APPRIVOISEZ VOTRE PINGOUIN !

GNU LINUX PRATIQUE

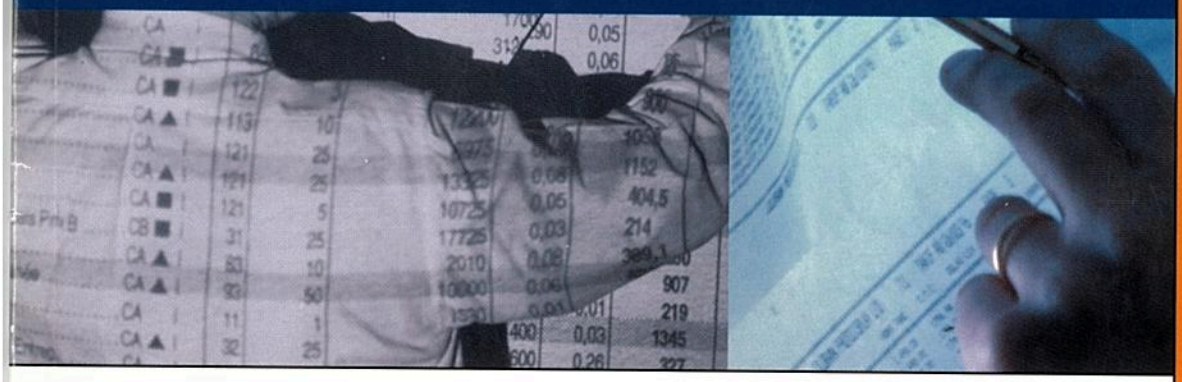
Nouvelle Formule !

32

Novembre Décembre 2005

OpenOffice 2.0

GÉREZ VOS BASES DE DONNÉES AVEC LE NOUVEAU MODULE BASE



SONDAGE – GRAND JEU CONCOURS
20 abonnements à gagner !!
Donnez-nous votre avis sur notre nouvelle formule !
(voir page 3)

sur le CD-ROM

ubuntu

TESTEZ LA NOUVELLE UBUNTU LIVE 5.10 SANS RIEN INSTALLER !

Cahier Web

FRANCE MÉTRO : 6,40 € - DOM 6,95 € -
BEL : 7,30 € - LUX : 7,30 € - PORT. CONT. : 7,30 €
CH : 13 FS - CAN : 12 $ - MAR : 65 DH

L 18864 - 32 - F: 5,95 € - RD

configurer

SERVEUR DE SAUVEGARDES INCRÉMENTALES

Pierre.Lafaye-de-Micheaux@upmf-grenoble.Fr - ddelseny@yahoo.fr

Dans cet article, nous allons vous apprendre comment mettre en place un serveur de sauvegardes incrémentales pour les données d'un petit réseau hétérogène.

1. LE BUT

Avez-vous déjà effacé par mégarde un fichier qui vous était cher ? Ou pire, lancé la commande `rm -rf *` à la place de la commande `rm -rf *temp` en voulant par exemple effacer tous les dossiers dont le nom se termine par `temp` (notez la proximité de la touche `*` et de la touche `[Entrée]` !) ? Plus d'inquiétudes, nous allons vous montrer comment prévenir vaut mieux que guérir !

Remarque : Au cas où le mal serait déjà fait, le petit utilitaire recover (http://recover.sourceforge.net/linux/recover/) pourrait vous être utile.

Nous allons donc vous montrer comment mettre en place votre propre serveur de sauvegardes. Cela peut par exemple être très intéressant pour une petite structure contenant une dizaine d'ordinateurs (ou plus).

Pour cela, vous devez disposer d'un ordinateur (le serveur) équipé d'un système d'exploitation Linux que nous nommerons par la suite « BACKUP ». Vous devez aussi avoir dans votre réseau d'autres ordinateurs (les clients) qui contiendront les données à sauvegarder.

Idéalement, et pour des raisons de sécurité, le serveur contiendra deux disques durs : un disque de faible capacité (disque 1) sur lequel un système Linux minimal sera installé et un second disque de grande capacité (disque 2) qui contiendra les sauvegardes des données des usagers du réseau. La capacité de ce disque devrait être légèrement supérieure à la taille totale de toutes les données que vous voulez sauvegarder. Notez que l'on trouve maintenant des disques durs de capacité supérieure à 300 Go pour un prix d'environ 300 euros.

