



Network File System

# Table des matières

1. Généralités .....	2
2. Installation .....	3
3. Configuration du serveur .....	4
3.1. Le fichier /etc/exports .....	4
3.2. Permissions sur les ressources .....	4
3.3. Cas concrets .....	5
3.4. La commande exportfs .....	5
3.5. La commande showmount .....	6
4. Configuration du client .....	7

NFS (Network File System) est un système de partage de fichiers via montage réseau.

# Chapitre 1. Généralités

NFS est basé sur un fonctionnement client-serveur : le serveur met à disposition des ressources pour tout ou partie du réseau (clients).

Le dialogue entre les clients et le serveur se fait grâce aux services RPC (Remote Procedure Call ou procédure d'appel à distance).

Les fichiers distants sont montés dans un répertoire et apparaissent comme un système de fichiers local. Les utilisateurs clients accèdent en toute transparence aux fichiers partagés par le serveur, en parcourant les répertoires comme s'ils étaient locaux.

## Chapitre 2. Installation

2 services sont nécessaires au fonctionnement de NFS :

- Le service network ;
- Le service rpcbind.

L'état des services peut être visualisé par les commandes :

```
service rpcbind status  
service network status
```

Si le paquet nfs-utils n'est pas installé :

```
yum install nfs-utils
```

Le paquet nfs-utils nécessite plusieurs dépendances dont nfs-utils-lib et rpcbind pour s'installer.

Le service NFS peut être démarré :

```
chkconfig nfs on  
service nfs start
```

L'installation du service NFS crée deux utilisateurs :

- nfsnobody : utilisé lors des connexions anonymes ;
- rpcuser : pour le fonctionnement du protocole RPC.

## Chapitre 3. Configuration du serveur



Les droits du répertoire et les droits NFS doivent être cohérents.

### 3.1. Le fichier `/etc/exports`

Le paramétrage des partages de ressources s'effectue dans le fichier `/etc/exports`. A une ligne de ce fichier correspond un partage NFS.

*Syntaxe du fichier `/etc/exports`*

```
/partage client1(permissions) client2(permissions)
```

- **/partage** : Chemin **absolu** du répertoire partagé ;
- **clients** : Machines autorisées à accéder aux ressources ;
- **(permissions)** : Permissions sur les ressources.

Les machines autorisées à accéder aux ressources peuvent être déclarées par :

- Adresse IP : 192.168.1.2
- Adresse réseau : 192.168.1.0/255.255.255.0 ou au format CIDR 192.168.1.0/24
- FQDN : `client_*.formatux.lan` : autorise les FQDN commençant par `client_` du domaine `formatux.lan` ;
- `*` pour tout le monde.

Plusieurs clients peuvent être spécifiés sur la même ligne en les séparant par un espace.

### 3.2. Permissions sur les ressources

Il existe deux types de permissions :

- `ro` : lecture seule ;
- `rw` : modification.

Si aucun droit n'est précisé, alors le droit appliqué sera lecture seule.

Par défaut les UID et GID des utilisateurs du client sont conservés (excepté pour `root`).

Pour forcer l'utilisation d'un UID ou d'un GID différent de l'utilisateur qui écrit la ressource, il faudra spécifier les options `anonuid=UID` et `anongid=GID` ou donner un accès anonyme aux données avec l'option **all squash** (squash dans sens d'écraser).



Il existe un paramètre, `no_root_squash`, qui permet d'identifier l'utilisateur root du client comme étant celui du serveur. Ce paramètre peut être dangereux pour la sécurité du système.

Par défaut, c'est le paramètre `root_squash` qui est activé (même si non précisé), identifiant root comme utilisateur anonyme.

### 3.3. Cas concrets

- `/partage client(ro,all_squash)`

Les utilisateurs du client n'ont accès qu'en lecture seule aux ressources et sont identifiés comme anonyme sur le serveur.

- `/partage client(rw)`

Les utilisateurs du client peuvent modifier les ressources et gardent leur UID sur le serveur. Seul root est identifié comme anonyme.

- `/partage client1(rw) client2(ro)`

Les utilisateurs du poste client1 peuvent modifier les ressources tandis que ceux du poste client2 n'ont qu'un accès en lecture seule.

Les UID sont gardés sur le serveur, seul root est identifié comme anonyme.

- `/partage client(rw,all_squash,anonuid=1001,anongid=100)`

Les utilisateurs du poste client1 peuvent modifier les ressources. Leurs UID sont transformés en 1001 et leur GID en 100 sur le serveur.

### 3.4. La commande `exportfs`

La commande `exportfs` (exported file systems) permet de gérer la table des fichiers locaux partagés avec les clients NFS.

*Syntaxe de la commande `exportfs`*

```
exportfs [-a] [-r] [-u partage] [-v]
```

*Table 1. Options de la commande `exportfs`*

Option	Description
<code>-a</code>	Active les partages NFS
<code>-r</code>	Prend en compte les partages du fichier <code>/etc/exports</code>
<code>-u partage</code>	Désactive un partage donné

Option	Description
-v	Affiche la liste des partages

## 3.5. La commande showmount

La commande showmount permet de surveiller les clients.

*Syntaxe de la commande showmount*

```
showmount [-a] [-e] [hôte]
```

*Table 2. Options de la commande showmount*

Option	Description
-e	Affiche les partages du serveur désigné
-a	Affiche tous les partages en cours sur le serveur

Cette commande permet aussi de savoir si le poste client a l'autorisation de monter les ressources partagées.



"showmount" trie et supprime les doublons dans les résultats (sort|uniq), il est donc impossible de déterminer si un client a fait plusieurs montages d'un même répertoire.



## Chapitre 4. Configuration du client

L'accès aux ressources partagé d'un serveur NFS se fait par point de montage sur le client.

Si besoin, créer le dossier local pour le montage :

```
[root]# mkdir /mnt/nfs
```

Lister les partages NFS disponibles du serveur :

```
[root]# showmount -e 172.16.69.237  
/partage *
```

Monter le partage NFS du serveur :

```
[root]# mount -t nfs 172.16.69.237:/partage /mnt/nfs
```

Le montage peut aussi être automatisé au démarrage du système dans le fichier `/etc/fstab` :

```
[root]# vim /etc/fstab  
172.16.69.237:/partage /mnt/nfs nfs defaults 0 0
```