2011

Administration d'un système GNU/Linux

(RedHat© / Centos / Fedora)

Stéphane DUFOUR
04/03/2011

Administration d’un système GNU/Linux (RedHat© / Fedora)

Support de cours : **Stéphane DUFOUR**

Tables des matières

Présentation de la formation

Présentation du site de formation

- Vie pratique
- Horaire
- Fiche de présence
- Repas
- Notation fin de stage

Présentation Stagiaires / Formateur

- Cursus,
- Affectation,
- Tâches quotidiennes,
- Attente par rapport à cette formation,
- Supports fournis :
    - DVD (documentations, TP Corrigés, outils divers)
    - Cours format papier
    - Image d’installation au format .ISO

Planning de la semaine

Lundi	Mardi	Mercredi	Jeudi	Vendredi
Présentation Cours ○ Présentation GNU/Linux et son architecture ○ Installation Centos Travaux pratiques	Cours ○ Le démarrage et arrêt du système ○ La gestion du système de fichiers Travaux pratiques	Cours ○ Configuration du réseau ○ Les services réseaux ○ SMB ○ FTP Travaux pratiques	Cours ○ Ordonnancement et automatisation ○ La supervision du serveur Travaux pratiques	Cours ○ La virtualisation
Cours ○ Les commandes et outils ○ Installation de logiciels supplémentaires ○ Les journaux systèmes Travaux pratiques	Cours ○ La sauvegarde et restauration Travaux pratiques	Cours ○ Gestion des utilisateurs ○ NFS ○ LDAP ○ HTTP Travaux pratiques	Cours ○ L'impression ○ Le noyau Linux ○ La gestion des modules Travaux pratiques	

Présentation GNU/Linux et son architecture

Introduction

Cette formation s'adresse à un **public ayant déjà des connaissances du système GNU/Linux** et plus spécialement sur les distributions basées sur RedHat©/Centos/Fedora.

L'administrateur système est responsable de l'intégrité et de la disponibilité du SI qu'il administre au quotidien. Pour ce faire il doit donc disposer de **solides connaissances des matériels et logiciels** sur lesquels il travaille. Il doit également avoir une **vue d'ensemble de l'infrastructure réseau** dans laquelle il évolue afin de pouvoir établir rapidement et précisément un diagnostic d'un éventuel incident.

On attend donc d'un administrateur système certaines qualités :

- autonomie,
- compétences techniques théorique et pratique,
- calme,
- expérience,
- curiosité,
- sens des responsabilités.

Ce support de cours fait le tour des connaissances nécessaires pour aborder sereinement un environnement de production et introduit quelques réflexions quant aux architectures systèmes possibles en utilisant GNU/Linux comme brique principale.

Historique, le libre

GNU/Linux n'est plus un simple effet de mode et d'annonce.
Depuis ses tous **premiers développements en 1991** et jusqu'à aujourd'hui Linux ne cesse d'évoluer, de changer.

Le monde de l'informatique est vivant. S'il n'évolue pas, il végète. Avec GNU/Linux, des millions de personnes ont trouvé enfin ce qu'elles cherchaient.

GNU/Linux n'est pas plus compliqué à utiliser que n'importe quel autre système. Le frein au développement de GNU/Linux auprès du plus grand nombre n'est pas lié à un quelconque niveau de difficulté. L'expérience acquise auprès de nombreux utilisateurs débutants ou confirmés, des groupes d'utilisateurs GNU/Linux et des acteurs professionnels montre qu'il s'agit surtout d'un **problème lié aux habitudes des gens**, accoutumés des années durant à un **système d'exploitation unique**.

En effet, ces habitudes doivent parfois être quelque peu modifiées pour s'adapter à un environnement GNU/Linux, tout comme un parfait conducteur de voiture familiale ne fait pas de vous un as de la conduite sportive en Ferrari.

On voit bien qu'avec d'autres systèmes comme GNU/Linux ou Mac OS X il est possible de « penser différemment » qu'avec des « fenêtres ».

Le **libre est le fer de lance de GNU/Linux**. Sans communauté du libre pas de GNU/Linux.

La **notion d'échange et de partage du savoir sont indissociables de « l'esprit GNU/Linux »**. Cependant il y a eu, il y a et il y aura des « égarements » concernant la mise à disposition des contributions aux codes sources GNU/Linux par certaines entreprises ou indépendants.

En ce sens des distributions du type Debian GNU/Linux ont les faveurs des puristes du libres : elles sont indépendantes.

Comprendre l'architecture GNU/linux

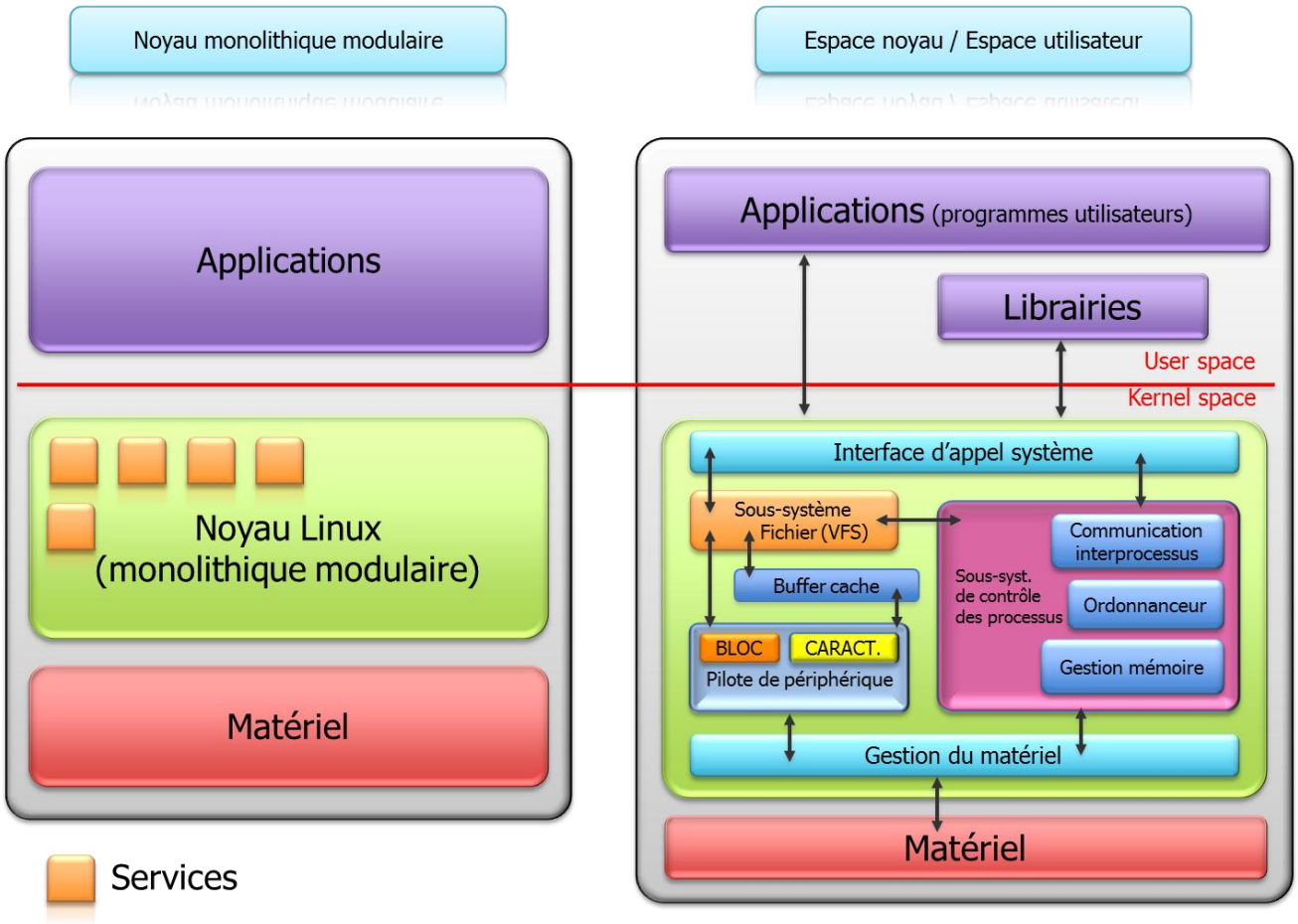
Tout d'abord il faut bien différencier GNU et Linux.
Linux est un noyau (Kernel) écrit en langage C dont le processus d'écriture a débuté en 1991 sous l'impulsion de Linus TORVALDS et se poursuit de nos jours.

Note : les numéros de versions de noyau impairs sont des versions expérimentales (Ex. 2.5.x)

Le projet **GNU est un ensemble de logiciels écrits autour ce noyau** pour former ce qu'on appelle communément une distribution Linux. Une des figures emblématiques de la FSF (Free Software Fondation) est Richard STALLMAN (auteur du logiciel Emacs entre autres)

GNU/Linux est un système d'exploitation:

- **multitâches**,
- **multi-utilisateurs**,
- supportant **la plupart des architectures matérielles (ARM, x86, Itanium, PowerPC etc.)**,
- **modulaire à souhait**,
- dont le **noyau est monolithique modulaire**.
- **conforme au standard** de l'informatique (POSIX, SYSTEM V, RFC, etc.)
- à haute **interopérabilité**.



Grâce aux deux schémas précédents on peut simplifier GNU/Linux ainsi :

- un noyau,
- un système de fichiers,
- un interpréteur de commande,
- des applications.

Le noyau, écrit en C, ne fera pas l'objet d'une étude poussée dans ce cours. Veuillez retenir qu'il assure principalement la gestion :

- des processus (IPC, Ordonnancement, etc.)
- de la mémoire,
- des différentes entrées sorties
- du matériel etc.

Le système de fichiers est une structure où sont stockées les données. Il en existe plusieurs types.
Nous distinguerons plusieurs familles :

- non journalisé (FATxx, ext2, HFS, UFS, HPFS, S5, FFS, etc.),
- journalisé (NTFS, ext3-4, HFS+, UFS+, JFS, XFS, ReiserFS, BeFS, NSS, etc.),
- systèmes de fichiers réseau,
- virtuels,
- spécialisés.
- Etc.

L'interpréteur de commande (**Shell**) est un programme particulier s'exécutant en « espace utilisateur » qui permet à l'utilisateur d'interagir avec le système d'exploitation et les applications.

Les programmes sont un ensemble de **commandes et services** disponibles sous une distribution GNU/Linux

Un système GNU/Linux est un système où **tout repose sur des fichiers de configuration**. A **un binaire** (exécutable) vous pouvez systématiquement faire **correspondre un fichier de configuration**.
Partant de cet état de fait, chaque **modification hasardeuse d'un fichier** peut très bien provoquer l'action attendue comme **mettre hors service votre système d'exploitation GNU/Linux** fraîchement installé.

Fort heureusement il y a des méthodes à utiliser pour régler efficacement votre système d'exploitation. De plus quelque soit l'action que vous aurez à effectuer sur le système : dans **99% des cas le redémarrage ne sera pas nécessaire** contrairement à de vieilles habitudes prises sur d'autres systèmes d'exploitation.

GNU/Linux est un système **non propriétaire**, bien que cela semble anodin à première vue, cela va vous permettre de voir ce qui se passe exactement dans ses entrailles : nous verrons que **durant les phases de mise en œuvre/au point ce détail sera primordial**.

GNU/Linux étant l'intégration d'un ensemble de programmes (libres et parfois commerciaux) autour d'un noyau, voyons maintenant quelles sont les principales distributions du marché.

Les distributions : RedHat© (Centos) – Fedora ?

Le choix d'une distribution n'est pas simple car l'offre est plutôt abondante. La plupart se valent en termes de qualité mais il y a des différences quant à leur statut moral et l'utilisation que l'on souhaite en faire.

Voici une présentation succincte des plus connues :

SUSE a été créée en 1993 à Nuremberg en Allemagne, elle a été rachetée par la société Novell à la fin de l'année 2003. Elle propose deux distributions principales : *SUSE Linux Enterprise* orientée vers les entreprises (certifications matérielles et logicielles nombreuses) et *openSUSE*, communautaire, orientée vers le grand public.

Debian est une **distribution non commerciale** régie par le ***contrat social Debian***. Elle se distingue par le très grand nombre d'architectures supportées, son importante logithèque et par son cycle de développement relativement long, gage d'une certaine stabilité. Sa **qualité et son sérieux** sont unanimement reconnus, mais elle garde l'image d'une distribution réservée aux experts, alors que son ergonomie a bien évolué.

Mandriva est la plus grande **distribution européenne**. C'est une distribution internationale d'origine **française** éditée par la société Mandriva. Très orientée vers le grand public, elle est conçue pour être facile d'installation et d'usage pour les débutants et les professionnels. Elle est disponible en plusieurs versions, certaines commerciales et d'autres (Live CD, DVD etc.)

Gentoo est une distribution caractérisée par sa gestion des paquetages à la manière des ports BSD, effectuant généralement la compilation des logiciels (X, OpenOffice, etc.) sur l'appareil de l'utilisateur. Elle est dédiée aux utilisateurs avancés, aux développeurs et aux passionnés. **La compilation des logiciels in situ** donne une liberté de choix de fonctionnalités et de dépendances très poussée, apportant davantage de souplesse dans la gestion des paquets que dans une distribution utilisant des paquets binaires.

Ubuntu, basée sur Debian. Distribution internationale orientée vers le grand public et sponsorisée par Canonical, édite des versions stables tous les 6 mois. Elle est disponible en live CD. Cette distribution dispose d'une communauté d'utilisateurs du monde entier très dynamique.

Red Hat© (officiellement *RedHat Enterprise Linux* ou RHEL) est une distribution commerciale largement **répandue dans les entreprises** (surtout aux États-Unis). La société Red Hat© qui la supervise a développé RPM, un gestionnaire de paquets sous licence GPL que d'autres distributions utilisent. Elle est la base de beaucoup de distributions du marché.

Fedora est une distribution communautaire supervisée par Red Hat. Elle est basée sur le système de gestion de paquetages logiciels RPM. Cette distribution est le **laboratoire de test de RedHat©**. Elle intègre les nouveautés en termes d'application et est plutôt orientée **grand public** donc pour un usage station de travail.

CentOS est une distribution GNU/Linux principalement **destinée aux serveurs**. Tous ses paquets, à l'exception du logo, sont des paquets compilés à partir des **sources de la distribution Linux Red Hat Enterprise Linux** (RHEL), éditée par la société Red Hat. Elle est donc quasiment **identique et 100% compatible**.

Voici venu le moment du choix.

En fait cela **dépendra de l'usage** que vous allez faire **de votre distribution**. Il n'y a pas de distribution universelle.

Néanmoins certains vous affirmeront qu'ils peuvent parfaitement adapter une « **Ubuntu Desktop »** pour en faire un très **bon serveur**, effectivement c'est parfaitement possible … mais à quel prix en termes de **temps de modification** etc. ?

Bref un petit jeu où ceux qui veulent aller à l'essentiel ne s'amuseront pas.

Prenons l'exemple de la **Gendarmerie nationale, le choix « Ubuntu »** a été fait pour le **poste de travail (70 000 postes)**. Cependant ce n'est pas cette même distribution qui a été retenu pour la partie serveur, où là, les architectes systèmes ont préféré une **distribution orientée « serveur », fiable, minimaliste et libérée de toutes contraintes commerciales : Debian** fût le choix idéal.

Cependant certains critères feront pencher la balance dans un sens, ces critères de choix peuvent être techniques mais aussi politiques. Pour notre part, au Ministère, une décision politique restreint le choix possible : **RedHat©**.
Entendons par là qu'il est donc également possible de choisir **Centos**.

Pour ce cours, « Administrateur Linux », nous nous baserons donc sur la distribution **Centos 5.5 32bits qui est le clone d'une RedHat© EL**, car directement construit à partir des sources RedHat© EL. Nous éviterons Fedora qui est une distribution dite « unstable » (testing etc.) et plutôt orientée poste de travail

Le matériel pour GNU/Linux

De nos jours le souci de choisir un matériel compatible avec le noyau Linux devient vraiment secondaire. On peut même affirmer que dans le cadre d'une utilisation professionnelle les distributions récentes embarquent nativement la plupart des pilotes.

Bien sûr il est conseillé de **vérifier, avant achat,** si votre **futur matériel est compatible GNU/Linux,** cela concerne surtout les derniers chipsets ou processeurs. Pour ce faire il faut connaître les éléments cruciaux de votre matériel. Voici une liste qui peut vous aider à valider votre configuration matérielle en vue d'une utilisation avec GNU/Linux sur un serveur :

- **Processeur** (32/64bits, **VT**, SSEx etc.),
- Chipset de la carte mère (**Southbridge** et **Northbridge**),
- **Chipset** de la carte contrôleur **RAID**,
- **Chipset** de la carte contrôleur **F**iber **C**hannel,
- **Chipset** de la carte **réseau**,
- Type de **mémoires**,
- **Chipset** de la carte **vidéo**.

Etant donné que la majorité des serveurs sont de marques **Dell, HP, IBM** il n'y aura généralement **pas de soucis** car ces constructeurs supportent ou **contribuent aux développements GNU/Linux** et proposent souvent leur code sources aux mainteneurs du noyau Linux.

Cependant la **multiplication exponentielle des serveurs est un souci majeur en terme** de :

- Sauvegarde de l'environnement (recyclage, consommation électrique, etc.),
- Coût de maintenance,
- Coût de supervision,
- Visibilité du parc,
- Flexibilité de mise en œuvre,
- Coût financier.

De **nombreuses études et audits convergent** tous vers un constat simple : le **matériel** est largement **sous-employé** lorsque l'on parle de serveurs physiques.
Les audits font donc plus état de **pics de charges éparses** que de charges continues. Depuis quelques années une réponse de l'industrie informatique a été apportée aux entreprises ayant besoin de cette flexibilité et d'une architecture matérielle à géométrie variable : **la virtualisation**.

Note : À ce sujet un « proof of concept » incluant clustering et virtualisation a été mis en œuvre en 2004 par Joe DE BAER de Novell Inc., nous y reviendrons plus tard.

La politique actuelle du Ministère va également dans ce sens : **les ressources vont être progressivement mutualisées**. Concrètement nous nous dirigeons vers des **Datacenter** où sera **virtualisée la majorité des serveurs physique actuels**.

Bien entendu cela sous-entend d'avoir des **liens réseaux irréprochables tant en terme de dimensionnement, de qualité de service que de haute disponibilité, des salles machines bien dimensionnées etc**.

Besoin d'aide ?

Savoir être autonome avec GNU/Linux …

Sous GNU/Linux, où tout est « script » et « code source», un bon administrateur système n'est pas celui qui connaît par cœur chaque procédure ou ligne de commande : il y en aurait tellement à apprendre !

En effet il existe une **infinité de situation critique à solutionner en environnement de production**, c'est rarement avec une recette toute faite que l'on résout un problème. Même si la « **cellule études et méthodes »** a bien fait son travail en **mettant à disposition des « administrateurs et exploitants » des procédures d'exploitation** il subsistera toujours des cas ou ce sera aux **administrateurs d'être force de proposition** pour solutionner l'incident.

Il est donc important de **comprendre le concept** qui se cache derrière une technologie, une brique système.
Connaître par cœur les lignes de commande qui permettent d'étendre un volume logique en le redimensionnant à la volée n'est pas la priorité : internet ou une base de connaissances quelconque sont là pour ça.
Par contre **savoir que cela est faisable** et surtout **comprendre le cadre d'utilisation de chaque commande** qui va être saisie constitue une **connaissance bien plus précieuse**.

N'oubliez jamais que l'on fait majoritairement appel aux services d'un administrateur système lorsqu'il faut effectuer des **manipulations délicates ou dans le cadre d'un sinistre bloquant la production** : alors **soyez calmes et posés** pour aborder ce genre de situation.

