

Module 5 : Partage de fichier Linux avec NFS

Objectifs

Pouvoir effectuer le partage de fichier sur un réseau Linux.

Table des matières

.....	1
Introduction.....	2
Considération réseaux.....	2
Configuration d'un serveur NFS.....	3
Installation.....	3
Installation des paquets.....	4
Fichier de configuration.....	4
Le fichier /etc/exports.....	4
Les fichiers /etc/hosts.allow et /etc/hosts.deny.....	6
Démarrage des services.....	7
Démarrage des autres « daemon ».....	7
Vérification que les “daemons” sont démarrés?.....	8
Changement au fichier /etc/exports.....	8
Accès au fichier partagé (mount et umount).....	9
Montage (mount).....	9
“Démontage” (umount).....	9
La commande showmount.....	9
Autres commandes d'administration.....	10

Introduction

Le système de fichier NFS permet aux systèmes Unix de partager les ressources disques. NFS fût développé par « Sun Microsystems » durant les années 1980. Le design qui provenait de « Sun » fût partagé par ces derniers ce qui permis au protocole de devenir un standard et par le fait même, d'éliminer tout conflits d'interopérabilité entre les différents systèmes d'exploitation. Linux, quant à lui, supportait NFS avant même la venue de la version 1.0. Le protocole NFS est donc ce qui permet le partage de fichier entre machine Linux.

Considération réseaux

Avant même d'installer NFS, vous devriez être certain que votre installation réseaux fonctionne adéquatement. Par exemple, on peut au moins tester si le serveur et le client peuvent connecter entre eux correctement.

Ainsi, si l'ordinateur sur lequel vous installez le serveur NFS possède l'adresse IP 172.17.113.1 et que le client possède l'adresse 172.17.113.2, vous devriez être en mesure de recevoir et d'envoyer des paquets par le biais des commandes suivantes :

du serveur	ping 172.17.113.2
du client	ping 172.17.113.1

Configuration d'un serveur NFS

Installation

Dans cette section, nous procéderons à l'installation du serveur et du client. Ainsi, l'installation se fera en deux étapes :

Préparation des fichiers de configuration pour NFS et démarrage des démons liés à NFS.

Il est important de connaître les différents fichiers qui sont contenus dans une installation NFS. Il y a principalement trois (3) programmes qui permettent le fonctionnement adéquat de NFS :

- portmap (Aussi appelé « **rpc.portmapper** », ce programme n'a pas de lien direct avec NFS mais permet de rediriger correctement les requêtes depuis d'autres machines vers le bon « démon » NFS.)
- nfs (Aussi appelé « **rpc.nfsd** », ce démon permet de faire la transformation entre les requêtes NFS et les requêtes sur le système de fichier local.)
- mountd (Aussi appelé « **rpc.mountd** », Ce « démon » s'occupe des requêtes pour « monter » et « démonter » les systèmes de fichier NFS.)

Note

Les programmes « nfs » et « mountd » sont utiles sur la machine qui sert de serveur NFS seulement. En fait, vous trouverez sûrement plus sage de ne pas faire rouler ces programmes sur la machine cliente pour des raisons de sécurité et pour éviter de consommer des ressources système inutilement.

Les clients NFS n'ont pas besoin de programmes spéciaux pour fonctionner. Le seul programme relatif à NFS qui devrait se retrouver sur la machine cliente est le programme « portmap » pour permettre les fonctionnalités des appels de procédures à distance (RPC) aux autres programmes qui nécessitent son usage.

Par défaut, ces programmes sont installés et chargés lors du démarrage du système. Pour le vérifier, utilisez la commande « rpcinfo » comme suit :

- rpcinfo -p

La sortie à l'écran montre tous les programmes qui sont enregistrés auprès du serveur RPC et qui sont en fonction.

Par exemple, sur ma machine :

```
[root@Atlantis root]# rpcinfo -p
```

program	vers	proto	port	
100000	2	tcp	111	portmapper
100000	2	udp	111	portmapper
100005	1	udp	821	mountd
100005	1	tcp	823	mountd
100003	2	udp	2049	nfs
100003	2	tcp	2049	nfs

Si vous ne voyez pas les programmes mountd et nfs, c'est que les "daemon" en questions ne sont pas en fonction ou que les paquetages n'ont pas été installés.

