

Monter automatiquement des disques distants ou locaux avec automount/autofs

31 mai 2014

(dernière révision le 31 mai 2014)

par **Winnt**

1 Introduction

Lorsque l'on veut avoir accès à des disques, qu'ils soient locaux ou distants, la réponse la plus fréquente que l'on lit sur les forums est d'éditer le fichier `/etc/fstab` et d'y inscrire l'accès au disque (local, réseau via NFS ou CIFS).

Cette réponse, bien que pertinente ne prends pas en compte un point important. Le profil de l'utilisateur et la fréquence d'accès au disque.

Si pour un disque local l'inscription directe de celui-ci dans `/etc/fstab` s'avère généralement le plus pertinent, ce n'est pas le cas pour les accès à des disques réseaux via NFS ou CIFS.

En effet, pour un particulier qui comme moi éteint ses ordinateurs pour la nuit, l'enchaînement des manipulations d'extinction est primordial.

Et si cet enchaînement n'est pas respecté, il peut être obligatoire d'éteindre sauvagement un ou plusieurs ordinateurs. ce que les systèmes de fichiers, bien que robustes sous Linux, n'apprécient pas du tout.

Fort de quelques mésaventures en la matière, je me suis penché sur la question. Et j'ai découvert cette petite pépite un peu oubliée — du fait des pseudo systèmes de fichiers que sont gvfs et kio fournis respectivement par Gnome et KDE — qu'est **automount/autofs**.

2 À quoi sert automount/autofs ?

Autofs permet de monter un disque réseau (via NFS ou CIFS), un lecteur cd (quoique udev s'en charge normalement très bien), un disque local non monté par défaut.

Jusque ici rien de bien formidable me direz-vous. C'est là que vous vous trompez.

Car une des particularité d'autofs c'est qu'il démontera tout seul, passé un délai défini dans son fichier de configuration, le disque réseau que vous venez de monter si vous n'y accédez plus.

C'est ce démontage automatique après un délai d'inactivité qui, à mon avis, fait le très grand intérêt de cet utilitaire.

3 Installation

Sous Debian l'installation est très simple puisqu'il suffit de taper la ligne suivante dans le terminal.

Installation d'autofs sous Debian

```
apt-get (ou aptitude) install autofs
```

Une fois ceci fait, il faut maintenant configurer notre outil selon nos besoins.

4 Configurations

La configuration d'autofs utilise un fichier maître nommé **auto.master** qui chargera des fichiers **auto.xxx**. Chaque fichier **auto.xxx** représentant un type d'accès au partage (CIFS ou NFS).

Le nom du fichier contenant les partages n'as pas grande importance. Mais la convention veut qu'il se nomme `auto.<extension indiquant le type de système de fichier>`.

Cette habitude permet de rapidement repérer les divers systèmes de fichiers utilisés ce qui simplifie grandement la maintenance. Et ce « *standard* » permet de voir facilement sur une machine inconnue ce qui est fait.

Il semble qu'il soit difficile d'utiliser plusieurs fichiers `auto.xxx` avec les montages indirects. En effet, en ayant plusieurs fichiers pour des montages indirects j'obtiens un répertoire vide.

J'ai donc opté pour des montages directs. Ce qui n'est guère plus compliqué.

Dans le cas ou le répertoire de montage n'existe pas, celui-ci est temporairement créé par automount.

Toutes les manipulations sur les fichiers de configuration sont effectuée avec l'utilisateur `root` (ou via `sudo` si installé sur votre distribution).

4.1 Configuration d'un partage NFS

Sous Linux, l'une des méthodes les plus simples est d'utiliser NFS pour partager des répertoires ou disques. Je partirai du principe que le partage NFS existe et qu'il est fonctionnel.

Voici les deux fichiers de configuration.

Fichier `auto.master`

```
/- /etc/auto.NFS --ghost,--timeout=60
```

Le fichier maître `auto.master` contient une ligne par type de partage.

La ligne de notre fichier `auto.master` se décompose en 3 parties distinctes :

- `/-` : indique que l'on utilise un montage direct.
- `/etc/auto.nfs` : indique le nom et le chemin d'accès au fichier contenant les instructions de montage des partages.
- `--ghost` : indique à `autofs` de toujours afficher les répertoire qui peuvent être montés.
- `--timeout=60` : limite d'inactivité pour le démontage automatique.

Chaque ligne du fichier **auto.master** doit impérativement comporter ces trois champs. Le dernier champs comprenant des options — sauf `--timeout=xx` qui est indispensable — peut varier selon les systèmes de fichiers ou les besoins.

Ainsi, on pourrait très bien rajouter les `gid` et `uid` de l'utilisateur (sous la forme `uid=<uid utilisateur>`, `gid=<gid utilisateur>` afin de s'assurer qu'il puisse lire et écrire dans le répertoire et les fichiers.