En cas d'oubli ou lorsque vous devez trouver une information technique voici les principales ressources qui sont à votre disposition :

Localement sur votre distribution GNU/Linux fraîchement installée :

- Les pages de manuels : **man,**
- La commande suivi de « **--help** »,
- La commande « **help** » suivi de la commande pour laquelle on désire une aide,
- ici : **doc/HOWTO**(CD-ROM), **/usr/share/doc/HTML**, **README**, INSTALL
- **/usr/share/doc/Deployment_Guide-fr-FR-5.2/** (**Centos**)

A distance, **sur internet**, il y a généralement une page qui répond dans 90% des cas à votre problématique, alors il ne faut pas hésiter à aller y faire un tour. N'appliquez pas bêtement un tutoriel glané sur internet : comprenez-le d'abord !

Pour ce faire :

- **Google** « is your friend »
- L'arrogant « **R**ead **T**he **F**…ing **M**anual » : ce satané manuel
- Les **sites officiels** RedHat©, Centos, Fedora
  http://docs.redhat.com/docs/fr-FR/index.html
  http://www.centos.org/docs/5/
- Les sites et forums communautaires
  http://lea-linux.org/, http://tldp.org, http://linuxdocs.org, http://linuxfr.org etc.

Installation de Centos

Anatomie d'une installation

Pour ce cours nous allons monter notre architecture de test dans un environnement totalement virtualisé. Nous utiliserons l'hyperviseur de type 2 « Virtual Box » de Sun/Oracle©.

Il y a des tâches de préparation de votre installation qui sont nécessaires afin que cette dernière soit un succès. Il faut donc :

- S'assurer que **l'architecture matérielle** (x86, ARM etc.-32 bits ou 64 bits) de la **distribution** que vous vous apprêtez à installer **corresponde** bien au **processeur** dont vous disposez.

  Note : le choix du **64 bits n'est pas une obligation** à ce jour (2011) car il pose encore pas mal de soucis d'incompatibilité, pensez-y et suivez les préconisations de l'intégrateur de la distribution à ce sujet.

- Disposer d'un **média d'installation fiable** (réseau opérationnel ou DVD graver à partir d'une image disque ISO contrôlée sans erreur via **MD5SUM**),

- Être en possession des **paramètres réseaux de votre futur serveur** (nom, IP, passerelle, DNS etc.). Bref, dès **maintenant constituez votre ficher serveur**, en respectant les **règles de nommage machine** de votre site ainsi que votre **plan d'adressage IP**.

- Avoir pris connaissance des caractéristiques techniques du matériel sur lequel vous projetez d'installer GNU/Linux (cpu, chipset, ram, disque RAID etc. le trio **x86 / 6GoHD / 256Mo** étant le minimum)

Une fois votre fiche serveur renseignée, et les prés requis validés, vous pouvez démarrer votre serveur (ou machine virtuelle) avec un media d'installation valide.

Vous pouvez installer GN/Linux de trois manières différentes :

- A l'aide de **DVD** (ou CD),

- Via des **sources** situées sur le **réseau** (HTTP, NFS, FTP),

- Ou en mode automatique à l'aide du fichier de réponse (Kick START) à l'aide d'une des deux méthodes citées ci-dessus.

  Le fichier « **anaconda-ks.cfg** » (Kick START) est automatiquement créé après une installation réussie :

```
/root/anaconda-ks.cfg
```

A propos des pilotes importés ou compilés

La compilation de module (pilote) apporte certes une solution à une éventuelle incompatibilité matérielle, toutefois il faut bien garder à l'esprit **qu'à chaque mise à jour du noyau il faudra recommencer la compilation du module**.

Voici les différentes phases d'une installation Centos :

Phase 1 : Installation du système de base

- Interface **Graphique** (**Texte** possible si soucis avec carte vidéo)
- Choix **Langue/Clavier** (français/français-latin1)
- Stratégie de partitionnement : prenez un **disque dur complet et choisissez le partitionnement par défaut** tout en gardant la possibilité d'« Examiner et de modifier la structure de partitionnement » pour vérifier que vous ne supprimez pas des données importantes.

Avant toutes **interventions** sur un **système de fichiers** un administrateur consciencieux aura pris soin de passer voir la cellule exploitation pour s'assurer que les **sauvegardes de la veille** sont **disponibles** et **utilisables**.

Bien que nous allons y revenir plus loin, dès le départ cassons certains mythes concernant le partitionnement.

Les choix techniques faits par défaut sur des distributions professionnel (Ex. Centos ou RedHat©) ont été fait par des personnels hautement qualifiés, ils sont souvent **cohérents et justifiés dans 99% des cas**. Il faut avoir de solides raisons pour les remettre en cause et savoir à quoi l'on s'engage.

Par exemple vouloir systématiquement disposer de plusieurs points de montage, partitions etc. (pour pouvoir gagner en souplesse les volumes exploités) est dépassé : avec **LVM nous conserverons ces possibilités et irons beaucoup plus loin en termes de souplesse** !
Certains défendent les **performances disques** pour justifier les points de montage : c'est **faux**, sur un serveur vous travaillez en RAID matériel.

Mais attention, LVM ne veut pas dire **sécurisation des données, le RAID** est fait pour ça.

- A ce stade vous avez indiqué à **DiskDruid** comment vous souhaitiez organiser votre sous-système disque,
- Paramétrage du **chargeur de démarrage** (Boot Loader) GRUB, **par défaut**,
- **Nom d'hôte** et **environnement réseau en mode statique (pas de DHCP sur un serveur)**, ici vous devez vous servir de **votre fiche serveur**,
- Choisissez votre **région géographique (Europe/Paris),**
- Choisissez un **mot de passe « root » complexe**,

- Vous devez choisir les **familles de paquetages à installer**. Nous installons un serveur, ne choisissez pas de paquetages superflus (ces derniers augmentent la surface de vulnérabilité de votre système). Donc :
  - Server,
  - Server – GUI uniquement.
- **ATTENTION : après cet écran tous les choix que vous avez faits vont être inscrits sur les disques durs,**
- Vous pouvez prendre un café. En effet la phase 1 de l'installation prend une bonne dizaine de mn. (Veuillez noter l'existence des fichiers /root/anaconda-ks.cfg, et /root/install.log)
- **Pensez à éjecter le DVD** d'installation à l'invite.

Phase 2: Configuration initiale (first boot)

Nous allons maintenant terminer l'installation de Centos en répondant à quelques questions concernant :

- La sécurité,
- Le débogage,
- La gestion de l'heure,
- La création d'un utilisateur pour l'administration de premier niveau et l'exploitation quotidienne : **exploit**,
- Finalisation.

Le pare-feu : En production il faut activer les réglages par défaut c'est-à-dire « activé » avec uniquement les ports TCP/UDP/ICMP (services réseaux) que vous proposerez à vos clients.

En cours désactivez-le par défaut.

SE Linux étant un module à part il est conseillé de le **désactiver** si l'on ne le maîtrise pas.

KDump (Kernel Dump) uniquement en environnement de test**, ici on le laisse désactivé**. Il permet d'analyser la cause d'un crash du noyau (chose très très rare sur un noyau non modifié).

La gestion de l'heure est importante notamment pour les fichiers de journaux (logs), de ce fait vous devez faire pointer votre serveur vers un serveur de temps officiel. Le système utilisera le **service et protocole NTP** pour assurer cette tâche.

Vous devez maintenant créer un premier compte utilisateur.

Il est très important de ne **pas travailler systématiquement avec** le compte **« root ».**
La mise en place d'une production robuste et performante repose sur le respect de cette règle.

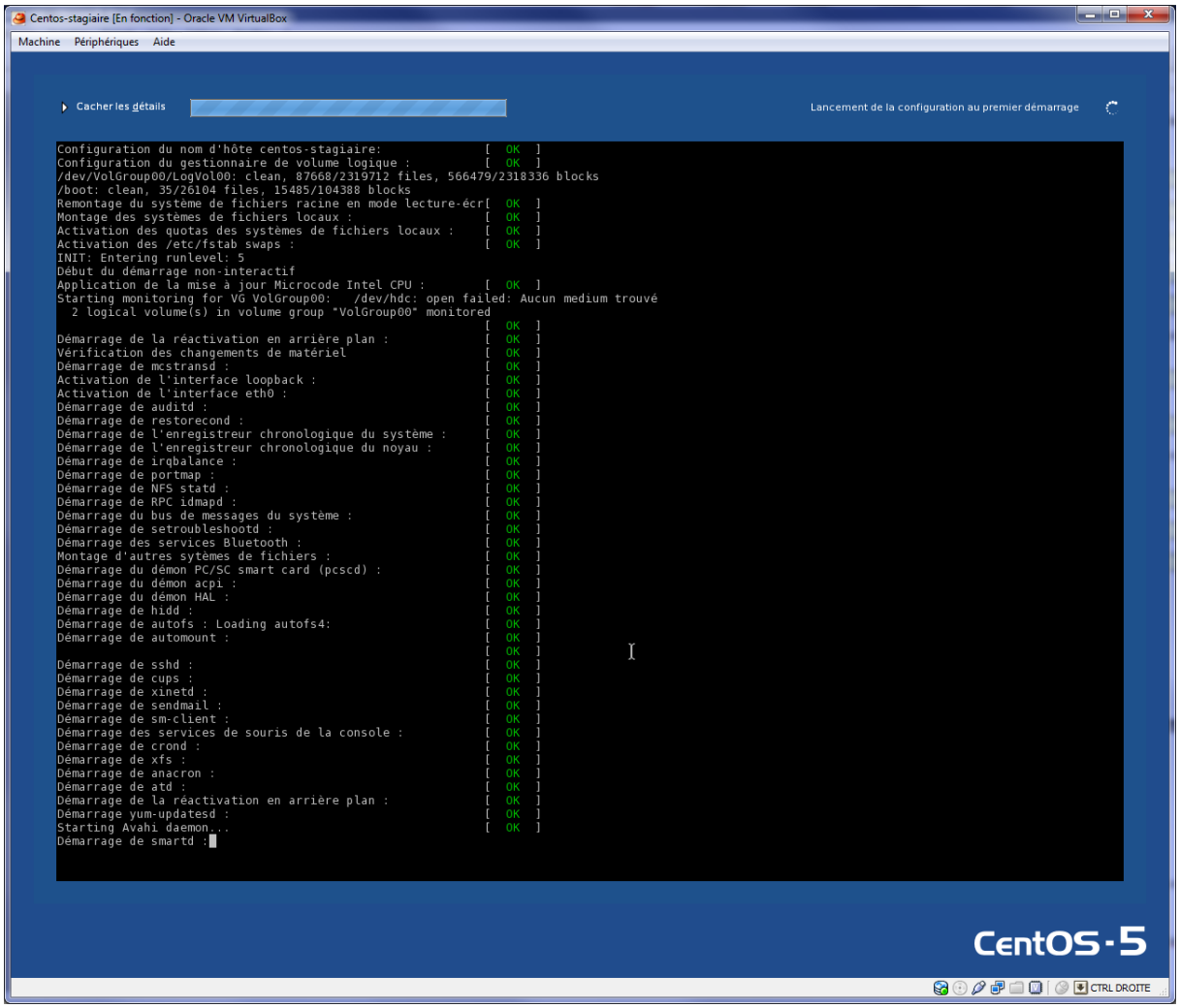
Pour ce faire il est vivement recommandé de traiter **l'exploitation quotidienne** du SI avec un **compte générique.** Ce dernier doit avoir les droits d'un utilisateur standard : créez le compte **« exploit »**.

Par exemple une sauvegarde MySQL, Oracle, une livraison J2EE, ou une mise en production ne nécessitent en aucun cas des droits administrateurs maximaux (avec « root ») si vous avez respecté l'architecture d'une distribution comme une RedHat© (Centos).

Vérification d'une installation

L'installation est terminée, maintenant nous allons vérifier que tout s'est déroulé correctement afin de disposer d'un système dans un état connu et stable.
Tout d'abord, au démarrage en mode graphique (ou texte), en cliquant sur **« Afficher les détails »**, vous ne devez voir apparaître aucun message d'erreur, dans le cas contraire il faudra investiguer.

Tous les services du processus de démarrage doivent être marqués **[OK]** comme suit :



Note : Cet écran de démarrage des services est également consigné dans le fichier **« /var/log/boot.log »**

A propos des répertoires `/proc` et `/sys`.

Ils contiennent un **système de fichiers virtuel** qui **« documente » à la volée le noyau et les différents processus du système**. Retenez juste que certains fichiers contenus dans ces répertoires nous permettront d'obtenir des informations précieuses sur l'état du système : le modèle et la fréquence du processeur, la mémoire vive disponible et la quantité de mémoire utilisée, et beaucoup d'autres choses encore. Quant au « système de fichiers virtuel », on peut le considérer comme un système de fichiers volatile, dont **il ne reste pas la moindre trace dès que vous éteignez la machine**. Nous les étudierons plus tard.

Dans un premier temps vous pouvez **contrôler quelques reflets directs de l'activité du noyau** dans le répertoire **/proc**, en voici quelques exemples importants :

Type de processeur reconnu

- cat /proc/cpuinfo

Mémoire reconnue

- cat /proc/meminfo

Disques durs reconnus

- cat /proc/scsi/scsi

Liste des partitions montées

- cat /proc/partitions

Ensuite il est intéressant de regarder la **prise en charge de votre matériel** en concordance à votre check-list de votre matériel faite lors de la préparation de l'installation :

Liste du matériel reconnu

- cat /etc/sysconfig/hwconf

Liste des périphériques branchés sur les slots PCi(e x)

- lspci

Liste des modules (pilotes) en cours d'utilisation

- lsmod

Enfin il y a les messages produits par **le système, le noyau et la séquence de démarrage** de GNU/Linux. Ils sont respectivement contenus dans :

- cat /var/log/messages

- cat /var/log/dmesg

- ~~cat /var/log/boot.log~~

Enfin vous pouvez consulter le fichier suivant pour vérifier l'installation des composants que vous avez sélectionnés :

- /root/install.log

Si vous désirez aller plus loin vous pouvez également parcourir l'arborescence, **/sys**.

Nous verrons par la suite comment nous servir de ces journaux systèmes (log) pour débusquer rapidement un souci sous GNU/Linux.

Les commandes et logiciels indispensables

Dans ce chapitre nous allons faire un bref rappel des **commandes utiles et indispensables** sous GNU/Linux puis faire un inventaire de **quelques outils qui pourront vous faciliter l'administration** (sous licence GPL) du système en étant **sous environnement Microsoft Windows**.
Si vous avez l'opportunité de pouvoir disposer d'un **poste dédié à l'administration** GNU/Linux installé précisément **sous Linux**, dans ce cas la plupart des **outils sont nativement présents**.

Bien que la majeure partie des distributions Linux proposent un mode graphique il est inconcevable pour un administrateur système GNU/Linux de travailler efficacement uniquement avec le mode graphique.

Information : un système BSD/GNU permet néanmoins d'administrer un serveur entièrement en **mode graphique, il s'agit de Mac OS X serveur** d'Apple Inc.

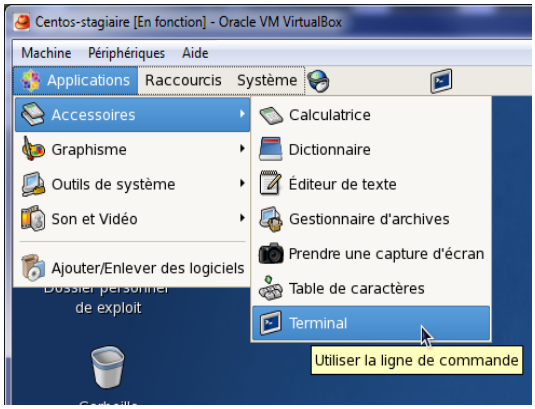
En fait le mode graphique ne permet pas d'assurer toutes les tâches d'administration du système. Pire, dans certains cas de figure une modification en **mode graphique pourra altérer vos réglages** en place dans les **fichiers de configurations** mettant votre système dans un état inconnu et instable (Exemple : configuration de Bind sous Centos).

Alors prenez dès le début de bonnes habitudes en **travaillant exclusivement en mode texte et sécurisé. Pour cela connectez-vous à GNU/Linux avec un client SSH reconnu**.

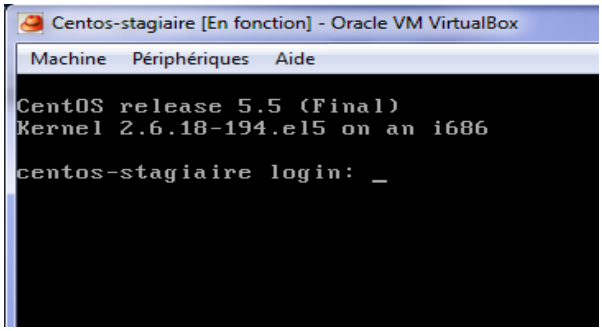
Ce dernier vous donnera accès à GNU/Linux de façon fiable et sécurisé. De plus vous vous apercevrez rapidement que vous allez gagner en rigueur d'administration : la ligne de commande adéquate ne s'improvise pas, on doit savoir ce que l'on fait.
Alors qu'avec un ensemble de clic on est toujours tenté de tâtonner pour arriver au résultat escompté.

Les commandes

Lorsque l'on lance l'application **« Terminal »** le premier contact avec le système est l'invite de commande Shell BASh. Par défaut en mode texte vous démarrez sous un terminal (TTY) sous Shell BASh :



OU



Il s'agit d'un interpréteur de commande en mode texte. Il en existe une dizaine. Nous nous concentrerons sur **BASh (Bourne Again Shell)** car ce dernier est présent sur la quasi-totalité des distributions GNU/Linux ainsi que sur la plupart des UNIX propriétaires.
Il vous permet de piloter totalement votre système.

- Veuillez noter que, par défaut, **Centos** propose **6 terminaux en mode texte** (TTY) qui sont déjà lancés au démarrage de votre système.
Ils sont **accessibles dès la fenêtre de connexion en mode graphique** ou texte via la combinaison de touche suivante :

Cette astuce vous permettra de piloter votre système même si il y a un souci avec le mode graphique ou le blocage d'un TTY.

- Pour sortir d'un programme ou d'une commande :

- Servez-vous de la touche suivante pour faire de l'auto-complétion de commandes ou de répertoires.

Les touches « flèches » du pavé numérique permettent de rappeler les commandes saisies précédemment (elles exploitent la commande **« history »)** :

La syntaxe générale d'une commande est la suivante :

- Commande [paramètres] [arguments]

Vous pouvez chaîner vos commandes avec : 

- Commande_1 **;** Commande_2 **;** Commande_3

Vous pouvez rediriger le(s) résultat(s) d'une commande dans le(s) paramètres d'entrée d'une autre commande avec : 

- Commande_1 | Commande_2

Vous pouvez lancer votre commande en arrière-plan avec : 

- Commande **&**

- La commande « **type** » vous permet de différencier une **commande interne ou externe** au Shell BASh :

```
[root@centos-stagiaire ~]# type ll
ll is aliased to `ls -l --color=tty'
[root@centos-stagiaire ~]# type pwd
pwd is a shell builtin
[root@centos-stagiaire ~]# type date
date is /bin/date
```

Une fonctionnalité va nous permettre de déboguer ou mettre au point la configuration d'un service réseau ou tout simplement de visualiser le comportement du système d'exploitation en temps réel : il s'agit de l'exploitation des fichiers de journaux (log).

Les fichiers de journalisation pour chaque service réseaux ou du système sont mis à jour en temps réel. Leur présence est précieuse car vous pourrez savoir exactement ce qui se passe en cas de souci, ils se trouvent sous :

- /var/log/

Les plus intéressants sont les journaux du système, des accès et du noyau au démarrage, voici comment les visualiser :

- tail -f -n 50 /var/log/messages
- tail -f -n 50 /var/log/secure
- tail -f -n 50 /var/log/dmesg

Astuce 1 :

Dans vos scripts pensez à **retourner** systématiquement **l'état de sortie de votre script : succès, échec** etc. Dans le cadre d'un **environnement** de production vous verrez que ce **détail est capital**. De plus utilisez la commande **« logger »** pour journaliser les actions de vos scripts.

