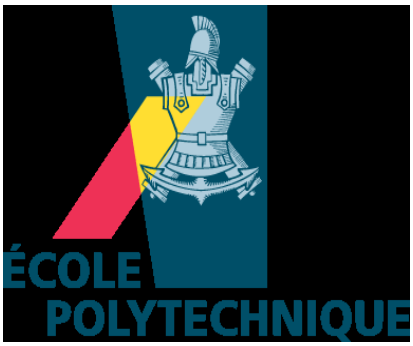
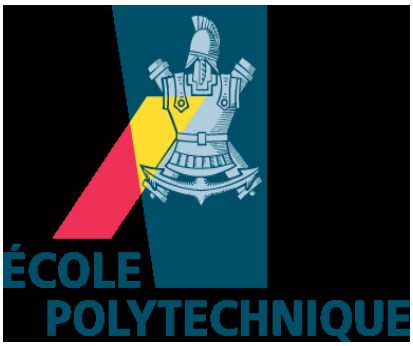


Rapport de stage
Virtualisation et administration d'un groupe de machines

A l'attention de
Monsieur James Régis: maître de stage
Monsieur Thierry Villemur: tuteur pédagogique

Yves Bonelli
Année 2009



Virtualisation et administration des machines du laboratoire du LIX avec les logiciels Ovirt et Func

A l'attention de
Monsieur James Regis: maître de stage
Monsieur Thierry Villemur: tuteur pédagogique

Yves Bonelli
Année 2009

Sommaire

Remerciements

Je tiens tout d'abord à remercier le directeur Philippe Baptiste qui m'a très gentiment accueilli au sein de son laboratoire pendant mon stage. Un grand merci aussi à James Régis, mon maître de stage, qui a su être à l'écoute pendant ces dix semaines. En effet, j'ai eus la chance de travailler dans son bureau et ses conseils relatifs à l'avancement de mon stage m'ont permis de mener à bien mon projet et de travailler efficacement.

Je remercie aussi Mathieu Guionnet, administrateur réseau, qui a su être à mon écoute face aux interrogations que j'avais. Je remercie toute l'équipe du LIX qui à fait preuve de beaucoup d'enthousiasme tout au long de mon stage.

Je remercie également Farid Haddad, Mustapha Khorchid et Romain Paul, trois stagiaires avec qui j'ai partagé de bons moments pendant mon stage et qui ont su m'aider dans certaines situations.

Je souhaite remercier Madame Labate qui m'a donné les informations et le contact relatifs à cette offre de stage. J'ai ainsi pu approfondir mes connaissances sur un sujet qui me tenais à cœur et découvrir le fonctionnement d'un laboratoire.

Enfin, je remercie Thierry Villemur, mon tuteur pédagogique qui a accepté de m'encadrer pendant ce stage.

Résumé

Pendant les dix semaines, mon travail a consisté a administrer des machines virtuelles depuis un serveur grâce à Ovirt et Func

Si ces logiciels s'étaient avérés fiables, ils auraient eut pour but d'aider les administrateurs systèmes du LIX (laboratoire d'informatique de l'Ecole Polytechnique) à centraliser la maintenance des machines hôtes par le biais de Func, et à héberger des serveurs facilement configurables grâce à Ovirt.

J'ai pour cela effectué de nombreux tests sur des machines virtuelles pour configurer au mieux ces logiciels et vérifier le bon fonctionnement de l'architecture du réseau. Cependant, j'ai rencontré de nombreuses difficultés avec le logiciel Ovirt qui est en phase de développement.

Les tests effectués sur Func se sont avérés concluants, c'est pourquoi j'ai créé un script d'automatisation pour l'installer sue les machines très rapidement.

Abstract

For the ten weeks of my training period at……, my job was to manage virtual machines from a server with Ovirt and Func.

If these software had been reliable, they would have helped the system administrators of LIX (laboratory of computing of Ecole Polytechnique) to centralize the maintenance of the machines by means of Func, and to accommodate severs easy to configure thanks to Ovirt..

For this I carried numerous tests on virtual machines to configure as well as possible the software and check the smooth running of the network architecture. However, I met many difficulties with Ovirt which is in its development phase. The tests performed on Func turned out to be quite decisive, that is why I created a script to install it on machines very quickly.

Glossaire

- API: (Application and Programming) Interface:Ensemble de bibliothèques permettant une programmation plus aisée car les fonctions deviennent indépendantes du matériel.

- DHCP: (Dynamic Host Configuration Protocol) Protocole d'attribution dynamique des adresses IP sur un réseau,

- DNS: (Domain Name System) Serveur spécial qui permet de faire correspondre une adresse Web avec une adresse physique comme une adresse IP

- Certificat : Document électronique qui atteste qu'une clé publique est bien liée à une organisation ou à une personne.

- GSSAPI: (Generic Security Service Application Program Interface) API d'authentification sous SASL.

- Kickstart: Script de configuration permettant l'installation d'un système d'exploitation sousLinux (type Redhat, CentOs personnalisé tant au niveau des paquets et logiciels installés qu'au niveau des droits attribués à l'utilisateur de la machine sur laquelle on installe l'OS

- MAC adresse : Adresse spécifique à chaque équipement réseau codée sur 6 octets (48 bits),soit 248 valeurs possibles.

- Minion: Nom donné par le logiciel Func aux machines dites esclaves qui sont gouvernées par une machine maître présente sur le même réseau.

- Paravirtualisation: Virtualisation spécifique qui présente une interface graphique permettant de gérer les machines virtuelles hébergées sur une machine hôte.

- PXE : (Preboot eXecution Environment) Désigne le milieu préalable au boot, grâce auquel on peut installer l'OS sur une machine à chaque démarrage.

- SASL : (Simple Authentification and Security Layer) C'est un procédé d'identification, spécifié par les RFC 2222, 2245 et 2444. Il est employé par les protocoles sous mode connecté.
- Swap: la mémoire virtuelle ou dans un cas particulier la partition utilisée par Linux pour effectuer ses opérations d'échanges

- UUID: Abréviation du terme anglais Universally Unique Identifier (identifiant unique universel). Ce code est un nombre de 128bit qui rend unique la machine.

- VM: (Virtual Machine) Machone virtuelle hébergée sur un ordinateur physique grâce à un logiciel de virtualisation.

- VMM: (Virtual Machine Monitor) Machine virtuelle hébergée par l'ordinateur qui va superviser les autres machines virtuelles, permettant ainsi de vérifier de prendre la main a distance sur chacune d'entre elles.

.

Introduction

La formation DUT Réseau et Télécommunications est une formation à but professionnalisante. Après deux années de formation au sein de l' IUT de Blagnac, j'ai donc l'occasion d'effectuer un stage au sein d'une entreprise ou d'une école, où un projet personnel m'est est confié. Mon stage se déroule au sein de l'école Polytechnique dans le laboratoire d'informatique nommé LIX. Pendant ces dix semaines, je vais ainsi pouvoir connaître le monde du travail, le fonctionnement du laboratoire très différent cependant de celui d'une entreprise et enfin rencontrer d'autres personnes thésards, chercheurs ou encore stagiaires qui travaillent sur des sujets très variés.

Le service dans lequel je travaille est le laboratoire d'informatique appelle LIX (laboratoire d'informatique de l'école polytechnique). Différents centres de recherche (24 exactement) y sont présents comme celui de la bio informatique, l'algorithmique et optimisation, la modélisation combinatoire, la cryptologie, les communications à haute performance, le développement des langages de programmation, l'ingénierie des systèmes industriels complexes et de nombreux autres encore.

Le bureau que j'occupe ne travaille pas dans ces différents groupes de recherche; il a pour but d'optimiser les réseaux présents au sein de ce laboratoire. James Régis, mon maître de stage et Mathieu Guionnet sont les administrateurs systèmes du LIX avec qui je vais partager le bureau.

Pendant dix semaines, je dois administrer des dizaines voire des centaines de machines virtuelles depuis un seul serveur grâce au logiciel Ovirt. Je vais alors effectuer de nombreux tests pour vérifier le bon fonctionnement des protocoles établis entre les machines. Si ces procédures se sont correctement déroulées, je effectuer la migration des ces machines vers un autre serveur et former enfin les personnes qui utiliseront les machines virtuelles mises a disposition de puis chaque ordinateur. Je vais également utiliser Func et exploiter ainsi ce logiciel pour permettre d'effectuer de nombreuses actions comme la mise à jour d'un logiciel, l'aperçu des ressources disponibles sur un groupe de machine et ce par le biais d'une seule commande. Ces deux principaux projets permettront ainsi d'optimiser la gestion des machines réelles et virtuelles au sein du LIX si les tests sont concluants.

Mon rapport aura donc pour but d'introduire tout d'abord le contexte de la virtualisation ainsi que les pré-requis nécessaires à la mise en place d'une telle plate forme. Ainsi je pourrais alors détailler l'intérêt d'utiliser un logiciel de virtualisation comme Ovirt et d'en faire une étude approfondie de celui-ci; Le but sera alors de le déployer au sein du LIX et ainsi d'effectuer la migration des serveurs du laboratoire sur la plate forme Ovirt, si les tests s'avèrent fructueux. Enfin, pour simplifier la gestion (installation d'un logiciel, démarrage d'un ou plusieurs services...) des groupes de machines présentes dans les différentes salles je ferais l'étude du logiciel Func et concevrais ensuite un script d'installation et de configuration complet.

1. Présentation de l’école

1.1 Historique

Plus de deux siècles d'histoire

1794

Création de l'École centrale des travaux publics qui prendra le nom d'École Polytechnique un an plus tard.

1804

Napoléon donne un statut militaire à l'École, et une devise "Pour la patrie, les sciences et la gloire".

1970

L'École devient un établissement public, sous la tutelle du Ministère de la Défense.

1972

Les jeunes filles sont admises à concourir.

1976

L'École est transférée à Palaiseau (au sud de Paris)

1995

Une nouvelle voie du concours est ouverte aux étrangers

2000

La réforme du Cycle Polytechnicien fixe la durée du cursus à 4 ans

1.2 Sa mission

L'École Polytechnique a pour mission de former des hommes et des femmes capables de concevoir et de mener des activités complexes et innovantes au plus haut niveau mondial, en s'appuyant sur une culture à dominante scientifique d'une étendue, d'une profondeur et d'un niveau exceptionnels, ainsi que sur une forte capacité de travail et d'animation.
Fidèle à son histoire et à sa tradition, l'École forme de futurs responsables de haut niveau, à forte culture scientifique, voués à jouer un rôle moteur dans le progrès de la société, par leurs fonctions dans les entreprises, les services de l'État et la recherche.
Notre projet pédagogique est de former des hommes et des femmes de caractère, équilibrés, aptes au travail en équipe, associant à la rigueur, l'écoute des autres et la liberté d'esprit, dotés d'une capacité exceptionnelle d'analyse et de synthèse et capables d'analyser, de concevoir, de construire et de mettre en œuvre des systèmes complexes.
Cette formation repose sur un programme éducatif unique réalisant un équilibre entre:
- un enseignement scientifique, pluridisciplinaire, de très haut niveau;
- une ouverture vers des disciplines littéraires et artistiques et la pratique de langues étrangères;
- une formation éthique, humaine et sportive.
Elle apprend à nos élèves à travailler avec rigueur, dans le respect des faits et l'honnêteté intellectuelle, à maîtriser les technologies actuelles et anticiper celles de demain. Elle leur permet aussi d'acquérir une culture d'une richesse et d'un niveau exceptionnel, tout en développant chez eux le travail en équipe, une ouverture aux problèmes de société et aux attentes de la collectivité et un sens aigu de la responsabilité individuelle.

