. With level 10, it is possible, for example, to create a 3-drive array configured to store only two copies of each piece of data, which then allows the overall array size to be 1.5 times the size of the smallest devices instead of only equal to the smallest device, similar to a 3-device, level 1 array. This avoids CPU process usage to calculate parity similar to RAID level 6, but it is less space efficient.
The creation of RAID level 10 is not supported during installation. It is possible to create one manually after installation.

Linear RAID

Linear RAID is a grouping of drives to create a larger virtual drive.
In linear RAID, the chunks are allocated sequentially from one member drive, going to the next drive only when the first is completely filled. This grouping provides no performance benefit, as it is unlikely that any I/O operations split between member drives. Linear RAID also offers no redundancy and decreases reliability. If any one member drive fails, the entire array cannot be used and data can be lost. The capacity is the total of all member disks.

18.4. SOUS-SYSTÈMES RAID LINUX

Les sous-systèmes suivants composent le RAID sous Linux :

Pilotes de contrôleurs RAID matériels pour Linux

Les contrôleurs RAID matériels n'ont pas de sous-système RAID spécifique sous Linux. Étant donné qu'ils utilisent des chipsets RAID spéciaux, les contrôleurs RAID matériels sont livrés avec leurs propres pilotes. Avec ces pilotes, le système détecte les jeux RAID comme des disques normaux.

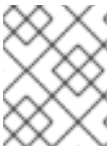
mdraid

Le sous-système **mdraid** a été conçu comme une solution RAID logicielle pour Linux. Il s'agit également de la solution préférée pour le RAID logiciel dans Red Hat Enterprise Linux. Ce sous-système utilise son propre format de métadonnées, appelé métadonnées MD natives.
Il prend également en charge d'autres formats de métadonnées, connus sous le nom de métadonnées externes. Red Hat Enterprise Linux 9 utilise **mdraid** avec des métadonnées externes pour accéder aux ensembles Intel Rapid Storage (ISW) ou Intel Matrix Storage Manager (IMSM) et au format Disk Drive Format (DDF) de la Storage Networking Industry Association (SNIA). Les ensembles du sous-système **mdraid** sont configurés et contrôlés par l'utilitaire **mdadm**.

18.5. CRÉATION D'UN RAID LOGICIEL PENDANT L'INSTALLATION

Les dispositifs RAID (Redundant Arrays of Independent Disks) sont constitués de plusieurs dispositifs de stockage agencés de manière à améliorer les performances et, dans certaines configurations, la tolérance aux pannes.

Un dispositif RAID est créé en une seule étape et les disques sont ajoutés ou supprimés si nécessaire. Vous pouvez configurer une partition RAID pour chaque disque physique de votre système, de sorte que le nombre de disques disponibles pour le programme d'installation détermine les niveaux de périphérique RAID disponibles. Par exemple, si votre système dispose de deux disques durs, vous ne pouvez pas créer un dispositif RAID 10, car il nécessite un minimum de trois disques distincts.



NOTE

Sur IBM Z 64 bits, le sous-système de stockage utilise RAID de manière transparente. Vous ne devez pas configurer manuellement le RAID logiciel.

Conditions préalables

- Vous devez avoir sélectionné deux disques ou plus pour l'installation avant que les options de configuration RAID ne soient visibles. Selon le type de RAID que vous souhaitez créer, au moins deux disques sont nécessaires.
- Vous avez créé un point de montage. En configurant un point de montage, vous pouvez configurer le périphérique RAID.
- Vous avez sélectionné la case d'option **Personnalisé** dans la fenêtre **Installation Destination**.

Procédure

1. Dans le volet gauche de la fenêtre **Manual Partitioning**, sélectionnez la partition requise.
2. Sous la section **Device(s)**, cliquez sur **Modifier**. La boîte de dialogue **Configure Mount Point** s'ouvre.
3. Sélectionnez les disques que vous souhaitez inclure dans le périphérique RAID et cliquez sur **Sélectionner**.

4. Cliquez sur le menu déroulant **Device Type** et sélectionnez **RAID**.
5. Cliquez sur le menu déroulant **File System** et sélectionnez votre type de système de fichiers préféré.
6. Cliquez sur le menu déroulant **RAID Level** et sélectionnez le niveau de RAID de votre choix.
7. Cliquez sur **Mettre à jour les paramètres** pour enregistrer vos modifications.
8. Cliquez sur **Terminé** pour appliquer les paramètres et revenir à la fenêtre **Installation Summary**.

Ressources supplémentaires

- Création d'un LV RAID avec intégrité DM

18.6. CRÉATION D'UN RAID LOGICIEL SUR UN SYSTÈME INSTALLÉ

Vous pouvez créer un réseau redondant de disques indépendants (RAID) sur un système existant à l'aide de l'utilitaire **mdadm**.

Conditions préalables

- Le paquet **mdadm** est installé.
- Deux partitions ou plus existent sur votre système. Pour des instructions détaillées, voir Création d'une partition avec parted .

Procédure

1. Créez un RAID de deux blocs, par exemple */dev/sda1* et */dev/sdc1*:

```
# mdadm --create /dev/md0 --level=2 --raid-devices=2 /dev/sda1 /dev/sdc1
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
```

   L'option *level_value* définit le niveau de RAID.

2. Facultatif : Vérifiez l'état du RAID :

```
# mdadm --detail /dev/md0
/dev/md0:
           Version : 1.2
     Creation Time : Thu Oct 13 15:17:39 2022
        Raid Level : raid0
        Array Size : 18649600 (17.79 GiB 19.10 GB)
      Raid Devices : 2
     Total Devices : 2
       Persistence : Superblock is persistent

       Update Time : Thu Oct 13 15:17:39 2022
             State : clean
    Active Devices : 2
   Working Devices : 2
```

```
    Failed Devices : 0
     Spare Devices : 0
[...]
```

3. Facultatif : Observez les informations détaillées concernant chaque périphérique du RAID :

```
# mdadm --examine /dev/sda1 /dev/sdc1
/dev/sda1:
          Magic : a92b4efc
        Version : 1.2
    Feature Map : 0x1000
     Array UUID : 77ddfb0a:41529b0e:f2c5cde1:1d72ce2c
           Name : 0
  Creation Time : Thu Oct 13 15:17:39 2022
     Raid Level : raid0
   Raid Devices : 2
[...]
```

4. Créez un système de fichiers sur le lecteur RAID :

```
# mkfs -t xfs /dev/md0
```

Remplacez *xfs* par le système de fichiers avec lequel vous avez choisi de formater le disque.

5. Créez un point de montage pour le lecteur RAID et montez-le :

```
# mkdir /mnt/raid1
# mount /dev/md0 /mnt/raid1
```

Remplacez */mnt/raid1* par le point de montage.

Si vous souhaitez que RHEL monte automatiquement le périphérique RAID **md0** au démarrage du système, ajoutez une entrée pour votre périphérique dans le fichier **/etc/fstab file**:

```
/dev/md0   /mnt/raid1 xfs  defaults   0 0
```

18.7. 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

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

18.8. EXTENSION DU RAID

Vous pouvez étendre un RAID en utilisant l'option **--grow** de l'utilitaire **mdadm**.

Conditions préalables

- Espace disque suffisant.
- Le paquet **parted** est installé.

Procédure

1. Étendre les partitions RAID. Pour plus d'informations, voir Redimensionnement d'une partition avec parted.
2. Étendre le RAID au maximum de la capacité de la partition :

```
# mdadm --grow --size=max /dev/md0
```

   Pour définir une taille spécifique, écrivez la valeur du paramètre **--size** en kB, par exemple **--size=*524228***.

3. Augmentez la taille du système de fichiers. Par exemple, si le volume utilise XFS et est monté sur */mnt/*, entrez :

```
# xfs_growfs /mnt/
```

Ressources supplémentaires

- La page de manuel **mdadm(8)**
- Gestion des systèmes de fichiers

18.9. RÉDUCTION DU RAID

Vous pouvez réduire le RAID en utilisant l'option **--grow** de l'utilitaire **mdadm**.

IMPORTANT

Le système de fichiers XFS ne prend pas en charge la réduction.

Conditions préalables

- Le paquet **parted** est installé.

Procédure

1. Réduire le système de fichiers. Pour plus d'informations, voir Gestion des systèmes de fichiers .
2. Réduire le RAID à la taille voulue, par exemple à *512 MB*:

```
# mdadm --grow --size=524228 /dev/md0
```

   Inscrivez le paramètre **--size** en kB.

3. Réduisez la partition à la taille dont vous avez besoin.

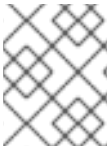
Ressources supplémentaires

- La page de manuel **mdadm(8)**
- Redimensionnement d'une partition avec parted .

18.10. CONVERSIONS RAID PRISES EN CHARGE

Il est possible de passer d'un niveau RAID à un autre. Par exemple, vous pouvez passer de RAID5 à RAID10, mais pas de RAID10 à RAID5. Le tableau suivant décrit les conversions RAID prises en charge :

Niveau de la source	Niveau de destination
RAID0	RAID4, RAID5, RAID10
RAID1	RAID0, RAID5
RAID4	RAID0, RAID5
RAID5	RAID0, RAID1, RAID4, RAID6, RAID10
RAID6	RAID5
RAID10	RAID0



NOTE

La conversion de RAID 5 en RAID0 et RAID4 n'est possible qu'avec la configuration **ALGORITHM_PARITY_N**.

Ressources complémentaires.

- La page de manuel **mdadm(8)**

18.11. CONVERSION D'UN NIVEAU RAID

Vous pouvez convertir le RAID en un niveau RAID différent si nécessaire. L'exemple suivant convertit le périphérique RAID **/dev/md0** de niveau 0 en 5 et ajoute un disque supplémentaire */dev/sdd* à la matrice.

Conditions préalables

- Suffisamment de disques pour la conversion.
- Le paquet **mdadm** est installé.
- Assurez-vous que la conversion envisagée est prise en charge. Voir Conversions RAID prises en charge.

Procédure

1. Convertir le RAID /dev/md0 au niveau RAID *5*:

```
# mdadm --grow --level=5 -n 3 /dev/md0 --force
```

2. Ajouter un nouveau disque à la matrice :

```
# mdadm --manage /dev/md0 --add /dev/sdd
```

Vérification

- Vérifiez que le niveau RAID est converti :

```
# mdadm --detail /dev/md0
/dev/md0:
           Version : 1.2
     Creation Time : Thu Oct 13 15:17:39 2022
        Raid Level : raid0
        Array Size : 18649600 (17.79 GiB 19.10 GB)
      Raid Devices : 5
[...]
```

Ressources supplémentaires

- La page de manuel **mdadm(8)**

18.12. CONVERSION D'UN DISQUE RACINE EN RAID1 APRÈS L'INSTALLATION

Cette section décrit comment convertir un disque racine non-RAID en un miroir RAID1 après l'installation de Red Hat Enterprise Linux 9.

Sur l'architecture PowerPC (PPC), suivez les étapes supplémentaires suivantes :

Conditions préalables

- Les instructions de l'article suivant de la base de connaissances de Red Hat sont terminées : Comment convertir mon disque racine en RAID1 après l'installation de Red Hat Enterprise Linux 7 ?

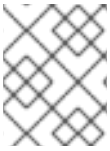
Procédure

1. Copiez le contenu de la partition de démarrage de la plate-forme de référence PowerPC (PReP) de */dev/sda1* à */dev/sdb1*:

```
# dd if=/dev/sda1 of=/dev/sdb1
```

2. Mettez à jour les drapeaux **prep** et **boot** sur la première partition des deux disques :

```
$ parted /dev/sda set 1 prep on
$ parted /dev/sda set 1 boot on

$ parted /dev/sdb set 1 prep on
$ parted /dev/sdb set 1 boot on
```



NOTE

L'exécution de la commande **grub2-install** ***/dev/sda*** ne fonctionne pas sur une machine PowerPC et renvoie une erreur, mais le système démarre comme prévu.

