SWILABUS

Laboratoire d'administration système

Partie « Linux »

par Louis SWINNEN

Septembre 2015

1. Introduction

Dans cette partie du cours, nous allons aborder des notions importantes en administration système sous GNU/Linux. Pour les aspects pratiques, nous utiliserons le logiciel VMware Workstation[1] ainsi que le système d'exploitation GNU/Linux CentOS 7[2].

Suite logique des cours de *systèmes d'exploitation et réseaux*, des notions importantes dans ces domaines sont requises pour pouvoir aborder le cours d'administration système sereinement.

Dans ce cours, nous aborderons les bases du système (gestion des utilisateurs, administration du disque dur, configuration réseau, scripting PERL) ainsi que les principales tâches dédiées aux serveurs Linux : serveur Web (hébergement de sites web PHP avec MySQL), serveur FTP, serveur DNS, accès à distance, firewall et sécurisation du système.

1.1 Pourquoi CentOS 7 ?

Le choix d'une distribution Linux est, avant tout, *une question de goût*. Il existe de nombreuses distributions : certaines très à la mode comme Ubuntu, d'autres commerciales comme RedHat, d'autres totalement libres et un peu moins accessibles comme Debian. Le choix de CentOS est, avant-tout, un choix de raison : cette distribution est **gratuite**, orientée **serveur** et **est compatible avec les versions RedHat® Enterprise Linux**.

Le choix s'impose donc assez aisément puisque cette distribution est très proche d'une version commerciale, disponible avec support et assez répandue dans le milieu professionnel.

Enfin, passer d'une distribution à une autre demande un peu de travail même si les bases restent les mêmes. Ainsi chaque distribution propose son système de *packages* (distribution de logiciels compilés), ses fichiers de configuration et sa sélection de programmes. En effet, sous Linux, il n'est pas rare de trouver beaucoup de programmes répondant aux mêmes attentes (par exemple le serveur FTP : il y a *ftpd*, *proftpd*, *wu-ftp* ou encore *vsftpd*). La distribution propose, en standard, une version du service avec lequel la configuration est proposée.

1.2 Cours, Notes, ECTS

Le cours d'*administration système et réseau* est fractionné en 2 parties distinctes pour le parcours *standard* : la première reprenant les notions d'administration sous Linux et la seconde reprenant les notions d'administration sous Windows Server.

Pour les étudiants inscrits au parcours Salto, le cours est limité aux notions d'administration sous Linux.

1.2.1 Le parcours standard

Le cours fait **70h** et **6** crédits ECTS. La partie *administration système Linux* reprendra :

[1] Une version plus limitée mais gratuite, appelée VMware View est disponible sur le site de VMware
[2] A l'instar de nombreuses distributions Linux, CentOS GNU/Linux est disponible gratuitement sur internet.

- Un cours de **35h** (uniquement au laboratoire)
- Une note comptant **pour la moitié des points** dans le cours complet.
    - Administration Système
        - Partie Windows, pondération 50 %
        - Partie Linux, pondération 50 %
- La proportion note année, note examen est la suivante : **pas de note année** + **100 % note examen**.

1.2.2 Le parcours Salto

Le cours fait **35h** et **3** crédits ECTS. La partie *administration système Linux* reprendra :

- Un cours de **35h** (uniquement au laboratoire)
- Administration Système
    - Partie Linux, pondération 100 %
- La proportion note année, note examen est la suivante : **pas de note année** + **100 % note examen**.

Il convient de se référer à la **fiche UE officielle** du cours pour ce point précis.

1.3 Examen

L'examen d'administration système, partie Linux a lieu, en **1ère session**, en **janvier**. La seconde session est programmée en août.

Il s'agit d'un **examen *pratique* à cours ouvert** dans lequel une configuration particulière vous sera demandée et vous disposerez d'un temps défini pour la réaliser. La configuration demandée peut reprendre n'importe quel aspect du cours et inclura l'écriture de scripts.

La seule manière de vous entraîner à ce cours est d'y participer activement en réalisant les exercices proposés à chaque séance, ne prenant pas de retard par rapport à la théorie et en posant des questions si le besoin s'en fait ressentir.

La matière est conséquente et une bonne préparation est vraiment nécessaire à la maîtrise des concepts vus ici.

Leçon 1 : Introduction à Linux

Dans cette leçon, nous allons découvrir le système Linux. Beaucoup d'information seront données dans cette leçon, certaines seront des révisions du laboratoire de système d'exploitation.

Avant de commencer, une référence intéressante pour cette leçon est :

[RHELC] T. BARTOLONE, *Red Hat Enterprise Linux CentOS – Mise en production et administration de serveurs*, 2ème édition, Editions ENI, février 2015

1.1 Déploiement de la machine virtuelle

Une machine virtuelle toute prête est disponible. Elle est déjà installée et certains outils ont déjà été ajoutés pour faciliter l'apprentissage. Le lien vers la machine virtuelle se trouve sur la page web décrivant le cours.

Pour l'utiliser dans VMware, il faut décompresser le fichier dans le dossier `c:\admsys`. Une fois décompressée, il faut démarrer VMware et choisir les options **File > Open** et sélectionner la machine virtuelle décompressée.

Avant de démarrer la machine virtuelle, n'oubliez pas de démarrer **le firewall pfSense** afin de connecter votre machine Linux à internet.

1.1.1 Connexion au réseau

Avant de commencer, il est nécessaire de bien comprendre la connexion au réseau au travers de VMware.

