

Projet Tuteurés -Outils de gestion centralisée de
machines virtuelles
Tuteur : Lucas Nussbaum

Sébastien Michaux - Augustin Bocca - Julien Tournois - Mathieu Lamouroux

21 mars 2012

Résumé

Remerciements
Toujours en cours de réflexion :)

Table des matières

1	Introduction	3
1.1	Présentation du projet	3
1.2	Introduction à la virtualisation	3
1.2.1	Machine virtuelle	3
1.2.2	Hyperviseur	3
1.2.3	Enjeux de la virtualisation	4
1.2.4	Histoire de la virtualisation	4
1.2.5	Au commencement, la virtualisation des mainframes	5
1.3	Présentation de Grid5000	5
1.3.1	Infrastructure des sites	7
1.3.2	Réseau	9
1.3.3	Environnement logiciel	9
1.4	Répartition des tâches	9
2	Ganeti	10
2.1	Présentation	10
2.2	Installation	10
2.3	Utilisation	11
3	OpenXENManager	12
3.1	Présentation	12
3.2	Installation	12
3.3	Utilisation	13
3.4	Installation côté serveur	13
3.5	Installation côté client	13
A	Sources	15
B	Scripts	16
B.1	réservation	16
B.2	déploiement	17
B.3	configuration de base	18
B.4	Listing d'une installation de Ganetti	19
C	Glossaire	21

Chapitre 1

Introduction

1.1 Présentation du projet

Mettre en place, évaluer et comparer différents outils permettant de gérer de manière centralisée et automatisée une infrastructure basée sur des machines virtuelles : Ganeti, OpenXenManager, virt-manager, Archipel...

1.2 Introduction à la virtualisation

1.2.1 Machine virtuelle

Une machine virtuelle est un conteneur isolé capable d'exécuter ses propres système d'exploitation et applications. Une machine virtuelle se comporte exactement comme un ordinateur physique et contient ses propres processeur, mémoire RAM, disque dur et carte d'interface réseau virtuels. Une machine virtuelle a pour but de générer sur une même machine un ou plusieurs environnements d'exécution applicative. On en distingue deux types d'application : d'une part la virtualisation par le biais d'un hyperviseur jouant le rôle d'émulateur de système (PC ou serveur), d'autre part la virtualisation applicative qui permet de faire tourner une application sur un poste client quelque soit le système sous-jacent.

1.2.2 Hyperviseur

La machine virtuelle avec hyperviseur est utilisée pour générer au dessus d'un système d'exploitation serveur une couche logicielle sous la forme d'un émulateur permettant de créer plusieurs environnements d'exécution serveur. Cet émulateur se place comme un niveau supplémentaire qui se greffe sur le système d'origine.

1.2.3 Enjeux de la virtualisation

Actuellement, les entreprises rencontrent des besoins qui ne sont pas couverts.

Au niveau de la sécurité, les entreprises souhaiteraient isoler les services sur des serveurs différents. Pour la maintenance, il serait utile d'améliorer des services tels que la disponibilité, la migration, la redondance, la flexibilité ou le temps de réponse. Il serait également bienvenu de tester, déléguer l'administration d'un système ...

Une des solutions pour répondre à ces besoins serait d'acquérir davantage de plateformes de travail.

La multiplication des serveurs pose cependant un certain nombre de problèmes, augmenter sans cesse son parc informatique est impossible pour plusieurs raisons :

- Tout d'abord au niveau écologique cela entraînerait un surplus de déchets électroniques, une consommation d'énergie directe et de l'énergie utilisée pour refroidir les salles serveur.
- Au niveau de la surface utilisée, les salles machine seraient vite encombrées, puis apparaîtra des problèmes tels que la nuisance sonore, le manque de puissance pour alimenter les salles serveur.
- Au niveau économique les coûts d'achat, de recyclage, de fonctionnement, de maintenance seraient trop chers. La mise en place de serveur de virtualisation est une solution pour résoudre ces problèmes.