Le premier disque dur contiendra un système d'exploitation Linux minimal sans gestionnaire de fenêtres graphiques tel KDE ou Gnome, mais fonctionnera uniquement en mode console.

2. LES INGREDIENTS

- `rsync` : petit logiciel permettant de transférer des fichiers sur une machine distante, en synchronisant la cible avec la source.
- `ssh` : permet d'obtenir un *shell* sur une machine distante de façon sécurisée ; peut interagir avec `rsync` pour sécuriser les transferts de fichiers.
- `Cron` : démon permettant d'automatiser des tâches à effectuer à des intervalles de temps réguliers.
- Liens « durs » (`cp -al`) : petite astuce qui nous permettra de sauver beaucoup d'espace disque.

3. LES PRINCIPES

On parle de sauvegarde incrémentale lorsqu'on ne prend en compte, à chaque nouvelle sauvegarde, que les différences avec celle qui la précède. Le serveur de sauvegarde sera une machine dédiée qui contiendra une copie de l'ensemble des fichiers présents sur les différents clients du réseau (en tout cas des fichiers que l'on désire sauvegarder).

Chaque client disposera, sur le serveur, d'un dossier nommé `sauvegardes` dans lequel il copiera, tous les jours, ses données (en fait cela pourra même être fait de manière automatisée, sans aucune intervention de l'utilisateur). Le serveur effectuera alors, de façon programmée, un roulement sur ces fichiers afin d'en conserver des versions journalières, hebdomadaires et mensuelles.

En cas de pépin sur l'une des machines clientes, l'utilisateur de cette machine pourra alors se connecter au serveur (sans avoir besoin de contacter l'administrateur réseau !) pour récupérer le fichier accidentellement effacé tel qu'il était la veille, la semaine dernière ou même le mois dernier.

4.LA PRATIQUE

4.1 Mise en place du serveur

Par la suite, je me baserai sur la distribution Gentoo (**http://www.gentoo.org/**), mais tout ceci est bien entendu possible sur toute autre distribution Linux. Utiliser `rpm` ou `apt-get` à la place de `emerge` sur une Fedora ou une Debian par exemple. Adaptez les commandes selon votre distribution si nécessaire.

Première chose à faire : il faut passer en mode super utilisateur dans une fenêtre terminal ou une console.

Puis vous devez commencer par installer ssh et faire en sorte que le démon ssh soit lancé à chaque démarrage de l'ordinateur :

```
# emerge openssh; emerge ssh
# rc-update add sshd default
# /etc/init.d/sshd start
```

Il vous faut ensuite faire la même chose avec rsync :

```
# emerge rsync
# rc-update add rsyncd default
# /etc/init.d/rsyncd start
```

Puis, il vous faut installer un gestionnaire de tâches automatisées comme dcron par exemple :

```
# emerge dcron
# rc-update add dcron default
# /etc/init.d/dcron start
# crontab /etc/crontab
```

Maintenant que les logiciels nécessaires à notre entreprise sont installés, il reste à configurer le disque dur de grande capacité pour qu'il soit en mesure de recevoir toutes les données des sauvegardes.

On commence par créer une partition primaire de type ext2 (83) sur le disque 2 (hdb pour *hard drive b*) à l'aide de l'utilitaire fdisk. Pour cela, tapez les instructions suivantes :

```
# fdisk /dev/hdb
Command (m for help):
n
p
1
[ENTREE]
[ENTREE]
t
83
w
```

Remarque : éventuellement, il vous faudra remplacer le b de hdb par une autre lettre. Pour cela, utilisez l'instruction suivante et repérez votre deuxième disque dur :

```
# dmesg | grep DISK | grep hd
```

Cela dépend de la façon dont vous avez branché les câbles plats et configuré les disques (en esclave ou en master) dans la tour de votre PC.

