

Admin Linux fiche : NFS

1 NFS - PARTAGE DE RÉPERTOIRES

NFS est un système de fichiers en réseau permettant de partager des données. Il permet aux clients d'avoir accès à un réseau. Nous allons voir ici comment rendre accessibles différents répertoires, publics et privés, via le protocole NFS.

1.1 Installation

1.1.1 Du côté serveur

Avant de pouvoir commencer à partager des fichiers il faut installer le module nous permettant de partager les dossiers aux clients.

```
apt-get install nfs-kernel-server nfs-common
```

nfs-kernel-server permet la gestion du serveur tandis que nfs common prend en charge les données partagées entre la machine serveur et la machine client.

relancer le service

Après avoir défini vos partages dans le fichier **/etc/exports** il suffit de relancer le service nfs:

```
sudo service nfs-kernel-server restart
```

Cette commande ne coupe pas les transferts en cours si la nouvelle configuration permet toujours leur accès au serveur. Vous pouvez donc la lancer plus ou moins à n'importe quel moment.

Pour vérifier que l'export a bien eu lieu, taper sur le serveur NFS la commande :

```
showmount -e
```

1.1.2 Du côté client

Du coté client il faut installer le module nfs-common et portmap.

```
apt-get install nfs-common portmap.
```

1.2 Configuration

1.2.1 Du côté serveur

Les 3 fichiers de configuration à modifier du côté serveur permettant le partage de fichiers sont /etc/exports, /etc/hosts.allow, /etc/hosts.deny. On peut modifier que /etc/exports pour partager des fichiers mais cela rendra le serveur très fragile aux attaques.

/etc/hosts.allow et /etc/hosts.deny spécifient qui a accès au serveur et qui ne l'a pas.

/etc/exports est composé de ligne d'entrée indiquant quel dossier est partagé et à quel machine.

```
/Dossier/APartager/ ip_master(rw, sync) ip_trusty(rw, sync)
```

Master et trusty correspondant aux machines à laquelle le dossier est partagé, pour plus de précision il est conseillé d'utiliser les IP des machines. Rw et no_root_squash sont des options qui décrivent l'accès qu'a à l'utilisateur au fichier, les plus importantes sont :

-ro : read only.

-rw : read write, l'utilisateur peut modifier le fichier.

-sync : l'accès au partage nfs se fasse de manière synchrone.

Pour rendre les lignes de configurations effectives il faut redémarrer le nfs kernel server avec la commande :

```
nfs-kernel-server restart
```

1.2.2 du coté client

Il y a 2 moyens de monter les fichiers chez le client on peut le faire manuellement avec mount, ou installer un module qui le fait automatiquement (autofs)

Coté client version manuelle :

Disons que nous avons un dossier home disponible sur le serveur nfs et qu'on voudrait le monter chez un client. On rentrera dans le terminal une ligne de code de ce genre :

```
mount -t nfs -o ro,soft,intr adresse_ip_serveur:/home client/ usrXX/nfs_public
```

Le dossier nfs_public doit être créé préalablement

-t signifie qu'on va spécifier le type de filesystem

-intr : Le processus de montage

-soft : va tenter de se connecter pendant un temps défini et en cas d'échec renvoie une erreur

-hard: va tenter de se connecter et ne s'arrêtera pas tant qu'il n'aura pas réussi.

Le dossier dans lequel on va partager le dossier du serveur doit être créé avant le montage.

On peut démonter le dossier partagé chez le client en utilisant la commande **umount**.

1.3 Monter les fichiers automatiquement avec autofs

AutoFs est un programme qui monte automatiquement les dossiers, il résout les problèmes du montage manuel ou `mount_fstab` tel que le fait que les dossiers restent montés et prennent de la ressource même lorsqu'ils ne sont pas utilisés ou encore que sans wifi l'auto-montage ne s'exécute pas.

Installation -> `apt-get install autofs`

Fichiers à configurer :

1. `/etc/auto.master`

`auto.master` est le fichier dans lequel il faut déclarer le répertoire parent de montage et le type de système de fichier

Une ligne de configuration doit être écrite sous ce format

```
/<point_de_montage_parent> /etc/auto.<type> --ghost,--timeout=30
```

où `point_de_montage_parent` est le dossier dans lequel on va monter les fichiers reçus du serveur nfs.

`type` : le type de système de fichier qu'il s'agit.

`--ghost` : créer des dossiers vides pour chaque point de montage

`--timeout`: indique le temps d'attente avant de démonter les fichiers

exemple: `/net /etc/auto.nfs --ghost,--timeout=30`

2.Solution au problème posé :

Le problème posé par le professeur était de partager à plusieurs utilisateurs un dossier public en read-only et que chaque utilisateur connecté au serveur ait son dossier privé seulement accessible à l'utilisateur lui-même.

Du côté client :

_____ -Nous avons créer des utilisateurs sur la machine client avec adduser et chaque utilisateur créé à un UID unique qui va permettre de le reconnaître sur la machine serveur.