Le but de la virtualisation est de donner un environnement système au programme pour qu'il croie être dans un environnement matériel. Pour cela, une machine virtuelle est utilisée. Ainsi, plusieurs environnements d'exécution sont créés sur une seule machine, dont chacun émule la machine hôte. L'utilisateur pense posséder un ordinateur complet pour chaque système d'exploitation alors que toutes les machines virtuelles sont isolées entre elles.

1.2.4 Histoire de la virtualisation

La virtualisation est un concept qui a été mis au point pour la première fois dans les années 1960 pour permettre la partition d'une vaste gamme de matériel mainframe et optimiser l'utilisation du matériel. De nos jours, les ordinateurs basés sur l'architecture x86 sont confrontés aux mêmes problèmes de rigidité et de sous-utilisation que les mainframes dans les années 1960. VMware a inventé la virtualisation pour la plate-forme x86 dans les années 1990 afin de répondre notamment aux problèmes de sous-utilisation, et a surmonté les nombreux défis émergeant au cours de ce processus. Aujourd'hui, VMware est devenu le leader mondial de la virtualisation x86 avec plus de 190 000 clients, dont la totalité des membres du classement Fortune 100.

1.2.5 Au commencement, la virtualisation des mainframes

La virtualisation a été mise en œuvre pour la première fois il y a plus de 30 ans par IBM pour partitionner logiquement des mainframes en machines virtuelles distinctes. Ces partitions permettaient un traitement « multitâche », à savoir l'exécution simultanée de plusieurs applications et processus. Étant donné que les mainframes consommaient beaucoup de ressources en même temps, le partitionnement constituait un moyen naturel de tirer pleinement parti de l'investissement matériel.

1.3 Présentation de Grid5000



Aujourd'hui, grâce à Internet, il est possible d'interconnecter des machines du monde entier pour traiter et stocker des masses de données. Cette collection hétérogène et distribuée de ressources de stockage et de calcul a donné naissance à un nouveau concept : les grilles informatiques.

L'idée de mutualiser les ressources informatiques vient de plusieurs facteurs, évolution de la recherche en parallélisme qui, après avoir étudié les machines homogènes, s'est attaquée aux environnements hétérogènes puis distribués ; besoins croissants des applications qui nécessitent l'utilisation toujours plus importante de moyens informatiques forcément répartis.

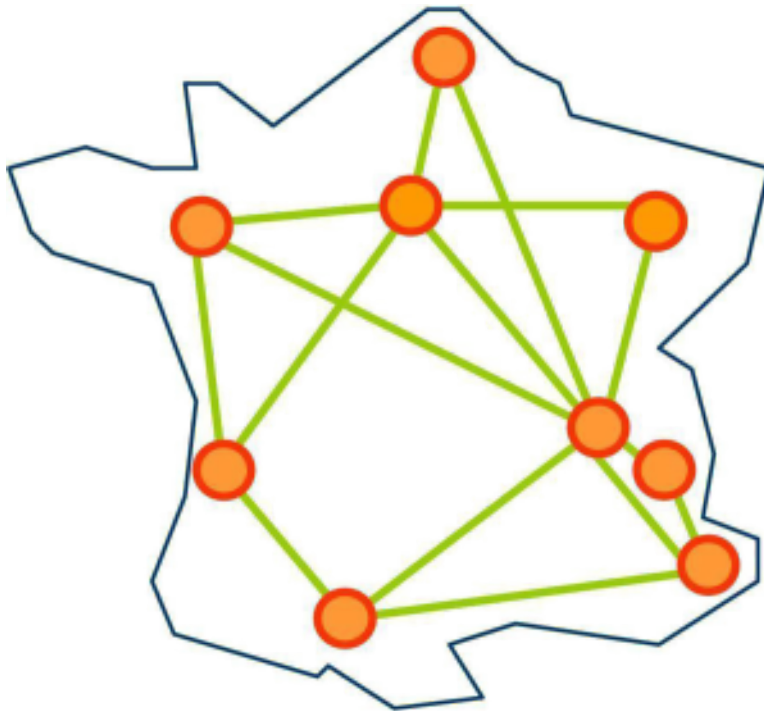
La notion de grille peut avoir plusieurs sens suivant le contexte : grappes de grappes, environnements de type GridRPC (appel de procédure à distance sur une grille), réseaux pair-à-pair, systèmes de calcul sur Internet, etc... Il s'agit d'une manière générale de systèmes dynamiques, hétérogènes et distribués à