18.13. CRÉATION DE DISPOSITIFS RAID AVANCÉS

Dans certains cas, vous pouvez vouloir installer le système d'exploitation sur une matrice créée avant la fin de l'installation. En général, cela signifie qu'il faut configurer le système de fichiers **/boot** ou le système de fichiers racine sur un périphérique RAID complexe. Dans ce cas, vous devrez peut-être utiliser des options de réseau qui ne sont pas prises en charge par le programme d'installation d'Anaconda. Pour contourner ce problème, suivez les étapes suivantes.

NOTE

Le mode Rescue limité du programme d'installation n'inclut pas de pages de manuel. Les pages de manuel **mdadm** et **md** contiennent des informations utiles pour la création de matrices RAID personnalisées et peuvent être nécessaires tout au long de la solution de contournement.

Procédure

1. Insérer le disque d'installation.
2. Lors du démarrage initial, sélectionnez **Rescue Mode** au lieu de **Install** ou **Upgrade**. Lorsque le système démarre complètement à **Rescue mode**, vous pouvez voir le terminal de ligne de commande.
3. À partir de ce terminal, exécutez les commandes suivantes :
   a. Créez des partitions RAID sur les disques durs cibles à l'aide de la commande **parted**.
   b. Créez manuellement des matrices raid en utilisant la commande **mdadm** à partir de ces partitions en utilisant tous les paramètres et options disponibles.
4. Facultatif : Après avoir créé des tableaux, créez également des systèmes de fichiers sur les tableaux.
5. Redémarrez l'ordinateur et sélectionnez **Install** ou **Upgrade** pour l'installation. Lorsque le programme d'installation d'Anaconda recherche les disques du système, il trouve les périphériques RAID préexistants.
6. Lorsque l'on vous demande comment utiliser les disques du système, sélectionnez **Custom Layout** et cliquez sur **Suivant**. Dans la liste des périphériques, les périphériques MD RAID préexistants sont répertoriés.
7. Sélectionnez un périphérique RAID et cliquez sur **Modifier**.
8. Configurez son point de montage et éventuellement le type de système de fichiers à utiliser si vous n'en avez pas créé un auparavant, puis cliquez sur **Terminé**. Anaconda s'installe sur ce périphérique RAID préexistant, en préservant les options personnalisées que vous avez sélectionnées lorsque vous l'avez créé en mode Rescue.

18.14. MISE EN PLACE DE NOTIFICATIONS PAR COURRIER ÉLECTRONIQUE POUR SURVEILLER UN RAID

Vous pouvez configurer des alertes par courrier électronique pour surveiller le RAID à l'aide de l'outil **mdadm**. Une fois que la variable **MAILADDR** est définie sur l'adresse électronique requise, le système de surveillance envoie les alertes à l'adresse électronique ajoutée.

Conditions préalables

- Le paquet **mdadm** est installé.
- Le service de messagerie est mis en place.

Procédure

1. Créez le fichier de configuration **/etc/mdadm.conf** pour la surveillance de la matrice en analysant les détails du RAID :

```
# mdadm --detail --scan >> /etc/mdadm.conf
```

   Notez que **ARRAY** et **MAILADDR** sont des variables obligatoires.

2. Ouvrez le fichier de configuration **/etc/mdadm.conf** avec un éditeur de texte de votre choix et ajoutez la variable **MAILADDR** avec l'adresse de courrier électronique pour la notification. Par exemple, ajoutez une nouvelle ligne :

```
MAILADDR example@example.com>
```

   Ici, *example@example.com* est une adresse électronique à laquelle vous souhaitez recevoir les alertes de la surveillance de la matrice.

3. Enregistrez les modifications dans le fichier **/etc/mdadm.conf** et fermez-le.

Ressources supplémentaires

- La page de manuel **mdadm.conf(5)**

18.15. REMPLACEMENT D'UN DISQUE DÉFAILLANT DANS UN RAID

Vous pouvez reconstruire les données des disques défaillants en utilisant les disques restants. Le niveau RAID et le nombre total de disques déterminent le nombre minimum de disques restants nécessaires pour une reconstruction réussie des données.

Dans cette procédure, le RAID */dev/md0* contient quatre disques. Le disque */dev/sdd* est défaillant et vous devez le remplacer par le disque */dev/sdf*.

Conditions préalables

- Un disque de rechange pour le remplacement.
- Le paquet **mdadm** est installé.

Procédure

1. Vérifiez le disque défaillant :
   a. Consulter les journaux du noyau :
      -

```
# journalctl -k -f
```

b. Recherchez un message similaire au suivant :

```
md/raid:md0: Disk failure on sdd, disabling device.

md/raid:md0: Operation continuing on 3 devices.
```

c. Appuyer sur **Ctrl**+**C** pour quitter le programme **journalctl**.

2. Marquer le disque défaillant comme défectueux :

```
# mdadm --manage /dev/md0 --fail /dev/sdd
```

3. Facultatif : Vérifiez si le disque défaillant a été marqué correctement :

```
# mdadm --detail /dev/md0
```

À la fin de la sortie se trouve une liste de disques dans le RAID */dev/md0* où le disque */dev/sdd* a le statut **faulty**:

```
Number   Major   Minor   RaidDevice State
   0       8      16        0      active sync   /dev/sdb
   1       8      32        1      active sync   /dev/sdc
   -       0       0        2      removed
   3       8      64        3      active sync   /dev/sde

   2       8      48        -      faulty   /dev/sdd
```

4. Retirez le disque défaillant du RAID :

```
# mdadm --manage /dev/md0 --remove /dev/sdd
```

AVERTISSEMENT

Si votre RAID ne peut pas supporter une autre défaillance de disque, ne retirez aucun disque tant que le nouveau disque n'a pas l'état **active sync**. Vous pouvez surveiller la progression à l'aide de la commande **watch cat /proc/mdstat**.

5. Ajoutez le nouveau disque au RAID :

```
# mdadm --manage /dev/md0 --add /dev/sdf
```

Le RAID */dev/md0* comprend maintenant le nouveau disque */dev/sdf* et le service **mdadm** commencera automatiquement à copier les données des autres disques vers ce disque.

Vérification

- Vérifiez les détails du tableau :

```
# mdadm --detail /dev/md0
```

Si cette commande affiche une liste de disques dans le RAID */dev/md0* où le nouveau disque a le statut **spare rebuilding** à la fin de la sortie, des données sont toujours copiées sur ce disque à partir d'autres disques :

```
Number   Major   Minor   RaidDevice State
   0       8       16        0      active sync   /dev/sdb
   1       8       32        1      active sync   /dev/sdc
   4       8       80        2      spare rebuilding   /dev/sdf
   3       8       64        3      active sync   /dev/sde
```

Une fois la copie des données terminée, le nouveau disque a l'état **active sync**.

Ressources supplémentaires

- Mise en place de notifications par courrier électronique pour surveiller un RAID

18.16. RÉPARATION DES DISQUES RAID

Cette procédure décrit comment réparer les disques d'une matrice RAID.

Conditions préalables

- Le paquet **mdadm** est installé.

Procédure

1. Vérifiez le comportement des disques défaillants dans la matrice :

```
# echo check > /sys/block/md0/md/sync_action
```

Cette opération vérifie le tableau et le fichier **/sys/block/md0/md/sync_action** montre l'action de synchronisation.

2. Ouvrez le fichier **/sys/block/md0/md/sync_action** avec l'éditeur de texte de votre choix et voyez s'il y a un message sur les échecs de synchronisation des disques.

3. Consultez le fichier **/sys/block/md0/md/mismatch_cnt**. Si le paramètre **mismatch_cnt** n'est pas **0**, cela signifie que les disques RAID doivent être réparés.

4. Réparez les disques de la matrice :

```
# echo repair > /sys/block/md0/md/sync_action
```

Cette opération répare les disques de la matrice et écrit le résultat dans le fichier **/sys/block/md0/md/sync_action**.

5. Visualiser l'état d'avancement de la synchronisation :

```
# cat /sys/block/md0/md/sync_action
```

```
repair

# cat /proc/mdstat
Personalities : [raid0] [raid6] [raid5] [raid4] [raid1]
md0 : active raid1 sdg[1] dm-3[0]
      511040 blocks super 1.2 [2/2] [UU]
unused devices: <none>
```

CHAPITRE 19. CHIFFREMENT DES BLOCS DE DONNÉES À L'AIDE DE LUKS

Le chiffrement de disque permet de protéger les données d'un bloc en les chiffrant. Pour accéder au contenu décrypté du périphérique, il faut entrer une phrase de passe ou une clé en guise d'authentification. Cette fonction est importante pour les ordinateurs mobiles et les supports amovibles, car elle permet de protéger le contenu du périphérique même s'il a été physiquement retiré du système. Le format LUKS est une implémentation par défaut du chiffrement par bloc des périphériques dans Red Hat Enterprise Linux.

19.1. CRYPTAGE DE DISQUE LUKS

Linux Unified Key Setup-on-disk-format (LUKS) fournit un ensemble d'outils qui simplifie la gestion des périphériques cryptés. Avec LUKS, vous pouvez chiffrer des périphériques en bloc et permettre à plusieurs clés d'utilisateur de déchiffrer une clé principale. Pour le chiffrement en bloc de la partition, utilisez cette clé principale.

Red Hat Enterprise Linux utilise LUKS pour effectuer le chiffrement du périphérique de bloc. Par défaut, l'option de chiffrement du périphérique de bloc n'est pas cochée lors de l'installation. Si vous sélectionnez l'option de chiffrer votre disque, le système vous demande une phrase de passe à chaque fois que vous démarrez l'ordinateur. Cette phrase de passe déverrouille la clé de chiffrement en bloc qui décrypte votre partition. Si vous souhaitez modifier la table de partition par défaut, vous pouvez sélectionner les partitions que vous souhaitez crypter. Cette option est définie dans les paramètres de la table de partitions.

Chiffres

Le chiffrement par défaut utilisé pour LUKS est **aes-xts-plain64**. La taille de clé par défaut de LUKS est de 512 bits. La taille de clé par défaut pour LUKS avec le mode **Anaconda** XTS est de 512 bits. Les codes disponibles sont les suivants :

- Norme de chiffrement avancée (AES)
- Twofish
- Serpent

LUKS effectue les opérations suivantes

- LUKS crypte des blocs entiers et est donc bien adapté à la protection du contenu des appareils mobiles tels que les supports de stockage amovibles ou les lecteurs de disques d'ordinateurs portables.
- Le contenu sous-jacent du bloc crypté est arbitraire, ce qui le rend utile pour crypter les périphériques d'échange. Cela peut également être utile pour certaines bases de données qui utilisent des périphériques de bloc spécialement formatés pour le stockage des données.
- LUKS utilise le sous-système de noyau existant pour le mappage de périphériques.
- LUKS permet de renforcer la phrase de passe, ce qui protège contre les attaques par dictionnaire.
- Les dispositifs LUKS contiennent plusieurs emplacements de clé, ce qui permet aux utilisateurs d'ajouter des clés de sauvegarde ou des phrases de passe.

