

## Les snapshots (instantannés) sous Linux et FreeBSD

### Principe

Un instantanné permet de pouvoir figer l'état du système de fichier à un instant  $t$  pour des besoins de sauvegarde.

Une sauvegarde peut parfois durer longtemps (plusieurs minutes ou heures) et il est important que pendant le temps que dure la sauvegarde les données ne soient pas modifiées, ceci pour assurer la cohérence des données. Ainsi il est souvent nécessaire de stopper toute activité logicielle qui utilise le système de fichier pour le temps d'une sauvegarde.

Grâce à un snapshot on sauvegarde en fait l'état des données à un instant donné. L'intérêt est que la création d'un snapshot est quelque chose de beaucoup plus rapide qu'une sauvegarde, cela permet de stopper les applications uniquement pour le temps de la création du snapshot.

Le principe du snapshot est en fait de sauvegarder les metadata (pointeurs sur les blocs de fichiers) et d'assurer que pendant la durée de vie du snapshot, aucune modification ne sera faite sur ses blocs. Ainsi, une fois le snapshot créé on peut effectuer la sauvegarde du système de fichier sans risque de corruption.

### Technologies utilisées

Chaque type de système d'exploitation possède sa technologie adaptée à l'utilisation de snapshots.

Sous linux il s'agit de LVM et sous FreeBSD, UFS2.

## 1. Linux et LVM

### **Présentation de LVM**

LVM pour « logical volume manager » est une technique qui permet de modifier la taille des partitions sur les disques durs sans avoir besoin de tout reformater, voire de créer des partitions s'étalant sur plusieurs disques. L'objectif est ainsi d'éviter arrêt et redémarrage d'une machine en production. Cette technique est disponible sur linux depuis la version 2.4 du noyau.

En plus de ces fonctionnalités, LVM permet la création de snapshots.

### **Organisation de LVM**

Le principe de LVM est de pouvoir regrouper *un ou plusieurs disques dur* dans un *groupe*. Ce groupe peut ensuite être redécoupé en *disques logiques*. Ce système très flexible permet beaucoup de souplesse dans la configuration de ses disques durs.

Attention toutefois, LVM ne doit pas être utilisé avec la partition `/boot`.

Les distributions récentes permettent pour la plupart de créer des volumes LVM dès l'installation, ce qui facilite son utilisation. Il est bien sûr possible de configurer LVM sur de nouveaux disques durs que l'on souhaite installer.

Lorsqu'on installe LVM sur un disque, toutes les données présentes sont effacées!

Si LVM est utilisé après l'installation du système d'exploitation, il est nécessaire de charger au démarrage le module qui prend en charge LVM. Ceci se fait grâce à la commande `lvmcreate_initrd`. Il faudra également modifier en conséquence le fichier `/etc/lilo.conf`.

Les termes utilisés dans LVM pour décrire son organisation sont :

- **Volume physique** (physical volume) : chaque disque dur deviendra un volume physique au moment de la création.
- **Groupe de volumes** (volume group) : les volumes physiques sont regroupés dans un groupe de volumes.
- **Volumes logiques** (logical volume) : un groupe de volumes est découpé en un ou plusieurs volumes logiques.

### ➤ **Création d'un volume physique**

On crée un volume physique grâce à la commande `pvcreate` (physical volume creation).

```
pvcreate [-f] /dev/hdxx où
```

-f : force la création du volume. A utiliser si le disque avait déjà été transformé en volume physique.

/dev/hdxx : fichier spécial du disque ou de la partition à transformer en volume physique

Par exemple, création d'un volume physique à partir du disque /dev/hdb1

```
# pvcreate /dev/hdb1
pvcreate -- physical volume "/dev/hdb1" successfully created
```

### ➤ **Création d'un groupe de volumes et ajout de volumes physiques**

Une fois qu'un volume physique est créé, il faut l'insérer dans un groupe de volumes. Cela se fait grâce à la commande `vgcreate`.

```
vgcreate nom_du_volume /dev/hdxx où
```

nom\_du\_volume : nom du groupe de volume - l'opération crée alors le répertoire

/dev/nom\_du\_volume

/dev/hdxx : fichier spécial du volume physique

Par exemple on crée le groupe de volumes `volume1` à partir du volume physique /dev/hdb1

```
# vgcreate volume1 /dev/hdb1
vgcreate -- INFO: using default physical extent size 4 MB
vgcreate -- INFO: maximum logical volume size is 255.99 Gigabyte
vgcreate -- doing automatic backup of volume group "volume1"
vgcreate -- volume group "volume1" successfully created and
activated
```