Astuce 2 :

L'ouvrage suivant est vraiment une bible des problématiques rencontrées par 90% des administrateurs système GNU/Linux : **bash Cookbook (Oreilly), Livre de Recette BASh**

MEMENTO COMMANDES UTILES

Gérer fichiers et répertoires

Créer un répertoire (**m**a**k**e **dir**ectory):
mkdir rép

Créer des répertoires imbriqués:
mkdir -p rép1/rép2

Changer de répertoire (**c**hange **d**ir):
cd nouveau_rép
cd .. (répertoire parent)
cd - (répertoire précédent)
cd (répertoire personnel)
cd ~bill (répertoire personnel de bill)

Afficher répertoire courant (**p**rint **w**orking **d**ir):
pwd

Copier un fichier vers un autre:
cp fichier_orig fichier_dest

Copier des fichiers dans un répertoire:
cp fichier1 fichier2 rép

Copier des répertoires entiers (**r**ecursively):
cp -r rép_orig rép_dest
rsync -a rép_orig/ rép_dest/

Créer un **lie**n symbolique:
ln -s fichier_orig lien

Renommer un fichier, lien ou répertoire:
mv fichier_orig fichier_dest

Supprimer (**rem**ove) des fichiers ou des liens:
rm fichier1 fichier2

Supprimer un répertoire (**r**emove **d**ir):
rmdir rép

Supprimer un répertoire non vide (**f**orce):
rm -rf rép

Afficher les noms de fichiers

Énumérer (**li**s**t**) les fichiers «ordinaires» (ne commençant pas par .) dans le rép. courant:
ls

Afficher une liste détaillée (**l**ong):
ls -l

Énumérer tous (**a**ll) les fichiers dans le rép. courant (y compris ceux commençant par .):
ls -a

Trier par date (**t**ime) (d'abord les plus récents):
ls -t

Trier par taille (**s**ize) (d'abord les plus gros)
ls -S

Affichier en inversant (**r**everse) l'ordre de tri:
ls -r

Affichage long, fichiers plus récents en dernier:
ls -ltr

Afficher le contenu des fichiers

Afficher bout à bout le contenu de fichiers:
cat fichier1 fichier2 (con**cat**enate)

Afficher le contenu de plusieurs fichiers (en faisant une pause à chaque page):
more fichier1 fichier2
less fichier1 fichier2 (plus de possibilités)

Afficher les 10 premières lignes d'un fichier:
head -10 fichier

Afficher les 10 dernières lignes d'un fichier:
tail -10 fichier

Modèles de noms de fichiers

Afficher bout à bout tous les fichiers ordinaires:
cat *

Afficher bout à bout tous les fichiers "cachés":
cat .*

Afficher tous les fichiers finissant par .log:
cat *.log

Les fichiers ordinaires avec bug dans leur nom:
ls *bug*

Lister tous les fichiers ordinaires finissant par . suivi d'un seul caractère:
ls *.?

Gérer le contenu des fichiers

N'afficher que les lignes d'un fichier contenant une sous-chaîne donnée:
grep sous-chaîne fichier

Recherche **i**nsensible aux majusc. / minusc.:
grep -i sous-chaîne fichier

Afficher toutes les lignes sauf celles qui contiennent une sous-chaîne:
grep -v sous-chaîne fichier

Recherche à travers tous les fichiers d'un rép.:
grep -r sous-chaîne rép

Trier les lignes d'un fichier:
sort fichier

Trier, n'afficher qu'1 fois les lignes identiques:
sort -u fichier (**u**nique)

Droits d'accès aux fichiers

Ajouter droits en écriture au propriétaire:
chmod u+w fichier (**u**ser, **w**rite)

Ajouter droits en lecture au groupe du fichier:
chmod g+r fichier (**r**ead)

Ajouter droits d'**ex**écution aux autres utilisat.:
chmod o+x fichier

Ajouter droits lecture / écriture à tous (**a**ll):
chmod a+rw fichier

Rendre fich. exécutables exécutables par tous:
chmod a+rX *

Rendre le répertoire et tous les fichiers qu'il contient accessibles par tous les utilisateurs:
chmod -R a+rX rép (**r**ecursive)

Comparer: fichiers, répertoires

Comparer 2 fichiers:
diff fichier1 fichier2

Comparer 2 fichiers (en mode graphique):
gvimdiff fichier1 fichier2
tkdiff fichier1 fichier2
kompare fichier1 fichier2

Comparer 2 répertoires:
diff -r rép1 rép2

Rechercher des fichiers

Rechercher tous les fichiers dans le répertoire courant (.) avec log dans leur nom:
find . -name "*log*"

Trouver tous les fichiers en .pdf dans rép et exécuter une commande sur chacun:
find . -name "*.pdf" -exec xpdf {} ';'

Recherche rapide dans tout système: (utilise un index, les fichiers récents peuvent manquer):
locate "*bar*"

Rediriger sortie de commande

Rediriger sortie de commande vers un fichier:
ls *.png > fichiers_image

Ajouter la sortie d'une commande à un fichier:
ls *.jpg >> fichiers_image

Rediriger la sortie d'une commande vers l'entrée d'une autre:
cat *.log | grep erreur

Contrôle de tâches

Afficher tous les processus exécutés:
ps -ef

Classement en direct des processus (P, M, T: trie par utilisation Processeur, Mémoire ou Temps):
top

Envoyer un signal d'arrêt à un processus:
kill <pid> (numéro indiqué par ps)

Faire tuer un processus par le système:
kill -9 <pid>

Tuer tous processus que l'on a le droit de tuer:
kill -9 -1

Tuer une application en mode graphique:
xkill (cliquer sur la fenêtre du programme)

Taille de fichiers et partitions

Afficher l'espace total occupé sur le disque par des fichiers ou des répertoires (**d**isk **u**sage)
du -sh rép1 rép2 fichier1 fichier2

Nombre de caractères, mots et lignes:
wc fichier (**w**ord **c**ount)

Afficher la taille, l'espace total et l'espace libre dans la partition courante:
df -h .

Afficher cette info pour toutes les partitions:
df -h

Compresser

Compresser un fichier:
bzip2 fichier (meilleur taux de compression)
gzip fichier

Décompresser un fichier:
bunzip2 fichier.bz2
gunzip fichier.gz

Manipuler des archives

Créer une archive compressée (**t**ape **ar**chive)
tar jcvf archive.tar.bz2 rép/ (le mieux!)
tar zcvf archive.tar.gz rép/

Tester (lister) une archive compressée:
tar jtvf archive.tar.bz2
tar ztvf archive.tar.gz

E**x**traire les fichiers d'une archive compressée:
tar jxvf archive.tar.bz2
tar zxvf archive.tar.gz

Options de tar:
c: **c**réer
t: **t**ester / lister
x: e**x**traire
j: (dé)compression bzip2 à la volée
z: (dé)compression gzip à la volée

Manipuler des archives zip:
zip -r archive.zip <files> (créer)
unzip -t archive.zip (tester / lister)
unzip archive.zip (extraire)

Imprimer

Envoyer fichiers PostScript ou texte sur queue:
lpr -Pqueue f1.ps f2.txt (**l**ocal **pr**inter)

Lister les tâches d'impression dans queue:
lpq -Pqueue

Annuler une tâche d'impression dans queue:
cancel 123 queue

Imprimer un fichier PDF:
pdf2ps doc.pdf
lpr doc.ps

Visualiser un fichier PostScript:
ps2pdf doc.ps
xpdf doc.pdf

Gestion des utilisateurs

Afficher les utilisateurs connectés au système:
who

Afficher sous quel utilisateur je suis connecté:
whoami

Afficher à quel groupe appartient utilisateur:
groups utilisateur

Afficher plus d'informations sur utilisateur:
finger utilisateur

Passer à l'utilisateur hulk:
su - hulk

Passer au super-utilisateur (root):
su -
su (sans changer de rép. ni d'environnement)

Gérer le temps

Attendre 60 secondes:
sleep 60

Afficher la date actuelle:
date

Mesurer le temps pris par une commande:
time trouve_prince_charmant -beau -riche

Aide sur les commandes

Aide de base (pour la plupart des commandes):
grep --help

Voir le manuel complet d'une commande:
man grep

Commandes diverses

Calculatrice simple en ligne de commande:
bc -l (**b**asic **c**alculator)

Bases d'administration système

Changer le propriétaire et le groupe d'un répertoire et tout ce qu'il contient:
chown -R nouvproprio:nouvgroupe rép

Redémarrer la machine dans 5 minutes:
shutdown -r +5

Éteindre la machine immédiatement:
shutdown -h now

Afficher toutes les interface réseau disponibles:
ifconfig -a

Assigner une adresse IP à une interface réseau:
ifconfig eth0 207.46.130.108

Désactiver une interface réseau:
ifconfig eth0 down

Définir une passerelle par défaut pour les paquets vers des machines hors du réseau:
route add default gw 192.168.0.1

Supprimer la route par défaut:
route del default

Tester la connexion réseau avec une machine:
ping 207.46.130.108

Créer ou supprimer des partitions sur le premier disque IDE:
fdisk /dev/hda1

Créer (formater) un système de fichiers ext3:
mkfs.ext3 /dev/hda1

Créer (formater) un système de fichiers FAT32:
mkfs.vfat -v -F 32 /dev/hda2

Monter une partition formatée:
mkdir /mnt/cleusb (nécessaire une seule fois)
mount /dev/uba1 /mnt/cleusb

Monter image de système de fichiers (loopback)
mount -o loop initrd.img /mnt/initrd

Démonter un système de fichiers:
umount /mnt/cleusb

Sources, traductions, mises à jour et détails sur les commandes disponibles avec nos supports de formation libres: http://free-electrons.com/training/intro_unix_linux

Remerciements à Michel Blanc, Hermann J. Beckers et Thierry Grellier.

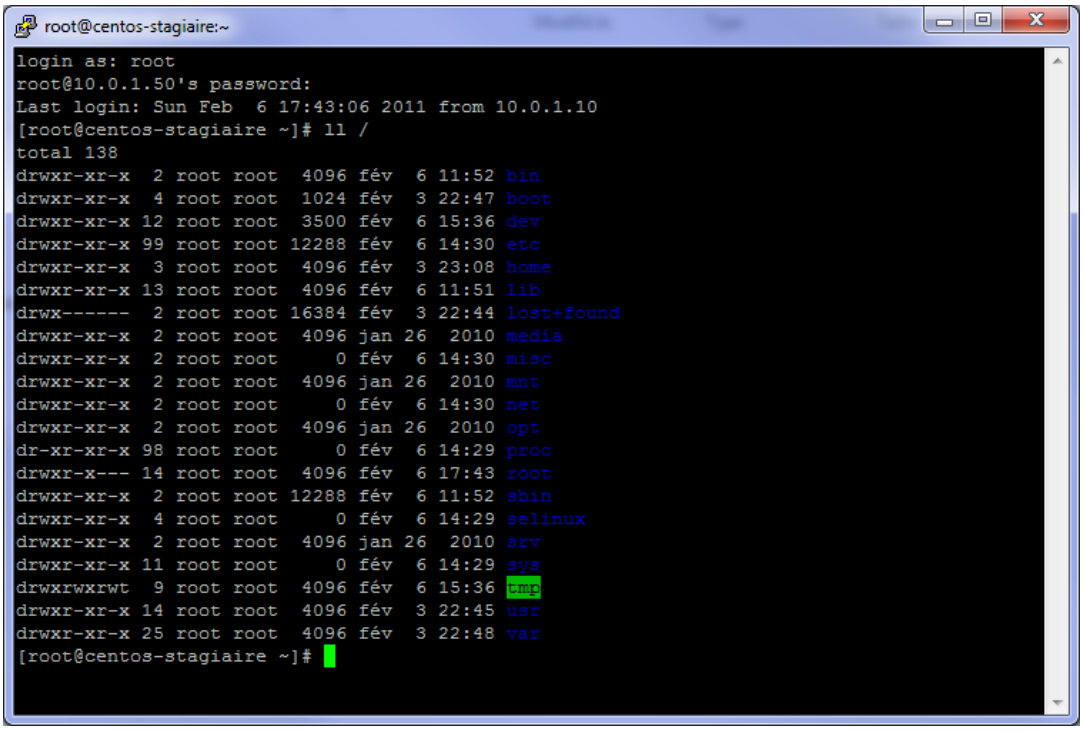
Les logiciels

Voici quelques outils qui vous faciliteront l'administration (tous sous licence GPL) du système GNU/Linux tout en étant sous environnement Microsoft Windows.

Les deux outils **PuTTy** et **PuTTYcm** vous permettront de :

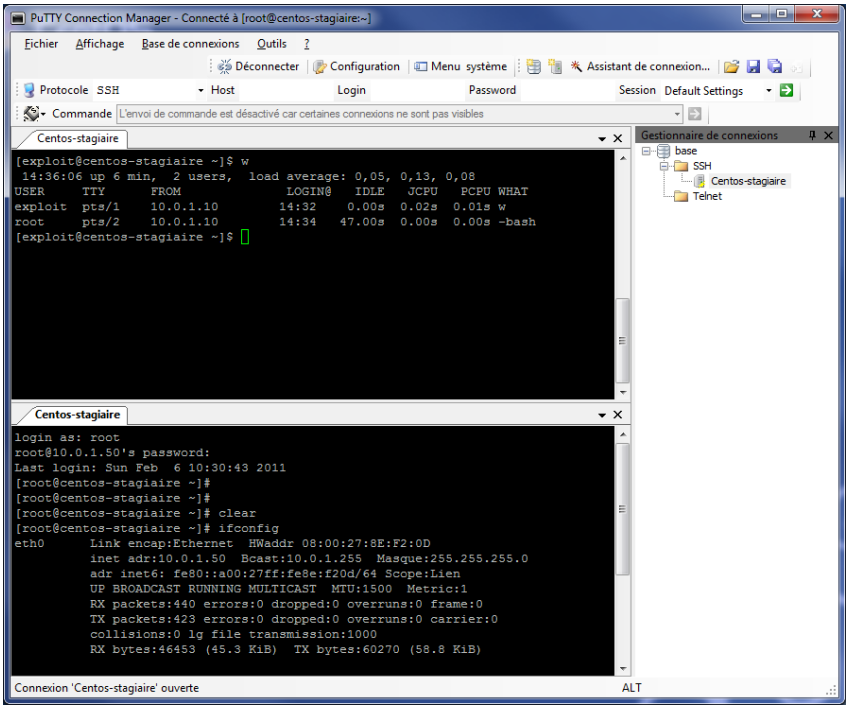
- **gérer à distance vos serveurs** en mode texte de façon **sécurisé**,
- gérer plusieurs serveurs **en simultané**,
- déboguer un script,
- **gérer vos connexions** à GNU/Linux **sur un parc conséquent**,
- naviguer entre vos différentes sessions **SSH via des onglets**,
- **déporter l'affichage X-Windows** de votre serveur,
- plein d'autres choses …

PuTTY permet de se connecter à votre distribution GNU/Linux de façon **sécurisé** contrairement à « **telnet » qui est à bannir**.
Il exploite le **protocole SSH** pour se connecter au service OpenSSH démarré (par défaut) sur GNU/Linux.

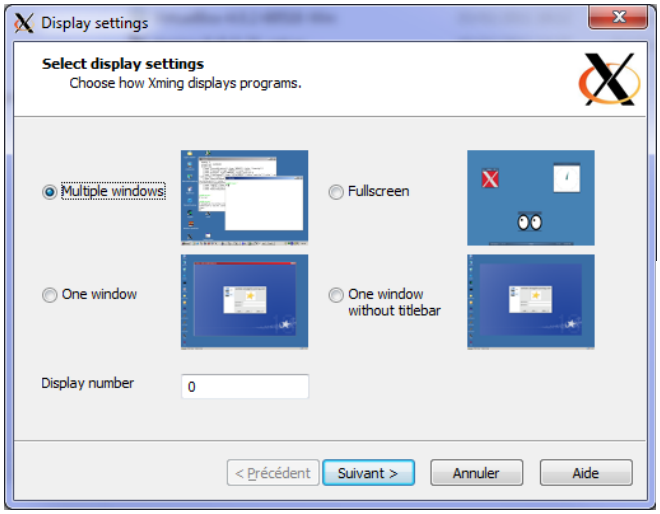


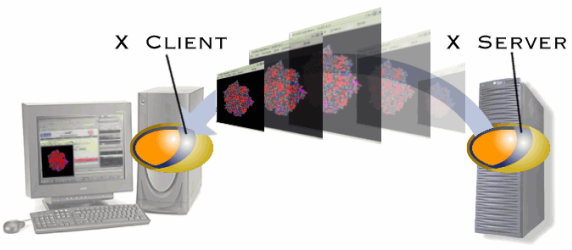
Astuce : Il est intéressant de choisir l'encodage **UTF-8** pour éviter les problèmes d'accents illisibles : allez sous « **Windows\Translation** ».
Mais aussi d'activer le déport d'affichage avec : « **Connection\SSH\X11 » en cochant « X11 Forwarding** ».
Puis sauvegardez vos modifications en tant que profile par défaut.

Vous pouvez ensuite installer une surcouche confortable pour l'utilisation de PuTTY : **PuTTY Connexion Manager** (il nécessite « **.NET RunTime** » et « **PuTTY »**) :



Enfin voici un serveur graphique X-Windows gratuit : **XMing**
Il vous permettra de profiter des **applications graphiques de GNU/Linux sur Windows**. On parle ici de **déport d'affichage via** le protocole X-Windows.