LUKS n'est pas recommandé dans les cas suivants

- Les solutions de chiffrement de disque telles que LUKS ne protègent les données que lorsque le système est éteint. Une fois que le système est en marche et que LUKS a décrypté le disque, les fichiers qui s'y trouvent sont accessibles à tous ceux qui y ont accès.
- Scénarios nécessitant que plusieurs utilisateurs disposent de clés d'accès distinctes au même appareil. Le format LUKS1 offre huit emplacements de clé et le format LUKS2 en offre jusqu'à 32.
- Applications nécessitant un cryptage au niveau du fichier.

Ressources supplémentaires

- Page d'accueil du projet LUKS
- Spécification du format LUKS sur disque
- FIPS 197 : norme de cryptage avancée (AES)

19.2. VERSIONS DE LUKS DANS RHEL

Dans Red Hat Enterprise Linux, le format par défaut pour le chiffrement LUKS est LUKS2. L'ancien format LUKS1 reste entièrement pris en charge et est fourni en tant que format compatible avec les versions antérieures de Red Hat Enterprise Linux. Le recryptage LUKS2 est considéré comme plus robuste et plus sûr que le recryptage LUKS1.

Le format LUKS2 permet des mises à jour futures des différentes parties sans qu'il soit nécessaire de modifier les structures binaires. En interne, il utilise le format de texte JSON pour les métadonnées, assure la redondance des métadonnées, détecte la corruption des métadonnées et répare automatiquement à partir d'une copie des métadonnées.

IMPORTANT

N'utilisez pas LUKS2 dans les systèmes qui ne prennent en charge que LUKS1. Red Hat Enterprise Linux 7 prend en charge le format LUKS2 depuis la version 7.6.

Depuis Red Hat Enterprise Linux 9.2, vous pouvez utiliser la commande **cryptsetup reencrypt** pour les deux versions de LUKS afin de chiffrer le disque.

Recryptage en ligne

Le format LUKS2 permet de recrypter les périphériques cryptés pendant qu'ils sont utilisés. Par exemple, il n'est pas nécessaire de démonter le système de fichiers sur le périphérique pour effectuer les tâches suivantes :

- Modifier la touche de volume
- Modifier l'algorithme de cryptage
  Lorsque vous cryptez un périphérique non crypté, vous devez toujours démonter le système de fichiers. Vous pouvez remonter le système de fichiers après une brève initialisation du chiffrement.

  Le format LUKS1 ne prend pas en charge le rechiffrement en ligne.

Conversion

Dans certaines situations, vous pouvez convertir LUKS1 en LUKS2. La conversion n'est pas possible dans les cas suivants :

- Un périphérique LUKS1 est marqué comme étant utilisé par une solution Clevis de décryptage basé sur des règles (PBD). L'outil **cryptsetup** ne convertit pas le périphérique lorsque certaines métadonnées **luksmeta** sont détectées.
- Un appareil est actif. L'appareil doit être dans un état inactif avant qu'une conversion ne soit possible.

19.3. OPTIONS DE PROTECTION DES DONNÉES PENDANT LE RECRYPTAGE LUKS2

LUKS2 propose plusieurs options qui donnent la priorité aux performances ou à la protection des données pendant le processus de recryptage. Il propose les modes suivants pour l'option **resilience**, et vous pouvez sélectionner n'importe lequel de ces modes à l'aide de la commande **cryptsetup reencrypt --resilience resilience-mode *<u>/dev/sdx</u>*** pour sélectionner l'un de ces modes :

checksum

Le mode par défaut. Il permet d'équilibrer la protection des données et les performances.
Ce mode stocke les sommes de contrôle individuelles des secteurs dans la zone de rechiffrement, que le processus de récupération peut détecter pour les secteurs qui ont été rechiffrés par LUKS2.
Ce mode exige que l'écriture du secteur du dispositif de bloc soit atomique.

journal

C'est le mode le plus sûr, mais aussi le plus lent. Étant donné que ce mode journalise la zone de rechiffrement dans la zone binaire, le LUKS2 écrit les données deux fois.

none

Le mode **none** donne la priorité aux performances et ne fournit aucune protection des données. Il protège les données uniquement en cas d'arrêt sûr du processus, comme le signal **SIGTERM** ou l'appui de l'utilisateur sur la touche **Ctrl**+**C** par l'utilisateur. Toute défaillance inattendue du système ou de l'application peut entraîner une corruption des données.

Si un processus de rechiffrement LUKS2 se termine inopinément par la force, LUKS2 peut effectuer la récupération de l'une des manières suivantes :

Automatiquement

L'exécution de l'une des actions suivantes déclenche l'action de récupération automatique lors de la prochaine action d'ouverture du périphérique LUKS2 :

- Exécution de la commande **cryptsetup open**.
- Attacher l'appareil avec la commande **systemd-cryptsetup**.

Manuellement

En utilisant la commande **cryptsetup repair *<u>/dev/sdx</u>*** sur le périphérique LUKS2.

Ressources supplémentaires

- **cryptsetup-reencrypt(8)** et **cryptsetup-repair(8)** pages de manuel

19.4. CHIFFREMENT DES DONNÉES EXISTANTES SUR UN DISPOSITIF DE BLOCAGE À L'AIDE DE LUKS2

Vous pouvez crypter les données existantes sur un dispositif non encore crypté en utilisant le format LUKS2. Un nouvel en-tête LUKS est stocké dans la tête du dispositif.

Conditions préalables

- L'unité de bloc dispose d'un système de fichiers.
- Vous avez sauvegardé vos données.

AVERTISSEMENT

Vous pouvez perdre vos données au cours du processus de cryptage en raison d'une défaillance matérielle, du noyau ou d'une défaillance humaine. Assurez-vous de disposer d'une sauvegarde fiable avant de commencer à chiffrer les données.

Procédure

1. Démontez tous les systèmes de fichiers sur le périphérique que vous prévoyez de chiffrer, par exemple :

```
# umount /dev/mapper/vg00-lv00
```

2. Libérez de l'espace pour stocker un en-tête LUKS. Utilisez l'une des options suivantes en fonction de votre scénario :
   - Dans le cas du chiffrement d'un volume logique, vous pouvez étendre le volume logique sans redimensionner le système de fichiers. Par exemple, il est possible d'étendre un volume logique sans redimensionner le système de fichiers :

```
# lvextend -L 32M /dev/mapper/vg00-lv00
```

   - Étendez la partition en utilisant des outils de gestion de partition, tels que **parted**.
   - Réduisez le système de fichiers sur le périphérique. Vous pouvez utiliser l'utilitaire **resize2fs** pour les systèmes de fichiers ext2, ext3 ou ext4. Notez que vous ne pouvez pas réduire le système de fichiers XFS.

3. Initialiser le cryptage :

```
# cryptsetup reencrypt --encrypt --init-only --reduce-device-size 32M /dev/mapper/vg00-lv00
lv00_encrypted

/dev/mapper/lv00_encrypted is now active and ready for online encryption.
```

4. Monter l'appareil :

```
# mount /dev/mapper/lv00_encrypted /mnt/lv00_encrypted
```

5. Ajouter une entrée pour un mappage persistant au fichier **/etc/crypttab**:

   a. Trouver le site **luksUUID**:

   ```
   # cryptsetup luksUUID /dev/mapper/vg00-lv00

   a52e2cc9-a5be-47b8-a95d-6bdf4f2d9325
   ```

   b. Ouvrez **/etc/crypttab** dans un éditeur de texte de votre choix et ajoutez un dispositif dans ce fichier :

   ```
   $ vi /etc/crypttab

   lv00_encrypted UUID=a52e2cc9-a5be-47b8-a95d-6bdf4f2d9325 none
   ```

   Remplacez *a52e2cc9-a5be-47b8-a95d-6bdf4f2d9325* par le site **luksUUID** de votre appareil.

   c. Rafraîchir les initramfs avec **dracut**:

   ```
   $ dracut -f --regenerate-all
   ```

6. Ajouter une entrée pour un montage persistant au fichier **/etc/fstab**:

   a. Recherchez l'UUID du système de fichiers du périphérique bloc LUKS actif :

   ```
   $ blkid -p /dev/mapper/lv00_encrypted

   /dev/mapper/lv00-encrypted: UUID="37bc2492-d8fa-4969-9d9b-bb64d3685aa9"
   BLOCK_SIZE="4096" TYPE="xfs" USAGE="filesystem"
   ```

   b. Ouvrez **/etc/fstab** dans un éditeur de texte de votre choix et ajoutez un dispositif dans ce fichier, par exemple :

   ```
   $ vi /etc/fstab

   UUID=37bc2492-d8fa-4969-9d9b-bb64d3685aa9 /home auto rw,user,auto 0
   ```

   Remplacez *37bc2492-d8fa-4969-9d9b-bb64d3685aa9* par l'UUID de votre système de fichiers.

7. Reprendre le cryptage en ligne :

```
# cryptsetup reencrypt --resume-only /dev/mapper/vg00-lv00

Enter passphrase for /dev/mapper/vg00-lv00:
Auto-detected active dm device 'lv00_encrypted' for data device /dev/mapper/vg00-lv00.
Finished, time 00:31.130, 10272 MiB written, speed 330.0 MiB/s
```

Vérification

1. Vérifier si les données existantes ont été cryptées :

```
# cryptsetup luksDump /dev/mapper/vg00-lv00

LUKS header information
Version: 2
Epoch: 4
Metadata area: 16384 [bytes]
Keyslots area: 16744448 [bytes]
UUID: a52e2cc9-a5be-47b8-a95d-6bdf4f2d9325
Label: (no label)
Subsystem: (no subsystem)
Flags: (no flags)

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

2. Affichez l'état du dispositif de bloc vierge crypté :

```
# cryptsetup status lv00_encrypted

/dev/mapper/lv00_encrypted is active and is in use.
  type:    LUKS2
  cipher:  aes-xts-plain64
  keysize: 512 bits
  key location: keyring
  device:  /dev/mapper/vg00-lv00
```

Ressources supplémentaires

- **cryptsetup(8)**, **cryptsetup-reencrypt(8)**, **lvextend(8)**, **resize2fs(8)**, et **parted(8)** pages de manuel

19.5. CHIFFREMENT DES DONNÉES EXISTANTES SUR UN PÉRIPHÉRIQUE DE BLOC À L'AIDE DE LUKS2 AVEC UN EN-TÊTE DÉTACHÉ

Vous pouvez chiffrer des données existantes sur un bloc sans créer d'espace libre pour le stockage d'un en-tête LUKS. L'en-tête est stocké dans un emplacement détaché, ce qui constitue également une couche de sécurité supplémentaire. La procédure utilise le format de cryptage LUKS2.

Conditions préalables

- L'unité de bloc dispose d'un système de fichiers.
- Vous avez sauvegardé vos données.

AVERTISSEMENT

Vous pouvez perdre vos données au cours du processus de cryptage en raison d'une défaillance matérielle, du noyau ou d'une défaillance humaine. Assurez-vous de disposer d'une sauvegarde fiable avant de commencer à chiffrer les données.

Procédure

1. Démonter tous les systèmes de fichiers sur l'appareil, par exemple :

```
# umount /dev/nvme0n1p1
```

2. Initialiser le cryptage :

```
# cryptsetup reencrypt --encrypt --init-only --header /home/header /dev/nvme0n1p1
nvme_encrypted

WARNING!
========
Header file does not exist, do you want to create it?