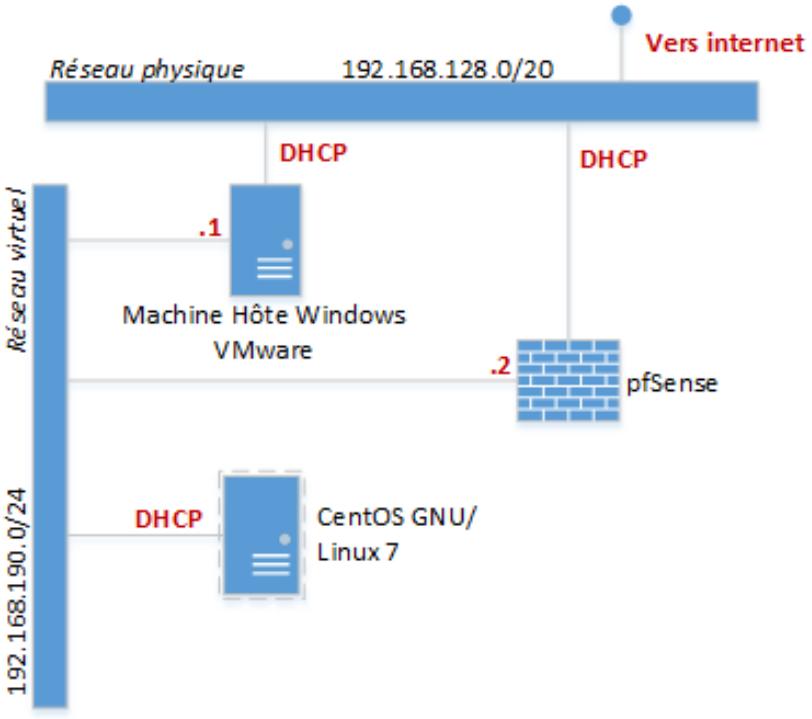

Figure 1.1 : Le réseau virtuel

Comme nous pouvons le voir sur la figure 1.1, il y a *le réseau physique* connecté au réseau de la haute école (ou de votre domicile), lui-même relié à Internet. Lors de l'installation de VMware, celui-ci ajoute des *réseaux virtuels*. Le réseau virtuel peut être utilisé pour interconnecter les machines virtuelles entre-elles et les relier à Internet.

Ainsi, sur le réseau configuré sur le campus Guillemins, nous avons le réseau virtuel *192.168.190.0/24* qui est dédié aux machines virtuelles. Il est configuré **de la même manière sur chaque machine qui tourne VMware** et cela ne pose aucun souci puisque ce réseau *est privé* et contenu sur la machine exécutant VMware.

Afin de connecter les machines virtuelles à Internet, un firewall gratuit virtuel, appelé pfSense, est installé. Il permet de relier le réseau virtuel au réseau physique. Il assure **également une complète isolation du réseau virtuel et du réseau physique**. Ainsi, si vous faites des erreurs de configuration sur vos machines virtuelles, cela n'aura pas d'impact sur le réseau du Campus Guillemins.

Voici un résumé des connexions prévues :

Machine	Réseau physique	Réseau virtuel
Machine Windows exécutant VMware	IP via DHCP	192.168.190.1
Firewall pfSense	IP via DHCP	192.168.190.2
Serveur virtuel CentOS	**Aucune connexion**	IP via DHCP

Configuration de VMware à domicile

Si vous souhaitez travailler ou faire fonctionner les machines virtuelles chez vous, il faut reproduire la configuration virtuelle. Pour ce faire, il y a 2 étapes à accomplir :

1. Via l'outil Virtual Network Editor installé avec VMware (accessible depuis le menu **Edit** de VMware Workstation), il faut vérifier les paramètres pour **VMnet1** :
    - VMnet Information : Host-Only
    - Vérifier que l'option *Connect a host virtual adapter to this network* est **activée**
    - Vérifier que l'option *Use local DHCP service to distribute IP address to VMs* est **désactivée**
    - Configurer les paramètres réseaux comme suit : *Subnet IP* : **192.168.190.0** et *Subnet mask* : **255.255.255.0**

Une fois cette étape terminée, votre configuration devrait ressembler à celle présentée sur la figure 1.2.

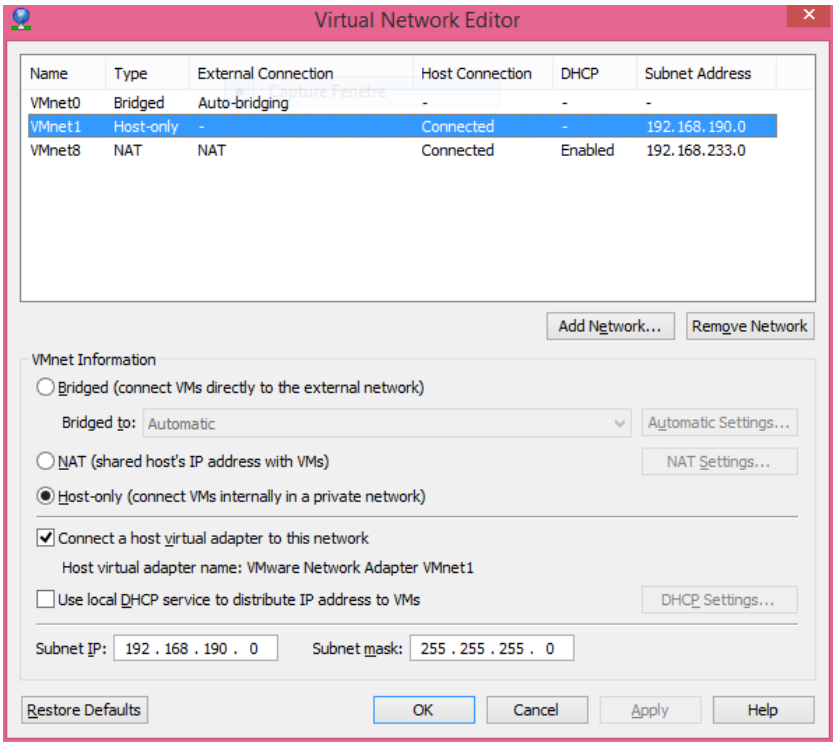

Figure 1.2 : Outil Virtual Network Editor

2. Pour la seconde étape, il faut ouvrir **le centre de réseau et de partage** et choisir l'option à gauche **Modifier les paramètres de la carte**, et faire un **clic-droit** sur l'interface *VMware Network Adapter VMnet1* et choisir **Propriétés**.

   Il faut ensuite choisir **Protocol Internet version 4 (TCP/IPv4)** et choisir **Propriétés**.

   Il faut alors vérifier que :

   - *Utiliser l'adresse IP suivante* est bien **sélectionnée**.
     - *Adresse IP :* `192.168.190.1`
     - Masque de sous-réseau : `255.255.255.0`

   Ce sont les seuls paramètres qui doivent être configurés

Une fois ces 2 étapes achevées, la configuration du réseau virtuel devrait être fonctionnelle et votre serveur virtuel devrait fonctionner aussi bien sur les machines de l'école que sur votre machine personnelle.

