La Sauvegarde de subvolumes BTRFS

- Objet : Les Snaphots et les sauvegardes en BTRFS
- Niveau requis :
  avisé
- Commentaires : *Différentes méthodes pour sauvegarder son système en BTRFS*
- 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 :

- La 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.
- Cette 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.

Ce Tutoriel fait suite à Btrfs - Les bases et Installation de Debian dans un Subvolume Btrfs
Assurez vous d'avoir bien lu ce précédent tuto avant d'envisager celui-ci !

Gestion des Snapshots

L'intérêt d'avoir la racine de son OS dans un subvolume est que l'on peut en faire des snapshots et booter dans ces snapshots en utilisant la methode ci dessus. Par exemple :

```
cp 40_custom 41_debian2
```

```
chmod +x 41_debian2
```

Changer les subvolumes dans le menuentry pour correspondre au snapshot.

```
update-grub
```

C'est pratique pour :

- Essayer un autre gestionnaire de fenêtre sans abîmer la config actuelle.
- Essayer systemd (puisque c'est d'actualité)
- Faire des bêtises avec rm ...

- Assurer ses arrières quand on est sous sid…
- Ai-je besoin d'énumérer toutes les raisons de casser son système ?? 😀

Pour cela je vais présenter 2 méthodes, une qui sera basée sur des scripts et **cron** et l'autre sur **snapper** qui est dans les dépôts

Sur un même volume physique

Dans ce cas on veut faire des snapshot et les garder dans la même partition. Ça protège des erreurs de manipulation mais pas des pannes de disques.

Snapper

Snapper est un programme qui gère les snapshots et qui a été développé par OpenSuse [2)].

```
apt-get install snapper
```

```
man snapper
```

Pour des snapshots automatiques de la racine :

```
snapper -c root create-config /
```

Ceci va créer un subvolume /.snapshot à la racine et les snapshots seront placés dedans. Les snaphots se font toutes les heures

Vous pouvez éditer /etc/snapper/configs/root pour changer le nombre de snaphsots que snapper va garder au fil du temps.

/etc/snapper/configs/root

```
# limits for timeline cleanup
TIMELINE_MIN_AGE="1800"
TIMELINE_LIMIT_HOURLY="10"
TIMELINE_LIMIT_DAILY="10"
TIMELINE_LIMIT_MONTHLY="10"
TIMELINE_LIMIT_YEARLY="10"
```

La configuration par défaut garde les 10 derniers snapshots (pris toutes les heures), 10 snapshots par jour, 10 snapshots par mois et 10 snapshots par an… Changez ces paramètres selon vos besoins.

Sachant que la taille d'un snapshot sur le disque correspond aux modifications sur les fichiers entre le snapshot et son subvolume parent. Je vous laisse donc imaginer la place pris sur le disque d'un snapshot qui a un an !

Notez encore une fois que les snapshots ne sont pas récursifs !
Si vous avez /home dans un subvolume, alors votre /home ne sera pas snapshotté.
recréez alors une configuration pour home

```
snapper -c home create-config /home
```

Voici ce que vous aurez une semaine après :

```
snapper list
```

retour de la commande

```
Type   | #   | Pré # | Date                            | Utilisateur |
Nettoyage | Description | Données utilisateur
-------+-----+-------+---------------------------------+-------------+--
---------+-------------+--------------------
single | 0   |       |                                 | root        |
| current     |
single | 475 |       | jeu. 06 mars 2014 00:17:01 CET | root        |
timeline  | timeline    |
single | 488 |       | ven. 07 mars 2014 00:17:01 CET | root        |
timeline  | timeline    |
single | 501 |       | sam. 08 mars 2014 00:17:01 CET | root        |
timeline  | timeline    |
single | 507 |       | dim. 09 mars 2014 00:17:01 CET | root        |
timeline  | timeline    |
single | 519 |       | lun. 10 mars 2014 00:17:01 CET | root        |
timeline  | timeline    |
single | 531 |       | mar. 11 mars 2014 00:17:01 CET | root        |
timeline  | timeline    |
single | 538 |       | mer. 12 mars 2014 09:17:01 CET | root        |
timeline  | timeline    |
single | 548 |       | mer. 12 mars 2014 19:17:01 CET | root        |
timeline  | timeline    |
single | 549 |       | mer. 12 mars 2014 20:17:01 CET | root        |
timeline  | timeline    |
single | 550 |       | mer. 12 mars 2014 21:17:01 CET | root        |
timeline  | timeline    |
single | 551 |       | mer. 12 mars 2014 22:17:01 CET | root        |
timeline  | timeline    |
single | 552 |       | mer. 12 mars 2014 23:17:01 CET | root        |
timeline  | timeline    |
single | 553 |       | jeu. 13 mars 2014 12:17:01 CET | root        |
timeline  | timeline    |
single | 554 |       | jeu. 13 mars 2014 13:17:02 CET | root        |
timeline  | timeline    |
single | 555 |       | jeu. 13 mars 2014 14:17:01 CET | root        |
timeline  | timeline    |
single | 556 |       | jeu. 13 mars 2014 16:17:01 CET | root        |
```

```
timeline  | timeline    |
single | 557 |       | jeu. 13 mars 2014 17:17:01 CET | root        |
timeline  | timeline    |
single | 558 |       | jeu. 13 mars 2014 18:17:01 CET | root        |
timeline  | timeline    |
single | 559 |       | jeu. 13 mars 2014 19:17:01 CET | root        |
timeline  | timeline    |
single | 560 |       | jeu. 13 mars 2014 20:17:01 CET | root        |
timeline  | timeline    |
single | 561 |       | jeu. 13 mars 2014 21:17:01 CET | root        |
timeline  | timeline    |
single | 562 |       | jeu. 13 mars 2014 22:17:01 CET | root        |
timeline  | timeline    |
```

Manuellement

Le principe de base est simple : un script bash et cron.

```
btrfs subvolume snapshot /home/btrfs/debian/home /home/btrfs/snapshots/home-backup-$(date)
```

Sauvegarde incrémentale

Entre deux volumes physiques

On a deux disques formatés en btrfs /dev/sda monté en /mnt/sda et /dev/sdb monté en /mnt/sdb.

