

# **Système de stockage distribué à faible coût**

---

**Travail de diplôme 2010**

Etudiant : Johan Söderström

Département : Technologie de l'Information et de la  
Communication

Filière : Réseaux et services

Mandant : Hervé Le Pezennec (HEIG-VD)

Responsable : Stephan Robert

Date : 30 juillet 2010

# **Cahier des charges**

## **Système de stockage distribué à faible coût**

### **Projet de Bachelor**

25 mars 2010

Etudiant : Johan Söderström  
Mandant : Hervé Le Pezennec (HEIG-VD)  
Responsable : Stephan Robert

# 1 Résumé

---

Le Service Informatique de la HEIG-VD recherche une solution de stockage à faible coût. Cette solution doit s'appuyer sur des serveurs x86 et être évolutive tout en maintenant un niveau de redondance minimum.

L'objectif de ce projet est d'évaluer plusieurs solutions qui permettent un stockage intelligent des fichiers pour les utilisateurs du réseau de l'école. Deux axes seront étudiés et une solution complète utilisant du logiciel Open Source sera testée et mise en place.

Axe numéro 1: Serveur de fichier avec stockage iSCSI.

Les utilisateurs accèdent en CIFS (Windows/Samba) un NAS en cluster (Network Access Server). Celui-ci stocke ses données sur un nuage (Cloud) de serveurs iSCSI répliqués et redondés. La problématique de la sauvegarde des données doit être abordée (Snapshot, etc...)

Axe numéro 2: File Area Network (FAN)

Les utilisateurs accèdent en CIFS (Windows/Samba) à un espace de stockage virtuel (Global Namespace). Cet espace virtuel va répartir les fichiers de manière intelligente (HSM/ILM) sur des serveurs NAS distribués. La problématique de la sauvegarde des données doit aussi être abordée (Snapshot, Restore Point, etc...)

Les arguments dont il faut tenir compte pour la solution à préconiser sont, dans l'ordre:

- Le faible coût par rapport à la taille du stockage
- L'évolutivité de la capacité de stockage
- La simplicité de gestion des stratégies de répartition des fichiers
- Le niveau de redondance pour tous les éléments de la solution
- Les facilités de sauvegarde et récupération de fichiers

Glossaire:

Hierarchical Storage Management (HSM)

Information Lifecycle Management (ILM)

## 2 Introduction

---

### 2.1 Contexte

Dans le monde informatique, le stockage est une variable en constante augmentation : on crée de plus en plus de nouvelles données sans supprimer les anciennes, on les copie, on les sauvegarde, même plusieurs fois, on les archive, on les publie et on les partage...

La HEIG-VD ne faillit pas à cette règle et doit même supérieure à la moyenne. Face à cette évidence, il est alors nécessaire de comprendre les besoins qui engendrent cet accroissement, d'étudier les processus qui se mettent en œuvre pour répondre à ces besoins et de mettre en place des solutions qui tentent de maîtriser les coûts.

### 2.2 Problématique

Il y a plusieurs familles de solutions pour ces besoins de stockage qui sont toutes différentes et qui ont chacune leur intérêt, mais toutes répondent à une loi fondamentale :

$$\text{fonctionnalités} * \text{performances} * \text{fiabilité} / \text{coût} = \text{CTE}$$

Pour notre projet nous voulons clairement baisser les coûts donc cela doit se faire au détriment des fonctionnalités, des performances, de la fiabilité ou un mix des trois. L'étude devra proposer différentes solutions en fonction des critères que l'on souhaite privilégier. En accord avec le mandant, il s'agira de choisir un bon compromis et de tester la faisabilité.

### 2.3 Stockage pour les étudiants

Les étudiants de la HEIG-VD ont l'obligation d'avoir un PC portable pendant leurs études. Aujourd'hui tous les PC portables offrent suffisamment de performances et d'espaces pour un usage normal pendant la durée des études. Alors quels sont les besoins autres pour du stockage pour les étudiants ?

- la messagerie officielle de l'école avec les documents annexés
- le partage de fichiers
- des espaces collaboratifs pour du travail en groupe
- des environnements dédiés pour des logiciels spécialisés (machines virtuelles)

Ce travail de Bachelor doit répondre à une partie de ces besoins.

## 3 But et objectifs

---

### 3.1 But du diplôme

Le but du diplôme est d'étudier et de mettre en place un banc d'essai concernant une solution matériel et logiciel pour la mise à disposition d'espaces privés et partagés pour les étudiants sur le réseau de la HEIG-VD.

### 3.2 Scénario retenu

Afin de donner un aspect plus concret à ce projet, nous proposons de suivre les scénarios suivants :

- Tous les étudiants bénéficient d'un service de backup personnel pendant leurs études avec un volume de stockage de 50 GB qu'ils peuvent accéder depuis le réseau de la HEIG-VD soit en interne soit via le VPN. Si l'étudiant désire plus d'espace, il peut demander une extension payante au service informatique.
- Les enseignants peuvent configurer des espaces de partage individuel ou en groupe pour les travaux avec les étudiants.
- Les administrateurs de réseau peuvent suivre l'évolution de l'utilisation des ressources partagées et ajouter de l'espace en cas de besoin. Ils ont à disposition des outils qui leur permettent de gérer ces espaces facilement.

A la fin du projet, l'étudiant pourra faire les démonstrations de ces scénarios. Il aura dû dans ce cas implémenter les interfaces, les outils et la documentation nécessaires.

### 3.3 Démarche

La première étape consistera à définir un scénario qui décrit l'utilisation type qu'un étudiant pourrait demander à une solution de stockage sur le réseau.

La seconde étape consistera à lister les différentes techniques et solutions existantes, en indiquant à chaque fois le besoin cible, les avantages et les inconvénients. L'objectif de cette étape sera d'élaborer un tableau des solutions types et de déterminer dans quel secteur du tableau nous allons continuer.

La troisième étape consistera à choisir la solution type qui répond au mieux aux scénarios choisis.

La quatrième étape permettra d'évaluer des produits et des systèmes qui correspondent à la solution type. Un choix technologique sera fait et permettra de mettre en place un environnement de test.

La cinquième étape permettra l'adaptation du produit pour répondre aux scénarios.

La sixième et dernière étape consistera à faire une synthèse permettant d'évaluer la qualité de la solution mise en œuvre ainsi que les moyens à mettre en œuvre pour son exploitation à l'échelle de la HEIG-VD.

## Table des matières

1.	Résumé .....	- 6 -
2.	Introduction.....	- 6 -
3.	Motivation .....	- 6 -
4.	Scénario d'utilisation.....	- 7 -
5.	Utilisation type d'un étudiant .....	- 8 -
6.	Solutions existantes.....	- 8 -
6.1	NAS .....	- 8 -
6.2	SAN .....	- 9 -
6.3	Système de fichiers en cluster.....	- 9 -
7.	Serveur de fichiers avec stockage ISCSI.....	- 10 -
7.1	SCSI Initiateur .....	- 10 -
7.2	SCSI cible .....	- 10 -
7.3	Logical Unit Number (LUN).....	- 11 -
8.	File Area Network (FAN).....	- 12 -
9.	Systèmes d'exploitation testés.....	- 13 -
9.1	CentOS.....	- 13 -
9.1.1	Procédure d'installation du service de haute disponibilité .....	- 13 -
9.1.2	Installation de la cible ISCSI .....	- 15 -
9.1.3	Configuration de l'initiateur .....	- 17 -
9.2	NexentaStor .....	- 20 -
9.2.1	Parcours des fonctionnalités offertes par le système .....	- 20 -
9.2.2	Conclusion sur le système NexentaStor .....	- 25 -
9.3	Opensolaris et Solaris10.....	- 26 -
9.4	GlusterFS.....	- 26 -
9.4.1	Qui utilise GlusterFS .....	- 26 -
9.4.2	Architecture de GlusterFS .....	- 27 -
9.4.3	Avantages et inconvénients de GlusterFS .....	- 27 -
9.4.4	Test du système GlusterFS .....	- 28 -
9.4.5	Conclusion sur le Système GlusterFS.....	- 28 -
10	Openfiler.....	- 29 -
10.1	Résumé des fonctionnalités .....	- 29 -
10.2	Utilisation du service DRBD .....	- 29 -
10.3	Schéma du réseau de stockage.....	- 30 -
10.4	Installation des serveurs Openfiler HA.....	- 30 -
10.5	Tests effectués .....	- 31 -

10.5.1	Arrêt du serveur secondaire et création d'un fichier sur un partage afin de vérifier la synchronisation des disques au démarrage du serveur secondaire .....	31 -
10.5.2	Création d'un nouveau partage sur le serveur primaire sans connexion avec le serveur secondaire .....	32 -
10.5.3	Extension du volume group avec une cible ISCSI .....	36 -
10.5.4	Contrôle de réplication du nouveau volume ISCSI .....	42 -
10.5.5	Test de perte d'une cible ISCSI sur le serveur primaire : .....	42 -
10.6	Split-Brain problem .....	44 -
10.7	Problèmes rencontrés lors d'ajout de cibles ISCSI .....	45 -
10.8	Création de nom de disques ISCSI persistants .....	46 -
10.9	Snapshot du système .....	49 -
10.10	Intégration des serveurs au domaine de la Heig-vd .....	50 -
10.11	Conclusion sur le système Openfiler .....	50 -
11	Déploiement pour la Heig-vd .....	51 -
12	Etat des lieux .....	51 -
13	Conclusion .....	51 -
14	Sources et Références .....	52 -
15	Annexes .....	53 -
15.1	Installation d'Openfiler .....	53 -
15.2	Configuration des serveurs Openfiler .....	61 -
15.2.1	Modification des fichiers host .....	61 -
15.2.2	Génération des clé SSH .....	62 -
15.2.3	Edition du fichier drbd.conf .....	63 -
15.2.4	Création des dossiers pour la réplication .....	65 -
15.2.5	Mise en route du service drbd .....	65 -
15.2.6	Edition du fichier /etc/lvm/lvm.conf .....	67 -
15.3	Configuration du Heartbeat .....	68 -
16	Installation de GlusterFS .....	71 -
16.1	Préparation de l'installation .....	71 -
16.2	Installation du serveur principal .....	71 -
16.3	Ajout d'un serveur au cluster .....	72 -
16.4	Console de management du cluster .....	73 -

## 1. Résumé

---

Le but du projet de diplôme est d'étudier et de mettre en place un banc d'essai concernant une solution matériel et logiciel pour la mise à disposition d'espaces de stockage privés et partagés pour les étudiants sur le réseau de la HEIG-VD. Il faut que le système de stockage soit facilement gérable par les administrateurs et adaptable au niveau du volume de stockage mis à disposition.

## 2. Introduction

---

Ce rapport de travail de bachelor va présenter différentes solutions de stockage actuellement utilisées par de grandes entreprises. L'approche du stockage par NAS et SAN seront brièvement décrites ainsi que le stockage par cluster.

Plusieurs systèmes d'exploitation testés seront détaillés avec leurs avantages et inconvénients. Une solution de stockage basée sur une distribution gratuite de linux sera décrite pour son déploiement au sein de la Heig-vd. Une description des tests effectués sur ce système sera documentée ainsi que la résolution de problèmes divers survenus. Une marche à suivre pour l'installation de base des serveurs est fournie en annexe.

## 3. Motivation

---

Ce projet est fort motivant de part son aspect pratique et par son utilité au sein de la Heig-vd ou toute autre entreprise privée désirant avoir un espace de stockage évolutif. Actuellement, le problème du stockage des données est une préoccupation majeure car les volumes à stocker ne cessent d'augmenter. Il faut donc trouver une solution évolutive à faible coût, ce qui est le but de ce projet.



## 4. Scénario d'utilisation

Ce scénario suivant reflète les cas d'utilisation du système de stockage comme il est spécifié dans le cahier des charges :

- Tous les étudiants bénéficient d'un service de backup personnel pendant leurs études avec un volume de stockage de 50 GB qu'ils peuvent accéder depuis le réseau de la HEIG-VD soit en interne soit via le VPN. Si l'étudiant désire plus d'espace, il peut demander une extension payante au service informatique.
- Les enseignants peuvent configurer des espaces de partage individuel ou en groupe pour les travaux avec les étudiants.
- Les administrateurs de réseau peuvent suivre l'évolution de l'utilisation des ressources partagées et ajouter de l'espace en cas de besoin. Ils ont à disposition des outils qui leurs permettent de gérer ces espaces facilement.

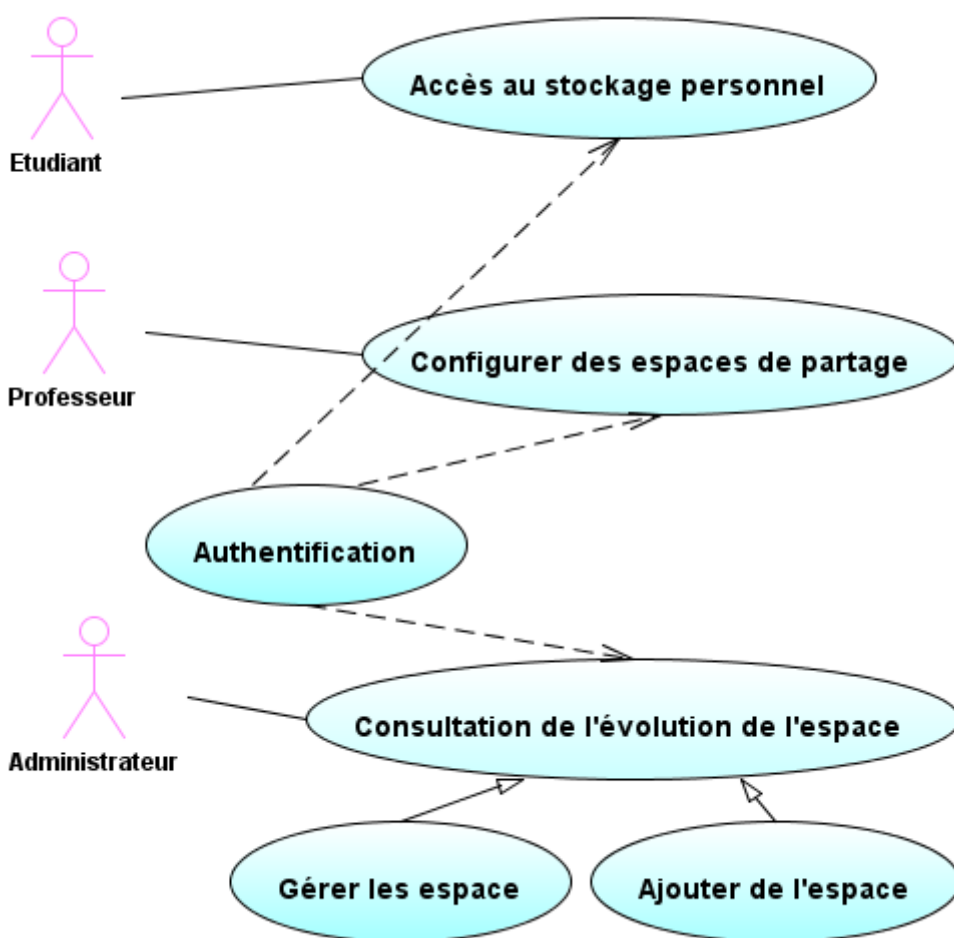


Figure 1 Diagramme des cas d'utilisation

## 5. Utilisation type d'un étudiant

---

Les étudiants au sein de la Heig-vd sont soumis à de nombreux travaux de laboratoires. Il est donc primordial de ne pas perdre ces données afin de rendre les travaux dans les délais impartis. Tous les étudiants ont leurs données stockées sur leur PC portable mais combien font des backups réguliers de leurs données? La perte d'un disque interne n'est pas très fréquente mais peut se révéler particulièrement gênante si aucune sauvegarde récente n'existe. Il faut donc offrir un moyen de pouvoir stocker ces données de manière fiable.

Le cas d'utilisation préconisé pour les étudiants sera la sauvegarde de ses répertoires en rapport avec ses travaux au sein de la Heig-vd, ce qui ne devrait jamais dépasser les 50GB.

## 6. Solutions existantes

---

Il existe de nombreuses solutions de stockage différentes. Les plus courantes pour de gros volumes de stockage sont basées sur des réseaux SAN. Les protocoles de stockage couramment utilisés sont Fibre Channel et iSCSI qui utilisent une baie de disques. Les solutions avec une baie de disques sont les plus utilisées car la maintenance et l'administration du stockage est centralisée, les I/O de réplication des données sont internes à la baie et ne surchargent pas le réseau.

### 6.1 NAS

Un moyen facile d'implémenter un stockage de fichiers est le NAS, il s'agit d'un serveur contenant une unité de stockage attachée directement au serveur. Cette méthode n'est pas optimale du point de vue de l'évolution de la capacité de stockage. Une fois le nombre maximum de disques que peut contenir l'unité de stockage atteinte, il n'est plus possible d'étendre le volume de données. Il faut pour cela racheter un autre système NAS. Cette méthode s'applique donc à un environnement où le stockage ne va pas croître rapidement.

Le système NAS est basé sur le système de transfert de fichiers CIFS ou NFS tandis que le système SAN est basé sur le transfert de bloc de données sur iSCSI ou de Fibre Channel.

Cette technologie fonctionne remarquablement bien. Les prix pour une solution NAS de 100TB sont d'environ 30'000 frs. Sur la figure suivante se trouve une illustration d'un serveur NAS pour 100TB de stockage sur un réseau ethernet dont le prix est de 31'000 \$.



Figure 2 AberNAS 890 LX-Serie

## 6.2 SAN

Les solutions basées sur un réseau de stockage séparé du LAN sont les plus évolutives au point de vue de l'augmentation de l'espace de stockage. Il n'y a presque pas de limitation du volume de stockage pouvant être géré par un SAN. Cependant, les technologies d'interconnexion au sein du SAN sont coûteuses car ils utilisent généralement de la fibre optique et les switches sont très onéreux. Le déploiement d'une solution basée sur un SAN basé sur de la fibre se chiffre en centaine de milliers de francs. Bien évidemment le système pourra gérer bien plus de 100TB très facilement et sans problème de performance. Avec l'arrivée du protocole iSCSI, la technologie de SAN est devenue abordable car elle peut utiliser des réseaux et des switches ethernet conventionnels qui ne sont pas coûteux.

## 6.3 Système de fichiers en cluster

Il s'agit d'un système de fichiers qui est monté simultanément sur plusieurs serveurs. Cette méthode permet d'interconnecter plusieurs machines physiques pour qu'elles communiquent entre elles afin de stocker des données de manière intelligente entre eux. Le principal avantage est qu'il n'y a pas besoin de matériel propriétaire à un fabricant spécifique. Le système peut être monté sur des machines bon marché et permet d'augmenter facilement la capacité de stockage avec l'ajout de nouvelles machines.

## 7. Serveur de fichiers avec stockage ISCSI

Il existe de nombreux systèmes d'exploitation permettant de monter des disques distants via le protocole ISCSI. Toutes les distributions linux disposent de package à installer.

### 7.1 SCSI Initiateur

Un initiateur permet de découvrir des disques au travers d'un réseau, cela permet au système d'exploitation de monter un disque distant et de l'utiliser comme s'il était directement connecté à la machine. Toutes les commandes SCSI de lecture/écriture sur le disque sont transportées par TCP/IP qui garantit un mode connecté afin de ne pas perdre de données.

### 7.2 SCSI cible

Une cible SCSI est un système d'exploitation qui va offrir son disque physique sur le réseau à un initiateur. C'est le serveur cible qui va accueillir physiquement les données sur le ou les disques durs mis à disposition. C'est l'initiateur qui va monter ces disques dans son système d'exploitation et les formater afin de les utiliser comme s'ils étaient en interne.

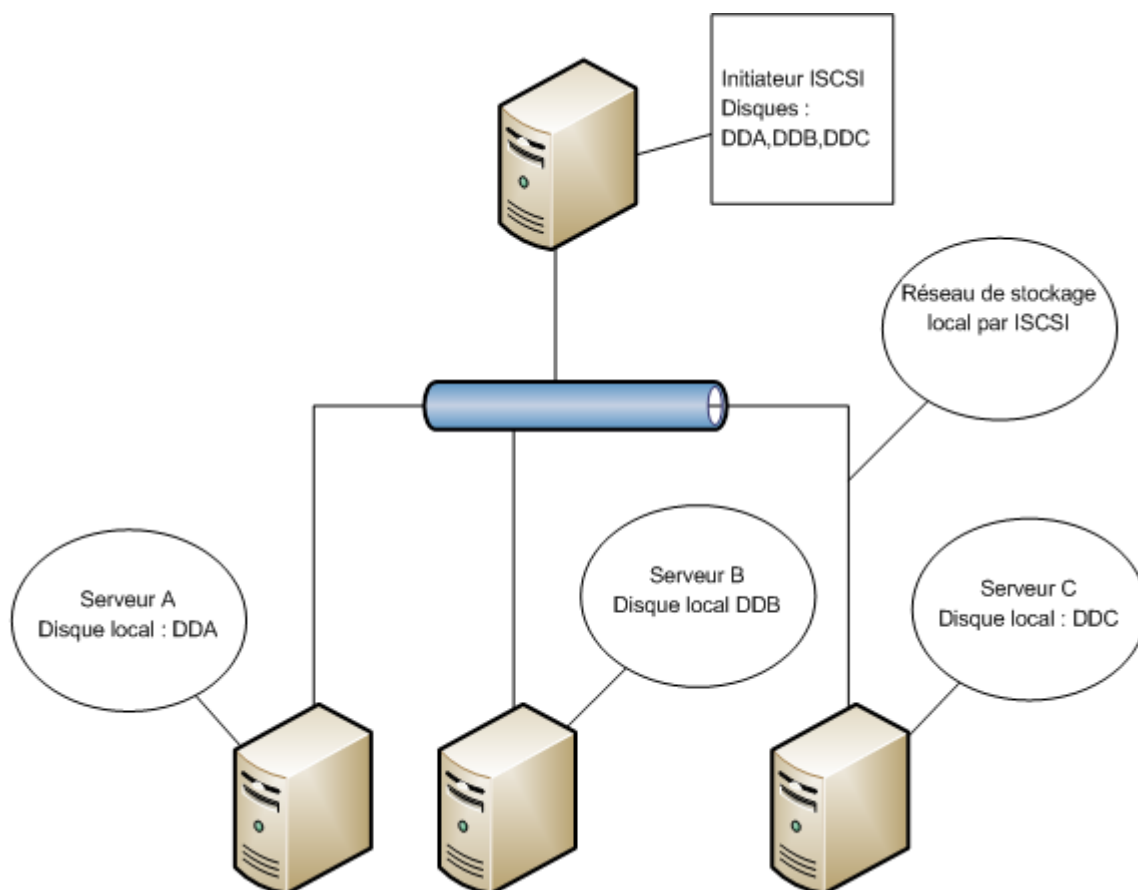


Figure 3 Schéma d'initiateur et target ISCSI

Les disques durs exportés par ISCSI sont présentés à l'initiateur comme des LUNs.

