


BTRFS

- Objet : Le système de fichiers BTRFS
- Niveau requis :
[avisé](#)
- Commentaires : *Les principes et commandes principales pour utiliser le BTRFS, les snapshots, les sauvegardes...*
- [Les sauvegardes](#) ; [Le partitionnement](#) ; [Le montage des partitions](#) 😊
- Suivi :
[à-tester](#)
 - Création par  [louispolaire](#) le 12/03/2014
 - Testé par le
- Commentaires sur le forum : [Lien vers le forum concernant ce tuto](#) ¹⁾

Ce Tuto est divisé en trois parties

- Cette première partie présente les principes du Btrfs et les commandes de base. À la fin de ce tuto vous serez capable de comprendre la terminologie de Btrfs, de créer un volume et de gérer les subvolumes et snapshots.
- [La deuxième partie présente l'installation de Debian dans un subvolume](#). À la fin de ce tuto vous serez capable d'installer Debian dans un subvolume et de modifier Grub pour y booter.
- [La troisième partie présente différentes méthodes pour réaliser des sauvegardes des subvolumes](#). A la fin de cette partie vous aurez des pistes pour élaborer une stratégie de backup de vos données sur Btrfs.

Principes de base

Btrfs est un système de fichiers pour Linux qui a pour but d'implémenter des fonctionnalités avancées tout en se focalisant sur la résistance aux pannes, les réparations et une administration simple. Sans entrer dans des détails techniques, voici quelques spécificités de BTRFS comparé aux systèmes de fichiers classiques ext3 et ext4.

Sous volumes - Subvolumes

Au sein d'un système de fichiers BTRFS, un "sous volume" (subvolume) est représenté comme un dossier classique. Il présente cependant les avantages suivants :

- C'est un sous-système de fichiers, il peut donc être monté, par exemple via la commande mount ou dans /etc/fstab.
- Il permet l'utilisation des snapshots.

Instantanés - Snapshots

Un instantané (snapshot) est un subvolume qui a été créé comme une "photographie" d'un autre subvolume à un moment donné. L'avantage d'un snapshot est que même si les données sont disponibles deux fois sur le système de fichiers, les données ne sont pas dédoublées sur le disque.

Seules les modifications faites au subvolume original ou au snapshot seront écrites sur le disque. Ainsi au moment de la création d'un snapshot, la taille sur le disque du snapshot est quasi-nulle. Ce n'est qu'au fil des modifications que sa taille sur le disque va augmenter puisque Btrfs va écrire les modifications sur le disque. Ceci est par exemple très utile pour créer des branches d'un système ou pour les sauvegardes.

RAID logiciel 0,1 & 10


Btrfs gère le RAID en logiciel, c-a-d la réplication des données sur plusieurs disques. Pour la théorie voir ici : https://fr.wikipedia.org/wiki/RAID_%28informatique%29 Pour l'instant Btrfs gère bien 3 types de RAID:

- RAID-0 : 

Une forme de RAID qui ne permet pas de réparer les erreurs, mais qui répartit une seule copie des données sur plusieurs volumes. Il y a un gain de performance (2X) en lecture et en écriture si les deux volumes sont sur des disques distincts.

- RAID-1 : 

Une forme de RAID qui stocke 2 copies complètes sur chacun des disques. Btrfs nécessite au minimum deux partitions pour utiliser le RAID-1. Les métadonnées sont en RAID-1 par défaut dans le cas d'un seul disque. On peut donc utiliser le RAID-1 avec Btrfs sur un seul disque. Il suffit d'y avoir 2 partitions (/dev/sda1 et /dev/sda2) et Btrfs répliquera les données entre ces deux partitions. Évidemment en cas de panne du disque ce type de RAID-1 ne sert à rien. Il faut utiliser 2 disques pour être protégé contre ce type de pannes.

- RAID-10 : 

Une forme de RAID qui stocke 2 copies complètes des données et répartit également chaque copie sur plusieurs volumes d'où un gain de performance. Il faut au minimum 4 volumes pour pouvoir utiliser le RAID-10.

- RAID-5 et RAID-6 ne sont pas encore totalement implémentés dans Btrfs.

