Duplication et synchronisation de données

Comme je l'ai déjà abordé, je suis d'un naturel assez pessimiste, même dans les sauvegardes, donc je prévois toujours un plan B.
Je pense que j'ai déjà fait le tour de la question dans de précédents articles (Faire des sauvegardes : Snapshot, Sauvegarde de données sous Linux).

Je n'ai pas abordé la duplication ou la synchronisation de données, mais a quoi ça sert, en effet votre système de sauvegarde est au point, pourquoi donc se prendre la tête là-dedans.

Je répondrai, les problèmes de sauvegarde sont en effet peu fréquents, mais quand il arrive ..., de plus je distingue 2 avantages à la synchronisation ou la duplication :

- L'avantage le plus important est la réduction d'espace occupé par les sauvegardes.
- Un avantage indirect, conséquence du précédent est la diminution de la bande passante nécessaire à la sauvegarde dans le cas de la dé-duplication à la source.

Je vois un autre avantage, oui je sais que j'avais dit 2, mais le dernier est la sérénité 😬 .

1. Copie binaire avec dd

La commande de copie bloc à bloc dd permet des copies de bas niveau d'un périphérique.
Elle est utilisée pour la duplication de disques durs, mais aussi pour la création d'images binaires de périphériques de stockage.
Pour ceux qui connaisse le Raspberry pi et oui vous ne fait que récupérer qu'une copie de Rasbian ou autres (pidora, etc).
Syntaxe de la commande dd :

```
dd if=entrée of=sortie bs=taille_blocs count=nombre_blocs
```

entrée : Le fichier à copier.
sortie : Le fichier vers lequel copier.
taille_blocs : Pas obligatoire, désigne la taille des blocs à copier.
nombre_blocs : Pas obligatoire, désigne le nombre de blocs à copier, si cela n'est pas précisé, la copie s'arrête dès qu'elle n'est plus possible.

Exemple
Copie de disque :

```
~# dd if=/dev/sda of=/dev/sdb
```

Réalisation d'une image ISO d'un CD :

```
~# dd if=/dev/cdrom of=/home/ksh/image.iso
```

Effacement d'une clé USB :

```
~# dd if=/dev/zero of=/media/sdc
```

Création d'un fichier vide de 100MO :

```
~# dd if=/dev/zero of=/home/ksh/swap.100 bs=1024 count=100000
```

2. Génération de fichiers ISO avec genisoimage

Les fichiers ISO sont des images binaires de CD ou DVD.
Les images ISO sont montables par la commande **mount**, gravables par n'importe quel logiciel de gravure et exploitables depuis les machines virtuelle où elles sont vue comme un CD.
Il peut être utile de générer des images ISO à partir d'une arborescence de fichiers et répertoires.

Syntaxe de la commande genisoimage :

```
genisoimage -J -o image répertoire
```

-j : Optionnel, génère des enregistrements en plus de la structure de nom iso9960.
-o image : Le fichier ISO qui sera généré, généralement suivi de l'extension `.iso`.
répertoire : Le répertoire à partir duquel l'image ISO sera générée.

Exemple d'utilisation :

```
~# mkdir tt
~# echo >tt/t
~# genisoimage -o image.iso tt
I: -input-charset not specified, using utf-8 (detected in locale settings)
Total translation table size: 0
Total rockridge attributes bytes: 0
Total directory bytes: 106
Path table size(bytes): 10
Max brk space used 0
175 extents written (0 MB)
~# mount -o loop image.iso /mnt/
mount: /dev/loop0 est protégé en écriture, sera monté en lecture seule
~# ls -l /mnt
total 1
-r-xr-xr-x 1 ksh ksh 1 nov.  24 23:38 t
```

3. Synchronisation de données avec rsync

Dans le cadre des stratégies de préservation des données, il peut être utile de répliquer des données d'un serveur sur un autre, soit afin de garantir une disponibilité géographique de données identiques, soit pour se préserver d'une défaillance d'un disque dur ou d'un serveur.

rsync propose plusieurs mode de fonctionnement, mais le plus courant dans le cadre de synchronisation de données est de disposer d'un service rsync sur un serveur et de planifier des synchronisations régulières depuis les machines contenant les données à répliquer.

3.1. Configuration d'un serveur rsync :

Modification du fichier de configuration rsync dans /etc/default

```
RSYNC_ENABLE=true
```

Ce paramètre permet le démarrage automatique ou manuel de rsyncen tant que service.

3.1.1. Création du fichier de configuration /etc/rsyncd.conf

```
uid = login
read only = false
[instance]
  path = répertoire
```

login : Le login au nom duquel les opérations d'écritures seront réalisées sur le serveur.
read only = false : indispensable pour que le service puisse écrire sur le disque.
instance : Nom au choix, il y aura autant d'instance que de clients à répliquer.

Il faudra relancer rsync :

```
/etc/init.d/rsync restart
```

3.1.2. Synchronisation des données depuis un client

La synchronisation se fera à la demande ou depuis une tâche planifiée avec la commande rsync
Syntaxe de la commande rsync :

```
rsync -av --delete /repertoire/ IP@serveur::instance
```

-a : mode archive, réplique les données à l'identique, en préservant les permissions et es propriétaires.
-v : Optionnel, affiche le détail de chaque opération.
–delete : Copie miroir, les données effacées sur le client le sont aussi sur le serveur.
repertoire : Le répertoire des données locales à dupliqués.
instance : Le nom de l'instance paramétrée dans /etc/rsyncd.conf sur le serveur.

3.2. Synchronisation sécurisée de données avec rsync

Si la synchronisation de données doit se faire en environnement hostile, il est possible de s'en remettre à SSH pour le transport des données.

Dans ce mode de fonctionnement, le démon `rsync` ne s'exécute pas sur le serveur et l'exécutable est lancé à la volée par SSH pour toute connexion entrante.
Syntaxe de la commande rsync

```
rsync -av --delete -e ssh repertoire login@serveur:/chemin_cible
```

-a : mode archive, réplique les données à l'identique, en préservant les permissions et es propriétaires.
-v : Optionnel, affiche le détail de chaque opération.
–delete : Copie miroir, les données effacées sur le client le sont aussi sur le serveur.
repertoire : Le répertoire des données locales à dupliqués.
login : Le login au nom duquel les opérations d'écritures seront réalisées sur le serveur.
serveur : Adresse IP du serveur SSH
chemin_cible :Répertoire cible pour la synchronisation de données sur la machine cible.

Je vous encourage à lire le man de la commande rsync ici, car en effet j'ai présenté 1/4 des options possibles de la commande