### 7.3 Logical Unit Number (LUN)

Une LUN est un terme utilisé dans le stockage des données. Cela représente un numéro d'unité logique qui permet de créer des partages ou d'être exporté pour un système distant. Une LUN peut correspondre physiquement à un ou plusieurs disques regroupés en un volume. Voici un schéma explicatif :

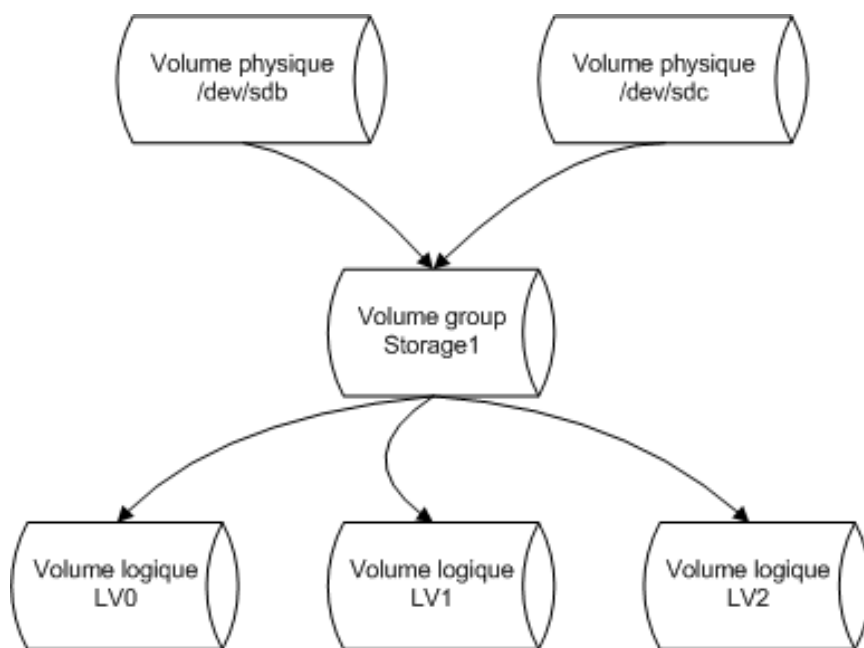


Figure 4 Explication des différents volumes

Les disques physiques sont regroupés en un groupe, dans le cas ci-dessus Storage1. Une LUN est en quelque sorte une vue logique de ce regroupement de disques physiques. Il est possible de créer plusieurs LUN par group de volume. Pour l'initiateur ISCSI, il n'est pas possible de savoir quel média de stockage est utilisé par la LUN. Cela peut-être un seul disque, plusieurs disques ou plus couramment un système de disques monté en Raid.

Terminologie :

- Un volume physique (PV) est un disque dur physique
- Un groupe de volume (VG) est un regroupement de disque dur
- Un volume logique (LV) est un disque logique créé sur un VG

## 8. File Area Network (FAN)

Un FAN est un concept qui permet de centraliser le management des données et ainsi de simplifier l'administration du stockage. Les éléments de stockage qui composent un FAN peuvent être multiples :

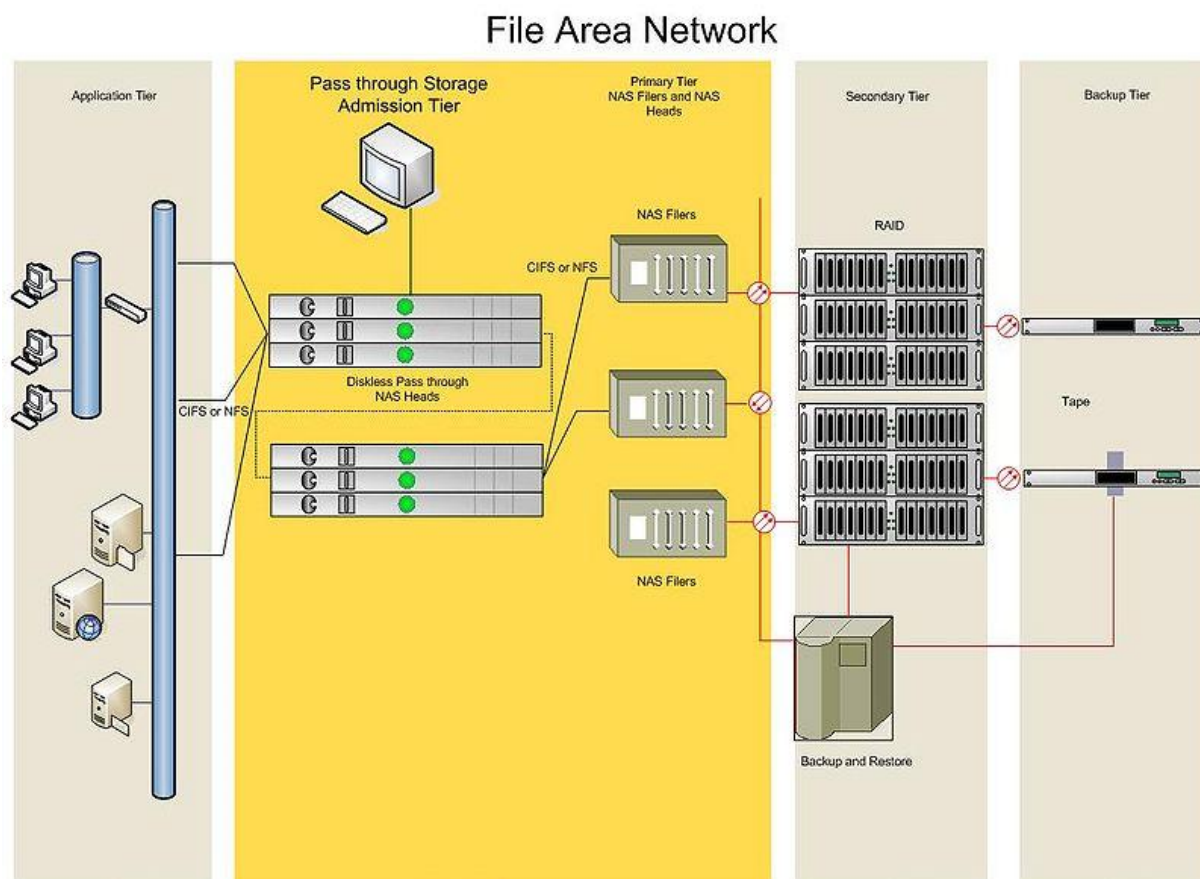


Figure 5 File Area Network

Le File Area Network peut utiliser des volumes de stockage en NAS ou en SAN. Il apporte une couche de virtualisation des données en offrant une vue unifiée de tous les éléments de stockage du réseau.

Voici les principaux fabricants de solution FAN sur le marché :

- Brocade.
- EMC
- Hewlett-Packard
- Microsoft
- Network Appliance

Tous ces fabricants proposent des solutions propriétaires et payantes. Cette solution de stockage ne seront donc pas étudiée pour un déploiement à la Heig-vd.

## 9. Systèmes d'exploitation testés

### 9.1 CentOS

Ce système d'exploitation est une distribution Linux libre et gratuite qui permet d'administrer facilement un serveur de stockage.

Le but des tests sur ce système était de concevoir un service de haute disponibilité entre deux serveurs CentOS.

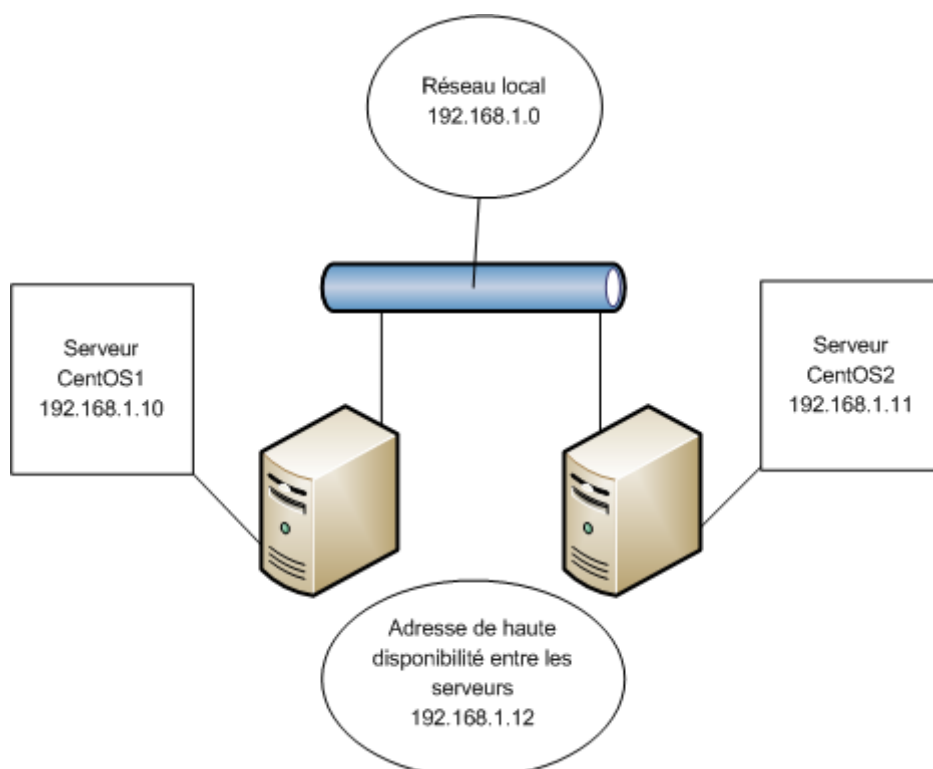


Figure 6 Haute disponibilité sur CentOS

#### 9.1.1 Procédure d'installation du service de haute disponibilité

Afin d'installer facilement le package heartbeat, il est préférable de disposer d'un accès internet sur les deux serveurs. Les serveurs ont été connectés en dhcp sur le réseau Heig-vd pour l'installation du package à l'aide de la commande suivante :

```
yum install heartbeat
```

Pour la configuration du service de haute disponibilité entre les serveurs, il est nécessaire de créer trois fichiers de configuration dans le dossier /etc/ha.d. Les fichiers sont :

- authkeys (fichier qui sert à définir la méthode d'authentification entre les serveurs)
- ha.cf (fichier qui définit les paramètres du service comme le numéro de port utilisé, la carte réseau utilisée, le temps avant la détection de la perte d'un serveur ainsi que les noms des serveurs)
- haresources (fichier qui définit les services qui utilisent le mode de haute disponibilité)

Edition du fichier authkeys :

```
auth 2
2 sha-1 unmotdepass
```

Changement des permissions sur le fichier authkeys :

```
chmod 600 /etc/ha.d/authkeys
```

Edition du fichier ha.cf :

```
logfile /var/log/ha-log
logfacility local0
keepalive 2
deadtime 30
initdead 120
bcast eth0
udpport 694
auto_failback on
node CentOS1
node CentOS2
```

Edition du fichier haresources :

```
CentOS1 192.168.1.12 httpd
```

Ce fichier haresources définit le serveur CentOS1 comme serveur primaire avec l'adresse IP de haute disponibilité 192.168.1.12 pour le service httpd.

Les deux serveurs doivent disposer des mêmes fichiers de configuration pour le bon fonctionnement du service heartbeat. On peut donc copier le contenu du dossier ha.d sur le serveur CentOS2 :

```
scp -r /etc/ha.d/ root@CentOS2:/etc/
```



Configuration de serveur Web pour tester la haute disponibilité entre les serveurs :

Edition du fichier /etc/httpd/conf/httpd.conf sur les deux serveurs :

```
Listen 192.168.1.12 :80
```

Création d'une page html sur CentOS1 :

```
echo "Serveur CentOS1" > /var/www/html/index.html
```

Création d'une page html sur CentOS2 :

```
echo "Serveur CentOS2" > /var/www/html/index.html
```

Sur chacun des serveurs, il faut activer le service heartbeat :

```
service heartbeat start
```

Une fois le service démarré, la page web est accessible sur l'adresse <http://192.168.1.12>

Le serveur répond avec le message : Serveur CentOS1

Arrêt du service heartbeat sur le serveur CentOS1 :

```
service heartbeat stop
```

Accès à la page <http://192.168.1.12>

Le serveur répond avec le message : Serveur CentOS2

Ce premier test du service de haute disponibilité n'est pas en relation directe avec le but du projet. Néanmoins, il a permis de comprendre facilement le fonctionnement du package heartbeat à l'aide des trois fichiers de configuration nécessaires.

### 9.1.2 Installation de la cible ISCSI

Le serveur CentOS2 sera la cible ISCSI. Pour configurer la cible, le package scsi-target-utils est nécessaire. Le serveur a été mis en dhcp afin d'avoir accès à internet pour l'installation, puis remis en adresse statique :

```
yum install scsi-target-utils  
chkconfig tgtd on  
service tgtd start  
ifconfig eth0 192.168.1.11 netmask 255.255.255.0
```

Pour créer une cible ISCSI sur le serveur, on utilise le package tgtadm installé comme suit :

```
tgtadm --lld iscsi --op new --mode target --tid 1 --T iqn.2010-07.centos.com :targetISCSI  
tgtadm --lld iscsi --op show --mode target
```

La figure suivante illustre la création de la cible et affiche le résultat :

```
[root@Centos2 ~]# tgtadm --lld iscsi --op new --mode target --tid 1 -T iqn.2010-07.centos.com:targetISCSI1  
[root@Centos2 ~]# tgtadm --lld iscsi --op show --mode target  
Target 1: iqn.2010-07.centos.com:targetISCSI1  
  System information:  
    Driver: iscsi  
    State: ready  
  I_T nexus information:  
  LUN information:  
    LUN: 0  
      Type: controller  
      SCSI ID: IET      00010000  
      SCSI SN: beaf10  
      Size: 0 MB  
      Online: Yes  
      Removable media: No  
      Backing store type: rdwr  
      Backing store path: None  
  Account information:  
  ACL information:
```

Figure 7 Création d'une cible ISCSI sur CentOS

Maintenant que le serveur a une cible ISCSI créée, on peut y attacher un disque à une LUN. Une partition de 4GB a été créée sur le disque sdb pour être exportée par la cible.

Dans cet exemple, on va mapper la partition sdb1 sur la LUN 1 à l'aide de la commande :

```
tgtadm --lld iscsi --op new --mode logicalunit --tid 1 --lun 1 /dev/sdb1
```

Il faut définir les autorisations sur la LUN, autrement dit, qui a la permission de se connecter et d'utiliser cette LUN à travers le réseau. Pour cette phase de test, on autorise tous les initiateurs :

```
tgtadm --lld iscsi --op bind --mode target --tid 1 -I ALL
```

Voici l'illustration de la création de la LUN 1 :

```
[root@Centos2 ~]# tgtadm --lld iscsi --op new --mode logicalunit --tid 1 --lun 1 -b /dev/sdb1
[root@Centos2 ~]# tgtadm --lld iscsi --op show --mode target
Target 1: iqn.2010-07.centos.com:targetISCSI1
System information:
  Driver: iscsi
  State: ready
I_T nexus information:
LUN information:
  LUN: 0
    Type: controller
    SCSI ID: IET      00010000
    SCSI SN: beaf10
    Size: 0 MB
    Online: Yes
    Removable media: No
    Backing store type: rdwr
    Backing store path: None
  LUN: 1
    Type: disk
    SCSI ID: IET      00010001
    SCSI SN: beaf11
    Size: 4294 MB
    Online: Yes
    Removable media: No
    Backing store type: rdwr
    Backing store path: /dev/sdb1
```

Figure 8 Création d'une LUN sur CentOS

### 9.1.3 Configuration de l'initiateur

L'initiateur est la machine CentOS1. Pour installer l'initiateur, il est à nouveau nécessaire de disposer d'une connexion à internet et d'utiliser la commande :

```
yum install iscsi-initiator-utils
```

Une fois le package initiateur ISCSI installé, il est possible de découvrir la cible au moyen de :

```
iscsiadm -m discovery -t st -p 192.168.1.11
iscsiadm -m node --login
```

```
[root@Centos1 ~]# iscsiadm -m discovery -t st -p 192.168.1.11
192.168.1.11:3260,1 iqn.2010-07.centos.com:targetISCSI1
[root@Centos1 ~]# iscsiadm -m node --login
Logging in to [iface: default, target: iqn.2010-07.centos.com:targetISCSI1, port
al: 192.168.1.11,3260]
Login to [iface: default, target: iqn.2010-07.centos.com:targetISCSI1, portal: 1
92.168.1.11,3260]: successful
```

Figure 9 Découverte et connexion à la cible ISCSI

Une fois l'initiateur connecté à la cible un nouveau disque apparaît dans le gestionnaire de volume logique du serveur.

Voici une illustration des volumes avant la connexion à la cible :

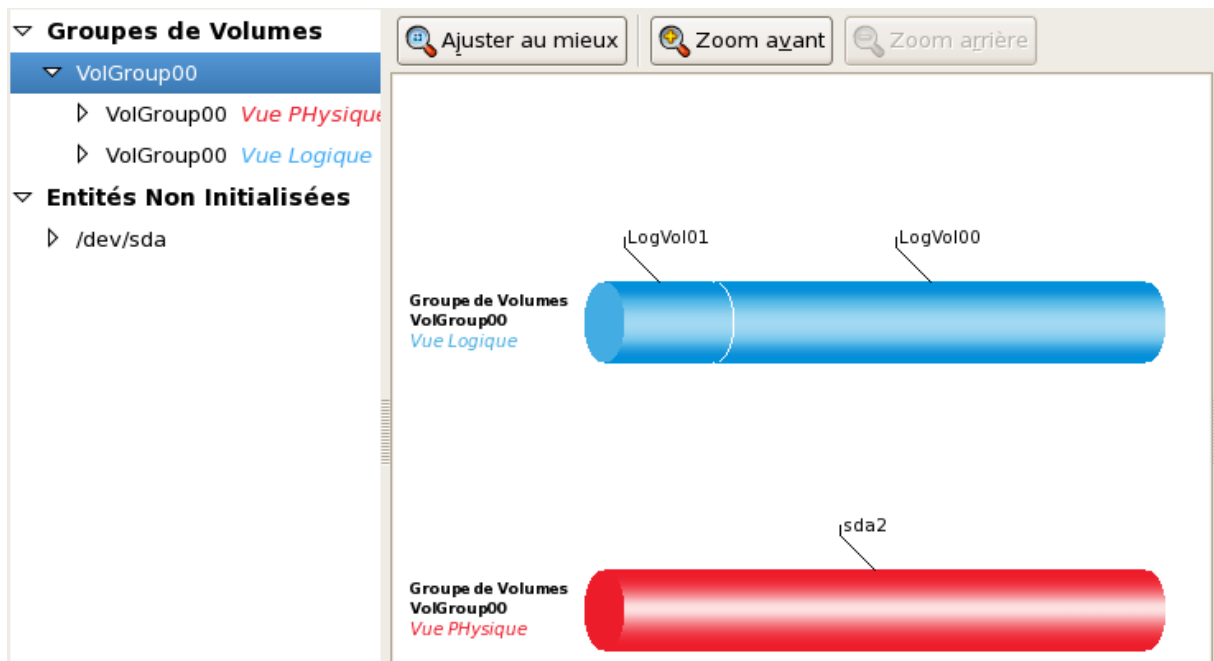


Figure 10 Volumes logiques du serveur avant la connexion iSCSI

Voici l'illustration des volumes logiques après la connexion iSCSI :



Figure 11 Volumes logiques après la connexion iSCSI

Un nouveau disque non initialisé est désormais disponible dans cette interface de management des volumes. Après l'initialisation du disque sdb, on peut l'ajouter au VG VolGroup00 existant :

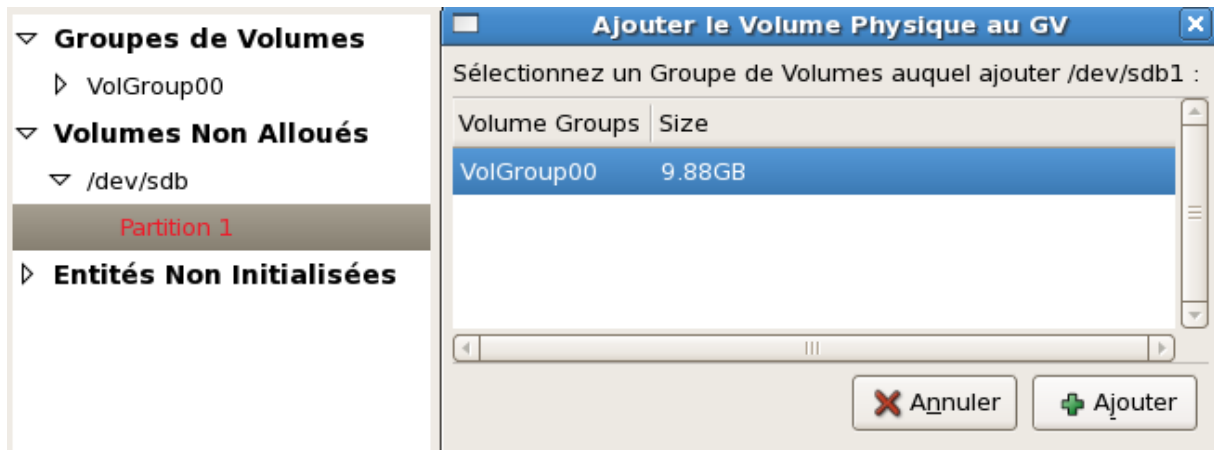


Figure 12 Ajout d'un disque physique à un VG existant

Une fois ajouté au group de volumes VolGroup00, la vue physique est :



Figure 13 Vue physique du volume group

Le groupe de volume est bien étendu avec le nouveau disque ISCSI exporté par le serveur CentOS2.

#### 9.1.4 Conclusion sur le système d'exploitation CentOS

Cette solution était prometteuse et a pris passablement de temps à la configuration des initiateurs et des cibles ISCSI. Ce système a été écarté en raison de la configuration en ligne de commande de la gestion des quotas des utilisateurs expliquée sur le lien suivant :

[http://www.centos.org/docs/5/html/Deployment\\_Guide-en-US/ch-disk-quotas.html](http://www.centos.org/docs/5/html/Deployment_Guide-en-US/ch-disk-quotas.html)

Le choix est donc de trouver une solution permettant une gestion des ressources plus facile et conviviale pour l'administrateur du système de stockage.

## 9.2 NexentaStor

Cette plateforme logicielle est basée sur le système d'exploitation SUN Solaris et permet une gestion unifiée des ressources de stockage. NexentaStor utilise le système de fichiers ZFS qui offre de nombreuses fonctionnalités intéressantes :

- Système de fichiers basé sur 128 bits
- $2^{48}$  snapshots possibles
- La taille maximale du système de fichiers est de 16 exabytes ( $2^{64}$  octets)
- Le nombre de fichiers maximal dans un dossier est de  $2^{48}$

[1] Citation de Jeff Bonwick, manager de l'équipe de développement du système de fichiers ZFS :

« Bien que nous aimerions tous que la Loi de Moore continue de s'appliquer pour toujours, la mécanique quantique impose quelques limites fondamentales sur les vitesses de calcul et les capacités de stockage de n'importe quel objet physique. En particulier, il a été montré qu'un kilogramme de matière contenue dans un volume d'un litre pouvait effectuer au maximum  $10^{51}$  opérations par secondes sur au maximum  $10^{31}$  bits d'information. Un espace de stockage 128 bits entièrement rempli contiendrait  $2^{128}$  blocs =  $2^{137}$  octets =  $2^{140}$  bits ; d'où la masse minimale nécessaire pour contenir les bits serait de  $(2^{140} \text{ bits}) / (10^{31} \text{ bits/kg}) = 136$  milliards de kg. Cependant, pour pouvoir fonctionner à cette limite de  $10^{31}$  bits/kg, la totalité de la masse de l'ordinateur devrait être composée d'énergie pure. Selon  $E=mc^2$ , l'énergie au repos de 136 milliards de kg est de  $1,2 \times 10^{28}$  joules. La masse des océans est d'environ  $1,4 \times 10^{21}$  kg. Il faut environ 4000 J pour élever la température d'un kg d'eau d'un degré Celsius, soit 400 000 J pour réchauffer de l'état gelé à l'ébullition. La chaleur latente de vaporisation ajoute encore 2 millions J/kg. Ainsi l'énergie nécessaire pour porter à ébullition les océans est d'environ  $(2,4 \times 10^6 \text{ J/kg}) \times (1,4 \times 10^{21} \text{ kg}) = 3,4 \times 10^{27} \text{ J}$ . Ainsi, remplir en totalité un espace de stockage 128 bits consommerait, littéralement, plus d'énergie que de faire bouillir les océans. »

Cette citation illustre bien à quel point il est pratiquement impossible d'atteindre les limites de stockage offertes par le système ZFS.

### 9.2.1 Parcours des fonctionnalités offertes par le système

L'interface de management est divisée en quatre catégories :

- Status, permet de voir l'état du serveur, du réseau ou du stockage
- Settings, permet de paramétrer le serveur. Intégration au domain Windows, utilisation d'un serveur LDAP, connexion de cibles ISCSI, réplication des volumes etc...
- Data Management, gère les disques connectés. Création de volumes RaidZ, gestion des permissions sur les partages et des quotas des utilisateurs.
- Analytics affiche les statistiques détaillées sur l'utilisation des ressources du serveur.