Are you sure? (Type 'yes' in capital letters): YES
Enter passphrase for /home/header:
Verify passphrase:
/dev/mapper/nvme_encrypted is now active and ready for online encryption.
```

Remplacez */home/header* par un chemin d'accès au fichier contenant un en-tête LUKS détaché. L'en-tête LUKS détaché doit être accessible pour déverrouiller ultérieurement le dispositif crypté.

3. Monter l'appareil :

```
# mount /dev/mapper/nvme_encrypted /mnt/nvme_encrypted
```

4. Reprendre le cryptage en ligne :

```
# cryptsetup reencrypt --resume-only --header /home/header /dev/nvme0n1p1

Enter passphrase for /dev/nvme0n1p1:
Auto-detected active dm device 'nvme_encrypted' for data device /dev/nvme0n1p1.
Finished, time 00m51s,   10 GiB written, speed 198.2 MiB/s
```

Vérification

1. Vérifiez si les données existantes sur un périphérique de bloc utilisant LUKS2 avec un en-tête détaché sont chiffrées :

```
# cryptsetup luksDump /home/header
```

```
LUKS header information
Version:        2
Epoch:          88
Metadata area:  16384 [bytes]
Keyslots area:  16744448 [bytes]
UUID:           c4f5d274-f4c0-41e3-ac36-22a917ab0386
Label:          (no label)
Subsystem:      (no subsystem)
Flags:        (no flags)

Data segments:
  0: crypt
 offset: 0 [bytes]
 length: (whole device)
 cipher: aes-xts-plain64
 sector: 512 [bytes]
[...]
```

2. Affichez l'état du dispositif de bloc vierge crypté :

```
# cryptsetup status nvme_encrypted

/dev/mapper/nvme_encrypted is active and is in use.
  type:    LUKS2
  cipher:  aes-xts-plain64
  keysize: 512 bits
  key location: keyring
  device:  /dev/nvme0n1p1
```

Ressources supplémentaires

- **cryptsetup(8)** et **cryptsetup-reencrypt(8)** pages de manuel

19.6. CHIFFREMENT D'UN BLOC VIERGE À L'AIDE DE LUKS2

Vous pouvez crypter un périphérique bloc vierge, que vous pouvez utiliser pour un stockage crypté à l'aide du format LUKS2.

Conditions préalables

- Un périphérique bloc vide. Vous pouvez utiliser des commandes telles que **lsblk** pour savoir s'il n'y a pas de données réelles sur ce périphérique, par exemple, un système de fichiers.

Procédure

1. Configurer une partition en tant que partition LUKS chiffrée :

```
# cryptsetup luksFormat /dev/nvme0n1p1

WARNING!
========
This will overwrite data on /dev/nvme0n1p1 irrevocably.
```

```
Are you sure? (Type 'yes' in capital letters): YES
Enter passphrase for /dev/nvme0n1p1:
Verify passphrase:
```

2. Ouvrez une partition LUKS chiffrée :

```
# cryptsetup open dev/nvme0n1p1 nvme0n1p1_encrypted

Enter passphrase for /dev/nvme0n1p1:
```

Cette commande déverrouille la partition et l'affecte à un nouveau périphérique en utilisant le mappeur de périphérique. Pour ne pas écraser les données chiffrées, cette commande avertit le noyau que le périphérique est un périphérique chiffré et qu'il est adressé par LUKS à l'aide de l'attribut **/dev/mapper/*device_mapped_name*** chemin.

3. Créez un système de fichiers pour écrire des données cryptées sur la partition, à laquelle il faut accéder par le nom mappé du périphérique :

```
# mkfs -t ext4 /dev/mapper/nvme0n1p1_encrypted
```

4. Monter l'appareil :

```
# mount /dev/mapper/nvme0n1p1_encrypted mount-point
```

Vérification

1. Vérifiez si le périphérique de blocage vierge est crypté :

```
# cryptsetup luksDump /dev/nvme0n1p1

LUKS header information
Version:        2
Epoch:          3
Metadata area:  16384 [bytes]
Keyslots area:  16744448 [bytes]
UUID:           34ce4870-ffdf-467c-9a9e-345a53ed8a25
Label:          (no label)
Subsystem:      (no subsystem)
Flags:          (no flags)

Data segments:
  0: crypt
 offset: 16777216 [bytes]
 length: (whole device)
 cipher: aes-xts-plain64
 sector: 512 [bytes]
[...]
```

2. Affichez l'état du dispositif de bloc vierge crypté :

```
# cryptsetup status nvme0n1p1_encrypted

/dev/mapper/nvme0n1p1_encrypted is active and is in use.
  type:    LUKS2
```

```
cipher:  aes-xts-plain64
keysize: 512 bits
key location: keyring
device:  /dev/nvme0n1p1
sector size:  512
offset:  32768 sectors
size:    20938752 sectors
mode:    read/write
```

Ressources supplémentaires

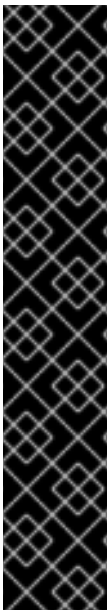
- **cryptsetup(8)**, **cryptsetup-open (8)**, et **cryptsetup-lusFormat(8)** pages de manuel

19.7. 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

CHAPITRE 20. UTILISATION DE LA MÉMOIRE PERSISTANTE NVDIMM

Vous pouvez activer et gérer différents types de stockage sur les modules de mémoire non volatile double en ligne (NVDIMM) connectés à votre système.

Pour installer Red Hat Enterprise Linux 9 sur un stockage NVDIMM, reportez-vous à la section Installation sur un périphérique NVDIMM à la place.

20.1. LA TECHNOLOGIE DE MÉMOIRE PERSISTANTE NVDIMM

Modules de mémoire double en ligne non volatile (NVDIMM) La mémoire persistante, également appelée mémoire de stockage ou **pmem**, est une combinaison de mémoire et de stockage.

Les NVDIMM combinent la durabilité du stockage avec la faible latence d'accès et la grande largeur de bande de la RAM dynamique (DRAM). Les autres avantages de l'utilisation des NVDIMM sont les suivants :

- Le stockage NVDIMM est adressable par octet, ce qui signifie qu'il est possible d'y accéder en utilisant les instructions de chargement et de stockage du CPU. Outre les appels système read() et write(), qui sont nécessaires pour accéder au stockage traditionnel par blocs, les NVDIMM prennent également en charge le chargement direct et un modèle de programmation store.
- Les caractéristiques de performance des NVDIMM sont similaires à celles de la DRAM, avec une latence d'accès très faible, typiquement de l'ordre de quelques dizaines à quelques centaines de nanosecondes.
- Les données stockées dans les NVDIMM sont conservées lorsque l'alimentation est coupée, comme dans le cas d'une mémoire persistante.
- Grâce à la technologie d'accès direct (DAX), les applications peuvent accéder directement au stockage de la carte mémoire sans passer par le cache de page du système. Cela libère de la DRAM pour d'autres usages.

Les NVDIMM sont utiles dans des cas d'utilisation tels que :

Bases de données

La réduction de la latence d'accès au stockage sur les NVDIMM améliore les performances des bases de données.

Redémarrage rapide

Le redémarrage rapide est également appelé effet de cache chaud. Par exemple, un serveur de fichiers n'a aucun contenu de fichier en mémoire après le démarrage. Au fur et à mesure que les clients se connectent et lisent ou écrivent des données, celles-ci sont mises en cache dans le cache des pages. Au final, le cache contient principalement des données chaudes. Après un redémarrage, le système doit recommencer le processus sur le stockage traditionnel.
Avec les NVDIMM, il est possible pour une application de conserver le cache chaud lors des redémarrages si l'application est conçue correctement. Dans cet exemple, il n'y aurait pas de cache de page : l'application mettrait les données en cache directement dans la mémoire persistante.

Cache d'écriture rapide

Souvent, les serveurs de fichiers n'accusent pas réception d'une demande d'écriture d'un client tant que les données ne sont pas sur un support durable. L'utilisation de NVDIMM en tant que cache

d'écriture rapide permet au serveur de fichiers d'accuser réception de la demande d'écriture rapidement, ce qui se traduit par une faible latence.

20.2. ENTRELACEMENT DES NVDIMM ET RÉGIONS

Les modules de mémoire double en ligne non volatile (NVDIMM) prennent en charge le regroupement en régions entrelacées.

Les dispositifs NVDIMM peuvent être regroupés en ensembles d'entrelacement de la même manière que la RAM dynamique ordinaire (DRAM). Un jeu d'entrelacement est similaire à une configuration de niveau RAID 0 (bande) sur plusieurs modules DIMM. Un jeu d'entrelacement est également appelé région.

L'entrelacement présente les avantages suivants :

- Les dispositifs NVDIMM bénéficient de performances accrues lorsqu'ils sont configurés en ensembles entrelacés.
- L'entrelacement permet de combiner plusieurs dispositifs NVDIMM plus petits en un dispositif logique plus grand.

Les jeux d'entrelacement NVDIMM sont configurés dans le BIOS du système ou dans le microprogramme UEFI. Red Hat Enterprise Linux crée un périphérique régional pour chaque jeu d'entrelacement.

20.3. ESPACES DE NOMS NVDIMM

Les régions des modules de mémoire double en ligne non volatile (NVDIMM) peuvent être divisées en un ou plusieurs espaces de noms en fonction de la taille de la zone d'étiquetage. En utilisant les espaces de noms, vous pouvez accéder au périphérique à l'aide de différentes méthodes, basées sur les modes d'accès de l'espace de noms tels que **sector**, **fsdax**, **devdax** et **raw**. Pour plus d'informations, consultez les modes d'accès aux NVDIMM.

Certains dispositifs NVDIMM ne prennent pas en charge plusieurs espaces de noms sur une région :

- Si votre périphérique NVDIMM prend en charge les étiquettes, vous pouvez subdiviser la région en espaces de noms.
- Si votre périphérique NVDIMM ne prend pas en charge les étiquettes, la région ne peut contenir qu'un seul espace de noms. Dans ce cas, Red Hat Enterprise Linux crée un espace de noms par défaut qui couvre l'ensemble de la région.

20.4. MODES D'ACCÈS AUX NVDIMM

Vous pouvez configurer les espaces de noms NVDIMM (Non-Volatile Dual In-line Memory Modules) pour qu'ils utilisent l'un des modes suivants :

sector

Présente le stockage comme un périphérique bloc rapide. Ce mode est utile pour les applications anciennes qui n'ont pas été modifiées pour utiliser le stockage NVDIMM, ou pour les applications qui utilisent la pile d'E/S complète, y compris Device Mapper.
Un périphérique **sector** peut être utilisé de la même manière que n'importe quel autre périphérique bloc sur le système. Vous pouvez y créer des partitions ou des systèmes de fichiers, le configurer dans le cadre d'un ensemble RAID logiciel ou l'utiliser comme périphérique de cache pour **dm-cache**.

Les appareils dans ce mode sont disponibles en tant que /**dev**/**pmem*N*s**. Après avoir créé l'espace de noms, consultez la liste des valeurs de **blockdev**.

devdaxou accès direct au dispositif (DAX)

Avec **devdax**, les dispositifs NVDIMM prennent en charge la programmation par accès direct, comme décrit dans la spécification du modèle de programmation de la mémoire non volatile (NVM) de la Storage Networking Industry Association (SNIA). Dans ce mode, les E/S contournent la pile de stockage du noyau. Par conséquent, aucun pilote Device Mapper ne peut être utilisé.
Device DAX fournit un accès brut au stockage NVDIMM en utilisant un nœud de périphérique à caractère DAX. Les données d'un périphérique **devdax** peuvent être pérennisées à l'aide d'instructions de vidage de cache et de clôture du CPU. Certaines bases de données et certains hyperviseurs de machines virtuelles peuvent bénéficier de ce mode. Les systèmes de fichiers ne peuvent pas être créés sur les périphériques **devdax**.

Les appareils dans ce mode sont disponibles en tant que /**dev**/**dax*N.M***. Après avoir créé l'espace de noms, consultez la liste des valeurs de **chardev**.

fsdaxou accès direct au système de fichiers (DAX)

Avec **fsdax**, les périphériques NVDIMM prennent en charge la programmation par accès direct, comme décrit dans la spécification du modèle de programmation de la mémoire non volatile (NVM) de la Storage Networking Industry Association (SNIA). Dans ce mode, les E/S contournent la pile de stockage du noyau et de nombreux pilotes Device Mapper ne peuvent donc pas être utilisés.
Vous pouvez créer des systèmes de fichiers sur des dispositifs DAX de système de fichiers.

Les appareils dans ce mode sont disponibles en tant que /**dev**/**pmem*N***. Après avoir créé l'espace de noms, consultez la liste des valeurs de **blockdev**.

IMPORTANT

La technologie DAX du système de fichiers est fournie uniquement en tant qu'aperçu technologique et n'est pas prise en charge par Red Hat.

raw

Présente un disque mémoire qui ne prend pas en charge DAX. Dans ce mode, les espaces de noms ont plusieurs limitations et ne doivent pas être utilisés.
Les appareils dans ce mode sont disponibles en tant que /**dev**/**pmem*N***. Après avoir créé l'espace de noms, consultez la liste des valeurs de **blockdev**.

20.5. INSTALLATION DE NDCTL

Vous pouvez installer l'utilitaire **ndctl** pour configurer et surveiller les modules de mémoire non volatile double en ligne (NVDIMM).

Procédure

- Installez l'utilitaire **ndctl**:

```
# dnf install ndctl
```

20.6. CRÉATION D'UN ESPACE DE NOMS DE SECTEUR SUR UN NVDIMM POUR AGIR EN TANT QUE PÉRIPHÉRIQUE DE BLOC

Vous pouvez configurer un périphérique NVDIMM (Non-Volatile Dual In-line Memory Modules) en mode sectoriel, également appelé mode hérité, pour prendre en charge le stockage traditionnel en mode bloc.

Vous pouvez soit

- reconfigurer un espace de noms existant en mode sectoriel, ou
- créer un nouvel espace de noms sectoriel s'il y a de l'espace disponible.

Conditions préalables

- Un périphérique NVDIMM est connecté à votre système.

20.6.1. Reconfiguration d'un espace de noms NVDIMM existant en mode secteur

Vous pouvez reconfigurer un espace de noms NVDIMM (Non-Volatile Dual In-line Memory Modules) en mode secteur pour l'utiliser en tant que périphérique de bloc rapide.

AVERTISSEMENT

La reconfiguration d'un espace de noms supprime les données précédemment stockées dans l'espace de noms.

Conditions préalables

- L'utilitaire **ndctl** est installé. Pour plus d'informations, voir Installation de ndctl.

Procédure

1. Afficher les espaces de noms existants :

```
# ndctl list --namespaces --idle
[
  {
    "dev":"namespace1.0",
    "mode":"raw",
    "size":34359738368,
    "state":"disabled",
    "numa_node":1
  },
  {
    "dev":"namespace0.0",
    "mode":"raw",
    "size":34359738368,
    "state":"disabled",
```

```
    "numa_node":0
  }
]
```

2. Reconfigurer l'espace de noms sélectionné en mode secteur :

```
# ndctl create-namespace --force --reconfig=namespace-ID --mode=sector
```

Exemple 20.1. Reconfiguration de l'espace de noms 1.0 en mode secteur

```
# ndctl create-namespace --force --reconfig=namespace1.0 --mode=sector
{
  "dev":"namespace1.0",
  "mode":"sector",
  "size":"755.26 GiB (810.95 GB)",
  "uuid":"2509949d-1dc4-4ee0-925a-4542b28aa616",
  "sector_size":4096,
  "blockdev":"pmem1s"
}
```

L'espace de noms reconfiguré est désormais disponible dans le répertoire **/dev** sous la forme du fichier **/dev/pmem1s**.

Vérification

- Vérifiez si l'espace de noms existant sur votre système est reconfiguré :

```
# ndctl list --namespace namespace1.0
[
  {
    "dev":"namespace1.0",
    "mode":"sector",
    "size":810954706944,
    "uuid":"2509949d-1dc4-4ee0-925a-4542b28aa616",
    "sector_size":4096,
    "blockdev":"pmem1s"
  }
]
```

Ressources supplémentaires

- La page de manuel **ndctl-create-namespace(1)**

20.6.2. Création d'un nouvel espace de noms NVDIMM en mode secteur

Vous pouvez créer un espace de noms NVDIMM (Non-Volatile Dual In-line Memory Modules) en mode secteur pour l'utiliser comme périphérique de bloc rapide s'il y a de l'espace disponible dans la région.

Conditions préalables

- L'utilitaire **ndctl** est installé. Pour plus d'informations, voir Installation de ndctl.

- Le périphérique NVDIMM prend en charge les étiquettes permettant de créer plusieurs espaces de noms dans une région. Vous pouvez le vérifier à l'aide de la commande suivante :

```
# ndctl read-labels nmem0 >/dev/null
  read 1 nmem
```

Cela indique qu'il a lu l'étiquette d'un périphérique NVDIMM. Si la valeur est **0**, cela signifie que votre périphérique ne prend pas en charge les étiquettes.

Procédure

1. Dressez la liste des régions **pmem** de votre système qui ont de l'espace disponible. Dans l'exemple suivant, de l'espace est disponible dans les régions *region1* et *region0*:

```
# ndctl list --regions
[
  {
    "dev":"region1",
    "size":2156073582592,
    "align":16777216,
    "available_size":2117418876928,
    "max_available_extent":2117418876928,
    "type":"pmem",
    "iset_id":-9102197055295954944,
    "badblock_count":1,
    "persistence_domain":"memory_controller"
  },
  {
    "dev":"region0",
    "size":2156073582592,
    "align":16777216,
    "available_size":2143188680704,
    "max_available_extent":2143188680704,
    "type":"pmem",
    "iset_id":736272362787276936,
    "badblock_count":3,
    "persistence_domain":"memory_controller"
  }
]
```

2. Attribuer un ou plusieurs espaces de noms sur l'une des régions disponibles :

```
# ndctl create-namespace --mode=sector --region=regionN --size=namespace-size
```

Exemple 20.2. Création d'un espace de noms de secteur de 36 Go sur region0

```
# ndctl create-namespace --mode=sector --region=region0 --size=36G
{
  "dev":"namespace0.1",
  "mode":"sector",
  "size":"35.96 GiB (38.62 GB)",
  "uuid":"ff5a0a16-3495-4ce8-b86b-f0e3bd9d1817",
  "sector_size":4096,
  "blockdev":"pmem0.1s"
}
```

Le nouvel espace de noms est désormais disponible à l'adresse **/dev/pmem0.1s**.

Vérification

- Vérifier si le nouvel espace de noms est créé en mode secteur :

```
# ndctl list -RN -n namespace0.1
{
  "regions":[
    {
      "dev":"region0",
      "size":2156073582592,
      "align":16777216,
      "available_size":2104533975040,
      "max_available_extent":2104533975040,
      "type":"pmem",
      "iset_id":736272362787276936,
      "badblock_count":3,
      "persistence_domain":"memory_controller",
      "namespaces":[
        {
          "dev":"namespace0.1",
          "mode":"sector",
          "size":38615912448,
          "uuid":"ff5a0a16-3495-4ce8-b86b-f0e3bd9d1817",
          "sector_size":4096,
          "blockdev":"pmem0.1s"
        }
      ]
    }
  ]
}
```

Ressources supplémentaires

- La page de manuel **ndctl-create-namespace(1)**

20.7. CRÉATION D'UN ESPACE DE NOMS DAX SUR UN NVDIMM

Configurez le périphérique NVDIMM connecté à votre système en mode DAX pour prendre en charge le stockage de caractères avec des capacités d'accès direct.

Envisagez les options suivantes :

- Reconfiguration d'un espace de noms existant en mode DAX.
- Création d'un nouvel espace de noms DAX pour les appareils, s'il y a de la place disponible.

20.7.1. NVDIMM en mode d'accès direct au périphérique

L'accès direct aux périphériques (device DAX, **devdax**) permet aux applications d'accéder directement au stockage, sans passer par un système de fichiers. L'avantage de device DAX est qu'il fournit une granularité de faute garantie, qui peut être configurée en utilisant l'option **--align** de l'utilitaire **ndctl**.

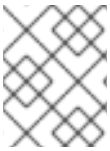
Pour les architectures Intel 64 et AMD64, les granularités de défaut suivantes sont prises en charge :

- 4 KiB
- 2 MiB
- 1 GiB

Les nœuds DAX ne prennent en charge que les appels système suivants :

- **open()**
- **close()**
- **mmap()**

Vous pouvez afficher les alignements pris en charge pour votre périphérique NVDIMM à l'aide de la commande **ndctl list --human --capabilities**. Par exemple, pour les visualiser pour le périphérique *region0*, utilisez la commande **ndctl list --human --capabilities -r region0**.



NOTE

Les appels système **read()** et **write()** ne sont pas pris en charge, car le cas d'utilisation du dispositif DAX est lié au modèle de programmation de la mémoire non volatile de la SNIA.

20.7.2. Reconfiguration d'un espace de noms NVDIMM existant en mode DAX de périphérique

Vous pouvez reconfigurer un espace de noms NVDIMM (Non-Volatile Dual In-line Memory Modules) existant en mode DAX.

AVERTISSEMENT

La reconfiguration d'un espace de noms supprime les données précédemment stockées dans l'espace de noms.

Conditions préalables

- L'utilitaire **ndctl** est installé. Pour plus d'informations, voir Installation de ndctl.

Procédure

1. Liste de tous les espaces de noms de votre système :

```
# ndctl list --namespaces --idle
[
```

```
  {
    "dev":"namespace1.0",
    "mode":"raw",
    "size":34359738368,
    "uuid":"ac951312-b312-4e76-9f15-6e00c8f2e6f4"
    "state":"disabled",
    "numa_node":1
  },
  {
    "dev":"namespace0.0",
    "mode":"raw",
    "size":38615912448,
    "uuid":"ff5a0a16-3495-4ce8-b86b-f0e3bd9d1817",
    "state":"disabled",
    "numa_node":0
  }
]
```

2. Reconfigurer tout espace de noms :

```
# ndctl create-namespace --force --mode=devdax --reconfig=namespace-ID
```

Exemple 20.3. Reconfiguration d'un espace de noms en tant que dispositif DAX

La commande suivante reconfigure **namespace0.1** pour le stockage de données prenant en charge DAX. Elle est alignée sur une granularité de panne de 2 Mo afin de garantir que le système d'exploitation effectue des pannes sur des pages de 2 Mo à la fois :

```
# ndctl create-namespace --force --mode=devdax  --align=2M --reconfig=namespace0.1
{
  "dev":"namespace0.1",
  "mode":"devdax",
  "map":"dev",
  "size":"35.44 GiB (38.05 GB)",
  "uuid":"426d6a52-df92-43d2-8cc7-046241d6d761",
  "daxregion":{
    "id":0,
    "size":"35.44 GiB (38.05 GB)",
    "align":2097152,
    "devices":[
      {
        "chardev":"dax0.1",
        "size":"35.44 GiB (38.05 GB)",
        "target_node":4,
        "mode":"devdax"
      }
    ]
  },
  "align":2097152
}
```

L'espace de noms est désormais disponible sur le site **/dev/dax0.1**.

Vérification

- Vérifiez si les espaces de noms existants sur votre système sont reconfigurés :