Installation des paquetages

Par défaut, sur Fedora 20, le paquetage nfs-server est déjà installé mais n'est pas actif. Si vous aviez à installer le paquet, la commande à faire serait :

```
sudo yum install nfs-utils-server
```

Démarrage et arrêt du service :

```
sudo systemctl start/stop nfs-server:
```

Pour redémarrer le service nfs :

```
sudo systemctl restart nfs-server
```

Fichier de configuration

Il y a principalement 3 fichiers qui servent à la configuration d'un serveur NFS:

```
/etc/exports,  
/etc/hosts.allow et  
/etc/hosts.deny.
```

Si on n'insiste pas sur la sécurité, seul le fichier "**/etc/exports**" est nécessaire pour donner les indications au serveur NFS.

Le fichier /etc/exports

Ce fichier contient une liste d'entrée. Chaque entrée indique le répertoire (ou volume) qui est partagé ainsi que les options de partage. Nous donnons ici un exemple mais il est fortement conseillé de consulter l'aide en ligne à propos de exports pour connaître les autres options possibles.

Une entrée dans le fichier "/etc/exports" ressemble à:

```
repertoire machine1(option1,option2, etc...) machine2(option1,option2, etc...)
```

où :

répertoire

le répertoire que vous voulez partager. Il est à noter que si vous partager un répertoire, alors tous les répertoires se situant en dessous du répertoire partagé seront également partagés.

machine1 et machine2

Ce sont les machines clientes qui auront accès au répertoire partagé. On peut employer le nom de la machine ou l'adresse IP de cette dernière (ex. : machine.compagnie.com ou 192.168.0.8).

optionx

la liste des options qui s'appliquent à chaque machine. Ces options décrivent le type d'accès que les machines clientes posséderont. Parmi les options importantes, mentionnons :

- **fsid** : Spécifie que le répertoire devient la racine des répertoires exportés. Habituellement, la valeur 0 est utilisée. **fsid=0**
- **ro**: Le répertoire est partagé en lecture seulement; la machine cliente ne pourra pas accéder ce système de fichier en écriture. C'est le défaut.
- **rw**: La machine cliente aura les droits d'écriture et de lecture dans le répertoire.
- **no_root_squash**: Par défaut, toute requête envoyer par l'utilisateur "root" sur la machine cliente est traitée par le serveur comme si la demande venait de l'utilisateur "nobody". Si l'option « no_root_squash » est sélectionnée, l'utilisateur « root » de la machine cliente aura les mêmes droits que l'utilisateur "root" sur le serveur. Ceci peut avoir de sérieuse conséquence au niveau de la sécurité. On applique rarement l'option « no_root_squash » à moins d'avoir de bonnes raisons de le faire.
- **no_subtree_check**: Si on exporte une partie d'un volume, une routine appelée "subtree_checking" vérifie que le fichier qui est demandé par le client est bel et bien dans le bon volume. Si le volume entier est exporté, il vaut mieux désactiver cette option ce qui permettra des transferts plus rapides.

Exemple 1: Partage de 2 répertoires à 2 machines distinctes.

Supposons que nous avons 2 machines clientes qui se nomme "ordi1" et "ordi2" et dont leur adresse IP est respectivement « 192.168.0.1 » et « 192.168.0.2 ».

Nous voulons partager les répertoires qui contiennent les programmes « binaires » sur le système ainsi que les répertoires « maison ». Voici le fichier /etc/exports qui permettra de réaliser les partages demandés:

```
/usr/bin    192.168.0.1(ro,fsid=0) 192.168.0.2(ro,fsid=0)
/home      192.168.0.1(rw,fsid=0) 192.168.0.2(rw,fsid=0)
```

Les machines ayant l'adresse 192.168.0.1 auront accès au repertoire « /usr/bin » en lecture seulement. Tandis que les machines ayant l'adresse 192.168.0.2 auront accès au répertoire « /home » en lecture et en écriture.

