2022-12-09: Migration du serveur Mutual

Table of Contents

Présentation

L'objectif est de configurer un serveur Rocky Linux 8.4 (socle2022_8.04) en remplacement du serveur MUTUAL.

Configuration précédente

La configuration du serveur Mutual repose sur **DRBL** (**D**iskless **R**emote **B**oot in **L**inux) un logiciel libre, open source pour gérer le déploiement de système d'exploitation GNU / Linux sur des postes clients.

Un environnement PXE DRBL met en oeuvre :

- un serveur DHCP;
- un serveur TFTP.
- un serveur NFS

Il n'est donc pas utile d'installer toute l'application sur le nouveau serveur, seuls les services (DHCPCD, TFTP et NFS) sont nécessaires pour une exploitation courante des données sauvegardées sur l'espace de stockage `/data`.

Organisation du nouveau serveur

- Avec 2 disques de 1,8 TB en RAID 1 on a 1,8 TB
- Avec 3 disques de 1,8 TB en RAID 5 on a 3,6 TB
- Avec 4 disques de 1,8 TB en RAID 5 on a 5,4 TB

Sur le précédent serveur on a 2398.0 GiB (2,3TiB).

Avec une répartition de 4x on a 5,4 TB d'expace de stockage alors qu'avec une répartion 3×3 on a 3,6 TB.

Pour se donner de la marge il est préférable de donner le maximum d'espace de stockage au serveur Mutualisé.

Configuration réseau

Le nouveau serveur dispose des cartes suivantes:

```
lshw -class network -short
H/W path             Device     Class          Description
=========================================================
/0/101/2/0           ens1f0     network        I350 Gigabit Network
Connection
/0/101/2/0.1         ens1f1     network        I350 Gigabit Network
Connection
/0/101/2/0.2         ens1f2     network        I350 Gigabit Network
Connection
/0/101/2/0.3         ens1f3     network        I350 Gigabit Network
Connection
/0/102/2/0           ens2f0     network        Ethernet Controller X710 for
10GbE
/0/102/2/0.1         ens2f1     network        Ethernet Controller X710 for
```

```
10GbE
/0/103/4/0          ens10f0    network        I350 Gigabit Network
Connection
/0/103/4/0.1        ens10f1    network        I350 Gigabit Network
Connection
/0/103/4/0.2        ens10f2    network        I350 Gigabit Network
Connection
/0/103/4/0.3        ens10f3    network        I350 Gigabit Network
Connection
/0/109/2/0          ens3f0     network        Ethernet Controller X710 for
10GbE
/0/109/2/0.1        ens3f1     network        Ethernet Controller X710 for
10GbE
```

Deux interfaces réseau et un bond constitué de 4 interfaces sont définis sur le serveur Mutual:

1. **bond0** est utilisé pour les montages NFS de l'espace secondaire,
2. **ens10f0** pour l'adresse primaire 10.103.212.14
3. **ens10f1** pour l'adresse secondaire 10.103.212.62 afin que les clients puissent sauvegarder leurs données même lorsqu'on fait un exercice de backup.

Configuration des interfaces

Définir l'interface ifcfg-ens10f0(provisoirement 10.103.212.15/27):

```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=none
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=no
NAME=ens10f0
UUID=db974012-217c-44b6-a199-0f33b940a934
DEVICE=ens10f0
ONBOOT=yes
IPADDR=10.103.212.15
PREFIX=27
GATEWAY=10.103.212.1
DNS1=10.154.59.104
DNS2=10.156.32.33
IPV6_DISABLED=yes
```

Définir l'interface ifcfg-ens10f1 (10.103.212.62/27):

```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
```

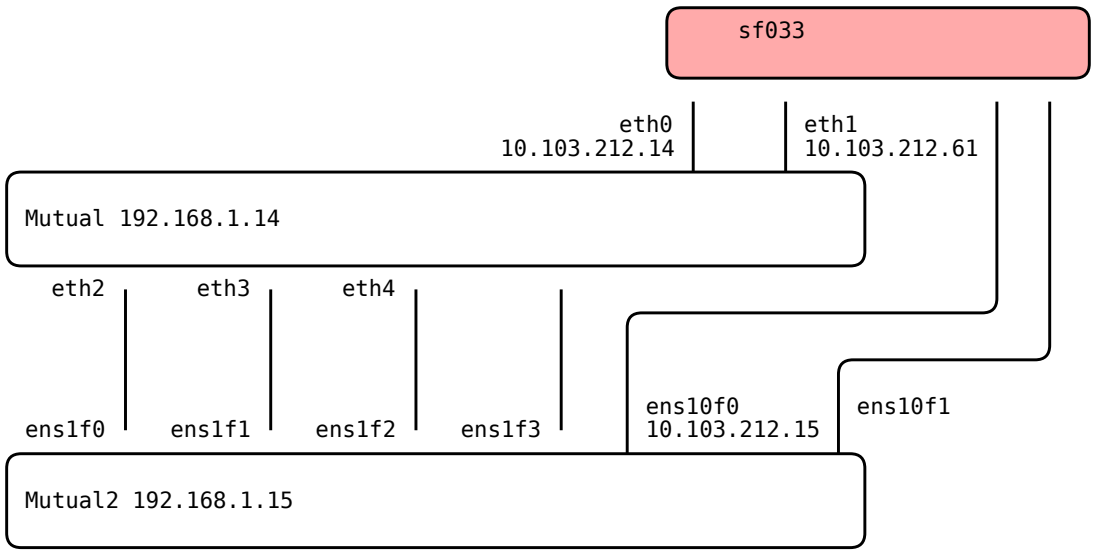
```
BOOTPROTO=static
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=no
IPV6_AUTOCONF=no
IPV6_DEFROUTE=no
IPV6_FAILURE_FATAL=no
NAME=ens10f1
UUID=4ac6c0d1-37a8-4325-b393-b8b79036bf16
DEVICE=ens10f1
ONBOOT=yes
IPADDR=10.103.212.62
PREFIX=27
```

Configuration des liens pour NFS

Le serveurs est configuré afin de partager espace de stockage sur un réseau interne avec l'ancien serveur.

Les ports des cartes filles des deux serveurs sont configurés afin d'établir un lien grande capacité.

Server	logical name	product	vendor	physical id	bus info	serial	capacity
mutual1	eth2	I350 Gigabit Network Connection	Intel Corporation	0	pci@0000:07:00.0	40:f2:e9:1d:66:80	1Gbit/s
	eth3	I350 Gigabit Network Connection	Intel Corporation	0.2	pci@0000:07:00.2	40:f2:e9:1d:66:80	1Gbit/s
	eth4	I350 Gigabit Network Connection	Intel Corporation	0.3	pci@0000:07:00.3	40:f2:e9:1d:66:80	1Gbit/s
mutual2	ens1f0	I350 Gigabit Network Connection	Intel Corporation	0	/0/101/2/0		1Gbit/s
	ens1f1	I350 Gigabit Network Connection	Intel Corporation	0.1	/0/101/2/0.1		1Gbit/s
	ens1f2	I350 Gigabit Network Connection	Intel Corporation	0.2	/0/101/2/0.2		1Gbit/s
	ens1f3	I350 Gigabit Network Connection	Intel Corporation	0.3	/0/101/2/0.3		1Gbit/s



Définir sur chacune des cartes les options suivantes

```
IPV6INIT=no
BOOTPROTO=none
ONBOOT=yes
MASTER=bond0
SLAVE=yes
USERCTL=no
```

Définir le bound0 qui sera utilisé pour les montages NFS des systèmes sur les serveurs distants:

```
cat > /etc/sysconfig/network-scripts/ifcfg-bond0<<'EOF'
DEVICE=bond0
BOOTPROTO=static
IPADDR=192.168.1.15
NETMASK=255.255.255.0
ONBOOT=yes
USERCTL=no
BONDING_OPTS='mode=1 miimon=100 primary=ens1f1'
EOF
```