Elle est mise en œuvre en associant à un corps enseignant de très haut niveau, un centre de recherche internationalement reconnu et un encadrement militaire à qui a été confié l'essentiel de la formation éthique, humaine et sportive.
Le régime de l'internat crée les conditions d'une vie associative et collective intense sur le campus et favorise l'initiative et la créativité de chacun.
Conformément aux valeurs et à la tradition qui sont les siennes, depuis plus de 200 ans, l'École est accessible à tous sans distinction d'origine ou de condition sociale: le seul critère d'admission est la sélection par concours des étudiants les plus aptes à se réaliser dans ce projet.
Pour former des polytechniciens ouverts sur le monde et capables d'exceller dans des environnements multiculturels et multinationaux, l'École accueille un fort contingent d'élèves étrangers et intègre dans son cursus des stages et des formations longues hors de France.
En associant à son cœur de formation, le cycle polytechnicien d'ingénieur, des formations aux normes internationales, masters et thèses, elle se positionne dans l'offre mondiale et valorise sa spécificité.

2. Contexte de la virtualisation

2.1 Machines virtuelles

La virtualisation est l'ensemble des techniques matérielles et/ou logicielles qui permettent de faire fonctionner sur une seule machine plusieurs systèmes d'exploitation et/ou plusieurs applications, séparément les uns des autres, comme s'ils fonctionnaient sur des machines physiques distinctes.

. Cela permet ainsi de disposer de plusieurs ordinateurs virtuels sur une seule machine physique.

Notons que chaque machine virtuelle bien qu'elle ai la possibilité d'utiliser les périphériques de la machine hôte possède un système d'exploitation et un espace de stockage isolé par rapport à la machine hôte et aux autres machines virtuelles. Le logiciel hôte qui fournit cette fonctionnalité est souvent dénommé superviseur ou hyperviseur. Pendant le stage j'ai pu utiliser un logiciel open source intégré sur la plupart des plates-formes Linux. Il s'agit de Virtual Machine Manager.

2.1.1 Fonctionnement

La couche matérielle est la première couche, c'est-à-dire la machine physiquement parlant.

Le système d'exploitation hôte est en relation directe avec les composants matériels de l'ordinateur.

Le réseau virtuel qui relie les machines clientes entre elles ainsi que les machines clientes aux machines hôtes est un réseau purement logiciel interne à la machine hôte, géré par celle-ci.

La mise en place de la virtualisation se déroule en plusieurs étapes successives :

On installe tout d'abord un système d'exploitation qui jouera le rôle de machine hôte ; c'est la machine qui est physiquement reliée à l'ordinateur.

Sur cette machine hôte il suffit d'installer un logiciel (superviseur ou hyperviseur) qui nous permettra d'émuler des machines virtuelles. Dans le cas de notre projet, il s'agira de Virtual Machine Manager.

Une machine virtuelle, parfois aussi appelée machine invitée est une machine qui fonctionne sur la machine hôte. Un logiciel de virtualisation permet de faire fonctionner simultanément un ou plusieurs OS invités. Grâce à l'interface Virtual Machine Manager, on va alors installer les systèmes d'exploitation sur les machines virtuelles. La machine hôte virtualise ou/et émule le matériel pour les OS invités ; ces derniers croient dialoguer directement avec la machine physique qui n'existe pas réellement. Toutes les machines sont donc isolées les une des autres mais utilisent le même support physique nécessaire dès l'installation de l'OS (lecteur CD-ROM partagé par la machine hôte et les machines virtuelles).

2.1.2 Avantages-Inconvénients

- La virtualisation permet une utilisation optimale des ressources d'un parc de machines en fonction de la sollicitation de chacune d'entre elles. En effet la mémoire dédiée

peut ainsi être adaptée en fonction du nombre d'applications exécutées sur une machine ou, de connections établies entre un serveur réalisé sur une machine virtuelle et ses clients. La mémoire vive peut ainsi être mieux gérée et allouée de manière dynamique en fonction des besoins de chaque application à un instant donné.

- Ceci présente notamment une diminution des risques liés au dimensionnement des serveurs lors de la définition de l'architecture d'une application ou d'un serveur (http, ftp, DNS ou autre…), l'ajout de puissance étant alors transparent.
- L'installation et le déploiement sont facilités par l'absence de câblages, de manipulations et d'infrastructures nécessaires à la création d'un réseau.
- La migration facile des machines virtuelles d'une machine physique à une autre en conservant une configuration précise fait également partie des atouts de la virtualisation (grâce à la copie du dossier /etc/…). Cette facilité à copier une configuration permet ainsi un gain de temps important.
- La mutualisation des ressources permet une économie sur le matériel et donc indirectement une diminution de la consommation électrique. L'entretien physique, la simplicité du monitoring et la compatibilité matérielle sont d'autres atouts de la mutualisation des ressources.
- Les phases d'installation, de tests et de développements sur les machines virtuelles ne présentent pas de danger pour la machine hôte mais aussi pour le matériel qui en temps réel peut être détérioré en cas de mauvaise manipulation.
- Les systèmes d'exploitation hôtes sont invisibles pour l'attaquant, ce qui permet la sécurisation et/ou l'isolation d'un réseau.
- Ainsi, différents utilisateurs simultanés d'une même machine peuvent travailler de manière autonome et isolée.

- La virtualisation demande cependant d'importantes ressources. Il est donc nécessaire d'avoir une machine puissante et fiable pour pouvoir héberger un serveur par exemple.

2.2 Virtual Machine Manager

2.2.1 Présentation de Virtual Machine Manager

Ce logiciel propose deux types de virtualisation qui ont des particularités bien distinctes.

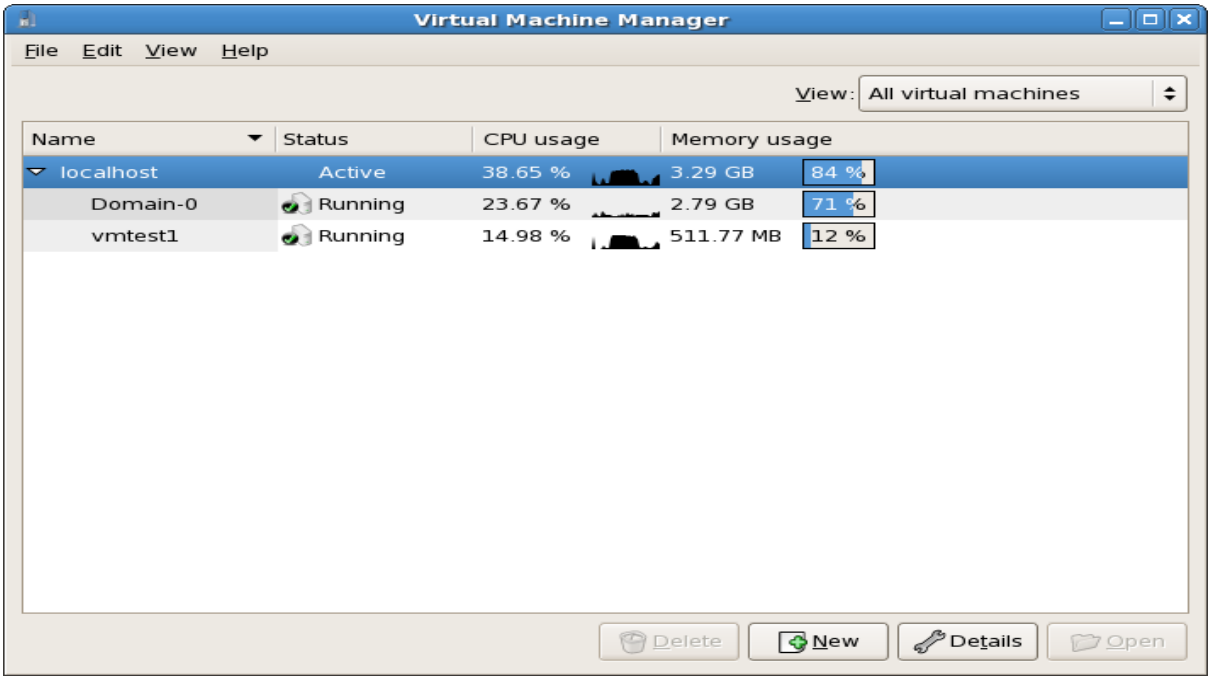
Machine paravirtualised:

Pour l'utilisation de ce type de virtualisation, le système d'exploitation doit bien souvent être modifié pour accepter les contraintes de l'architecture du processeur. La version CentOs, que j'utilise possède un kernel Linux de type 2.6.18-128.el5 qui grâce aux améliorations subies par rapport aux anciennes versions permet la virtualisation de type paravirtualised. Afin d'éviter les modifications au niveau du kernel,les processeurs de la marque AMD et Intel ont été adaptés pour palier à ce problème et améliorer la vitesse d'exécution des process.

Machine fullvirtualised:

Ce mode de virtualisation est très simple cependant, les performances obtenues par la machine virtualisée sont bien moins satisfaisantes que dans le cas de l'autre type de virtualisation. On utilisera ce logiciel directement implémenté par la version CentOs 5.3 utilise le démon Xen. Son interface graphique permet de lancer l'installation, la configuration et la gestion des performances de chaque machine de manière séparée et totalement cloisonnée. Des interfaces réseau peuvent être définies sur chaque machine virtuelle pour leur permettre de communiquer , entre elles et également, mais aussi avec les machines physiques présentes au sein du réseau de l'école.

Aperçu de l'interface graphique:



2.2.2 Installation d'une machine virtuelle

La configuration pour l'installation d'une machine virtuelle se fait en quelques minutes seulement.

En effet, quelques étapes suffisent pour paramétrer une machine fullvirtualised ou paravirtualised.

Tout d'abord il est nécessaire d'avoir installé au préalable les démons Xen et Virtual Machine Manager grâce à la commande suivante:

```
yum install xen ;yum install virt-manager
```

On lance alors virt-manager dans le shell et après avoir sélectionné la machine locale (machine physique) on clique alors sur l'onglet new situé en bas de l'interface graphique. La configuration se fait alors pas à pas.

On sélectionne alors le mode paravirtualised ou fullvirtualised. J'ai personnellement choisi d'utiliser le mode paravirtualised pour pouvoir obtenir de meilleures performances systèmes.

Il faut ensuite indiquer le chemin de l'image présente sur le serveur du LIX et le kickstart qui a été préalablement défini par les administrateurs. J'ai utilisé les OS CentOs et Fedora Core 10 pour des raisons de compatibilité avec les logiciels Ovirt et Func. Notons cependant que CentOs s'installe plus rapidement car le kickstart que l'on utilise comporte une installation plus légère.

On choisit ensuite la partition définie précédemment avec LVM,sur laquelle on installe la machine virtuelle. Il est important de choisir astucieusement la mémoire RAM attribuée pour chaque machine. En effet, une fois que la machine virtuelle est démarrée, toute la taille de la mémoire RAM choisie, lui est alors entièrement allouée et ne peut donc être partagée avec une autre machine.

La machine sur laquelle je travaille possède 4GB de RAM. J'ai donc choisit d'affecter 512 MB à chaque machine virtuelle.
Enfin, la dernière étape permet de choisir le type d'interface que l'on attribue à la machine virtuelle, offrant soit la possibilité d'isoler la machine, soit de la « bridger » avec la machine qui l'héberge.

Une fois ces étapes effectuées, il ne reste alors plus qu'à lancer l'installation. Afin d'optimiser mon temps, j'ai pris l'habitude de créer des machines virtuelles successivement que j'utilise ainsi selon mon besoin.

2.2.3 Avantages-Inconvénients

Virtual Machine Manager a développé une interface graphique bien pensée qui affiche l'état de la mémoire de chaque machine virtuelle et de la machine physique, grâce à des diagrammes qui sont actualisés chaque seconde.