large échelle. Un grand nombre de problématiques de recherche sont soulevées par les grilles informatiques. Elles touchent plusieurs domaines de l'informatique : algorithmique, programmation, intergiciels, applications, réseaux.

L'objectif de GRID'5000 est de construire un instrument pour réaliser des expériences en informatique dans le domaine des systèmes distribués à grande échelle (GRID).

Cette plate-forme, ouverte depuis 2006 aux chercheurs de la communauté grille, regroupe un certain nombre de sites répartis sur le territoire national. Chaque site héberge une ou plusieurs grappes de processeurs. Ces grappes sont alors interconnectées via une infrastructure réseau dédiée à 10 Gb/s fournie par RENATER. À ce jour, GRID'5000 est composé de 9 sites : Lille, Rennes, Orsay, Nancy, Bordeaux, Lyon, Grenoble, Toulouse et Nice.

Début 2007, GRID'5000 regroupait plus de 2500 processeurs et près de 3500 cœurs.



Architecture Grid5000

1.3.1 Infrastructure des sites

Chaque site héberge :

- un frontend, serveur permettant d'accéder aux clusters disponibles ,
- un serveur de données, pour centraliser les données utilisateurs ,
- plusieurs clusters, c'est-à-dire des grappes de machines homogènes, appelées noeuds (nodes).



L'utilisateur de Grid 5000 accède à chaque site par son frontend en utilisant le protocole SSH.
 Commande :

```
1 ssh jdoe@access.grid5000.fr
```

Sur tous les serveurs du site, un répertoire home, local à chaque site, est monté avec NFS 2 . A partir du frontend, il est possible d'accéder aux machines des clusters en exectuant des réservations à l'aide de la commande :

```
1 oarsub
```

Grâce à notre tuteur, M. Lucas Nussbaum nous avons pu visiter la salle serveurs du site de Nancy située au Loria, ainsi qu'une présentation de la plate-forme (matériel utilisé, connexions réseau, administration).

Une description détaillée du site de Nancy est disponible sur le site de Grid 5000.

1.3.2 Réseau

Les sites et toutes les machines qu'ils comprennent sont interconnectés par RENATER 3 en 10Gbits/s. De plus, chaque site peut disposer de plusieurs réseaux locaux 4 :

- réseau en ethernet, 1 Gb/s
- réseaux hautes performances (Infiniband 20 Gb/s ou 10 Gb/s, et Myrinet 20 Gb/s)

1.3.3 Environnement logiciel

Tous les serveurs de Grid 5000 fonctionnent sous Debian GNU/Linux. A partir du frontend, l'utilisateur peut réserver des machines en utilisant la suite de logiciels OAR dédiée à la gestion de ressources de clusters, et déployer ses propres images de systèmes à l'aide des outils kadeploy. Il y a deux types de réservation :

- par défaut, pour des besoins de calcul avec OpenMPI;
- pour le déploiement d'environnements (deploy).

1.4 Répartition des tâches

Nous avons commencé par prendre en main Grid5000 durant les 2 premières semaines du projet. Pour ce faire nous avons suivi avec soin les tutoriels mis à notre disposition sur le site www.grid5000.fr.

Une fois les manipulations de bases bien assimilées. Nous nous sommes divisés en 2 sous-groupes pour tester les différents outils du projet :

- Julien et Augustin se sont chargés de Ganeti.
- Sébatien et Mathieu pour OpenXenManager.