1.1.2 Accès au firewall pfSense

Le firewall pfSense dispose d'une interface web permettant de configurer celui-ci. Nous aurons, dans la suite du cours, l'occasion de revenir sur ses paramètres. Notons surtout que pfSense permet une isolation complète entre le réseau virtuel et le réseau du campus.

Pour y accéder, il faut simplement se connecter à l'adresse : `http://192.168.190.2`

- Login : `admin`
- Mot de passe : `rootroot`

Ouverture de ports dans pfSense

Pour ouvrir des ports dans pfSense, il faut se connecter à l'interface web de configuration du firewall et puis aller dans le menu **Firewall > NAT**.

Supposons que nous souhaitions ouvrir et rediriger le port 21 (=FTP) et tous les ports entre 15000 et 15500 vers notre machine virtuelle. Nous allons devoir ajouter 2 règles[3] pour permettre la connexion sur ces ports depuis le réseau de l'école :

1. Interface : **WAN** ; Protocol : **TCP** ; Destination : **WAN address** ; Destination port range from : **(other) 15000** to **(other) 15500** ; Redirect target IP : **<ip de votre serveur Linux>** ; Redirect target port : **15000** ; Cliquer sur SAVE

2. Interface : **WAN** ; Protocol : **TCP** ; Destination : **WAN address** ; Destination port range from : **FTP** to **FTP** ; Redirect target IP : **<ip de votre serveur Linux>** ; Redirect target port : **FTP** ; Cliquer sur SAVE

   Cliquer sur **Apply Changes**

   Les ports en question sont désormais redirigé vers la machine virtuelle. Ainsi, toute connexion sur un de ces ports en utilisant l'adresse IP extérieur provoquera la connexion vers la machine virtuelle sur ces mêmes ports.

1.2 Présentation de Linux

Linux est un système d'exploitation particulier : en effet, il s'agit d'un système d'exploitation *libre*. C'est à dire que, parmi les droits qui sont concédés aux utilisateurs, il y a la *liberté de modifier* le système et de l'adapter comme souhaité. Linux est d'ailleurs présent dans beaucoup d'environnements embarqués. Ainsi, beaucoup de « box internet » utilisent Linux pour interconnecter votre réseau personnel à celui de votre fournisseur internet (FAI ou ISP).

Outre la liberté, Linux est également un système extrêmement stable. Ce qui le rend particulièrement adapté à l'exécution de programmes serveurs. Ainsi, beaucoup de services critiques sur Internet, comme le DNS ou le mail, utilisent abondamment des serveurs Linux.

On trouve également beaucoup de serveurs LAMP (acronyme de Linux Apache MySQL Php) permettant de faire tourner des sites web dynamiques.

Pour **trouver de l'aide** sous Linux, Internet est une véritable mine d'information. Il ne faut pas négliger non plus **les pages de manuels** qui présentent toutes les commandes et les options :

```
$ man ls
```

Affiche la page de manuel pour la commande `ls`. La navigation dans le manuel est simple : `<espace>` permet de passer à la page suivante, `b` à la page précédente et `q` permet de quitter. Les fleches du clavier peuvent également être utilisées pour naviguer dans le manuel.

[3] Les ports proposés sont donnés à titre d'exemple (21 et ceux compris entre 15000 et 15500)

1.2.1 Disque, partition, LVM et système de fichiers

Un disque dur est un espace disponible pour le système. On trouve aujourd'hui des disques durs jusqu'à 4 To, bon marché. Afin d'exploiter efficacement cet espace, il est possible de *créer des partitions* (i.e. découpe logique du disque en plusieurs unités). Comme tout périphérique, le disque dur est renseigné dans le dossier `/dev`. Le premier disque dur SATA ou SCSI porte le nom `/dev/sda`, le second `/dev/sdb`, et ainsi de suite.

La gestion des partitions se fait soit en utilisant MBR (Master Boot Record - implémenté en 1983) ou GPT (GUID Partition Table - développé début 2000). Les deux technologies sont toujours présentes aujourd'hui même si l'antique MBR tend à disparaître avec l'apparition des BIOS compatibles UEFI et les disques durs de plus en plus gros. Ce qu'il faut simplement retenir est qu'il est possible d'utiliser l'un ou l'autre système et que la manière dont Linux interprète les informations est simple.

En mode MBR, on peut découper un disque en partition primaire (4 maximum) ou étendue (3 primaires et 1 étendue). Une partition étendue peut contenir autant de partitions logiques que nécessaire. De plus, il est possible d'installer Linux sur n'importe quel type de partition. La taille maximale du disque est de 2 To. Ainsi, en Linux, la 1ère partition principale sera renseignée par `/dev/sda1` (si elle se situe sur le 1ier disque), `/dev/sda2` pour la seconde et ainsi de suite. Les partitions logiques commencent avec `/dev/sda5`.

En mode GPT, il est possible de créer autant de partition que souhaitée. Il n'y a pas de limite quant au nombre, ni à l'espace alloué. Les disques modernes (de plus de 2 To) sont supportés sans problème. Avec ce mécanisme, les partitions sont identifiées à l'aide d'un identifiant global et sont ensuite numérotées par le système `/dev/sda1`, `/dev/sda2`, ….

Et LVM dans tout ça …

Logical Volume Manager ou LVM est un mécanisme sous Linux permettant de dissocier un système de fichiers de son emplacement sur un ou plusieurs disques ou partitions. En fait, LVM introduit la notion de *volume physique (PV)* qui représente un disque ou une partition. La seconde notion importante est *le volume group (VG)* qui peut regrouper plusieurs volumes physiques. Grâce à ces VG, il est possible de créer un espace allant au-delà d'un seul volume physique.

Enfin, la dernière notion importante est *le volume logique (LV)* qui est l'espace destiné à un système de fichiers. Le LV est localisé sur un VG (qui lui peut recouvrir plusieurs disques ou partitions). On voit ainsi apparaître une grande flexibilité : il est possible d'agrandir un volume logique en lui dédiant un nouveau disque que l'on ajoute au VG configuré.