Il n'est pas obligatoire d'utiliser les adresses IP, on peut entrer les informations de la machine cliente de 4 façons:

- Par leur nom . (/home machine(ro))
- Par le nom du groupe dans lequel fait parti l'ordinateur. Par exemple, si l'ordinateur client fait parti du groupe "OrdiNet", on pourra écrire: /home @OrdiNet(ro)
- En utilisant le caractère « * ». Ce caractère peut prendre n'importe quelle valeur. Ainsi, pour désigner tout les ordinateurs du réseau linux.org, on peut utiliser : *.linux.org .
- On peut utiliser la combinaison adresse/Masque de réseau. Par exemple, pour spécifier tout les ordinateurs dans le réseau 192.168.42.0 avec un masque de 255.255.255.0, on peut utiliser la notation : 192.168.42.0/24.

Les fichiers /etc/hosts.allow et /etc/hosts.deny

Ces deux fichiers permettent de spécifier quels ordinateurs sur le réseau peuvent utiliser les services sur votre machine. Chaque ligne spécifie un service et une liste d'ordinateur. Lorsque le serveur reçoit une requête d'une machine, le serveur procède aux tests suivants:

- Vérifie si le client se retrouve dans le fichier "hosts.allow". Si oui, le client aura accès.
- Si la machine cliente n'a pas d'entrée dans le fichier "host.allow", le serveur va alors vérifier si ce client est dans le fichier "hosts.deny". Si oui, ce client se verra refuser l'accès au service demandé.
- Si le client n'est pas dans l'un de ces fichiers, alors l'accès est accordé.

En général, c'est une bonne idée avec NFS (comme avec la plupart des services Internet), d'explicitement interdire l'accès au service que vous n'utilisez pas.

La première étape consiste à entrer la ligne suivante dans le fichier "/etc/hosts.deny" :

```
portmap:ALL
```

Avec la venue de la version 0.2.0 de nfs-utils, vous pouvez même contrôler l'accès au "démon" individuellement. Voici un exemple:

```
lockd:ALL  
mountd:ALL  
rquotad:ALL  
statd:ALL
```

Même si vous avez une version plus vieille de nfs-utils, l'ajout des lignes précédentes ne causera aucun dommage à votre serveur même si ces entrées ne sont pas reconnues. Certains administrateurs systèmes vont même jusqu'à placer la ligne

```
ALL:ALL
```

dans le fichier "/etc/hosts.deny", ce qui a pour effet de bloquer tous les services à toutes les machines clientes qui tentent d'accéder à ces services.

Ensuite, il s'agit d'ajouter les machines dans le fichier "/etc/hosts.allow" pour lesquelles on veut donner accès aux différents services. Le format de ce fichier est le suivant:

```
service: host [or network/netmask] , host [or network/netmask]
```

Supposons que nous voulions donner accès aux machines (machine1.foo.com et machine2.foo.com) et supposons que les adresses IP de ces machines sont: 192.168.0.1 et 192.168.0.2 respectivement. On peut alors ajouter les lignes suivantes dans le fichier /etc/hosts.allow:

```
portmap: 192.168.0.1 , 192.168.0.2
```

Démarrage des services

Démarrage des autres « daemon ».

Les autres "daemon" sont : nfs ou rpc.nfsd, qui fait le plus gros de l'ouvrage, rpc.lockd and rpc.statd, qui s'occupe des accès aux fichiers; rpc.mountd, qui s'occupe des requêtes traitants des montages de partition NFS et rpc.rquotad, qui gère les "quotas" sur les volumes exportés.

Dans les versions plus récentes de Linux, on peut maintenant partir d'un seul coup tout les "daemons" associés au service NFS avec la commande:

```
systemctl start nfs-server
```

Vérification que les “daemons” sont démarrés?