A l'heure actuelle nous n'avons pas encore eu le temps de nous projeter sur les autres outils et ne les avons donc pas répartis.

Chapitre 2

Ganeti

2.1 Présentation

Ganeti est un outil de gestion de machines virtuelles se basant sur les technologies de virtualisation existantes comme XEN et KVM. Ganeti nécessite un logiciel de virtualisation pré-installé sur les serveurs afin de pouvoir fonctionner.

Une fois installé, l'outil prendra en charge la partie gestion des instances virtuelles (Xen DomU), par exemple, la gestion de création de disque, l'installation du système d'exploitation (en coopération avec les scripts d'installation du système d'exploitation spécifique), et le démarrage, l'arrêt, le basculement entre les systèmes physiques. Il a été conçu pour faciliter la gestion de cluster de serveurs virtuels et de fournir une récupération rapide et simple.

2.2 Installation

installer ganeti :

```
1 apt-get install ganeti2 ganeti-htools ganeti-instance-  
debootstrap
```

Dans `/etc/host` ajouter l'adresse IP du node avec le nom de l'host ainsi que l'adresse du cluster :

```
1 10.144.64.1 cluster1  
2 172.16.65.56 griffon-56.nancy.grid5000.fr
```

Lorsque on aura plusieurs nodes il faudra une autre adresse Ip au cluster car elle doit être accessible à tous les nodes

Dans `/boot/` créer des liens symboliques :

```
1 ln -s vmlinuz-2.6.32-5-xen-amd64 vmlinuz-2.6.xenU
```

```
2 ln -s initrd.img-2.6.32-5-xen-amd64 initrd.img-2.6.xenU
```

Créer un LVM d'au moins 20Go (Obligatoire) :

```
1 umount /dev/sda5
2 pvcreate /dev/sda5
3 vgcreate xenvg /dev/sda5
```

Dans /ect/network/interface remplacer le paragraphe de eth0 par celle du brige xen-br0 :

```
1 auto xen-br0
2 iface xen-br0 inet static
3   address 172.16.65.56 #Adresse du node
4   netmask 255.255.240.0
5   network 172.16.64.0
6   broadcast 172.16.79.255
7   gateway 10.144.64.254
8   bridge_ports eth0
9   bridge_stp off
10  bridge_fd 0
```

2.3 Utilisation

Initialiser le cluster avec ganeti :

```
1 gnt-custer init --no-drbd-storage cluster1
```

ajouter le node

```
1 gnt-node add griffon-56.nancy.grid5000.fr
```

et vérifier avec :

```
1 gnt-node list
```

Qui devrait renvoyer quelque chose comme cela :

Node	Pinst	Sinst	DTotal	DFree	MTotal	MNode	MFree
griffon-56.nancy.grid5000.fr			283.2G	283.2G	-	-	-
	0						0

Chapitre 3

OpenXENManager

3.1 Présentation

XenseMaking Project développe un client lourd, ainsi qu'un client web, pour manager XenServer. C'est un clone du XenCenter, qui fonctionne avec Linux, BSD, Windows et MacOSX, alors que le XenCenter ne fonctionne qu'avec Windows. OpenXenManager/OpenXenCenter est le client lourd qui permet de manager XenServer. Il a été développé en Python avec pygtk et gtk-vnc. Les fonctionnalités actuellement implémentées sont les suivantes :

- monitoring des machines virtuelles - accès à la console des machines virtuelles
- opérations d'administration (démarrage, arrêt, reboot, ...)
- création de machines virtuelles

3.2 Installation

Pour l'installation nous avons besoin des paquets suivant :

```
1 apt-get install subversion bzip2 python-glade2 python-gtk-vnc  
shared-mime-info graphviz
```

On télécharge la dernière version d'openxenmanager dans le dépôt subversion

```
1 svn co https://openxenmanager.svn.sourceforge.net/svnroot/  
openxenmanager openxenmanager
```

On se déplace dans le répertoire trunk :

```
1 cd openxenmanager/trunk
```

Finalement on lance openxenmanager avec la commande suivante

```
1 python window.py
```

Une interface graphique d'openxenmanger apparait.