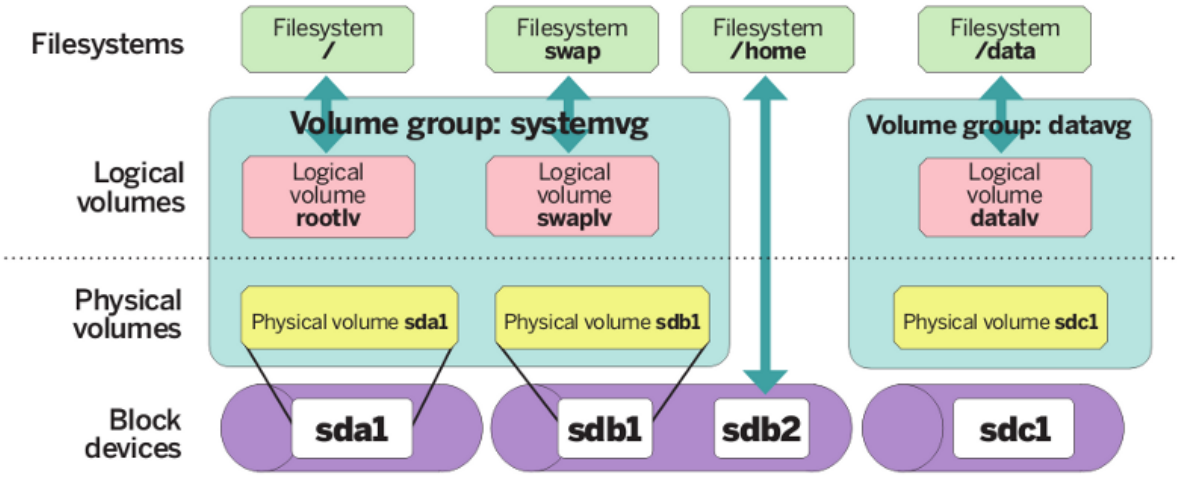

Figure 1.3 : Exemple d'utilisation de LVM[4]

Sur la figure 1.3, nous avons 3 disques durs SATA ou SCSI nommés `sda`, `sdb` et `sdc`. Sur le 1ier et le dernier disque, il y a une seule partition (`sda1` et `sdc1`) tandis que le second disque contient deux partitions (`sdb1` et `sdb2`).

Dans cet exemple, 3 volumes physiques sont configurés pour utiliser LVM: `sda1` et `sdb1` d'une part et `sdc1` d'autre part. Enfin, 2 volumes groups sont ajoutés : le 1ier `systemvg` utilise les volumes physiques `sda1` et `sdb1`, le second, `datavg` utilise le dernier volume physique `sdc1`.

Enfin, les volumes logiques sont créés : `rootlv` et `swaplv` sur le VG `systemvg`, `datalv` sur le VG `datavg`. Les systèmes de fichiers sont attachés au LV : la racine (le `/`) sur `rootlv`, la mémoire virtuelle sur `swaplv` et le dossier `/data` sur `datalv`. Il reste le dossier /home qui est directement contenu dans la partition `/dev/sdb2` (sans passer par LVM donc).

Sur notre machine virtuelle, MBR et LVM sont utilisés conjointement :

```
$ fdisk /dev/sda
```

Cette commande permet de gérer les partitions MBR sur le disque dur.

```
$ pvdisplay
$ vgdisplay
$ lvdisplay
```

Toutes ces commandes permettent de visualiser les informations concernant les PV, VG et LV configurés. Dans notre système, il y a 1 seul PV qui est `sda2`. Il y a un volume group reprenant le PV configuré (et se nommant `centos`).

Sur ce VG, il y a 3 volumes logiques (LV) nommés `root`, `swap` et `home`. Le premier reprend le système de fichiers racine (i.e. le `/`), le second est utilisé pour la mémoire virtuelle et le dernier reprend le stockage des dossiers personnels (i.e. `/home`). Ils sont accessibles au travers des noms de périphériques particuliers : `/dev/mapper/centos-root`, `/dev/mapper/centos-swap` et `/dev/mapper/centos-home`.

[4] La figure est extraite de : http://www.tuxradar.com/content/lvm-made-easy

1.2.2 Les répertoires

Les systèmes UNIX proposent une structure de répertoire assez commune. Ainsi, on trouve généralement les dossiers suivants :

Dossiers	Utilisation habituelle
/etc	Dossier contenant les fichiers de configuration du système (configuration réseau, système, mais aussi les différents services installés)
/dev	Dossier pointant vers les périphériques attachés au système. Il est possible ainsi de désigner un disque, une partition, un port d'E/S (USB par exemple) ou encore des périphériques particuliers (appelé pseudo-périphérique) comme `/dev/zero` (qui retourne uniquement des zeros) ou `/dev/random` (qui retourne des nombres aléatoires) ou `/dev/null` (qui absorbe tout et ne retourne rien).
/media	Dossier vide qui peut être utilisé par l'administrateur pour *monter* manuellement un périphérique (lecteur DVD, clé USB, dossier réseau distant, …)
/home	Dossier standard pour contenir l'ensemble des répertoires personnels des utilisateurs (excepté pour l'utilisateur root). Il arrive souvent que son contenu soit stocké sur une autre partition ou sur un serveur distant.
/root	Répertoire personnel de l'administrateur (utilisateur root)
/var	Dossier contenant des informations *en cours de traitement*. Ainsi, on y trouve les files d'attente pour l'impression (`/var/spool/lpd`) ou les boîtes aux lettres des utilisateurs (`/var/spool/mail`), les pages web – si le serveur web est installé – (dans `/var/www`), les bases de données MySQL (`/var/lib/mysql`), …
/bin & /sbin **/usr/bin** **/usr/sbin**	Dossiers contenant les programmes exécutables installés sur le système. Les dossiers `bin` contiennent les programmes disponibles pour tous les utilisateurs alors que les dossiers `sbin` contiennent plutôt les programmes qui peuvent être lancés par l'administrateur
/tmp	Dossier temporaire
/proc	Dossier particulier (il s'agit d'un système de fichier monté) contenant des informations sur les processus en cours d'exécution mais aussi sur le matériel et la configuration logicielle du système d'exploitation. Ce dossier est automatiquement généré par le système. Il ne faut donc rien y ajouter. Ex : `/proc/cpuinfo` identifie le processus du système

1.2.3 Administration du serveur par le Web

Un outil particulier est installé pour permettre l'administration du serveur par une interface web. Cet outil, nommé *Webmin*, permet d'ajouter, adapter, supprimer des éléments de configuration comme : ajouter des utilisateurs ou des groupes, administrer le disque dur, voir/planifier des tâches, configurer certains services, adapter les paramètres réseaux, …

Cependant, dans notre étude du système Linux, **nous préférerons souvent la ligne de commande** qui a le grand avantage de permettre de scripter facilement toutes ces tâches.