Fichier `auto.nfs`

```
/mnt/Partage -fstype=nfs,rw,intr 192.168.100.100:/home/export
ou
/mnt/Partage -fstype=nfs,rw,intr mon_serveur:/home/export
```

Le fichier de `auto.nfs` contenant nos partages NFS contient une ligne constituée de trois champs.

- `/mnt/Partage` : le répertoire local qui contiendra les fichiers et répertoires auxquels nous voulons accéder.
- `-fstype=nfs` : type de système de fichier (ici NFS).
- `rw` : droits d'écriture et lecture sur le système de fichier (toujours utile non ?).
- `intr` : autorisation d'interruption des requêtes NFS si le serveur est en panne ou ne peut pas être contacté (toujours prudent de le mettre).
- `192.168.100.100:/home/export` : répertoire NFS exporté par la machine `192.168.100.100` (vous pouvez mettre le nom dns directement à la place de l'adresse si vous avez un serveur dns).

Une fois les fichiers de configuration mis à jour, il faut faire recharger la nouvelle configuration à `autofs`.

Rechargement de la configuration d'`autofs` sous Debian

```
/etc/init.d/autofs reload
```

En supposant que le répertoire `/home/export` contienne un fichier nommé `super_serveur.txt`, voici ce que l'on obtient en faisant un `cd /mnt/Partage` suivi d'un `ls` :

```
super_serveur.txt
```

Vous avez désormais facilement accès au répertoire `/home/export` de `mon_serveur` sans d'autre manipulation que vous déplacer dans `/mnt/Partage` si vous êtes en console ou un simple clic si vous utilisez un gestionnaire de fenêtre tel que KDE, Gnome, Xfce...

4.2 Configuration d'un partage FTP

Afin de pouvoir monter un répertoire FTP comme s'il s'agissait d'un répertoire local nous allons utiliser *Fuse* le système de fichiers virtuel en espace utilisateur qui est disponible sur toutes les distributions *Linux*.

Pour ce faire nous installerons *curlftpfs* le système de fichier FTP et *fuse-utils* les outils de montage manuel.

```
apt-get install curlftpfs fuse-utils
```

Pour avoir accès à *Fuse* il nous faut bien sûr être dans le **groupe fuse**.

Vérifions cela avec l'une des commandes suivantes dans une console :

```
groups ou groups <utilisateur> ou encore grep <utilisateur> /etc/group
```

Exécuter la commande suivante dans une console en tant que `root` permet d'ajouter l'utilisateur au groupe `fuse`.

```
adduser <utilisateur> fuse
```

Pour que la nouvelle appartenance au **groupe fuse** soit prise en compte, il faut se déconnecter de la session actuelle, puis se reconnecter.

Une fois reconnecté, notre utilisateur apparaît bien comme appartenant au groupe fuse.

Nous insérons dans notre fichier auto.master la ligne suivante :

Fichier auto.master

```
/- /etc/auto.ftp --ghost,--timeout=60
```

Puis nous créons un fichier auto.ftp et y insérer la ligne suivante :

Fichier auto.ftp

```
/mnt/ftp -fstype=fuse,allow_other,user=login:password \_  
:curlftpfs\#ftp.domaine.tld\:/
```

Attention toute la commande tient sur une ligne le symbole _ indique un retour à la ligne pour une question d'affichage.

Quelques explications concernant la ligne du fichier auto.ftp me semblent utiles.

- /mnt/ftp : répertoire local qui contiendra les fichiers et répertoires du site FTP auquel nous souhaitons accéder.
- -fstype=fuse : système de fichier utilisé.
- allow_other : permet à un utilisateur non root d'accéder à ce partage (en l'occurrence votre utilisateur).
- user=login:password : identifiant et mot de passe de l'utilisateur du site FTP.
- :curlftpfs\#ftp.domaine.tld\:/ : Site FTP auquel vous voulez accéder à l'aide de curlftpfs.

Une fois les fichiers de configuration mis à jour, il faut recharger la nouvelle configuration.

Rechargement de la configuration d'autofs sous Debian

```
/etc/init.d/autofs reload
```

Il ne reste plus qu'à se rendre dans le répertoire /mnt/ftp afin de constater que celui-ci contient bien les fichiers de notre site internet personnel.

4.3 Configuration d'un partage CIFS (Samba)

Tout comme avec NFS et FTP, nous allons mettre en place un montage de répertoire partagé via Samba afin de pouvoir y avoir accès rapidement. Nous utiliserons toujours *Fuse* le système de fichiers virtuel en espace utilisateur dont nous avons vu l'installation en section 4.2 page 3.

Nous éditons notre fichier auto.master afin d'y insérer la ligne suivante (cela devrait maintenant vous être familier).