3.3 Utilisation

Pour l'instant nous n'avons toujours pas réussi à utiliser ce logiciel. DU a diverses difficultés (pas de service distant sous squeeze et plus de xen lors de la migration sous unstable).

Chapitre 4

Installation de virt-manager

4.1 Installation côté serveur

Cette partie est facile, un simple apt-get install suffit. Nous installons le paquet qui communique avec Xen et remonte les informations au client virt-manager.

```
1 apt-get install libvirt-bin
```

Du côté de Xen, nous devons vérifier qu'il peut communiquer avec libvirt.

Libvirt accède aux données de Xen via un socket unix. La configuration consiste à activer cette option dans Xen et à relancer les services. Nous éviterons ainsi l'erreur libvirtError : internal error failed to connect to xend dont on trouve peu d'explication sur le net.

On édite le fichier de configuration xen

```
1 nano /etc/xen/xend-config.sxp
```

on active la ligne suivante

```
1 (xend-unix-server yes)
```

Enfin on relance le service xen avec `/etc/init.d/xend restart`

4.2 Installation côté client

Pour gérer nos serveurs, nous installons virt-manager avec la commande suivante :


```
1 apt-get install virt-manager
```

Création d'hôtes virtualisée avec virt-manager

1) Pour commencer on démarre virt-manager, puis on lance le gestionnaire de machines virtuelles à partir du menu en cliquant sur l'icône en forme de pc.

3) La fenêtre du gestionnaire de machine virtuelle nous autorise à en créer de nouvelles. On clique sur création de nouvelle machine virtuelle pour faire apparaître l'assistant qui va nous aider pour élaborer notre hôte. L'assistant décompose la création en cinq étapes :
-La localisation et la configuration des supports d'installation -La configuration de la mémoire et les options de CPU -La configuration du stockage de l'invité -La configuration réseau, l'architecture, et d'autres paramètres matériels

Le processus de création d'hôte commence avec la sélection d'un nom et le type d'installation. _____ ON MET UNE IMAGE POUR MONTRER _____ -Local install media(ISO image or CDROM) Cette méthode utilise un CD-ROM,DVD ou une image iso.

-Network install Cette méthode utilise le réseau pour installer le système d'exploitation.

-Import existing disk image Cette méthode nous permet de créer un nouvelle hôte et d'y importer une image disque.

La prochaine étape consiste à configurer l'installation. On configure le type de système d'exploitation et sa version qui sera installé, cela dépend de la méthode d'installation que l'on a choisie. _____

_____ ON MET UNE IMAGE _____
_____ ON
MET UNE IMAGE _____

Configuration du CPU et de la mémoire La prochaine étape consiste à configurer le nombre de CPU et la quantité de mémoire à allouer à la machine virtuelle. L'assistant indique le nombre de processeurs et la quantité de mémoire que l'on peut lui allouer. _____

_____ ON MET UNE IMAGE _____

_____ Configuration de l'espace de stockage _____ ON MET UNE IMAGE _____ Si l'on a choisi d'importer une image de disque existante au cours de la première étape, virt-manager va sauter cette étape. On doit attribuer un espace suffisant pour notre machine virtuelle et toutes les applications que l'hôte a besoin.

Configuration finale On vérifie les paramètres de la machine virtuelle et on clique sur Terminer lorsqu'on est satisfait, cela permettra de créer l'hôte avec les paramètres réseau par défaut, le type de virtualisation, et l'architecture.

Annexe A

Sources

www.grid5000.fr : le wiki disponible sur le site internet de grid5000 fut principale source de renseignements pour le démarrage du projet.

www.loria.fr/lnussbau/ : nous avons pu y consulter des anciens projets sur Grid5000 ce qui nous a permis d'avoir un premier aperçu de ses possibilités.