Maintenant, on formate cette nouvelle partition.

```
# mke2fs /dev/hdb1
```

Notre disque est maintenant prêt à accueillir tous les dossiers et sous-dossiers nécessaires à la sauvegarde incrémentale des données de votre réseau.

Toutes les données sauvegardées se trouveront dans des sous-répertoires de /mnt/backup:

```
# mkdir /mnt/backup
```

On monte la partition ainsi créée (sans *rebooter* l'ordinateur):

```
# mount /mnt/backup
```

Utiliser aussi votre éditeur de texte favori (vi par exemple) pour rajouter la ligne suivante au fichier /etc/fstab:

```
/dev/hdb1   /mnt/backup   ext2   auto,noexec,nouser,rw   0 0
```

Cela permettra de monter automatiquement la partition /dev/hdb1 sur /mnt/backup en cas de redémarrage accidentel de la machine BACKUP.

On peut maintenant créer l'ensemble des couples groupe/utilisateur qui auront accès à ce système de sauvegardes. Je vais vous montrer comment faire pour un seul groupe/utilisateur. Il vous sera alors très facile d'adapter la procédure pour un nombre plus important de personnes (quitte à faire un petit script bash pour automatiser tout cela !).

On crée un groupe nommé guser1 :

```
# groupadd guser1
```

Ensuite, on crée un utilisateur nommé user1:

```
# useradd -m -d /mnt/backup/user1 user1 -g guser1
```

Et on définit son mot de passe (par exemple backuppl2m) :

```
# passwd user1
New UNIX password: backuppl2m
Retype new UNIX password: backuppl2m
```

On efface quelques fichiers inutiles ajoutés par la commande useradd :

```
# rm /mnt/backup/user1/.*
```

On met ensuite en place un ensemble de répertoires, ainsi que les droits et propriétaires à y associer pour assurer une sécurité optimale, c'est-à-dire pour que seul l'intéressé (et root bien entendu) puisse accéder aux données qui sont les siennes sans voir celles des autres. Notons qu'il est même possible de crypter ses données pour que le super-utilisateur soit lui aussi incapable de voir vos données, mais ce sera pour une autre fois !

Pour ceux qui sont un peu perdus dans ce qui va suivre, merci de lire l'article sur la gestion des permissions des fichiers écrit par Fleur Brosseau dans le LP31.

```
# chgrp guser1 /mnt/backup/user1
# chown root /mnt/backup/user1
# chmod o-rwx /mnt/backup/user1
# mkdir /mnt/backup/user1/sauvegardes
```

```
# mkdir /mnt/backup/user1/jour
# mkdir /mnt/backup/user1/semaine
# mkdir /mnt/backup/user1/mois
# chmod -R o-rwx /mnt/backup/user1/
# chgrp -R guser1 /mnt/backup/user1/
# chmod g+w /mnt/backup/user1/sauvegardes
```

On crée aussi un dossier nommé `.ssh` qui nous servira par la suite (à nous passer du *password*) :

```
# mkdir /mnt/backup/user1/.ssh
# chmod o+rx /mnt/backup/user1
# chgrp -R guser1 /mnt/backup/user1/.ssh
# chmod g+w /mnt/backup/user1/.ssh
```

Maintenant, on automatise tout le processus de façon incrémentale. Nous allons faire en sorte, qu'une fois par mois, semaine puis jour, le contenu de chaque dossier (inférieur d'un point de vue chronologique) soit remonté d'un cran.

Vous noterez l'utilisation de la commande `cp -al` qui permet de créer ce que l'on appelle des « liens durs » (*hard links* en anglais). Créer un lien *hard* vers un fichier consiste en quelque sorte à en créer une copie (sorte de lien symbolique) qui n'utilisera aucune place supplémentaire sur le disque. Il est alors impossible de dire lequel, du fichier ou de sa copie, est l'original. On peut créer plusieurs liens hards vers un même fichier. Le fichier en question existera tant que l'un des liens hards subsistera.