Pour ce faire, lancer une requête au “daemon” portmapper avec la commande

```
rpcinfo -p
```

Vous devriez recevoir une sortie qui ressemble à ceci :

```
program vers proto  port  service
100000    4   tcp    111  portmapper
100000    3   tcp    111  portmapper
100000    2   tcp    111  portmapper
100000    4   udp    111  portmapper
100000    3   udp    111  portmapper
100000    2   udp    111  portmapper
100024    1   udp   47246 status
100024    1   tcp   44866 status
100003    3   tcp    2049 nfs
100003    4   tcp    2049 nfs
100227    3   tcp    2049 nfs_acl
100003    3   udp    2049 nfs
100003    4   udp    2049 nfs
100227    3   udp    2049 nfs_acl
100021    1   udp   39629 nlockmgr
100021    3   udp   39629 nlockmgr
100021    4   udp   39629 nlockmgr
100021    1   tcp   40940 nlockmgr
100021    3   tcp   40940 nlockmgr
100021    4   tcp   40940 nlockmgr
100011    1   udp     875 rquotad
100011    2   udp     875 rquotad
100011    1   tcp     875 rquotad
100011    2   tcp     875 rquotad
100005    1   udp   20048 mountd
100005    1   tcp   20048 mountd
100005    2   udp   20048 mountd
100005    2   tcp   20048 mountd
100005    3   udp   20048 mountd
100005    3   tcp   20048 mountd
```

Si vous ne voyez pas au moins une ligne avec le nom "portmapper", une ligne avec le nom "nfs" et une ligne qui possède le nom "mountd" alors vous devez repartir les “daemons” et essayer la séquence de nouveau.

Changement au fichier /etc/exports

Dès que le fichier /etc/exports est modifié, on doit en aviser les « daemons » en faisant la commande :

```
exportfs -r
```

Ceci signale aux « daemons » nfs et mountd qu’il faut relire les informations contenues dans le fichier /etc/exports.

Accès au fichier partagé (mount et umount)

Montage (mount)

Pour monter un système de fichier NFS distant, utilisez la commande:

```
mount nom_du_serveur:/repertoire/a/exporter point_de_montage
```

où :

nom_du_serveur est le nom du serveur duquel on veut exporter les fichiers,

repertoire/a/exporter est le nom du répertoire qui figure dans le fichier `/etc/exports` et

point_de_montage est le répertoire local sur votre machine à partir duquel vous voulez accéder aux fichiers.

Par exemple :

Je veux monter un répertoire nommé « `/usr/jeux/xbill` » de la machine « `castor` » dans le répertoire « `/jeux` » de la machine local. Il faut taper:

```
mount castor:/usr/jeux/xbill /jeux
```

Les mêmes options que pour le montage de partitions peuvent être utilisées.

“Démontage” (umount)

Pour « démonter » un répertoire NFS, utilisez la commande « **umount** ». Par exemple, pour « démonter » le répertoire monter précédemment, faites la commande suivante :

```
umount /jeux
```

La commande showmount

Cette commande permet d'interroger un hôte distant sur les services NFS qu'il offre, et notamment les volumes qu'il exporte.

showmount -e AdresseIP_ou_NomIP lancée à partir d'un client nous affichera la liste des ressources offertes par *sAdresseIP_ou_NomIP* (=serveur).

Sur le serveur, **showmount -a** nous affichera la liste des clients connectés sur chacune de nos ressources.

De même, sur le serveur, la command **showmount -e** affiche le liste des partages en cours.

Autres commandes d'administration

rpcinfo : (par exemple **rpcinfo -p** consulte le catalogue des applications RPC (nfsd, mountd sont des applicatifs RPC parmi d'autres).

nfsstat : fournit des statistiques d'utilisation de NFS.

La commande **exportfs** permet elle aussi d'obtenir la liste des partages en cours, de relancer le service (pour la prise en compte d'éventuelles modifications du fichier `/etc/exports`, voir même d'effectuer un partage à la volée (sans passer par `/etc/exports`).

exportfs -v affiche les partages en cours.

exportfs -r active les changements fait dans le fichier de configuration de partage NFS (il fait relire le fichier `/etc/exports` par le programme serveur).