Annexe B

Scripts

B.1 réservation

```
1 puts "-----"
2 puts "Souhaitez vous reserver des noeuds?(y/n)"
3 loop do
4   test = gets.chomp
5   if test.eql?("n")
6     puts "#####"
7     puts "#sortie du programme#"
8     puts "#####"
9     break;
10  end
11  if test.eql?("y")
12    #Reservation de machines
13    puts "-----Script de reservation-----"
14    puts "Choisir un nombre de noeud:"
15    noeuds = gets.chomp.to_i
16    puts "Choisir un temps de reservation(HH:MM:SS):"
17    temps = gets
18    puts "\nVous avez reserve #{noeuds} noeuds
19    pour une duree de #{temps}"
20    puts "-----"
21    exec "oarsub -I -t deploy -n'virtu' -l
22    slash_22=1+nodes=#{noeuds},walltime=#{temps}"
23    break;
24  end
25 end
```

B.2 déploiement

```
1 #Deployer une image cree
2 puts "Voulez vous deployer une image?(y/n)"
3 loop do
4   test = gets.chomp
5   if test.eql?("n")
6     puts "#####"
7     puts "#sortie du programme#"
8     puts "#####"
9     break;
10  end
11  if test.eql?("y")
12    #choix de l'image
13    puts "-----"
14    puts "image disponibles:"
15    puts `ls /home/\$USER/image | grep .env`
16    puts "-----"
17    puts "Choix de la distribution(tout saisir):"
18    debian = gets.chomp
19    puts "-----"
20    exec"kadeploy3 -f \$OAR_FILE_NODES -a #{debian} -k
21    \$HOME/.ssh/id_rsa.pub"
22    break;
23  end
24 end
25 #Deploiment de l environnement sur les noeuds reserves
26 puts "Voulez vous deployer un environnement?(y/n)"
27 loop do
28   test = gets.chomp
29   if test.eql?("n")
30     puts "#####"
31     puts "#sortie du programme#"
32     puts "#####"
33     break;
34   end
35   if test.eql?("y")
36     #choix de la version a deployer
37     puts "-----"
38     puts "distributions disponibles:"
39     puts `kaenv3 -l | cut -d - -f1 | uniq | tail -n +3`
40     puts "-----"
41     puts "Choix de la distribution:"
42     debian = gets.chomp
43     puts "-----"
44     puts "version de la distribution:"
45     puts `kaenv3 -l | grep #{debian}| cut -d ' ' -f1`
46     debian = debian+"-x64-"
47     puts "-----"
```

```

48     puts "Choix de la version de la distribution a deployee
49     (sans #{debian}):"
50     version = gets.chomp
51     version = debian+version
52     puts version
53     exec"kadeploy3 -e #{version} -f \${OAR_FILE_NODES} -k
54     \${HOME}/.ssh/id_rsa.pub"
55     break;
56 end
57 end

```

B.3 configuration de base

```

1 #####
2 #-----Configuration minimale-----#
3 #-----#
4
5 #recuperation des noeuds reserves
6 cat \${OAR_FILE_NODES} | uniq > \${HOME}/script_base/list_nodes
7 list_nodes="\${HOME}/script_base/list_nodes"
8
9 echo "-----"
10 echo "Liste des machines reservee:"
11 cat \${list_nodes}
12 echo "-----"
13 echo "Copie des clees SSH vers toutes les machines."
14 for node in \$(cat \${list_nodes})
15 do
16     scp \${HOME}/.ssh/id_rsa* root@\${node}:~/.ssh/
17 done
18 echo "-----"
19
20 #mise a jour des noyaux
21 taktuk -l root -f \${list_nodes} broadcast exec [ apt-get update
22 ]
23 taktuk -l root -f \${list_nodes} broadcast exec [ apt-get dist-
24 upgrade
25 -q -y --force-yes ]
26 #changement des mdp root
27 taktuk -l root -f \${list_nodes} broadcast exec [ 'echo -e
28 pttvirtu\npttvirtu" | passwd root' ]