Redémarrer le réseau:

```
systemctl restart NetworkManager.service
```

Configuration des routes pour les ops

Définir les routes alternatives pour que les serveurs puissent opérer leurs sauvegardes en utilisant l'interface ens10f1 (ex eth1)

```
```

```
cat > /etc/sysconfig/network-scripts/route-ens10f1<< 'EOF'
10.153.150.25/32 via 10.103.212.34 dev ens10f1
10.153.150.27/32 via 10.103.212.34 dev ens10f1
10.153.150.50/32 via 10.103.212.34 dev ens10f1
10.153.150.70/32 via 10.103.212.34 dev ens10f1
10.37.221.50/32 via 10.103.212.34 dev ens10f1
10.37.221.70/32 via 10.103.212.34 dev ens10f1
10.153.150.43/32 via 10.103.212.34 dev ens10f1
10.153.150.45/32 via 10.103.212.34 dev ens10f1
10.37.221.25/32 via 10.103.212.34 dev ens10f1
10.37.221.27/32 via 10.103.212.34 dev ens10f1
10.37.221.43/32 via 10.103.212.34 dev ens10f1
10.37.221.45/32 via 10.103.212.34 dev ens10f1
10.153.113.154/32 via 10.103.212.34 dev ens10f1
10.153.113.35/32 via 10.103.212.34 dev ens10f1
10.153.113.38/32 via 10.103.212.34 dev ens10f1
10.153.113.55/32 via 10.103.212.34 dev ens10f1
10.153.113.56/32 via 10.103.212.34 dev ens10f1
10.153.113.58/32 via 10.103.212.34 dev ens10f1
10.156.251.153/32 via 10.103.212.34 dev ens10f1
10.156.251.154/32 via 10.103.212.34 dev ens10f1
10.153.113.96/32 via 10.103.212.34 dev ens10f1
10.153.113.98/32 via 10.103.212.34 dev ens10f1
10.87.254.50/32 via 10.103.212.34 dev ens10f1
10.67.254.50/32 via 10.103.212.34 dev ens10f1
EOF
```

Configuration de l'espace de stockage

Les images bootables du serveur PXE sont stockées dans le dossier `/tftpboot`.

L'espace de stockage `/data` du serveur mutual comporte les sous dossiers `home`, `ops`, `ops-sav`, `partimag` et `petra`.

Configuration avec LVM/XFS

device	nom du volume	taille	capacité
/dev/vg00/vol_partimag	vol_partimag	1%VG	<50,30 GiB
/dev/vg00/vol_backups	vol_backups	30%VG	1,47 TiB
/dev/vg00/vol_data	vol_data	100%FREE	<3,39 TiB

Créer un volume physique LVM sur le disque raid /dev/sdb:

```
pvcreate /dev/sdb1
  Physical volume "/dev/sdb1" successfully created.
```

```
pvdisplay
```

```
  "/dev/sdb1" is a new physical volume of "4,91 TiB"
  --- NEW Physical volume ---
  PV Name               /dev/sdb1
  VG Name
  PV Size               4,91 TiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               K8oQ3C-hXv8-wbfU-Yfxl-xsGE-oeej-hXTEdz
```

Créer le groupe de volumes vg00:

```
vg create vg00 /dev/sdb1
```

Créer les volumes logiques:

```
lvcreate -n vol_partimag -l 1%VG vg00
lvcreate -n vol_backups -l 30%VG vg00
lvcreate -n vol_data -l 100%FREE vg00
```

Formater les volumes logiques:

```
mkfs.xfs /dev/vg00/vol_partimag
mkfs.xfs /dev/vg00/vol_backups
mkfs.xfs /dev/vg00/vol_data
```

AJouter dans `/etc/fstab` les points de montages des nouveaux volumes:

```
/dev/mapper/vg00-vol_partimag     /partimag                   xfs
defaults        0 0
/dev/mapper/vg00-vol_backups      /nfsshare                   xfs
defaults        0 0
/dev/mapper/vg00-vol_data         /data                       xfs
defaults        0 0
```

Créer les points de montage

```
mkdir /partimag
mkdir /nfsshare
mkdir /data
```

Monter les volumes sur les points de montage:

```

```

```
mount /partimag
mount /nfsshare
mount /data
```

Créer les points de montage :

```
mkdir -p /data/{petra,ops,nfsshare}
mkdir -p /home/petra
```

Pour s'assurer que le système de fichier est bien monté on va créer un drapeau qui n'apparaitra pas quand tous les systèmes sont montés:

`touch /home/petra/ceci-est-cache-quand-le-fs-est-monte`

Configuration avec ZFS

La déduplication est gourmande en mémoire. Lorsque le système ne contient pas suffisamment de RAM, il ne peut pas mettre en cache le DDT dans la mémoire lors de la lecture et les performances du système peuvent diminuer.

La RAM requise dépend de la taille du DDT et de la quantité de données qui seront stockées dans le pool. De plus, plus les données sont dupliquées, moins il y a d'entrées et plus le DDT est petit. Les pools adaptés à la déduplication, avec des taux de déduplication de 3x ou plus (la taille des données peut être réduite à un tiers ou moins), peuvent n'avoir besoin que de 1 à 3 Go de RAM pour 1 To de données. La taille réelle du DDT peut être estimée en dédupliquant une quantité limitée de données dans un pool de test temporaire ou en utilisant `zdb -S` dans une ligne de commande.

Le réglage **vfs.zfs.arc.meta_min (type=LOADER, value=bytes)** peut être utilisé pour forcer ZFS à réserver au moins la quantité de RAM donnée pour la mise en cache des métadonnées.

Pour simplifier l'installation, les distributions basées sur RHEL et CentOS dans le référentiel OpenZFS, un package zfs-release est fourni qui inclut un fichier de configuration et une clé de signature publique.

Tous les packages OpenZFS officiels sont signés à l'aide de cette clé et, par défaut, yum ou dnf vérifiera la signature d'un package avant d'autoriser son installation.

```
dnf install https://zfsonlinux.org/epel/zfs-release.el8_4.noarch.rpm
```

Pour installer des packages de style DKMS, exécuter les commandes suivantes. Ajouter d'abord le référentiel EPEL fourni par DKMS en installant le package epel-release, puis les packages kernel-devel

et zfs.

```
dnf install epel-release
dnf install kernel-devel zfs
```

Pour pouvoir compiler le **dkms** celui-ci doit correspondre à la version du noyau retournée avec la commande `uname -r`.
Mais il arrive que le paquet installé soit en avance, si l'installation du **dkms** s'est bien passée la commande `dkms status` doit retourner la version du noyau avec la mention **installed** comme suit

```
zfs/2.1.7, 4.18.0-305.3.1.el8_4.x86_64, x86_64: installed
```

si ce n'est pas le cas il faut forcer l'installation de la bonne version ainsi:
`yum install "kernel-devel == $(uname -r)"`

Mais il peut arriver que la version attachée au noyau ne soit plus disponible il faut donc récupérer la version des RPM sur les dépots puis l'installer manuellement:

```
wget
https://koji.mbox.centos.org/pkgs/packages/kernel/4.18.0/305.3.1.el
8/x86_64/kernel-headers-4.18.0-305.3.1.el8.x86_64.rpm
yum install kernel-headers-4.18.0-305.3.1.el8.x86_64.rpm
wget
https://koji.mbox.centos.org/pkgs/packages/kernel/4.18.0/305.3.1.el
8/x86_64/kernel-devel-4.18.0-305.3.1.el8.x86_64.rpm
yum install kernel-devel-4.18.0-305.3.1.el8.x86_64.rpm
```