Un second outil, *Usermin*, est également installé. Il permet la modification des paramètres personnels de chaque utilisateur. Voici comment accéder à ces outils :

- Webmin : https://localhost:10000
- Usermin : https://localhost:20000

1.2.4 Les utilisateurs globaux et locaux

Comme tous les systèmes d'exploitations actuels, Linux supporte à la fois des utilisateurs *locaux* (i.e. définit *localement* sur la machine) et les utilisateurs *globaux* (i.e. définit dans un annuaire partagés entre plusieurs machines).

Les utilisateurs ***globaux*** sont utilisés lorsqu'il faut permettre à des utilisateurs de se connecter sur n'importe quelle machine du réseau. On définit alors ces utilisateurs dans un annuaire LDAP et on renseigne cet annuaire au niveau des différentes machines. Ainsi, l'authentification est *déléguée* à l'annuaire (et les informations concernant ces utilisateurs sont centralisées en 1 point) qui vérifie si le nom d'utilisateur et le mot de passe sont corrects.

A l'inverse, les utilisateurs ***locaux*** sont utilisés pour permettre l'accès au serveur considéré. Ainsi, il est possible de permettre la connexion sur ce seul serveur puisque l'utilisateur est seulement connu par celui-ci.

Il faut bien remarquer que dans l'annuaire, il n'y a que les utilisateurs globaux alors que dans les fichiers de configuration de la machine, il n'y a que les utilisateurs locaux.

Les utilisateurs et groupes locaux

Les utilisateurs **locaux** (uniquement) sont renseignés dans le fichier `/etc/passwd`. Chaque ligne de ce fichier texte décrit un utilisateur existant sur le système. Les informations suivantes sont renseignées pour chaque utilisateur, dans cet ordre (séparé par le symbole « : ») :

Information	Explication
login	Nom utilisé pour la connexion sur le serveur, c'est l'identifiant texte. Il doit être unique.
Mot de passe	Le mot de passe n'est pas précisé dans ce fichier mais dans le fichier `/etc/shadow` sous un format haché. C'est pourquoi celui-ci est remplacé par `x` dans ce fichier.
UID	Identifiant numérique unique associé à l'utilisateur. L'utilisateur `root` a toujours l'UID `0`. Les utilisateurs locaux ont un UID >= `1000`. Notons que les identifiants < `500` sont réservés pour **des comptes systèmes**.
GID	Identifiant numérique du groupe principal de l'utilisateur. Précisons que le groupe `root` à le GID `0`. Le groupe `users` a le GID `100`.
Informations utilisateur	Le champ suivant reprend les informations sur l'utilisateur : son nom, son prénom, son téléphone, son bureau, …
Dossier personnel	L'avant dernier champ mentionne le dossier personnel de l'utilisateur. A l'exception des comptes systèmes et de l'utilisateur `root`, les dossiers des utilisateurs sont stockés dans `/home`. Ainsi, l'utilisateur `lsw` a son répertoire personnel dans `/home/lsw`. L'utilisateur `root` a son répertoire dans `/root`.
Le shell	Le dernier argument est le shell à lancer lors de la connexion de l'utilisateur. Nous utiliserons toujours `bash` comme shell en précisant : `/bin/bash` comme shell.

Les commandes pour gérer les utilisateurs sont :

useradd

Cette commande permet d'ajouter un utilisateur sur le système. Par défaut sur les systèmes RedHat/CentOS, cette commande crée un groupe du nom de l'utilisateur et intègre l'utilisateur dans ce groupe principal. Comme ce comportement n'est pas souhaité, il convient de préciser les arguments suivants :

Exemple :

```
$ useradd -g users superswila
```

Cette commande ajoute l'utilisateur (dont le login est `superswila`) dans la liste des utilisateurs locaux du système. Cet utilisateur est placé dans le groupe principal `users`. Par defaut, le système lui attribue le shell `bash` et détermine le chemin vers le dossier personnel dans `/home/superswila`. Consulter la page de manuel pour plus d'information[5].

Le compte de l'utilisateur **est désactivé tant qu'aucun mot de passe n'est précisé**.

usermod

Cette commande permet de modifier les paramètres d'un utilisateur local déjà créé (par exemple, l'ajout dans un groupe secondaire, …). Consulter la page de manuel pour plus d'information[5].

userdel

Cette commande permet de supprimer un utilisateur local existant. Sans option particulière, le dossier personnel de l'utilisateur est conservé. Consulter la page de manuel pour plus d'information[5].

chfn

Cette commande permet de changer le nom de l'utilisateur, préciser son bureau, et toutes les informations utilisateurs qui lui sont attachées. Consulter la page de manuel pour plus d'information[5].

passwd

Sans paramètre, cette commande permet **de changer le mot de passe de l'utilisateur courant**. L'administrateur peut préciser le login d'un utilisateur pour changer ou fixer le mot de passe de celui-ci. Quand le mot de passe est mentionné pour la 1ère fois, le compte est automatiquement activé.

id

Sans paramètre, cette commande permet de connaître le nom d'utilisateur courant. Il précise également les groupes (principaux et secondaires) de cet utilisateur. Il est possible de préciser le login d'un utilisateur, la commande retourne alors les informations de cet utilisateur.

chsh

Cette commande permet de changer le shell d'un utilisateur.

Les groupes **locaux** sont mentionnés dans le fichier `/etc/group`. Le format est analogue à celui des utilisateurs et il indique :

Information	Explication

[5] Pour rappel : `man useradd` (`<espace>` page suivante, `b` page précédente, `q` quitter).

groupname	Nom textuel utilisé pour identifier ou afficher le groupe
x	Ce champ ne sert pas ici
GID	L'identifiant numérique associé au groupe. Le groupe `root` utilise l'identifiant `0`, le groupe `users` utilise l'identifiant `100`.
liste des utilisateurs séparés par « , »	Précise les utilisateurs ayant **comme groupe <u>secondaire</u> le groupe en question**. Ainsi, si l'utilisateur lsw a comme groupe principal le groupe `users` et comme groupe secondaire `audio`, nous aurons : l'identifiant `100` précisé comme GID dans le fichier `passwd` pour cet utilisateur et, dans les membres du groupe `audio`, le login `lsw` apparaitra. Par contre, il n'apparaitra pas pour le groupe `users` (ce n'est pas un groupe secondaire).