```

B.4 Listing d'une installation de Ganeti

```
1 #!/bin/bash
2 #passage en wheezy
3 rm /etc/apt/sources.list
4 echo "## wheezy security" > /etc/apt/sources.list
5 echo "deb http://security.debian.org/ wheezy/updates main
   contrib non-free" >> /etc/apt/sources.list
6 echo "deb-src http://security.debian.org/ wheezy/updates main
   contrib non-free" >> /etc/apt/sources.list
7 echo " " >> /etc/apt/sources.list
8 echo "#wheezy" >> /etc/apt/sources.list
9 echo "deb http://ftp.fr.debian.org/debian/ wheezy main contrib
   non-free" >> /etc/apt/sources.list
10 echo "deb-src http://ftp.fr.debian.org/debian/ wheezy main
   contrib non-free" >> /etc/apt/sources.list
11 #Installation de ganeti
12 apt-get update
13 apt-get dist-upgrade -y --force-yes
14 apt-get install -y --force-yes ganeti2 ganeti-htools ganeti-
   instance-debootstrap
15 echo "Ajout du node dans /etc/hosts"
16 hostname='cat /etc/hostname'
17 #recuperation des variables
18 #ip du node
19 ifconfig eth0 > troll
20 ipnode='head -2 troll | tail -1 | cut -d\:' -f2 | cut -d' ' -
   f1'
21 echo \$ipnode \$hostname >> /etc/hosts
22 #ip du broadcast
23 ipbroadcast='head -2 troll | tail -1 | cut -d'B' -f2 | cut -d
   \:' -f2 | cut -d' ' -f1'
24 #ip du masque de sous reseau
25 ipmask='head -2 troll | tail -1 | cut -d'M' -f2 | cut -d\:' -
   f2 | cut -d' ' -f1'
26 #ip du reseau
27 ipnetwork='head -1 ipnetwork'
28 #ip de la passerelle
29 ipgateway='head -1 ipgateway'
30 #ajout de cluster1 dans /etc/hosts
31 echo "ajout de cluster1 dans /etc/hosts"
32 ipcluster='cat ipcluster'
33 echo \$ipcluster "cluster1" >> /etc/hosts
34 #Dans /boot/ creer des liens symboliques :
35 ln -s /boot/vmlinuz-2.6.32-5-xen-amd64 /boot/vmlinuz-2.6.xenU
36 ln -s /boot/initrd.img-2.6.32-5-xen-amd64 /boot/initrd.img
   -2.6.xenU
37 #Pour le moment changera surement.
38 echo "creation du LVM"
```

```

39 umount /dev/sda5
40 pvcreate /dev/sda5
41 vgcreate xenvg /dev/sda5
42 #Creation du bridge xen-br0
43 echo " " >> /etc/network/interfaces
44 echo "auto xen-br0" >> /etc/network/interfaces
45 echo "iface xen-br0 inet static" >> /etc/network/interfaces
46 echo "address" \$ipnode >> /etc/network/interfaces
47 echo "netmask " \$ipmask >> /etc/network/interfaces
48 echo "network" \$ipnetwork >> /etc/network/interfaces
49 echo "gateway" \$ipgateway >> /etc/network/interfaces
50 echo "broadcast" \$ipbroadcast>> /etc/network/interfaces
51 echo "bridge_ports eth0" >> /etc/network/interfaces
52 echo "bridge_stp off" >> /etc/network/interfaces
53 echo "bridge_fd 0" >> /etc/network/interfaces
54 #Suppression des ligne de eth0
55 sed -i '9d' /etc/network/interfaces
56 sed -i '9d' /etc/network/interfaces
57 #supression des fichier temporaires
58 rm troll ipcluster ipgateway ipnetwork
59 #initialisation du cluster
60 gnt-cluster init --no-drbd-storage cluster1
61 #ajouter le node
62 gnt-node add \$hostname
63 #et verifier
64 gnt-node list

```

Annexe C

Glossaire