Ce logiciel permet de créer une machine virtuelle en quelques minutes seulement ce qui s'avère très avantageux lorsqu'on effectue des tests sur un logiciel comme Ovirt.

On peut aussi allumer, éteindre, redémarrer, ou encore détruire chaque machine en une seule étape à partir de la fenêtre principale. L'accès aux machines virtuelles se fait par le biais d'une interface graphique. On peut ainsi accéder à chaque machine virtuelle et à toutes ses fonctionnalités depuis la machine physique.

Virtual Machine Manager est donc une solution stable, rapide et efficace et surtout gratuite qui ressemble très fortement à Vmware Server.

Pour fonctionner dans des conditions optimales, la machine physique se doit d'être suffisamment puissante. Seul un processeur récent conçu pour la virtualisation permettra de faire fonctionner ce logiciel appartenant au monde libre ce qui n'est pas nécessaire lorsqu'on utilise Vmware Server.

Enfin, les fonctionnalités réseau sont moins évoluées sur ce logiciel que sur Vmware Server ou encore Virtual Box. Son utilisation sera alors plus adaptée pour l'hébergement de machines « bridgées ».

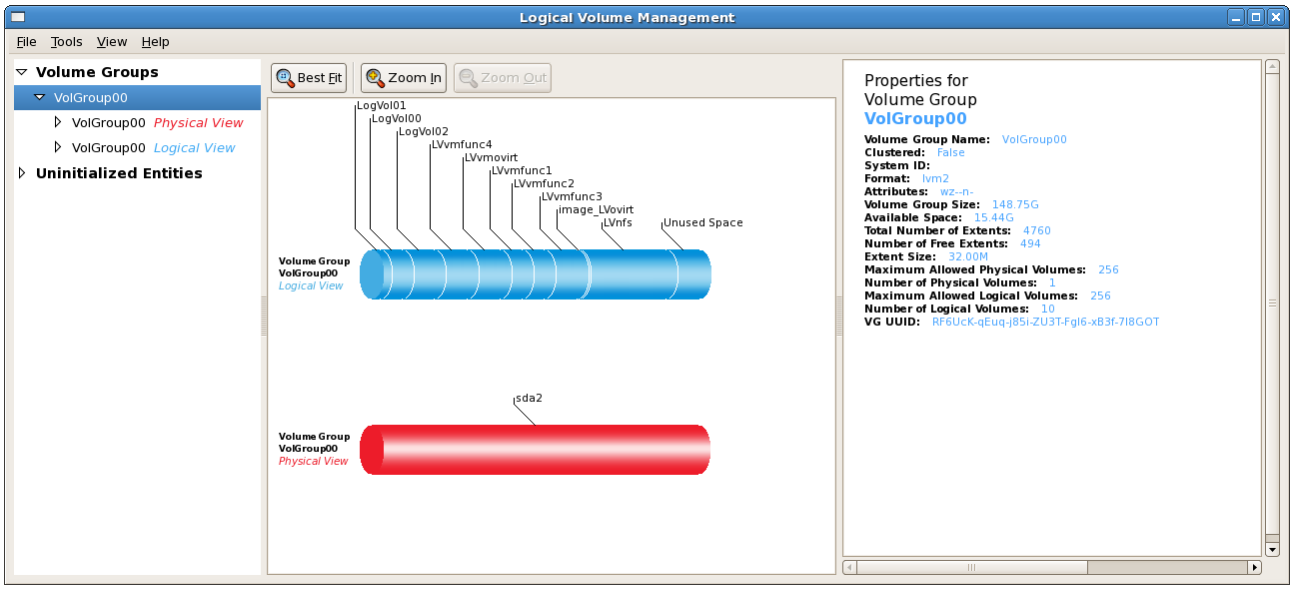
2.3 LVM : Logical Volume Manager

2.3.1 Présentation du logiciel

Tout système de fichiers est habituellement dépendant de contraintes matérielles. Avec un système d'exploitation classique, il est possible de partitionner les disques en partitions primaires et étendues (avec "lecteurs" logiques) mais ce, à condition de rester dans les limites imparties par la taille de ces disques. LVM permet de créer des volumes logiques de très grandes tailles en regroupant plusieurs disques durs et en ne considérant qu'un groupe de volume.

Le Logical Volume Manager est un sous-système pour la gestion du stockage des données sur les disques. En effet, il se glisse entre les périphériques et l'interface d'entrée/sortie du noyau..

Dans ce groupe de volumes, il est possible de créer un ou plusieurs volumes logiques. Ceux-ci sont considérés par le système comme des partitions classiques où il est possible de créer un système de fichiers ou une zone de swap.



Volume physique

Un volume physique ou « PV » pour « physical volume » est un disque ou une partition. Il s'agit donc d'un espace de stockage bien réel, que l'on va exploiter et partitionner avec à LVM. Attention avant l'utilisation de ce logiciel il faut penser à sauvegarder les données car tout ce qui était présent sur la partition sera alors effacé lors de la création du volume physique.

Groupe de volume

Un groupe de volumes ou « VG » pour « volume group » est, comme son nom l'indique, un ensemble de volumes physiques. On a donc un ou plusieurs volumes physiques dans un groupe de volumes, et pour utiliser LVM, il faut obligatoirement au moins un groupe de volumes. Habituellement, sur les gros serveurs, on essaye de regrouper les disques en

fonction de leur caractéristiques (capacités, performances, etc.). Pour un particulier, le fait de mettre plusieurs disques dans un même groupe de volume peut permettre « d'étaler » un système de fichiers sur plusieurs disques, et d'avoir donc /home par exemple qui utiliserait 2 disques.

Volume logique

Un volume logique ou « LV » pour « logical volume » est ce que je vais utiliser au final. Un volume logique est un espace « quelque part dans un groupe de volume » où l'on peut mettre un système de fichiers. C'est donc ce qui remplace les partitions. On peut donc utiliser un volume logique pour créer un espace de swap, ou encore le dossier /home.

Chaque volume group a un répertoire sous /dev, c'est dans ce répertoire que l'on retrouve les fichiers spéciaux des logical volumes. La convention pour nommer ce répertoire est de l'appeler /dev/vgn avec n le numéro du volume group, par défaut si on a installé le système avec LVM, le volume group root a comme répertoire /dev/vg00.

2.3.2 Installation

Cette étape s'effectue très facilement; il suffit de taper la commande suivante dans un shell:

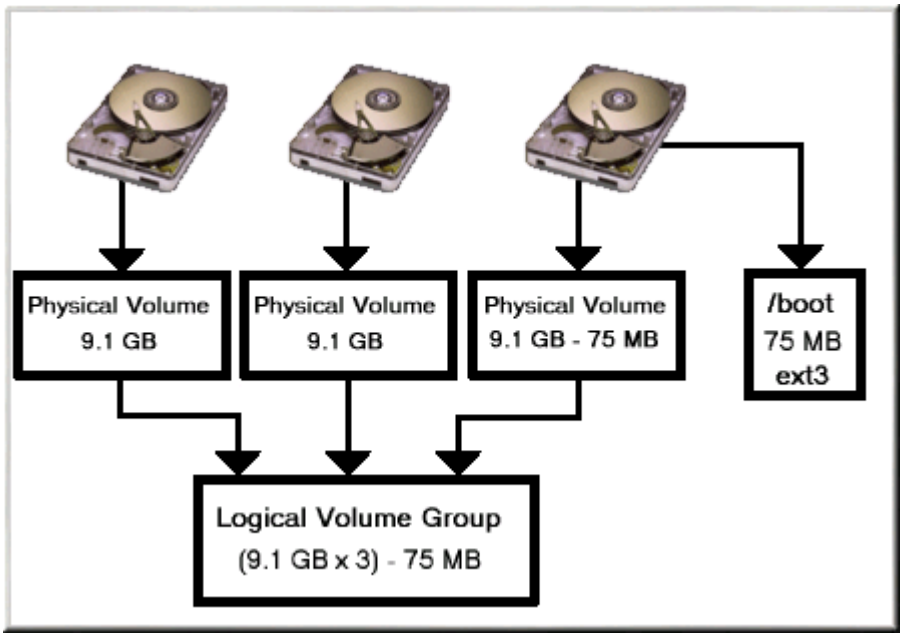
```
yum install system-config-lvm
```

Cependant, dans les configurations standards Linux, ce logiciel est bien souvent installé par défaut.

2.3.3 Avantages Inconvénients

Inconvénient	Avantages
Une fois que celui-ci a « pris le contrôle » d'un disque ou d'une partition, Windows ne pourra plus y accéder (A moins d'installer Explore2FS).	Pas de limitations comme avec les partitions (primaire, étendue, etc.) On ne se préoccupe plus de l'emplacement exact des données On peut conserver quelques giga-octets de libres pour pouvoir les ajouter n'importe où et n'importe quand. Les opérations de redimensionnement deviennent quasiment sans risques.

Exemple de stockage possible:
Les volumes physiques sont associés en groupes de volumes logiques, à l'exception de la partition /boot. La partition /boot ne peut pas se trouver sur un groupe de volumes logiques car le chargeur d'amorçage ne peut pas le lire. Si l'on souhaite que la partition root / se trouve sur un volume logique, on doit créer une partition /boot séparée, qui ne fera pas partie d'un groupe de volumes.
Voici un exemple de configuration possible:



2.4 Connexions SSH

2.4.1 Pourquoi utiliser une connexion SSH?

Secure Shell (SSH) est à la fois un programme informatique et un protocole de communication sécurisé. Ce protocole de connexion impose un échange de clés de chiffrement en début de connexion. Une fois cette étape réalisée, les trames sont alors chiffrées et il devient donc impossible d'analyser la trame avec Wireshark par exemple pour voir ce que fait l'utilisateur. Le protocole SSH a été conçu avec l'objectif de remplacer les différents programmes rlogin, telnet et rsh.

Par mesure de sécurité, j'utilise donc ce protocole pour me connecter à chacune des machines virtuelles sur lesquelles je travaille
J'ai appris a créer une clé SSH de type RSA qui me permet de me connecter à distance, et prendre la main sur une machine (hôte ou virtuelle). J'ai donc pu mettre en pratique les notions sur les connexions SSH que nous avions abordé à l'IUT, et me familiariser davantage avec cet outil que j'utilise maintenant facilement.

2.4.2 Création d'une clé SSH

La commande suivante permet de générer une clé SSH qui me servira à chaque connexion:

```
ssh-keygen -t dsa -f clebonelli
```

On doit alors entrer une suite de caractères (composée à la fois de chiffres et/ou de lettres) nommée passphrase:

```
Generating public/private dsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in bonelli.
Your public key has been saved in bonelli.pub.
The key fingerprint is:
f7:35:e1:eb:c8:73:6a:75:01:51:62:ab:14:0f:55:91 bonelli@lilas
```

Mes clés se trouvent maintenant sous mon home dans le répertoire .ssh (cd ~/.ssh). Il y a 2 fichiers test et test.pub.
test est ma clé privé
test. pub est ma clé public
Au moment de la génération de la clé, je peux la protéger par un mot de passe qui me sera demandé ensuite à chaque fois que j'utilise ma clé.
Partage de la clé publique
Pour pouvoir se connecter sur une autre machine avec cette clé, je dois copier ma clé publique sur le serveur de destination. Cependant par mesure de sécurité et pour éviter toute mauvaise manipulation qui puisse s'avérer dangereuse, James régis a effectué cette étape.