Puis forcer l'installation du bon dkms:

```
dkms --force install zfs/2.1.7
```

Ensuite il faut adopter le lien qui pointe vers la mauvaise référence

```
ln -s /lib/modules/$(uname -r)/build /usr/src/kernels/$(uname -r)/
```

Chaque sous-dossier de `/data` sont associés à un système de fichiers ZFS:

sous dossier btrfs	système de fichiers zfs	Déduplication
/data/ops	pool1/ops	oui[1)]
/data/partimag	pool1/partimag	non
/data/petra	pool1/petra	oui[2)]
/nfsshare	pool1/nfsshare	non

Activer le module zfs:

```
/sbin/modprobe zfs
```

Créer un nouveau pool ZFS pool1 sur le périphérique de stockage /dev/sdb, exécuter la commande suivante :

```
zpool create -f pool1 /dev/sdb
zpool status
  pool: pool1
 state: ONLINE
config:

    NAME        STATE     READ WRITE CKSUM
    pool1       ONLINE       0     0     0
      sdb       ONLINE       0     0     0
```

Créer les systèmes de fichiers dans `pool1/petra`:

```
zfs create pool1/petra
for _rep in eas sns t13a t44c t51a txxa1 li87a st67a t44a t44v t63a; do zfs
create pool1/petra/${_rep}; zfs set dedup=on pool1/petra/${_rep}; done
```

La déduplication n'est pas activée par défaut, il faut activer la déduplication sur les système de fichiers ZFS comme suit : `zfs set dedup=on pool1/petra/${_rep}`

Créer les systèmes de fichiers dans `pool1/ops`:

```
zfs create pool1/ops
for _rep in o13b o44a o44c o44e o51b o63b o77e w75b o13e o44b o44d o44f o51e
o63e w75e; do zfs create pool1/ops/${_rep}; zfs set dedup=on
pool1/ops/${_rep}; done
```

La déduplication n'est pas activée par défaut, il faut activer la déduplication sur les système de fichiers ZFS comme suit : `zfs set dedup=on pool1/ops/${_rep}`

Créer les systèmes de fichiers ZFS partimag et nfssahre sur le pool ZFS pool1 comme suit :

```
zfs create pool1/partimag
zfs create pool1/nfsshare
```

Vérifier que les systèmes de fichiers sont bien créés.

```

```

```
 zfs list
NAME                 USED  AVAIL     REFER  MOUNTPOINT
pool1                431G  4.36T      411G  /data
pool1/nfsshare        96K  4.36T       96K  /data/nfsshare
pool1/ops           1.65M  4.36T      252K  /data/ops
pool1/ops/o13b        96K  4.36T       96K  /data/ops/o13b
pool1/ops/o13e        96K  4.36T       96K  /data/ops/o13e
pool1/ops/o44a        96K  4.36T       96K  /data/ops/o44a
pool1/ops/o44b        96K  4.36T       96K  /data/ops/o44b
pool1/ops/o44c        96K  4.36T       96K  /data/ops/o44c
pool1/ops/o44d        96K  4.36T       96K  /data/ops/o44d
pool1/ops/o44e        96K  4.36T       96K  /data/ops/o44e
pool1/ops/o44f        96K  4.36T       96K  /data/ops/o44f
pool1/ops/o51b        96K  4.36T       96K  /data/ops/o51b
pool1/ops/o51e        96K  4.36T       96K  /data/ops/o51e
pool1/ops/o63b        96K  4.36T       96K  /data/ops/o63b
pool1/ops/o63e        96K  4.36T       96K  /data/ops/o63e
pool1/ops/o77e        96K  4.36T       96K  /data/ops/o77e
pool1/ops/w75b        96K  4.36T       96K  /data/ops/w75b
pool1/ops/w75e        96K  4.36T       96K  /data/ops/w75e
pool1/partimag     19.7G  4.36T     19.7G  /data/partimag
pool1/petra        1.21M  4.36T      188K  /data/petra
pool1/petra/eas       96K  4.36T       96K  /data/petra/eas
pool1/petra/li87a     96K  4.36T       96K  /data/petra/li87a
pool1/petra/sns       96K  4.36T       96K  /data/petra/sns
pool1/petra/st67a     96K  4.36T       96K  /data/petra/st67a
pool1/petra/t13a      96K  4.36T       96K  /data/petra/t13a
pool1/petra/t44a      96K  4.36T       96K  /data/petra/t44a
pool1/petra/t44c      96K  4.36T       96K  /data/petra/t44c
pool1/petra/t44v      96K  4.36T       96K  /data/petra/t44v
pool1/petra/t51a      96K  4.36T       96K  /data/petra/t51a
pool1/petra/t63a      96K  4.36T       96K  /data/petra/t63a
pool1/petra/txxa1     96K  4.36T       96K  /data/petra/txxa1
```

Monter le pool ZFS pool1:

```
zfs set mountpoint=/data pool1
```

Configuration de PXE

DRBL démarre les services suivants :

- dhcpd
- inetd(tftp)
- rpcbind
- iptables
- nfsd

Serveur dhcp

Installer le package dhcp-server

```
dnf -y install dhcp-server
```

Sauvegarder la configuration d'origine du serveur

```
cp -a /etc/dhcp/dhcpd.conf /etc/dhcp/dhcpd.old
```

récupérer la configuration sur l'ancien serveur:

```
rsync -aPHv  -e root@10.103.212.14:/etc/dhcp/dhcpd.conf
/etc/dhcp/dhcpd.conf
```

serveur tftp

Installer les paquets **tftp** et **tftp-server**

```
dnf install -y tftp-server tftp
```

L'installation crée automatiquement un service tftp et un fichier socket pour systemd sous `/usr/lib/systemd/system`. Il faut copier les fichiers de configuration systemd dans le répertoire `/etc/systemd/system`.

```
cp /usr/lib/systemd/system/tftp.service /etc/systemd/system/tftp-
server.service
cp /usr/lib/systemd/system/tftp.socket /etc/systemd/system/tftp-
server.socket
```

Modifier le fichier `/etc/systemd/system/tftp-server.service`:

```
[Unit]
Description=Tftp Server
Requires=tftp-server.socket
Documentation=man:in.tftpd

[Service]
ExecStart=/usr/sbin/in.tftpd -c -p -s /tftpboot
StandardInput=socket

[Install]
Also=tftp-server.socket
```

Signification des options ajoutées à la ligne ExecStart:

* **-c**: Autorise la création de nouveaux fichiers.
* **-p**: utilisé pour qu'aucune vérification d'autorisation supplémentaire ne soit effectuée au-delà des contrôles d'accès normaux fournis par le système.
* **-s** : Recommandé pour la sécurité et la compatibilité avec les anciennes ROM de démarrage.

Exécuter `systemctl daemon-reload` pour recharger les unités.

Vérifier que le démon tftp est bien activé:

```
systemctl status tftp-server
```

Il est possible d'utiliser la méthode avec **xinetd**:

Installer **xinetd**

```
dnf install xinetd
```

Définir un service dans `/etc/xinetd.d/tftp`:

```
# default: off
# description: The tftp server serves files using the trivial file
transfer \
#       protocol.  The tftp protocol is often used to boot
diskless \
#       workstations, download configuration files to network-
aware printers, \
#       and to start the installation process for some operating
systems.
service tftp
{
        socket_type             = dgram
        protocol                = udp
        wait                    = yes
        user                    = root
        server                  = /usr/sbin/in.tftpd
        server_args             = -c -s /tftpboot/nbi_img
        disable                 = no
        per_source              = 11
        cps                     = 100 2
        flags                   = IPv4
}
```

Démarrer xinetd

```
systemctl start xinetd
systemctl enable xinetd
```

rpcbind

rpcbind est utilisé pour **NFS** (**N**etwork **F**ile **S**ystem), qui permet de « monter » un système de fichier distant comme s'il était local, le démon rpcbind, écoute et répond sur le port 111.