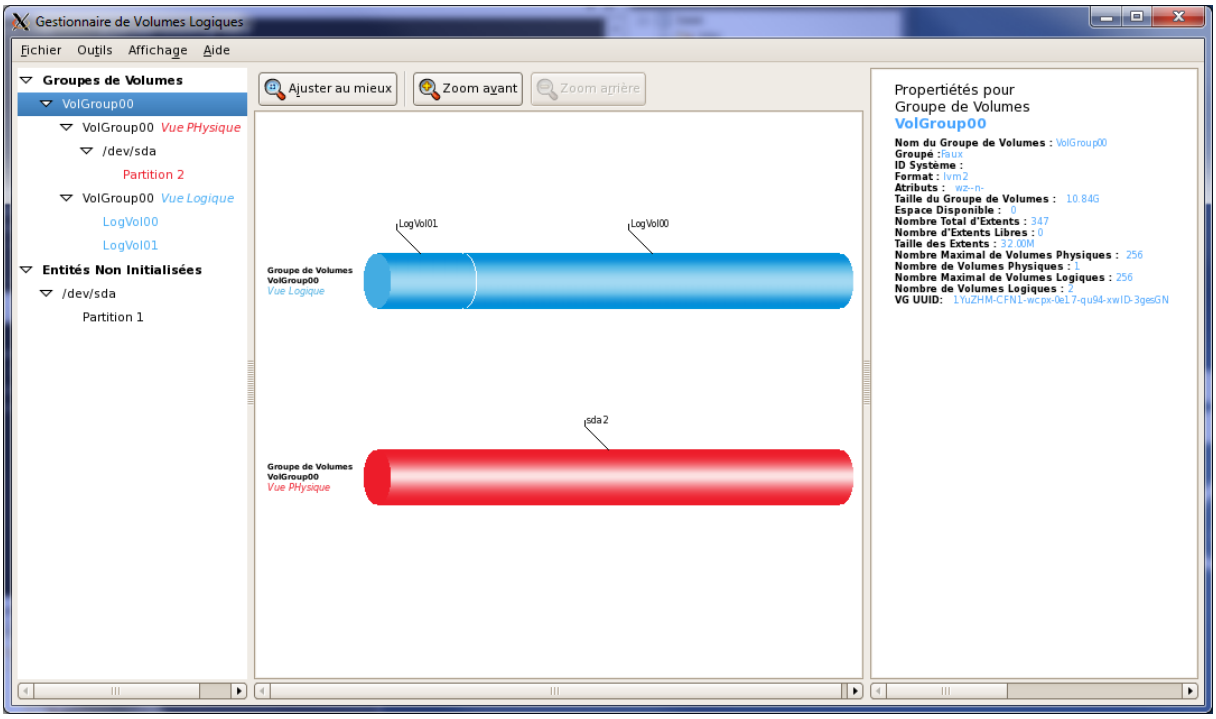




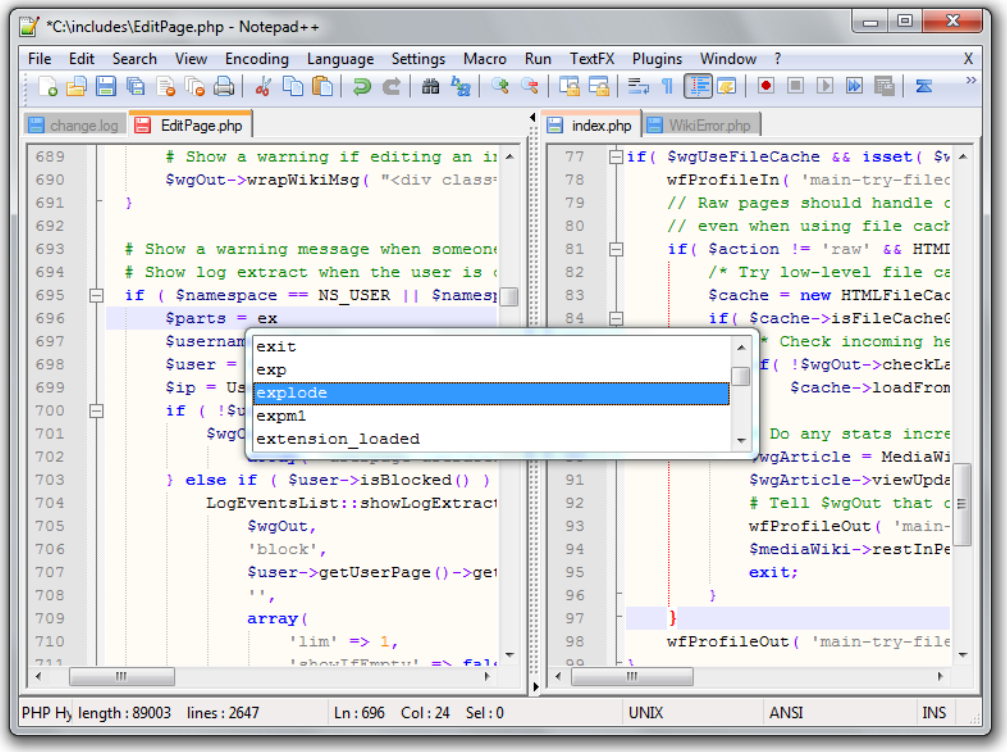
Pour en profiter, lancez simplement XMing (tout choix par défaut).
Puis une **connexion SSH avec l'option X-Forwarding** (Connection\SSH\X11)
Une fois sous la session SSH lancez une application X-Windows (gnome-kde etc …)

- system-config-lvm &

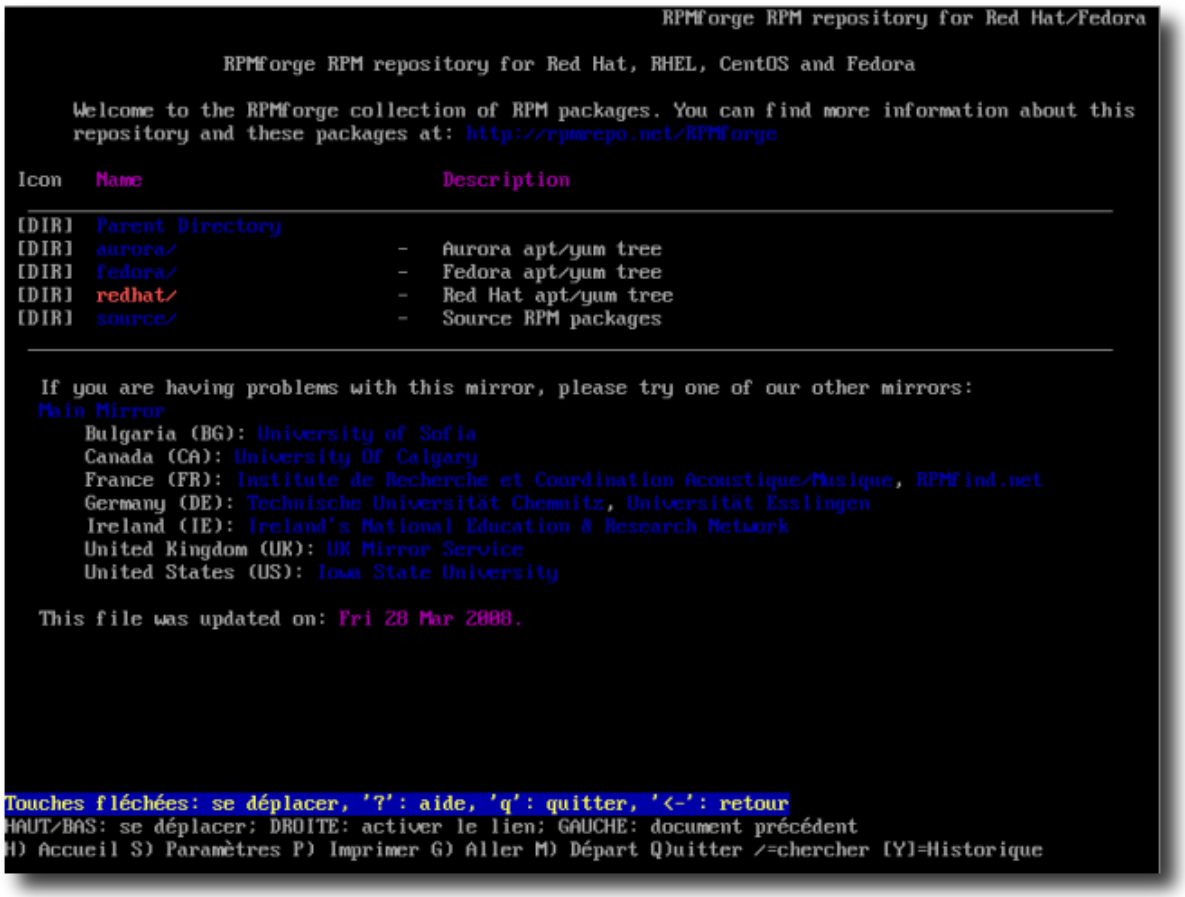
Vous obtenez sous Windows une fenêtre GNU/linux (ici Gnome LVM)



Pour éditer aisément des fichiers en mode Unix/Linux ou Windows (à partir de Windows) vous pouvez utiliser **Notepad++** :



Bien pratique quand vous êtes en mode texte voici LYNX un navigateur web.



L'installation de logiciels supplémentaires

Sous GNU/Linux vous disposez principalement de **deux manières pour installer des logiciels** supplémentaires :

- Via manipulation de **paquetages**
- Via **compilation des sources**.

La manipulation via paquetages est vivement conseillée. Pourquoi ?

Par le simple fait qu'après avoir choisi le bon paquetage pour votre distribution, version : vous êtes quasi sure d'avoir une installation adaptée à votre distribution et 100% compatible et opérationnelle. Tout au plus vous allez devoir effectuer une légère édition d'un fichier de configuration.

Si vous optez pour la solution via compilation des sources vous vous embarquez dans une avalanche d'incompatibilités possibles avec votre distribution. Sans comptez que cela suggère que vous allez devoir adapter/créer une multitude de scripts pour prendre en charge les binaires fraîchement compilés : bref un travail de « packageur/développeur ».

De plus la compilation ne donne pas toujours de meilleur performance quant au(x) binaire(s) obtenu(s) et en cas d'erreur de compilation : bon courage, votre productivité va en prendre un coup.

Si nous prenons l'exemple de vouloir installer « Apache » à partir des sources vous pouvez déjà être sur de devoir écrire ou adapter les fichiers de démarrage pour le SYSTEM V, etc.

Quelques fois la compilation s'impose : alors sachez que vous allez devoir largement mettre les mains dans le cambouis.

Voici donc la marche à suivre pour une installation par compilation :

- **Installation d'un environnement de compilation complet (gcc, make, etc.),**
- **Récupération des sources compressées sur internet,**
- **Décompression des sources,**
- **Lecture des fichiers README ou INSTALL si ils existent, à défaut la documentation officielle,**
- **./configure + directive de configuration (voir ./configure --help)**
- **make (validation des prérequis)**
- **make install**
- **la gestion de conflits de bibliothèques partagées,**
- **adaptation/intégration des scripts pour votre distribution.**

La tâche principale d'un administrateur système étant avant tout … d'administrer un système d'exploitation et non de le redévelopper les choses s'imposent d'elles-mêmes : les paquetages.

Un paquetage est un regroupement de fichiers (binaire, sources, texte etc.) et de règles dans un seul fichier. De plus le RPM (RedHat Package Manager) est utilisé par LSB (Linux Standard Base).

Un paquetage est spécifique à :
- Une distribution (Centos, Suse Novell, Debian, Ubuntu, Mandriva etc.),
- Une version de distribution (Centos 5.1, 5.2 5.3 etc.),
- Une architecture donnée (32bits, 64bits, ARM, x86, PPC etc.) ou non (noarch)

Il en existe deux principaux types :
- RPM (RedHat Package Manager),
- DEB (DEBian).

Les règles régissant un paquetage sont les suivantes:
- Gestion des dépendances,
- Tâches pré et post installation

Un des gros inconvénients des paquetages résident dans **leur gestion et les dépendances** qu'ils ont entre eux. Nous verrons plus loin qu'il existe des solutions efficaces pour s'affranchir de ces deux contraintes

Où se procurer les paquetages pour un logiciel désiré ?
Tout d'abord il faut savoir, que pour une utilisation type serveur, 99% des paquetages indispensables sont livrés sur le DVD d'installation de votre distribution : alors inutile d'aller les récupérer sur internet.

Au cas où l'offre du DVD ne comble pas vos besoins vous pouvez récupérer vos paquetages sur internet. Voici une liste de sites incontournables :

Officiels
- http://mirror.centos.org/
- http://ftp.proxad.net/mirrors/ftp.centos.org/

Tiers
- http://rpm.pbone.net/
- http://www.rpmfind.net/
- http://www.rpm.org/
- http://pkgs.org/
- http://rpmrepo.org/RPMforge/
- http://www.linuxpackages.net/

Cependant attention, un des inconvénients **d'utiliser les paquetages RPM (.rpm) unitairement** en provenance d'internet, outre les problèmes de **virus (passage sur station blanche)**, provient du fait que vous devrez être **attentifs aux dépendances de votre paquetage par rapport à ceux déjà installés**.

Soyons honnête, cela peut parfois relever du véritable **parcours du combattant pour satisfaire toutes les dépendances de votre paquetage,** sans parler des **conflits de versions avec les paquetages installés**. Bien entendu il existe **une solution** que nous étudierons plus tard : les **gestionnaires de dépendances** qui se connectent sur des **dépôts** pour faire tout le travail à notre place : ce sont des sortes d' « App Store ».

Prenons un exemple simple, vous avez besoin du logiciel **« htop»**, vous le trouvez sur http://pkgs.org/. En lisant attentivement la page on nous signale que ce **paquetage requiert** :

Requires

- libc.so.6
- libc.so.6(GLIBC_2.0)
- libc.so.6(GLIBC_2.1)
- libc.so.6(GLIBC_2.3)
- libc.so.6(GLIBC_2.3.4)
- libc.so.6(GLIBC_2.4)
- libm.so.6
- libm.so.6(GLIBC_2.0)
- libncurses.so.5
- rpmlib(CompressedFileNames)
- rpmlib(PayloadFilesHavePrefix)
- rtld(GNU_HASH)

... si ces bibliothèques sont déjà en place sur votre système avec une version satisfaisante : alors tout va bien. Dans le cas contraire c'est le début de la pêche au paquetage…

Voyons comment manipuler les paquetages RPM (.rpm).
Il se manipule avec la commande suivante :

- rpm -option_principale[option_secondaire] [le_paquetage]

Memento de la commande RPM.

Commande	Action
rpm -i <nom_paquetage>-<version>.<arch>.rpm	Installer le paquetage
rpm -ivh <nom_paquetage>-<version>.<arch>.rpm	Installer le paquetage en visualisant la progression de l'installation
rpm -Uvh <nom_paquetage>-<version>.<arch>.rpm	Mettre à jour le paquetage et l'installer s'il n'existe pas
rpm -e <nom_paquetage>-<version>.<arch>.rpm	Désinstaller le paquetage
rpm -q <nom_paquetage>-<version>.<arch>.rpm	Vérifier si le paquetage est installé
rpm -qa | more	Lister tous les paquetages installés
rpm -qa | grep <nom_paquetage>	Vérifier si le paquetage est installé
rpm -qi <nom_paquetage>-<version>.<arch>.rpm	Information sur un paquetage installé
rpm -qpi <nom_paquetage>-<version>.<arch>.rpm	Information sur un paquetage non installé
rpm -qf /path/command	Connaître le paquetage d'origine d'un fichier
rpm -ql <nom_paquetage>-<version>.<arch>.rpm	Connaître la liste des fichiers composant un paquetage
rpm -qR <nom_paquetage>-<version>.<arch>.rpm	Connaître les dépendances d'un paquetage
rpm -V <nom_paquetage>-<version>.<arch>.rpm	Vérifie l'intégrité d'un paquetage
rpm -q -d <nom_paquetage>-<version>.<arch>.rpm	Affiche les fichiers de documentation que fournit un paquetage
rpm -Uvh ftp://<machine_reseau> /path/ <nom_paquetage>-<version>.<arch>.rpm	Installer un paquetage stocké sur un serveur ftp accessible

Note : le « p » derrière l'option d'interrogation « -q » permet de faire une interrogation entre un paquetage déjà installé et un paquetage sous forme de fichier rpm -qR ou rpm -qpR(.rpm).

La commande RPM permet non seulement d'installer, supprimer, mettre à jour un paquetage mais elle permet aussi d'interroger l'ensemble des paquetages déjà installés. Pour se faire la commande RPM s'appuie sur une base de données intégrée DBM. Cette base se trouve ici :

- /var/lib/rpm/__db*

En cas de gros soucis sur cette base vous pouvez la reconstruire en faisant comme suit :

- rm -rf /var/lib/rpm/__db*
- rpm -vv --rebuild

Cependant certaines erreurs comme les **forçages d'installation ou désinstallation de RPM** (que volontairement nous ne documenterons pas) ne pourront être réparées de cette manière. Une recherche sur internet sera alors votre seul salut.

Astuce : une sauvegarde journalière de la base RPM n'est pas un luxe.

Inutile de cacher que si l'on travaille systématiquement sur des **fichiers .rpm** l'installation de logiciels supplémentaires devient vite un **casse-tête de dépendances et de conflits de versions.**

Fort heureusement, et ce depuis longtemps, des outils permettent de s'affranchir de cette tâche improductive pour un administrateur système : **les gestionnaires de dépendances**.

Le plus puissant étant APT-GET, il est natif sur une distribution Debian. Chez RedHat (Centos, Fedora etc.) un équivalent a été développé : il s'agit de **YUM (Yellowdog Updater Modified).**

Cet outil agit en **véritable chef d'orchestre de l'installation et la mise à jour des logiciels** ainsi que de la résolution des dépendances inter-paquetages. Bien entendu toutes les notions que nous avons vues précédemment restent valides.
Appliqué à l'ensemble de votre installation GNU/Linux il va vous permettre :

- D'automatiser la recherche de logiciels installés ou installables,
- D'installer automatiquement les dépendances et gérer les conflits de versions,
- De mettre à jour tout ou partie des logiciels de votre système,
- De désinstaller des logiciels.

YUM nécessite un ou plusieurs dépôts locaux ou distants.
Un dépôt est un répertoire contenant des fichiers RPM (.rpm) indexés par des fichiers .XML compressés au format Gunzip (.xml.gz).

Il faut savoir que le DVD d'installation Centos est, par défaut, un dépôt local.

Attention : Pour pouvoir manipuler les logiciels vous devez avoir **un accès « root »**

La commande permettant de créer un dépôt YUM est :

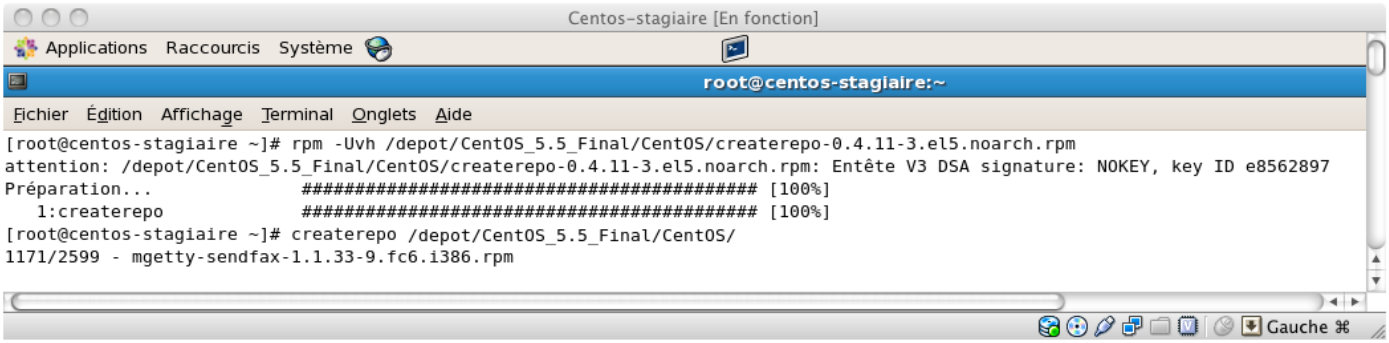
- createrepo /path/RPMs

Voici comment la mettre en œuvre.

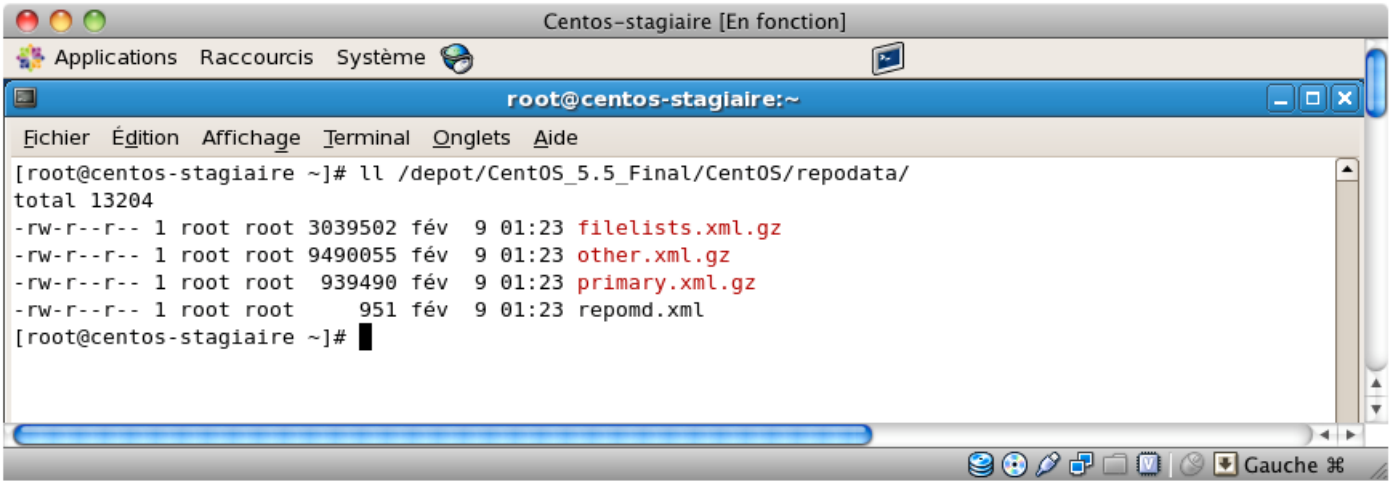
Après avoir copié le contenu entier du DVD sous « **/depot** »,

- mkdir /depot
- cp -rp /media/CentOS_5.5_Final /depot/

Vous allez installer l'outil « **createrepo** » via la commande native **« rpm -Uvh »** puis vous en servir pour créer un dépôt comme suit :



Vous disposez d'un dépôt local construit à partir du DVD officiel Centos. Vous n'aurez plus besoin du DVD pour installer un logiciel contenu sur ce dernier, vous pouvez donc l'éjecter.
Ce dépôt est défini dans le répertoire « **./repodata » du dépôt** par les fichiers XML compressés au format Gunzip **(.xml.gz)** :



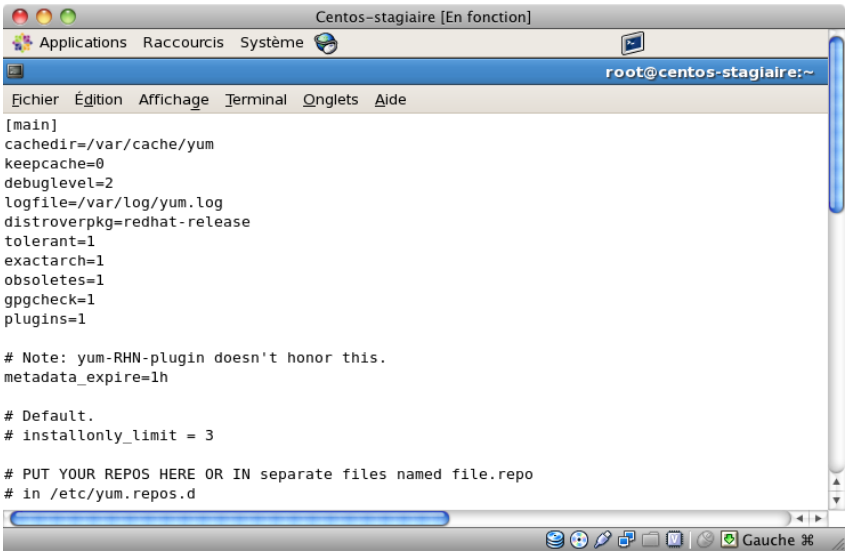
Pour utiliser ce dépôt vous devez maintenant **indiquer à YUM la définition des dépôts dans lesquels il pourra puiser** pour installer, supprimer, mettre à jour etc. vos paquetages (logiciels).