Pour cela:

```
ssh-copy-id -i ~/.ssh/bonelli.pub regis@serveursshdistant
```

On demande alors le mot de passe du login distant. La clé est copiée et configurée pour être utilisée.
Connexion à un hôte distant:

```
ssh bonelli@lilas
```

A chaque connexion, il m'est demandé d'entrer ma passsphrase. Pour éviter cela, il suffit de taper cette commande

```
ssh - add
```

On entre alors son mot de passe qui sera ensuite retenu par la machine.

2.4.3 Configuration:

Pour simplifier les connexions SSH et travailler avec des noms de machine, on modifie /etc/hosts afin d'associer un nom de machine à une adresse ip comme suit

```
192.168.112.143 minion1
```

Pour afficher le nom de la machine sur le prompt,il suffit d'écrire hostname :

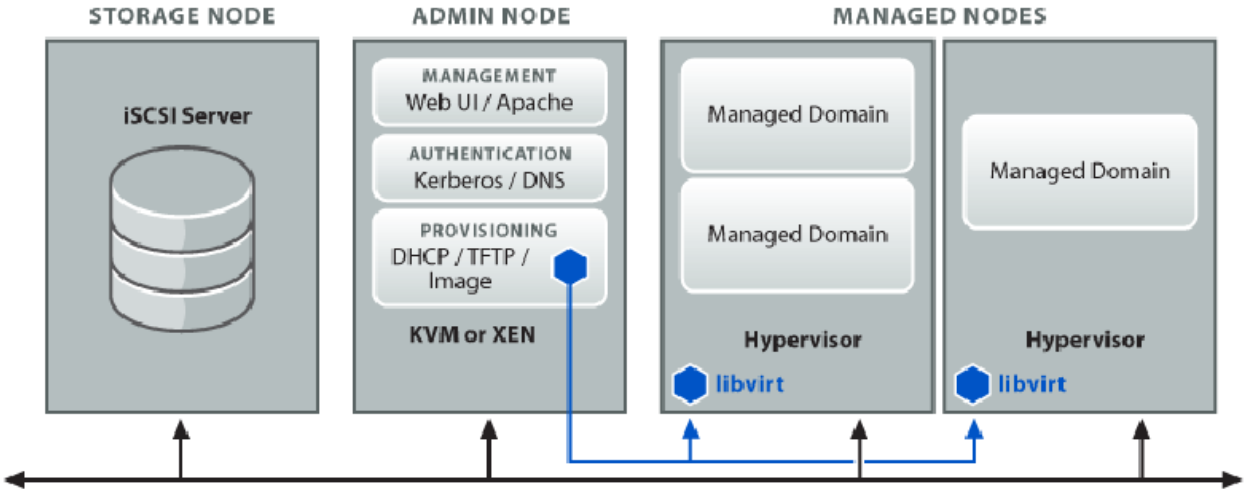
```
hostname minion1
```

Attention le nom de la machine ne pourra être reconnu que sur celle ou le fichier hosts a été modifié. La résolution de nom est donc locale. Ce procédé est donc pratique lorsqu'il y a peu de machines mais fastidieux lorsque celles-ci sont nombreuses. Dans ce cas il sera plus facile de mettre en place un serveur DNS.

3. Ovirt

3.1 Présentation

Qu'est ce qu´Ovirt?



Ovirt est un logiciel open source, multi-plate-formes de virtualisation et de management de système. Ovirt offre à la fois une petite image qui s'installe sur une machine hôte quelconque (mais de préférence puissante)et qui fournit des services de virtualisation semblables à son concurrent Vmware Server. Il possède également et une console de gestion qui vous permettant d'affecter des utilisateurs à des machines, de créer des espaces de stockage adaptés et d'installer et supprimer les machines virtuelles très facilement.

La gestion des ressources (mémoire vive, espace de stockage, périphériques locaux ou périphériques réseaux) peut s'effectuer pour une machine en particulier ou pour un grand groupe de machines. Ovirt est conçu aussi bien pour un petit groupe d'utilisateurs ayant peu de nécessité de contrôle d'accès et de gestion des quotas, que pour des milliers d'hôtes robustes nécessitant un contrôle de groupement, des autorisations et restrictions pour les utilisateurs, et des quotas.

Ovirt peut fonctionner sur un ordinateur équipé de Fedora Core 10 ou d'une version plus récente. L'installation de Ovirt a une taille très faible de 352 MB; il est donc très facile de stocker son image sur un CD-ROM ,une clé USB, ou sur un serveur distant pour lancer l'exécution via le PXE.

Ovirt est basé sur Libivirt, API de virtualisation open source qui fournit des outils pour manipuler des machines virtuelles par le biais d'un canal sécurisé authentifié.

Ovirt intègre notamment Collectd qui permet la collecte de statistiques pour mesurer les performances d'une machine qu'elle soit virtuelle ou réelle ou encore d'un parc de machines.

Grâce à ces deux outils, il est ainsi possible de consulter les configurations, les statistiques ou modifier les paramètres importants.

Une console de gestion, accessible depuis un navigateur internet permet de gérer tous les aspects de la machine hôte et ainsi de la paramétrer de manière plus aisée en utilisant les outils Libivirt et Collectd décrits précédemment.

La migration des serveurs et des machines virtuelles peut s'effectuer depuis une pool (groupe) de machines vers une autre. L'architecture du réseau peut ainsi être modulée en fonction des besoins de l'entreprise.

3.2 Installation

3.2.1 Configuration requise

Pour effectuer l'installation Ovirt, il est nécessaire d'avoir une machine physique ayant un processeur récent ainsi que 2GB de mémoire RAM et l'OS Fedora Core 10 ou d'une version plus récente. On télécharge ensuite les dernières mises à jour du système d'exploitation.
Ce logiciel en phase de test permet d'héberger seulement des machines virtuelles possédant le système d'exploitation Fedora Core 10.

3.2.2 Étapes de l'installation

On installe la base de repos des paquets qui sera nécessaire pour obtenir les paquets spécifiques au serveur Ovirt

```
sudo rpm -ivh http://ovirt.org/repos/ovirt/ovirt-release-LATEST.noarch.rpm
```

On met à jour la base de repos Ovirt afin d'obtenir la dernière version du logiciel.

```
sudo yum update --enablerepo=ovirt
```

On peut maintenant lancer l'exécution des paquets recensés dans la base de repos.

```
sudo yum install --enablerepo=ovirt ovirt-server ovirt-server-installer \ ovirt-node-image ovirt-node-image-pxe
```

On paramètre alors l'environnement local grâce à la commande:

```
ovirt-installer
```

Après avoir lancé ovirt-installer on autorise SELINUX en mode permissive de sorte qu'ovirt puisse utiliser le processus.

```
SELinux application, voulez-vous mettre à permissive? |y|
Setting SELinux permissive Réglage de SELinux permissive
```

On doit également associer une interface au réseau des machines hôtes et une autre qui va être utilisée pour administrer les machines.
Les interfaces présentes sur la machine sont listées comme ci-dessous:

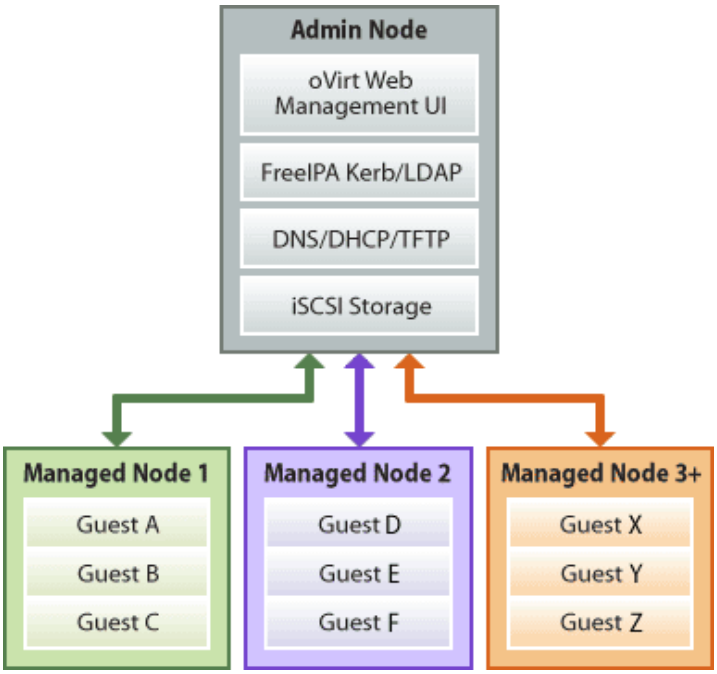
```
Below are the detected networking devices
mac address          interface       ip address
00:00:00:00:00:00:eth1:192.168.50.2
3e:f1:f4:2d:d6:93:virbr0:192.168.100.1
00:1b:77:02:85:25:eth0:192.168.1.197
```

Je conseille d'associer l'interface eth0 pour le Guest Network et l'interface virbr0 (virtual bridge 0) à l'admin network.

```
Enter the interface for the Guest network: |eth0| eth0
Enter the interface for the Admin network: |virbr0| virbr0
```

On indique ensuite le nom de la machine hôte Ovirt:

```
Enter the hostname of the oVirt management server (example:
management.example.com): management.ovirt.priv
```



3.2.3 Configuration du serveur DNS

Ovirt propose l'utilisation du ou des serveurs DNS qu'il a détectés sur le réseau mais pour éviter de destabiliser le réseau du LIX, James m'a demandé de ne pas utiliser cette option. Cette fonctionnalité peut cependant s'avérer très utile lorsque le parc de machines, et, ou de serveurs est relativement important. On peut alors dialoguer en utilisant le nom de machine pour identifier le destinataire et non plus l'adresse IP, ce qui simplifie la tâche pour l'utilisateur.

```
The following DNS servers were found:
nameserver 172.16.52.28
nameserver 10.11.255.27
Use this systems's dns servers? n
```

3.2.4 Configuration du serveur DHCP

On peut notamment utiliser le DHCP du LIX pour attribuer une adresse IP à chaque machine de manière dynamique, mais pour les raisons évoquées précédemment, on utilisera le serveur DHCP implémenté par Ovirt..

```
Does your Admin network already have dhcp? n
Enter the first 3 octets of the dhcp network you wish to use (example:  192.168.50):192.168.50
Enter the dhcp pool start address (example:3):3
Enter the dhcp pool end addess (example:100):50
Enter the dhcp domain you wish to use (example: example.com):|localdomain| ovirt.priv
Enter the network gateway for your Admin network (example:192.168.50.254): 192.168.50.1
Provide pxe/tftp capability? y
```

3.2.5 Configuration de Cobbler

Cobbler permet de créer les images des systèmes d'exploitation que l'on va ensuite installer sur les machines hébergées par ovirt. Cette étape va permettre de configurer les options basiques utiles à l'administrateur du réseau.

```
Do you have a cobbler instance already that you wish to use? n
We will setup a cobbler instance, please provide the following information
Enter your cobbler username:cobbler
Enter your cobbler user password:*******
```

3.2.6 PostgreSQL

Ce programme génère une base de données qui va contenir les donnés et paramètres essentiels tels que les configurations de chacune des machines virtuelles (adresses IP, droits de chaque utilisateur sur cette machine...). Cette Ovirt data base permettra ainsi de connaître de manière synthétique les configurations des différents nœuds du réseau.\

```
Enter a password for the ovirt postgres account: ********
```

3.2.7 Kerberos

Pour éviter à l'utilisateur de partager son mot de passe sur le réseau et ainsi de devenir vulnérable , Ovirt a décidé d'intégrer sur sa plate forme le serveur Kerberos.