Le client **NFS** utilise le service **rpcbind** sur le serveur pour découvrir le numéro de port utilisé par nfsd.

Les fichiers hosts.allow et hosts.deny de l'ancien serveur sont vides:

```
cat /etc/hosts.allow
#
# hosts.allow   This file contains access rules which are used to
#       allow or deny connections to network services that
#       either use the tcp_wrappers library or that have been
#       started through a tcp_wrappers-enabled xinetd.
#
#       See 'man 5 hosts_options' and 'man 5 hosts_access'
#       for information on rule syntax.
#       See 'man tcpd' for information on tcp_wrappers
#
```

```
cat /etc/hosts.deny
#
# hosts.deny    This file contains access rules which are used to
#       deny connections to network services that either use
#       the tcp_wrappers library or that have been
#       started through a tcp_wrappers-enabled xinetd.
#
#       The rules in this file can also be set up in
#       /etc/hosts.allow with a 'deny' option instead.
#
#       See 'man 5 hosts_options' and 'man 5 hosts_access'
#       for information on rule syntax.
#       See 'man tcpd' for information on tcp_wrappers
#
```

firewalld

Sur l'ancien serveur aucune règles **iptables** étaient définies dans l'ancien serveur, mais dans le

nouveau serveur **firewalld est activé**:

```
systemctl status firewalld.service
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled;
vendor p>
   Active: active (running) since Wed 2022-12-14 02:09:17 EST; 1 day 2h ago
     Docs: man:firewalld(1)
 Main PID: 1915 (firewalld)
    Tasks: 2 (limit: 821162)
   Memory: 30.5M
   CGroup: /system.slice/firewalld.service
           └─1915 /usr/libexec/platform-python -s /usr/sbin/firewalld --
nofork >

déc. 14 02:09:16 localhost.localdomain systemd[1]: Starting firewalld -
dynamic>
déc. 14 02:09:17 localhost.localdomain systemd[1]: Started firewalld -
dynamic >
déc. 14 02:09:17 localhost.localdomain firewalld[1915]: WARNING:
AllowZoneDrift>
```

NFS

Installer le paquet **nfs-utils**:

```
dnf install nfs-utils -y
```

Indiquer dans le fichier `/etc/exports` les dossiers exportés pour PXE:

```
cat > /etc/exports<<'EOF'
/data/ops 192.168.1.14(rw,async,no_root_squash)
/data/petra 192.168.1.14(rw,async,no_root_squash)
EOF
```

Redémarrer le serveur nfs:

```
systemctl restart nfs-server.service
```

Pour les systèmes avec le service Firewalld en cours d'exécution, autoriser les ports NFS:

```
firewall-cmd --add-service=nfs
firewall-cmd --add-service={nfs3,mountd,rpc-bind}
firewall-cmd --runtime-to-permanent
```

Reprise des données

Avant d'utiliser **rsync** avec **ssh** il faut adapter la configuration du démon sshd sur le serveur (cf ssh-session-timeout)

Ajouter les lignes suivantes à `/etc/ssh/sshd_config`:
`ClientAliveInterval 600`
`ClientAliveCountMax 0`

Montage des systèmes de fichiers

Monter les systèmes de fichiers sur nfsshare

```
mount -t nfs 192.168.1.15:/data/petra /nfsshare/petra
mount -t nfs 192.168.1.15:/data/ops /nfsshare/ops
```

Automatisation de la synchronisation:

- transfert de la cle publique de **mutual1** vers **mutual-new**: `scp ~/.ssh/id_rsa_SNS.pub root@10.103.212.15:`
- création de `authorized_keys` sur le root de **mutual-new**: `touch ~/.ssh/authorized_keys`
- intégration de la clé publique de **mutual1** dans le fichier `authorized_keys` du root de **mutual-new**: `cat ~/id_rsa_SNS.pub >> ~/.ssh/authorized_keys`

Modifier le script de synchronisation dans le /root de **mutual1**:

```
cat > ~/syncro-nfsshare.sh << 'EOF'
#!/bin/bash
set -o errexit
set -o nounset
set -o pipefail
date > root/syncro-nfsshare.log
rsync -av --no-i-r --delete /data/ops/ /nfsshare/ops/ >> /root/syncro-
nfsshare.log && \
rsync -av --no-i-r --delete /data/petra/ /nfsshare/petra/ >> /root/syncro-
nfsshare.log && \
ssh root@10.103.212.15 "zpool list" >> ~/syncro-nfsshare.log
tail -n40 /root/synchro-nfsshare.log | mailx -s "synchro nfsshare $(date)" -
S smtp=smtp://smtp.oc.dgfip:25 -S
from="esi.marseille.snsreseau@dgfip.finances.gouv.fr"
esi.marseille.snsreseau@dgfip.finances.gouv.fr
EOF
```

Activer le programmation du contrôle dans cron de **mutual1** avec `crontab -e`:

```
00 09 * 6 * /root/syncro-nfsshare.sh
```

Synchronisation du répertoire home

Les répertoires des utilisateurs du système (à l'exclusion du répertoire `/root` et `/home/petra`) doivent être récupérés.

```
rsync -aPHv  -e 'ssh --exclude /home/petra /home root@10.103.212.15:/
```

Synchronisation du répertoire tftpboot

Les images **PXE** sont dans le répertoire des images d'amorçage `/tftpboot` (3,8G), ce répertoire doit être intégralement récupéré.

```
rsync -aPHv  -e ssh /tftpboot root@10.103.212.15:/
rsync -aL  -e ssh /tftpboot root@10.103.212.15:/
```

La seconde synchronisation avec l'opérateur **-L** permet de récupérer les cibles de liens symboliques et non pas les liens.

