

# NFS SERVER

- [Installation d'un serveur NFS sur un RAID 1 sur une VM](#)

# Installation d'un serveur NFS sur un RAID 1 sur une VM

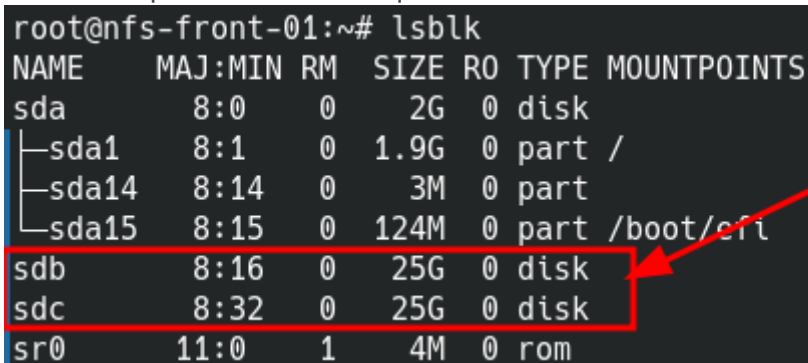
## Prérequis

- Avoir une VM Debian/Ubuntu et pas un container
- 512MB de RAM
- 1 vcpu
- 2 disque supplémentaire pour le RAID

## Mise en place du RAID 1

1. Vérifier la présence des disques:

```
root@nfs-front-01:~# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda         8:0    0   2G  0 disk
├─sda1      8:1    0  1.9G  0 part /
├─sda14     8:14   0    3M  0 part
└─sda15     8:15   0  124M  0 part /boot/efi
sdb         8:16    0  25G  0 disk
sdc         8:32    0  25G  0 disk
sr0        11:0    1    4M  0 rom
```



On voit que les deux disques sont là !!

2. Installer le paquet mdadm

```
apt update && apt install mdadm
```

3. Paramétrer les disques libres dans mon exemple sdb et sdc avec la commande fdisk. Pour l'exemple je le fais sur sdb il faudra aussi le faire sur le deuxième disque (ici sdc).

1. `fdisk /dev/sdb`

2. Entrez l'option `n` pour créer une nouvelle partition.

```
Command (m for help): n
Partition type
  p   primary (0 primary, 0 extended, 4 free)
  e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-104857599, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-104857599, default 104857599):

Created a new partition 1 of type 'Linux' and of size 50 GiB.
```

3. Paramétrer le type de partition, entrer `t` puis `83`

```
Command (m for help): t
Selected partition 1
Hex code or alias (type L to list all): 83
Changed type of partition 'Linux' to 'Linux'.
```

4. Enregistrer en tapant `w`

4. Faire la même chose sur l'autre disque. Vous devriez avoir un résultat comme ça:

```
root@nfs-front-01:~# lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda       8:0    0   15G  0 disk
├─sda1    8:1    0   14G  0 part /
├─sda2    8:2    0    1K  0 part
└─sda5    8:5    0   975M  0 part [SWAP]
sdb       8:16   0   50G  0 disk
└─sdb1    8:17   0   50G  0 part
sdc       8:32   0   50G  0 disk
└─sdc1    8:33   0   50G  0 part
sr0       11:0    1   629M  0 rom
```

5. Création du raid 1

```
root@nfs-front-01:~# mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sdb1
/dev/sdc1
mdadm: Note: this array has metadata at the start and
      may not be suitable as a boot device.  If you plan to
      store '/boot' on this device please ensure that
      your boot-loader understands md/v1.x metadata, or use
      --metadata=0.90
Continue creating array? yes
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
```

On peut voir l'avancée de la création du raid avec les commandes suivante:

```
root@nfs-front-01:~# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 sdc1[1] sdb1[0]
      52393984 blocks super 1.2 [2/2] [UU]
      [===>.....]  resync = 18.7% (9824064/52393984) finish=3.5min
      speed=196481K/sec
```

```

root@nfs-front-01:~# mdadm --detail /dev/md0
/dev/md0:
    Version : 1.2
  Creation Time : Fri Mar  8 17:55:59 2024
    Raid Level : raid1
    Array Size : 52393984 (49.97 GiB 53.65 GB)
  Used Dev Size : 52393984 (49.97 GiB 53.65 GB)
    Raid Devices : 2
  Total Devices : 2
    Persistence : Superblock is persistent

    Update Time : Fri Mar  8 17:58:42 2024
      State : clean, resyncing
  Active Devices : 2
 Working Devices : 2
 Failed Devices : 0
  Spare Devices : 0

Consistency Policy : resync

    Resync Status : 62% complete

    Name : nfs-front-01:0 (local to host nfs-front-01)
    UUID : dad7d1cc:3bed76c7:f30c0a98:bfe003f0
    Events : 10

   Number   Major   Minor   RaidDevice State
    -----   -----   -----   -
    0         8       17         0     active sync  /dev/sdb1
    1         8       33         1     active sync  /dev/sdc1