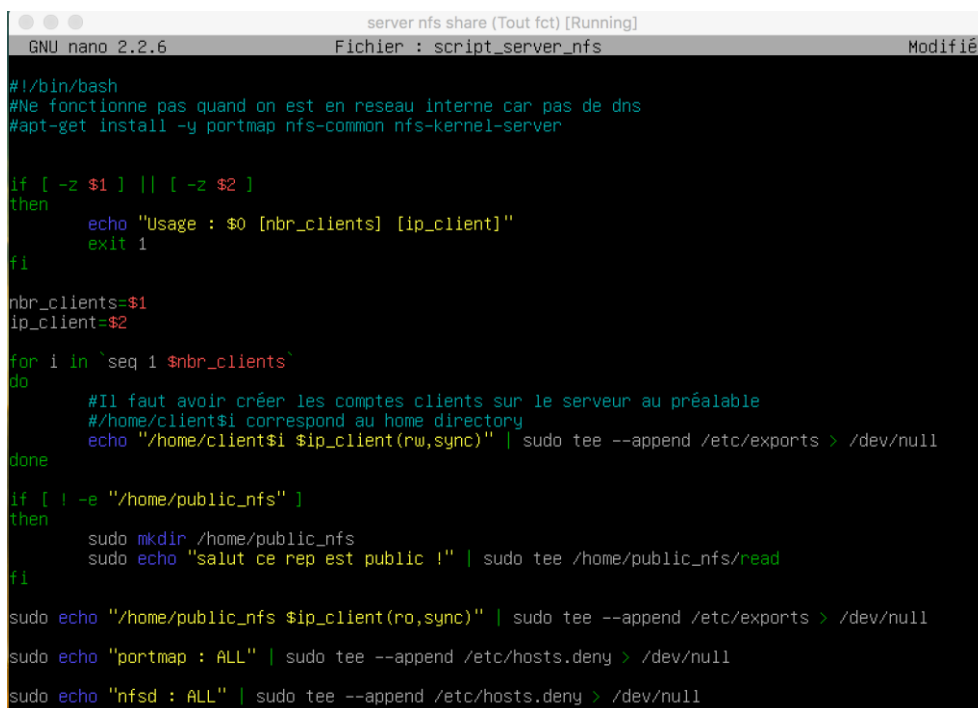
Du côté serveur :

_____ -Nous avons aussi créer des utilisateurs qui ont un UID correspondant à ceux créés du côté client.

-Script qui crée un dossier public et qui mets un fichier dedans, on ajoute ensuite une ligne de configuration dans /etc/exports qui partage le dossier en read only à l'IP de la machine client.

- Ensuite le script ajoutera une ligne de configuration, pour chaque utilisateur de la machine client, dans /etc/exports qui partagera le dossier de l'utilisateur créé sur le serveur qui a un UID correspondant à celui sur le client en rw.

-Et pour finir nous rajoutons dans /etc/hosts.deny et /etc hosts.allow des lignes de configurations qui vont donner l'accès à seulement l'ip de la machine cliente.



```
server nfs share (Tout fct) [Running]
GNU nano 2.2.6      Fichier : script_server_nfs      Modifié

#!/bin/bash
#Ne fonctionne pas quand on est en reseau interne car pas de dns
#apt-get install -y portmap nfs-common nfs-kernel-server

if [ -z $1 ] || [ -z $2 ]
then
    echo "Usage : $0 [nbr_clients] [ip_client]"
    exit 1
fi

nbr_clients=$1
ip_client=$2

for i in `seq 1 $nbr_clients`
do
    #Il faut avoir créer les comptes clients sur le serveur au préalable
    #/home/client$i correspond au home directory
    echo "/home/client$i $ip_client(rw,sync)" | sudo tee --append /etc/exports > /dev/null
done

if [ ! -e "/home/public_nfs" ]
then
    sudo mkdir /home/public_nfs
    sudo echo "salut ce rep est public !" | sudo tee /home/public_nfs/read
fi

sudo echo "/home/public_nfs $ip_client(ro,sync)" | sudo tee --append /etc/exports > /dev/null
sudo echo "portmap : ALL" | sudo tee --append /etc/hosts.deny > /dev/null
sudo echo "nfsd : ALL" | sudo tee --append /etc/hosts.deny > /dev/null
```

```

sudo echo "mountd : ALL" | sudo tee --append /etc/hosts.deny > /dev/null

sudo echo "portmap : $ip_client" | sudo tee --append /etc/hosts.allow > /dev/null
sudo echo "nfsd : $ip_client" | sudo tee --append /etc/hosts.allow > /dev/null
sudo echo "mountd : $ip_client" | sudo tee --append /etc/hosts.allow > /dev/null

sudo /etc/init.d/nfs-kernel-server restart

```

Du coté client :

-On crée des dossiers pour chaque clients sur la machine et on mount pour chaque client son dossier privé.

```

GNU nano 2.2.6      Fichier : script montage
#!/bin/bash

if [ -z $1 ]
then
    echo "Usage : $0 [nbr_clients]"
    exit 1
fi

if [ ! -e /nfs ]
then
    mkdir /nfs
fi

for i in `seq 1 $1`
do
    if [ ! -e /nfs/priv_client$i ]
    then
        mkdir /nfs/priv_client$i
    fi

    mount -t nfs -o rw,soft,intr "192.168.0.1:/home/client$i" "/nfs/priv_cl$

done

if [ ! -e /nfs/public_nfs ]
then
    mkdir /nfs/public_nfs
fi
mount -t nfs -o ro,soft,intr 192.168.0.1:/home/public_nfs /nfs/public_nfs

```

3.NFS VS SAMBA :

Comme dit plus haut, nfs est surtout utilisé dans le monde Unix et permet le partage de dossier. Samba lui est utilisé afin d'interconnecter le monde Windows au monde Unix.

La différence est que nfs ne nécessite pas l'utilisateur d'un utilisateur et d'un mot de passe mais bien juste sur base d'une identification d'hôte et non pas d'utilisateur.

Nfs devient dès lors très complexe si on doit bien séparer les dossiers par rapport au home/utilisateurs et l'utilisation de Samba devient plus efficace.

Par contre s'il s'agit de logiciels, CD-ROM ou encore des mises à jour, il est plus simple d'utiliser NFS qui reste très simple d'utilisation dans ces cas ci.