Liens pointant vers un emplacement inexistant:

```
symlink has no referent: "/tftpboot/node_root/boot/boot"
symlink has no referent: "/tftpboot/node_root/etc/grub.conf"
symlink has no referent: "/tftpboot/node_root/etc/httpd/modules"
symlink has no referent: "/tftpboot/node_root/lib/cpp"
symlink has no referent:
"/tftpboot/node_root/lib/modules/2.6.32-358.el6.x86_64/build"
symlink has no referent:
"/tftpboot/node_root/lib/modules/2.6.32-358.el6.x86_64/source"
symlink has no referent: "/tftpboot/node_root/tmp/boot/boot"
symlink has no referent:
"/tftpboot/node_root/usr/lib64/libustr-1.0.so.1"
symlink has no referent:
"/tftpboot/nodes/10.103.212.20/etc/grub.conf"
symlink has no referent:
"/tftpboot/nodes/10.103.212.20/etc/httpd/modules"
```

```
symlink has no referent:
"/tftpboot/nodes/10.103.212.20/etc/sysconfig/network-scripts/ifup"
symlink has no referent:
"/tftpboot/nodes/10.103.212.20/etc/sysconfig/network-
scripts/ifdown"
symlink has no referent:
"/tftpboot/nodes/10.103.212.21/etc/grub.conf"
symlink has no referent:
"/tftpboot/nodes/10.103.212.21/etc/httpd/modules"
symlink has no referent:
"/tftpboot/nodes/10.103.212.21/etc/sysconfig/network-scripts/ifup"
symlink has no referent:
"/tftpboot/nodes/10.103.212.21/etc/sysconfig/network-
scripts/ifdown"
symlink has no referent:
"/tftpboot/nodes/10.103.212.22/etc/grub.conf"
symlink has no referent:
"/tftpboot/nodes/10.103.212.22/etc/httpd/modules"
symlink has no referent:
"/tftpboot/nodes/10.103.212.22/etc/sysconfig/network-scripts/ifup"
symlink has no referent:
"/tftpboot/nodes/10.103.212.22/etc/sysconfig/network-
scripts/ifdown"
symlink has no referent:
"/tftpboot/nodes/10.103.212.23/etc/grub.conf"
symlink has no referent:
"/tftpboot/nodes/10.103.212.23/etc/httpd/modules"
symlink has no referent:
"/tftpboot/nodes/10.103.212.23/etc/sysconfig/network-scripts/ifup"
symlink has no referent:
"/tftpboot/nodes/10.103.212.23/etc/sysconfig/network-
scripts/ifdown"
symlink has no referent:
"/tftpboot/nodes/10.103.212.24/etc/grub.conf"
symlink has no referent:
"/tftpboot/nodes/10.103.212.24/etc/httpd/modules"
symlink has no referent:
"/tftpboot/nodes/10.103.212.24/etc/sysconfig/network-scripts/ifup"
symlink has no referent:
"/tftpboot/nodes/10.103.212.24/etc/sysconfig/network-
scripts/ifdown"
symlink has no referent:
"/tftpboot/nodes/10.103.212.25/etc/grub.conf"
symlink has no referent:
"/tftpboot/nodes/10.103.212.25/etc/httpd/modules"
symlink has no referent:
"/tftpboot/nodes/10.103.212.25/etc/sysconfig/network-scripts/ifup"
symlink has no referent:
"/tftpboot/nodes/10.103.212.25/etc/sysconfig/network-
scripts/ifdown"
symlink has no referent:
```

```
"/tftpboot/nodes/10.103.212.26/etc/grub.conf"
symlink has no referent:
"/tftpboot/nodes/10.103.212.26/etc/httpd/modules"
symlink has no referent:
"/tftpboot/nodes/10.103.212.26/etc/sysconfig/network-scripts/ifup"
symlink has no referent:
"/tftpboot/nodes/10.103.212.26/etc/sysconfig/network-
scripts/ifdown"
symlink has no referent:
"/tftpboot/nodes/10.103.212.27/etc/grub.conf"
symlink has no referent:
"/tftpboot/nodes/10.103.212.27/etc/httpd/modules"
symlink has no referent:
"/tftpboot/nodes/10.103.212.27/etc/sysconfig/network-scripts/ifup"
symlink has no referent:
"/tftpboot/nodes/10.103.212.27/etc/sysconfig/network-
scripts/ifdown"
symlink has no referent:
"/tftpboot/nodes/10.103.212.28/etc/grub.conf"
symlink has no referent:
"/tftpboot/nodes/10.103.212.28/etc/httpd/modules"
symlink has no referent:
"/tftpboot/nodes/10.103.212.28/etc/sysconfig/network-scripts/ifup"
symlink has no referent:
"/tftpboot/nodes/10.103.212.28/etc/sysconfig/network-
scripts/ifdown"
symlink has no referent:
"/tftpboot/nodes/10.103.212.29/etc/grub.conf"
symlink has no referent:
"/tftpboot/nodes/10.103.212.29/etc/httpd/modules"
symlink has no referent:
"/tftpboot/nodes/10.103.212.29/etc/sysconfig/network-scripts/ifup"
symlink has no referent:
"/tftpboot/nodes/10.103.212.29/etc/sysconfig/network-
scripts/ifdown"
rsync error: some files/attrs were not transferred (see previous
errors) (code 23) at main.c(1039) [sender=3.0.6]
```

L'image de boot de DRBL utilise le répertoire `/usr` du serveur, mais **clonezilla** a été compilé pour utiliser les librairies de l'ancien serveur il faut donc le récupérer de l'ancien serveur :

```
rsync -aPHv  -e ssh /usr/ root@10.103.212.15:/tftpboot/node_usr/
rsync -aL  -e ssh /usr/ root@10.103.212.15:/tftpboot/node_usr/
```

Liens pointant vers un emplacement inexistant:

```
symlink has no referent: "/usr/lib/gcc/x86_64-redhat-
linux/4.4.4/32/libmudflap.a"
symlink has no referent: "/usr/lib/gcc/x86_64-redhat-
linux/4.4.4/32/libgomp.so"
symlink has no referent: "/usr/lib/gcc/x86_64-redhat-
linux/4.4.4/32/libmudflapth.a"
symlink has no referent: "/usr/lib/gcc/x86_64-redhat-
linux/4.4.7/32/libmudflap.a"
symlink has no referent: "/usr/lib/gcc/x86_64-redhat-
linux/4.4.7/32/libgomp.so"
symlink has no referent: "/usr/lib/gcc/x86_64-redhat-
linux/4.4.7/32/libmudflapth.a"
symlink has no referent: "/usr/lib/mkpxeinitrd-net/initrd-
skel/etc/resolv.conf"
symlink has no referent: "/usr/share/doc/redhat-release"
rsync error: some files/attrs were not transferred (see previous
errors) (code 23) at main.c(1039) [sender=3.0.6]
```

- **libmudflap**: contient la bibliothèque de support partagé GCC qui est nécessaire pour le support de mudflap (Mudflap a été supprimé de GCC 4.9 et versions ultérieures)
- **libgomp**: contient la bibliothèque de support partagé GCC qui est nécessaire pour le support OpenMP v3.0

Il faut modifier le fichier `/etc/fstab` dans chaque node du répertoire `/tftpboot/node_root/` afin de monter le répertoire `/tftpboot/node_usr/` à la place de `/usr`:

```
10.103.212.14:/tftpboot/node_usr             /usr        nfs
ro,soft,nfsvers=3,tcp,,defaults        0 0
```

Si **SELinux** est activé il faut vérifier les permissions:

```
ls -Z /tftpboot/
unconfined_u:object_r:default_t:s0 5510HI-CMW710-R1311P02.ipe
unconfined_u:object_r:default_t:s0 nbi_img
unconfined_u:object_r:default_t:s0 node_root
unconfined_u:object_r:default_t:s0 nodes
```

Les permissions par défaut **object_r** et **default_t** doivent être modifiées en **system_r** et **tftpd_t**:

```
chcon -R -r system_r /tftpboot/
chcon -R -t tftpd_t /tftpboot/
```

- **-u, –user=UTILISATEUR**: définit l'utilisateur UTILISATEUR dans le contexte de sécurité cible
- **-r, –role=RÔLE**: définit le rôle RÔLE dans le contexte de sécurité cible
- **-t, –type=TYPE**: définit le type TYPE dans le contexte de sécurité cible

```
ls -Z /tftpboot/
unconfined_u:system_r:tftpd_t:s0 5510HI-CMW710-R1311P02.ipe
unconfined_u:system_r:tftpd_t:s0 nbi_img
unconfined_u:system_r:tftpd_t:s0 node_root
unconfined_u:system_r:tftpd_t:s0 nodes
```

Vérifier ensuite les autorisations **tftp** dans **SELinux**:

```
getsebool -a | grep tftp
tftp_anon_write --> off
tftp_home_dir --> off
```

Si l'écriture **tftp** est désactivée comme indiqué ci-dessus, il faut l'activer avec la commande **setsebool**:

```
setsebool -P tftp_anon_write 1
setsebool -P tftp_home_dir 1
```

Les modifications ci-dessus apportées à **SELinux** sont permanentes, il n'est donc plus nécessaire de modifier les fichiers de configuration de **SELinux**.

```
getsebool -a | grep tftp
tftp_anon_write --> on
tftp_home_dir --> on
```

Synchronisation du répertoire partimag

Les images **clonezilla** sont dans le répertoire des images `/data/partimag` (20G), ce répertoire doit être intégralement récupéré.

```
rsync -aPHv  -e ssh /data/partimag/ root@10.103.212.15:/home/partimag/

sent 21153316226 bytes  received 4029 bytes  112219205.60 bytes/sec
total size is 21150720341  speedup is 1.00
```