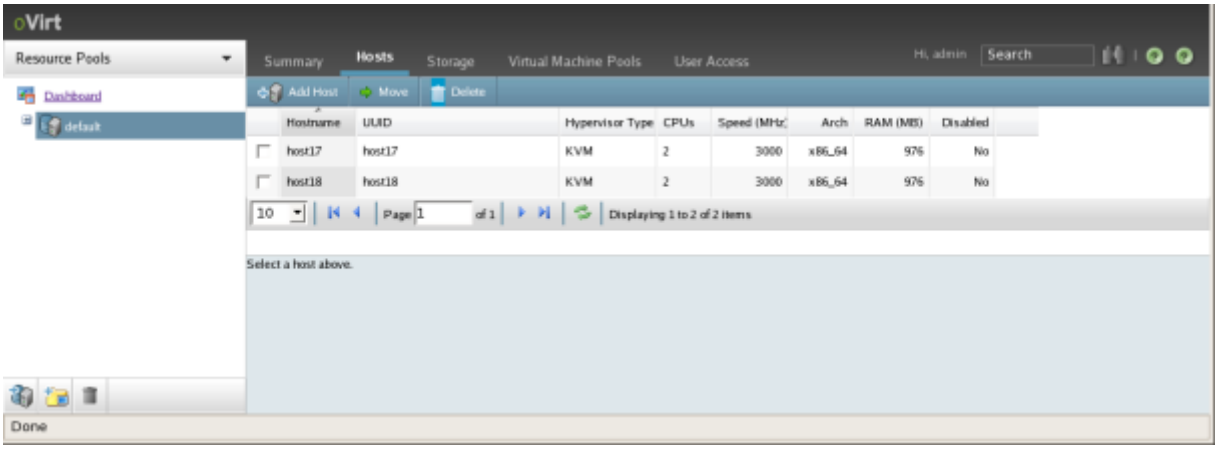
Le protocole Kerberos repose sur un système de cryptographie à base de clés secrètes (clés symétriques ou clés privées), avec l'algorithme DES. Kerberos partage avec chaque client du réseau une clé secrète faisant office de preuve d'identité.
Le principe de fonctionnement de Kerberos repose sur la notion de « tickets » :
Afin d'obtenir l'autorisation d'accès à un service, un utilisateur distant doit envoyer son identifiant au serveur d'authentification.
Le serveur d'authentification vérifie que l'identifiant existe et envoie un ticket initial au client distant, chiffré avec la clé associée au client. Le ticket initial contient :

- une clé de session, faisant office de mot de passe temporaire pour chiffrer les communications suivantes :
- un ticket d'accès au service de délivrement de ticket.

Le client distant déchiffre le ticket initial avec sa clé et obtient ainsi un ticket et une clé de session. Grâce à son ticket et sa clé de session, le client distant peut envoyer une requête chiffrée au service de délivrement de ticket, afin de demander l'accès à un service.

3.3 Utilisation

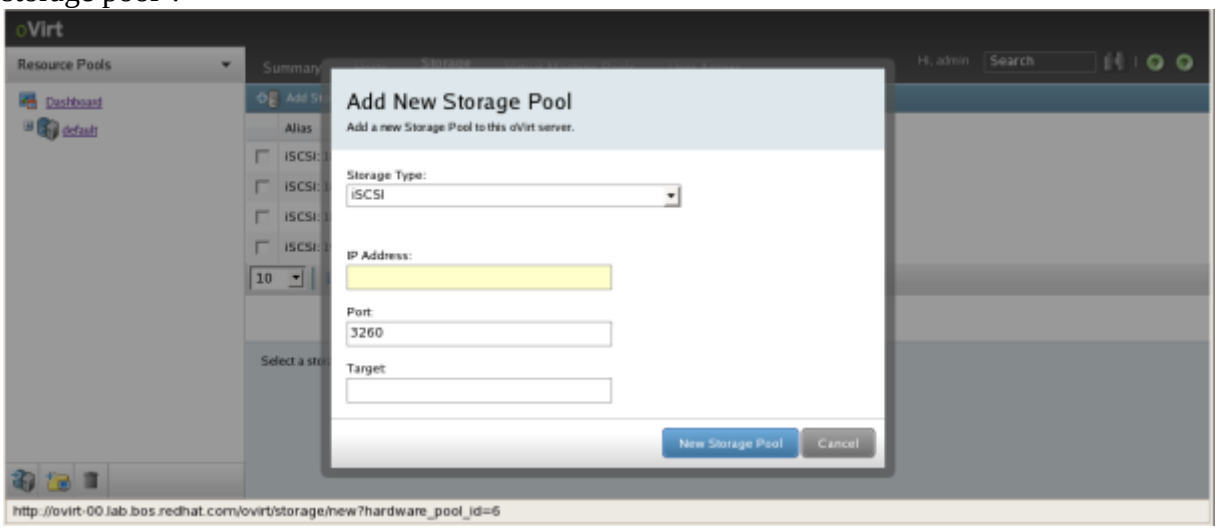
Voici à quoi ressemble la fenêtre principale de la plate forme Ovirt



3.3.1 Création d'un espace de stockage

Afin de pouvoir gérer un groupe de machines, il faut déclarer dans un premier temps un espace de stockage. Pour cela, il est important de choisir entre un serveur de type NFS ou iSCSI selon les besoins de l'entreprise et le matériel dont elle dispose.

Cela se fait en cliquant sur le "Storage" onglet, puis en cliquant sur "Add a new storage pool".



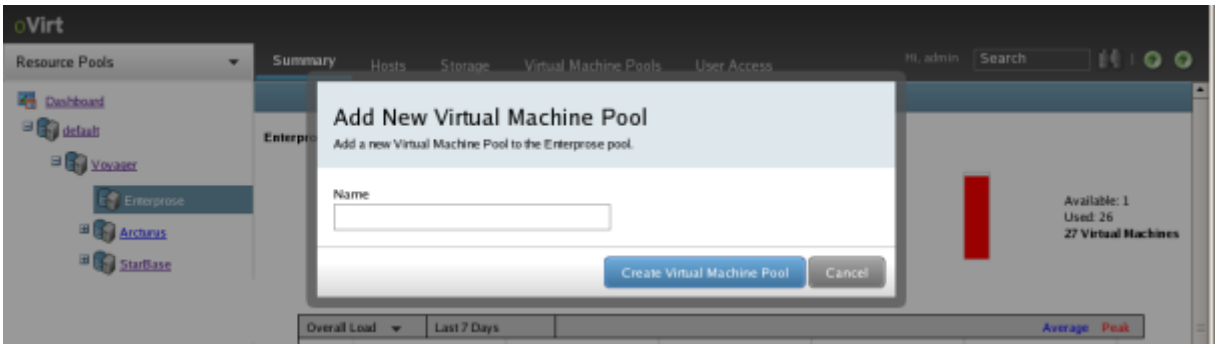
On indique alors le type de stockage (NFS ou iSCSI). L'adresse ip du serveur de stockage et le port de connexion sont indispensables. Ces deux éléments permettent ainsi à Ovirt de se connecter à ce serveur pour gérer l'espace disque des machines virtuelles que l'on va par la suite installer.

Dans le cas d'une configuration iSCSI on utilisera les informations suivantes:
Adresse IP: 192.168.50.2
Port: 3260

Target: ovirtpriv:storage
Dans le cas d'une configuration NFS on utilisera les informations suivantes:
IP address: 192.168.50.2
Export Path: /ovirtnfs
L'installation que j'ai effectuée était celle basée sur un serveur NFS.

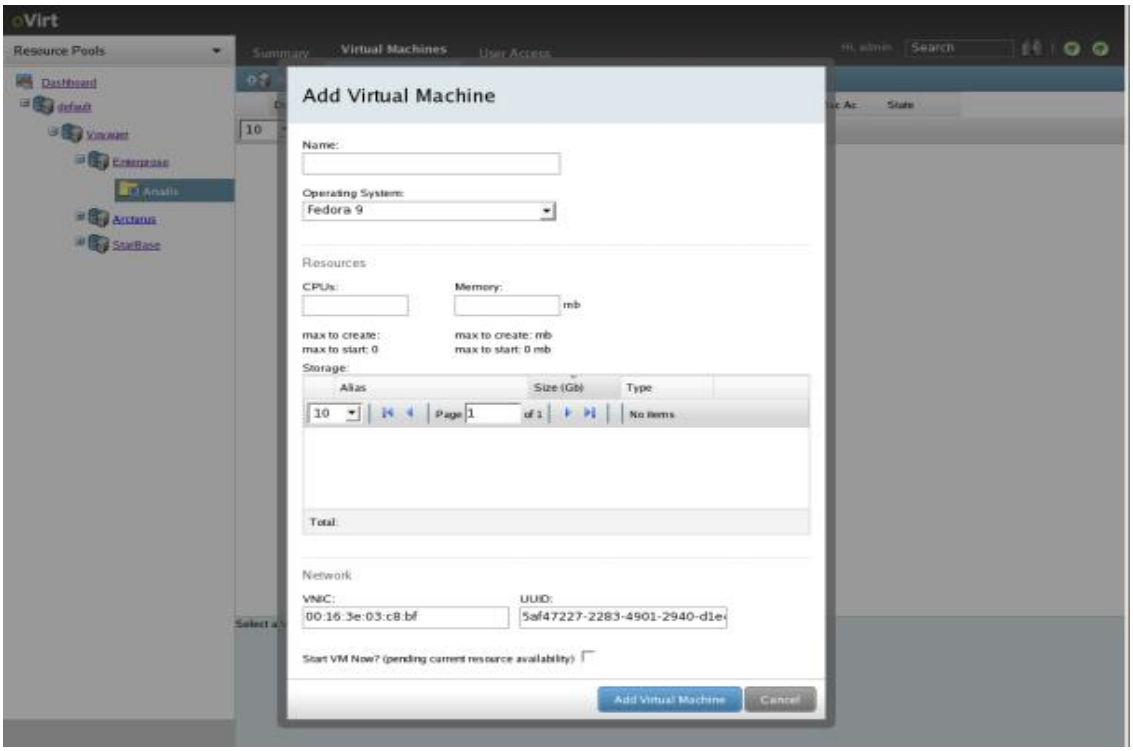
3.3.2 Création d'une pool de machines

Maintenant que le matériel physique a été installé, il est alors indispensable de créer une pool de machines pour permettre de gérer un groupe de machines de manière plus centralisée. Pour créer une pool de machines virtuelles, on clique sur le "Virtual Machine Pool" onglet, puis sur "New Virtual Machine Pool »:



Il est alors nécessaire d'indiquer le nom que l'on souhaite donner à ce groupe. Une fois cette étape réalisée, la pool va alors apparaître sur l'onglet de la fenêtre de droite dans les « Virtual Machine Pools ».