Les commandes pour gérer les groupes sont :

groupadd

Cette commande permet d'ajouter un groupe sur le système. Consulter la page de manuel pour plus d'information.

groupmod

Cette commande permet de modifier un groupe déjà existant. Consulter la page de manuel pour plus d'information.

groupdel

Cette commande permet de supprimer un groupe existant. Consulter la page de manuel pour plus d'information.

groups

Cette commande permet de connaître les groupes auxquels l'utilisateur appartient. Consulter la page de manuel pour plus d'information.

Les utilisateurs et groupes globaux

Les utilisateurs et groupes **globaux** sont précisés dans l'annuaire LDAP. L'annuaire est déjà installé sur le serveur CentOS virtuel. Il est possible d'accéder, en mode graphique, à ce dernier comme suit :

```
$ 389-console
```

Cette commande permet de démarrer l'interface d'administration graphique en Java. Il faut préciser les informations de connexion suivantes :

- User ID : `cn=Directory Manager`
- Password : `rootroot`
- Administration URL : `http://localhost:9830`

L'interface est alors démarrée. Si l'on déploie l'arbre, on voit apparaître deux éléments importants :

- *Administration Server :* cette option reprend tous les paramètres de configuration **du serveur d'administration**. Il n'est pas nécessaire d'y accéder.
- *Directory Server (localhost) :* cette option reprend la configuration et le contenu **de l'annuaire LDAP**. C'est cette partie que nous allons explorer.

Si l'on l'explore l'élément *Directory Server*, on voit apparaitre la fenêtre de gestion de l'annuaire comportant plusieurs onglets. L'onglet *Task* permet de relancer le service, gérer les certificats, etc.

L'onglet *Configuration* permet de déterminer comment se connecter à l'annuaire (activation de SSL/TLS), configurer la réplication, etc. Ensuite, l'onglet *Directory* permet de parcourir le contenu de l'annuaire. Enfin, l'onglet *Status* regroupe toutes les informations concernant l'état du système, les fichiers journaux, etc.

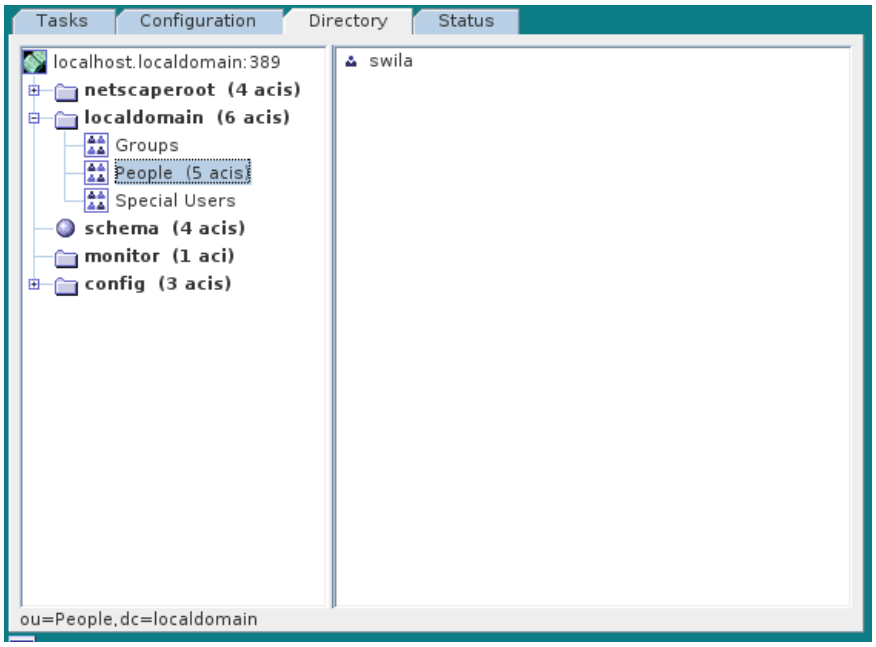

Figure 1.4 : Le contenu de l'annuaire LDAP

Comme on peut le voir sur la figure 1.4, nous pouvons consulter le contenu de l'annuaire. Il y a déjà plusieurs *unités d'organisation* qui existent : `Groups`, `People` et `Special Users`. Ces unités permettent de structurer l'annuaire de sorte à ce qu'il reproduise l'organisation de l'entreprise. Ainsi, il n'est pas rare de créer des unités organisationnelles (abrégées en OU) pour chaque service de l'entreprise. Une OU est un conteneur pouvant accueillir des utilisateurs ou des groupes globaux. Ainsi, dans la OU `People`, nous avons déjà l'utilisateur `swila` qui est présent. Il s'agit donc d'un utilisateur global.

Pour **créer une OU**, il faut simplement faire un **clic-droit** dans l'arbre à gauche, dans le conteneur parent et choisir **New > Organizational Unit**. Il faut préciser le nom (qui est obligatoire) et éventuellement une description.

Il est également possible de créer cette OU par la ligne de commande. Il faut commencer par créer un fichier texte reprenant l'unité organisationnelle à créer (par exemple : `/tmp/ou.ldif`):

```
dn: ou=Student, ou=People, dc=localdomain
ou: Student
description: Tous les etudiants
objectClass: top
objectClass: organizationalUnit
```

Nous pouvons ensuite importer les informations dans l'annuaire avec la commande :

```
$ ldapadd -D 'cn=Directory Manager' -f /tmp/ou.ldif -x -W
```

Le mot de passe est demandé. L'unité `Student` est ensuite créée dans l'annuaire dans l'unité `People`.