On crée le fichier `/etc/cron.monthly/user1m.cron` qui contient les lignes suivantes :

```
#! /bin/bash
rm -rf /mnt/backup/user1/mois
mv /mnt/backup/user1/semaine /mnt/backup/user1/mois
mv /mnt/backup/user1/jour /mnt/backup/user1/semaine
cp -al /mnt/backup/user1/sauvegardes /mnt/backup/user1/jour
```

Puis, on fait un `chmod u+x /etc/cron.monthly/user1m.cron`

On crée le fichier `/etc/cron.weekly/user1w.cron` qui contient les lignes suivantes :

```
#! /bin/bash
rm -rf /mnt/backup/user1/semaine
mv /mnt/backup/user1/jour /mnt/backup/user1/semaine
cp -al /mnt/backup/user1/sauvegardes /mnt/backup/user1/jour
```

Puis, on fait un `chmod u+x /etc/cron.weekly/user1w.cron`

On crée le fichier `/etc/cron.daily/user1d.cron` qui contient les lignes suivantes :

```
#! /bin/bash
rm -rf /mnt/backup/user1/jour
cp -al /mnt/backup/user1/sauvegardes /mnt/backup/user1/jour
```

Puis, on fait un `chmod u+x /etc/cron.daily/user1d.cron`

Attention ! Il faut que l'heure de la sauvegarde soit antérieure à celle de la semaine, qui doit elle-même être antérieure à celle du jour.

Il pourrait en effet se produire le problème suivant. Admettons que la sauvegarde du mois se fasse tous les 1er, à cinq heures du matin, celle de la semaine tous les lundis à quatre heures et celle du jour à trois heures. Supposons maintenant que l'on modifie par mégarde (sur le PC client) le contenu d'un fichier `toto` le dimanche 31 dans la journée et que le client le sauvegarde (de façon automatique ou non) le soir même sur le serveur dans le dossier `sauvegardes`. Alors, le lendemain, qui est un lundi 1er, le fichier `toto` va d'abord être transféré dans le dossier `jour` à trois heures, puis dans le dossier `semaine` à quatre heures et enfin dans le dossier `mois` à cinq heures.

Au bout du compte, le mauvais fichier `toto` se retrouve dans tous les dossiers !

Pour parer à cette éventualité, vous pouvez utiliser le fichier `/etc/crontab` suivant :

```
# minute   hour   day   month   dayofweek   command
*/15 * * * *     test -x /usr/sbin/run-crons && /usr/sbin/
run-crons
0  *  * * *      rm -f /var/spool/cron/lastrun/cron.hourly
0  5  * * *      rm -f /var/spool/cron/lastrun/cron.daily
0 4  * * 1      rm -f /var/spool/cron/lastrun/cron.weekly
0 3  1 * *      rm -f /var/spool/cron/lastrun/cron.monthly
```

Le format de chaque entrée est :

```
minute(1 à 60) heure(1 à 24)   jour(1 à 31)   mois(1 à 12)
joursemaine(1 à 7)  commande
```

La commande `run-crons` (exécutée toutes les 60/15=4 secondes) vérifie s'il y a des scripts à lancer dans `/etc/cron.[hourly|daily|weekly|monthly]`. Les informations sur le dernier lancement d'un programme sont enregistrées dans le dossier `/var/spool/cron/lastrun` afin qu'un nouveau lancement du même programme ne se produise pas avant la fin du précédent.

La dernière ligne signifie ainsi que les tâches mensuelles seront exécutées tous les jours portant le numéro 1 (c'est-à-dire chaque 1er du mois) à 3 heures (du matin) et 0 minutes.

Tapez enfin l'instruction :

```
# crontab /etc/crontab
```

Remarque : Vous pouvez changer le degré de finesse de vos sauvegardes en rajoutant par exemple un dossier pour chaque jour de la semaine et pour chaque mois de l'année. Il pourrait aussi être judicieux d'ajouter des quotas disque pour chacun des utilisateurs afin qu'ils ne grignotent pas votre disque de sauvegarde en quelques jours avec leurs photos et leurs vidéos !

4.2 Installation côté client

Pour les clients Linux

Créez, sur chaque poste client, le répertoire /home/backup et changez le propriétaire de ce fichier en lui donnant le vôtre (qui idéalement devrait être le même que celui sur la machine BACKUP) :

```
# chown user1 /home/backup
# chgrp users /home/backup
```

À partir de maintenant, vous copierez toutes les données (du poste client) que vous voulez sauvegarder dans /home/backup.