Opérations à chaud

La plupart des opérations se font à chaud : le redimensionnement du système de fichiers, l'ajout d'un disque, le changement de type de redondance (RAID), l'équilibrage des données et métadonnées sur les différents disques (balance), la vérification et correction du système de fichiers, la défragmentation, la compression... Ceci est très pratique car en se dédouanant de la limitation des tables de partitions classiques, on évite les manipulations risquées et longues des redimensionnement de partitions.

Limites

Btrfs a cependant certaines limites :

- Il n'a pas de fonction de cryptage des données. Il est cependant possible de créer un volume Btrfs dans une partition créée avec dm-crypt.
- Il gère pas les fichiers de swap. Il est cependant possible de contourner le problème au prix de la performance ²⁾
- Les snapshots ne sont pas récursifs. C'est à dire que si on réalise un snapshot d'un subvolume comportant des subvolumes-enfants, dans le snapshot les subvolumes-enfants apparaîtront comme des dossiers vides.



Ce tuto présente des commandes qui, mal utilisées peuvent mener à la perte de vos données.
Sauvegardez vos données !!



Btrfs est en cours de développement. Bien que le format du système de fichier soit considéré comme stable par les développeurs ³⁾, les outils qui permettent de le manipuler sont eux en développement.
Il est donc **important d'utiliser un kernel à jour (≥ 3.12) et la version la plus à jour de Btrfs-tools (3.12-1 à ce jour)**

Installation

Btrfs est inclus dans le kernel de base, Cependant il est vivement recommandé d'utiliser un kernel le plus à jour possible (et ≥ 3.12)

Les outils de gestion doivent être installés.

```
apt-get install btrfs-tools
```

Commandes de Base

Création d'un nouveau système de fichiers

Pour formater la partition /dev/sdX1 taper :

```
mkfs.btrfs -L label /dev/sdX1
```

Il peut être utile d'ajouter l'option -f si la partition n'est pas vide.

Le partitionnement

Btrfs peut occuper tout un disque et remplacer la table de partition. Les subvolumes peuvent être utilisés pour remplacer les partitions. Il y a cependant des limites à cette utilisation :

- Il n'est pas possible d'utiliser d'autres systèmes de fichiers sur différents points de montage.
- Il n'est pas possible d'utiliser de swap sur ce disque.
- Il n'est pas possible de booter sur ce disque en UEFI.

Pour effacer la table des partitions et utiliser tout le disque /dev/sdX comme un volume Btrfs :

```
mkfs.btrfs /dev/sdX
```

Convertir un système ext3 / ext4

Il est possible de convertir un système ext3 ou ext4 en Btrfs. Si on a une partition /dev/sdX1 en ext4

```
umount /dev/sdX1
```

```
fsck.ext3 -f /dev/sdX1
```

```
btrfs-convert /dev/sdX1
```

```
mount /dev/sdX1 /point-de-montage
```

Remontez manuellement la partition et vérifiez que tous les fichiers sont là. Si la partition se chargeait via /etc/fstab modifiez le ([cf Section sur fstab](#)). À la racine de la partition /dev/sdX1 vous trouverez un subvolume "ext2_saved" qui comporte une image de l'ancien système de fichiers.

Si tout va bien et que vous avez tous vos fichiers on supprime cette image et le subvolume qui la contient :

```
rm /point-de-montage/ext2_saved/*
```

```
btrfs subvolume delete /ext2_saved
```

```
btrfs balance start /point-de-montage
```

Si tout ne va pas bien, qu'il y a eu un problème ou que vous avez les jetons 🤪 :

```
umount /dev/sdX1
```

```
btrfs-convert -r /dev/SdX1
```

Options de montage

En des différentes options disponibles lors de la création du système de fichiers, le nombreuses options de montage pour Btrfs peuvent drastiquement changer ses performances.