Pour **créer un utilisateur** dans une OU, il faut simplement faire un clic-droit dans le conteneur souhaité et choisir l'option **New > User**. Dans les options à préciser, il faut remplir *le first name*, le *last name*, le *common name*, le *user ID* (= login) et le *password*. Il faut également **remplir les informations Posix User** pour permettre la connexion de cet utilisateur. Ainsi, il faut spécifier son *UID Number* (= son identifiant numérique, idéalement >= 5000 pour éviter tout conflit avec les utilisateurs locaux), *GID Number* (= identifiant numérique du groupe principal – peut être `100` si le groupe principal est `users`), le *Home directory* (= le chemin vers son dossier personnel – idéalement `/home/<login>` qu'il faut créer à la main), et le *Login Shell* (= `/bin/bash`).

Pour créer un utilisateur global par ligne de commande, il faut créer un fichier texte (par exemple : `/tmp/user.ldif`) :

```
dn: uid=ldapswila,ou=Student,ou=People,dc=localdomain
objectClass: top
objectClass: inetorgperson
objectClass: posixAccount
cn: Louis SWINNEN
sn: SWINNEN
givenname: Louis
userPassword: rootroot
gidNumber: 100
uidNumber: 5001
homeDirectory: /home/ldapswila
loginShell: /bin/bash
```

L'ajout dans l'annuaire se fait par la commande :

```
$ ldapadd -D 'cn=Directory Manager' -f /tmp/user.ldif -x -W
```

Cette commande provoque l'ajout de l'utilisateur ldapswila dans l'OU Student de l'annuaire LDAP.

Pour créer **un groupe**, il faut faire **un clic-droit** sur le conteneur et choisir **New > Group**. Il faut mentionner le *Group Name* et, éventuellement une description. **Il faut également remplir les informations *Posix Group*** pour indiquer le *GID Number* (= l'identifiant numérique du groupe), et *inclure les utilisateurs* ayant ce groupe comme groupe secondaire.

Pour créer un groupe global par ligne de commande, il faut créer un fichier texte (par exemple : `/tmp/group.ldif`) :

```
dn: cn=bac1,ou=Groups,dc=localdomain
description: bac1
objectClass: top
objectClass: groupOfUniqueNames
objectClass: posixGroup
gidNumber: 5000
```

L'ajout dans l'annuaire se fait par la commande :

```
$ ldapadd -D 'cn=Directory Manager' -f /tmp/group.ldif -x -W
```

Cette commande provoque l'ajout du groupe `bac1` dans l'OU `Groups`.

Pour ajouter un utilisateur dans un groupe existant (`/tmp/addmember.ldif`) :

```
dn: cn=bac1,ou=Groups,dc=localdomain
changetype: modify
```

```
add: memberuid
memberuid: ldapswila
```

L'ajout dans l'annuaire se fait par la commande :

```
$ ldapadd -D 'cn=Directory Manager' -f /tmp/addmember.ldif -x -W
```

Cette commande provoque l'ajout de l'utilisateur `ldapswila` dans le groupe `bac1`.

1.3 Les commandes de base[6]

1.3.1 Les commandes sur les dossiers

ls

Cette commande permet de lister les répertoires et les fichiers.

```
ls [-l|-a|-h] [fichier...]
```

Exemples:

```
$ ls
$ ls -l
$ ls -a -l est équivalent à ls -al
$ ls -la ~
```

cd

Cette commande permet de changer le répertoire courant

```
cd [répertoire]
```

Exemples:

```
$ cd /tmp
$ pwd
$ cd ..
$ cd -
$ cd
```

mkdir

Permet de créer un répertoire

```
mkdir [-m mode|-p] répertoire...
```

Exemples:

```
$ mkdir rep1 rep2
$ mkdir -p rep3/test/exemple doc/louis/linux
```

rmdir

Permet de supprimer un répertoire vide

```
rmdir [-p] répertoire...
```

Exemples:

```
$ rmdir rep2
$ rmdir -p rep3/test/exemple
```

[6] Cette partie est extraite du cours *d'introduction à Linux* du NamurLUG (`http://www.namurlug.org`).

1.3.2 Les commandes sur les fichiers

mv

Cette commande permet de changer le nom d'un répertoire ou d'un fichier. Cette commande permet également de déplacer un fichier ou un répertoire.

```
mv [-i] source dest
mv [-i] source... répertoire
```

Exemples:

```
$ mv rep1 jef
$ mv jef doc/
```

cp

Permet de copier un fichier ou un répertoire.

```
cp [-i | -r] source dest
cp [-i | -r] source... répertoire
```

Exemples:

```
$ cp -r /etc doc/jef
$ cp /etc/nsswitch.conf doc/louis
```

rm

Permet de supprimer un fichier ou un répertoire

```
rm [-i | -f | -r] fichier...
```

Exemples:

```
$ rm -rf doc/jef
$ rm -i doc/louis/nsswitch.conf
```

file

Cette commande permet de connaître le type d'un fichier

```
file fichier...
```

Exemples:

```
$ file /usr/bin/man
$ file /etc/nsswitch.conf
$ file doc
```

more et **less**

Ces commandes permettent de visualiser le contenu d'un fichier texte, page par page.

```
more fichier
less fichier
```

Exemples:

```
$ more /etc/nsswitch.conf
$ less /etc/nsswitch.conf
```

cat

Cette commande permet de fusionner plusieurs fichiers et afficher le résultat sur la sortie standard (écran, par défaut). Cette commande est également utilisée pour afficher le contenu d'un fichier à l'écran.

```
cat fichier...
```

Exemples:

```
$ cat /etc/nsswitch.conf /etc/fstab
```

find

Cette commande permet d'effectuer une recherche d'un fichier.

```
find répertoire... expression
```

Exemples:

```
$ find / -name man
$ find / -name emacs -exec file {} \;
```

grep

Cette commande permet de faire une recherche à l'intérieur d'un fichier texte. Elle affiche les lignes contenant au moins une occurrence de l'expression recherchée.

```
grep [-r] [-i] "chaîne" fichiers...
```

Exemples:

```
$ grep "files" /etc/nsswitch.conf
```


1.3.3 Les redirections

Le système Linux définit 3 flux standards :

- *stdin :* l'entrée standard, par défaut attachée au clavier
- *stdout :* la sortie standard, par défaut attachée à l'écran
- *stderr :* la sortie pour les erreurs, par défaut attachée à l'écran

Chaque commande *peut* utiliser ces 3 flux standards. Par exemple, lors d'un *printf* en C, le résultat est par défaut envoyé vers la sortie standard. Il est possible de rediriger ces flux vers des fichiers.