Pour ce faire vous pouvez éventuellement éditer le **fichier de configuration principal de YUM** mais il est indiqué de modifier les **fichiers de définitions des dépôts** :

- /etc/yum.conf (fichier de configuration principal YUM)
- /etc/yum.repo/*.repo (fichiers de définitions des dépôts)

Regardons à quoi ressemble le fichier de configuration principal de YUM.

Ce fichier fonctionne dans l'esprit du fichier de configuration principal du serveur Web « Apache ».



Astuce : Les directives **« name » et « baseurl » sont obligatoires** pour définir un dépôt.
Si aucune option n'est définie dans le fichier /etc/yum.repos.d/*.repo alors celles de **/etc/yum.conf s'appliquent par défaut**.
Si vous voulez prendre connaissance des options il existe un **« man yum.conf »**

Dans la pratique les différents **dépôts ne sont pas définis dans ce fichier de configuration principale** mais en créant/éditant des **fichiers de définitions de dépôts** qui peuvent contenir la définition de un ou plusieurs dépôts.
Ces fichiers ont pour **extension .repo**, ils sont situés dans le répertoire suivant :

- /etc/yum.repos.d/

De base nous disposons de deux fichiers de définitions de dépôts :

- **CentOS-Base.repo** (ces dépôts sont les **dépôts officiels qui sont disponibles sur internet**, plus précisément sur les miroirs de CentOS.org),
- **CentOS-Media.repo** (ce dépôt est le **dépôt local** qui définit le contenu du **DVD CentOS qui a servi à l'installation**)

Voici un synopsis des sources de téléchargement préconfigurées dans du fichier « CentOS-Base.repo ».

Chacune des six sections (stances) commence par le nom du dépôt correspondant entre crochets : [base], [updates], [addons], [extras], [centosplus] et [contrib].

Voici une petite synopsis qui explique brièvement à quoi correspond chacun de ces dépôts de téléchargement.

- [base]: les paquetages de base de CentOS, tels qu'on les trouve dans les ISO (le DVD, les CD).
- [updates] : les mises à jour de [base] publiées après les ISO de CentOS. Il s'agit de mises à jour de sécurité, de corrections de bogues ou d'améliorations des paquetages de [base].
- [addons] : les paquetages requis pour la compilation des paquetages de [base], même s'ils ne font pas partie de ce groupe. On peut considérer qu'il s'agit d'une extension de [base]. Cette archive peut être désactivée, étant donné qu'elle ne contient pas de paquetages pour CentOS 5.

- [extras] : les paquetages non compris dans RedHat Enterprise Linux, compilés et gérés par les développeurs de CentOS, et qui ajoutent certaines fonctionnalités à la distribution de base. L'ensemble de ces paquetages est dûment testé et n'interfère pas avec la distribution de base.
- [centosplus] : les paquetages provenant de la contribution des développeurs et des utilisateurs de CentOS, mais susceptibles de remplacer des paquetages de [base].
- [contrib] : les paquetages provenant de la contribution des utilisateurs de CentOS.

Astuce : Pour **désactiver un fichier de définitions de dépôts** rajouter simplement **l'extension .orig au fichier** (Ex. : CentOS-Base.repo**.orig**).

Si vous ne **disposez pas d'une connexion permanente à internet** commencez par **désactiver le dépôt : CentOS-Base.repo** (en renommant ce fichier en CentOS-Base.repo.orig). Cela évitera des connexions incessantes vers internet.

Architecture:
Vous avez donc trois choix possibles pour profiter de la souplesse de YUM via les dépôts de logiciels :

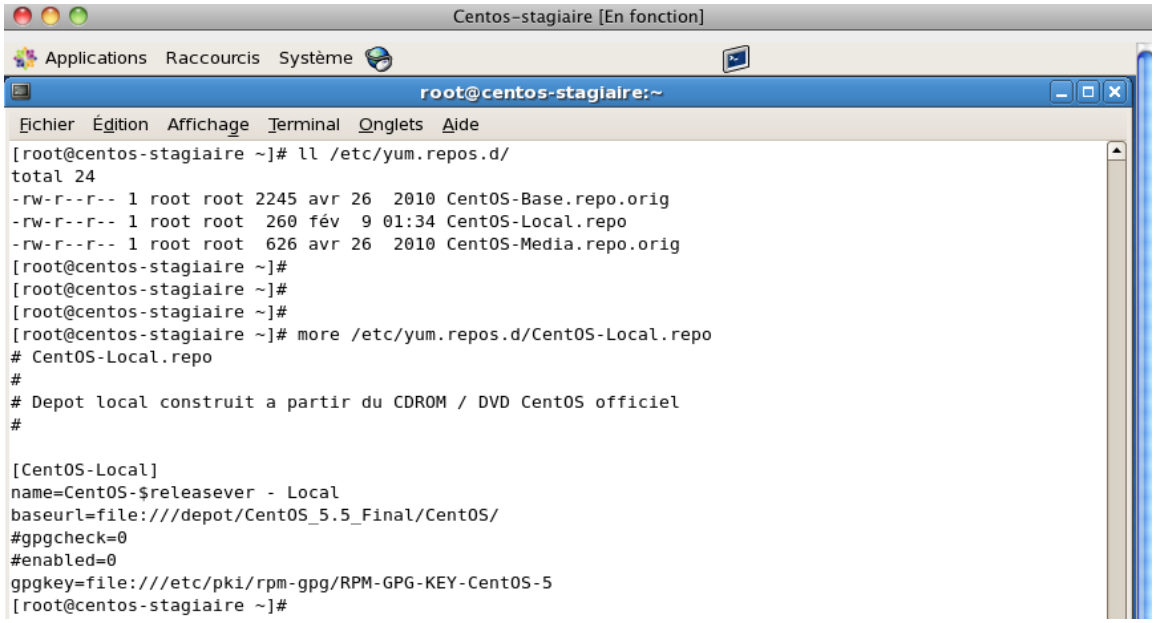
- **Travailler avec le dépôt du DVD**, là vous ne bénéficiez d'aucune mise à jour ou de logiciels supplémentaires hormis ceux du DVD officiel CentOS,
- **Récupérer les dépôts CentOS officiels complets** sur un des miroirs d'internet et le **copier** (station blanche) sur un **serveur de l'intranet** qui fera office de dépôt de référence pour votre site (vous devrez **maintenir à jour ces dépôts vous-même en mettant un place un miroir CentOS**, RSYNC est la solution pour assurer cette tâche),
- **Se connecter sur les dépôts CentOS** existants et à jour au **sein de votre intranet (solution préférable).**

Prenons le cas d'école du dépôt contenu sur le DVD officiel.

Travailler avec un DVD inséré dans le lecteur de vos serveurs en production n'est pas une solution confortable, notamment en cas de redémarrage à distance si le DVD est le premier périphérique de démarrage.
Nous allons **désactiver le dépôt du DVD (Media)** : CentOS-Media.repo (en renommant ce fichier en CentOS-Media.repo.orig). A ce stade nous nous retrouvons sans aucun dépôt de logiciels actifs.

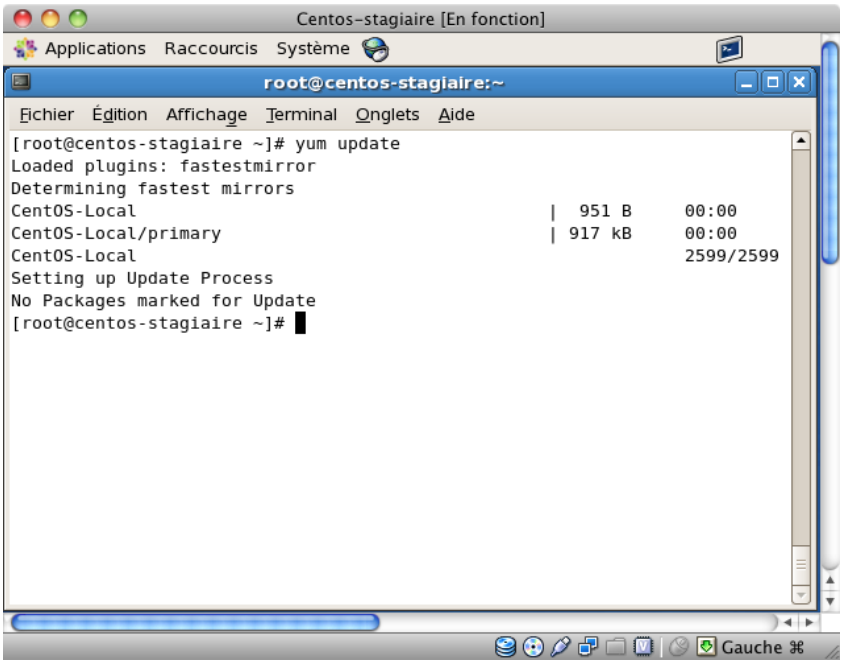
Précédemment nous avons copié le contenu de ce DVD sous le répertoire **« /depot »** en y copiant tous les RPM disponibles. Puis nous avons créé un dépôt avec ces mêmes RPM.
Nous allons donc nous en servir pour cette formation.

Maintenant voyons comment indiquer à YUM qu'il doit travailler à partir de ce dépôt pour gérer vos logiciels. Pour ce faire **créer le fichier « CentOS-Local.repo »** comme suit :



YUM dispose maintenant de toutes les informations pour pouvoir travailler à partir de ce nouveau dépôt **« CentOS-Local ».**
Il ne reste plus qu'à **mettre à jour** ce qui doit l'être, en l'occurrence ici les dépôts utilisés.
Nous utilisons donc la commande :

- yum update



Maintenant vous disposez d'un GNU/linux CentOS prêt à être utilisé en production.

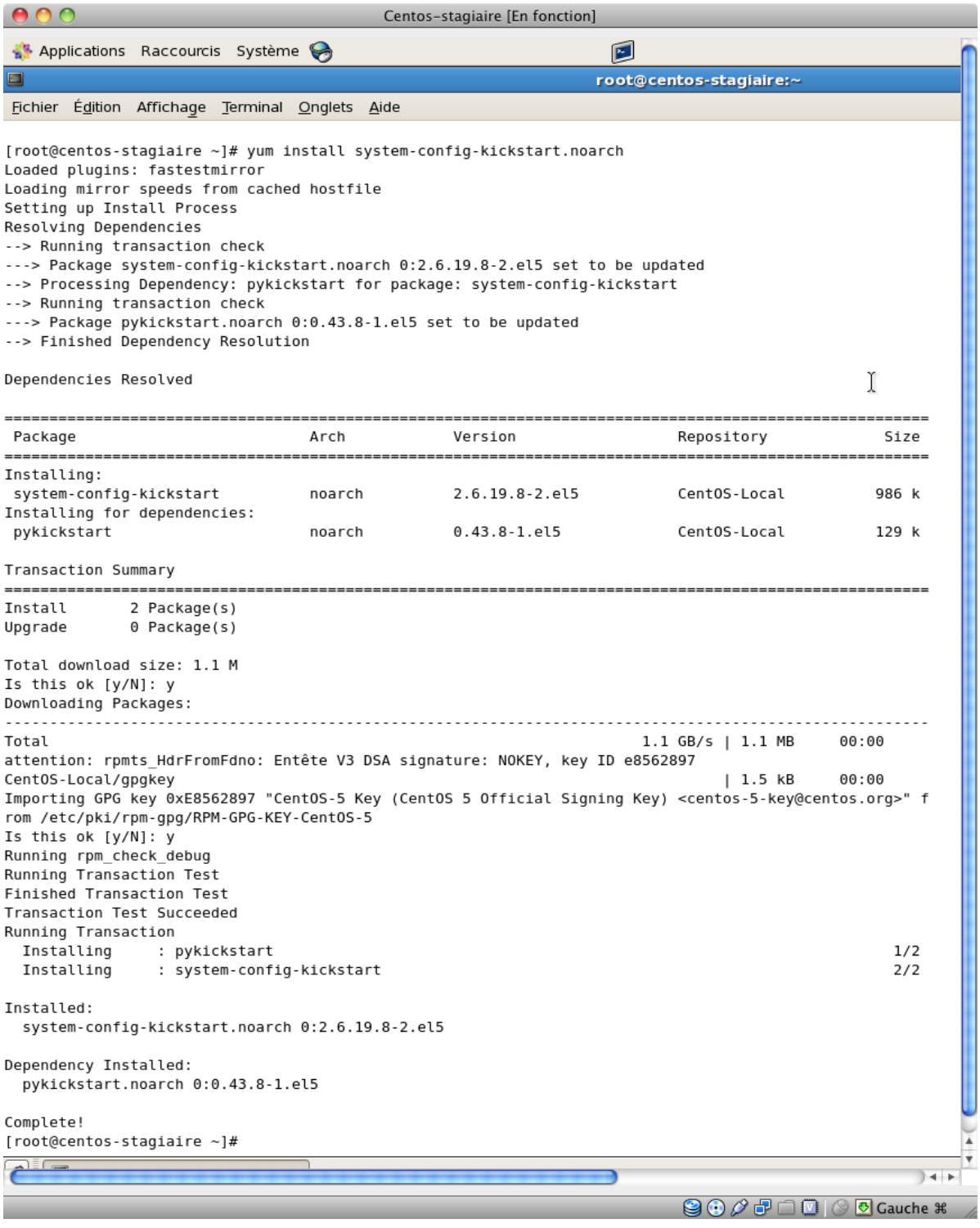
Voyons comment il est simple d'installer, supprimer, chercher des logiciels.

Voici une memento simplifié de la commande **YUM**.

Les caractères jokers sont autorisés pour filtrer et affiner vos manipulations.

Commande	Action
yum list all	Lister tous les paquetages installés et présents dans les dépôts
yum list available	Lister tous les paquetages présents dans les dépôts (et non installés)
yum list installed	Lister tous les paquetages installés
yum list recent	Lister les derniers paquetages ajoutés dans le dépôt
yum info <paquetage>	Donne des informations détaillées sur le paquetage
yum install <paquetage>	Installer le paquetage
yum update <paquetage>	Mettre à jour un paquetage s’il est installé
yum check-update <paquetage>	Vérifie si une mise à jour est disponible pour ce paquetage
yum upgrade	Attention va mettre à jour toute votre distribution si des mises à jour sont disponibles
yum remove <paquetage>	Supprimer un paquetage installé
yum clean all	Nettoyage du cache de YUM
yum makecache	Force la mise à jour du cache de YUM
yum search <paquetage>	Permet de rechercher un paquetage dans les dépôts

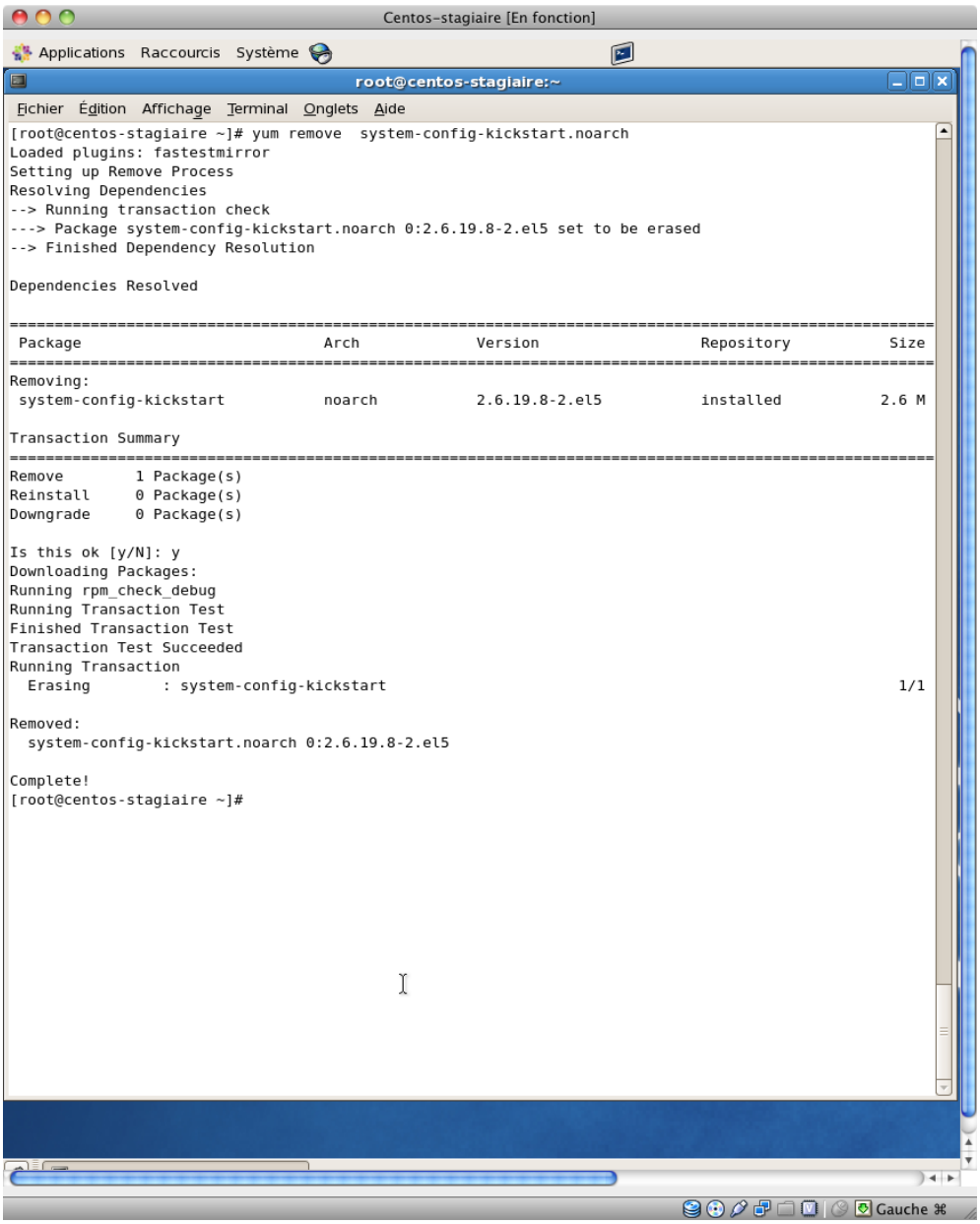
Voici comment installer le logiciel de configuration graphique Kick Start (déploiement automatique de la distribution CentOS).