```
# ndctl list --namespace namespace0.1
[
  {
    "dev":"namespace0.1",
    "mode":"devdax",
    "map":"dev",
    "size":38048628736,
    "uuid":"426d6a52-df92-43d2-8cc7-046241d6d761",
    "chardev":"dax0.1",
    "align":2097152
  }
]
```

Ressources supplémentaires

- La page de manuel **ndctl-create-namespace(1)**

20.7.3. Création d'un nouvel espace de noms NVDIMM en mode device DAX

Vous pouvez créer un nouvel espace de noms DAX sur un périphérique NVDIMM (Non-Volatile Dual In-line Memory Modules) s'il y a de l'espace disponible dans la région.

Conditions préalables

- L'utilitaire **ndctl** est installé. Pour plus d'informations, voir Installation de ndctl.
- Le périphérique NVDIMM prend en charge les étiquettes permettant de créer plusieurs espaces de noms dans une région. Vous pouvez le vérifier à l'aide de la commande suivante :

```
# ndctl read-labels nmem0 >/dev/null
read 1 nmem
```

Cela indique qu'il a lu l'étiquette d'un périphérique NVDIMM. Si la valeur est **0**, cela signifie que votre périphérique ne prend pas en charge les étiquettes.

Procédure

1. Dressez la liste des régions **pmem** de votre système qui ont de l'espace disponible. Dans l'exemple suivant, de l'espace est disponible dans les régions *region1* et *region0*:

```
# ndctl list --regions
[
  {
    "dev":"region1",
    "size":2156073582592,
    "align":16777216,
    "available_size":2117418876928,
    "max_available_extent":2117418876928,
    "type":"pmem",
    "iset_id":-9102197055295954944,
```

```
    "badblock_count":1,
    "persistence_domain":"memory_controller"
  },
  {
    "dev":"region0",
    "size":2156073582592,
    "align":16777216,
    "available_size":2143188680704,
    "max_available_extent":2143188680704,
    "type":"pmem",
    "iset_id":736272362787276936,
    "badblock_count":3,
    "persistence_domain":"memory_controller"
  }
]
```

2. Attribuer un ou plusieurs espaces de noms sur l'une des régions disponibles :

```
# ndctl create-namespace --mode=devdax --region=region_N_ --size=namespace-size
```

Exemple 20.4. Création d'un espace de noms sur une région

La commande suivante crée un espace de noms DAX de 36 Go sur region0. Il est aligné sur une granularité de panne de 2 Mo afin de garantir que le système d'exploitation effectue des pannes sur des pages de 2 Mo à la fois :

```
# ndctl create-namespace --mode=devdax --region=region0 --align=2M --size=36G
{
  "dev":"namespace0.2",
  "mode":"devdax",
  "map":"dev",
  "size":"35.44 GiB (38.05 GB)",
  "uuid":"89d13f41-be6c-425b-9ec7-1e2a239b5303",
  "daxregion":{
    "id":0,
    "size":"35.44 GiB (38.05 GB)",
    "align":2097152,
    "devices":[
      {
        "chardev":"dax0.2",
        "size":"35.44 GiB (38.05 GB)",
        "target_node":4,
        "mode":"devdax"
      }
    ]
  },
  "align":2097152
}
```

L'espace de noms est désormais disponible à l'adresse **/dev/dax0.2**.

Vérification

- Vérifier si le nouvel espace de noms est créé en mode secteur :

```
# ndctl list -RN -n namespace0.2
{
  "regions":[
    {
      "dev":"region0",
      "size":2156073582592,
      "align":16777216,
      "available_size":2065879269376,
      "max_available_extent":2065879269376,
      "type":"pmem",
      "iset_id":736272362787276936,
      "badblock_count":3,
      "persistence_domain":"memory_controller",
      "namespaces":[
        {
          "dev":"namespace0.2",
          "mode":"devdax",
          "map":"dev",
          "size":38048628736,
          "uuid":"89d13f41-be6c-425b-9ec7-1e2a239b5303",
          "chardev":"dax0.2",
          "align":2097152
        }
      ]
    }
  ]
}
```

Ressources supplémentaires

- La page de manuel **ndctl-create-namespace(1)**

20.8. CRÉATION D'UN ESPACE DE NOMS DAX DE SYSTÈME DE FICHIERS SUR UN NVDIMM

Configurez un périphérique NVDIMM attaché à votre système, en mode système de fichiers DAX pour prendre en charge un système de fichiers avec des capacités d'accès direct.

Envisagez les options suivantes :

- Reconfiguration d'un espace de noms existant en mode DAX du système de fichiers.
- Création d'un nouvel espace de noms DAX du système de fichiers s'il y a de l'espace disponible.

IMPORTANT

La technologie DAX du système de fichiers est fournie uniquement en tant qu'aperçu technologique et n'est pas prise en charge par Red Hat.

20.8.1. NVDIMM en mode d'accès direct au système de fichiers

Lorsqu'un périphérique NVDIMM est configuré en mode d'accès direct au système de fichiers (système

de fichiers DAX, **fsdax**), vous pouvez créer un système de fichiers au-dessus de lui. Toute application qui effectue une opération **mmap()** sur un fichier de ce système de fichiers obtient un accès direct à son stockage. Cela permet d'utiliser le modèle de programmation à accès direct sur les NVDIMM.

À partir de Red Hat Enterprise Linux 8, les nouvelles options **-o dax** suivantes sont désormais disponibles, et le comportement d'accès direct peut être contrôlé via un attribut de fichier si nécessaire :

-o dax=inode

Il s'agit de l'option par défaut lorsque vous ne spécifiez aucune option dax lors du montage d'un système de fichiers. Cette option vous permet de définir un drapeau d'attribut sur les fichiers afin de contrôler si le mode dax peut être activé. Si nécessaire, vous pouvez définir cet indicateur sur des fichiers individuels.
Vous pouvez également définir cet attribut pour un répertoire et tous les fichiers de ce répertoire seront créés avec le même attribut. Vous pouvez définir cet attribut en utilisant la commande **xfs_io -c 'chattr x'** nom-du-répertoire.

-o dax=never

Avec cette option, le mode dax ne sera pas activé même si l'indicateur dax est défini sur un mode **inode**. Cela signifie que l'attribut dax par nœud est ignoré et que les fichiers définis avec cet attribut ne seront jamais activés pour l'accès direct.

-o dax=always

Cette option est équivalente à l'ancien comportement de **-o dax**. Avec cette option, vous pouvez activer le mode d'accès direct pour n'importe quel fichier du système de fichiers, quel que soit le drapeau d'attribut dax.

AVERTISSEMENT

Dans les versions ultérieures, il se peut que **-o dax** ne soit pas pris en charge et, le cas échéant, vous pouvez utiliser **-o dax=always** à la place. Dans ce mode, chaque fichier peut être en accès direct.

Attribution de métadonnées par page

Ce mode nécessite l'allocation de métadonnées par page dans la DRAM du système ou sur le périphérique NVDIMM lui-même. La surcharge de cette structure de données est de 64 octets par page de 4 Ko :

- Sur les petits appareils, la quantité de frais généraux est suffisamment faible pour être intégrée sans problème dans la DRAM. Par exemple, un espace de noms de 16 Go ne nécessite que 256 Mo pour les structures de pages. Étant donné que les dispositifs NVDIMM sont généralement petits et coûteux, il est préférable de stocker les structures de données de suivi des pages dans la DRAM.

- Sur les dispositifs NVDIMM de plusieurs téraoctets ou plus, la quantité de mémoire nécessaire pour stocker les structures de données de suivi des pages peut dépasser la quantité de DRAM du système. Un TiB de NVDIMM nécessite 16 GiB pour les structures de page. Par conséquent, il est préférable de stocker les structures de données sur le NVDIMM lui-même dans de tels cas.
Vous pouvez configurer l'endroit où les métadonnées par page sont stockées en utilisant l'option **--map** lors de la configuration d'un espace de noms :

- Pour allouer de la mémoire vive au système, utilisez **--map=mem**.
- Pour allouer sur le NVDIMM, utilisez **--map=dev**.

20.8.2. Reconfiguration d'un espace de noms NVDIMM existant en mode DAX de système de fichiers

Vous pouvez reconfigurer un espace de noms NVDIMM (Non-Volatile Dual In-line Memory Modules) existant en mode DAX du système de fichiers.

AVERTISSEMENT

La reconfiguration d'un espace de noms supprime les données précédemment stockées dans l'espace de noms.

Conditions préalables

- L'utilitaire **ndctl** est installé. Pour plus d'informations, voir Installation de ndctl.

Procédure

1. Liste de tous les espaces de noms de votre système :

```
# ndctl list --namespaces --idle
[
  {
    "dev":"namespace1.0",
    "mode":"raw",
    "size":34359738368,
    "uuid":"ac951312-b312-4e76-9f15-6e00c8f2e6f4"
    "state":"disabled",
    "numa_node":1
  },
  {
    "dev":"namespace0.0",
    "mode":"raw",
    "size":38615912448,
    "uuid":"ff5a0a16-3495-4ce8-b86b-f0e3bd9d1817",
    "state":"disabled",
    "numa_node":0
  }
]
```

2. Reconfigurer tout espace de noms :

```
# ndctl create-namespace --force --mode=fsdax --reconfig=namespace-ID
```

Exemple 20.5. Reconfiguration d'un espace de noms en tant que système de fichiers DAX

Pour utiliser **namespace0.0** pour un système de fichiers qui prend en charge DAX, utilisez la commande suivante :

```
# ndctl create-namespace --force --mode=fsdax --reconfig=namespace0.0
{
  "dev":"namespace0.0",
  "mode":"fsdax",
  "map":"dev",
  "size":"11.81 GiB (12.68 GB)",
  "uuid":"f8153ee3-c52d-4c6e-bc1d-197f5be38483",
  "sector_size":512,
  "align":2097152,
  "blockdev":"pmem0"
}
```

L'espace de noms est désormais disponible sur le site **/dev/pmem0**.

Vérification

- Vérifiez si les espaces de noms existants sur votre système sont reconfigurés :

```
# ndctl list --namespace namespace0.0
[
  {
    "dev":"namespace0.0",
    "mode":"fsdax",
    "map":"dev",
    "size":12681478144,
    "uuid":"f8153ee3-c52d-4c6e-bc1d-197f5be38483",
    "sector_size":512,
    "align":2097152,
    "blockdev":"pmem0"
  }
]
```

Ressources supplémentaires

- La page de manuel **ndctl-create-namespace(1)**

20.8.3. Création d'un nouvel espace de noms NVDIMM en mode DAX du système de fichiers

Vous pouvez créer un nouvel espace de noms DAX de système de fichiers sur un périphérique NVDIMM (Non-Volatile Dual In-line Memory Modules) s'il y a de l'espace disponible dans la région.

Conditions préalables

- L'utilitaire **ndctl** est installé. Pour plus d'informations, voir Installation de ndctl.

- Le périphérique NVDIMM prend en charge les étiquettes permettant de créer plusieurs espaces de noms dans une région. Vous pouvez le vérifier à l'aide de la commande suivante :

```
# ndctl read-labels nmem0 >/dev/null
read 1 nmem
```

Cela indique qu'il a lu l'étiquette d'un périphérique NVDIMM. Si la valeur est **0**, cela signifie que votre périphérique ne prend pas en charge les étiquettes.

Procédure

1. Dressez la liste des régions **pmem** de votre système qui ont de l'espace disponible. Dans l'exemple suivant, de l'espace est disponible dans les régions *region1* et *region0*:

```
# ndctl list --regions
[
  {
    "dev":"region1",
    "size":2156073582592,
    "align":16777216,
    "available_size":2117418876928,
    "max_available_extent":2117418876928,
    "type":"pmem",
    "iset_id":-9102197055295954944,
    "badblock_count":1,
    "persistence_domain":"memory_controller"
  },
  {
    "dev":"region0",
    "size":2156073582592,
    "align":16777216,
    "available_size":2143188680704,
    "max_available_extent":2143188680704,
    "type":"pmem",
    "iset_id":736272362787276936,
    "badblock_count":3,
    "persistence_domain":"memory_controller"
  }
]
```

2. Attribuer un ou plusieurs espaces de noms sur l'une des régions disponibles :

```
# ndctl create-namespace --mode=fsdax --region=regionN --size=namespace-size
```

Exemple 20.6. Création d'un espace de noms sur une région

La commande suivante crée un espace de noms DAX de 36 Go sur *region0*:

```
# ndctl create-namespace --mode=fsdax --region=region0 --size=36G
{
  "dev":"namespace0.3",
  "mode":"fsdax",
  "map":"dev",
  "size":"35.44 GiB (38.05 GB)",
  "uuid":"99e77865-42eb-4b82-9db6-c6bc9b3959c2",
```

```
    "sector_size":512,
    "align":2097152,
    "blockdev":"pmem0.3"
  }
```

L'espace de noms est désormais disponible à l'adresse **/dev/pmem0.3**.

Vérification

- Vérifier si le nouvel espace de noms est créé en mode secteur :

```
# ndctl list -RN -n namespace0.3
{
  "regions":[
    {
      "dev":"region0",
      "size":2156073582592,
      "align":16777216,
      "available_size":2027224563712,
      "max_available_extent":2027224563712,
      "type":"pmem",
      "iset_id":736272362787276936,
      "badblock_count":3,
      "persistence_domain":"memory_controller",
      "namespaces":[
        {
          "dev":"namespace0.3",
          "mode":"fsdax",
          "map":"dev",
          "size":38048628736,
          "uuid":"99e77865-42eb-4b82-9db6-c6bc9b3959c2",
          "sector_size":512,
          "align":2097152,
          "blockdev":"pmem0.3"
        }
      ]
    }
  ]
}
```

Ressources supplémentaires

- La page de manuel **ndctl-create-namespace(1)**

20.8.4. Création d'un système de fichiers sur un dispositif DAX de système de fichiers

Vous pouvez créer un système de fichiers sur un périphérique DAX de système de fichiers et monter le système de fichiers. Après avoir créé un système de fichiers, l'application peut utiliser la mémoire persistante et créer des fichiers dans le répertoire *mount-point*, ouvrir les fichiers et utiliser l'opération **mmap** pour mapper les fichiers en vue d'un accès direct.

Sur Red Hat Enterprise Linux 9, les systèmes de fichiers XFS et ext4 peuvent être créés sur les NVDIMM en tant qu'aperçu technologique.

Procédure

1. Facultatif : Créez une partition sur le périphérique DAX du système de fichiers. Pour plus d'informations, voir Création d'une partition avec parted .

NOTE

Lors de la création de partitions sur un périphérique **fsdax**, les partitions doivent être alignées sur les limites de page. Sur les architectures Intel 64 et AMD64, un alignement d'au moins 4 KiB est requis pour le début et la fin de la partition. 2 MiB est l'alignement préféré.

Par défaut, l'outil **parted** aligne les partitions sur des limites de 1 Mo. Pour la première partition, indiquez 2 Mo comme début de la partition. Si la taille de la partition est un multiple de 2 MiB, toutes les autres partitions sont également alignées.

2. Créez un système de fichiers XFS ou ext4 sur la partition ou le périphérique NVDIMM :

```
# mkfs.xfs  -d su=2m,sw=1  fsdax-partition-or-device
```

NOTE

Les fichiers compatibles avec dax et les fichiers réassociés peuvent désormais coexister sur le système de fichiers. Cependant, pour un fichier individuel, dax et reflink s'excluent mutuellement.

Pour XFS, désactivez les extents de données partagés en copie sur écriture car ils sont incompatibles avec l'option de montage dax. En outre, afin d'augmenter la probabilité de mappages de pages de grande taille, définissez l'unité de bande et la largeur de bande.

3. Monter le système de fichiers :

```
# mount f_sdax-partition-ou-dispositif mount-point_
```

Il n'est pas nécessaire de monter un système de fichiers avec l'option dax pour activer le mode d'accès direct. Si vous ne spécifiez pas d'option dax lors du montage, le système de fichiers est en mode **dax=inode**. Définissez l'option dax sur le fichier avant d'activer le mode d'accès direct.

Ressources supplémentaires

- La page de manuel **mkfs.xfs(8)**
- NVDIMM en mode d'accès direct au système de fichiers

20.9. SURVEILLANCE DE L'ÉTAT DES NVDIMM À L'AIDE DE S.M.A.R.T.

Certains modules de mémoire double en ligne non volatile (NVDIMM) prennent en charge les interfaces S.M.A.R.T. (Self-Monitoring, Analysis and Reporting Technology) pour l'extraction d'informations sur la santé.

IMPORTANT

Surveillez régulièrement l'état des NVDIMM pour éviter toute perte de données. Si S.M.A.R.T. signale des problèmes liés à l'état de santé d'un périphérique NVDIMM, remplacez-le comme indiqué dans la section Détection et remplacement d'un périphérique NVDIMM défectueux.

Conditions préalables

- Facultatif : Sur certains systèmes, téléchargez le pilote **acpi_ipmi** pour récupérer des informations sur la santé à l'aide de la commande suivante :

```
# modprobe acpi_ipmi
```

Procédure

- Accéder aux informations sur la santé :

```
# ndctl list --dimms --health
[
  {
    "dev":"nmem1",
    "id":"8089-a2-1834-00001f13",
    "handle":17,
    "phys_id":32,
    "security":"disabled",
    "health":{
      "health_state":"ok",
      "temperature_celsius":36.0,
      "controller_temperature_celsius":37.0,
      "spares_percentage":100,
      "alarm_temperature":false,
      "alarm_controller_temperature":false,
      "alarm_spares":false,
      "alarm_enabled_media_temperature":true,
      "temperature_threshold":82.0,
      "alarm_enabled_ctrl_temperature":true,
      "controller_temperature_threshold":98.0,
      "alarm_enabled_spares":true,
      "spares_threshold":50,
      "shutdown_state":"clean",
      "shutdown_count":4
    }
  },
[...]
]
```

Ressources supplémentaires

- La page de manuel **ndctl-list(1)**

20.10. DÉTECTION ET REMPLACEMENT D'UN DISPOSITIF NVDIMM CASSÉ

Si vous trouvez des messages d'erreur relatifs aux modules de mémoire non volatile double en ligne (NVDIMM) dans votre journal système ou par S.M.A.R.T., cela peut signifier qu'un périphérique NVDIMM est défaillant. Dans ce cas, il est nécessaire de :

1. Détecter le périphérique NVDIMM défaillant
2. Sauvegarder les données qui y sont stockées
3. Remplacer physiquement le dispositif

Procédure

1. Détecter le dispositif cassé :

```
# ndctl list --dimms --regions --health
{
  "dimms":[
    {
      "dev":"nmem1",
      "id":"8089-a2-1834-00001f13",
      "handle":17,
      "phys_id":32,
      "security":"disabled",
      "health":{
        "health_state":"ok",
        "temperature_celsius":35.0,
        [...]
      }
[...]
}
```

2. Trouvez l'attribut **phys_id** du NVDIMM cassé :

```
# ndctl list --dimms --human
```

D'après l'exemple précédent, vous savez que **nmem0** est le NVDIMM cassé. Par conséquent, trouvez l'attribut **phys_id** de **nmem0**.

Exemple 20.7. Les attributs phys_id des NVDIMM

Dans l'exemple suivant, l'adresse **phys_id** est **0x10**:

```
# ndctl list --dimms --human

[
  {
    "dev":"nmem1",
    "id":"XXXX-XX-XXXX-XXXXXXXX",
    "handle":"0x120",
    "phys_id":"0x1c"
  },
  {
```

```
    "dev":"nmem0",
    "id":"XXXX-XX-XXXX-XXXXXXXX",
    "handle":"0x20",
    "phys_id":"0x10",
    "flag_failed_flush":true,
    "flag_smart_event":true
  }
]
```

3. Trouvez l'emplacement de mémoire du NVDIMM cassé :

```
# dmidecode
```

Dans le résultat, recherchez l'entrée dont l'identifiant **Handle** correspond à l'attribut **phys_id** du NVDIMM cassé. Le champ **Locator** indique l'emplacement de mémoire utilisé par le NVDIMM cassé.

Exemple 20.8. Liste des emplacements de mémoire NVDIMM

Dans l'exemple suivant, le périphérique **nmem0** correspond à l'identifiant **0x0010** et utilise l'emplacement de mémoire **DIMM-XXX-YYYY**:

```
# dmidecode

...
Handle 0x0010, DMI type 17, 40 bytes
Memory Device
        Array Handle: 0x0004
        Error Information Handle: Not Provided
        Total Width: 72 bits
        Data Width: 64 bits
        Size: 125 GB
        Form Factor: DIMM
        Set: 1
        Locator: DIMM-XXX-YYYY
        Bank Locator: Bank0
        Type: Other
        Type Detail: Non-Volatile Registered (Buffered)
...
```

4. Sauvegardez toutes les données contenues dans les espaces de noms du NVDIMM. Si vous ne sauvegardez pas les données avant de remplacer le NVDIMM, elles seront perdues lorsque vous retirerez le NVDIMM de votre système.

AVERTISSEMENT

Dans certains cas, par exemple lorsque le NVDIMM est complètement cassé, la sauvegarde peut échouer.

Pour éviter cela, surveillez régulièrement vos périphériques NVDIMM à l'aide de S.M.A.R.T. comme décrit dans la section Surveillance de l'état des NVDIMM à l'aide de S.M.A.R.T. et remplacez les NVDIMM défaillants avant qu'ils ne se cassent.

5. Dressez la liste des espaces de noms sur le NVDIMM :

```
# ndctl list --namespaces --dimm=DIMM-ID-number
```

Exemple 20.9. Liste des espaces de noms NVDIMM

Dans l'exemple suivant, le périphérique **nmem0** contient les espaces de noms **namespace0.0** et **namespace0.2**, que vous devez sauvegarder :

```
# ndctl list --namespaces --dimm=0

[
  {
    "dev":"namespace0.2",
    "mode":"sector",
    "size":67042312192,
    "uuid":"XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
    "raw_uuid":"XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
    "sector_size":4096,
    "blockdev":"pmem0.2s",
    "numa_node":0
  },
  {
    "dev":"namespace0.0",
    "mode":"sector",
    "size":67042312192,
    "uuid":"XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
    "raw_uuid":"XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
    "sector_size":4096,
    "blockdev":"pmem0s",
    "numa_node":0
  }
]
```

6. Remplacer physiquement le NVDIMM cassé.

Ressources supplémentaires

- Les pages de manuel **ndctl-list(1)** et **dmidecode(8)**

CHAPITRE 21. 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.

21.1. 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.

Recommendations

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.

21.2. 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.

21.3. 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.

21.4. 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 22. RETRAIT DES PÉRIPHÉRIQUES DE STOCKAGE

Vous pouvez retirer en toute sécurité un périphérique de stockage d'un système en cours d'exécution, ce qui permet d'éviter la surcharge de la mémoire du système et la perte de données.

Conditions préalables

- Avant de retirer une unité de stockage, vous devez vous assurer que vous disposez de suffisamment de mémoire système libre en raison de l'augmentation de la charge de la mémoire système lors d'un rinçage d'E/S. Utilisez les commandes suivantes pour afficher la charge de mémoire actuelle et la mémoire libre du système :