3.3.3Création de machines virtuelles



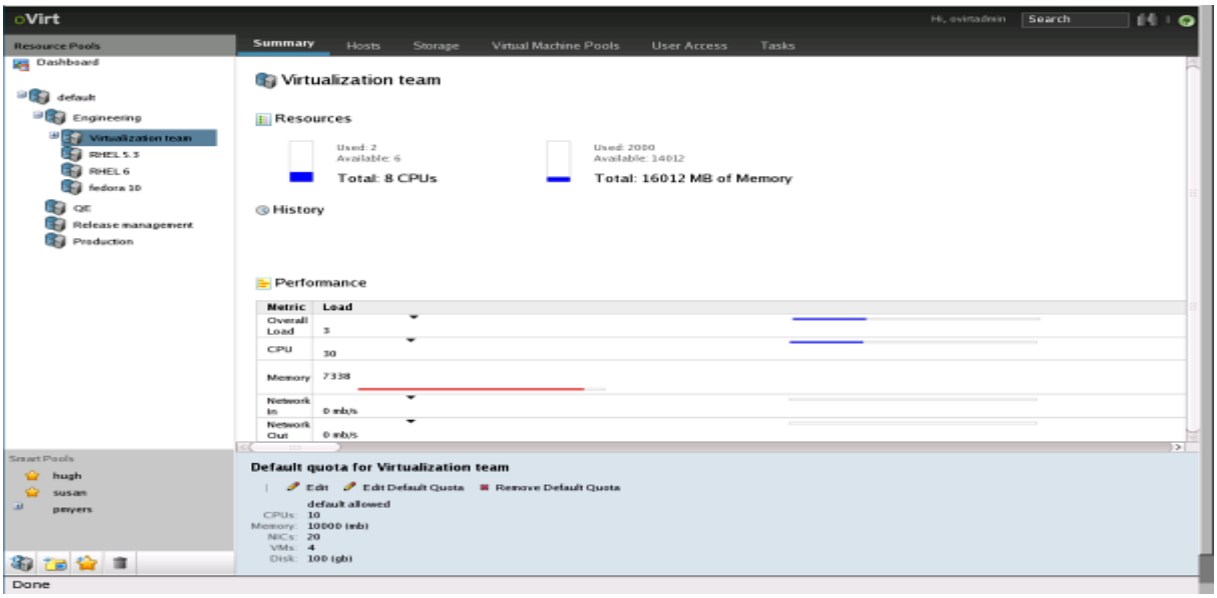
Créons maintenant une machine hôte qui pourra ensuite être intégrée au sein d'une pool de machines. Pour cela, on clique alors sur le signe + à côté de « default » qui va alors indiquée la pool de machines créées précédemment. On clique alors sur ce groupe de machines, puis sur « Add Virtual Machine ». on obtient alors l'écran suivant: On remplit alors les champs nécessaires comme le nom de la machine et le système d'exploitation que l'on souhaite y installer. Notons qu 'Ovirt ne permet d'installer pour l'instant uniquement Fedora Core 9 ou une version ultérieure, ce qui peut s'avérer contraignant pour une personne qui a par exemple l'habitude d'utiliser Windows XP. En effet, sur Vmware Server, il est déjà possible d'installer tout type d'OS. Cependant les évolutions Ovirt permettront peut être de palier à ce problème.

Il faut ensuite attribuer le nombre de processeurs de la machine physique que l'on souhaite attribuer à la machine virtuelle. Il est donc important de faire fonctionner ovirt avec une machine puissante possédant plusieurs processeurs.

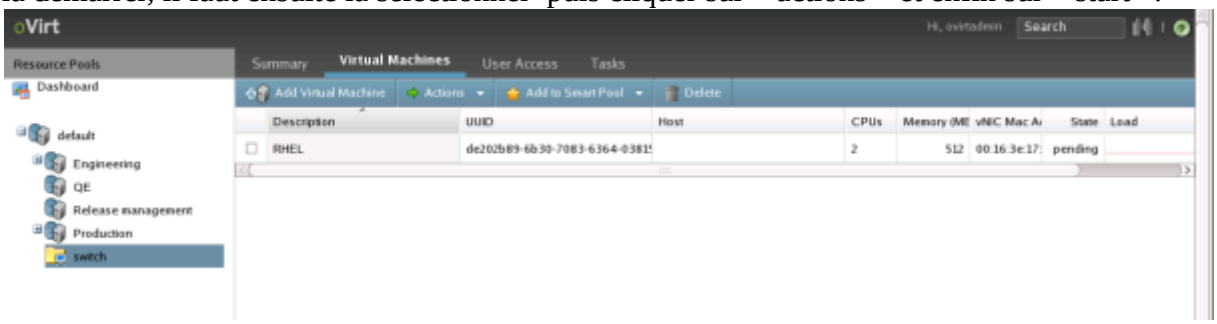
Dans le cadre de mon projet, je possédais une machine avec un bi-processeur (Intel Core 2 Duo cadencé à 3Ghz). J'ai donc été restreint au niveau des tests.

La gestion de la mémoire doit également être adaptée aux possibilités de la machine physique.

Enfin, on peut paramétrer l'interface réseau en lui donnant l'adresse MAC et un UUID que l'on choisit personnellement. Je déconseille cependant ces manipulations, qui peuvent rendre instable le système voire « planter » la machine physique.



Le dernier onglet "Start VM Now" permet dans le cas où on le coche de créer la machine virtuelle que l'on vient de configurer. Cependant celle-ci ne sera pas démarrée. Pour la démarrer, il faut ensuite la sélectionner puis cliquer sur « actions » et enfin sur « start ».



Après de nombreuses tentatives, et des configurations différentes adaptées pour palier aux erreurs rencontrées précédemment, je n'ai cependant pas pu installer l'OS sur cette machine virtuelle. J'ai pour cela effectuer de nombreuses recherches, consulter des forums et fait appel à James Régis mais comme l'indique le site d'Ovirt et le forum des bugs de Redhat, Ovirt connaît encore de nombreux points faibles concernant la stabilité de sa plate forme. J'ai donc été relativement limité à cause de ce problème majeur qui ne m' a pas permis de tester toutes les fonctionnalités de ce logiciel.

3.3.4 Attribuez des droits à un utilisateur

Dans les étapes précédentes, nous avons pu ajouter des machine virtuelles, un espace de stockage, ainsi que des groupes de machines.

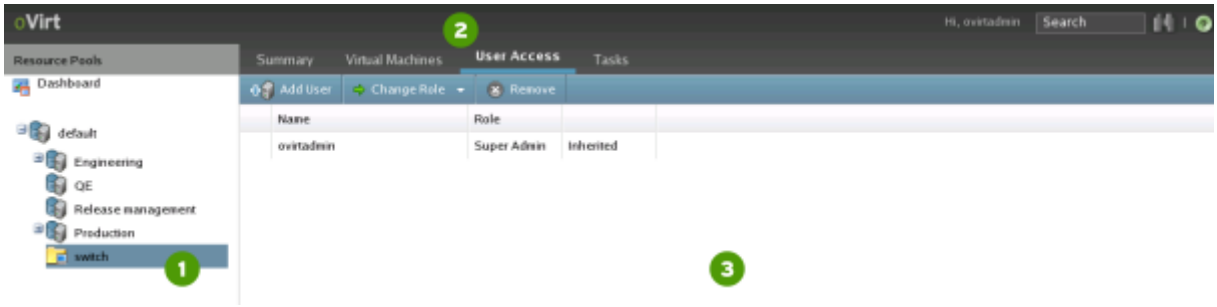
Cette dernière action permet d'attribuer des permissions à des utilisateurs pour un groupe de machines, ou une machine en particulier.

Dans cet exemple, les utilisateurs auront des droits spécifiques sur un groupe de machines. Pour cela:

1) Sélectionnez la pool de machines virtuelles que l'on sélectionne grâce au navigateur situé à gauche de l'écran. La pool de machines virtuelles et ses caractéristiques vont alors s'afficher dans le volet de droite.

2)On clique ensuite sur « User Access »

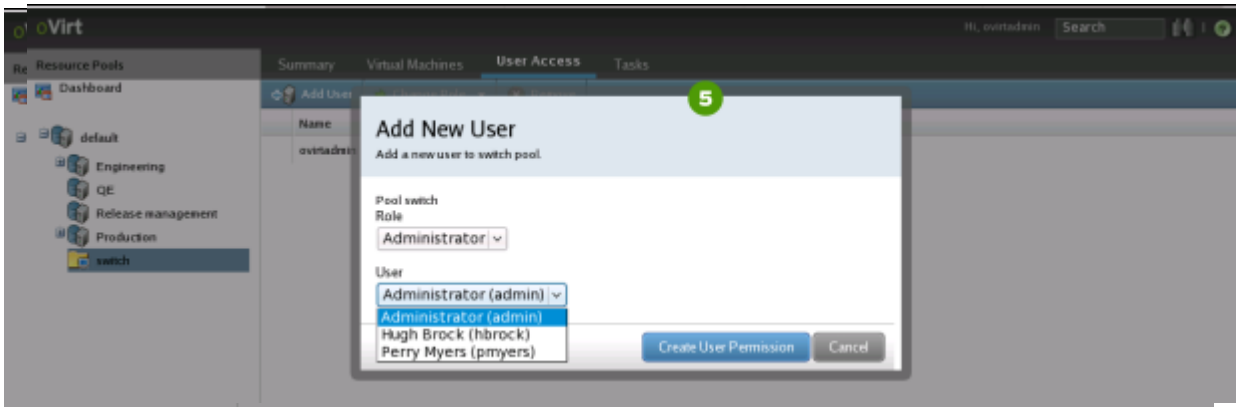
3) La page de configuration apparaît alors dans le volet principal.



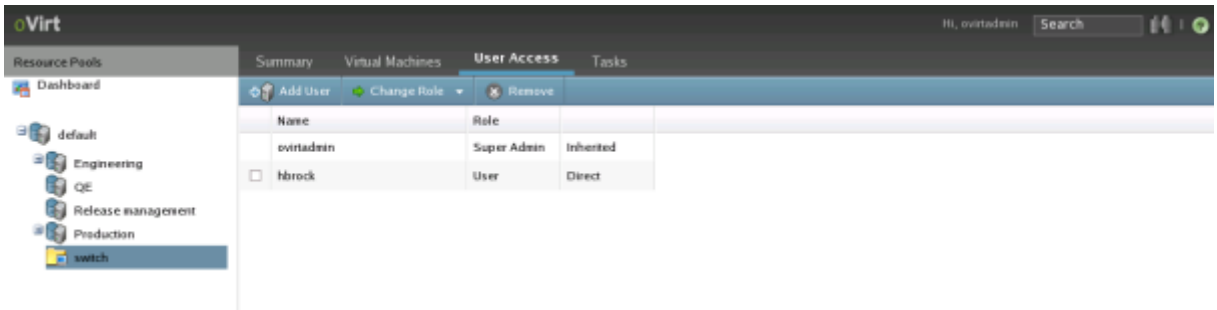
On clique sur « Add New User » dans le menu
On sélectionne ensuite le rôle de l'utilisateur que l'on souhaite attribuer :

- Administrateur
- Super Administrateur
- Utilisateur
- Spectateur

On choisit l'utilisateur situé dans la liste des utilisateurs LDAP.



7) On clique enfinsur « Create User Permission »
Les paramètres utilisateurs sont alors définis et affichés sur la page de la pool de machines.



3.4 Difficultés rencontrées

La découverte de ce logiciel m'a permis d'acquérir une certaine méthodologie tant dans le cadre de ma recherche d'informations mais aussi dans ce lui de la résolution de problèmes. J'ai donc effectué un travail méthodique pour pouvoir exploiter au mieux ce logiciel.

J'ai tout d'abord lu les tutoriels écrits en anglais sur le site officiel de Ovirt détaillant les différentes fonctionnalités d'Ovirt, les étapes de l'installation et son utilisation.
Lors de la première installation, effectuée a la fin de la première semaine, j'ai rencontré de nombreuses difficultés concernant les miroirs permettant d'accéder aux téléchargements des paquets Ovirt. En effet, ils s'avéraient bien souvent inactifs, le paquet n'étant pas situé à l'adresse URL indiquée.

J'ai alors recherché les paquets sur les serveurs Redhat pour trouver le paquet correspondant. Je n'ai pas réussi à trouver la version du paquet utilisée pour l'installation (0.96) mais la version antérieure (0.95). J'ai poursuivi l'installation en y intégrant des paquets de deux versions différentes. Cette étape terminée, j'ai alors pu découvrir le logiciel qui gênerait bien souvent des erreurs affichées sur l'interface (de type 500) sûrement causées par l'incompatibilité des paquets.
J'ai souhaité redémarrer les démons Ovirt après avoir effectué uns mise a jour. L'interface graphique est alors devenue inaccessible.