L'interface de management du serveur est simple et intuitive :

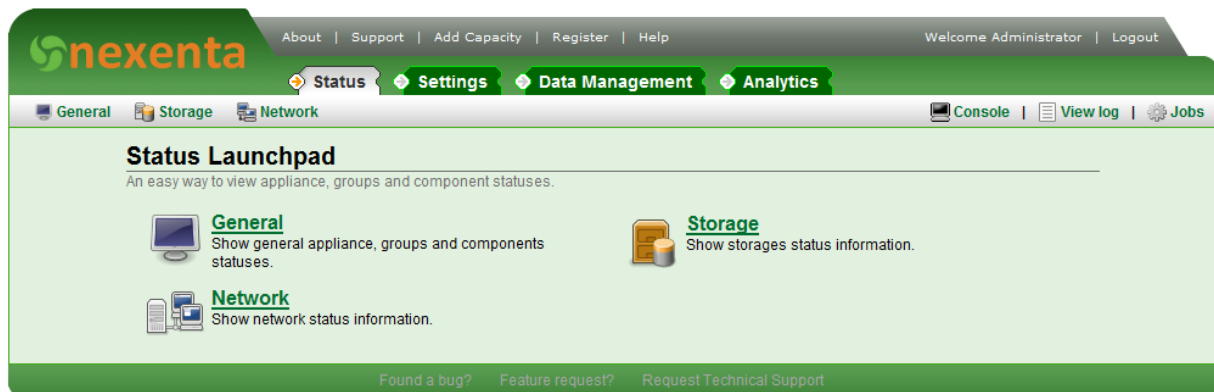


Figure 14 Interface principale de NexentaStor

L'ajout de cibles iSCSI est fait aisément à l'aide de l'interface graphique suivante :

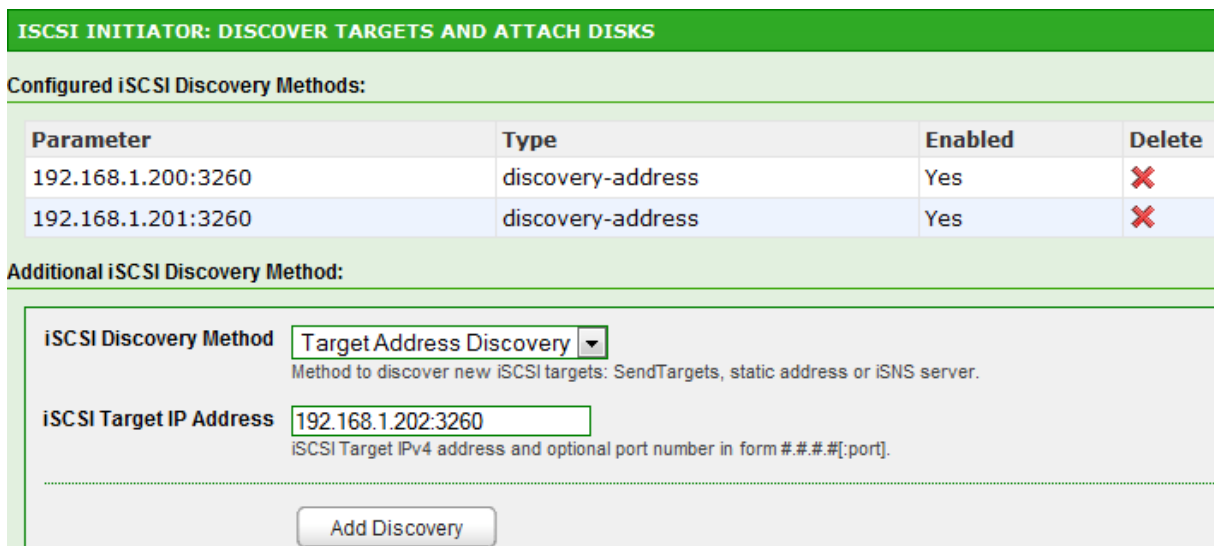


Figure 15 Ajout de cibles iSCSI

Une fois les disques connectés, il est possible de créer un volume ZFS en sélectionnant les disques connectés par iSCSI :

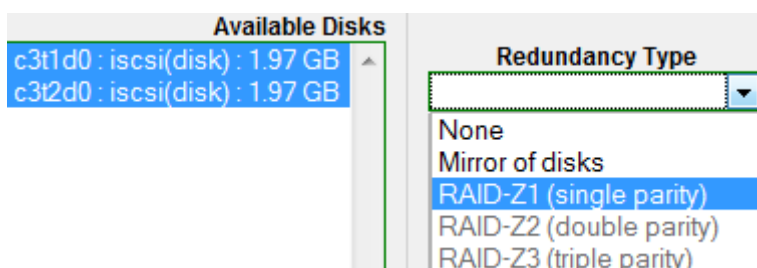


Figure 16 Création d'un volume ZFS

Un volume Data a été créé sur ces disques, puis un dossier partagé users. Il est très facile de gérer les permissions sur les dossiers en ajoutant des utilisateurs ou un groupe d'utilisateurs. La figure suivante montre le dossier users et les opérations possibles à effectuer :

**EDIT FOLDER: DATA/USERS**

Read-Only Parameters:

Name	Value
name	Data/Users
creation	Thu Jul 29 16:31 2010
used	35K
available	1.92G
referenced	35K
compressratio	1.00x
mountpoint	/volumes/Data/Users
casesensitivity	mixed

Quota:

[Edit folder quotas](#)

Access Control List: found 1 ACL entry(s)

Entity	Allow	Deny	Delete
user:smb	list_directory, read_data, add_file, write_data, add_subdirectory, append_data, read_xattr, write_xattr, execute, delete_child, write_attributes, write_acl, write_owner		✗

(+) [Add Permissions for User](#)

(+) [Add Permissions for Group](#)

Figure 17 Droits d'accès à un partage

Ajout d'un quota pour un utilisateur sur le partage users :

**CREATE USER QUOTA WITHIN FOLDER: DATA/USERS**

User

POSIX

johan

User name.

Quota

100M

Example: 1000K, 1M, 10G, 1T or none. Beware, set value to

Create

Figure 18 Ajout d'un quota de 100M sur le partage



Après l'ajout d'un quota de 100M sur le partage un fichier de 621M est transféré. Le transfert est interrompu et le système d'exploitation avertit qu'il manque d'espaces sur le disque :

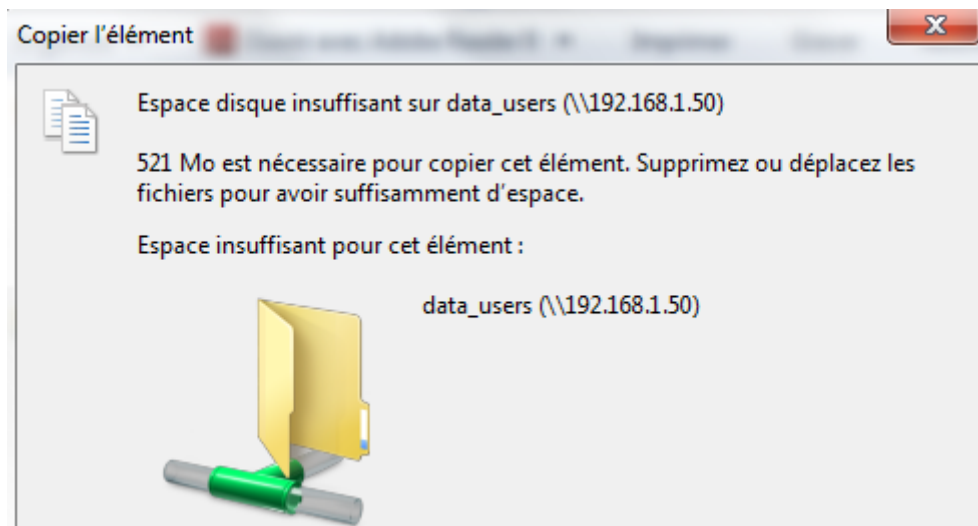


Figure 19 Erreur de transfert de fichier, quota dépassé

Lors du transfert de fichiers, il est possible de consulter les ressources utilisées par le serveur dans l'interface graphique. L'image suivante indique l'utilisation du processeur, la bande passante utilisée ainsi que l'écriture sur le disque :

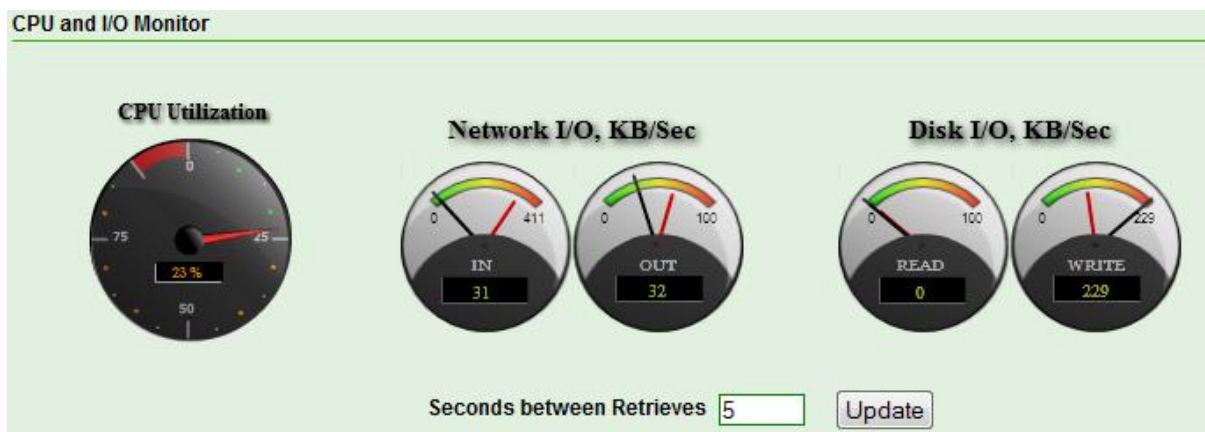


Figure 20 Ressources utilisées par le serveur

Il existe la possibilité de consulter graphiquement l'utilisation des volumes, des dossiers partagés ou encore de l'espace disque. Voici un exemple des quelques graphiques possibles :

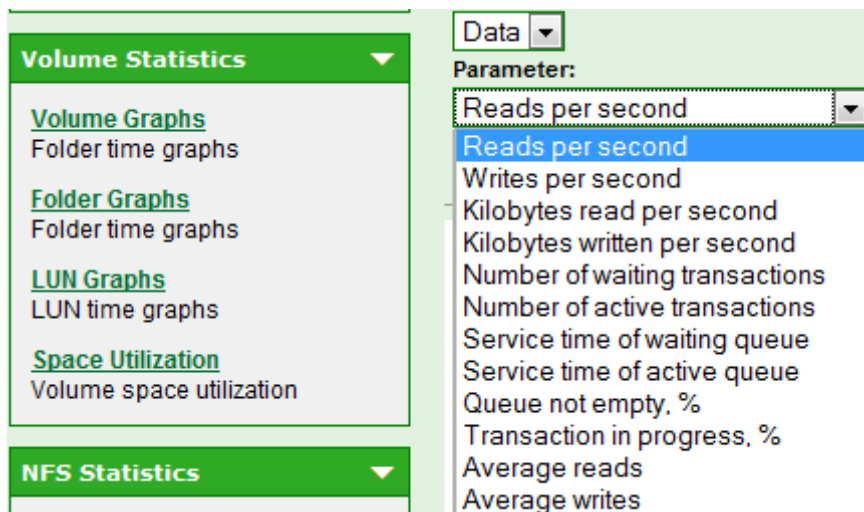


Figure 21 Exemple de graphique de statistiques

Voici encore un exemple de statistiques disponibles :

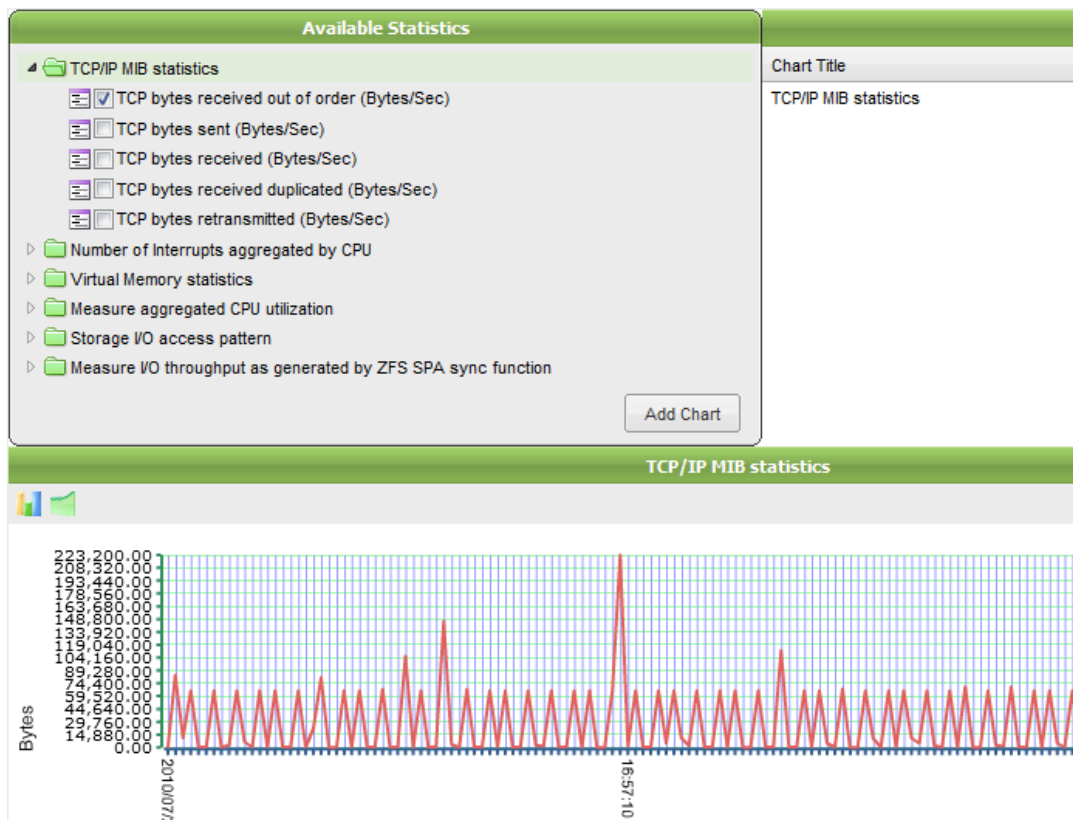


Figure 22 Statistiques plus spécifiques au serveur

Afin de garantir la fiabilité des données, il est possible d'activer une réplication automatique des dossiers ou volumes sur un autre serveur.

**CREATE AUTO-SYNC SERVICE**

**Direction** To Remote Host   
Direction of replication flow.

**Local Source Folder/Zvol:** Data/Users

**Replicate Content** ☒  
Check to replicate content of selected source or leave it unchecked

**Periodic Interval**  
Frequency: hourly  
Period: 1

**Transport Protocol** rsync+ssh  
RSYNC over SSH protocol.

**Remote Destination Host**   
Existing remote destination host to where to replicate.

**Remote Destination Folder:**   
Existing remote destination folder to where to replicate.

Figure 23 Réplication des données

Ce parcours des fonctionnalités offertes par NexentaStor n'est qu'un bref aperçu des possibilités.

### 9.2.2 Conclusion sur le système NexentaStor

Ce système de gestion de stockage est de qualité professionnelle : son interface de management est simple et intuitive. La gestion est facile et centralisée, malheureusement tout cela a un coût. Ce logiciel coûte 13990 \$ pour la gestion de 128TB de données plus 4900 \$ pour avoir un cluster de deux machines en actif/actif. Malgré les fonctionnalités offertes par ce système, cette solution est bien trop onéreuse.

## 9.3 Opensolaris et Solaris10

Opensolaris est la version gratuite du système d'exploitation de SUN. Le but de tester Opensolaris était de continuer à utiliser le système de fichiers ZFS. La version Solaris10 est devenue payante depuis le mois de mars 2010 après l'achat de SUN par Oracle. Le système d'exploitation n'est pas vendu mais il nécessite un contrat de maintenance annuel pour son exploitation.

Oracle s'est engagé à fournir des correctifs pour Opensolaris uniquement pendant six mois. Il est donc fort possible que ce système ne soit plus mis à jour et ne dispose pas de nouvelles fonctionnalités.

Les tests de ce système ont donc été très succincts et se sont arrêtés à la simple configuration d'un volume de données sur une machine virtuelle.

[2] Un manuel d'installation d'une solution de haute disponibilité basée sur Opensolaris est fourni sur le DVD du travail de diplôme. La configuration et la gestion du système sont entièrement faits en ligne de commande.

## 9.4 GlusterFS

Gluster Storage Platform est une solution open source de stockage en cluster. Ce système est une solution puissante et flexible qui simplifie la tâche de gestion des données allant de quelques téraoctets à plusieurs pétaoctets. Gluster supporte l'interconnexion par Gigabit Ethernet et ne nécessite pas de système en rack, ce qui en fait un système parfaitement adaptable au besoin de stockage. Il supporte l'agrégation de plusieurs médias de stockage différents pour ne fournir qu'un seul nom global à l'ensemble du volume de stockage.

### 9.4.1 Qui utilise GlusterFS

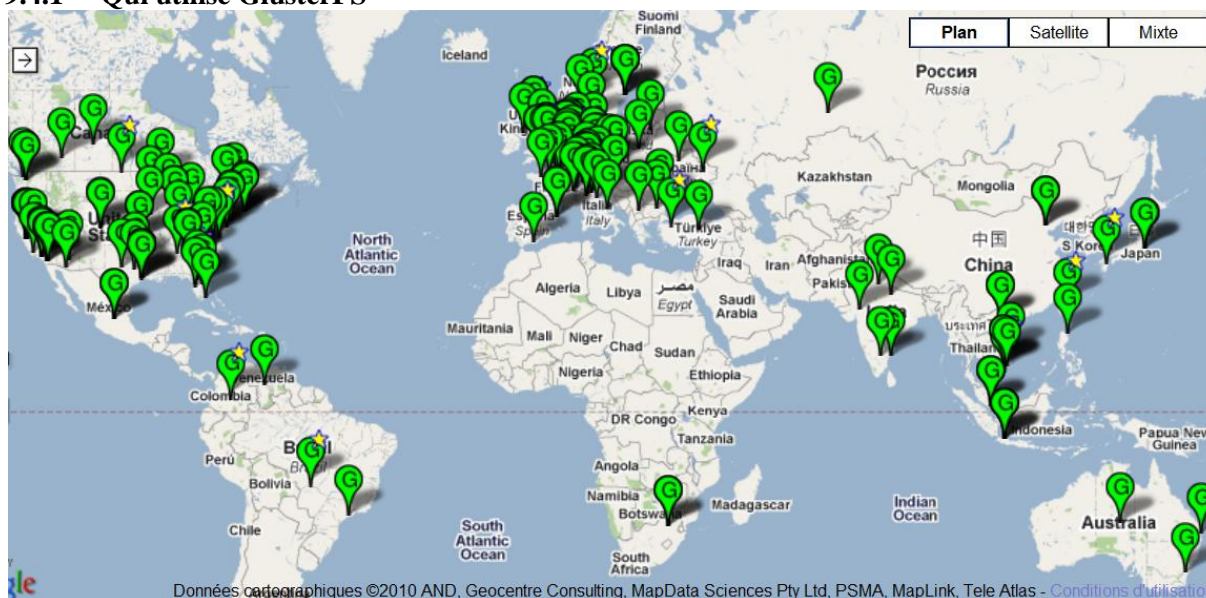


Figure 24 Carte d'utilisation de GlusterFS

Il existe plus de 170 entreprises à travers le monde qui utilisent ce système de stockage. La majeure partie des implémentations est de l'ordre de 2TB à 30TB avec un nombre restreint de serveurs allant de deux à une dizaine. Il existe cependant aussi quelques implémentations plus volumineuses dont une qui gère 100TB avec 500 serveurs ou encore 320TB avec 34 serveurs.

#### 9.4.2 Architecture de GlusterFS

La figure suivante montre comment la compatibilité entre les réseaux de stockage et les clients Windows est réalisée. Une passerelle entre le stockage GlusterFS et le réseau Lan des clients est nécessaire car le système ne gère pas nativement les droits des utilisateurs ou des quotas sur les volumes de stockage.

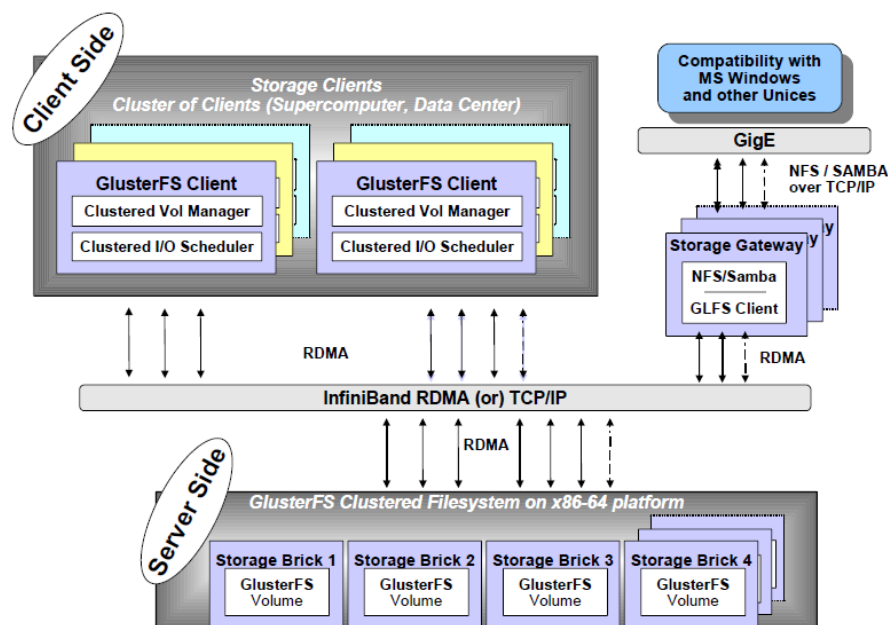


Figure 25 Passerelle pour clients MS Windows

#### 9.4.3 Avantages et inconvénients de GlusterFS

Avantages	Inconvénients
Adaptable à du hardware bon marché	Projet jeune
Solution free basée sur linux	Architecture 64 bit
Scalabilité linéaire	Passerelle pour Windows
Pas de serveur unique contenant les metadata	Gestion des droits d'accès aux fichiers
Facilité de gestion des volumes physiques	

Le système Gluster ne repose pas sur l'indexation centralisée des metadata comme la majeure partie des systèmes de stockage. Cela implique qu'il n'y a pas de Single Point of Failure ou de NameNode unique comme dans beaucoup de systèmes de stockage centralisés. Il faut toutefois implémenter une passerelle afin de pouvoir utiliser CIFS depuis un poste utilisateur Windows.

L'augmentation du volume de stockage du système est facile à adapter. Un nœud de stockage peut être rajouté à l'aide d'une interface Web. L'installation des serveurs est expliquée en annexe.

#### **9.4.4 Test du système GlusterFS**

Six ordinateurs ont été mis en place sur le banc de test. La création de volumes de données en Raid1 a été testée, mais certains bugs sont apparus :

- Impossible de supprimer un volume créé
- Après démontage d'un volume, l'interface de management ne permettait plus de remonter le volume bien que celui-ci soit toujours actif et partagé sur le réseau
- Impossible d'ajouter des serveurs à un volume déjà créé

#### **9.4.5 Conclusion sur le Système GlusterFS**

Ce système est très facile pour mettre en place un grand volume de stockage mais il nécessite un système de gestion des droits des utilisateurs et des partages à implémenter. Il n'est donc pas directement adapté aux besoins de la Heig-vd.