```
# vmstat 1 100
# free
```

- Red Hat ne recommande pas de supprimer un périphérique de stockage sur un système où :
  - La mémoire libre est inférieure à 5 % de la mémoire totale dans plus de 10 échantillons sur 100.
  - La permutation est active (colonnes **si** et **so** non nulles dans la sortie de la commande **vmstat**).

22.1. RETRAIT EN TOUTE SÉCURITÉ DES DISPOSITIFS DE STOCKAGE

Pour retirer en toute sécurité un périphérique de stockage d'un système en cours d'exécution, il faut adopter une approche de haut en bas. Commencez par la couche supérieure, qui est généralement une application ou un système de fichiers, et travaillez vers la couche inférieure, qui est le périphérique physique.

Vous pouvez utiliser les périphériques de stockage de plusieurs façons et ils peuvent avoir différentes configurations virtuelles au-dessus des périphériques physiques. Par exemple, vous pouvez regrouper plusieurs instances d'un périphérique dans un périphérique à chemins multiples, l'intégrer à un RAID ou à un groupe LVM. En outre, il est possible d'accéder aux périphériques via un système de fichiers ou directement, comme dans le cas d'un périphérique "brut".

En utilisant l'approche de haut en bas, vous devez vous assurer que

- le dispositif que vous souhaitez supprimer n'est pas utilisé
- toutes les E/S en attente vers le périphérique sont effacées
- le système d'exploitation ne fait pas référence au périphérique de stockage

22.2. SUPPRESSION DES PÉRIPHÉRIQUES DE BLOC ET DES MÉTADONNÉES ASSOCIÉES

Pour supprimer en toute sécurité un périphérique de bloc d'un système en cours d'exécution, afin d'éviter une surcharge de la mémoire système et une perte de données, vous devez d'abord supprimer les métadonnées qu'il contient. Traitez chaque couche de la pile, en commençant par le système de fichiers, puis le disque. Ces actions permettent d'éviter de placer votre système dans un état incohérent.

Utilisez des commandes spécifiques qui peuvent varier en fonction du type de dispositifs que vous supprimez :

- **lvremove vgremove** et sont spécifiques à LVM. **pvremove**
- Pour le RAID logiciel, exécutez **mdadm** pour supprimer la matrice. Pour plus d'informations, voir Gestion du RAID.
- Pour les dispositifs de blocage cryptés à l'aide de LUKS, il existe des étapes supplémentaires spécifiques. La procédure suivante ne fonctionnera pas pour les périphériques de bloc chiffrés à l'aide de LUKS. Pour plus d'informations, voir Chiffrement des périphériques de bloc à l'aide de LUKS.

AVERTISSEMENT

Le fait de renumériser le bus SCSI ou d'effectuer toute autre action qui modifie l'état du système d'exploitation sans suivre la procédure décrite ici peut entraîner des retards dus aux délais d'entrée/sortie, à la suppression inattendue de périphériques ou à la perte de données.

Conditions préalables

- Vous disposez d'une pile de périphériques de bloc existante contenant le système de fichiers, le volume logique et le groupe de volumes.
- Vous vous êtes assuré qu'aucune autre application ou service n'utilise le dispositif que vous souhaitez supprimer.
- Vous avez sauvegardé les données de l'appareil que vous souhaitez supprimer.
- Facultatif : si vous souhaitez supprimer un périphérique à chemins multiples et que vous ne pouvez pas accéder à ses périphériques de chemin, désactivez la mise en file d'attente du périphérique à chemins multiples en exécutant la commande suivante :

```
# multipathd disablequeueing map multipath-device
```

Cela permet aux E/S de l'appareil de tomber en panne, ce qui permet aux applications qui utilisent l'appareil de s'arrêter.

NOTE

La suppression des périphériques et de leurs métadonnées, couche par couche, garantit qu'aucune signature périmée ne subsiste sur le disque.

Procédure

1. Démonter le système de fichiers :

```
# umount /mnt/mount-point
```

2. Retirer le système de fichiers :

```
# wipefs -a /dev/vg0/myvol
```

NOTE

Si vous avez ajouté une entrée dans le fichier **/etc/fstab** pour établir une association persistante entre le système de fichiers et un point de montage, vous devez également modifier **/etc/fstab** à ce stade pour supprimer cette entrée.

Poursuivez les étapes suivantes, en fonction du type de dispositif que vous souhaitez supprimer :

3. Supprimez le volume logique (LV) qui contenait le système de fichiers :

```
# lvremove vg0/myvol
```

4. S'il ne reste aucun autre volume logique dans le groupe de volumes (VG), vous pouvez supprimer en toute sécurité le VG qui contenait le périphérique :

```
# vgremove vg0
```

5. Supprimer les métadonnées du volume physique (PV) du ou des périphériques PV :

```
# pvremove /dev/sdc1
```

```
# wipefs -a /dev/sdc1
```

6. Supprimer les partitions qui contenaient les PV :

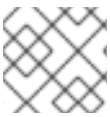
```
# parted /dev/sdc rm 1
```

NOTE

Ne suivez les étapes suivantes que si vous souhaitez effacer complètement l'appareil.

7. Supprimez la table de partition :

```
# wipefs -a /dev/sdc
```



NOTE

Ne suivez les étapes suivantes que si vous souhaitez retirer physiquement l'appareil.

- Si vous supprimez un périphérique à chemins multiples, exécutez les commandes suivantes :

  a. Afficher tous les chemins d'accès à l'appareil :

```
# multipath -l
```

La sortie de cette commande est nécessaire dans une étape ultérieure.

i. Rincer les E/S et retirer le dispositif à trajets multiples :

```
# multipath -f multipath-device
```

- Si le périphérique n'est pas configuré comme périphérique à chemins multiples, ou s'il est configuré comme périphérique à chemins multiples et que vous avez précédemment transmis des E/S aux chemins individuels, videz toutes les E/S en attente sur tous les chemins de périphérique utilisés :

```
# blockdev --flushbufs device
```

Ceci est important pour les dispositifs auxquels on accède directement et pour lesquels les commandes **umount** ou **vgreduce** n'effacent pas les E/S.

- Si vous retirez un périphérique SCSI, exécutez les commandes suivantes :

a. Supprimez toute référence au nom de l'appareil basé sur le chemin d'accès, tel que ***/dev/sd***, ***/dev/disk/by-path***ou le numéro **major:minor**, dans les applications, les scripts ou les utilitaires du système. Cela permet de s'assurer que les différents dispositifs ajoutés à l'avenir ne seront pas confondus avec le dispositif actuel.

b. Retirer du sous-système SCSI chaque chemin d'accès au périphérique :

```
# echo 1 > /sys/block/device-name/device/delete
```

Ici, le ***device-name*** est extrait de la sortie de la commande **multipath -l**, si le périphérique a été précédemment utilisé comme périphérique à trajets multiples.

8. Retirer le périphérique physique d'un système en cours d'exécution. Notez que les entrées/sorties vers d'autres périphériques ne s'arrêtent pas lorsque vous retirez ce périphérique.

Vérification

- Vérifiez que les périphériques que vous souhaitez supprimer ne sont pas affichés dans la sortie de la commande **lsblk**. Voici un exemple de sortie :

```
# lsblk

NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda      8:0    0    5G  0 disk
sr0     11:0    1 1024M  0 rom
vda    252:0    0   10G  0 disk
|-vda1 252:1    0    1M  0 part
|-vda2 252:2    0  100M  0 part /boot/efi
`-vda3 252:3    0  9.9G  0 part /
```

Ressources supplémentaires

- Les pages de manuel **multipath(8)**, **pvremove(8)**, **vgremove(8)**, **lvremove(8)**, **wipefs(8)**, **parted(8)**, **blockdev(8)** et **umount(8)**.

CHAPITRE 23. 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.

23.1. 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

23.2. 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

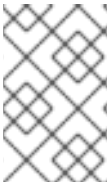
Chaque pool peut contenir un ou plusieurs systèmes de fichiers, qui stockent des fichiers.
Les systèmes de fichiers sont finement provisionnés et n'ont pas une taille totale fixe. La taille réelle d'un système de fichiers augmente avec les données qui y sont stockées. Si la taille des données approche 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.

23.3. 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.

23.4. 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
```

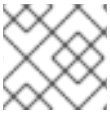
23.5. CRÉATION D'UN POOL STRATIS NON CHIFFRÉ

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
```

23.6. 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

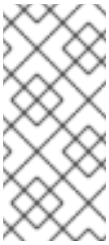
   Fait référence à la clé qui existe dans le trousseau de clés du noyau, que vous avez créé à l'étape précédente.

   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
```

4. Vérifiez que le nouveau pool Stratis a été créé :

```
# stratis pool list
```

23.7. 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 :

1. 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.

2. 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

1. 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
```

2. 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.

3. 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* .

23.8. 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

23.9. 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.

23.10. 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
```

23.11. 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
```

23.12. 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

23.13. 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
```

23.14. 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 23.1. 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 .

23.15. 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 .

23.16. 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 23.2. 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 23.3. 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 .

23.17. 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 :

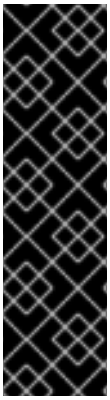
```
# /dev/stratis/[STRATIS_SYMLINK] [MOUNT_POINT] xfs defaults, x-
systemd.requires=stratis-fstab-setup@[POOL_UUID].service,x-systemd.after=stratis-stab-
setup@[POOL_UUID].service <dump_value> <fsck_value>
```

Ressources supplémentaires

- Montage persistant des systèmes de fichiers .

CHAPITRE 24. 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.

24.1. 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

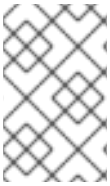
Chaque pool peut contenir un ou plusieurs systèmes de fichiers, qui stockent des fichiers.
Les systèmes de fichiers sont finement provisionnés et n'ont pas une taille totale fixe. La taille réelle d'un système de fichiers augmente avec les données qui y sont stockées. Si la taille des données approche 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.

24.2. 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

24.3. RESSOURCES SUPPLÉMENTAIRES

- Le site web *Stratis Storage*

CHAPITRE 25. 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.

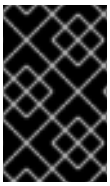
IMPORTANT

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.

25.1. 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.

25.2. 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

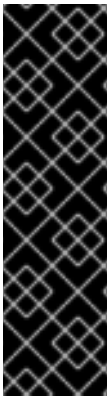
- **stratis(8)** page de manuel.

25.3. RESSOURCES SUPPLÉMENTAIRES

- Le site web *Stratis Storage*

CHAPITRE 26. 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.

26.1. 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.

26.2. 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.

26.3. 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.

26.4. 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
```

2. Démonter et supprimer le système de fichiers d'origine :

```
# umount /dev/stratis/my-pool/my-fs
# stratis filesystem destroy my-pool my-fs
```

3. 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
```

4. 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.

26.5. 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

1. Démonter l'instantané :

```
# umount /dev/stratis/my-pool/my-fs-snapshot
```

2. Détruire l'instantané :

```
# stratis filesystem destroy my-pool my-fs-snapshot
```

Ressources supplémentaires

- **stratis(8)** page de manuel.

26.6. RESSOURCES SUPPLÉMENTAIRES

- Le site web *Stratis Storage*

CHAPITRE 27. 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.

27.1. 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.

27.2. 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.

27.3. 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.

27.4. RESSOURCES SUPPLÉMENTAIRES

- Le site web *Stratis Storage*