Option de montage propres à Btrfs	Rôle
autodefrag	Autorise la défragmentation en arrière plan
compress=zlib	La compression la plus importante. Par défaut

Option de montage propres à Btrfs	Rôle
compress=lzo	La compression la plus rapide
compress=no	Pas de compression
degraded	Utilisé dans le cas des RAID. Permet de monter le volume même si certains disques sont manquants
discard	Utilise le TRIM pour les disques SSD
recovery	Lance la réparation automatique après le montage. A utiliser si le volume ne veut plus se monter. une fois monté réaliser un scrub pour réparer les erreurs. (Ne fonctionne qu'avec les volumes créés avec la version 3.2 ou +)
ssd	Utilise l'optimisation pour disques SSD de Btrfs. C'est utilisé par défaut si votre système reconnaît le SSD. Cette option ne lance pas le TRIM !
subvol=/chemin/vers/subvolume	Permet de monter un subvolume
subvol=id	Permet de monter un subvolume par son id. La racine a un id=0

Pour un SSD

Par exemple

```
mount /dev/sdX1 -o rw,noatime,ssd,discard,autodefrag,compress=lzo,space_cache,inode_cache /mnt
```

Les options les plus importantes pour un ssd sont "ssd" et "discard".

Pour un HDD

Par exemple

```
mount /dev/sdX1 -o rw,noatime,compress=zlib,autodefrag
```

En utilisant /etc/fstab

Je vous conseille d'utiliser les UUID. Si vous utilisez plusieurs disques pour le même volume peu importe lequel vous montez. Les options dump et pass dans les deux derniers champs doivent être égal à 0.

Exemple de ligne dans /etc/fstab :

```
UUID=a69d9182-f4c7-4276-b35d-7d5f9bd50a57 /mnt Btrfs
rw,noatime,ssd,discard,autodefrag 0 0
```

Outils d'audit

Pour voir l'espace utilisé dans le volume Btrfs monté dans /mnt :

```
btrfs filesystem df /mnt
```

[retour de la commande](#)

```
Data, single: total=27.00GiB, used=24.28GiB
System, DUP: total=32.00MiB, used=4.00KiB
Metadata, DUP: total=2.50GiB, used=1.70GiB
```

il y a 3 lignes : Data correspond aux données, System aux fichiers système du volume et Metadata aux métadonnées associées aux différents fichiers (sommes de contrôles etc...) la somme des used= correspond à l'espace utilisé sur le disque. Vous noterez la différence avec la commande df -h.

Pour avoir des informations sur le disque, l'UUID du volume etc :

```
btrfs filesystem show /mnt
```

[retour de la commande](#)

```
Label: none  uuid: a69d9182-f4c7-4276-b35d-7d5f9bd50a57
Total devices 1 FS bytes used 25.98GiB
devid      1 size 115.43GiB used 32.06GiB path /dev/sdb6

Btrfs v3.12
```

Dans le cas d'un volume en RAID (ici RAID 10) on peut voir la répartition des données sur les différents disques :

[retour de la commande](#)

```
Label: stockage  uuid: ca7fc4a0-3bf6-4d86-b4b9-3ef025af5d31
Total devices 4 FS bytes used 350.62GiB
devid      1 size 931.51GiB used 182.04GiB path /dev/sdd
devid      2 size 931.51GiB used 182.03GiB path /dev/sda
devid      3 size 931.51GiB used 182.03GiB path /dev/sdb
devid      4 size 931.51GiB used 182.03GiB path /dev/sdc

Btrfs v3.12
```

On peut aussi voir si le disque a fait des erreurs

```
btrfs device stats /dev/sdX
```

[retour de la commande](#)

```
[/dev/sdX].write_io_errs    0
[/dev/sdX].read_io_errs    0
```

```
[/dev/sdX].flush_io_errs    0  
[/dev/sdX].corruption_errs 0  
[/dev/sdX].generation_errs 0
```

Ici tout va bien.

Les Subvolumes

Pour créer un subvolume :

```
btrfs subvolume create NOM
```

Un subvolume apparaît dans votre gestionnaire de fichiers comme un dossier. Vous pouvez le manipuler comme tel. Par contre la suppression devra se faire avec **btrfs subvolume delete SNAP**.

Pour avoir la liste des subvolumes dans le volume Btrfs :

```
btrfs subvolume list /mnt
```

Pour supprimer un subvolume :

```
btrfs subvolume delete NOM
```

Monter un subvolume

Pour monter un subvolume présent dans /dev/sdX :

```
mount /dev/sdX -o ..options...,subvol=NOM /point-de-montage
```

Dans /etc/fstab il suffit de rajouter également l'option de montage **subvol=chemin vers le subvolume**. Par exemple si on a un subvolume "SUB" dans un dossier "DOS" à la racine du volume Btrfs on rajoutera subvol=DOS/SUB aux options de montage.