Il existe cependant des implémentations de GlusterFS sur certains systèmes linux comme le montre ce tutoriel : <http://blogama.org/node/96>

## 10 Openfiler

---

Ce système d'exploitation est soumis à la License GNU GPL version2, cela implique que son utilisation est gratuite. Il permet de concevoir facilement un réseau de stockage basé sur NAS ou SAN.

### 10.1 Résumé des fonctionnalités

- Interface d'administration des volumes de données
- Support de protocoles multiples, NFS, SMB/CIFS, WebDAV, FTP
- Authentification des utilisateurs à l'aide d'un annuaire LDAP ou d'un domaine Active Directory
- Fonctionne comme initiateur iSCSI ou cible iSCSI
- Agrégation de cartes réseau (NIC Bonding)
- Supporte jusqu'à 60TB de stockage
- Permet d'utiliser un système UPS
- Configuration de snapshots réguliers
- Gestion de quotas par dossier partagé ou par utilisateur

Au vu de ces fonctionnalités, cela fait d'Openfiler un candidat idéal pour la réalisation d'un réseau de stockage à faible coût.

### 10.2 Utilisation du service DRBD

DRBD est l'acronyme de Distributed Replicated Block Device. C'est un système de réplication de bloc de données entre deux serveurs mis en cluster de haute disponibilité. C'est en quelque sorte un RAID1 sur un réseau ethernet. Ce package gratuit a cependant des limitations, la capacité maximale d'un volume de réplication est de 4TB sur les systèmes 32bits et de 8TB sur 64bits. La configuration des cibles iSCSI utilisées permet d'utiliser un maximum de quatre disques. Il ne faut donc pas excéder cette taille de LUN par cible iSCSI car chaque cible va être répliquée avec drbd.

Ce service est configurable à l'aide d'un fichier /etc/drbd.conf qui définit toutes les ressources à répliquer entre les serveurs. Il nécessite la création d'une partition pour gérer les metadata du cluster, la taille de cette partition peut être approximée à l'aide de la formule :

[3] Taille de la partition [MB] < (Capacité du système [MB] /32768) +1

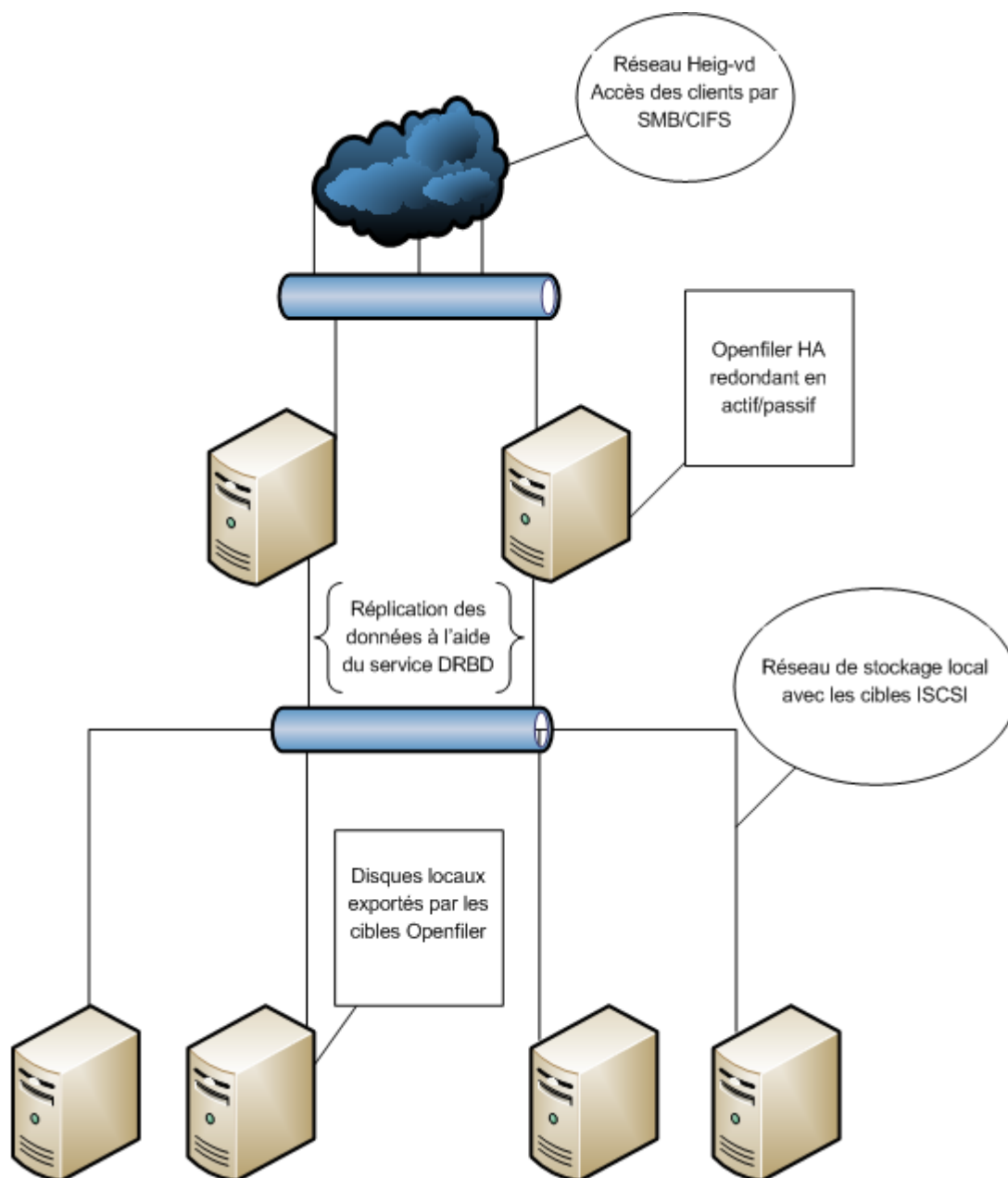
Dans notre configuration, pour 60TB de données, une partition de 1832MB est nécessaire.

Les détails de la configuration du service sont mis dans l'installation et la configuration des serveurs Openfiler est fourni en annexe.

Pour plus d'informations sur le fonctionnement ou la configuration et résolution de problèmes, veuillez vous référer au site [www.drbd.com](http://www.drbd.com)



### 10.3 Schéma du réseau de stockage



### 10.4 Installation des serveurs Openfiler HA

Veuillez vous référer au manuel d'administration fourni en annexe.



## 10.5 Tests effectués

Les tests ont été effectués sur des machines virtuelles dont voici la configuration :

Nom du serveur	Adresse IP eth0	Adresse IP eth1 (réplication)
filer01	192.168.1.10	192.168.1.20
filer02	192.168.1.11	192.168.1.21
Target1	192.168.1.200	
Target2	192.168.1.201	

L'adresse IP de haute disponibilité entre les serveurs est 192.168.1.12

### 10.5.1 Arrêt du serveur secondaire et création d'un fichier sur un partage afin de vérifier la synchronisation des disques au démarrage du serveur secondaire

Pour contrôler la réplication, le serveur primaire est arrêté pour que le serveur secondaire reprenne le contrôle. Une fois que le serveur secondaire a détecté la perte du serveur primaire, il devient lui-même serveur primaire dans un état WFConnection car il n'est plus connecté avec l'autre serveur :

```
m:res          cs          st          ds          p  mounted
      fstype
0:cluster_metadata WFConnection Primary/Unknown UpToDate/DUnknown C /cluste
r_metadata ext3
1:vg0drbd        WFConnection Primary/Unknown UpToDate/DUnknown C
[root@filer02 ~]#
```

Figure 26 Statu du service drbd sur le filer02

Les partages sont toujours accessibles sur l'adresse IP de haute disponibilité et le fichier créé sur le filer01 a bien été répliqué :

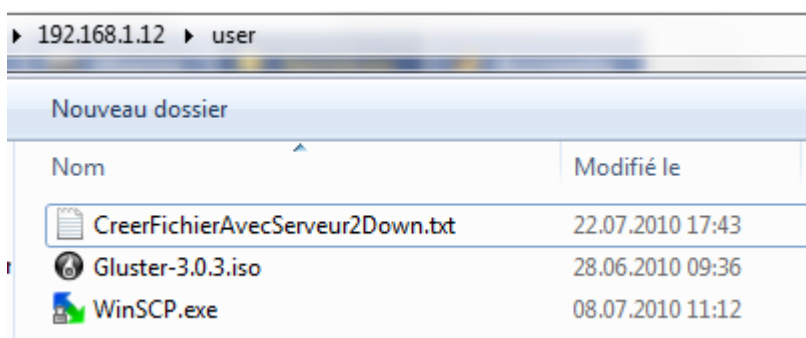


Figure 27 Création de fichier avec le serveur secondaire down

Pour contrôler la réplication lors de la remise en route du filer01, on répète l'opération de créer un fichier sur le partage. Le filer01 est redémarré :

```
m:res          cs          st          ds          p  mounted
      fstype
0:cluster_metadata Connected Secondary/Primary UpToDate/UpToDate C
1:vg0drbd        Connected Secondary/Primary UpToDate/UpToDate C
[root@filer01 ~]#
```

Figure 28 État drbd après redémarrage du filer01

Le filer01 est passé en secondaire, il faut donc arrêter le filer02 afin de contrôler la création du fichier sur le partage :

```
m:res          cs          st          ds          p  mounted
      fstype
0:cluster_metadata WFCConnection Primary/Unknown UpToDate/DUnknown C /cluste
r_metadata ext3
1:vg0drbd         WFCConnection Primary/Unknown UpToDate/DUnknown C
[root@filer01 ~]#
```

Figure 29 État drbd après arrêt du filer02

Le filer01 est bien passé en primaire et comme précédemment, il est en attente d'une connexion (WFCConnection) avec le filer02 qui est arrêté. L'accès au partage est disponible et le fichier a bien été répliqué sur le filer01 :

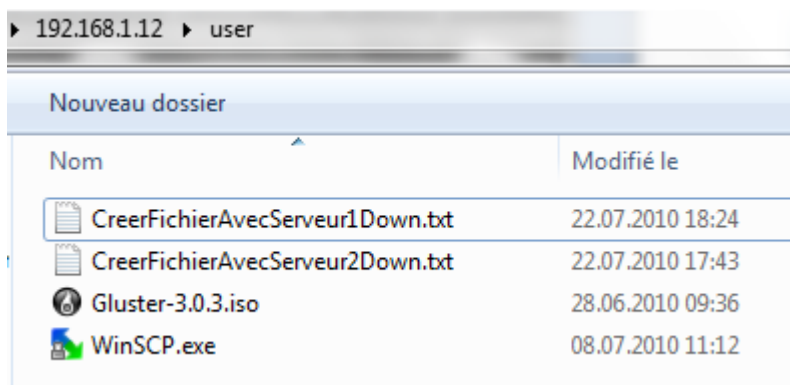


Figure 30 Création de fichier avec le filer01 down

La synchronisation des fichiers est donc correctement gérée par le service drbd entre les serveurs.

### 10.5.2 Création d'un nouveau partage sur le serveur primaire sans connexion avec le serveur secondaire

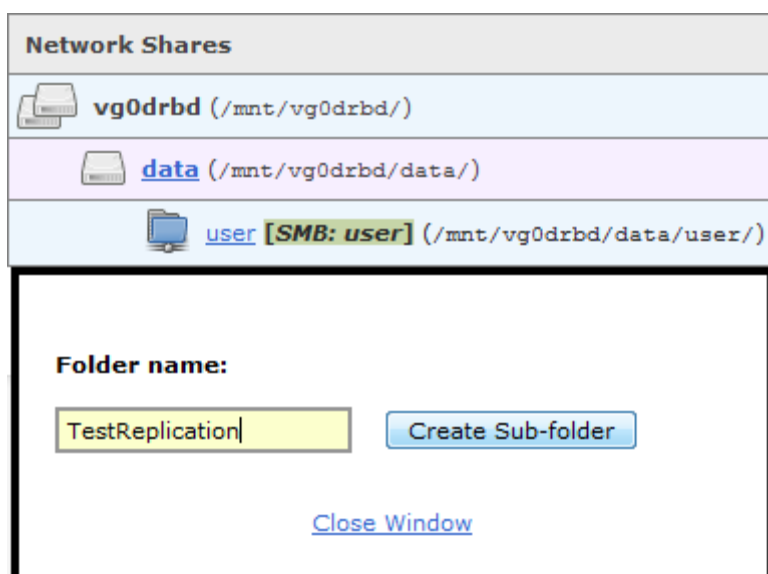


Figure 31 Création d'un dossier à partager

Une fois le dossier créé, il faut encore le définir comme un partage en cliquant sur le nom du dossier. La fenêtre suivante apparaît :

Figure 32 Création du partage TestReplication

Une fois le dossier partagé, on peut changer le nom affiché sur le réseau sinon le dossier apparaîtra sous la forme : vg0drbd.data.TestReplication

Edit share /mnt/vg0drbd/data/TestReplication/		
<p>Please use unique SMB share name overrides as duplicates automatically have a suffix attached to them. Existing shares with duplicate names can have their suffix changed every time more duplicates are created.</p>		
Share name:	<input type="text" value="TestReplication"/>	<button>Change</button>
Share description:	<input type="text" value="TestReplication"/>	<button>Change</button>
Override SMB/Rsync share name:	<input type="text" value="TestReplication"/>	<button>Change</button>

Figure 33 Changement du nom SMB

Il faut encore ajouter les permissions de lecture/écriture sur le dossier en autorisant le réseau à accéder au partage via SMB/CIFS :

**Host access configuration (/mnt/vg0drbd/data/TestReplication/)**

[\[ Back to shares list \]](#)

Name	Network	SMB/CIFS			NFS				HTTP(S) / WebDAV			FTP			Rsync		
		SMB/CIFS Options													Rsync Options		
		<input checked="" type="checkbox"/> Restart services															
		No	RO	RW	No	RO	RW	Options	No	RO	RW	No	RO	RW	No	RO	RW
local	192.168.1.0	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<a href="#">Edit</a>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>

[Update](#)

Figure 34 Ajout des permissions SMB/CIFS sur le partage

Contrôle de création du partage TestRéplication :

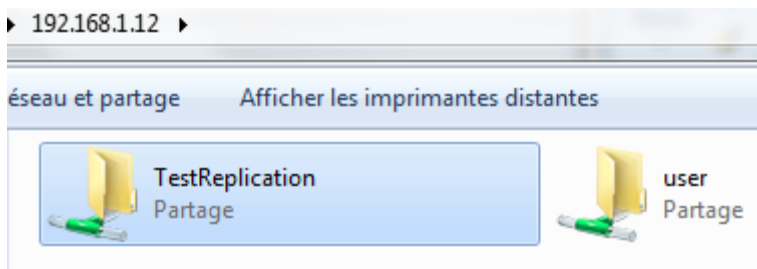


Figure 35 Accès au nouveau partage réseau

Une fois le partage créé et visible à travers le réseau, le filer02 est redémarré afin de se synchroniser avec le serveur primaire.

```
11-12 16:47:11
m:res          cs          st          ds          p  mounted
fstype
0:cluster_metadata Connected Secondary/Primary UpToDate/UpToDate C
1:vg0drbd       Connected Secondary/Primary UpToDate/UpToDate C
[root@filer02 ~]# _
```

Figure 36 État drbd du filer02

Le filer02 apparaît comme synchronisé et à jour. On peut donc arrêter le filer01 et contrôler la réplication du partage sur le filer02 :

```
m:res          cs          st          ds          p  mounted
fstype
0:cluster_metadata WFConnection Primary/Unknown UpToDate/DUnknown C /cluste
r_metadata ext3
1:vg0drbd         WFConnection Primary/Unknown UpToDate/DUnknown C
[root@filer02 ~]# _
```

Figure 37 État drbd du filer02 avec filer01 down

Le filer02 présente deux partages, user et test. Le partage test a été créé pour une première expérience de la réplication des dossiers, mais ce partage a été supprimé sur le filer01 avant le

commencement de l'expérience avec le dossier TestReplication. Il y a donc un problème comme le montre la figure suivante :

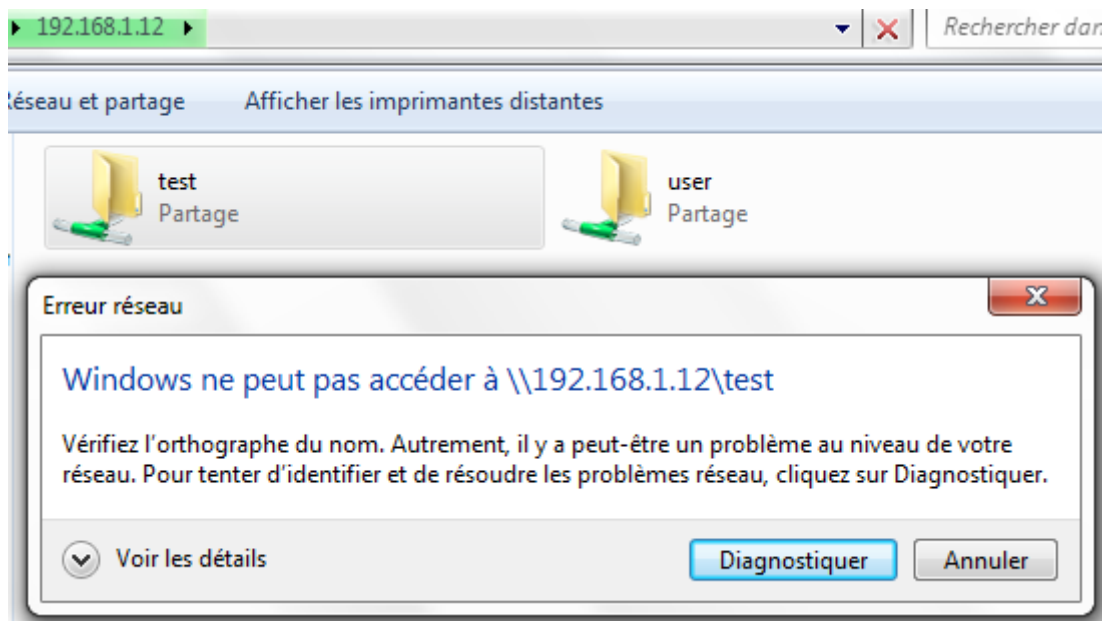


Figure 38 Contrôle de la création du partage

Pourtant, dans l'interface de management du serveur, le partage TestReplication est bien présent et il n'y a pas de trace du dossier test :

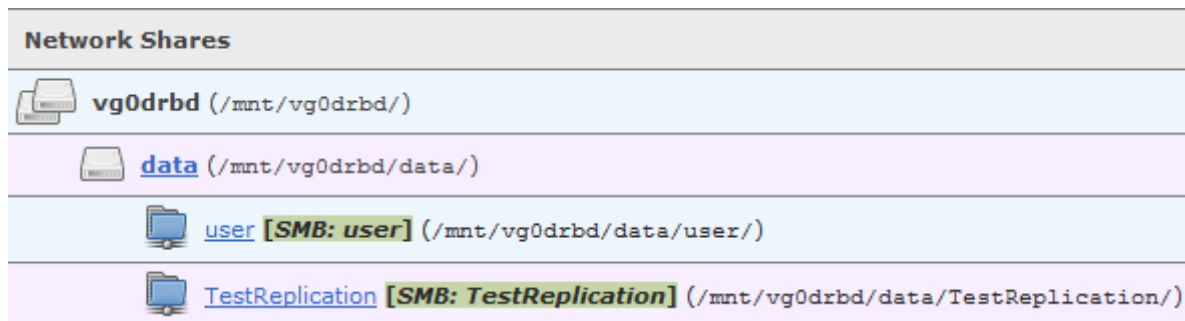


Figure 39 Partages disponibles sur le serveur

Le problème est au niveau du service SMB/CIFS du serveur, il n'a pas pris en compte les modifications sur les nouveaux partages. Il faut donc arrêter et redémarrer le service sur le serveur :

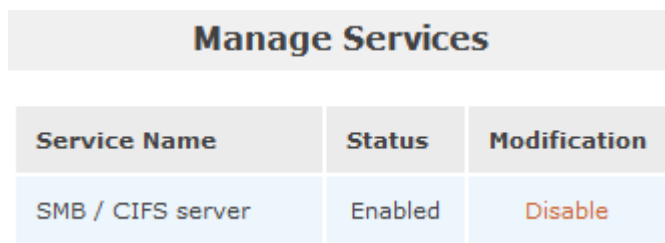


Figure 40 Redémarrage du service SMB/CIFS

Une fois le service redémarré, le partage apparaît correctement sur le réseau :

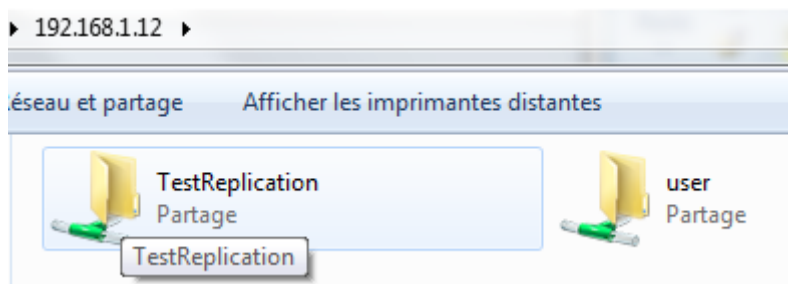


Figure 41 Partage TestReplication

Il y a donc un petit problème de synchronisation des partages SMB/CIFS sur le nœud secondaire. Le problème peut être facilement résolu, mais il nécessite une intervention de la part d'un administrateur. Cela est tout de même fâcheux car le but dans un système de haute disponibilité est de ne pas avoir à intervenir lors d'une défaillance d'un des serveurs pour que les services continuent de fonctionner normalement. Néanmoins, les anciens partages fonctionnent sans redémarrage du service SMB/CIFS.

### 10.5.3 Extension du volume group avec une cible ISCSI

Pour connecter une cible ISCSI sur le filer01, il faut utiliser le service iscsiadm à l'aide des commandes suivantes suivi d'un exemple sur le filer01 :

```
iscsiadm -m discovery -t st -p AdresseIP_de_la_Cible
iscsiadm -m node --login
```

```
[root@filer01 ~]# iscsiadm -m discovery -t st -p 192.168.1.200
192.168.1.200:3260,1 iqn.2006-01.com.openfiler:tsn.3d44cea5684d
[root@filer01 ~]# iscsiadm -m node --login
Login session [iface: default, target: iqn.2006-01.com.openfiler:tsn.3d44cea5684d, portal: 192.168.1.200,3260]
scsi 1:0:0:0: Direct-Access      OPNFILER VIRTUAL-DISK      0      PQ: 0 ANSI: 4
sd 1:0:0:0: [sdb] 4128768 512-byte hardware sectors (2114 MB)
sd 1:0:0:0: [sdb] Write Protect is off
sd 1:0:0:0: [sdb] Write cache: disabled, read cache: disabled, doesn't support D
PO or FUA
sd 1:0:0:0: [sdb] 4128768 512-byte hardware sectors (2114 MB)
sd 1:0:0:0: [sdb] Write Protect is off
sd 1:0:0:0: [sdb] Write cache: disabled, read cache: disabled, doesn't support D
PO or FUA
sd 1:0:0:0: [sdb] Attached SCSI disk
[root@filer01 ~]#
```

Figure 42 Connexion à une cible ISCSI

Afin que le disque ISCI soit reconnecté au démarrage de la machine, il faut exécuter :

```
Iscsiadm -m discovery -t st -p 192.168.1.200

iscsiadm -m node -T iqn.2010-07.com.openfiler:tsn.openfiler -p 192.168.1.200 --op update -n
node.startup -v automatic
```

```
[root@filer01 ~]# iscsiadm -m discovery -t st -p 192.168.1.200
192.168.1.200:3260,1 ign.2010-07.com.openfiler:tsn.target1
[root@filer01 ~]# iscsiadm -m node -T ign.2010-07.com.openfiler:tsn.target1 -p 1
92.168.1.200 --op update -n node.startup -v automatic
```

Figure 43 Connexion iSCSI au démarrage du serveur

Une fois le disque connecté au serveur, il faut le partitionner comme suit :

```
fdisk /dev/sdb
```

Dans le menu de partitionnement de fdisk, choisissez les options dans l'ordre :

- n (Création d'une nouvelle partition)
- p (Création d'une partition primaire sur le disque)
- enter (Sélection du premier cylindre de la partition, laissez par défaut et presser enter)
- enter (Sélection du dernier cylindre de la partition, laissez par défaut et presser enter)
- t (Changement du type de la partition créée)
- 1 (sélection de la partition à modifier)
- 8e (type de la partition linux LVM)
- w (écriture de la partition)

Illustration du formatage du disque :

```
[root@filer01 ~]# fdisk /dev/sdb

Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-1008, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-1008, default 1008):
Using default value 1008

Command (m for help): t
Selected partition 1
Hex code (type L to list codes): 8e
Changed system type of partition 1 to 8e (Linux LVM)

Command (m for help): w_
```

Figure 44 Partitionnement d'un disque iSCSI

La nouvelle partition créée est sdb1, il est nécessaire de l'initialiser avant de créer les volumes drbd:

```
dd if=/dev/zero of=/dev/sdb1
```

Cette opération peut prendre un certain temps suivant la taille de votre partition.