Pour maintenant ajouter d'autres volumes physiques au groupe de volume, on utilise la commande `vgextend`

```
vgextend nom_volume /dev/hdxx
```

nom\_volume : nom du groupe de volume à redimensionner

/dev/hdxx : volume physique à ajouter ou retirer du groupe de volumes

Ajout de `hdb2` au groupe de volumes `volume1`

```
# vgextend volume1 /dev/hdb2
vgextend -- INFO:maximum logical volume size is 255.99 Gigabyte
vgextend
--doing automatic backup of volume group "volume1" vgextend
--volume group "volume1" successfully extended
```

### ➤ **Création d'un volume logique**

Il s'agit de la dernière étape de la configuration de volumes LVM. Lorsqu'un groupe de volumes a été créé, il faut le séparer en un ou plusieurs volumes logiques.

La commande `lvcreate` permet de créer un volume logique de la taille qu'on désire avec le nom qu'on désire dans tel groupe de volumes. Une fois cette étape passée, un « disque » LVM est créé dans `/dev/nom_du_groupe_de_volume/nom_du_volume_logique`

```
lvcreate -L tailleK|M|G [-n nom] <nom_volume> où  
-L tailleK|M|G : taille du volume logique exprimable en Ko, Mo ou Go  
-n nom : nom du volume logique - l'opération crée un fichier spécial portant ce nom pour le volume  
logique et sera placé dans le répertoire /dev/nom_volume  
<nom_volume> : nom du groupe de volumes dans lequel sera créé le volume logique.
```

On crée un volume logique de 600 Mo nommé `part1` dans le groupe de volume « `volumel` »

```
# lvcreate -L 600 -n part1 volumel  
lvcreate -- doing automatic backup of "volumel"  
lvcreate -- logical volume "/dev/volumel/part1" successfully  
created
```

### ➤ **Informations sur les composants de LVM**

On peut obtenir des informations sur chaque composant : volume physique, groupe de volumes et volume logique grâce aux commandes :

```
pvdisplay /dev/hdxx  
vgdisplay volumel  
lvdisplay /dev/volumel/part1
```

## **Création de snapshot**

La création de snapshot avec LVM se fait de façon simple : on crée un nouveau volume logique qui contient le snapshot, on monte ce volume, on effectue sa sauvegarde, on démonte le volume puis on le détruit.

### **1. Création du snapshot**

On crée ici le snapshot qui apparaîtra comme un duplicat du volume logique qu'on souhaite sauvegarder. Il faut prévoir suffisamment de place pour ce volume logique (environ 500Mo) et spécifier à la création qu'il s'agit d'un volume logique

```
# lvcreate -L500M -s -n snapshot /dev/volumel/part1  
lvcreate -- WARNING: the snapshot must be disabled if it gets full  
lvcreate -- INFO: using default snapshot chunk size of 64 KB for  
"/dev/volumel/snapshot"  
lvcreate -- doing automatic backup of "volumel"  
lvcreate -- logical volume "/dev/volumel/snapshot" successfully  
created
```

Si il n'y a pas assez de place pour le snapshot, celui ci est désactivé.

Si le système de fichier utilisé est XFS, il faut utiliser la commande `xfs_freeze -f` pour verrouiller le système de fichier avant la création du snapshot puis le déverrouiller avec `xfs_freeze -u`.

## 2. Montage du snapshot

On crée le point de montage et on monte le snapshot.

```
# mkdir /mnt/snapshot
# mount /dev/volume1/snapshot /mnt/snapshot
mount: block device /dev/volume1/snapshot is write-protected,
mounting read-only
```

Avec un système de fichier XFS il faut spécifier l'option `-onouuid,ro` à `mount`.