Snapshots

Pour réaliser un snapshot du subvolume NOM :

```
btrfs subvolume snapshot NOM SNAP
```

Il est possible de réaliser des snapshots en lecture seule avec l'option **-r**.

Un snapshot est un subvolume comme un autre, c'est juste sa méthode de création qui change. En tant que subvolume vous pouvez le manipuler comme un dossier, avec cd, ls, mv etc... La suppression devra se faire avec **btrfs subvolume delete SNAP**.

Par exemple si vous voulez remplacer le subvolume NOM par le subvolume(ex-snapshot) SNAP

```
ls /point-de-montage
```

```
NOM SNAP
```

```
cd /point-de-montage
```

```
mv NOM NOM_old
```

```
mv SNAP NOM
```

```
ls /point-de-montage/
```

```
NOM NOM_old
```

Et on peut faire un snapshot de snapshot...

```
btrfs subvolume snapshot NOM NOM_snap
```

```
btrfs subvolume snapshot NOM_snap NOM_snap_snap
```

```
ls /point-de-montage
```

```
NOM NOM_old NOM_snap NOM_snap_snap
```

Et pendant ce temps là si aucune modification n'est faire dans les subvolumes/snapshots... L'espace libre sur le disque reste le même.

La compression

Btrfs peut compresser le système de fichiers ce qui veut dire que tous les fichiers dans le volume sont compressés. Il y a donc une réduction de la taille des fichiers sur le disque mais aussi un gain de performances, spécialement dans le cas de la compression lzo.

Pour autoriser la compression sur un volume on ajoute l'option de montage **compress=lzo** ou **compress=zlib**

Pour compresser un volume non compressé ou changer le type de compression (à chaud bien sur !):

```
btrfs filesystem defragment -r -v -clzo /point-de-montage
```

Maintenance

Scrub

Scrub permet de vérifier à chaud le système de fichiers. Les données et métadonnées sont lues et les sommes de contrôle sont vérifiées, les données corrompues sont identifiées et réparées.

```
btrfs scrub start /point-de-montage
```

On peut changer la commande **start** avec **cancel** pour stopper la vérif, **pause**, et **status** pour voir l'état d'une vérif en cours.

Balance

Balance ré-alloue les données dans le système de fichier. C'est particulièrement utile lorsqu'on a un système de fichiers en RAID et qu'un disque est ajouté ou retiré. Les copies manquantes sont ainsi régénérées.

```
btrfs balance start /point-de-montage
```

Idem que pour scrub pour les commandes **start**, **cancel** et **status**.

RAID 0 1 et 10

Si on dispose de plusieurs disques il est possible d'activer la redondance logicielle des données. Si vous voulez plus d'information sur le RAID : https://fr.wikipedia.org/wiki/RAID_%28informatique%29

Par exemple si on a deux disques /dev/sda et /dev/sdb

On crée un système de fichier sur ces deux disques :

```
mkfs.btrfs /dev/sda /dev/sdb
```

On monte le volume :

```
mount /dev/sda /mnt
```

Si on décide d'augmenter la taille de ce volume (à chaud) en ajoutant un disque /dev/sdc :

```
btrfs device add /dev/sdc
```

puis on ré-alloue les données en utilisant les 3 disques.

```
btrfs balance start /mnt
```

Par défaut Btrfs utilise RAID0, pour passer en RAID1 :

```
btrfs balance start -dconvert=raid1 -mconvert=raid1 /mnt
```

Il est aussi possible de supprimer des disques si nécessaire.

Références

- [Le wiki de Btrfs](#)
- [La page sur Btrfs sur le wiki d'Archlinux](#)

¹⁾

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

²⁾

https://btrfs.wiki.kernel.org/index.php/FAQ#Does_btrfs_support_swap_files.3F

³⁾

https://btrfs.wiki.kernel.org/index.php/Main_Page#Stability_status

From:

<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:

<http://debian-facile.org/doc:systeme:btrfs>

Last update: **01/07/2023 08:06**