Configuration du fichier drbd.conf à ajouter en fin de fichier sur le filer01:

```
resource vg1drbd {
  protocol C;
  startup {
    wfc-timeout 0; ## Infinite!
    degr-wfc-timeout 120; ## 2 minutes.
  }

  disk {
    on-io-error detach;
  }

  net {
    # timeout 60;
    # connect-int 10;
    # ping-int 10;
    # max-buffers 2048;
    # max-epoch-size 2048;
  }

  syncer {
    after "cluster_metadata";
  }

  on filer01 {
    device /dev/drbd2;
    disk /dev/sdb1;
    address 192.168.1.20:7790;
    meta-disk internal;
  }

  on filer02 {
    device /dev/drbd2;
    disk /dev/sdb1;
    address 192.168.1.21:7790;
    meta-disk internal;
  }
}
```

Une fois le fichier drbd.conf édité, il est nécessaire de le copier sur le filer02 :

```
scp /etc/drbd.conf root@filer02:/etc/drbd.conf
```

Ainsi nous pouvons continuer la configuration du nouveau volume vg1drbd :

```
[root@filer01 ~]# drbdadm create-md vg1drbd
[root@filer02 ~]# drbdadm create-md vg1drbd
[root@filer01 ~]# drbdadm attach vg1drbd
[root@filer02 ~]# drbdadm attach vg1drbd
[root@filer01 ~]# drbdadm connect vg1drbd
[root@filer02 ~]# drbdadm connect vg1drbd
[root@filer01 ~]# drbdsetup /dev/drbd2 primary -o
[root@filer01 ~]# drbdadm adjust vg1drbd
```

Ces commandes ont pour effet de créer le volume de réplication vg1drbd et de débiter une synchronisation des disques comme le montre la figure suivante :



```
[root@filer01 ~]# drbdadm attach vg1drbd
[root@filer01 ~]# drbdadm connect vg1drbd
[root@filer01 ~]# drbdsetup /dev/drbd2 primary -o
[root@filer01 ~]# drbdadm adjust vg1drbd
[root@filer01 ~]# service drbd status
drbd driver loaded OK; device status:
version: 8.2.7 (api:88/proto:86-88)
GIT-hash: 61b7f4c2fc34fe3d2acf7be6bcc1fc2684708a7d build by phil@fat-tyre, 2008-
11-12 16:47:11
m:res          cs          st          ds          p  mou
nted          fstype
0:cluster_metadata Connected Primary/Secondary UpToDate/UpToDate C /c1
uster_metadata ext3
...          sync'ed: 17.3% (1709788/2062236)K
1:vg0drbd Connected Primary/Secondary UpToDate/UpToDate C
2:vg1drbd SyncSource Primary/Secondary UpToDate/Inconsistent C
```

Figure 45 Synchronisation vg1drbd

Il faut éditer le fichier /etc/lvm/lvm.conf et ajouter le filtre suivant :

```
filter = [ "r|/dev/sda4|", "r|/dev/sdb1|" ]
```

Création du volume physique /dev/drbd2, cette opération n'est à réaliser que sur le serveur primaire :

```
pvccreate /dev/drbd2
```

```
"/dev/drbd2" is a new physical volume of "1.97 GB"
--- NEW Physical volume ---
PV Name          /dev/drbd2
VG Name
PV Size          1.97 GB
Allocatable      NO
PE Size (KByte)  0
Total PE         0
Free PE          0
Allocated PE     0
PV UUID          92mork-71HU-kC02-3LEW-aTtJ-i0wY-majmgx
```

Figure 46 Nouveau volume physique drbd2

Une fois ce volume créé, il est possible d'étendre notre volume group vg0drbd. Voici le volume group vg0drbd avant son extension :

```
--- Volume group ---
VG Name          vg0drbd
System ID
Format           lvm2
Metadata Areas   1
Metadata Sequence No 3
VG Access        read/write
VG Status        resizable
MAX LV           0
Cur LV          2
Open LV          1
Max PV           0
Cur PV          1
Act PV           1
VG Size          2.63 GB
PE Size          4.00 MB
Total PE         674
Alloc PE / Size  674 / 2.63 GB
Free PE / Size   0 / 0
VG UUID          2fs0C3-9nLR-grRR-HGhI-w00S-uKY7-0Hkhw1

[root@filer01 ~]# pvdisplay
```

Figure 47 Volume group vg0drbd

Pour étendre ce volume, il faut utiliser la commande :

```
vgextend vg0drbd /dev/drbd2
```

Le volume vg0drbd a bien été étendu :

```
--- Volume group ---
VG Name          vg0drbd
System ID
Format           lvm2
Metadata Areas   2
Metadata Sequence No 4
VG Access        read/write
VG Status        resizable
MAX LV           0
Cur LV          2
Open LV          1
Max PV           0
Cur PV          2
Act PV           2
VG Size          4.60 GB
PE Size          4.00 MB
Total PE         1177
Alloc PE / Size  674 / 2.63 GB
Free PE / Size   503 / 1.96 GB
VG UUID          2fs0C3-9nLR-grRR-HGhI-w00S-uKY7-0Hkhw1
```

Figure 48 Extension du volume group

Dans l'interface de management d'openfiler, le volume a lui aussi été augmenté :

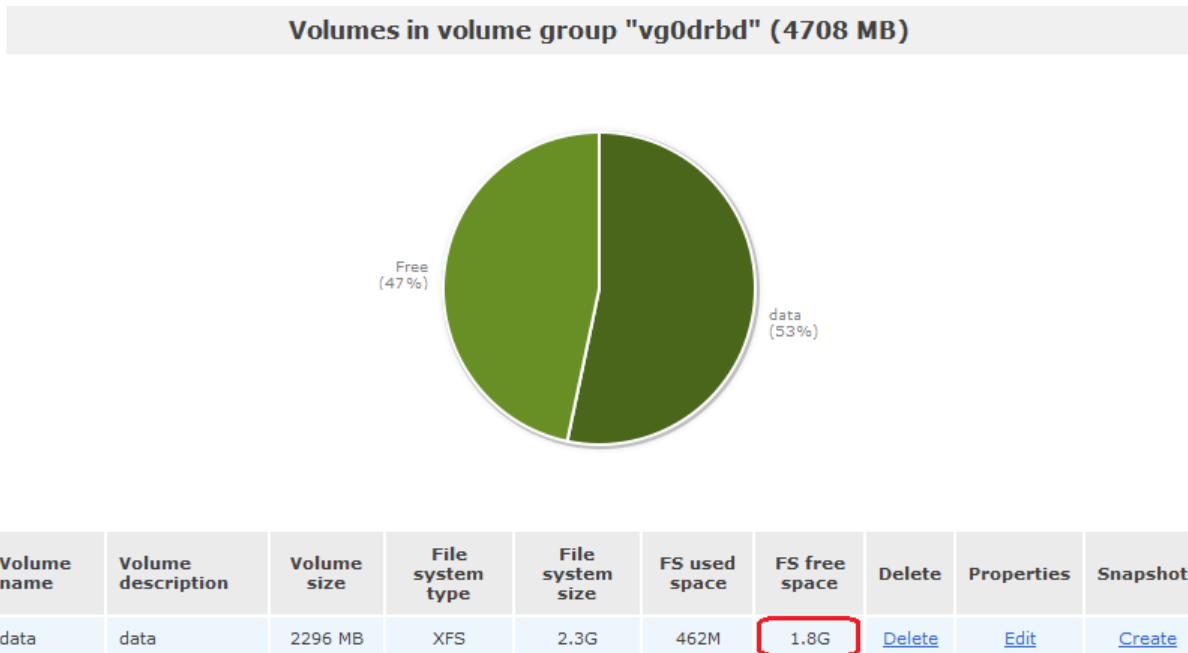


Figure 49 Vue graphique du volume vg0drbd

Le volume logique data dispose donc d'un espace non utilisé, il faut donc l'étendre à son tour afin d'utiliser tout l'espace disque mis à disposition :

```
lvextend /dev/vg0drbd/data /dev/drbd2
```

```
[root@filer01 ~]# lvextend /dev/vg0drbd/data /dev/drbd2
Extending logical volume data to 4.21 GB
Logical volume data successfully resized
[root@filer01 ~]#
```

Figure 50 Extension du volume logique

Voici le volume logique étendu :

```
[root@filer01 ~]# lvdisplay
--- Logical volume ---
LV Name                /dev/vg0drbd/data
VG Name                vg0drbd
LV UUID                8mK5m1-8bgo-87If-QVcF-jCuR-DFbA-XzKA0U
LV Write Access        read/write
LV Status              available
# open                 1
LV Size                4.21 GB
Current LE             1077
Segments               2
Allocation             inherit
Read ahead sectors     auto
- currently set to     256
Block device           253:1
```

Figure 51 Extension du volume logique

#### 10.5.4 Contrôle de réplication du nouveau volume ISCSI

Pour contrôler que la réplication du volume est effective sur le serveur secondaire, le serveur primaire est arrêté. Une fois que le serveur secondaire détecte la perte du lien, il devient serveur primaire en attente de connexion (WFConnection).

```
m:res          fstype      cs          st          ds          p  mounted
0:cluster_metadata  ext3      WFConnection  Primary/Unknown  UpToDate/DUnknown  C  /cluster
1:vg0drbd          WFConnection  Primary/Unknown  UpToDate/DUnknown  C
2:vg1drbd          WFConnection  Primary/Unknown  UpToDate/DUnknown  C

[root@filer02 ~]# lvdisplay
--- Logical volume ---
LV Name                /dev/vg0drbd/data
VG Name                vg0drbd
LV UUID                8mK5m1-8bgo-87If-QVcF-jCuR-DFbA-XzKA0U
LV Write Access        read/write
LV Status              available
# open                 1
LV Size                4.21 GB
Current LE             1077
Segments               2
Allocation             inherit
Read ahead sectors     auto
- currently set to     256
Block device           253:0

[root@filer02 ~]#
```

Figure 52 Réplication du volume augmenté

Le volume logique /dev/vg0drbd/data est bien répliqué sur le deuxième serveur et le volume a été augmenté. La réplication des informations du volume logique fonctionne donc parfaitement.

#### 10.5.5 Test de perte d'une cible ISCSI sur le serveur primaire :

Le serveur cible ISCSI du serveur primaire a été stoppé pour simuler la panne. Cela a pour effet que le serveur primaire informe le secondaire que son disque est manquant et l'état drbd est :

```
m:res          fstype      cs          st          ds          p  mounted
0:cluster_metadata  ext3      Connected    Primary/Secondary  UpToDate/UpToDate  C  /cluster
1:vg0drbd          Connected    Primary/Secondary  UpToDate/UpToDate  C
2:vg1drbd          Connected    Primary/Secondary  Diskless/UpToDate   C

[root@filer01 ~]#
```

Figure 53 État drbd avec perte d'un lien ISCSI

Le problème de cette configuration est que le serveur qui a perdu le disque reste en primaire, ce qui empêche les utilisateurs d'accéder aux données si elles se trouvaient sur ce disque. Pour rétablir le

bon fonctionnement des serveurs, il faut impérativement basculer le serveur secondaire en primaire. Pour cela, la méthode la plus rapide est d'arrêter le serveur primaire. Ainsi, les utilisateurs peuvent continuer d'utiliser les partages sans une grande interruption des services jusqu'à la restauration de la cible manquante.

La résolution du problème dépend de la panne de la cible ISCSI :

- Si la cible est juste momentanément inaccessible ou que le PC a redémarré mais sans perte de ses disques durs, le redémarrage du serveur va se reconnecter à la cible et résout ce problème.
- Si les disques durs de la machine sont perdus ou que le système est corrompu, il faut réinstaller une cible ISCSI et exporter une LUN de même taille que celle perdue. Une fois la cible reconnectée sur le serveur, il faut encore recréer une partition sur le disque comme décrit à la section 10.5.3. Une fois le disque correctement partitionné, on indique au service drbd de réajuster la synchronisation du disque avec la commande :

```
drbdadm adjust vg1drbd
```

La figure suivante montre l'état drbd avec le disque manquant, puis après l'ajustement du volume vg1drbd qui est en cours de synchronisation :

```
0:cluster_metadata Connected Secondary/Primary UpToDate/UpToDate C
1:vg0drbd Connected Secondary/Primary UpToDate/UpToDate C
2:vg1drbd Connected Secondary/Primary Diskless/UpToDate C
[root@filer01 ~]# drbdadm adjust vg1drbd
[root@filer01 ~]# service drbd status
drbd driver loaded OK; device status:
version: 8.2.7 (api:88/proto:86-88)
GIT-hash: 61b7f4c2fc34fe3d2acf7be6bcc1fc2684708a7d build by phil@fat-tyre, 2008-
11-12 16:47:11
m:res cs st ds p mou
nted fstype
0:cluster_metadata Connected Secondary/Primary UpToDate/UpToDate C
... sync'ed: 6.2% (1938524/2062236)K
1:vg0drbd Connected Secondary/Primary UpToDate/UpToDate C
2:vg1drbd SyncTarget Secondary/Primary Inconsistent/UpToDate C
```

Figure 54 Synchronisation drbd avec la nouvelle cible ISCSI

Pour que la synchronisation drbd fonctionne correctement, il faut toutefois que la nouvelle cible ISCSI soit mappée avec le même nom de disque comme celle précédemment perdue. Un moyen de s'assurer que chaque cible ISCSI soit correctement mappée par le système sera expliqué ultérieurement.

## 10.6 Split-Brain problem

Afin de tester la connexion de cibles multiples ISCSI sur un serveur, trois cibles ont été mises en place. La cible nécessaire au volume vg1drbd ainsi que deux autres sans partition initialisée. Une cible portait un nom iqn-2006-01.com.openfiler:tsn.49be4af9d845 comme le montre la figure suivante :

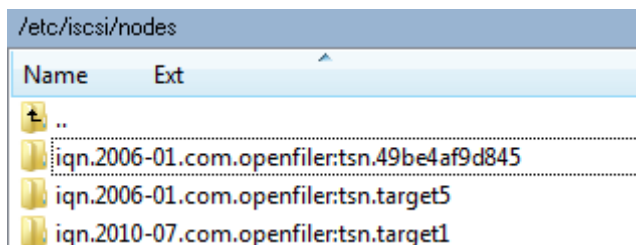


Figure 55 Cibles ISCSI du filer01

Les cibles sont classées dans le dossier /dev/iscsi/node du serveur par ordre alphabétique. Cette cible est classée avant la cible nécessaire au volume vg1drbd iqn-2010-07.com.openfiler:tsn.target1. Lors du redémarrage du serveur, cela a eu pour effet de connecter la cible iqn-2006-01 en premier et donc de prendre le nom du périphérique sdb à la place de la cible voulue. Comme le service drbd est basé sur des noms de périphériques fixes, le service a voulu répliquer ses données sur une cible non initialisée. Cela a conduit à un problème de synchronisation entre les serveurs qui mène au Split-Brain.

```
[root@filer01 ~]# drbdadm adjust vg1drbd
drbd2: No usable activity log found.
drbd2: Split-Brain detected, dropping connection!
```

Figure 56 Problème de Split-Brain

Ce problème survient généralement lorsque les deux serveurs perdent le lien de réplication entre eux et chacun devient serveur primaire en pensant que l'autre serveur est déconnecté. Cela implique que chaque serveur conserve l'adresse de haute disponibilité et continuent de fonctionner en mode StandAlone. Une fois le lien de réplication remis entre les serveurs, ils se demandent qui doit être le serveur primaire et le secondaire. Comme les deux serveurs pensent avoir les données à jour, il y a ce problème de cerveau partagé littéralement traduit. Le résultat est que le volume concerné reste en mode StandAlone. Pour résoudre cette situation, il faut décider quel serveur a les données à jour et forcer une synchronisation manuelle à l'aide de la commande :

```
Drbdadm -- --discard-my-data connect vg1drbd
```

Comme le montre la figure suivante, il n'est pas possible de demander au serveur primaire de forcer la synchronisation :

```
0:cluster_metadata Connected Primary/Secondary UpToDate/UpToDate C /cluster_metadata_ext3
1:vg0drbd Connected Primary/Secondary UpToDate/UpToDate C
2:vg1drbd StandAlone Primary/Unknown UpToDate/DUnknown -
[root@filer01 ~]# drbdadm -- --discard-my-data connect vg1drbd
/dev/drbd2: Failure: (123) --discard-my-data not allowed when primary.
Command 'drbdsetup /dev/drbd2 net 192.168.1.20:7790 192.168.1.21:7790 C --set-de
faults --create-device --discard-my-data' terminated with exit code 10
[root@filer01 ~]#
```

Figure 57 Discard-Data sur le serveur primaire

Dans mon cas de figure c'est bel et bien le serveur primaire qui ne détient pas les données à jour. Il faut donc le passer en serveur secondaire afin de pouvoir exécuter la commande. Un redémarrage du serveur ou un arrêt du service drbd permet de faire cela.

Une fois la commande exécutée sur le serveur secondaire, les serveurs indiquent qu'il y a eu un problème de Split-Brain et que le problème a été résolu manuellement. La figure suivante montre la résolution et l'état drbd mis à jour pour le volume vg1drbd.

```
0:cluster_metadata Connected Secondary/Primary UpToDate/UpToDate C
1:vg0drbd Connected Secondary/Primary UpToDate/UpToDate C
2:vg1drbd WfConnection Secondary/Unknown UpToDate/DUnknown C
[root@filer01 ~]# drbd2: Split-Brain detected, manually solved. Sync from peer n
ode
service drbd status
drbd driver loaded OK; device status:
version: 8.2.7 (api:88/proto:86-88)
GIT-hash: 61b7f4c2fc34fe3d2acf7be6bcc1fc2684708a7d build by phil@fat-tyre, 2008-
11-12 16:47:11
m:res cs st ds p mounted
fstype
0:cluster_metadata Connected Secondary/Primary UpToDate/UpToDate C
1:vg0drbd Connected Secondary/Primary UpToDate/UpToDate C
2:vg1drbd Connected Secondary/Primary UpToDate/UpToDate C
[root@filer01 ~]#
```

Figure 58 Résolution du problème de Split-Brain

## 10.7 Problèmes rencontrés lors d'ajout de cibles ISCSI

Lors de la mise en place de cibles ISCSI sur le filer01, un test de reboot sur le serveur primaire a été effectué afin de contrôler le login de la cible ISCSI au redémarrage. Le résultat de cette opération s'est avéré concluant pour le filer01 qui s'est bien connecté sur la cible mais en revanche le service drbd a rencontré un problème. Le serveur 2 ne disposant pas de disque ISCSI connecté a refusé de devenir le serveur primaire, ce qui en soit est une bonne chose car il ne dispose pas de disque. L'état drbd est passé en Secondaire/Secondaire pour les deux serveurs, ce qui rend les partages et la réplication inactifs, comme le montre la figure suivante :

```
drbd2: State change failed: Refusing to be Primary without at least one UpToDate
disk
drbd2:  state = { cs:WfConnection st:Secondary/Unknown ds:Diskless/DUnknown r--
- }
drbd2:  wanted = { cs:WfConnection st:Primary/Unknown ds:Diskless/DUnknown r---
}

[root@filer02 ~]# service drbd status
drbd driver loaded OK; device status:
version: 8.2.7 (api:88/proto:86-88)
GIT-hash: 61b7f4c2fc34fe3d2acf7be6bcc1fc2684708a7d build by phil@fat-tyre, 2008-
11-12 16:47:11
m:res          cs          st          ds          p  mounte
d fstype
0:cluster_metadata Connected Secondary/Secondary UpToDate/UpToDate C
1:vg0drbd      Connected Secondary/Secondary UpToDate/UpToDate C
2:vg1drbd      Connected Secondary/Secondary Diskless/UpToDate C
[root@filer02 ~]# _
```

Figure 59 Problème de drbd avec cible ISCSI non connectée

Connexion de la cible ISCSI sur le filer02 et reboot de la machine :

```
[root@filer02 ~]# iscsiadm -m discovery -t st -p 192.168.1.201
192.168.1.201:3260,1 iqn.2010-07.com.openfiler:tsn.target2
[root@filer02 ~]# iscsiadm -m node -T iqn.2010-07.com.openfiler:tsn.target2 -p 1
92.168.1.201 --op update -n node.startup -v automatic
[root@filer02 ~]# reboot_
```

Figure 60 Connexion à la cible ISCSI

Une fois le filer02 redémarré, le service drbd a repris un état correct :

```
m:res          cs          st          ds          p  mounted
d fstype
0:cluster_metadata Connected Secondary/Primary UpToDate/UpToDate C
1:vg0drbd      Connected Secondary/Primary UpToDate/UpToDate C
2:vg1drbd      Connected Secondary/Primary UpToDate/UpToDate C
[root@filer02 ~]# _
```

Figure 61 État drbd après reboot

Cette manipulation n'est pas un problème majeur mais il faut tout de même faire attention à la démarche utilisée lors d'ajout de cible ISCSI, si l'opération doit être réalisée avec les serveurs en production.

## 10.8 Création de nom de disques ISCSI persistants

Pour résoudre le problème de noms statiques des disques dans la configuration du service drbd, le but est de ne plus utiliser un nom statique comme sdb1 ou sdc3 mais d'utiliser le nom de la cible ISCSI connectée.

Deux fichiers sont nécessaires pour établir un lien symbolique entre le nom de la cible ISCSI et son nom de disque sdX. Ces opérations sont à réaliser sur les deux serveurs.