Les permissions par défaut **object_r** et **default_t** doivent être modifiées en **system_r** et **tftpd_t**:

```
chcon -R -r system_r /home/
chcon -R -t tftpd_t /home/
```

- **-u, –user=UTILISATEUR**: définit l'utilisateur UTILISATEUR dans le contexte de sécurité cible
- **-r, –role=RÔLE**: définit le rôle RÔLE dans le contexte de sécurité cible
- **-t, –type=TYPE**: définit le type TYPE dans le contexte de sécurité cible

Synchronisation des répertoires petra et ops

Les sauvegardes distantes sont dans le répertoire `/data/petra` et `/data/ops`, ces répertoires doivent être intégralement récupérés.

```
rsync -av --no-i-r --delete /data/ops/ /nfsshare/ops/
sent 248215746694 bytes  received 132142800 bytes  9685766.25 bytes/sec
total size is 247792694310  speedup is 1.00

rsync -av --no-i-r --delete /data/petra/ /nfsshare/petra/
sent 995329445884 bytes  received 15745231 bytes  58953723.52 bytes/sec
total size is 995153645621  speedup is 1.00
```

Migration du serveur Mutual

Le jour du remplacement du serveur

Import du trousseau de clef

Le répertoire `~/.ssh` héberge une clé privée `~/.ssh/id_rsa` et une clé publique qui peut par ex. s'appeler `id_rsa.pub` (sur l'ancien serveur elle s'appelle `id_rsa_SNS.pub`).

Ce trousseau est, en principe, unique pour chaque utilisateur et différent sur chaque machine. Il est utilisé lors de l'authentification sans mot de passe.

Afin de permettre aux clients du serveur Mutual de continuer à pousser leurs sauvegardes sur le nouveau serveur avec le compte **root**, il faut implanter l'ancien couple de ce compte à la place de celui créé lors de l'installation du nouveau serveur.

Sur le nouveau serveur (10.103.212.15), sauvegarder le trousseau:

```
tar czf /home/root-ssh-mutual-new-$(date +%Y-%m-%d).tgz /root/.ssh
tar czf /home/root-etc-mutual-new-$(date +%Y-%m-%d).tgz /etc/ssh/ssh_*
```

Sur l'ancien serveur (10.103.212.14), sauvegarder le trousseau:

```
tar czf /home/root-ssh-mutual-old-$(date +%Y-%m-%d).tgz /root/.ssh
tar czf /home/root-etc-mutual-old-$(date +%Y-%m-%d).tgz /etc/ssh/ssh_*
```

Sur l'ancien serveur (10.103.212.14), implanter le trousseau de clef du nouveau serveur:

```
scp root@10.103.212.15:/home/root-ssh-mutual-new-$(date +%Y-%m-%d).tgz ~/
scp root@10.103.212.15:/home/root-etc-mutual-new-$(date +%Y-%m-%d).tgz ~/
mv ~/.ssh ~/.ssh-old
```

```
cp -a /etc/ssh/ /etc/ssh-old
tar xzf root-ssh-mutual-new-$(date +%Y-%m-%d).tgz -C /
tar xzf root-etc-mutual-new-$(date +%Y-%m-%d).tgz -C /
```

La nouvelle clé privée doit correspondre à la clé `id_rsa_SNS.pub`, dans le répertoire `~/.ssh`, utiliser la commande **diff** suivante pour voir si les clés vont ensemble:

```
cd
diff -q <( ssh-keygen -y -e -f "~/root/.ssh/id_rsa" ) <( ssh-keygen -y -e -f
"~/root/.ssh/id_rsa.pub" )
```

Sur le nouveau serveur (10.103.212.15), implanter le trousseau de clef de l'ancien serveur:

```
scp root@10.103.212.14:/home/root-ssh-mutual-old-$(date +%Y-%m-%d).tgz ~/
scp root@10.103.212.14:/home/root-etc-mutual-old-$(date +%Y-%m-%d).tgz ~/
mv ~/.ssh ~/.ssh-old
cp -a /etc/ssh /etc/ssh-old
tar xzf root-ssh-mutual-old-$(date +%Y-%m-%d).tgz -C /
tar xzf root-etc-mutual-old-$(date +%Y-%m-%d).tgz -C /
```

La nouvelle clé privée doit correspondre à la clé `id_rsa_SNS.pub`, dans le répertoire `~/.ssh`, utiliser la commande **diff** suivante pour voir si les clés vont ensemble:

```
diff -q <( ssh-keygen -y -e -f "~/root/.ssh/id_rsa" ) <( ssh-keygen -y -e -f
"~/root/.ssh/id_rsa_SNS.pub" )
```

Chaque utilisateur a une clé privée, c'est un document sensible très personnel, il faut y appliquer des droits très restrictifs : `r-x------ (500)` pour le répertoire `.ssh` et `r-------- (400)` pour le fichier `id_rsa`.

Re configuration des interfaces

Définir l'interface ifcfg-ens10f0(10.103.212.14/27):

```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=none
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=no
NAME=ens10f0
UUID=db974012-217c-44b6-a199-0f33b940a934
```

```
DEVICE=ens10f0
ONBOOT=yes
IPADDR=10.103.212.14
PREFIX=27
GATEWAY=10.103.212.1
DNS1=10.154.59.104
DNS2=10.156.32.33
IPV6_DISABLED=yes
```

Re définir le bound0 qui sera utilisé pour les montages NFS:

```
cat > /etc/sysconfig/network-scripts/ifcfg-bond0<<'EOF'
DEVICE=bond0
BOOTPROTO=static
IPADDR=192.168.1.14
NETMASK=255.255.255.0
ONBOOT=yes
USERCTL=no
BONDING_OPTS='mode=1 miimon=100 primary=ens1f1'
EOF
```

Redémarrer le réseau:

```
systemctl restart NetworkManager.service
```

Montage de /home/petra

Ajouter dans /etc/fstab le montage automatique:

```
/data/petra /home/petra          none    bind            0 0
```