La deuxième semaine, James Régis m'a alors suggéré d'installer le système d'exploitation CentOs sur ma machine physique et de travailler sur une machine virtuelle avec le l'os Fedora Core 10. J'ai donc eus l'occasion de travailler avec le logiciel LVM, pour créer une partition logique pour chaque machine virtuelle, et Virtual Machine Manager pour gérer les machines virtuelles (mémoire allouée, espace disque réservé...). Les machines virtuelles ont un intérêt majeur, car elles permettent de créer de moduler aisément les performances que l'on souhaite attribuer a chaque machine virtuelle. En créant plusieurs machines on peut alors travailler de manière efficace, écrasant la configuration d'une machine pour la reconstruire, pendant que l'on travaille sur l'autre. J'ai à cinq reprises l'obligation de réinstaller cet agent d'administration.

Le site Ovirt et ses tutoriels ont été mis à jour pendant le weekend , le processus d'installation a été alors très différent. Cependant j'ai rencontré à nouveau de nombreux problèmes dont celui des miroirs inexistants. J'ai été dans l'obligation d'abandonner temporairement l'étude de ce logiciel pour mettre à profit celui de Func.

En effet, Ovirt est en perpétuel développement et présente encore de nombreux bugs qui sont plus ou moins bien détaillés et résolus sur le site de Bugzilla. Je n'ai malheureusement pas réussi à exécuter les fonctionnalités de base comme l'installation d'une machine virtuelle, et la gestion d'une pool de machine stockée sur un serveur NFS. Ce logiciel est relativement instable et une mauvaise manipulation met bien souvent en péril l'installation et la configuration effectuée.

Au cours des cinq semaines, j'ai donc principalement travaillé sur Ovirt en essayant de le configurer pas à pas. De nombreuses améliorations restent cependant, à mettre en œuvre pour pouvoir exploiter correctement Ovirt et le déployer pour la gestion de machines virtuelles et réelles.

4 Func

4.1 Utilisation de Func

Func est un programme python permettant de gérer de manière centralisée des postes clients. Il permet de lancer des commandes, ou voir même des scripts python sans devoir se connecter par SSH. Ceci en fait un outil très pratique si vous avez un réseau à gérer. Il est ainsi possible de réaliser de nombreuses actions comme l'installation d'un programme sur un groupe de machines en effectuant une seule commande.

4.2 Installation de Func

4.2.1 Sur la machine maitre

il faut commencer par installer.

```
yum install func
```

Cette commande va installer le démon Func et le démon Certmaster. Ce dernier utilise des certificats de confidentialité pour permettre au maitre de dialoguer de manière sécurisée avec ses minions.

Il faut ensuite ouvrir les ports 51234 et 51235 (Protocole tcp), l'un utilise par Certmaster et l'autre par Func.

```
iptables -I INPUT -p tcp --dport 22 -j ACCEPT; iptables -I INPUT -p tcp --dport 51235 -j ACCEPT; iptables -I INPUT -m state RELATED,ESTABLISHED -j ACCEPT
```

Il faut alors paramétrer Certmaster pour que celui-ci agisse au niveau 3,4 et 5 de Linux, puis démarrer le service.

```
/sbin/chkconfig --level 345 certmaster on
/sbin/service certmaster start
```

4.2.2 Sur les machines esclaves

il faut commencer par installer:

```
yum install func
```

il faut également ouvrir les ports 51234 et 51235 pour que les minions puissent dialoguer avec la machine.

```
iptables -I INPUT -p tcp --dport 22 -j ACCEPT; iptables -I INPUT -p tcp --dport 51234 -j ACCEPT; iptables -I INPUT -m state RELATED,ESTABLISHED -j ACCEPT
```

On édité ensuite le fichier minion.conf de manière à ce que le minion identifie son maitre également serveur NFS Ovirt dans le cas présent.

/etc/certmaster/minion.conf

```
[main]
certmaster = nfs
log_level = DEBUG
cert_dir = /etc/pki/certmaster
```

Il faut alors paramétrer Certmaster de la même manière que sur la machine maître et démarrer le service.

```
/sbin/chkconfig --level 345 certmaster on
/sbin/service certmaster start
```

Il faut démarrer Func

```
/etc/init.d/funcd start
```

Ajout de l'esclave dans le maître :
il faut que le maître connaisse les esclaves. Pour cela, il va falloir récupérer leur certificat.

```
certmaster-ca –list
```

L'adresse des minions correctement configures va alors apparaître comme dans l'exemple ci-dessous

```
minion1
minion2
```

On doit alors voir apparaître l'adresse des minions. On doit maintenant le faire accepter :

```
certmaster-ca --sign minion1
```

Pour éviter ces manipulations, il est possible de modifier une option qui permettra la signature automatique des certificats. Sur la machine maitre il faut éditer le fichier

/etc/certmaster/certmaster.conf en modifiant l'option de signature automatique.

```
[main]
autosign = yes
listen_addr =
listen_port = 51235
cadir = /etc/pki/certmaster/ca
cert_dir = /etc/pki/certmaster
certroot = /var/lib/certmaster/certmaster/certs
csrroot = /var/lib/certmaster/certmaster/csrs
cert_extension = cert
```

Test du bon fonctionnement de la configuration:
Le test va consister a ouvrir le lecteur de la machine esclave depuis la machine esclave

```
func adresse.esclave.fr call command run "eject /dev/cdrom"
```

4.2.3 Création de groupes de machines

Vous pouvez, si vous le souhaiter configurer des groupes de sous-fifres pour les manipuler plus facilement en ligne de commande. Pour cela créer un fichier /etc/func/groups. Il doit être construit selon le schéma suivant :

```
[groupe_test]
host= minion1; minion2;
```

On peut alors utiliser ce groupe ainsi :

```
func @minion command run "eject /dev/cdrom"
```

Suppression de certificat avec certmaster:

```
certmaster-ca --clean minion1
```

La machine minion1 ne sera alors plus accessible de puis le maitre.

4.2.4 Liste des commandes utiles

Copie d'un fichier

func @minion copyfile -f GOOD --remotepath /home/ok

func minion1 copyfile -f GOOD --remotepath /home/ok

Gérer l'état d'un service

Arrêter un service:
func minion1 call service stop httpd

Démarrer un service:
func minion1 call service start httpd

statut d'un service:
func minion1 call service status httpd
Si le service est arrete fun c retourne:{'minion1': 3}

Si le service est actif func retourne:{'minion1': 0}

Remarque: lorsqu'il n'y a pas d'erreur lors du démarrage d'un service ou de son arrêt Func renvoie:{'minion1': 0}

En cas d'erreur on a alors:{'minion1': 3}

Gestion des paquets RPM

Permet de lister la totalité des paquets rpm installés sur le minion.

func minion1 call rpms inventory

Tests réseaux

Pour effectuer des tests réseaux, il faut tout d'abord installer le logiciel args *magic avec la commande yum install.

Ping depuis la machine distante sur une autre machine

```
 func minion1 call networktest ping www.google.fr -c 2
{'minion1': ['PING www.l.google.com (74.125.77.99) 56(84) bytes of data.',
        '64 bytes from ew-in-f99.google.com (74.125.77.99): icmp_seq=1 ttl=238 time=23.0 ms',
        '64 bytes from ew-in-f99.google.com (74.125.77.99): icmp_seq=2 ttl=238 time=23.0 ms',
        '',
        '--- www.l.google.com ping statistics ---',
        '2 packets transmitted, 2 received, 0% packet loss, time 999ms',
        'rtt min/avg/max/mdev = 23.070/23.076/23.082/0.006 ms',
        '']}
```

Netstat:
Obtenir des informations sur les routes et les autres spécificités du réseau avec la commande netstat

```
func minion1 call networktest netstat
{'minion1': ['Active Internet connections (w/o servers)',
        'Proto Recv-Q Send-Q Local Address           Foreign Address         State      ',
        'tcp        0      0 minion1.localdomain:51234   nfs:55192               ESTABLISHED ',
        'tcp        0      0 minion1.localdomain:ssh     ::ffff:192.168.112.1:55125  ESTABLISHED ',
        'Active UNIX domain sockets (w/o servers)',
        'Proto RefCnt Flags       Type       State         I-Node Path',
        'unix  11     [ ]         DGRAM                    4897   /dev/log',
        'unix  2      [ ]         DGRAM                    1328   @/org/kernel/udev/udevd',
        'unix  2      [ ]         DGRAM                    6046   @/org/freedesktop/hal/udev_event',
        'unix  2      [ ]         DGRAM                    54279  ',
        'unix  2      [ ]         DGRAM                    8773   ',
        'unix  3      [ ]         STREAM     CONNECTED     6371
/var/run/dbus/system_bus_socket',
        'unix  3      [ ]         STREAM     CONNECTED     6370   ',
        'unix  3      [ ]         STREAM     CONNECTED     6365   @/tmp/fam-root-',
        'unix  3      [ ]         STREAM     CONNECTED     6364   ',
        'unix  3      [ ]         STREAM     CONNECTED     6350
/var/run/dbus/system_bus_socket',
        'unix  3      [ ]         STREAM     CONNECTED     6349   ',
        'unix  3      [ ]         STREAM     CONNECTED     6041   @/var/run/hald/dbus-
po4SnPLtsN',
        'unix  3      [ ]         STREAM     CONNECTED     6040   ',
…
```

Traceroute:
Consulter les nœuds du réseau traversés pour dialoguer avec une machine distante;

```
func minion1 call networktest netstat
{'minion1': ['Active Internet connections (w/o servers)',
        'Proto Recv-Q Send-Q Local Address           Foreign Address         State      ',
        'tcp        0      0 minion1.localdomain:51234   nfs:55192               ESTABLISHED ',
        'tcp        0      0 minion1.localdomain:ssh     ::ffff:192.168.112.1:55125  ESTABLISHED ',
```

```
        'Active UNIX domain sockets (w/o servers)',
        'Proto RefCnt Flags       Type       State         I-Node Path',
        'unix  11     [ ]         DGRAM                    4897   /dev/log',
        'unix  2      [ ]         DGRAM                    1328   @/org/kernel/udev/udevd',
        'unix  2      [ ]         DGRAM                    6046   @/org/freedesktop/hal/udev_event',
        'unix  2      [ ]         DGRAM                    54279  ',
        'unix  2      [ ]         DGRAM                    8773   ',
        'unix      3              [ ]                      STREAM          CONNECTED          6371
/var/run/dbus/system_bus_socket',
        'unix  3      [ ]         STREAM     CONNECTED     6370   ',
        'unix  3      [ ]         STREAM     CONNECTED     6365   @/tmp/fam-root-',
        'unix  3      [ ]         STREAM     CONNECTED     6364   ',
        'unix      3              [ ]                      STREAM          CONNECTED          6350
/var/run/dbus/system_bus_socket',
        'unix  3      [ ]         STREAM     CONNECTED     6349   ',
        "]}
```

Dig :

Obtenir des informations relatives au serveur consulté

```
 func minion1 call networktest dig redhat.com
{'minion1': ['',
        '; <<>> DiG 9.3.4-P1 <<>> redhat.com',
        ';; global options:  printcmd',
        ';; Got answer:',
        ';; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 35441',
        ';; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 3',
        '',
        ';; QUESTION SECTION:',
        ';redhat.com.\t\t\tIN\tA',
        '',
        ';; ANSWER SECTION:',
        'redhat.com.\t\t60\tIN\tA\t209.132.177.50',
        '',
        ';; AUTHORITY SECTION:',
        'redhat.com.\t\t600\tIN\tNS\tns2.redhat.com.',
        'redhat.com.\t\t600\tIN\tNS\tns3.redhat.com.',
        'redhat.com.\t\t600\tIN\tNS\tns1.redhat.com.',
        '',
        ';; ADDITIONAL SECTION:',
        'ns1.redhat.com.\t\t81525\tIN\tA\t66.187.233.210',
        'ns2.redhat.com.\t\t81525\tIN\tA\t209.132.183.2',
        'ns3.redhat.com.\t\t81525\tIN\tA\t66.187.229.10',
        '',
        ';; Query time: 120 msec',
        ';; SERVER: 192.168.112.1#53(192.168.112.1)',
        ';; WHEN: Tue May  5 14:23:27 2009',
        ';; MSG SIZE  rcvd: 146',
        '',
        "]}
```

Isportopen:

Cette commande est très utile lorsque l'on veut installer une application nécessitant l'ouverture d'un ou plusieurs ports spécifiques à son application. On peut alors vérifier si les ports de la machine sont ouverts ou non.

```
func minion2  call networktest isportopen minion2 22
{'minion2': [0, 'connection to minion2:22 succeeded']}
```

Dans le cas de la machine minion2 le port 22 est effectivement ouvert. L'exécution des commandes SSH pourra donc se dérouler correctement.