Dans le dossier /etc/udev/rules.d, créez un fichier 55-openiscsi.rules contenant le script :

```
# /etc/udev/rules.d/55-openiscsi.rules
```



```
KERNEL=="sd*", BUS=="scsi", PROGRAM="/etc/udev/scripts/iscsidev.sh %b",SYMLINK+="iscsi/%c/part%n"
```

Dans le dossier `/etc/udev/scripts`, créez le script `iscsidev.sh` contenant :

```
#!/bin/sh

# FILE: /etc/udev/scripts/iscsidev.sh

BUS=${1}
HOST=${BUS%%:*}

[ -e /sys/class/iscsi_host ] || exit 1

file="/sys/class/iscsi_host/host${HOST}/device/session*/iscsi_session*/targetname"

target_name=$(cat ${file})

# This is not an open-scsi drive
if [ -z "${target_name}" ]; then
    exit 1
fi

# Check if QNAP drive
check_qnap_target_name=${target_name%%:*}
if [ $check_qnap_target_name = "iqn.2004-04.com.qnap" ]; then
    target_name=`echo "${target_name%.*}"`
fi

echo "${target_name}##*."
```

Changez les droits d'accès au script avec la commande :

```
chmod 755 /etc/udev/scripts/iscsidev.sh
```

Après la mise en place de ce script de mappage des cibles iSCSI, voici la configuration de la ressource `vg1drbd` à l'aide des liens symboliques créés :

```
resource vg1drbd {
    protocol C;
    startup {
        wfc-timeout 0; ## Infinite!
        degr-wfc-timeout 120; ## 2 minutes.
    }

    disk {
        on-io-error detach;
    }

    net {
        # timeout 60;
    }
}
```

```
# connect-int 10;
# ping-int 10;
# max-buffers 2048;
# max-epoch-size 2048;
}

syncer {
  after "cluster_metadata";
}

on filer01 {
  device /dev/drbd2;
  disk /dev/iscsi/target1/part1;
  address 192.168.1.20:7790;
  meta-disk internal;
}

on filer02 {
  device /dev/drbd2;
  disk /dev/iscsi/target2/part1;
  address 192.168.1.21:7790;
  meta-disk internal;
}
}
```

Cette solution de création de lien symbolique entre le disque physique et le nom de la cible ISCSI permet de :

- Eviter le problème de mauvaise synchronisation des disques avec drbd si un cible est manquante
- Isoler rapidement une cible ISCSI défectueuse
- Simplifier la gestion de la configuration du fichier drbd.conf

## 10.9 Snapshot du système

Openfiler permet de créer un snapshot du système depuis l'interface de management du serveur sous l'onglet système puis Backup/Restore :

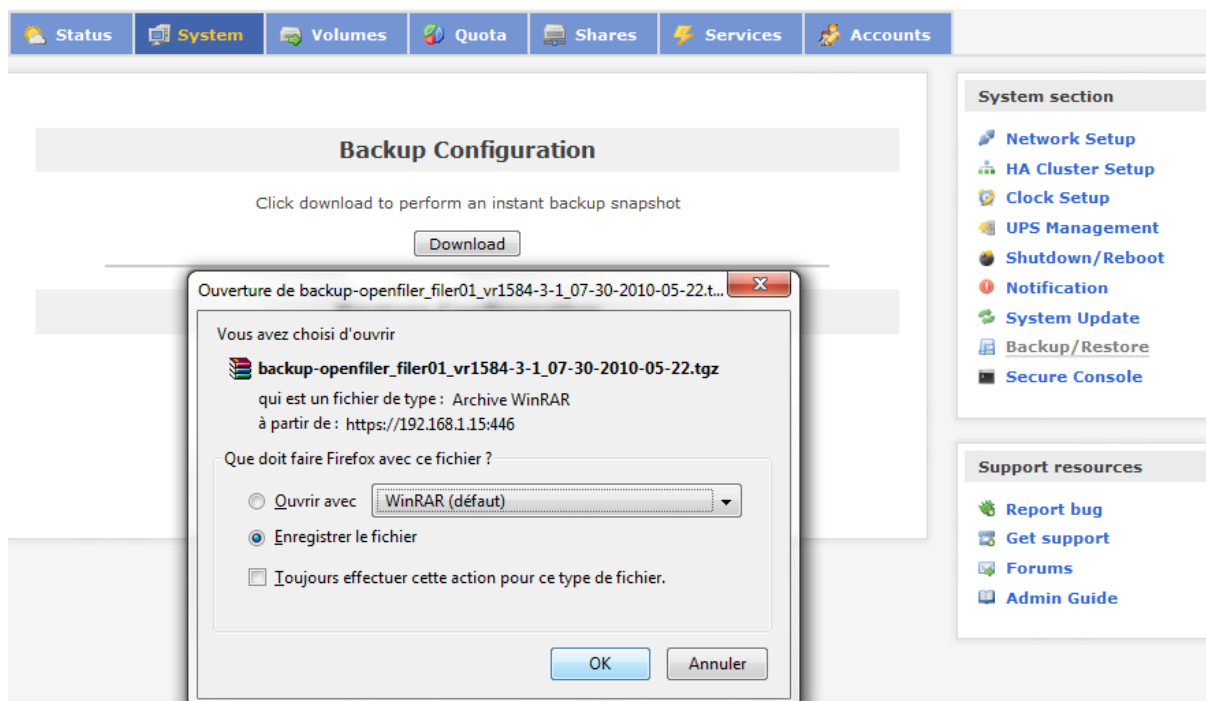


Figure 62 Snapshot du système

La restauration du système se fait par la même interface en uploadant la sauvegarde du système créée précédemment :

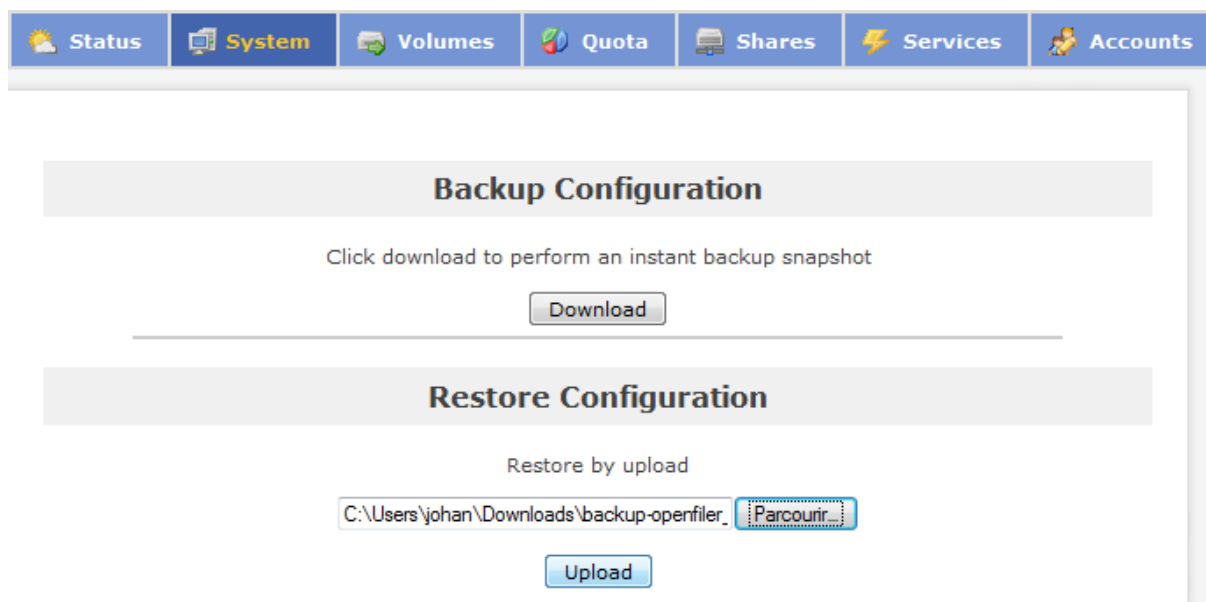


Figure 63 Restauration de la configuration du système

Pour créer un snapshot du serveur secondaire, il est nécessaire de la passer en serveur primaire et de recommencer l'opération de sauvegarde.

En cas de corruption du système Openfiler d'un des serveurs, il est nécessaire de recommencer l'installation du serveur comme il est décrit en annexe et de reconfigurer les cibles ISCSI. Cette opération peut prendre passablement de temps. Il est donc préférable de créer régulièrement une image Ghost du serveur ce qui simplifie et réduit le temps de réinstallation d'un serveur.

## **10.10 Intégration des serveurs au domaine de la Heig-vd**

L'intégration des serveurs au domaine n'a pas pu être réalisée. Une tentative d'authentification des utilisateurs au moyen du service LDAP a été faite. Le résultat de cette opération a rendu les deux serveurs inaccessibles à travers l'interface de management Web. Le choix a été pris de continuer les tests sur le système local et d'intégrer les serveurs au domaine en août, pour la défense du diplôme.

## **10.11 Conclusion sur le système Openfiler**

Ce système gratuit répond aux attentes demandées. La consultation du volume de données peut se faire grâce à l'interface graphique, ce qui est simple et rapide. La configuration doit toutefois être faite en majeure partie en ligne de commande linux, ce qui est le cas pour tous les systèmes de stockage gratuits actuels. Ce système a déjà fait ses preuves en production mais il manque encore de support sur le forum dédié d'Openfiler.

---

## 11 Déploiement pour la Heig-vd

---

Pour déployer cette solution basée sur Openfiler il est recommandé d'utiliser deux serveurs avec une architecture 64bits avec 4GB de ram pour le cluster de haute disponibilité. Pour assurer un minimum de redondance des cartes réseau, il est préférable d'utiliser une carte dédiée à la réplication des données. Les serveurs peuvent donc être connectés via un câble croisé directement ce qui limite le risque de Split-Brain décrit dans le cas de la perte du switch du réseau local. Les cartes ethernet du des serveurs en cluster peut être de type TOE (TCP Offload Engine). Cela décharge le processeur du serveur, car il devra traiter beaucoup de requêtes TCP sur le réseau de stockage et de la part des clients. Ces cartes sont accessibles à partir de 200Fr, ce qui est abordable et donc vivement conseillé. Pour l'interface ethernet sur le réseau Heig-vd, il est possible de créer une interface agrégée avec deux cartes. Les performances peuvent ainsi être améliorées pour les utilisateurs.

Les cibles ISCSI peuvent être des ordinateurs communs basés sur 32 bits avec 1GB de mémoire ram.

---

## 12 Etat des lieux

---

Une solution de stockage des données basée sur un logiciel libre et gratuit est détaillée et testée sur des machines virtuelles. En raison d'une installation des serveurs de haute disponibilité quelque peu aléatoire, le fichier haresources n'est pas toujours généré par Openfiler. Ce problème est connu mais il n'y a pas de solution disponible sur le forum actuellement. Le banc de tests définitifs n'a pas encore pu être installé correctement.

Le banc de test final sera installé pendant le mois d'août pour une démonstration du fonctionnement du système lors de la défense du projet de diplôme.

---

## 13 Conclusion

---

Ce projet de stockage est fort intéressant et m'a permis d'acquérir de nombreuses connaissances dans le domaine de la réplication des données ainsi que les différentes méthodes de stockage.

Après avoir configuré les serveurs basés sur Openfiler et sur CentOS, le choix du système de cluster de haute disponibilité n'a peut-être pas été judicieux. Le système CentOS a été écarté en raison de la configuration complexe en ligne de commande, mais cela a été pareil pour Openfiler. Le choix de conserver Openfiler a été prise car je possédais une plus grande expérience dans la configuration de ce système d'exploitation.

## 14 Sources et Références

---

[1] Citation de Jeff Bonwick, manager de l'équipe de développement du système de fichiers ZFS :

<http://fr.wikipedia.org/wiki/ZFS>

[2] Manuel d'installation d'Opensolaris en cluster HA :

<http://hub.opensolaris.org/bin/download/Project+colorado/files/Whitepaper-OpenHAClusterOnOpenSolaris-external.pdf>

[3] Formule de calcul de l'espace nécessaire au metadata du cluster drbd :

<http://www.drbd.org/users-guide/ch-internals.html>

<http://www.gluster.org/>

<http://en.wikipedia.org>

[http://www.howtoforge.com/high\\_availability\\_heartbeat\\_centos](http://www.howtoforge.com/high_availability_heartbeat_centos)

[http://wiki.fluidvm.com/index.php?title=SAN\\_Setup\\_on\\_CentOS\\_5.3](http://wiki.fluidvm.com/index.php?title=SAN_Setup_on_CentOS_5.3)

<http://www.howtoforge.com/installing-and-configuring-openfiler-with-drbd-and-heartbeat>

[https://project.openfiler.com/tracker/browser/openfiler/trunk/doc/cluster\\_guide/openfiler-ha.html?format=raw](https://project.openfiler.com/tracker/browser/openfiler/trunk/doc/cluster_guide/openfiler-ha.html?format=raw)

<https://forums.openfiler.com/>

<http://www.linux-ha.org/doc/>

[http://www.idevelopment.info/data/Unix/Linux/LINUX\\_ConnectingToAniSCSITargetWithOpen-iSCSIInitiatorUsingLinux.shtml](http://www.idevelopment.info/data/Unix/Linux/LINUX_ConnectingToAniSCSITargetWithOpen-iSCSIInitiatorUsingLinux.shtml)

<http://www.drbd.org/>

<http://wiki.centos.org/HowTos/Ha-Drbd>

## 15 Annexes

### 15.1 Installation d'Openfiler

Insérez le CD d'installation d'Openfiler et configurez le bios du PC afin de démarrer sur le lecteur de CD. Choisissez l'installation graphique du système d'exploitation en pressant enter.

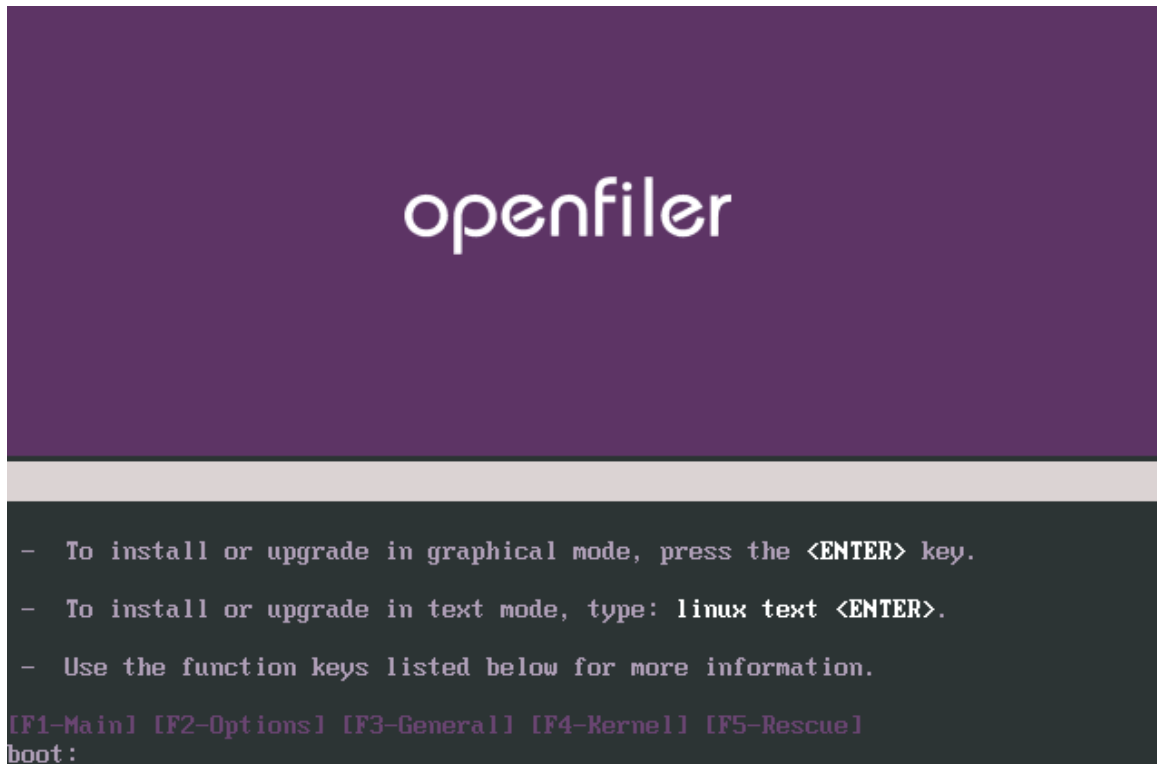


Figure 64 Début de l'installation du système

Il est possible de tester le CD d'installation, choisissez Skip.



Figure 65 Test du CD d'installation

Début de l'installation du système, cliquez sur Next.

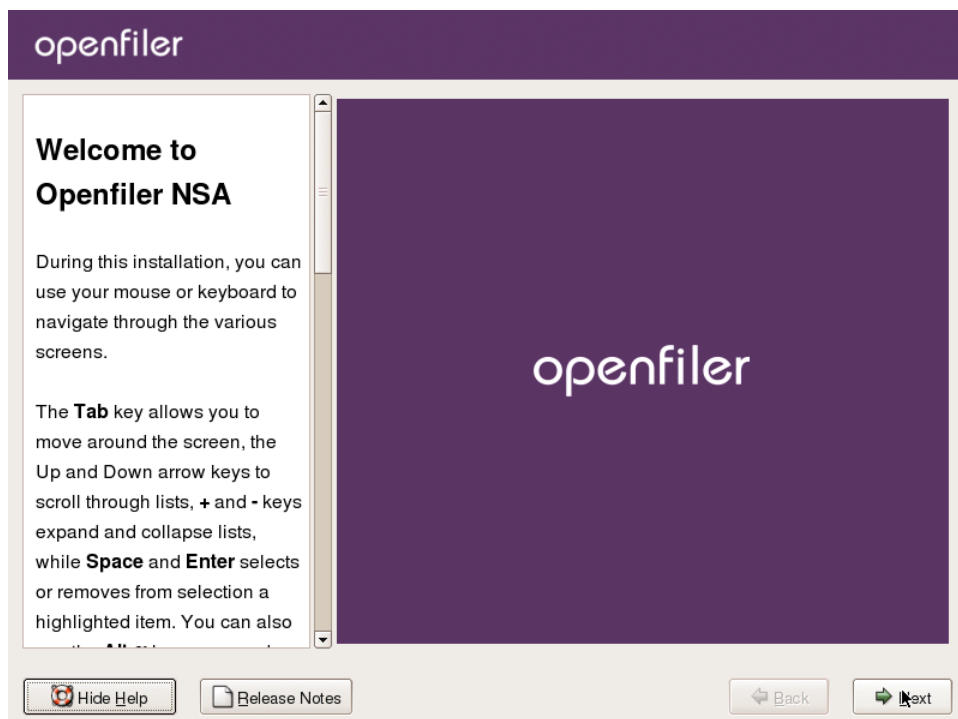


Figure 66 Début de l'installation du système

Sélection du choix de la langue du clavier du système, sélectionnez Swiss French et cliquez sur Next.

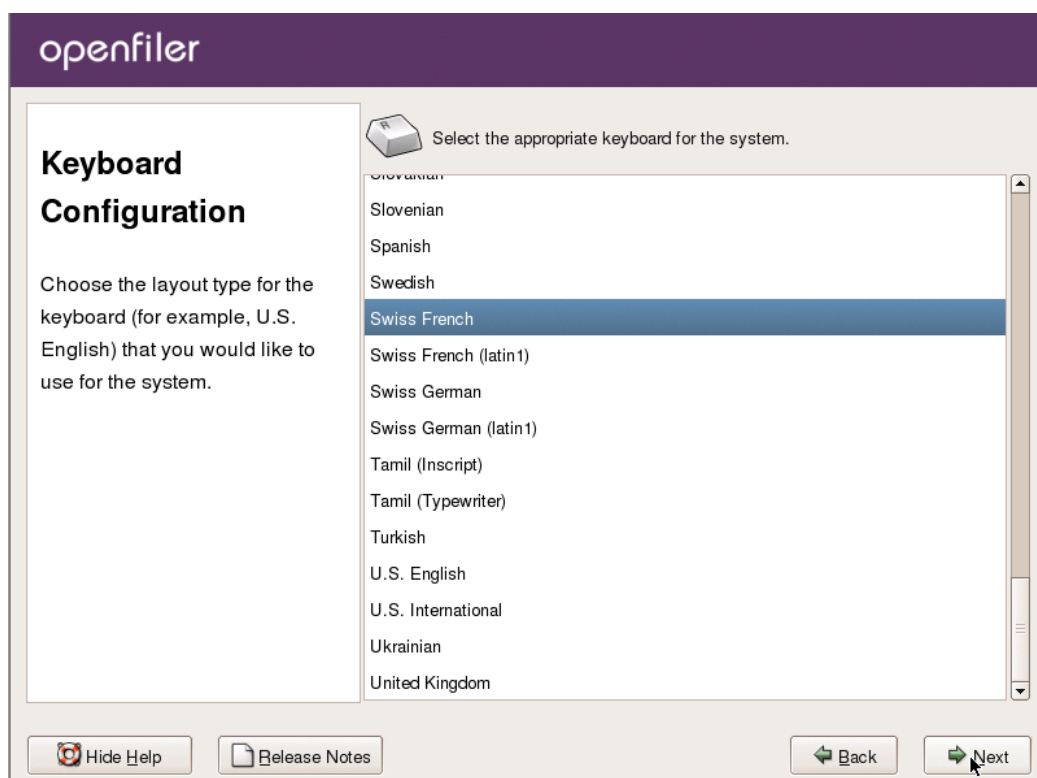


Figure 67 Sélection de la langue du clavier



Sélectionnez Manually partition with Disk Druid :

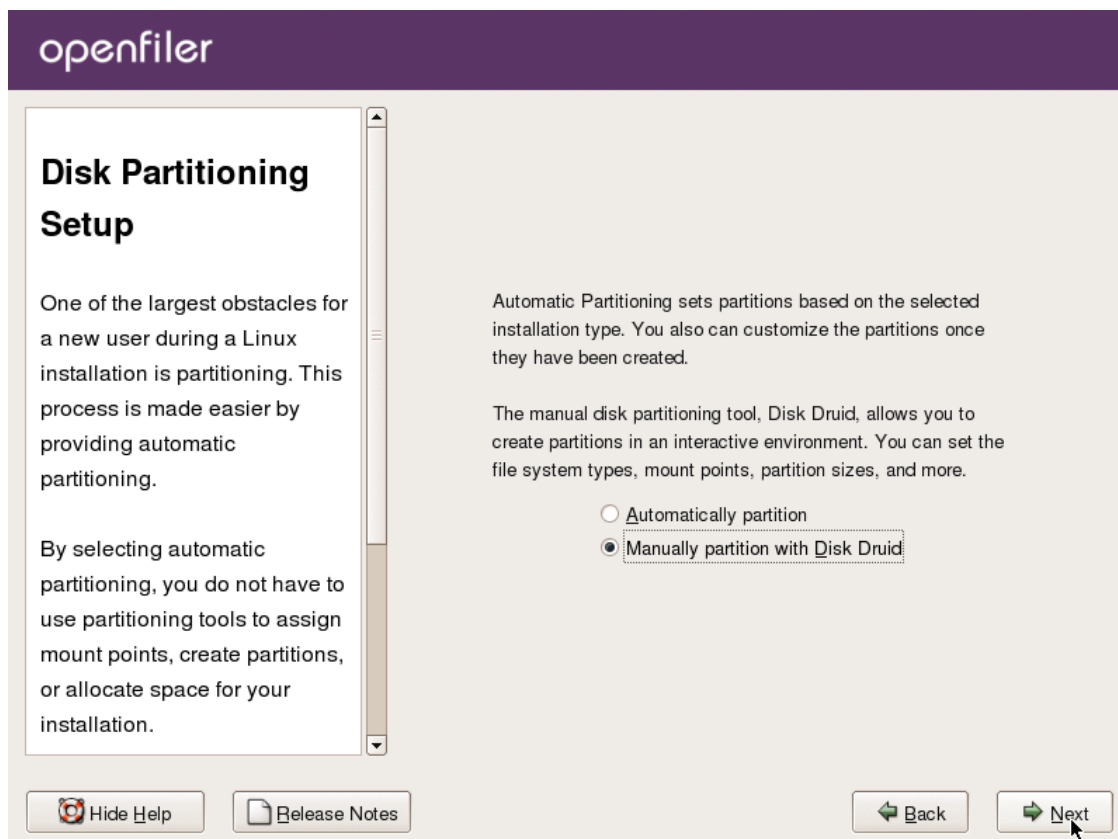


Figure 68 Sélection du partitionnement des disques

Voici la page de partitionnement des disques, cliquez sur la touche RAID :