Comme vous pouvez le voir YUM a résolu de lui-même une dépendance et a installé le paquetage **« pykickstart.noarch »** sans que nous lui ayons explicitement indiqué.

Ce cas est simple imaginez quand vous avez plus d'une vingtaine de dépendances, YUM est indispensable.

Et enfin de le supprimer dans la foulée … aussi simplement.



Pour aller plus loin …
Il faut savoir que YUM permet de gérer des plug-ins (extensions). Parmi-eux vous avez le plug-in **« yum-priorities »**

Cette extension permet de définir des priorités pour les différents dépôts de téléchargement. Il suffit de définir une variable `priority=N`, avec `N` compris entre 1 et 99, 1 représentant la priorité la plus haute et 99 la plus basse. Concrètement, si vous avez défini une priorité de 1 pour le dépôt `[base]` et une priorité de 10 (ou 30, ou 99, peu importe) pour `[rpmforge]`, les paquetages de ce dernier ne pourront jamais remplacer des paquetages de `[base]`. Yum les en empêchera tout simplement en les excluant de la liste.

Maintenant que vous savez ajouter, supprimer des briques logicielles à votre système GNU/Linux nous allons pouvoir commencer à comprendre son fonctionnement interne.

Les journaux systèmes

Bien utilisés, les journaux systèmes (log) permettent de débusquer rapidement un souci sous GNU/Linux.

Il est très important de vous référer à ces fichiers lorsque :

- vous avez un incident,
- vous tentez en vain de mettre en œuvre un service,
- vous avez un disfonctionnement du système.

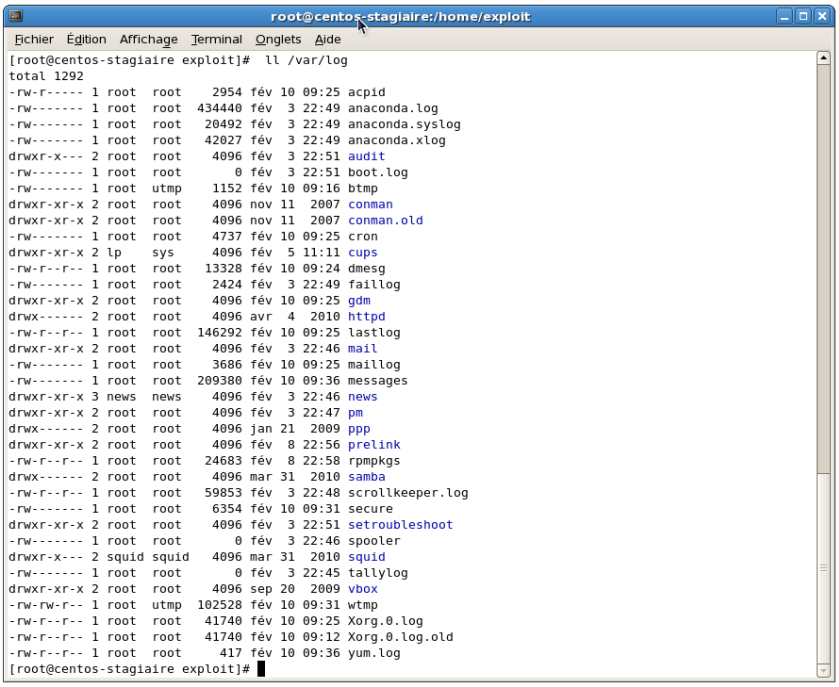
Dans 90% des cas, ce sera par l'analyse des journaux systèmes que vous pourrez résoudre vos soucis.

Ces derniers sont quasi tous localisés dans :

- /var/log/

Cependant si vous avez installé un **logiciel/service via compilation ou import de source** il se peut que cet élément **ne journalise pas par défaut dans ce répertoire**, donc ce sera à vous de **trouver où ce dernier journalise** ces messages de fonctionnements ou disfonctionnements.

Suite à une installation type « serveur basique » voici les journaux auxquels vous avez déjà accès :



Note : Tous ces journaux contiennent des **informations horodatées**.

La commande « **dmesg** » affiche les messages du **noyau notamment lors du démarrage.**
Ce journal se paramètre ici « **/etc/rc.sysinit** » et « **/etc/sysconfig/init** ». Cette commande travaille sur un tampon circulaire. Donc les derniers messages viennent écraser les premiers.
Cependant ces messages ne seront pas perdus car le fichier **/var/log/messages** stocke aussi ces messages.

/var/log/messages est le fichier qui centralise presque tous les messages du système y compris ceux du noyau. Il est alimenté par le **service (démon) syslog** vous pouvez le configurer avec le fichier **/etc/syslog.conf.** Ces messages sont archivés au fil du temps dans des fichiers **/var/log/messages.x.gz** orchestré par une **rotation des journaux paramétrable** (**/etc/logrotate.conf**) via l'ordonnanceur (/etc/crontab.conf).

Pour obtenir plus d'informations sur ces paramétrages de journaux veuillez consultez les pages de **man**.

Un des manières la plus efficace pour débusquer une erreur de paramétrage, un souci ou tout simplement prendre connaissance du comportement du système à l'une de vos actions, **consiste à tracer vos journaux en temps réel**.

Prenons l'exemple de l'insertion d'un DVD dans le lecteur : **vous ne savez pas sous quel périphérique /dev/** votre DVD a été monté (on peut tout aussi bien prendre l'exemple d'une clef USB ou d'un disque dur externe)

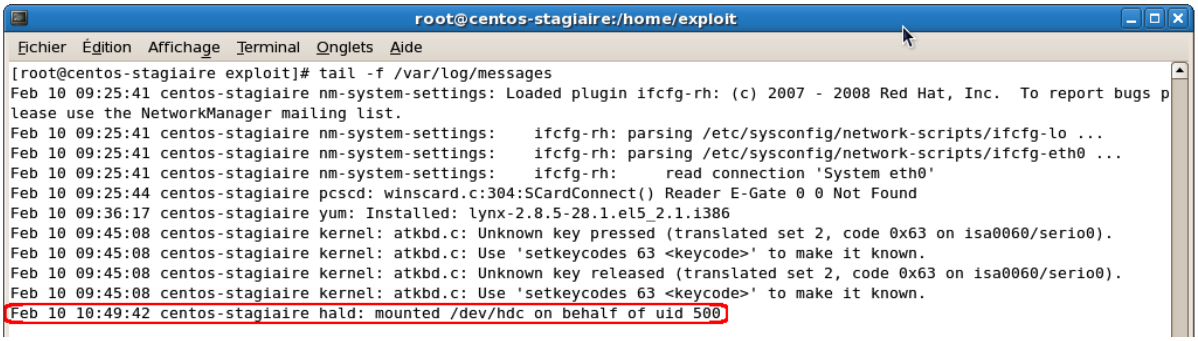
Avec les journaux visualisés en temps réel, rien de plus facile.

Dans un terminal commencez par vous placer en « root », puis lancez la commande suivante :

- tail -f /var/log/messages

Ensuite insérer votre DVD dans le lecteur.

Voici le message produit par l'action d'insérer un DVD dans le lecteur de la machine virtuelle (VM) si vous êtes connecté en session graphique :



Le DVD utilise le périphérique : **/dev/hdc** et le programme assurant cette tâche et le service **hald.**
Le DVD a été **monté** le **10 février à 10 :49** sur la machine **centos-stagiaire** par l'utilisateur dont **l'UID est 500** (exploit).

Pour le test d'une commande le principe sera le même sauf que vous aurez à **ouvrir deux terminaux**. Dans l'un vous lancerez la visualisation en temps réel (tail -f) de votre journal système dans l'autre vous exécuterez vos commandes. Vous visualiserez instantanément leurs résultats : bon ou mauvais. Avec les messages d'erreurs ou d'informations affichés vous serez capables de **solutionner rapidement le problème**. Bien entendu cette méthode de travail sous-entend de comprendre ce que vous cherchez. **Vous pouvez l'appliquer aux autres journaux présents. (mail, samba, named etc.)**

Comprendre le démarrage du système

La séquence d'amorçage : standard

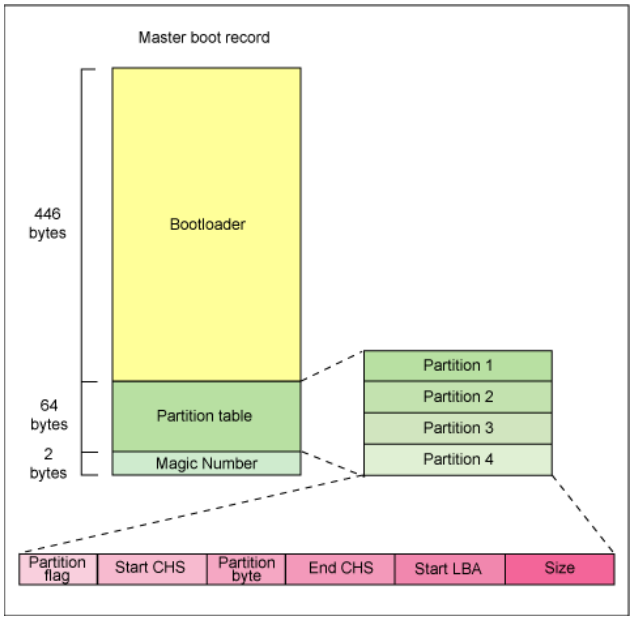
Chaque ordinateur, qu'il soit serveur ou station, dispose d'un **BIOS (ou EFI)** qui est la **première couche logicielle** exécutée lors du démarrage de votre machine.

Prenons l'exemple le plus courant : **le BIOS**. Ce dernier active le **chargeur de programme initial** ou **primaire** (**I**nitial **P**rogram **L**oader) situé dans le **MBR (Master Boot Record)**, il s'agit des **512 premiers octets** du support de démarrage choisi.
Note : Le MBR n'est pas le seul format, il existe aussi GPT qui autorise plus de partitions etc.

Astuce : Pour afficher le contenu du MBR (disque de boot : /dev/sda)

- dd if=/dev/sda of=mbr.bin bs=512 count=1
- od -xa mbr.bin

Voici son anatomie :



Pour les plus curieux : http://www.ibm.com/developerworks/linux/library/l-linuxboot/

Pour GNU/Linux ces **512 octets ne suffisent pas pour loger un menu de démarrage et lancer le chargement du système**. Le chargeur de programme initial va donc charger un **chargeur de démarrage secondaire.** Ce dernier disposera de plus d'espace disque pour pouvoir proposer un menu de démarrage et autoriser la saisie de commandes de démarrage particulières. Il faut que ce dernier se **situe sur une partition dite « active ».**

C'est durant cette seconde étape qu'il sera possible de choisir :

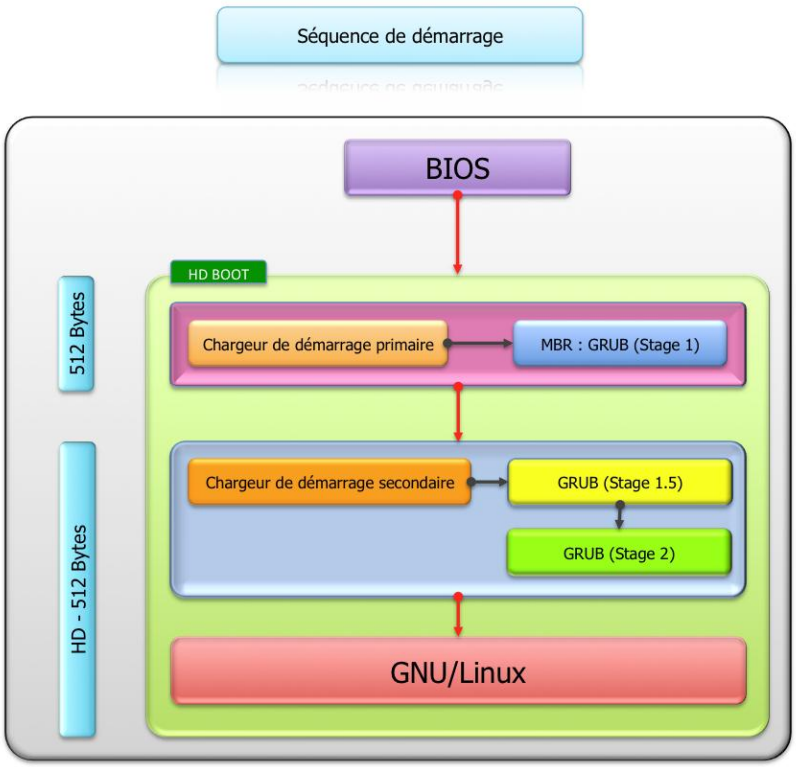
- Le **noyau** Linux à démarrer,
- Les différentes **options de démarrage** à passer en paramètre à ce noyau,

- Le démarrage d'un autre système d'exploitation : cas du **« multiboot »**

Note : Sur un serveur de production le **multiboot est rarement utilisé**, il s'agit plutôt d'une option intéressante pour les stations de travail. Cependant si l'on désire disposer de plusieurs systèmes d'exploitations sur une même machine physique **on préfèrera virtualiser** les systèmes invités **sur un système hôte disposant d'un hyperviseur de type 1**.

Enfin le système d'exploitation pourra commencer son démarrage.

Voici un résumé d'une **séquence d'amorçage standard**



Note : Il existe aussi d'autres séquences d'amorçage, exemple par le réseau via PXE (**P**re-boot e**X**ecution **E**nvironnement) en récupérant une image de système d'exploitation sur un serveur TFTP accessible.

GRUB

La quasi-totalité des systèmes GNU/Linux utilise maintenant GRUB (**GR**ant **U**nified **B**ootloader).
Voici ses principales fonctionnalités :

- Contrairement à LILO, GRUB n'a pas besoin d'être réinstallé pour mettre à jour sa configuration. GRUB prend en compte les modifications de son fichier de configuration dynamiquement.
- Au cas où le fichier de configuration serait incorrect, GRUB peut fournir un interpréteur de commandes pour permettre à l'utilisateur de charger un système d'exploitation manuellement.
- GRUB est très portable : il permet de charger aussi bien des systèmes compatibles avec le multiboot que des systèmes non-compatibles avec cette fonction (comme Microsoft Windows). GRUB

supporte en outre beaucoup de systèmes de fichiers comme ext3, VFAT ou NTFS. GRUB est également compatible avec le mode Logical block addressing (LBA).

- GRUB peut être utilisé avec différentes interfaces. Beaucoup de distributions GNU/Linux utilisent le support graphique de GRUB pour afficher au démarrage de l'ordinateur un menu avec une image de fond, et parfois une prise en charge de la souris.
- GRUB peut télécharger des images de systèmes d'exploitation depuis un réseau, et supporte donc les ordinateurs sans disques. GRUB peut donc décompresser ces images pour les charger ensuite.

Voici son fonctionnement.
Pour lancer un noyau ou un autre système d'exploitation GRUB doit lancer successivement 3 exécutables (cf. schéma page précédente) :

- Stage 1 : le **premier** situé dans le MBR, sert à **lancer le « Stage 1_5 »** qu'il trouvera en cherchant sur une partition active (voir la table des partitions contenu dans le MBR),
- Stage 1_5 : le **second**, situé **sur une partition active**, assure une prise en charge minimal du **système de fichiers** (reiserfs, ext2, etx3 etc.) via des versions spécifiques (Ex. : reiserfs_stage1_5, e2fs_stage1_5 etc.),
- Stage 2 : le **troisième** permet **d'afficher le menu**, de **sélectionner un noyau** ou un **autre système** d'exploitation que GNU/Linux et enfin de passer en **ligne de commande** (en voici quelques une : **e**, d, o, b**)**.
  Cet exécutable, comme tout logiciel GNU/Linux se paramètre via un fichier de configuration : **/etc/grub.conf**, ci-dessous.

```
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE:  You have a /boot partition.  This means that
#          all kernel and initrd paths are relative to /boot/, eg.
#          root (hd0,0)
#          kernel /vmlinuz-version ro root=/dev/VolGroup00/LogVol00
#          initrd /initrd-version.img
#boot=/dev/sda
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title CentOS (2.6.18-194.el5)
        root (hd0,0)
        kernel /vmlinuz-2.6.18-194.el5 ro root=/dev/VolGroup00/LogVol00 rhgb qui
et
        initrd /initrd-2.6.18-194.el5.img
```

Bien que nous n'allons pas détailler ici toutes les options de GRUB voici un descriptif des principales :

Paramètre	Fonctionnalité
timeout	Nombre de secondes avant le démarrage par défaut
default n	Démarrage par défaut (0=1er titre, 1=2ème titre, etc.)
gfxmenu	Chemin vers un menu graphique
title nnnnn	Début d'une section ou entrée de menu GRUB
root(hdx,y)	Définition de la partition racine (root) de démarrage, Ex 1. : **root(hd0,0)** indique une racine située sur la **première partition** du **premier disque** détecté par le BIOS, Ex 2. : **root(hd2,3)** indique une racine située sur la **quatrième partition** du **troisième disque** détecté par le BIOS
kernel	Le nom de l'image du noyau Linux, suivi de ses paramètres de démarrage. Attention : le « /vmlinuz …. » n'indique pas la racine du système de fichiers mais celle de root(hd0,0) soit **/boot/**vmlinuz-2.6.xxxxxx
initrd	INITial RamDisk. Le noyau charge se fichier comme disque dur en mémoire (RAMDISK) afin de disposer d'un système d'exploitation minimal (pilotes initiaux et configurations de bases)

rootnoverify	La racine spécifiée, à ne pas monter par GRUB
	Par exemple lorsque GRUB ne supportait pas la lecture sur système de fichiers NTFS nous avions ceci,
	Ex 3 : `title Windows XP`
	`rootnoverify (hd1,0)`
	`chainloader +1`
	Ici on donne la main au bootloader « NTLDR » de Windows xp
	
	Attention sous Vista ou 7 le bootloader a changé, de plus GRUB supporte maintenant la lecture en NTFS donc :
	Ex 4 : `title Windows 7`
	`root (hd1,0)`
	`makeactive`
	`chainloader +1`

chainloader +1	Démarrer le premier secteur de la racine spécifiée par « root » ou « rootnoverify ».
hiddenmenu	Cette commande masque le menu complet au démarrage (attention)

Pour installer GRUB ou le réinstaller procéder comme suit en choisissant bien votre premier support inscriptible de démarrage sélectionné dans le BIOS :

- grub-install /dev/sd**x**

Vous pouvez également mettre un mot de passe sur le menu GRUB. Cependant rappelons qu'une fois la machine **à disposition physique d'un éventuel assaillant**, elle est considérée comme **compromise**.