Mise à jour initiale

On veut sauvegarder le subvolume home qui se trouve dans /mnt/sda/home et l'envoyer dans /mnt/sdb.

On réalise un snapshot de home en lecture seule

```
btrfs subvolume snapshot -r /mnt/sda/home /mnt/sda/home@a-envoyer
```

On l'envoie sur /mnt/sdb

```
btrfs send /mnt/sda/home@a-envoyer | btrfs receive /mnt/sdb/
```

On a maintenant la configuration suivante :

/dev/sda	**/dev/sdb**
home	

/dev/sda	/dev/sdb
home@a-envoyer	home@a-envoyer

On renomme les snapshots pour l'étape suivante et on réalise un snapshot de home@a-envoyer qui ne soit pas en lecture seule :

```
mv /mnt/sda/home@a-envoyer /mnt/sda/home@envoye
```

```
mv /mnt /sdb/home@a-envoyer /mnt/sdb/home@envoye
```

```
btrfs subvolume snapshot /mnt/sdb/home@envoye /mnt/sdb/home
```

On a ainsi deux subvolumes home qui sont identiques sur sda et sdb.

/dev/sda	/dev/sdb
home	home
home@envoye	home@envoye

Maintenant pour la mise a jour incrémentale :

Après avoir fait des modification dans /mnt/sda/home on réalise un nouveau snapshot en lecture seule :

```
btrfs subvolume snapshot -r home home@a-envoyer
```

On a donc :

/dev/sda	/dev/sdb
home (modifié)	home
home@envoye	home@envoye
home@a-envoyer (modifié)	

On signifie à **btrf send** qu'il ne faut envoyer que les différences entre home@envoye et home@a-envoyer

Ce qui est très intéressant parce que les différences ne se font pas au niveau des fichiers mais des clunks, ainsi si vous modifiez le une partie seulement d'un gros fichier, seuls les clunks correspondant aux modifications seront envoyés... du coup ça va beaucoup plus vite.
Mais cela veut aussi dire qu'il est important que le subvolume home@envoye soit présent et identique sur les deux disques et en lecture seule.

```
 btrfs send -p /mnt/sda/home@envoye /mnt/sda/home@a-envoyer | btrfs receive /mnt/sdb/
```

On a maintenant :

/dev/sda	/dev/sdb
home (modifié)	home
home@envoye	home@envoye
home@a-envoyer (modifié)	home@a-envoyer(modifié)

Il ne reste plus qu'a faire le ménage pour se préparer à répéter cette itération :

```
btrfs subvolume delete /mnt/sda/home@envoye
```

```
mv /mnt/sda/home@a-envoyer /mnt/sda/home@envoye
```

```
btrfs subvolume delete /mnt/sdb/home@envoye
```

```
mv /mnt/sdb/home@a-envoyer /mnt/sdb/home@envoye
```

```
btrfs subvolume delete /mnt/sdb/home
```

```
btrfs subvolume snapshot /mnt/sdb/home@a-envoyer /mnt/sdb/home
```

Ce qui nous donne :

/dev/sda	/dev/sdb
home (modifié)	home (modifié)
home@envoye (modifié)	home@envoye (modifié)

On est prêt à recommencer dès que /mnt/sda/home sera modifié ! Je vous conseille un petit script et une tache cron pour automatiser tout ça.

Entre deux machines via ssh

Les principes de bases sont les mêmes. Dans ce cas /mnt/sdb se trouve sur le serveur root@stockage.local

Pour la mise a jour initiale :

```
btrfs send /mnt/sda/home@a-envoyer | ssh root@stockage.local /sbin/btrfs
receive /mnt/sdb
```

Pour la mise a jour incrémentale :

```
btrfs send -p /mnt/sda/home@envoye /mnt/sda/home@a-envoyer | ssh
root@stockage.local /sbin/btrfs receive /mnt/sdb
```

Note : pour la lisibilité du tuto je me connecte en root sur le serveur, ce n'est pas la meilleure idée.

Il est crucial que les subvolumes home@envoye soient identiques sur le serveur et le client. Sinon la transaction échouera. Je conseille de vérifier que le transfert s'est bien passé comparant les md5sums. Inconvénient : c'est long et ça utilise le processeur.

Il n'y a pas, à ma connaissance, de système de checksum accessible à l'utilisateur intégré à btrfs. J'utilise donc un script qui compare les md5sums des ls chez le client et le serveur. Sur le client :

```
export LANG=C
```

```
cd /mnt/sda/home@envoye
```

```
ls -lnARs > /tmp/md5
```

```
md5sum /tmp/md5
```

Sur le Serveur:

```
ssh root@stockage.local "LANG=C cd /mnt/sdb/home@envoye && ls -lnARs" > /tmp/md5
```

```
md5sum /tmp/md5
```

Si les deux md5sum ne sont pas identiques alors il faut refaire une mise à jour complète.

Références

- Le wiki de Btrfs
- La page sur Btrfs sur le wiki d'Archlinux
- [[http://zythmer.acyclic.org/blog/2013/01/27/debian-full-disk-encryption-btrfs-subvolumes.html|http://zythmer.acyclic.org/blog/2013/01/27/debian-full-disk-encryption-btrfs-subvolumes.html]] - **Lien Obsolète**

1)
N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !
2)
http://snapper.io/manpages/snapper.html