Fichier auto.master

```
/- /etc/auto.cifs --ghost,--timeout=60
```

Comme précédemment, nous créons un fichier auto.cifs dans lequel nous insérons la ligne suivante :

Fichier auto.cifs

```
/mnt/samba -fstype=cifs,rw,guest ://nom_du_serveur/répertoire_partagé
```

- /mnt/samba : répertoire local dans lequel nous retrouverons les fichiers et répertoires partagés par nom_du_serveur .
- -fstype=cifs : le type de système de fichier fuse (cela dira quelque chose à ceux qui compilent leur noyau ou font joujou avec Samba).
- rw : droits d'écriture et lecture sur le système de fichier.
- guest : nous nous identifions en tant qu'utilisateur guest (invité).
- ://nom_du_serveur/répertoire_partagé cela parle de lui même.

Il peut être utile de rajouter `domain=<workgroup ou domaine` et de s'assurer que le fichier `smb.conf` contient les paramètres suivants :

smb.conf de Samba 4 pour un partage public

```
Section globale :
security = user
map to guest = Bad Password
guest account = <utilisateur invité>

Pour chaque partage :
guest ok = yes
force user = <utilisateur invité>
```

Et une fois de plus, nous relançons `autofs`.

Rechargement de la configuration d'`autofs` sous Debian

```
/etc/init.d/autofs reload
```

Une fois ceci fait, nous devrions voir dans notre répertoire `/mnt/samba` les fichiers présent sur notre serveur Samba.

Si le serveur Samba exige une identification utilisateur et mot de passe quelques modifications sont à apportées au fichier `auto.cifs`.

Il faut remplacer `guest` par l'une des deux lignes suivantes :

- `username=xxx,password=xxx`
- `credentials=/etc/credentials.txt`. Ce fichier contiendra `username=xxx` et `password=xxx` chacun sur une ligne et devra être protégé contre les indiscretions par un `chmod 600 /etc/credentials.txt`.

5 Aïe ça ne marche pas

Si vous rencontrez des problèmes commencez par :

1. arrêter `autofs` avec `/etc/init.d/autofs stop`.
2. commentez toutes les lignes de votre fichier `auto.master` sauf une.
3. ouvrir une autre console et lancer la commande `automount -f -v`¹.
4. Recommencez jusqu'à ce que vous trouviez la ligne qui coince.

Pour arrêter `automount`, il suffit de taper `CTRL+C` dans la console ou il est actif.

Vous pouvez aussi consulter les pages de man suivante :

- `automount`
- `autofs`
- `auto.master`
- `nfs`
- `mount.nfs`
- `mount.cifs`

En procédant à divers tests, je me suis aperçu que pour deux montages ftp absolument identiques (en dehors des identifiants et mots de passe) pour l'un j'obtenais un résultat parfait alors que pour l'autre j'avais un message d'erreur.

Celui pour lequel le transfert d'un fichier me retournai une erreur ne posai aucun souci lors de l'utilisation d'un client ftp traditionnel (chez moi Filezilla). Alors que le même fichier copié dans mon répertoire monté par `curlftpfs` me retournais un fichier de taille zéro et un message.

Mais en vérifiant via Filezilla, le fichier était complet sur le serveur FTP.

Je me suis donc mis à la recherche du pourquoi. Et j'ai découvert qu'il s'agissait d'un bug non encore corrigé qui ne semble pas affecter l'intégrité du fichier.

Ce souci ne touche que `curlftpfs`. NFS n'est pas sujet à ce bug.

1. Cette commande lance `automount` en mode verbeux. Ainsi vous verrez très exactement là où se situe le souci.

6 Conclusion

J'espère qu'avec ce petit tutoriel vous aurez découvert et apprécié ce petit utilitaire qu'est automount/autofs.

Personnellement, la découverte de cet utilitaire m'a permis de revoir et étendre mes partages tout en allégeant mon fichier `/etc/fstab` des répertoires réseaux.

En ne montant mes répertoires réseaux que lors d'un accès et grâce à la faculté d'autofs de démonter un montage à partir d'une durée d'inactivité.

Je m'évite tous les soucis liés à un démontage effectué dans le désordre tel qu'un arrêt forcé de la machine, une perte de données, un contrôle du système de fichiers (et c'est long, très long sur de gros disques)...

Ce tutoriel n'est absolument pas complet pour les raisons suivantes :

- je n'ai pas abordé la possibilité de faire des montages réseaux des répertoires utilisateurs via NFS dans le cas de machine partagées ;
- je n'ai pas abordés les maps indirectes ;
- je n'est pas évoqué SSHFS ;
- et sûrement plusieurs autres choses.

Peut-être compléterai-je ce tutoriel plus tard. Mais pour le moment, il me semble être déjà assez complet pour le but que je m'étais fixé.