Attention :

Les périphériques reconnus par GRUB sont indiqués dans le fichier suivant:

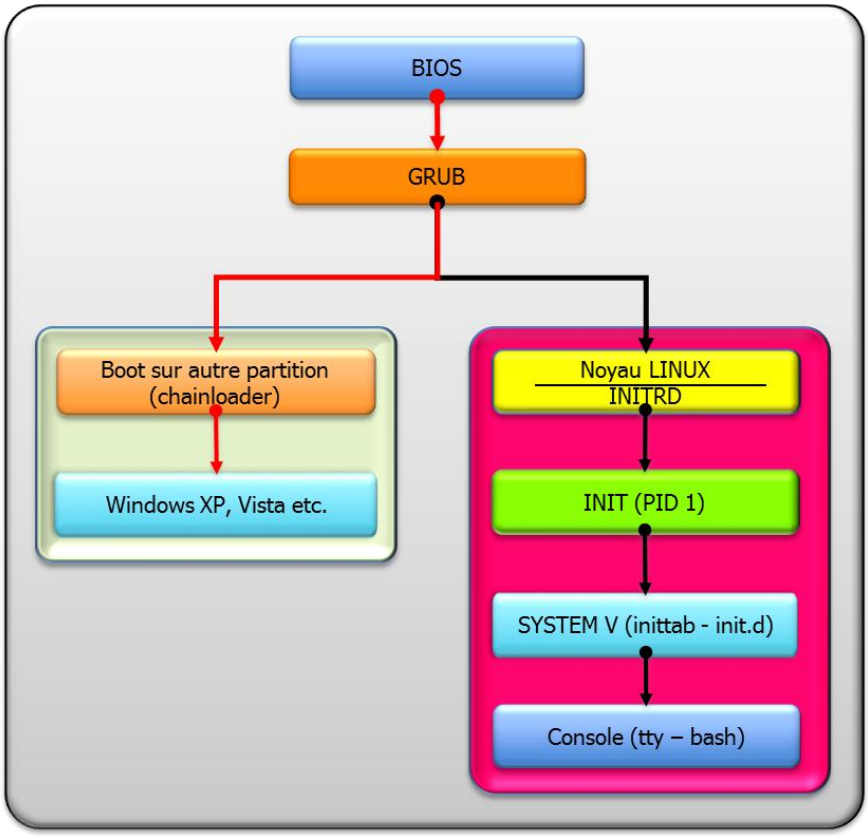
- cat /boot/grub/device.map

N'oubliez pas : GRUB dispose de sa propre convention de nommage concernant les partitions et périphérique.

```
[root@centos-stagiaire ~]# cat /boot/grub/device.map
# this device map was generated by anaconda
(hd0)     /dev/sda
[root@centos-stagiaire ~]# _
```

Maintenant que nous avons **amorcé notre système d'exploitation**, nous allons passer à la séquence de démarrage complète de GNU/Linux.

Voici une synthèse du démarrage de GNU/Linux :



Au chargement du noyau LINUX une multitude d'information s'affiche. On peut retrouver dans **/var/log/dmesg** les journaux du processus INIT.
A l'initialisation du noyau :

- Le matériel est détecté,
- Initrd est chargé, les modules présents éventuellement chargés,
- Le système de fichiers racine est monté en lecture seule,
- Une console TTY crée (sans BAsh à ce niveau)
- Le premier processus est lancé « init » dont le **P**rocessus **ID**entifier est 1 (PID 1)

Note : INIT est le dernier processus à être stoppé.

Le processus INIT contrôle le **lancement ou l'arrêt des services** (cf. SYSTEM V Unix). Il gère également les différents **niveaux d'exécution du système d'exploitation**. INIT est le **processus père** de tout les autres processus. Il se configure dans le fichier **/etc/inittab**.

Astuce : En cas **d'erreur de configuration de ce fichier**, il faudra démarrer le système avec le niveau d'exécution 1 (S, single user).

Vous pouvez également contrôler le démarrage des services un à un en appuyant sur la **touche « I »**. Vous aurez ensuite le choix de démarrer tel ou tel service.

Le fichier **/etc/inittab** contient trois type de lignes :

- Des commentaires (avec le traditionnel #),
- Des actions à lancer,
- Un type de gestion pour un niveau d'exécution donné.

Voici le format d'une ligne :

- id : runlevel : action : command

Par exemple : Le choix du niveau par défaut (RunLevel) qui sera atteint après le démarrage de GNU/Linux se règle dans le fichier de configuration du processus INIT, **/etc/inittab**, voici son contenu pour démarrage en mode graphique (5):

- id : 5 : initdefault :

```
# Default runlevel. The runlevels used by RHS are:
#   0 - halt (Do NOT set initdefault to this)
#   1 - Single user mode
#   2 - Multiuser, without NFS (The same as 3, if you do not have networking)
#   3 - Full multiuser mode
#   4 - unused
#   5 - X11
#   6 - reboot (Do NOT set initdefault to this)
#
id:5:initdefault:
```

Attention : Veuillez noter la définition des Runlevel pour une RedHat (RHS).
Ce **fichier** est très **sensible**, hormis le niveau de démarrage initial, vous n'aurez pas à le modifier.

Pour résumer voici ce que le fichier **inittab** va exécuter (par ordre) :

- Définir le **niveau d'exécution initial** de GNU/Linux,
- Lancer le script d'initialisation du système : **/etc/rc.d/rc.sysinit**,
- **Gestion des services** par niveau d'exécution (SYSTEM V),
- Action lors de l'appel **CTRL+ALT+DEL**,
- **Comportement** de GNU/Linux en fonction de **l'alimentation électrique** (coupure, remise du courant, etc.),
- Lancement des **consoles virtuelles TTY** (accessibles via ALT+Fx),
- Lancement du **gestionnaire de connexion graphique** (basique) au système GNU/Linux.

Le fichier **/etc/rc.d/rc.sysinit** est la pièce maitresse qui s'occupe de la configuration de base du système. Sur une distribution RedHat (vs Debian ou OpenSuse) il est monolithique donc conséquent (environ 985 lignes de script).
Il effectue les tâches suivantes, dans cet ordre :

- Mise en place des **paramètres du noyau** contenu dans **/etc/sysctl.conf**,
- Configuration de l'**horloge** système,
- Peuplement de **l'arborescence des fichiers de périphérique** (**/dev** via udev),
- Chargement des **tables de caractères** du clavier,
- Définition du **nom d'hôte** du système,
- Contrôle et mise en place des **sous-systèmes disque de bas niveau** (LVM)
- Contrôle et montage du système de fichiers **racine en lecture écriture** cette fois,
- Contrôle et montage des **autres systèmes de fichiers locaux**,
- Activation des éventuels **quotas disque locaux**,
- Activation des partitions de **SWAP**,
- **Nettoyage des verrous** et des fichiers **temporaires** (notamment lors d'un arrêt précédent brutal).

Les niveaux d'exécution : le SYSTEM V

Sur un système GNU/Linux un **niveau d'exécution ou Runlevel** est un état dans lequel se trouve le système à un instant donné. Il est possible de basculer d'un niveau à un autre. Chaque niveau a ses propres propriétés.
Voici un tableau résumant les différents niveaux d'exécution :

Niveau	Description
0	Halt : stoppe le système, éteint la machine électriquement si l'ACPI est ok
1 (S)	Mono-utilisateur (Single User) utilisé pour la maintenance, mode console
2	Multi-utilisateur, sans réseau, mode console
3	Multi-utilisateur, avec réseau, mode console
4	Idem que 3, à la convenance de l'administrateur
5	Multi-utilisateur, avec réseau, mode graphique X Window
6	Reboot : redémarrage de la machine

En général on travaille sur un des 3 niveaux d'exécution :

- Le 1 ou S, ce niveau vous permet de récupérer le mot de passe « root », réparer ou manipuler les partitions verrouillées en temps normal,
- Le 3, ce niveau est le plus utilisé pour la production car il ne consomme pas de ressource graphique et offre les services réseaux que vous avez choisis,
- Le 5, pour utiliser votre distribution en mode graphique.

Evidement lors d'une séquence de démarrage ou d'arrêt on passe par plusieurs niveaux.
Imaginons que vous êtes au niveau 3, un ordre d'arrêt (« **halt** » vous fera passer du niveau 3 au niveau 0.

Attention : les fonctionnalités de chaque niveau ne sont pas standardisées, des différences mineures peuvent exister entre distribution GNU/Linux.

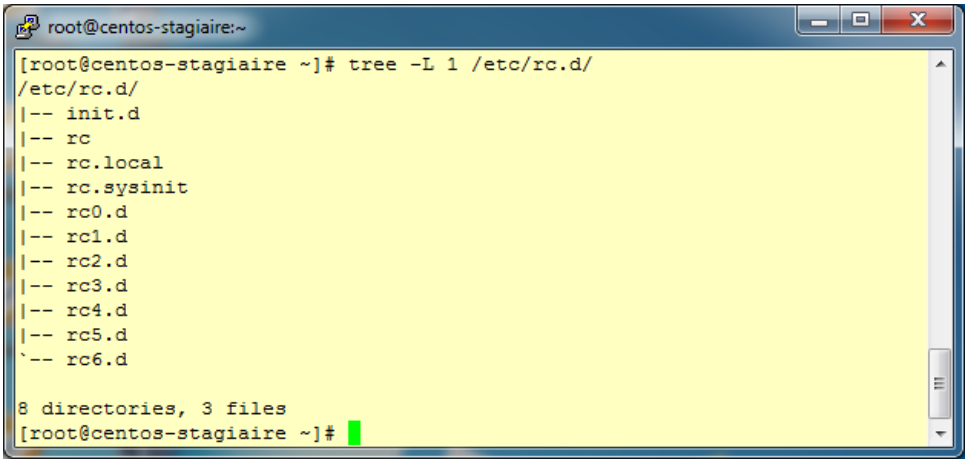
Vous pouvez **changer** de **RunLevel**, en invoquant la commande **« init »** (lien symbolique **telinit**):

- telinit <RunLevel_choisi>

Pour connaître le RunLevel en cours et le précédent :

- runlevel
- who -r

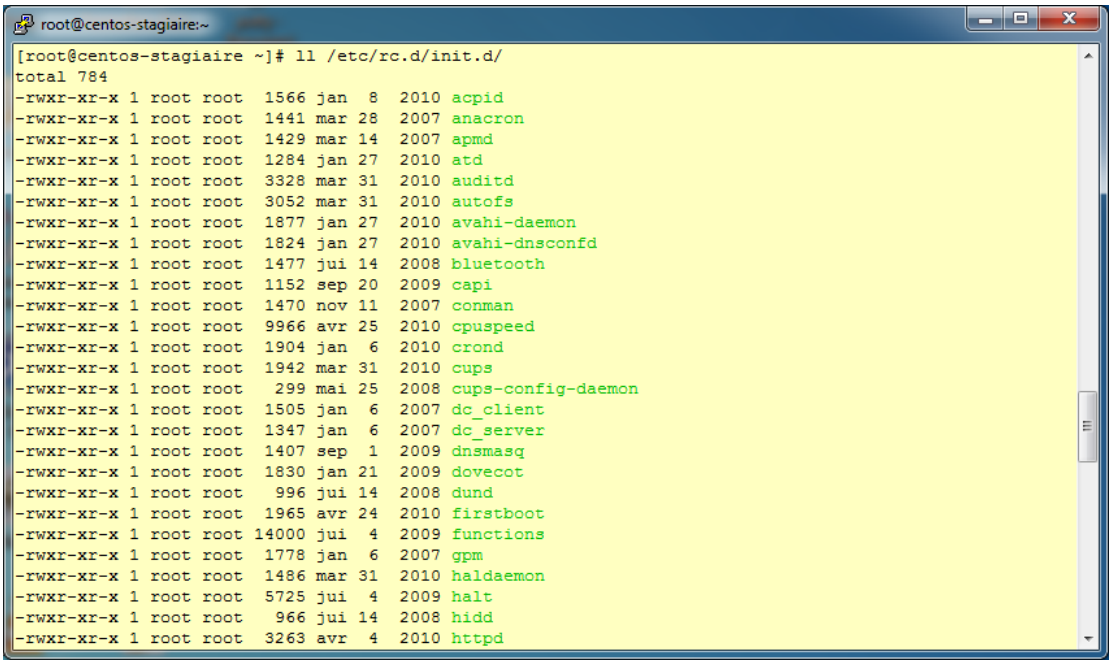
Note : N 5 signifie que le système se trouve au **niveau d'exécution 5**, le **N : None** donc pas de niveau précédent.
Le SYSTEM V assure la gestion des services du système GNU/Linux en fonction du niveau d'exécution, voici son arborescence générale :



Il faut savoir que les services qu'ils soient **réseaux** (Samba, SSHD, NFS etc.) ou **non** (ACPI, HAL, CRON etc.) sont lancés (**S**tart) ou stoppés (**K**ill) en fonction du niveau d'exécution courant.

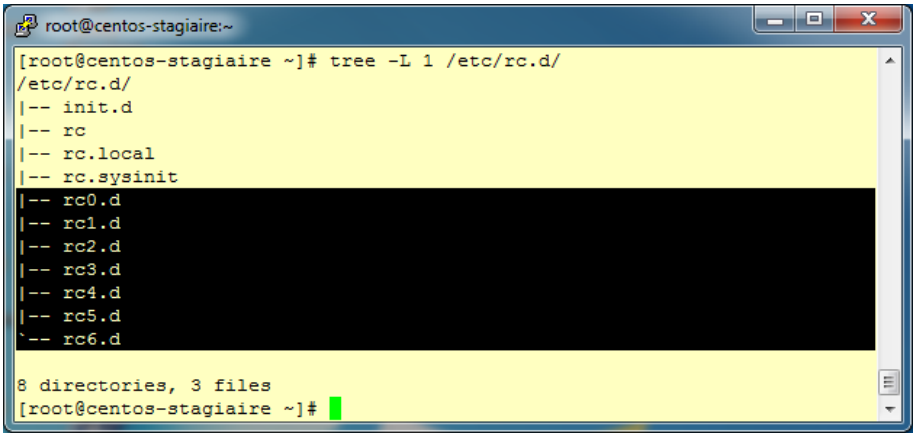
Le fichier **/etc/rc.d/rc :**
Ce script qui assure la bascule vers un niveau d'exécution choisi (il est utilisé dans **inittab** par exemple). Ce script compare les services qui doivent être arrêtés ou démarrés entre l'ancien et le nouveau niveau d'exécution. Si un service est commun aux deux niveaux, il est maintenu en l'état.

Les services disponibles sur la machine sont tous définis par des scripts, lesquels sont contenus dans le répertoire **/etc/rc.d/init.d/** (ou /etc/init.d qui est un lien symbolique) comme suit :



Il faut savoir que, en fonction du niveau d'exécution où se trouve GNU/Linux, seuls certains services sont activés, tous ne le sont pas.

La mise en fonction d'un service à un niveau d'exécution donné est définie dans les répertoires **/etc/rc.d/rcX.d.**
X (0, 1, 2, 3, 4, 5, 6 sont les différents niveaux d'exécution)



Le passage d'un niveau à un autre lancera ou stoppera des services via un mécanisme de liens symboliques pointant vers les scripts des services. Voici la règle de nommage de ces liens symboliques :

[SK]pp<nom_du_service>

Avec :

- **S** : start, soit le démarrage du service,
- **K** :Kill, soit l'arrêts du service,
- **pp** : la position de démarrage ou d'arrêt du service (00=premier et 99=dernier)
- **<nom_du_service>** : le nom du service contenu dans /etc/rc.d/init.d

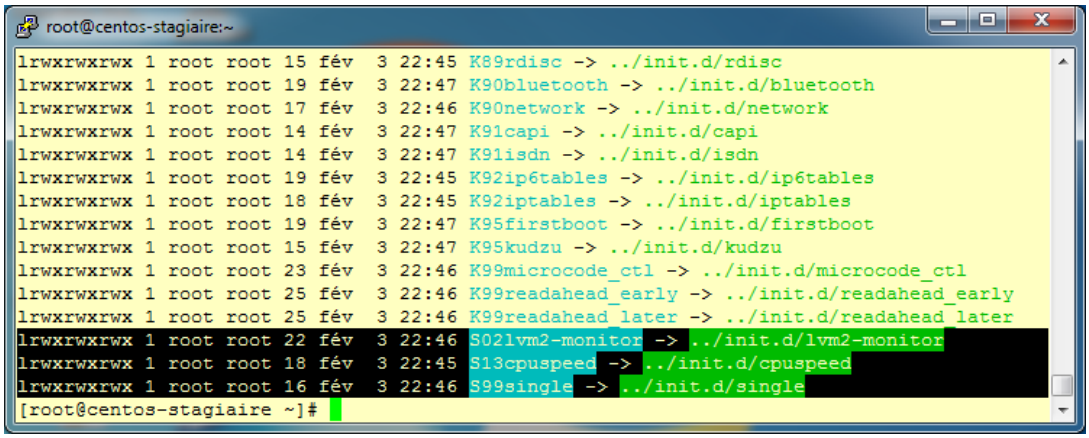
Il faut savoir que le script d'un service contenu dans **/etc/rc.d/init.d** prend, au minimum, les paramètres en entrées suivants :

- start,
- stop,
- status.

Pour voir la structure d'un script de service, saisissez (exemple : <nom_du_service> = cupsd) :

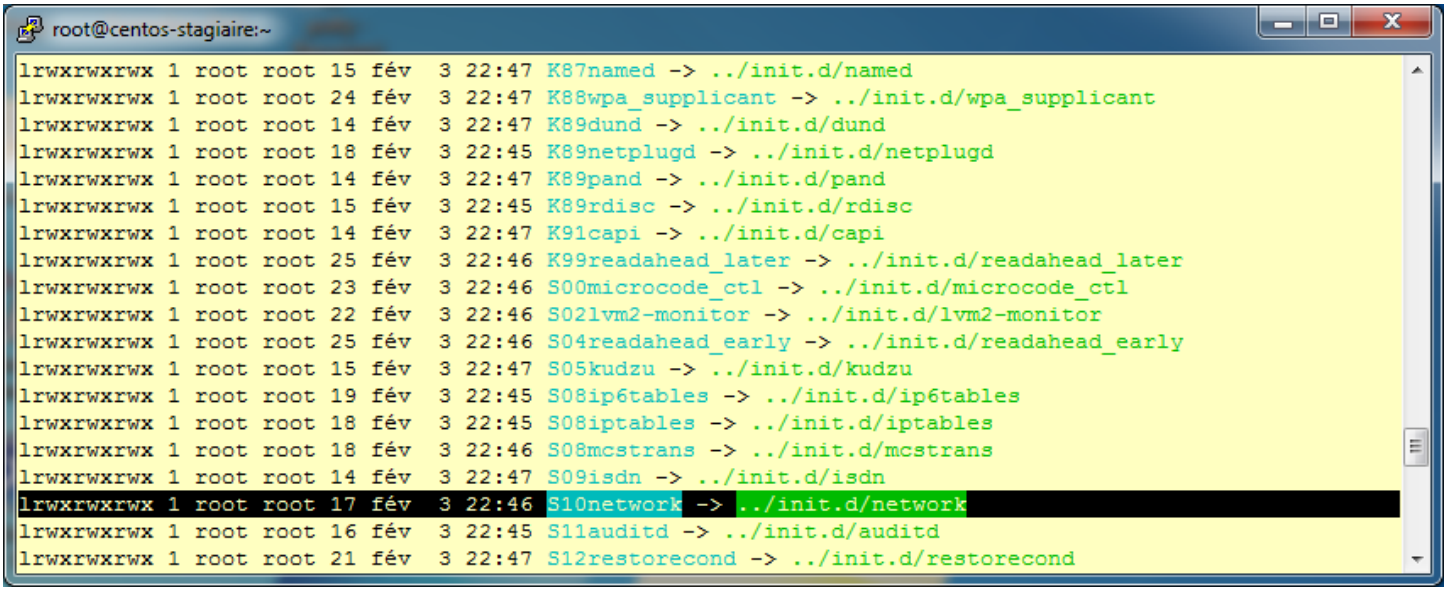
➢ vim /etc/rc.d/init.d/<nom_du_service>

De ce fait en fonction de l'appel, le script du service sait s'il doit démarrer ou arrêter le programme visé (ici **« cupsd »**).
Voyons le cas du niveau d'exécution 1 (S), le mode de maintenance de GNU/Linux, aucun service réseau n'est démarré. Seul 3 services sont démarrés (**S**xx), les autres sont arrêtés (**K**xx) :



Autre exemple, au **niveau d'exécution 3** (support réseau, Multi-utilisateur, sans mode graphique) défini sous **/etc/rc.d/rc3.d**, vous devez saisir cette commande pour voir ce qui est arrêté ou démarré :

➢ ll /etc/rc.d/rc3.d/



On constate, entre autre, qu'au niveau d'exécution 3, le service réseau (S**10**network) est bien démarré. Le service DNS, en haut, (K**87**named) est stoppé.

Pour illustrer la position de lancement des scripts, veuillez noter le lancement du pare-feu de GNU/Linux (S**08**iptables) **avant** de monter le réseau (S**10**network) : on protège le système avant de l'ouvrir au réseau.

Le SYSTEM V est majoritairement constitué de scripts et de liens symboliques pour la partie ordonnancement des arrêts et démarrages des services
La lourdeur d'administration est réelle et source d'erreur, pour palier à cette dernière il existe un outil incontournable : « **chkconfig »**.