Redirection de la sortie standard

Le résultat est alors envoyé dans un fichier.

```
commande > fichier      le fichier est créé ou écrasé
commande >> fichier     le résultat est ajouté au fichier
```

Exemples:

```
$ ls > out
$ ls -l >> out
```

Redirection du standard d'erreur

Les messages d'erreurs sont alors envoyés dans un fichier.

```
commande 2> fichier    Redirige le standard d'erreur
commande &> fichier    Redirige la sortie standard + erreur
```

Exemples:

```
$ find / -name emacs 2> find-error
```

Redirection de l'entrée standard

Les données ne sont plus demandées au clavier et sont prises à partir d'un fichier.

```
commande < fichier
```

Exemples:

```
$ mail root < /etc/inittab
```

Liens entre plusieurs commandes

L'intérêt majeur des redirections est de pouvoir lier la sortie d'une commande à l'entrée d'une autre. Les systèmes UNIX permettent de réaliser cela en une seule opération. Il est alors possible de chaîner plusieurs commandes réalisant des opérations ciblées.

```
commande1 | commande2
```

Exemples:

```
$ cat /etc/fstab | less
$ cat /etc/nsswitch.conf | grep "files"
$ cat /etc/aliases | grep "news"
```


1.3.4 Commandes sur la gestion des partitions

La description des différentes partitions du système est concentrée dans le fichier `/etc/fstab`. On trouve, pour chaque partition le point de montage associée, le système de fichier utilisé et les options utilisées lors du montage (nous verrons les différentes options lors d'une prochaine séance).

fdisk

Cette commande permet de modifier les partitions du système (à utiliser avec précaution !). Elle permet également de connaître l'état des partitions. Il suffit d'utiliser l'option p du menu pour connaître les différentes partitions du système.

```
fdisk periphérique
```

Exemples:

```
$ fdisk /dev/sda
```

mount

Cette commande permet de monter une partition dans le système de fichier. Par cette commande, on rend accessible un système de fichier. Il peut s'agit d'une clé ou d'un disque USB, d'un DVD (`/dev/sr0`), d'une partition sur un disque.

```
mount [-t système_fichier] périphérique point_de_montage/
```

Exemples:

```
$ mount /home
$ mount /dev/sr0 /media
$ mount -t vfat /dev/sdb1 /media
```

umount

Cette commande permet de supprimer l'association entre le système de fichier et le périphérique. Il est indispensable de démonter un périphérique avant de l'emporter. Cela assure que toutes les opérations ont bien été effectuées sur le disque.

```
umount peripherique
umount point_de_montage/
```

Exemples:

```
$ umount /home
$ umount /media
$ umount /dev/sdb1
```

e2fsck

Cette commande permet de vérifier la cohérence et l'intégrité du système *ext2/ext3/ext4*. Elle est lancée par le système lors d'un arrêt brutal (coupure de courant, problème matériel, ...). Dans certains cas, lorsque les erreurs sont graves, le système demande d'exécuter cette commande manuellement.

Cette commande ne peut être exécutée que sur des systèmes non montés !

Exemples:

```
$ umount /dev/mapper/centos-home
$ e2fsck -c /dev/mapper/centos-home
$ e2fsck -f /dev/mapper/centos-home
$ e2fsck -y /dev/mapper/centos-home
```

Dans ces exemples, nous réparons le système de fichier attaché au volume `centos-home`. Il est, en effet, formaté en *ext4*. Sans option particulière, le programme vérifie le système de manière interactive (si des problèmes sont rencontrés, l'utilisateur doit prendre une décision). L'option `-c` permet de lancer une vérification des secteurs défectueux, l'option `-f` force la vérification du système même si l'état est cohérent, l'option `-y` suppose la réponse *yes* à toutes les questions (correction des erreurs).

xfs_repair

Cette commande permet de vérifier la cohérence et l'intégrité des systèmes *xfs*. Comme la commande précédente, elle est lancée en cas d'arrêt brutal du système. Il est parfois nécessaire de lancer cette commande manuellement.

Cette commande ne peut être exécutée que sur des systèmes non montés !

Exemples :

```
$ umount /dev/sda1
$ xfs_repair -n /dev/sda1
$ xfs_repair /dev/sda1
```

Dans ces exemples, nous réparons le système de fichier présent sur la 1ère partition du disque. Il est, en effet, formaté en *xfs*. L'option `-n` (*no modify*) n'effectue aucune correction sur le système de fichier, il permet de savoir si celui-ci est corrompu et s'il doit être réparé. Sans option, le système tente de réparer les erreurs rencontrées.

1.4 Exercices

1. Au moyen de **la ligne de commande** :
    a. Créer un utilisateur local correspondant à votre login HELMo et placer le dans le groupe principal *users*. Précisez également vos informations (nom, prenom, …) et fixez le mot de passe.
    b. Créer un groupe local *friends*
    c. Créer un compte pour votre voisin dans le groupe principal *users* et dans le groupe secondaire *friends*.
    d. Modifier (usermod) votre compte pour qu'il appartienne également au groupe secondaire *friends*
    e. Vérifier avec les commandes et en consultant les fichiers adéquats que tout est correct. Tentez de vous connecter !

2. Au moyen de **la console d'administration LDAP** :
    a. Créez une OU *student* avec, à l'intérieur, une autre unité *BM*
    b. Dans l'OU *Groups,* créer 4 groupes globaux : *biomed1, biomed2, biomed3* et *etudiants*
    c. Dans l'OU *BM*, créer 3 utilisateurs globaux : *bm1*, *bm2*, et *bm3* appartenant au groupe principal *étudiants* et au groupe *biomedX* correspondant.

3. Au moyen **de la ligne de commande, insérer dans LDAP** :
    a. Une OU *Info* à l'intérieur de la OU *student*
    b. Dans l'OU *Groups*, les 3 groupes globaux : *info1*, *info2* et *info3*
    c. Dans l'OU *Info*, créer 3 utilisateurs globaux *inf1*, *inf2* et *inf3* appartenant au groupe principal *etudiants* et au groupe *infoX* correspondant.

    Pour ce faire, créer les fichiers LDIF et importez-les.