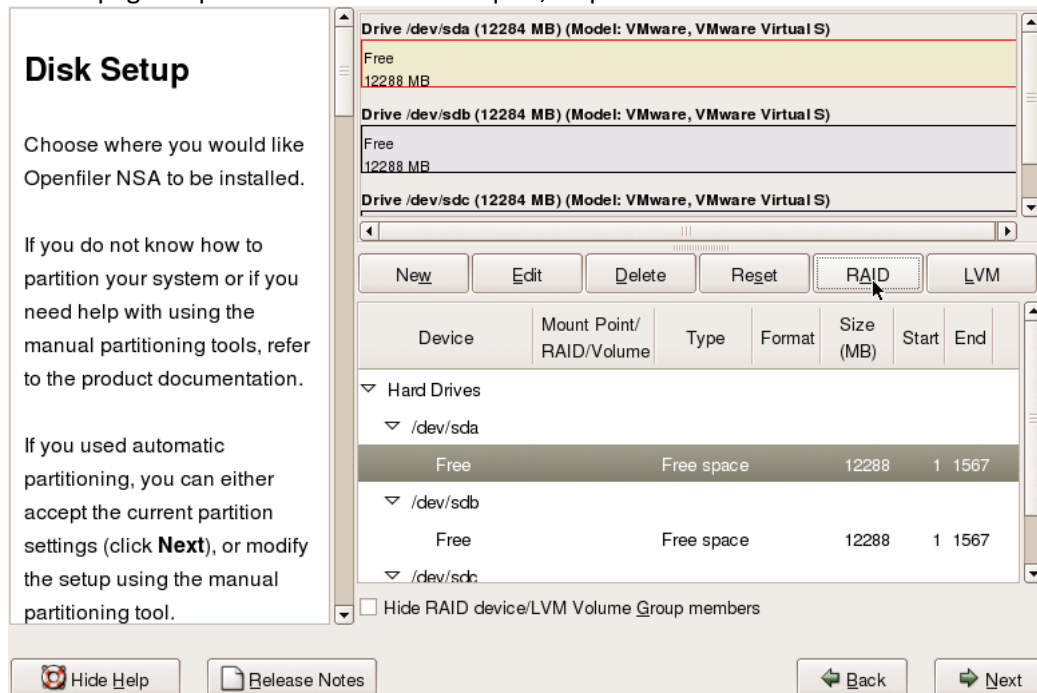


Figure 69 Partitionnement des disques

Le système indique qu'il n'y pas de partitions RAID software, cliquez sur OK :

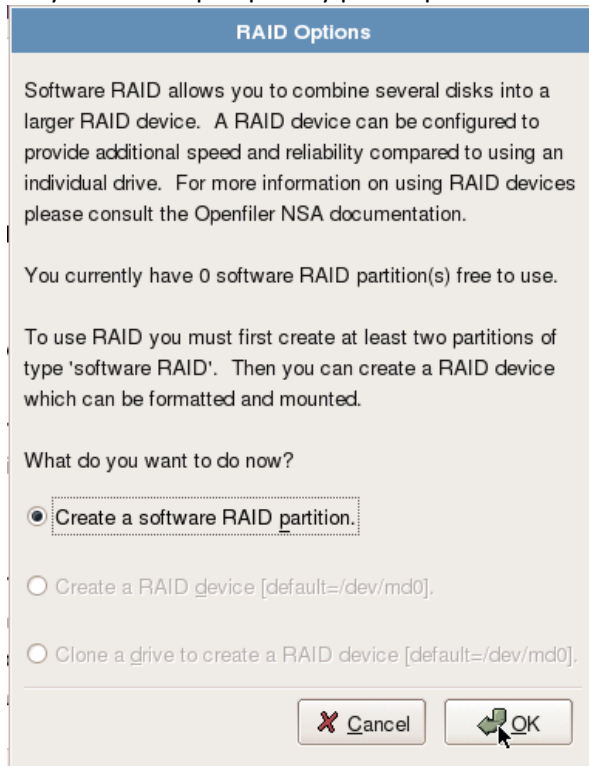


Figure 70 Début de la création du RAID

La fenêtre suivante apparaît, sélectionnez un disque et fixez une taille de partition à 6000MB :

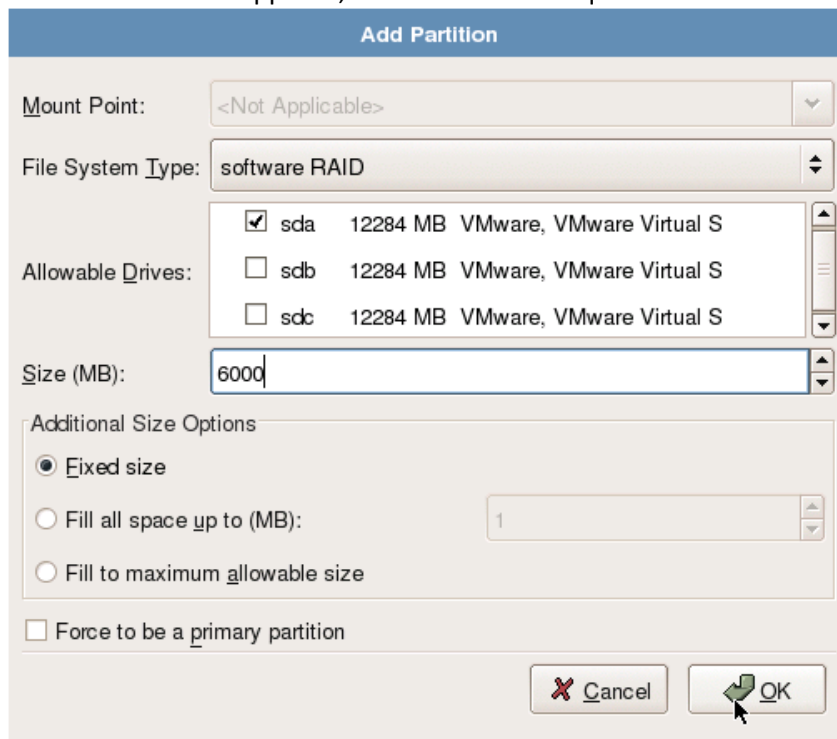
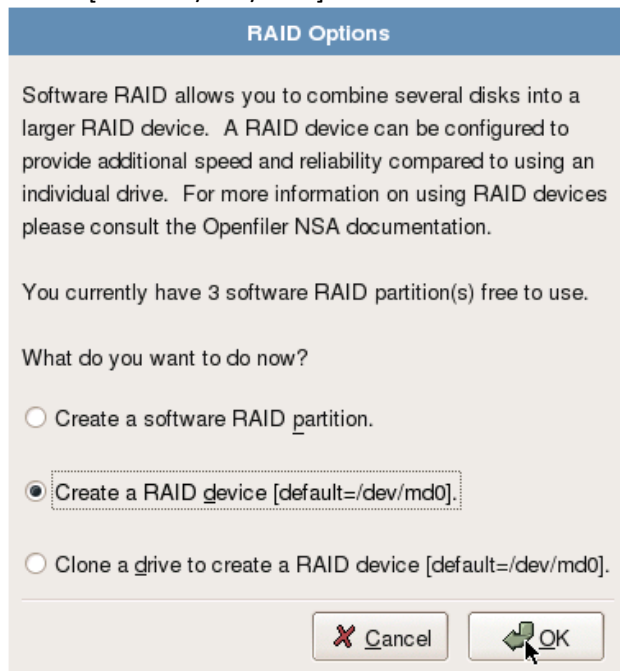


Figure 71 Sélection des disques pour le RAID

Répétez l'opération de création d'une partition RAID pour le deuxième disque sdb.

Une fois les trois partitions créées, cliquez encore sur le bouton RAID et sélectionnez Create RAID device [default=/dev/md0].



**RAID Options**

Software RAID allows you to combine several disks into a larger RAID device. A RAID device can be configured to provide additional speed and reliability compared to using an individual drive. For more information on using RAID devices please consult the Openfiler NSA documentation.

You currently have 3 software RAID partition(s) free to use.

What do you want to do now?

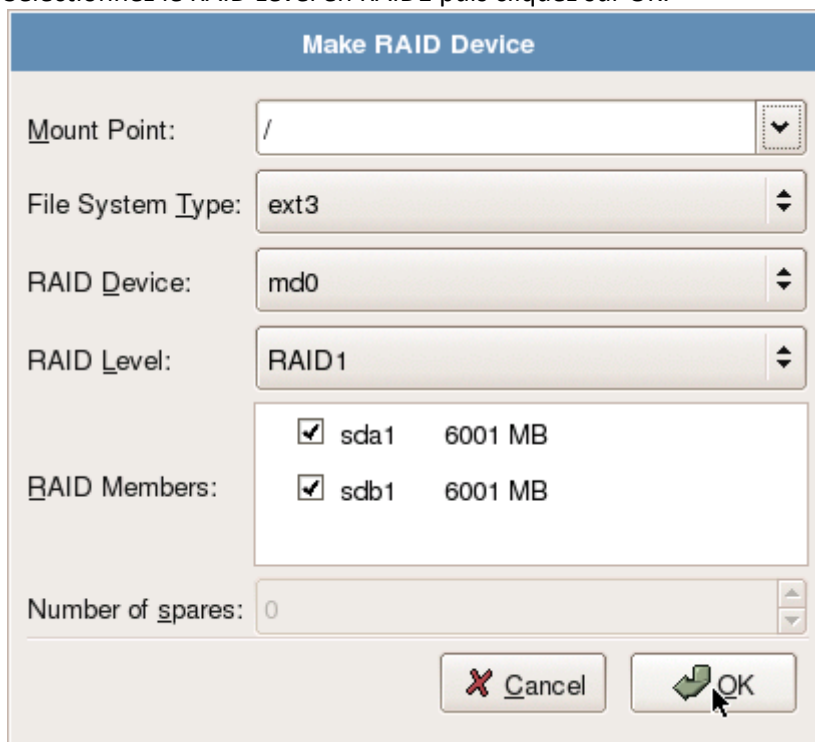
☐ Create a software RAID partition.

☒ Create a RAID device [default=/dev/md0].

☐ Clone a drive to create a RAID device [default=/dev/md0].

Figure 72 Création du raid md0

Sélectionnez le Mount Point à /. Cela indique que le système sera installé sur le RAID md0. Sélectionnez le RAID Level en RAID1 puis cliquez sur OK.



**Make RAID Device**

Mount Point: /

File System Type: ext3

RAID Device: md0

RAID Level: RAID1

RAID Members:

<input checked="" type="checkbox"/>	sda1	6001 MB
<input checked="" type="checkbox"/>	sdb1	6001 MB

Number of spares: 0

Figure 73 Choix du point de montage du système

Il est nécessaire de créer une partition de 2000MB de swap pour les systèmes linux, cette partition est à créer de préférence sur le troisième disque sdc.

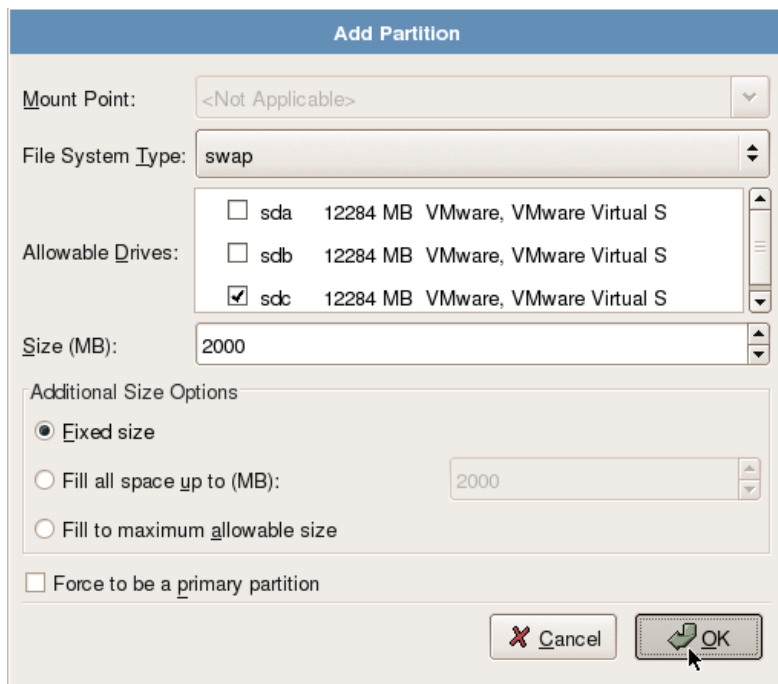


Figure 74 Création de la partition de swap

Une fois cette configuration des partitions terminée cliquez sur le Bouton OK. La fenêtre de configuration des cartes réseaux apparaît.

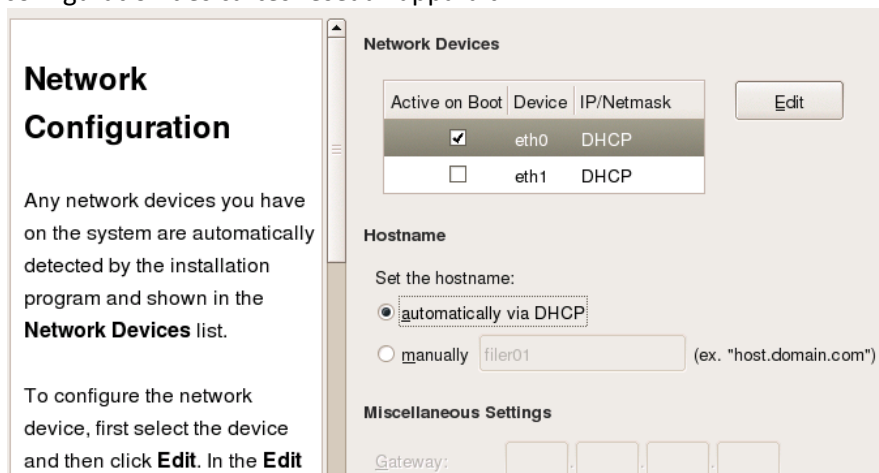


Figure 75 Configuration du réseau

Sélectionnez la carte réseau eth0 et cliquez sur le bouton Edit.

Décochez la case Configure using DHCP et entrez l'adresse IP désirée.

Figure 76 Configuration de l'adresse IP sur eth0

Configurez la carte eth1 en choisissant une adresse IP de réseau local, cette carte sera utilisée pour la réplication des données entre les deux serveurs et à accéder aux cibles ISCSI. Voici la configuration des cartes réseaux avec l'adresse IP 192.168.1.10 sur eth1.

Active on Boot	Device	IP/Netmask
<input checked="" type="checkbox"/>	eth0	10.192.49.216/255.255.248.0
<input checked="" type="checkbox"/>	eth1	192.168.1.10/255.255.255.0

**Hostname**

Set the hostname:

☐ automatically via DHCP

☒ manually  (ex. "host.domain.com")

**Miscellaneous Settings**

Gateway:  .  .  .

Primary DNS:  .  .  .

Secondary DNS:  .  .  .

Tertiary DNS:  .  .  .

Figure 77 Configuration finale des cartes réseaux

Sélectionnez la zone Europe/Zurich.

**Time Zone Selection**

Set your time zone by selecting your computer's physical location.

On the interactive map, click on a specific city (marked by a yellow dot) and a red X appears indicating your selection.

You can also scroll through the list of locations to select your desired time zone.

Please select the nearest city in your timezone:

Location Description

Europe/Zurich

☐ System clock uses UTC

Hide Help Release Notes Back Next

Figure 78 Sélection de la zone d'heure du système

Finalement pour terminer la configuration de l'installation, choisissez un mot de passe pour l'utilisateur root du système. Ce mot de passe est nécessaire à la configuration ultérieure du système.

The root account is used for administering the system.  
Enter a password for the root user.

Root Password: \*\*\*\*\*

Confirm: \*\*\*\*\*

Figure 79 Choix du mot de passe root

Cliquez sur le Bouton Next et l'installation peut débuter. Une fois l'installation terminée, retirez le CD d'installation et cliquez sur le bouton de redémarrage du serveur.

L'installation du serveur openfiler2 est similaire au premier serveur, choisissez comme adresses IP :  
Eth0 : 10.192.49.217 /255.255.248.0  
Eth1 : 192.168.1.11 /255.255.255.0

## 15.2 Configuration des serveurs Openfiler

### 15.2.1 Modification des fichiers host

Afin que les deux serveurs Openfiler puissent communiquer entre eux sur le réseau local, il faut ajouter une ligne de configuration dans le fichier `/etc/host` de chacun :

Il est possible de modifier le fichier directement en ligne de commande ou d'utiliser le logiciel winSCP.exe fournit sur le DVD pour plus de facilité :

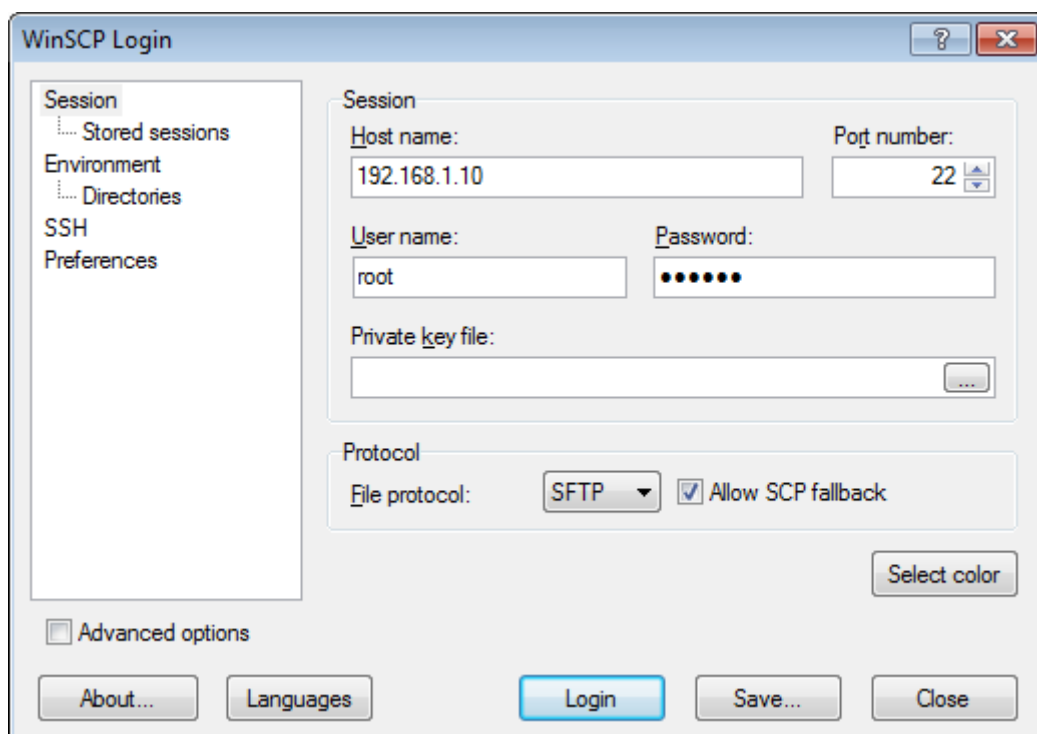


Figure 80 Connection au serveur openfiler

Il est dès lors facile de naviguer dans le système de fichier du serveur et d'éditer le fichier `/etc/host` :

Modification du fichier `/etc/host` sur openfiler1 :

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1          openfiler1 localhost.localdomain localhost
192.168.1.11      openfiler2
```

Modification du fichier `/etc/host` sur openfiler2 :

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1          openfiler2 localhost.localdomain localhost
192.168.1.10      openfiler1
```



### 15.2.2 Génération des clé SSH

Pour permettre au deux serveurs de communiquer entre eux sans utiliser de mot de passe, il faut générer une paire de clés ssh à l'aide de la commande :

```
root@openfiler1 ~# ssh-keygen -t dsa
```

La génération de la clé demande une phrase comme mot de passe, ce qui n'est pas nécessaire. Laissez la phrase vide et validez, voici l'illustration de la génération :

```
[root@openfiler1 ~]# ssh-keygen -t dsa
Generating public/private dsa key pair.

Enter file in which to save the key (/root/.ssh/id_dsa): Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_dsa.
Your public key has been saved in /root/.ssh/id_dsa.pub.
The key fingerprint is:
8d:2a:7a:11:26:df:d5:58:2b:69:29:8f:0e:e7:4b:f2 root@openfiler1
```

Figure 81 Génération de clé ssh

Une fois la clé générée, il faut l'exporter vers l'autre serveur à l'aide de la commande :

```
root@openfiler1 ~# scp ~/.ssh/id_dsa.pub root@openfiler2:~/.ssh/authorized_keys2
```

Cette commande nécessite le mot de passe root du serveur défini lors de l'installation. Voici l'illustration de l'exportation de la clé :

```
[root@openfiler1 ~]# scp ~/.ssh/id_dsa.pub root@openfiler2:~/.ssh/authorized_keys2
root@openfiler2's password:
id_dsa.pub                                100% 605      0.6KB/s   00:00
[root@openfiler1 ~]#
```

Figure 82 Copie de la clé sur l'autre serveur

Ensuite il faut répéter l'opération sur le serveur openfiler2 et exporter sa clé sur openfiler1 :

```
root@openfiler2 ~# ssh-keygen -t dsa
root@openfiler2 ~# scp ~/.ssh/id_dsa.pub root@openfiler1:~/.ssh/authorized_keys2
```



### 15.2.3 Edition du fichier drbd.conf

A l'aide de winSCP, éditez le fichier drbd.conf sur openfiler1 comme suit :

```
global {
    # minor-count 64;
    # dialog-refresh 5; # 5 seconds
    # disable-ip-verification;
    usage-count no;
}

common {
    syncer { rate 40M; }
}

resource cluster_metadata {

    protocol C;

    handlers {
        pri-on-incon-degr "echo O > /proc/sysrq-trigger ; halt -f";
        pri-lost-after-sb "echo O > /proc/sysrq-trigger ; halt -f";
        local-io-error "echo O > /proc/sysrq-trigger ; halt -f";
        # outdate-peer "/usr/sbin/drbd-peer-outdater";
    }

    startup {
        # wfc-timeout 0;
        degr-wfc-timeout 120; # 2 minutes.
    }

    disk {
        on-io-error detach;
    }

    net {
        after-sb-0pri disconnect;
        after-sb-1pri disconnect;
        after-sb-2pri disconnect;
        rr-conflict disconnect;
    }

    syncer {
        # rate 10M;
        # after "r2";
        al-extents 257;
    }

    on openfiler1 {
        device /dev/drbd0;
        disk /dev/sdc2;
        address 192.168.1.10:7788;
        meta-disk internal;
    }
}
```

```
on openfiler2 {
  device /dev/drbd0;
  disk /dev/sdc2;
  address 192.168.1.11:7788;
  meta-disk internal;
}
}

resource vg0_drbd {

  protocol C;
  startup {
    wfc-timeout 0; ## Infinite!
    degr-wfc-timeout 120; ## 2 minutes.
  }

  disk {
    on-io-error detach;
  }

  net {
    # timeout 60;
    # connect-int 10;
    # ping-int 10;
    # max-buffers 2048;
    # max-epoch-size 2048;
  }

  syncer {
    after "cluster_metadata";
  }

  on openfiler1 {
    device /dev/drbd1;
    disk /dev/sdc3;
    address 192.168.1.10:7789;
    meta-disk internal;
  }

  on openfiler2 {
    device /dev/drbd1;
    disk /dev/sdc3;
    address 192.168.1.11:7789;
    meta-disk internal;
  }
}
```

Il faut faire attention aux lignes mises en évidence en rouge, /dev/sdc2 doit correspondre à la partition créée pour les metadata du cluster et /dev/sdc3 correspond à une partition créée pour un volume de données à répliquer. Il faut peut-être adapter ces lignes en fonction de votre partitionnement du système

Le serveur openfiler2 nécessite le même fichier de configuration, soit vous le copiez via winSCP où utilisez la commande :

```
root@openfiler1 ~# scp /etc/drbd.conf root@openfiler2:/etc/drbd.conf
```

Initialisation des partitions /dev/sda3 et /dev/sda4 sur les deux serveurs :

```
root@openfiler1 ~# dd if=/dev/zero bs=1M count=1 of=/dev/sdc2
root@openfiler1 ~# dd if=/dev/zero bs=1M count=1 of=/dev/sdc3
root@openfiler2 ~# dd if=/dev/zero bs=1M count=1 of=/dev/sdc2
root@openfiler2 ~# dd if=/dev/zero bs=1M count=1 of=/dev/sdc3
```

#### 15.2.4 Création des dossiers pour la réplication

Il faut maintenant initialiser les metadata sur /dev/drbd0 et /dev/drbd1 à l'aide des commandes suivantes :

```
root@openfiler1 ~# drbdadm create-md cluster_metadata
root@openfiler1 ~# drbdadm create-md vg0_drbd
root@openfiler2 ~# drbdadm create-md cluster_metadata
root@openfiler2 ~# drbdadm create-md vg0_drbd
```

Ces commandes vont utiliser le fichier drbd.conf créée précédemment afin d'initialiser correctement cluster\_metadata sur la partition /dev/sda3 et vg0\_drbd sur /dev/sda4.

