

INSTALLATION D'UN SERVEUR NFS

http://fr.wikipedia.org/wiki/Network_File_System
<http://web.mit.edu/rhel-doc/3/rhel-rg-fr-3/ch-nfs.html>

I) Serveur NFS

I-A) Installation du Serveur « NFS »

Notre Serveur a pour adresse IP : 192.168.10.254 / 24

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

I-B) Configuration du Serveur « NFS »

Je vais créer un répertoire « nfs » partagé dans « /var ».
Mais j'aurais bien pû le créer ailleurs, comme dans « /home ».

```
# mkdir /var/nfs  
# chown -R nobody:nogroup /var/nfs  
# chmod -R 755
```

On va créer le partage et mettre les droits sur le dossier partagé et quels utilisateurs pourront se connecter dans le fichiers
« **/etc/exports** »

```
# emacs /etc/exports &  
/var/nfs 192.168.10.0/24(rw,sync,no_subtree_check) # Si vous voulez que l'accès soit en lecture/écriture.  
/var/nfs 192.168.10.0/24(ro,sync,no_subtree_check) # Si vous voulez que l'accès soit en lecture seulement.
```

Vous pourrez trouver des explications sur les options pour le fichier « exports » sur le lien suivant :

<http://manpages.ubuntu.com/manpages/jaunty/fr/man5/exports.5.html>

Sinon, vous avez le fichier « **NFS Options du Fichier Exports.pdf** » dans le dossier « **Annexe** ».

Prise en compte des modifications du fichier « /etc/exports ».

```
# exportfs -a
```

Préparation du test, on va créer un fichier et voir si le client pourra le voir.

```
# cd /var/nfs  
# touch test.txt
```

I-C) Sécurisation du Serveur « NFS »

I-C-1) D'abord Tout Interdire

Editez le fichier « **/etc/hosts.deny** » et copiez les lignes suivantes :

```
# Tout interdire sur portmap, nfsd et mountd  
portmap:ALL  
nfsd:ALL  
mountd:ALL
```

I-C-2) Autoriser les échanges voulus

Editez le fichier « **/etc/hosts.allow** » et copiez les lignes suivantes :

```
#Autoriser ce client à se connecter aux services (essentiellement : portmap, nfsd et mountd) :
portmap: 192.168.1.2
lockd: 192.168.1.2
nfsd: 192.168.1.2
mountd: 192.168.1.2
rquotad: 192.168.1.2
statd: 192.168.1.2
```

NB : Il est possible d'autoriser plusieurs clients.

En définissant une plage, par exemple : 192.168.0.0/192.168.0.20

En listant les IP locales autorisées (séparées par des espaces), comme par exemple : 192.168.0.1 192.168.0.3 192.168.0.8

Ou encore en utilisant la syntaxe 192.168. qui autorise toutes les machines dont l'IP locale commence par 192.168. à se connecter.

II) Client(s) NFS

Client (DHCP 192.168.10.0 / 24)

II-A) Installation des outils « NFS » pour le client

En principe, par défaut, les outils sont installés, sinon, vous pouvez par précaution faire l'installation.

```
# apt-get install nfs-common
```

II-B) Monter le Dossier Partagé via NFS

II-B-1) Via Ligne de Commande

Attention, le fait de « monter » le partage via la ligne de commande, il sera démonter automatiquement quand le serveur redémarrera.

Il vous faut créer un dossier dans le lequel vous allez « pointer » (mettre) le dossier partagé.

```
# mkdir -p /mnt/nfs/var/nfs
```

Ensuite, il vous faut « monter » le dossier partagé.

```
# mount 192.168.10.254:/var/nfs /mnt/nfs/var/nfs
```

Allons vérifier si nous avons bien le partage de « monter » et si nous voyons le fichier « test.txt »

```
# cd /mnt/nfs/var/nfs
```

```
# ls
```

Pour voir le montage nfs

```
# mount
```

Pour voir l'espace disque utilisé

```
# df -h
```

Pour démonter le montage nfs, il faut le faire en dehors du point de montage.

```
# umount /mnt/nfs/var/nfs
```

II-B-2) Via le fichier « /etc/fstab »

Configurer le fichier « **/etc/fstab** » permettra d'avoir le « montage » du partage à chaque démarrage.

La structure à mettre dans le fichier est ci-dessous, je rajoute un « montage » sur même « partage NFS » sous 2 formes.