4.3 Création d'un script d'automatisation de l'installation

Ce script a pour but de reprendre les étapes de l'installation que je vous ai présenté précédemment et de les automatiser pour permettre un gain de temps et une plus grande simplicité.

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

import os
import commands
import subprocess

def main():

    #----------------------------MODULE DE CHOIX DE L'INSTALLATION DU LOGICIEL FUNC-----------------------------------------
    print "================================================================================================="
    print "****************Bienvenue dans l'assistant d'installation du logiciel FUNC*****************************"
    print "================================================================================================="

    #Choix du Module

    print "1-INSTALLATION DE FUNC"

    print "2-SUPPRESSION DE FUNC"


    num = raw_input("Entrez le numéro de l'action correspondante souhaitée: ")

    if num=='1':

        #Installation de func


        print "1-INSTALLATION DU SERVEUR FUNC"

        print "2-INSTALLER FUNC SUR UN MINION EN ENTRANT SON ADRESSE IP"

        print "3-INSTALLER FUNC SUR LES MINIONS DONT L'ADRESSE EST PRESENTE DANS LE FICHIER FICIP"
```

```
        num = raw_input("Entrez le numéro de l'action correspondante souhaitée: ")



        if num=='1':
          installation_serv()

       # elif num=='2':
        #   digit=raw_input("Entrer le dernier octet de l'adresse IP du minion sur lequel vous
voulez installer Func")


        elif num=='3':
          installation_group_minion()

    elif num=='2':
        #Suppression de Func
        print "1-SUPPRESSION DU SERVEUR FUNC"

        print "2-SUPPRIMER FUNC D'UN MINION EN ENTRANT SON ADRESSE IP"

        print "3-SUPPRIMER FUNC DES MINIONS DONT L'ADRESSE EST PRESENTE
DANS LE FICHIER FICIP"

        num = raw_input("Entrez le numéro de l'action correspondante souhaitée: ")

        if num=='1':
           suppression_serv()

       # elif num=='2':
        #     digit=raw_input("Entrer le dernier octet de l'adresse IP du minion pour lequel vous
voulez désinstaller Func")

        elif num=='3':
            suppression_group_minion()


def installation_serv():
    #----------------------------MODULE D'INSTALLATION DE FUNC----------------------------
-----------------------------------
    print
"============================================================================
================================================="
    print "********************** INSTALLATION DU SERVEUR
FUNC************************************************************"
    print
"============================================================================
================================================="
    #Installation et configuration de func et certmaster sur la machine serveur
```

```
    commands.getstatusoutput("su -c 'rpm -Uvh
http://download.fedora.redhat.com/pub/epel/5/i386/epel-release-5-3.noarch.rpm'")
    commands.getstatusoutput("yum -y install func")

    #Configuration du firewall du serveur
    commands.getstatusoutput("iptables -I INPUT -p tcp --dport 22 -j ACCEPT; iptables -I
INPUT -p tcp --dport 51235 -j ACCEPT; iptables -I INPUT -m state
RELATED,ESTABLISHED -j ACCEPT")


    commands.getstatusoutput("/etc/init.d/funcd start")
    commands.getstatusoutput("/sbin/chkconfig --level 345 certmaster on")
    commands.getstatusoutput("/sbin/service certmaster start")

def installation_group_minion():
    print
"=============================================================================
======================================="
    print "********************** INSTALLATION DES MINIONS
*********************************************************"
    print
"=============================================================================
======================================="
    #Extraction du dernier octet de l'@ IP du minion indiquée dans ficip.txt
    print
"=============================================================================
======================================="
    print "********************** INSTALLATION DU MINION $i
*********************************************************"
    print
"=============================================================================
======================================="

    fs = open("ficip.txt", 'r')
    for line in fs.readlines():
        digit = string.splitfields(line, '.')
        print digit[3]

        # liste des repos nécessaires au téléchargement de func et certmaster
        commands.getstatusoutput("ssh -l root %s su -c 'rpm -Uvh
http://download.fedora.redhat.com/pub/epel/5/i386/epel-release-5-3.noarch.rpm'") %(digit[3])

        # Installation de func et de certmaster, paquet contenu dans func
        commands.getstatusoutput("ssh -l root $i yum -y  install func")
        commands.getstatusoutput(" ssh -l root $i /etc/init.d/funcd start")

        # Copie du fichier de configuration du minion
        commands.getstatusoutput(" scp /func/minion.conf $i:/etc/certmaster/")

        # Copie du fichier de configuration /etc/hosts
```

```
        commands.getstatusoutput("cp /func/hosts_type /func/hosts")

        #Modification du fichier /etc/hosts de la machine hôte
        commands.getstatusoutput("echo $i|gawk -F. '{print $4}'>/func/digit_list")

        #Enregistrement des digits dans le fichier digit_list
        #commands.getstatusoutput("for digit in `less digit_list`;do")
        print "La machine distante porte le nom suivant: minion%s" %(digit[3])
        commands.getstatusoutput(" done")

        print "%s minion%s >> /func/hosts;" %(digit[3], digit[3])
        commands.getstatusoutput("scp /func/hosts %s:/etc/;"%(digit[3]))


        #Suppression du fichier temporaire hosts
        commands.getstatusoutput("rm -f /func/hosts;")

        #Configuration du firewall du minion
        commands.getstatusoutput("ssh -l root %s 'iptables -I INPUT -p tcp --dport 22 -j
ACCEPT; iptables -I INPUT -p tcp --dport 51234 -j ACCEPT; iptables -I INPUT -m state
RELATED,ESTABLISHED -j ACCEPT'")%(digit[3])

        #Attribution du hostname associé à la machine
        commands.getstatusoutput("host_name='minion%s'")%(digit[3])
        commands.getstatusoutput("ssh -l root %s hostname %s")%(digit[3],host_name)

        # Configuration et démarrage des programmes
        commands.getstatusoutput("ssh -l root %s \"/sbin/chkconfig --level 345 funcd
on\""%(digit[3]))
        commands.getstatusoutput("ssh -l root %s \"/etc/init.d/funcd start; /etc/init.d/certmaster
start\""%digit[3])

        print "********************** Statut des daemons de la machine $i
****************"
        # Vérification de l´état du daemon certmaster et funcd
        commands.getstatusoutput("ssh -l root %s \"/etc/init.d/funcd status; /etc/init.d/certmaster
status\""%digit[3])
        commands.getstatusoutput("sleep 6")

        #signature du certifcat du minion
        commands.getstatusoutput("certmaster-ca -s %s"%host_name)


        #Liste des minions ayant obtenu leur certificat
        print "*****Liste des minions ayant obtenu leur certificat************"
        commands.getstatusoutput("certmaster-ca --list-signed")

        # Fin du script lorsque celui-ci s'est correctement déroulé
        commands.getstatusoutput("exit 0  # sortie normale (code de retour 0 )")
```

```
def suppression_serv ():

   #---------------MODULE DE SUPPRESSION DE FUNC-----------------------------------------
-----
   print
"================================================================================
================================================="
   print "********************** SUPPRESSION DU SERVEUR
FUNC************************************************************************"
   print
"================================================================================
================================================="

   # Suppression de func et certmaster sur  le serveur
   commands.getstatusoutput("yum -y remove certmaster")
   commands.getstatusoutput("yum -y remove func")
   commands.getstatusoutput("rm -rf /etc/pki/certmaster")

def suppression_group_minion ():

   print
"================================================================================
======================================="
   print "********************** SUPPRESSION DES MINIONS
*****************************************************"
   print
"================================================================================
======================================="

   print
"================================================================================
======================================="
   print "********************** SUPPRESSION DU MINION $i
*****************************************************"
   print
"================================================================================
======================================="
   #exécution de la liste de commandes pour chaque machine dont l'@ IP est indiquée dans
ficip.txt

   fs = open("ficip.txt", 'r')
   for line in fs.readlines():
      digit = string.splitfields(line, '.')
      print digit[3]

   # Suppression des daemons certmaster et func
   commands.getstatusoutput("ssh -l root %s \"yum -y remove certmaster; yum -y remove
func;rm -rf /etc/pki/certmaster \""%digit[3])
```

```
    #Suppression des certificats depuis le serveur Func
    commands.getstatusoutput("echo %s|gawk -F. '{print $4}'>/func/digit_list"%digit[3])

    #Enregistrement des digits dans le fichier digit_list

      #print "Suppression du certificat de la machine: minion"digit[3]
    print "Suppression du certificat de la machine: minion%s" %digit[3]


if __name__ == "__main__":
    main()
```

5 Retour d’expérience

Ce stage de dix semaines au cœur du LIX m'a beaucoup apporté tant sur le plan technique que relationnel. J'ai travaillé dans le bureau de Mathieu Guionnet et de James Régis, administrateurs systèmes du laboratoire, ce qui m'a permis au quotidien de comprendre plus précisément quelles étaient les tâches qui leur étaient confiées et quels types de problèmes ils devaient résoudre.

Le projet qui m’a été confié m ‘a beaucoup apporté. Ayant commencé mon stage avec le logiciel Ovirt, je me suis heurté à de nombreuses difficultés. J’ai donc appris à chercher efficacement des informations sur Internet mais aussi dans des livres qui étaient à ma disposition.

De plus, les nombreuses erreurs auxquelles je me suis heurté m’ont demandé une certaine rigueur pour pouvoir résoudre les problèmes. J’ai notamment appris à travailler en autonomie et à prendre des initiatives. J’ai enfin utilisé de nombreux outils intégrés à Linux comme l’éditeur Vim et j’ai acquis une certaine logique, ce pourquoi j’utilise maintenant plus facilement.

Parallèlement à mon projet de stage, j'ai eus l'occasion de voir la salle des serveurs du LIX et d'autres laboratoires. James m'a également expliqué la topologie du réseau depuis notre laboratoire jusqu'au réseau RENATER, en me détaillant la fonction de chaque nœud, et son adresse qui lui a été attribuée. Lors du remplacement d'un disque dur monté en raid sur un serveur, James m'a expliqué les différentes étapes de la réinstallation et le fonctionnement de la baie du serveur. Toutes ces découvertes m’ont beaucoup plu et m’ont donné envie de poursuivre ma voie dans le domaine du réseau

J'ai eus la chance d'échanger et de partager de bons moments avec les stagiaires présents au sein du LIX qui avaient chacun un sujet de stage très différent. Ainsi, j’ai pu m’intéresser davantage aux innovations électroniques qu’ils étudiaient. Je garde un très bon souvenir de ce stage au cours duquel j’ai connu de très bon moments..