Monter le répertoire `/home/petra`:

```
mount /home/petra
```

Exports des fs PXE

Recharger le fichier /etc/exports avec les dossiers nécessaires à PXE

```
cat > /etc/exports<<'EOF'
# Generated by DRBL at 15:21:52 2019/07/11

/tftpboot/node_root 10.103.212.20(ro,async,no_root_squash,no_subtree_check)
/usr 10.103.212.20(ro,async,no_root_squash,no_subtree_check)
```

```
/home 10.103.212.20(rw,sync,no_root_squash,no_subtree_check,crossmnt)
/var/spool/mail 10.103.212.20(rw,sync,root_squash,no_subtree_check)
/opt 10.103.212.20(ro,async,no_root_squash,no_subtree_check,crossmnt)

/tftpboot/node_root 10.103.212.21(ro,async,no_root_squash,no_subtree_check)
/tftpboot/node_usr 10.103.212.21(ro,async,no_root_squash,no_subtree_check)
/home 10.103.212.21(rw,sync,no_root_squash,no_subtree_check,crossmnt)
/var/spool/mail 10.103.212.21(rw,sync,root_squash,no_subtree_check)
/opt 10.103.212.21(ro,async,no_root_squash,no_subtree_check,crossmnt)

/tftpboot/node_root 10.103.212.22(ro,async,no_root_squash,no_subtree_check)
/tftpboot/node_usr 10.103.212.22(ro,async,no_root_squash,no_subtree_check)
/home 10.103.212.22(rw,sync,no_root_squash,no_subtree_check,crossmnt)
/var/spool/mail 10.103.212.22(rw,sync,root_squash,no_subtree_check)
/opt 10.103.212.22(ro,async,no_root_squash,no_subtree_check,crossmnt)

/tftpboot/node_root 10.103.212.23(ro,async,no_root_squash,no_subtree_check)
/tftpboot/node_usr 10.103.212.23(ro,async,no_root_squash,no_subtree_check)
/home 10.103.212.23(rw,sync,no_root_squash,no_subtree_check,crossmnt)
/var/spool/mail 10.103.212.23(rw,sync,root_squash,no_subtree_check)
/opt 10.103.212.23(ro,async,no_root_squash,no_subtree_check,crossmnt)

/tftpboot/node_root 10.103.212.24(ro,async,no_root_squash,no_subtree_check)
/tftpboot/node_usr 10.103.212.24(ro,async,no_root_squash,no_subtree_check)
/home 10.103.212.24(rw,sync,no_root_squash,no_subtree_check,crossmnt)
/var/spool/mail 10.103.212.24(rw,sync,root_squash,no_subtree_check)
/opt 10.103.212.24(ro,async,no_root_squash,no_subtree_check,crossmnt)

/tftpboot/node_root 10.103.212.25(ro,async,no_root_squash,no_subtree_check)
/usr 10.103.212.25(ro,async,no_root_squash,no_subtree_check)
/home 10.103.212.25(rw,sync,no_root_squash,no_subtree_check,crossmnt)
/var/spool/mail 10.103.212.25(rw,sync,root_squash,no_subtree_check)
/opt 10.103.212.25(ro,async,no_root_squash,no_subtree_check,crossmnt)

/tftpboot/node_root 10.103.212.26(ro,async,no_root_squash,no_subtree_check)
/tftpboot/node_usr 10.103.212.26(ro,async,no_root_squash,no_subtree_check)
/home 10.103.212.26(rw,sync,no_root_squash,no_subtree_check,crossmnt)
/var/spool/mail 10.103.212.26(rw,sync,root_squash,no_subtree_check)
/opt 10.103.212.26(ro,async,no_root_squash,no_subtree_check,crossmnt)

/tftpboot/node_root 10.103.212.27(ro,async,no_root_squash,no_subtree_check)
/tftpboot/node_usr 10.103.212.27(ro,async,no_root_squash,no_subtree_check)
/home 10.103.212.27(rw,sync,no_root_squash,no_subtree_check,crossmnt)
/var/spool/mail 10.103.212.27(rw,sync,root_squash,no_subtree_check)
/opt 10.103.212.27(ro,async,no_root_squash,no_subtree_check,crossmnt)

/tftpboot/node_root 10.103.212.28(ro,async,no_root_squash,no_subtree_check)
/tftpboot/node_usr 10.103.212.28(ro,async,no_root_squash,no_subtree_check)
/home 10.103.212.28(rw,sync,no_root_squash,no_subtree_check,crossmnt)
/var/spool/mail 10.103.212.28(rw,sync,root_squash,no_subtree_check)
/opt 10.103.212.28(ro,async,no_root_squash,no_subtree_check,crossmnt)
```

```
/tftpboot/node_root 10.103.212.29(ro,async,no_root_squash,no_subtree_check)
/tftpboot/node_usr 10.103.212.29(ro,async,no_root_squash,no_subtree_check)
/home 10.103.212.29(rw,sync,no_root_squash,no_subtree_check,crossmnt)
/var/spool/mail 10.103.212.29(rw,sync,root_squash,no_subtree_check)
/opt 10.103.212.29(ro,async,no_root_squash,no_subtree_check,crossmnt)

/tftpboot/nodes/10.103.212.20/
10.103.212.20(rw,sync,no_root_squash,no_subtree_check)
/tftpboot/nodes/10.103.212.21/
10.103.212.21(rw,sync,no_root_squash,no_subtree_check)
/tftpboot/nodes/10.103.212.22/
10.103.212.22(rw,sync,no_root_squash,no_subtree_check)
/tftpboot/nodes/10.103.212.23/
10.103.212.23(rw,sync,no_root_squash,no_subtree_check)
/tftpboot/nodes/10.103.212.24/
10.103.212.24(rw,sync,no_root_squash,no_subtree_check)
/tftpboot/nodes/10.103.212.25/
10.103.212.25(rw,sync,no_root_squash,no_subtree_check)
/tftpboot/nodes/10.103.212.26/
10.103.212.26(rw,sync,no_root_squash,no_subtree_check)
/tftpboot/nodes/10.103.212.27/
10.103.212.27(rw,sync,no_root_squash,no_subtree_check)
/tftpboot/nodes/10.103.212.28/
10.103.212.28(rw,sync,no_root_squash,no_subtree_check)
/tftpboot/nodes/10.103.212.29/
10.103.212.29(rw,sync,no_root_squash,no_subtree_check)
EOF
```

Redémarrer le serveur nfs:

```
systemctl restart nfs-server.service
```

Vérifier que le démon **NFS** est bien activé:

```
systemctl status nfs-server
● nfs-server.service - NFS server and services
   Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; disabled;
vendor>
  Drop-In: /run/systemd/generator/nfs-server.service.d
           └─order-with-mounts.conf
   Active: active (exited) since Wed 2023-01-11 15:05:44 CET; 1min 44s ago
  Process: 8511 ExecStopPost=/usr/sbin/exportfs -f (code=exited,
status=0/SUCCE>
  Process: 8508 ExecStopPost=/usr/sbin/exportfs -au (code=exited,
status=0/SUCC>
  Process: 8505 ExecStop=/usr/sbin/rpc.nfsd 0 (code=exited,
status=0/SUCCESS)
  Process: 8536 ExecStart=/bin/sh -c if systemctl -q is-active gssproxy;
then s>
```

```
  Process: 8524 ExecStart=/usr/sbin/rpc.nfsd (code=exited, status=0/SUCCESS)
  Process: 8522 ExecStartPre=/usr/sbin/exportfs -r (code=exited,
status=0/SUCCE>
 Main PID: 8536 (code=exited, status=0/SUCCESS)

janv. 11 15:05:44 localhost.localdomain systemd[1]: Starting NFS server and
ser>
janv. 11 15:05:44 localhost.localdomain systemd[1]: Started NFS server and
serv>
lines 1-15/15 (END)
```

Ativer le service dhcpcd

```
systemctl enable --now dhcpd
```

Vérifier que le serveur est actif:

```
systemctl status dhcpd
● dhcpd.service - DHCPv4 Server Daemon
   Loaded: loaded (/usr/lib/systemd/system/dhcpd.service; enabled; vendor
prese>
   Active: active (running) since Wed 2023-01-11 14:24:12 CET; 8min ago
     Docs: man:dhcpd(8)
           man:dhcpd.conf(5)
 Main PID: 8259 (dhcpd)
   Status: "Dispatching packets..."
    Tasks: 1 (limit: 821162)
   Memory: 8.9M
   CGroup: /system.slice/dhcpd.service
           └─8259 /usr/sbin/dhcpd -f -cf /etc/dhcp/dhcpd.conf -user dhcpd -
grou>

janv. 11 14:24:12 localhost.localdomain dhcpd[8259]:
janv. 11 14:24:12 localhost.localdomain dhcpd[8259]:
janv. 11 14:24:12 localhost.localdomain dhcpd[8259]: No subnet declaration
for >
janv. 11 14:24:12 localhost.localdomain dhcpd[8259]: ** Ignoring requests on
en>
janv. 11 14:24:12 localhost.localdomain dhcpd[8259]:    you want, please
write >
janv. 11 14:24:12 localhost.localdomain dhcpd[8259]:    in your dhcpd.conf
file>
janv. 11 14:24:12 localhost.localdomain dhcpd[8259]:    to which interface
ens2>
janv. 11 14:24:12 localhost.localdomain dhcpd[8259]:
janv. 11 14:24:12 localhost.localdomain dhcpd[8259]: Sending on
Socket/fallba>
janv. 11 14:24:12 localhost.localdomain dhcpd[8259]: Server starting
service.
```

```
lines 1-22/22 (END)
```