```

**Tant que le Resync Status n'est pas à 100% on ne peut pas continuer !!**

6. On va formater le nouveau disque en EXT4 (tout laisser par défaut).

```

root@nfs-front-01:~# mkfs.ext4 /dev/md0
mke2fs 1.47.0 (5-Feb-2023)
Discarding device blocks: done
Creating filesystem with 13098496 4k blocks and 3276800 inodes
Filesystem UUID: 57e01ca3-e8a9-4098-b875-b135a85a6103
Superblock backups stored on blocks:

```

```
32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
4096000, 7962624, 11239424
```

```
Allocating group tables: done
```

```
Writing inode tables: done
```

```
Creating journal (65536 blocks): done
```

```
Writing superblocks and filesystem accounting information:
done
```

## Montage du nouveau disque.

1. Créer le point de montage, pour moi ce sera /NFS

```
root@nfs-front-01:~# mkdir /NFS
```

2. Trouver le UUID du nouveau disque

```
root@nfs-front-01:~# blkid |grep md0
/dev/md0: UUID="57e01ca3-e8a9-4098-b875-b135a85a6103" BLOCK_SIZE="4096" TYPE="ext4"
```

3. Ajouter une entrée dans le fichier /etc/fstab:

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# systemd generates mount units based on this file, see systemd.mount(5).
# Please run 'systemctl daemon-reload' after making changes here.
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda1 during installation
UUID=2516b302-d073-4803-bf2f-1e6118ce3c89 / ext4 errors=remount-ro
0 1
# swap was on /dev/sda5 during installation
UUID=0929167e-0fd7-4211-8d49-c025c871d2fa none swap sw
0 0
/dev/sr0 /media/cdrom0 udf,iso9660 user,noauto 0 0
UUID=57e01ca3-e8a9-4098-b875-b135a85a6103 /NFS ext4 defaults
0 0
```

On voit que j'ai récupéré le UUID du disque /dev/MD0 et que je le monte sur /NFS

4. Monter le nouveau disque avec la commande `mount -a` et mettre à jour le système avec `systemctl daemon-reload`
5. Vérifier que le disque est bien monté

```
root@nfs-front-01:~# df -h |grep md0
/dev/md0          49G   24K   47G   1% /NFS
```

## Installation du serveur NFS

1. Installer le paquet NFS:

```
apt install nfs-kernel-server -Y
```

2. Démarrage du service et activation:

```
systemctl enable nfs-kernel-server && systemctl restart nfs-kernel-server
```

3. Paramétrage de la configuration en éditant le fichier `/etc/exports`:

```
# /etc/exports: the access control list for filesystems which may be exported
#                to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes      hostname1(rw,sync,no_subtree_check)
hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4       gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes gss/krb5i(rw,sync,no_subtree_check)
#
#Donner acces a tout le monde
/NFS * *(rw,sync,no_subtree_check)
#Donner acces a une iP
/NFS * 172.16.0.2(rw,sync,no_subtree_check)
```

4. Redémarrer le service NFS

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

## Test de montage sur un serveur distant (DEBIAN)

1. Installer le paquet

```
apt install nfs-common -Y
```

2. Choisir un Point de montage pour moi ce sera `/mnt/nfs`

3. On peut faire le montage de deux façon:

- En ligne de commande mais au prochain redémarrage le montage n'existera plus:

1. Monter le dossier distant:

```
mount -t nfs4 -o 'vers=4.2' 172.16.0.8:/NFS /mnt
```

2. Vérifier:

```
df -h |grep "/mnt"  
172.16.0.8:/NFS          49G   0  47G   0% /mnt
```

On voit bien que le montage est présent.

On peut aussi tester en envoyant un fichier dans le dossier fraîchement monter et voire s'il est bien sur le serveur distant.

- En le mettant dans le fichier fstab ce qui permettre le de le monter à chaque redémarrage système

1. Éditer le fichier fstab

◦ `vim /etc/fstab`

```
172.16.0.8:/NFS /NFS    nfs  
rw,relatime,vers=4.2,rsiz=1048576,wsiz=1048576,namlen=255,hard  
0 0
```

2. Vérifier:

```
df -h |grep NFS  
172.16.0.8:/NFS          49G   0  47G   0% /NFS
```