#### 15.2.5 Mise en route du service drbd

Exécutez les commandes suivantes sur chacun des serveurs :

```
root@openfiler1 ~# service drbd start
root@openfiler2 ~# service drbd start
```

Cela a pour effet de démarrer le service drbd. Si toutes les opérations se sont déroulées correctement, les deux serveurs doivent accepter la mise en route du service drbd comme suit :

```
[root@openfiler2 ~]# service drbd start
Starting DRBD resources:  [ d(cluster_metadata) d(vg0_drbd) s(cluster_metadata) s(vg0_drbd) n(
cluster_metadata) n(vg0_drbd) ].
.....[root@openfiler2 ~]#
[root@openfiler2 ~]#
```

Figure 83 Activation du service drbd

Contrôle du service drbd à l'aide de la commande :

```
service drbd status
```

L'état des partitions cluster\_metadata et vg0\_drbd doit être connecté et de type secondaire/secondaire comme le montre la figure suivante :

```
[root@openfiler1 ~]# service drbd status
drbd driver loaded OK; device status:
version: 8.2.7 (api:88/proto:86-88)
GIT-hash: 61b7f4c2fc34fe3d2acf7be6bcc1fc2684708a7d build by phil@fat-tyre, 2008-11-12 16:47:11
m:res          cs          st          ds          p mounted fstyp
e
0:cluster_metadata Connected Secondary/Secondary Inconsistent/Inconsistent C
1:vg0_drbd      Connected Secondary/Secondary Inconsistent/Inconsistent C
```

Figure 84 Statut du service drbd

On peut dès lors définir openfiler1 comme serveur primaire :

```
root@openfiler1 ~# drbdsetup /dev/drbd0 primary -o
root@openfiler1 ~# drbdsetup /dev/drbd1 primary -o
```

On contrôle la configuration du serveur :

```
root@openfiler1 ~# service drbd status
```

Le serveur est passé en mode Primaire/secondaire comme le montre la figure suivante :

```
[root@openfiler1 ~]# service drbd status
drbd driver loaded OK; device status:
version: 8.2.7 (api:88/proto:86-88)
GIT-hash: 61b7f4c2fc34fe3d2acf7be6bcc1fc2684708a7d build by phil@fat-tyre, 2008-11-12 16:47:11
m:res          cs          st          ds          p mounted fstype
...           sync'ed:    7.0%    (3595476/3855444)K
0:cluster_metadata SyncSource Primary/Secondary UpToDate/Inconsistent C
1:vg0_drbd      PausedSyncS Primary/Secondary UpToDate/Inconsistent C
```

Figure 85 Contrôle du serveur primaire

Création du système de fichier cluster\_metadata afin de contenir les fichiers de configuration d'openfiler nécessaires à la haute disponibilité :

```
root@openfiler1 ~# mkfs.ext3 /dev/drbd0
```

Cette commande initialise la partition en ext3 pour contenir les fichiers du cluster :

```
[root@openfiler1 ~]# mkfs.ext3 /dev/drbd0
mke2fs 1.40.8 (13-Mar-2008)
Warning: 256-byte inodes not usable on older systems
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
241440 inodes, 963861 blocks
48193 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=989855744
30 block groups
32768 blocks per group, 32768 fragments per group
8048 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736

Writing inode tables: done
Creating journal (16384 blocks): █
```

Figure 86 Système de fichiers pour la haute disponibilité entre les serveurs

### 15.2.6 Edition du fichier /etc/lvm/lvm.conf

Remplacer la ligne du fichier lvm.conf : filter = [ "a/\*/" ]  
Par l'expression suivante : filter = [ "r|/dev/sdc3|" ]

Sdc3 doit correspondre à la partition créée pour le volume de données. Si vous avez dû modifier le fichier drbd.conf, il faut l'adapter en conséquence avec la bonne partition.

Création du volume physique sur la partition sdc3

```
root@openfiler1 ~# pvcreate /dev/drbd1
```

```
[root@openfiler1 ~]# pvcreate /dev/drbd1
Physical volume "/dev/drbd1" successfully created
[root@openfiler1 ~]# █
```

Figure 87 Création du volume physique

Cette opération n'est à effectuer que sur le serveur primaire car cela sera répliqué automatiquement sur le secondaire via drbd.

## 15.3 Configuration du Heartbeat

Le heartbeat contrôle le bon fonctionnement du service openfiler entre les deux serveurs. Ce service envoie des informations de manière régulière afin de détecter une défaillance possible d'un serveur.

Edition du fichier /etc/ha.d/ha.cf sur les deux serveurs :

```
auth 2  
2 crc
```

Création du fichier /etc/ha.d/authkeys sur les deux serveurs :

```
debugfile /var/log/ha-debug  
logfile /var/log/ha-log  
logfacility local0  
bcast eth1  
keepalive 5  
warntime 10  
deadtime 120  
initdead 120  
udpport 694  
auto_failback off  
node openfiler1  
node openfiler2
```

Changement des permissions du fichier authkeys pour le compte root uniquement :

```
root@openfiler1 ~# chmod 600 /etc/ha.d/authkeys  
root@openfiler2 ~# chmod 600 /etc/ha.d/authkeys
```

Activation du service au démarrage :

```
root@openfiler1 ~# chkconfig --level 2345 heartbeat on  
root@openfiler2 ~# chkconfig --level 2345 heartbeat on
```

Configuration des données nécessaires à openfiler

Précédemment, une partition a été créée pour accueillir les données nécessaires au fonctionnement du cluster. Le but de cette opération est de déplacer les fichiers contenus dans le dossier /opt/openfiler dans la partition cluster\_metadata qui est répliquée entre les serveurs afin de conserver les paramètres dans le cas de la perte d'un serveur. Pour le bon fonctionnement, il faut créer un lien symbolique entre le dossier /opt/openfiler et le dossier /cluster\_metadata/opt/openfiler :

```
root@openfiler1 ~# mkdir /cluster_metadata  
root@openfiler1 ~# mount /dev/drbd0 /cluster_metadata
```

```
root@openfiler1 ~# mv /opt/openfiler/ /opt/openfiler.local
root@openfiler1 ~# mkdir /cluster_metadata/opt
root@openfiler1 ~# cp -a /opt/openfiler.local /cluster_metadata/opt/openfiler
root@openfiler1 ~# ln -s /cluster_metadata/opt/openfiler /opt/openfiler
root@openfiler1 ~# rm /cluster_metadata/opt/openfiler/sbin/openfiler
root@openfiler1 ~# ln -s /usr/sbin/httpd /cluster_metadata/opt/openfiler/sbin/openfiler
root@openfiler1 ~# rm /cluster_metadata/opt/openfiler/etc/rsync.xml
root@openfiler1 ~# ln -s /opt/openfiler.local/etc/rsync.xml /cluster_metadata/opt/openfiler/etc/
```

Edition du fichier /opt/openfiler.local/etc/rsync.xml

```
<?xml version="1.0" ?>
<rsync>
<remote hostname="192.168.1.11"/> ## IP du second serveur
<item path="/etc/ha.d/haresources"/>
<item path="/etc/ha.d/ha.cf"/>
<item path="/etc/ldap.conf"/>
<item path="/etc/openldap/ldap.conf"/>
<item path="/etc/ldap.secret"/>
<item path="/etc/nsswitch.conf"/>
<item path="/etc/krb5.conf"/>
</rsync>
```

Configuration du serveur openfiler2

```
root@openfiler2 ~# mkdir /cluster_metadata
root@openfiler2 ~# mv /opt/openfiler/ /opt/openfiler.local
root@openfiler2 ~# ln -s /cluster_metadata/opt/openfiler /opt/openfiler
```

Edition du fichier /opt/openfiler.local/etc/rsync.xml

```
<?xml version="1.0" ?>
<rsync>
<remote hostname="192.168.1.10"/> ## IP du second serveur
<item path="/etc/ha.d/haresources"/>
<item path="/etc/ha.d/ha.cf"/>
<item path="/etc/ldap.conf"/>
<item path="/etc/openldap/ldap.conf"/>
<item path="/etc/ldap.secret"/>
<item path="/etc/nsswitch.conf"/>
<item path="/etc/krb5.conf"/>
</rsync>
```

## Configuration du heartbeat entre les deux serveurs

Edition du fichier /cluster\_metadata/opt/openfiler/etc/cluster.xml sur openfiler1 uniquement :

```
<?xml version="1.0" ?>
<cluster>
<clustering state="on" />
<nodename value="openfiler1" />
<resource value="MailTo::it@company.com::ClusterFailover"/>
<resource value="IPAddr::10.192.49.218/24" />
<resource value="drbddisk::"/>
<resource value="LVM::vg0drbd"/>
<resource value="Filesystem::/dev/drbd0::cluster_metadata::ext3::defaults,noatime"/>
<resource value="MakeMounts"/>
</cluster>
```

Ce fichier permet au heartbeat de monter la partition /dev/drbd0 et le volume de réplication vg0drbd. L'adresse IP spécifiée est l'adresse de haute disponibilité désirée pour le cluster.

Création du volume group vg0drbd sur la partition drbd1

```
root@openfiler1 etc# vgcreate vg0drbd /dev/drbd1
```

```
[root@openfiler1 ~]# vgcreate vg0drbd /dev/drbd1
Volume group "vg0drbd" successfully created
```

Figure 88 Création du volume vg0drbd

```
root@openfiler1 ~# rm /opt/openfiler/etc/httpd/modules
root@openfiler1 ~# ln -s /usr/lib/httpd/modules /opt/openfiler/etc/httpd/modules
root@openfiler1 ~# service openfiler restart
```

Le redémarrage du service openfiler crée le fichier haresources nécessaire au fonctionnement du cluster. Copiez ce fichier sur le serveur openfiler2 :

```
root@openfiler1 ~# scp /etc/ha.d/haresources root@openfiler2:/etc/ha.d/haresources
```

Pour terminer la configuration du service de haute disponibilité, il est nécessaire de créer un volume logique sur le groupe de volume vg0drbd afin que le service heartbeat puisse démarrer correctement.

```
root@openfiler1 ~# lvcreate -L 40M -n filer vg0drbd
```

Rebootez le serveur openfiler1 puis le serveur openfiler2. L'interface de management des serveurs est désormais disponible sur l'adresse de haute disponibilité désirée.



## 16 Installation de GlusterFS

### 16.1 Préparation de l'installation

Il existe deux méthodes possibles pour l'installation du système :

- Création d'un CD à l'aide de l'image GlusterFS3.0.4.iso
- Création d'une clé USB

Afin d'installer le système à partir de la clé usb il faut la formater, cela nécessite un PC avec un système linux et les droits administrateurs. Une fois la clé usb reconnue par le pc linux, il faut déterminer son nom. Pour cela une méthode facile est d'ouvrir le dossier /dev et de consulter les fichiers présents sda,sdb,sdc, etc... Une fois la clé usb insérée elle sera automatiquement reconnue et un nouveau fichier sera créé portant le nom sd<Votre périphérique usb> à déterminer. Une fois le nom identifié vous pouvez formater la clé en utilisant la commande suivante :

« dd if=Gluster-3.0.4.img of=/dev/sd<Votre périphérique USB> bs=1M »



Cette étape requiert une grande prudence, car le formatage d'un autre disque que la clé usb souhaitée va entraîner la perte totale des données présentes sur le disque cible.

### 16.2 Installation du serveur principal

Une fois votre média d'installation prêt, il faut sélectionner le bon périphérique dans le Bios du PC afin de débiter l'installation du système GlusterFS. Lors du chargement de l'installation, sélectionnez Start First Server Installation :

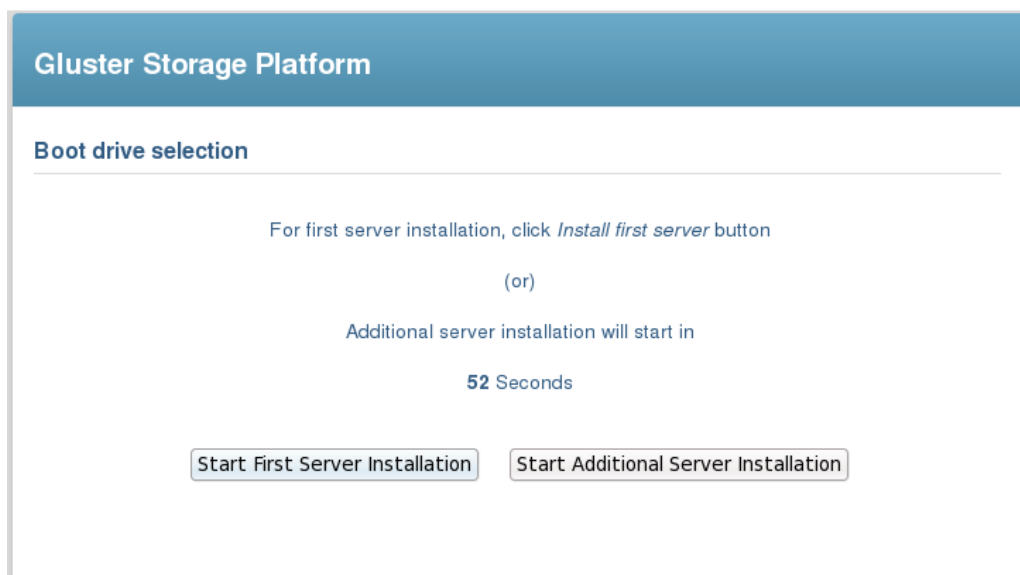


Figure 89 Installation de GlusterFS

La fin de l'installation est identique à l'ajout d'un serveur additionnel.

## 16.3 Ajout d'un serveur au cluster

Lors de l'installation d'un nouveau serveur il faut assigner :

- Un nom de serveur
- Un nom de domaine
- Une ou plusieurs adresses de serveur DNS
- Sélectionner les disques de stockage si plusieurs disques sont installés
- Une adresse IP et le masque de sous-réseau
- Cocher la case pour activer la configuration par la console de management

**Add Server**

Network Configuration | General | eth0

Hostname:

Domain Name:

Primary DNS:

Secondary DNS:

Third DNS:

Disk Configuration:

Storage:  ?

Time Settings:

Time zone:

Network time server:   
(example: pool.ntp.org or 192.168.1.1)

Figure 90 Configuration générale du serveur

**Add Server**

Network Configuration | General | eth0

Device: Intel Corporation 82567LM-3 Gigabit Network Connection

MAC Address:

IP Address:

Netmask:

Gateway:

Management Console: ☒

Figure 91 Configuration ethernet du serveur

## 16.4 Console de management du cluster

Une fois l'ajout du serveur terminé, celui-ci apparaît dans l'interface Web comme ci-dessous.

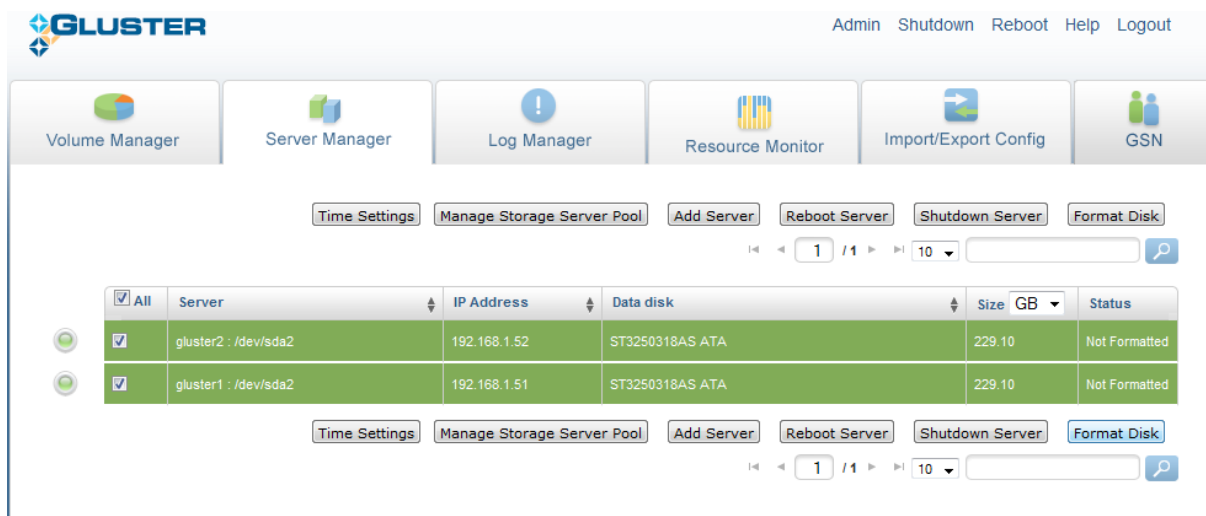


Figure 92 Console de management

L'interface permet de formater les disques disponibles dans le cluster :

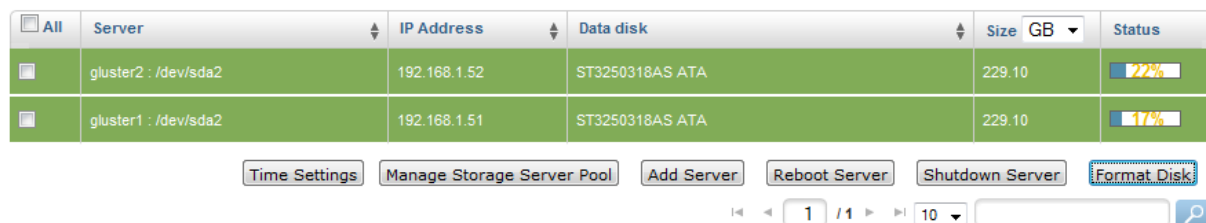


Figure 93 Formatage des disques en cours

Une fois le formatage des disques effectué, il est possible de créer un nouveau volume de données. Il faut pour cela cliquer dans l'onglet Volume manager comme le montre la figure suivante :

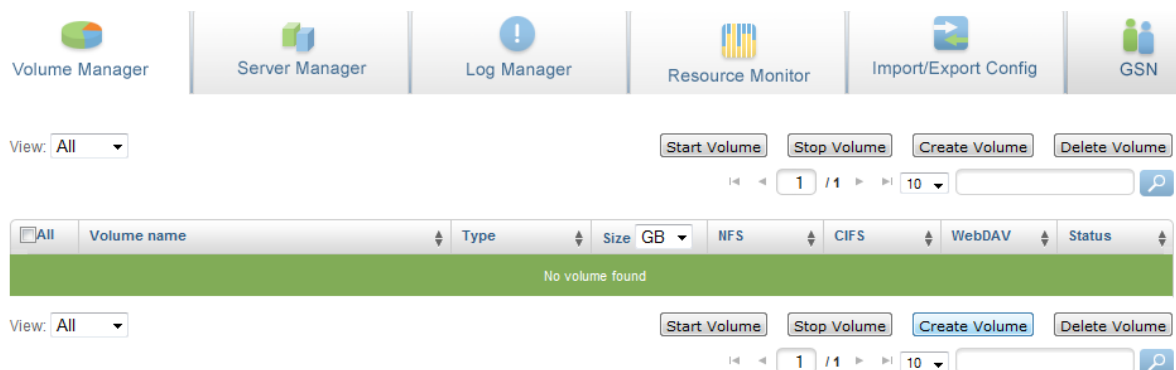


Figure 94 Volume manager

Pour créer un volume il suffit de cliquer sur le bouton Create Volume, ce qui fournit l'interface suivante :

**Create Volume**

Volume name:

Volume type: ☐ None ☒ Mirror ☐ Stripe ?

Transport type: ☒ Ethernet ☐ Infiniband

Size (in Gigabytes):

Storage servers:

<input checked="" type="checkbox"/> All	Server	Capacity
<input checked="" type="checkbox"/>	gluster2: /dev/sda2	229.10
<input checked="" type="checkbox"/>	gluster1: /dev/sda2	229.10

Volume access control:  ?  
Comma separated IP addresses in wildcard pattern

Exported as:

- ☒ GlusterFS Native
- ☒ NFS
- ☒ CIFS
  - User name:
  - Password:
  - Confirm Password:
  - Password matches
- ☐ WebDAV

Figure 95 Création d'un volume de donnée

L'interface permet de sélectionner les serveurs sur lesquels sera créé le volume. Il faut cocher le Volume type en Mirror afin d'activer la fonction de Raid1 et la réplication des données sur les serveurs.

Une fois le volume créé, le système indique comment le volume peut être monté suivant les protocoles GlusterFS, NFS ou accéder directement par CIFS. Pour activer ce volume, il faut valider en cliquant sur Start Volume.

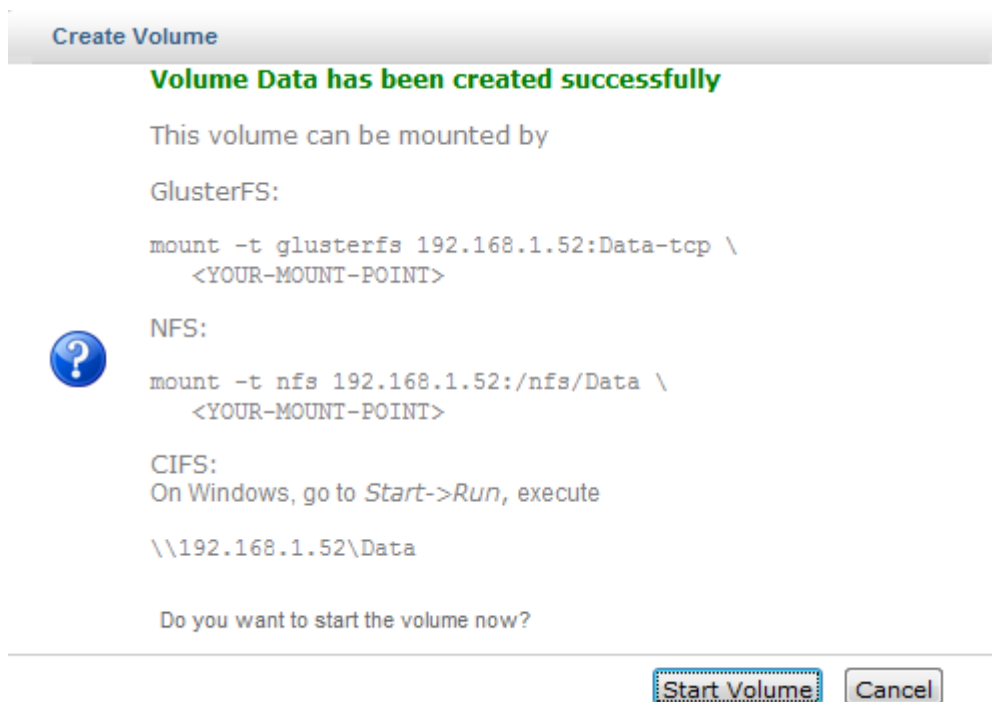


Figure 96 Activation du volume de donnée

Le système confirme l'activation du volume :

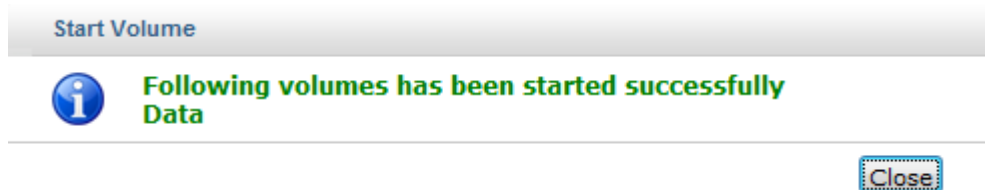


Figure 97 Activation du volume terminée

Le volume Data est prêt à être utilisé par les serveurs de haute disponibilité via le protocole NFS.