Pour les systèmes avec le service Firewalld en cours d'exécution, autoriser les ports DHCP (Le serveur DHCP utilise [67/UDP]):

```
firewall-cmd --add-service=dhcp
firewall-cmd --runtime-to-permanent
```

Activer le service tftp

```
sudo systemctl enable --now tftp-server
```

Vérifier que le démon est bien activé:

```
systemctl status tftp-server
● tftp-server.service - Tftp Server
   Loaded: loaded (/etc/systemd/system/tftp-server.service; indirect; vendor
pr>
   Active: active (running) since Wed 2023-01-11 14:54:45 CET; 59s ago
     Docs: man:in.tftpd
 Main PID: 8484 (in.tftpd)
    Tasks: 1 (limit: 821162)
   Memory: 376.0K
   CGroup: /system.slice/tftp-server.service
           └─8484 /usr/sbin/in.tftpd -c -p -s /tftpboot

janv. 11 14:54:45 localhost.localdomain systemd[1]: Started Tftp Server.
```

Pour les systèmes avec le service Firewalld en cours d'exécution, autoriser les ports TFTP :

```
sudo firewall-cmd --add-service=tftp
sudo firewall-cmd --runtime-to-permanent
```

Récupération des scriptes

Récupérer tous les scriptes dans le répertoire `/root` de l'ancien serveur:

```
scp root@10.103.212.15:~/syncro* ~/
```

Et adapter les scriptes `syncro-nfsshare-ops.sh` et `syncro-nfsshare-petra.sh`.

```
#!/bin/bash

# Script pour faire un snapshot en utilisant les liens physiques

set -o errexit
set -o nounset
set -o pipefail

readonly SOURCE_DIR="/data/ops/"
readonly BACKUP_DIR="/data/ops/"
LC_TIME="fr_FR.UTF-8" BACKUP_PATH="/data/backups/$(date -d yesterday
+%d)/ops/"

#rsync -av --delete ${SOURCE_DIR} ${BACKUP_DIR}>> /root/synchro-nfsshare-
ops.log
LC_TIME="fr_FR.UTF-8" TIMEFORMAT="Sauvegarde ${SOURCE_DIR} de "$(date -d
yesterday +%A)" en %R secondes ..."
{ time {
cd ${BACKUP_DIR}
for _rep in */ ;  do
  echo "ammorce de ${BACKUP_PATH}${_rep%/}/"
  LATEST_LINK="${BACKUP_PATH}${_rep%/}/"
  if [ -d "${LATEST_LINK}" ]
    then
    rm -rf "${LATEST_LINK}"
  fi
  mkdir -p "${LATEST_LINK}"
# La commande cp ci-dessous equivaut à la commande rsync avec --link-dest
#  cp -al "${BACKUP_DIR}/${_rep}" "${LATEST_LINK}"
  rsync -a  --link-dest="${BACKUP_DIR}/${_rep}" "${BACKUP_DIR}/${_rep}"
"${LATEST_LINK}"
  touch "${LATEST_LINK}/$(date -d yesterday +%A-%w-%d)"
done
"$@" 2>&3
}
} 3>&2 2>> /root/synchro-nfsshare-ops.log
tail -n40 /root/synchro-nfsshare-ops.log | mailx -s "synchro nfsshare ops
$(date)" -S smtp=smtp://smtp.oc.dgfip:25 -S
from="esi.marseille.snsreseau@dgfip.finances.gouv.fr"
esi.marseille.snsreseau@dgfip.finances.gouv.fr
```

```
#!/bin/bash

# Script pour faire un snapshot en utilisant les liens physiques

set -o errexit
set -o nounset
```

```
set -o pipefail

readonly SOURCE_DIR="/data/petra/"
readonly BACKUP_DIR="/data/petra/"
LC_TIME="fr_FR.UTF-8" BACKUP_PATH="/data/backups/$(date -d yesterday
+%d)/petra/"

#rsync -av --delete ${SOURCE_DIR} ${BACKUP_DIR}>> /root/synchro-nfsshare-
petra.log
LC_TIME="fr_FR.UTF-8" TIMEFORMAT="Sauvegarde ${SOURCE_DIR} de "$(date -d
yesterday +%A)" en %R secondes ..."
{ time {
cd ${BACKUP_DIR}
for _rep in */ ;  do
  echo "ammorce de ${BACKUP_PATH}${_rep%/}/"
  LATEST_LINK="${BACKUP_PATH}${_rep%/}/"
  if [ -d "${LATEST_LINK}" ]
    then
    rm -rf "${LATEST_LINK}"
  fi
  mkdir -p "${LATEST_LINK}"
# La commande cp ci-dessous equivaut à la commande rsync avec --link-dest
#  cp -al "${BACKUP_DIR}/${_rep}" "${LATEST_LINK}"
  rsync -a  --link-dest="${BACKUP_DIR}/${_rep}" "${BACKUP_DIR}/${_rep}"
"${LATEST_LINK}"
  touch "${LATEST_LINK}/$(date -d yesterday +%A-%w-%d)"
done
"$@" 2>&3
}
} 3>&2 2>> synchro-nfsshare-petra.log
tail -n40 /root/synchro-nfsshare-petra.log | mailx -s "synchro nfsshare
petra $(date)" -S smtp=smtp://smtp.oc.dgfip:25 -S
from="esi.marseille.snsreseau@dgfip.finances.gouv.fr"
esi.marseille.snsreseau@dgfip.finances.gouv.fr
```

Installer la crontab

```
# SYNCHRONISATION
30 5,21 * * * /home/petra/t13a/eas/synchro/synchro.pl 2>&1 >>
/home/petra/t13a/eas/synchro/log
40 5,21 * * * /home/petra/t51a/eas/synchro/synchro.pl 2>&1 >>
/home/petra/t51a/eas/synchro/log
50 5,21 * * * /home/petra/t63a/eas/synchro/synchro.pl 2>&1 >>
/home/petra/t63a/eas/synchro/log
55 5,21 * * * /home/petra/t44a/eas/synchro/synchro.pl 2>&1 >>
/home/petra/t44a/eas/synchro/log
10 1    * * * /home/petra/t44c/eas/synchro/synchro.pl 2>&1 >>
/home/petra/t44c/eas/synchro/log
########################################
```

```
# synchro t44v temporairement desactivee
20 1    * * * /home/petra/t44v/eas/synchro/synchro.pl 2>&1 >>
/home/petra/t44v/eas/synchro/log
# synchro alternative temporaire pour t44v (sur /nfsshare)
#20 1    * * * /home/petra/synct44v.pl 2>&1 >>
/nfsshare/t44v/eas/synchro/log
###################################
00 20   * * 1-5 /home/petra/li87a/eas/synchro/synchro.pl 2>&1 >>
/home/petra/li87a/eas/synchro/log
00 3    * * * /home/petra/st67a/eas/synchro/synchro.pl 2>&1 >>
/home/petra/st67a/eas/synchro/log
00 7  * * * /root/syncro-nfsshare.sh
# minute heure jour-du-mois numero-du-mois jour-de-la-semaine(6=samedi)
#0 8 25 8 6 rsync -avP /data/ops/o13e /data/ops-o13e-25aout2018/ 2>&1 >>
/data/sauveOPS.log
```

Sur l'ancien serveur toutes les entrées de la crontab sont commentées

[1)](#) [2)](#)
,
Il faut définir un système de fichiers ZFS pour chacun des sous répertoires de petra sinon
- **rsync** va répliquer sur la racine
- On ne pourra pas créer de snapshots individuels