## 3. Sauvegarde

On utilise le mode de sauvegarde qu'on veut : `tar`, `rsync`, etc.

```
#tar -cvzf /backup.tar.gz /mnt/snapshot
```

## 4. Démontage et destruction du snapshot

Une fois que la sauvegarde est finie, on a plus besoin du snapshot, on le détruit donc.

```
# umount /mnt/snapshot
# lvremove /dev/volume1/snapshot
lvremove -- do you really want to remove "/dev/volume1/snapshot"?
[y/n]: y
lvremove -- doing automatic backup of volume group "volume1"
lvremove -- logical volume "/dev/volume1/snapshot" successfully
removed
```

## Informations supplémentaires

LVM sur lea-linux.org : [http://lea-linux.org/leapro/pro\\_sys/lvm.html](http://lea-linux.org/leapro/pro_sys/lvm.html)

Howto de LVM, partie snapshots :

[http://www.tldp.org/HOWTO/LVM-HOWTO/snapshots\\_backup.html](http://www.tldp.org/HOWTO/LVM-HOWTO/snapshots_backup.html)

## 2. FreeBSD et UFS2

UFS2 est le système de fichier utilisé par défaut par FreeBSD à partir de sa version 5. Ce système de fichier permet la création de snapshot.

Ici le principe est différent de celui de LVM, il existe deux types d'utilisation :

- la sauvegarde du système de fichier avec l'outil `dump`
- la création d'un snapshot pour naviguer dans l'image figée à la date de création du snapshot.

### Sauvegarde avec `dump`

Le principe est de faire une sauvegarde complète d'un système de fichier grâce à l'outil `dump`. On pourra plus tard restaurer les données avec la commande `restore`.

Attention, un système de fichier qu'on souhaite sauvegarder avec `dump` est un de ceux définis dans le fichier `/etc/fstab`.

Avant de faire le dump, créer un dossier « .snap » à la racine du système de fichier à sauvegarder. Le propriétaire doit être « root », le groupe « operator » et le mode « 0770 »

Exemple : sauvegarde du système de fichier /home

```
# cd /home
# mkdir .snap
```

Si le dossier n'a pas les bonnes propriétés de propriétaire, groupe, mode énoncées au dessus : les changer. Enfin, effectuer la sauvegarde effective : dump de /home dans fichier\_de\_sauvegarde

```
# dump -L -f fichier-de-sauvegarde /home
```

L'option -L permet de créer un dump basé sur un snapshot.

Pour plus d'informations, consulter les pages de manuel de `dump(8)` et `restore(8)`.

## Exploration d'un snapshot

Au départ on crée un snapshot du système de fichier que l'on souhaite sauvegarder. Le snapshot sera en fait un fichier.

Exemple : création du snapshot de /home

```
mount -u -o snapshot /home/fichier_snapshot /home
```

Cela crée le snapshot « fichier\_snapshot » du système de fichier /home grâce à l'option « -o snapshot » de la commande `mount`. Attention, le fichier du snapshot doit être créé dans le système de fichier dont on crée le snapshot.

Il faut maintenant monter le snapshot pour pouvoir naviguer dans l'image figée.

```
# mdconfig -a -t vnode -f /home/fichier_snapshot -u 1
# mkdir /mnt/image_figee
# mount -o ro /dev/md1 /mnt/image_figee
```

Avant de monter l'image, on crée un « disque dur virtuel » /dev/md1 à partir de fichier\_snapshot grâce à la commande `mdconfig`. On monte ensuite ce disque virtuel en lecture seule (option « -o ro ») dans le dossier `image_figee`.

Une fois qu'on n'a plus besoin d'utiliser l'image figée du système de fichier, on démonte l'image, on désactive le disque virtuel et enfin on supprime le fichier du snapshot.

```
# umount /mnt/image_figee
# mdconfig -d -u 1
# rm -f /home/fichier_snapshot
```

Pour simplifier la gestion des snapshot (créer, monter, démonter, supprimer) on peut se tourner vers l'outil `snapshot` présenté sur cette page : <http://people.freebsd.org/~rse/snapshot/>

Consulter également la page de manuel de `mount(8)`.

Il est à noter que le gestionnaire de volume `vanium` de FreeBSD permettra dans ses versions futures de créer des snapshots sur le même principe que LVM.