[Partage NFS]	[Point de Montage]	[Type de fichier]	[Options]
192.168.10.254:/var/nfs	/mnt/nfs/var/nfs	ext4	defaults
192.168.10.254:/var/nfs	/mnt/nfs/var/nfs	ext4	rw,suid,dev,exec,auto,nouser,async

Les options sont séparés par une virgule et sans espace.

Ci-dessous, une liste des « **Types de fichier** ».

Ext2FS	ext2	Extended File System v2
Ext3FS	ext3	Extended File System v3
Ext4FS	ext4	Extended File System v4
FAT 16	fat	File Allocation Table 16 bits
FAT 32	vfat	File Allocation Table 32 bits
NTFS	ntfs	New Technology FileSystem
XFS	xfs	
ReiserFS	reiserfs	
ISO 9660	iso9660	Format des CDROM
HFS		Format de MacOS

Et il y a également adfs, affs, autofs, coda, coherent, cramfs, devpts, efs, hpfs, jfs, minix, msdos, ncpfs, nfs, proc, qnx4, ramfs, romfs, smbfs, sysv, tmpfs, udf, ufs, umsdos, xenix, xfs, xiafs.

Ci-dessous, une explication des options.

defaults	Correspond a rw,suid,dev,exec,auto,nouser et async
discard	Compatible ext4 SSD seulement, active le TRIM
auto	Montage automatique lors d'un appel mount -a (par défaut)
noauto	Pas de montage automatique
nouser	Seul le compte root peut monter/démonter le système de fichier (par défaut)
user	Autorise l'utilisateur courant à monter/démonter le système de fichier. Ceci entraîne l'utilisation des options noexec, nosuid, et nodev (à moins qu'elles ne soient explicitement surchargées, comme dans une ligne d'option user,exec,dev,suid).
rw	Montage en lecture/écriture
ro	Montage en lecture seule
atime	Met à jour la date à chaque manipulation (par défaut)
noatime	Pas de mise à jour de la date (Augmente la durée de vie de votre SSD)
exec	Autorise l'exécution des programmes (par défaut)
noexec	Pas d'exécution
suid	Les bits SUID et SGID sont pris en compte (par défaut)
nosuid	Les bits SUID et SGID ne sont pas pris en compte
async	Montage asynchrone (par défaut)
sync	Montage synchrone
dev	Interprète les fichiers spéciaux de périphériques présents sur le système (par défaut)
nodev	N'interprète pas les fichiers spéciaux de périphériques présents sur le système de fichiers
uid=	Spécifie le n° du user propriétaire des fichiers (si omis : root) [n° du user]
gid=	Spécifie le n° du groupe propriétaire des fichiers (si omis : root)
umask=	Spécifie les droits d'usage des fichiers et dossiers
dmask=	Spécifie les droits d'usage des dossiers
fmask=	Spécifie les droits d'usage des fichiers
utf8=	Spécifie l'encodage utf8 pour les noms de fichiers
sw	Montage des partitions swap

Explications de fmask, dmask et umask

Comme pour les droits avec « chmod », nous le faisons pour les 3 entités : User (ou Propriétaire) / Group / Others (Les Autres)

fmask sont les droits en octal pour les fichiers.

Afin de connaître la correspondance d'un fmask en droits version chmod, on fait :

666 - {fmask} = Droits chmod pour les fichiers

Exemple : fmask=022
Droits chmod = 666
 - 022
 644

Droits chmod = 644 => rw-/r--/r--

dmask sont les droits en octal pour les dossiers
777 - {dmask} = Droits chmod pour les dossiers

Exemple : dmask=022
Droits chmod = 777
 - 022
 755

Droits chmod = 755 => rwx/r-x/r-x

umask sont les droits en octal pour les fichiers et dossiers, donc il faut prendre le « mask » et faire le calcul pour les fichiers et dossiers.

Exemple : umask=022 => Droits des fichiers = 644 et Droits des dossiers = 755

Voici la version courte pour les « **dmask** » UNIQUEMENT

Valeur Octale	Permissions (VA)	Droits (VF)	chmod
0	Read, Write, Execute	Lecture, Ecriture, Execution	7
1	Read and Write	Lecture et Ecriture	6
2	Read and Execute	Lecture et Execution	5
3	Read Only	Lecture Seule	4
4	Write and Execute	Ecriture et Execution	3
5	Write Only	Ecriture Seulement	2
6	Execute Only	Execution Seulement	1
7	No Permissions	Pas de Droits	0