Créez dans /usr/sbin le fichier nommé sauvegardes.cron qui contient les deux lignes :

```
date  > /home/user1/rsync.log
rsync -e ssh -av --delete  --hard-links --progress "/home/backup/" \ user1@BACKUP.domaine.com:./sauvegardes/ >> rsync.log
```

Remplacez bien entendu BACKUP.domaine.com par le nom de votre machine BACKUP (ou par son adresse IP donnée par la commande ifconfig | grep inet).

Ensuite, on rend ce fichier exécutable par l'utilisateur user1 :

```
# chown user1 /usr/sbin/sauvegardes.cron
# chgrp users /usr/sbin/sauvegardes.cron
# chmod u+x /usr/sbin/sauvegardes.cron
```

Ainsi, à chaque fois que l'on voudra sauvegarder ses données, il suffira de lancer la commande /usr/sbin/sauvegardes.cron dans un terminal.

Enfin, pour que la sauvegarde des fichiers de user1 se fasse automatiquement, sans intervention de sa part :

```
# cp /usr/sbin/sauvegardes.cron /etc/cron.daily
```

Il reste toutefois une dernière chose à faire. En effet, vous pouvez noter que le mot de passe pour se connecter par SSH à la machine BACKUP n'est pas présent dans le fichier sauvegardes.cron. Et il n'est pas souhaitable qu'il nous soit demandé tous les jours. A chaque fois la procédure de sauvegarde automatique se lance !

L'astuce est donc d'utiliser les clés privées et publiques de SSH de la façon suivante :

```
# ssh-keygen -t rsa (et appuyer 3 fois sur [ENTREE])
# cat ~/.ssh/id_rsa.pub | ssh user1@BACKUP.domaine.com 'cat - >> ~/.ssh/authorized_keys'
# ssh root@BACKUP.domaine.com
# chmod go-w /mnt/backup/user1/.ssh
```

Pour les clients Windows

Créez un dossier nommé Sauvegardes dans C:\. Installez le logiciel cwRsync que vous pouvez télécharger ici :

http://prdownloads.sourceforge.net/sereds/cwRsync_2.0.3_Installer.zip

Créez le fichier backup.bat dans le dossier C:\Program Files\cwRsync\bin qui contient les lignes suivantes :

```
@echo off
set PATH=c:\PROGRA~1\cwRsync\bin
date /T >>rsync.log
time /T >>rsync.log
rsync -e ssh -av --delete --progress "/cygdrive/c/Sauvegardes/" user1@BACKUP.domaine.com:./sauvegardes/ >> rsync.log
echo TERMINE
```

Remplacez bien entendu BACKUP.domaine.com par le nom de votre machine BACKUP (ou par son adresse IP donnée par la commande ipconfig tapée dans une fenêtre de commande DOS).

Dans ***Démarrer > Panneau de configuration > Performances et maintenance > Tâches planifiées***, créez une tâche planifiée tous les jours de la semaine à la même heure pour exécuter backup.bat.

Choisir une heure où votre ordinateur est connecté au réseau et où vous ne vous en servez pas. Entre midi et deux, par exemple.

La même procédure que sous Linux concernant les clés privées et publiques de SSH devra être mise en place.

4.3 Accéder à ses fichiers sauvegardés

On espère ne pas en avoir besoin, mais on sera bien content d'avoir mis en place cette méthode le jour où un drame arrivera !

Sous Linux, vous pouvez utiliser sftp (ou scp) pour récupérer vos données, le plus simple étant d'utiliser Konqueror en tapant dans la barre d'adresse de ce navigateur :

```
sftp://user1@BACKUP.domaine.com
```

Sous Windows, il faut installer SSH et SFTP (**http://ftp.ssh.com/pub/ssh/SSHSecureShellClient-3.2.9.exe**). Ensuite, on accède à ses données par SFTP sur BACKUP.domaine.com

```
login: user1
password: backuppl2m
```

Et voilà, comme cela vous ne perdrez plus jamais de fichiers importants !

LIEN

Excellent site en anglais sur le sujet :
http://www.mikerubel.org/computers/wrsync_snapshots/