Attention : Cette commande ne reflète pas l'état des services à l'instant temps, mais ce qu'ils devraient être pour un niveau d'exécution donné (arrêtés ou démarrés).
En effet on peut très bien arrêter/démarrer un service manuellement **après un changement** de niveau d'exécution. Dans ce cas « **chkconfig »** ne fournit pas les informations réelles sur ce service.

Cette commande permet de choisir le fait qu'un service soit présent ou non à un certain niveau d'exécution, ainsi que la gestion du gestionnaire de **service à la demande « xinetd »**.

Voici ses options :

chkconfig --list	Liste l'état des services en fonction des niveaux d'exécution x:marche/arrêt
chkconfig --list <nom_du_service>	Liste la configuration d'un service donné
chkconfig --level xxx <nom_du_service> on/off	Active ou désactive le service pour les niveaux d'exécution indiqués (xxx)
chkconfig --add <nom_du_service>	Ajoute le service indiqué dans la configuration SYSTEM V
chkconfig --del <nom_du_service>	Supprime le service indiqué de la configuration du SYSTEM V

Bien sûr il est tout à fait possible de créer soi-même son propre service et l'intégrer dans le SYSTEM V. Néanmoins, le code du fichier de script d'un service est complexe de par ses multiples dépendances et implique une maîtrise totale de :

- **L'architecture SYSTEM V de votre distribution** (entête de script etc.),
- La gestion des divers **processus du service à intégrer**,
- La gestion des **verrous système**,
- Du **Scripting** en BASh avec gestion des **codes erreurs**.

De ce fait, avant de prendre la décision de développer vos propres scripts SYSTEM V etc., **cherchez si le paquetage n'est pas présent dans les dépôts ou sur internet.**
Beaucoup de disfonctionnements viennent **d'une intégration hasardeuse dans le SYSTEM V** d'un logiciel/service compilé manuellement. C'est pour cela que des mainteneurs de paquetages sont désignés au plus haut niveau.

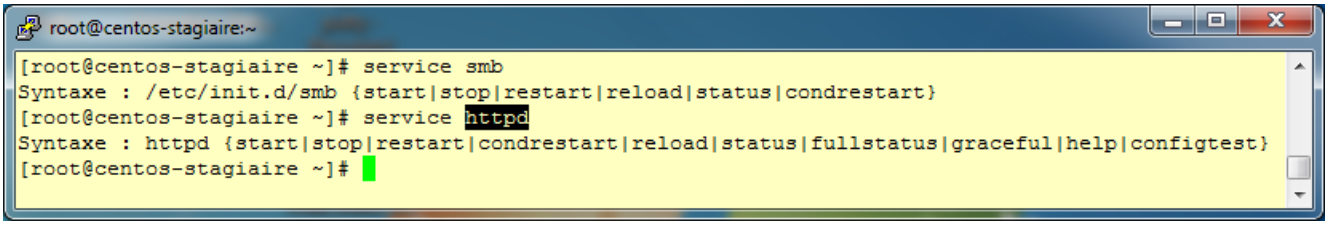
Maintenant voyons comment manipuler les services installés.

Rappel : Les **noms exacts des services** sont disponibles sous **/etc/rc.d/init.d**.

Commande	Fonctionnalité
service <nom_du_service> start	**Démarrer** un service
service <nom_du_service> stop	**Stopper** un service
service <nom_du_service> restart	**Redémarrer** un service
service <nom_du_service> status	Connaître le **statut** d'un service
Idem avec ➔/etc/rc.d/init.d/sshd restart	Redémarrage du service SSHd

Note : Certains services proposent une multitude d'option de manipulation (Ex. : Apache avec « **httpd** »).
Pour en prendre connaissance vous pouvez lancer les commandes suivantes :

- **service <nom_du_service>**, (idem avec cat /etc/rc.d/init.d/<nom_du_service> | more)



Enfin voici les commandes pour arrêter ou redémarrer votre système, en espérant que vous n'aurez pas à les utiliser souvent.

Pour l'arrêt : **« halt »**
Pour le redémarrage : **« reboot »**
Pour plus d'option voir la commande suivante : **shutdown <param> <délai> <message>**

Vous avez maintenant une vue globale et précise du démarrage d'un système GNU/Linux.
On s'aperçoit donc que le démarrage est avant tout un ordonnancement savant et complexe de scripts et de quelques binaires.
Par conséquent la modification d'un des scripts touchant la séquence de démarrage de GNU/Linux implique une très bonne connaissance de la distribution et peut complètement mettre hors d'usage votre serveur.

Alors faites toujours des **copies des fichiers que vous allez modifier** et **préparez vos interventions en lisant la documentation** cela permettra d'éviter des soirées en salle serveurs.

La gestion du système de fichiers

Le système de fichiers est l'élément visible dans lequel les données sont stockées. Il s'appuie sur une multitude de couche logique et physique.

Dans ce chapitre nous allons donc étudier :

- La présentation bas niveau (physique),
- Le partitionnement (logique),
- Le système de fichiers (vue haut niveau),
- La gestion des droits d'accès (ACL).

Au niveau matériel (bas niveau)

Il faut rester vigilant quant à la séparation des couches dites physiques et logiques.
De nos jours la notion de disque dur « physique » (ou tout autre média matériel) et de disque dur « logique » devient extrêmement floue et difficile à définir lorsque que vous commencez à travailler sur des systèmes de stockage professionnel.

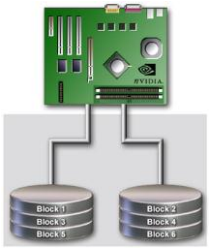
Attention : La difficulté de distinction des couches peut également s'appliquer à d'autre domaine que le stockage, cela est en partie dû à l'utilisation massive de la virtualisation.
Exemple : le système GNU/Linux sur lequel vous êtes en train d'expérimenter (en cours) : il n'y a rien de physique et pourtant …

Sur un système d'exploitation vous pouvez manipuler plusieurs types de disque dur.
Entendons par **disque dur,** un disque qui peut être présenté à GNU/Linux comme un **périphérique de type bloc**.
En voici un échantillon que vous serez amené à côtoyer quotidiennement :

- Les disques dur IDE (aussi appelé PATA) sont nommés par le noyau Linux comme suit **« hdx » :**
  hda : IDE0, Master
  hdb : IDE0, Slave
  hdc : IDE1, Master
  hdd : IDE1, Slave
  etc.
- Les disques durs SATA, SCSI, SAS etc. ou périphérique USB, FireWire etc. qui sont reliés à des contrôleurs SCSI, iSCSI (target/initiator), SCA, SATA, SAS, FiberChannel(FC), USB, FireWire etc. sont nommés par le noyau comme suit **« sdx » :**
  sda : premier disque
  sdb : deuxième disque
  sdc : troisième disque
  etc.

Note : Vous pourrez aussi trouver des périphériques sur scdX, srX (sur des graveurs, lecteurs DVD, Blue-Ray, etc) ou cXdYpZ (sur SmartArray HP, X = slot, Y = disque et Z= partition).

Prenons le cas courant des **disques durs physiques** cités ci-dessus (SATA, SCSI, SAS, etc.) connectés sur une **carte contrôleur RAID** (Redundant Array of Independent Disks) **matériel d'un serveur.**
En fonction du type de RAID matériel mise en place dans le BIOS de votre **carte RAID, celle-ci présentera** un ou plusieurs **disques logiques à GNU/Linux**. Ce dernier le **prendra en charge** comme un ou plusieurs **disques durs physiques** (un périphérique de bloc).



Note : Ne confondez pas RAID **matériel, logiciel** voir **semi-matériel** (logiciel propriétaire en fait).
En effet le RAID semi-matériel n'est pas supporté par le noyau Linux, en lieu et place vous disposerez de l'ensemble des disques dur présents sur ce type de contrôleur vu par le noyau Linux.

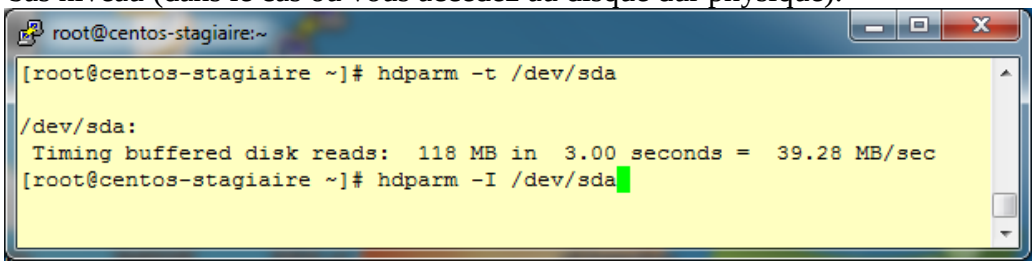
Il faut savoir que la plupart des **cartes RAID matériels** embarquent un mini-système Linux Embedded pour gérer les composants électroniques qu'elles contiennent : il s'agit la encore d'une **surcouche logiciel qui est masqué !**

Si vous ne parvenez pas à identifier sur quel périphérique de type bloc ou **« device » (/dev/xxx)** votre disque est placé, une fois de plus reportez-vous **aux fichiers « /var/log/message » ou « dmesg » :** ils contiennent généralement l'information qu'il vous manque pour retrouver le nom exact de votre disque vu par le noyau Linux.

Information : Sur des serveurs disposant d'un ou plusieurs disques durs SATA, SCSI ~~**sans RAID matériel**~~ vous pouvez utiliser les commandes suivantes qui agissent directement sur le disque dur (vu qu'il n'y a pas de surcouche logiciel vs RAID matériel) :

- **hdparm**
- **sdparm**

Ces 2 commandes permettent de faire des tests de vitesse, i/o, tunning etc. mais aussi de régler certains paramètres de bas niveau (dans le cas où vous accédez au disque dur physique).



Pour aller plus loin il existe l'outil : Bonnie++, il permet de faire des tests assez poussés en ce qui concerne les entrées/sorties (I/O) de vos serveurs.

Le partitionnement

Les types de partitions

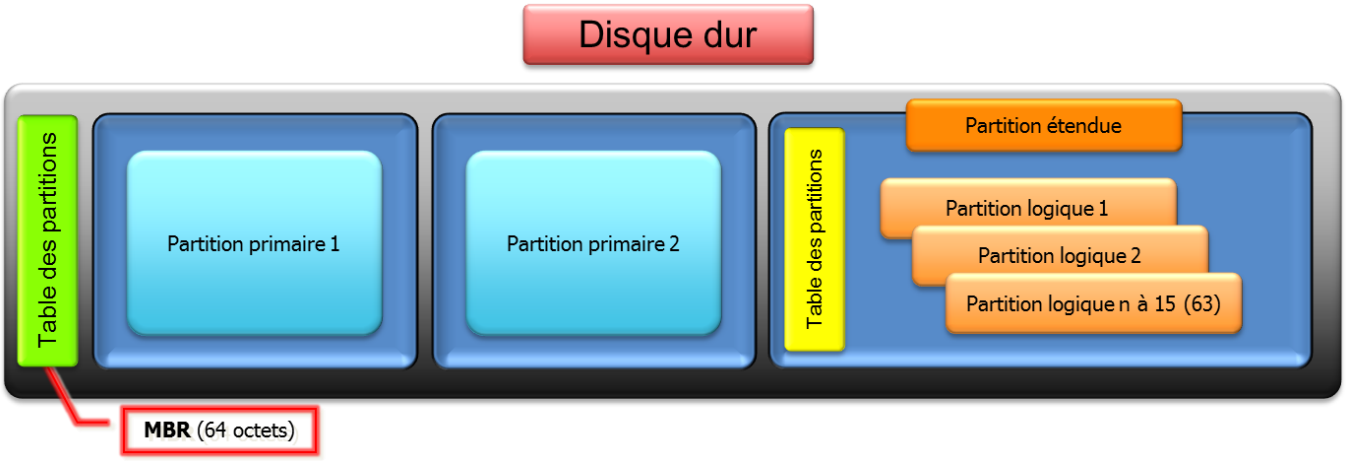
De façon simplifiée, il s'agit du **fractionnement d'un disque dur** en plusieurs « disques virtuels » appelés **partitions**.
Avec la technologie Intel, via le MBR, vous disposez de **4 partitions primaires** (définies dans les 4 entrées des **64 octets du MBR**).

Pour palier cette limitation la notion de **partition primaire étendue** a été introduite. On les appelle couramment **partitions étendues**.

Par disque, il est donc possible de reserver une des 4 partitions primaires et de la définir **comme partition étendue**. **Dans cette partition étendue** vous pouvez définir à nouveau des nouvelles **partitions dites « logiques** », avec un **maximum de 63 en IDE et 15 en SCSI (ou via la libata).**
Il faut savoir que la limite actuelle est de 15 partitions / disques durs pour tous les types de disques avec les derniers noyaux et l'API « **libata ».**

Attention : **Un disque dur** ne peut contenir **qu'une seule partition étendue**.
Pour **dépasser ces limites** en nombre de partition il est possible d'utiliser **LVM** que nous étudierons plus loin dans ce chapitre

Voici un exemple de partitionnement simple :



Pour aller plus loin …

Il existe également le **partitionnement** en technologie **GPT** (**G**uid **P**artition **T**able). Ce type de partitionnement sera, à l'avenir, le plus utilisé.
En effet le partitionnement en mode GPT permet de s'affranchir des **limites de la technologie MBR** et de :

- **Dépasser les 2,19 To** par partition,
- **Redonder** l'entête/descripteur et la table de partition en début et fin de disque,
- N'avoir aucune contrainte en **nombre de partition**,
- Travailler avec **l'EFI** (**E**xtensible **F**irmware **I**nterface),
- Etc.

Sous GNU/Linux les **disques durs** sont repérés par **une lettre** et les **partitions** sont **numérotées de 1 à 15 (ou 63)**.

Exemples :

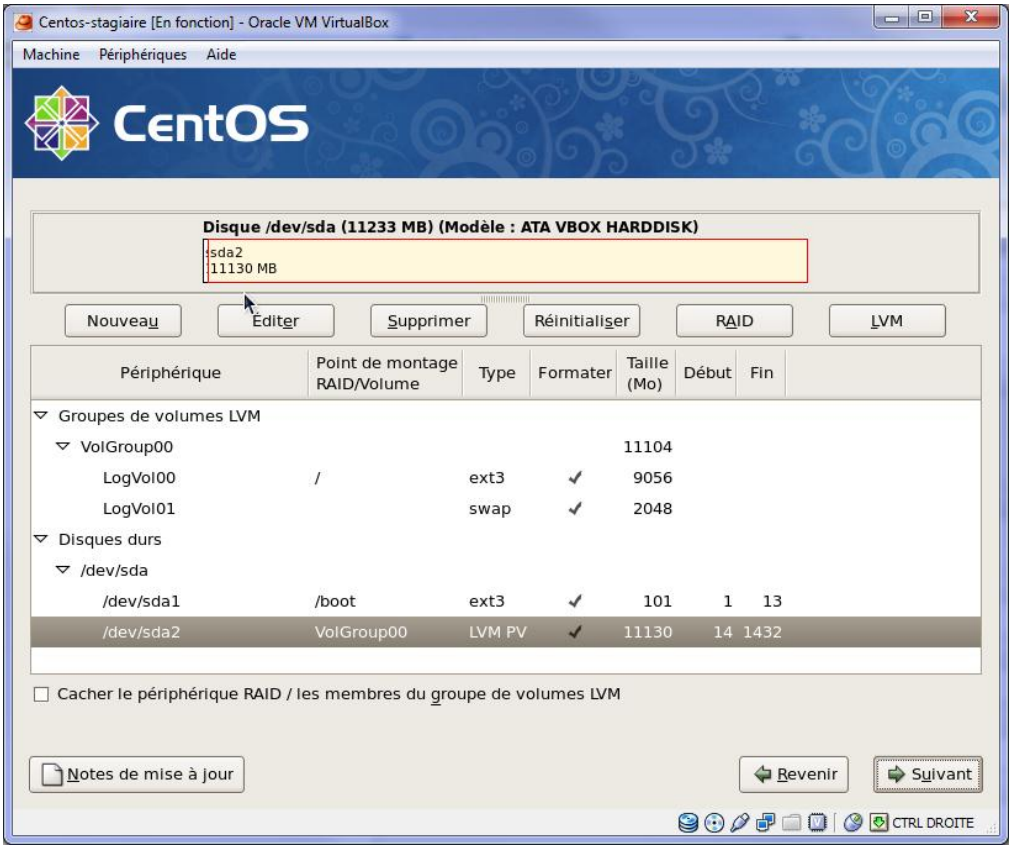
hda1 : première partition primaire du premier disque IDE,
hdb5 : cinquième partition, première partition logique du second disque IDE,
sda1 : première partition primaire, du premier disque SCSI / libata,
sda2 : deuxième partition primaire, du premier disque SCSI / libata,
sdc4 : quatrième partition primaire, du troisième disque SCSI / libata

La gestion des partitions

Vous pouvez manipuler les partitions avec un large éventail d'outils, en voici quelques-uns.

Mode graphique :

- diskdruid (uniquement durant la phase d'installation d'une distribution Fedora, RedHat©, Centos),
- gparted.



Mode texte :

- fdisk (installé de base),
- parted,
- cfdisk,
- sfdisk.

Tout comme « Vi », **« fdisk »** présente l'énorme avantage d'être présent sur tout GNU/Linux dès l'installation et oblige l'utilisateur à savoir exactement ce qu'il fait. Nous allons donc travailler avec cet outil universel.
Il est possible de lancer **« fdisk »** avec l'action à réaliser :

- **fdisk -[options] <périphérique>**

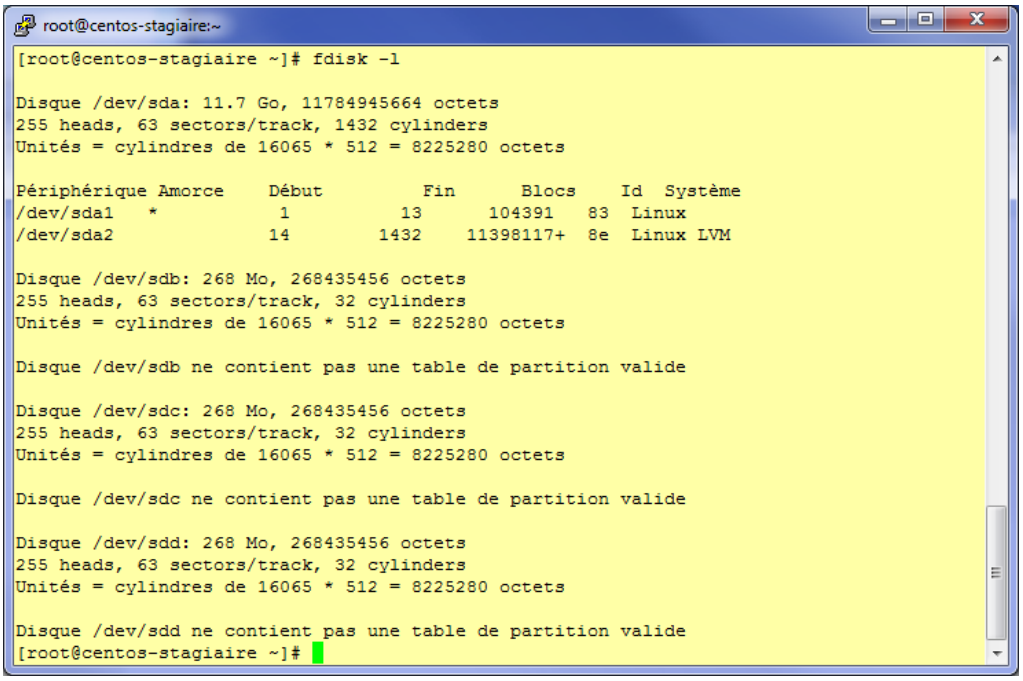
Exemple :

- **fdisk -lu <périphérique>** **unité secteur (vs cylindre)**

- **fdisk -s <périphérique>** **taille de la partition en bloc**

Commençons par **lister tous les disques durs et partitions** reconnus par le noyau, avec l'option **« -l »**.

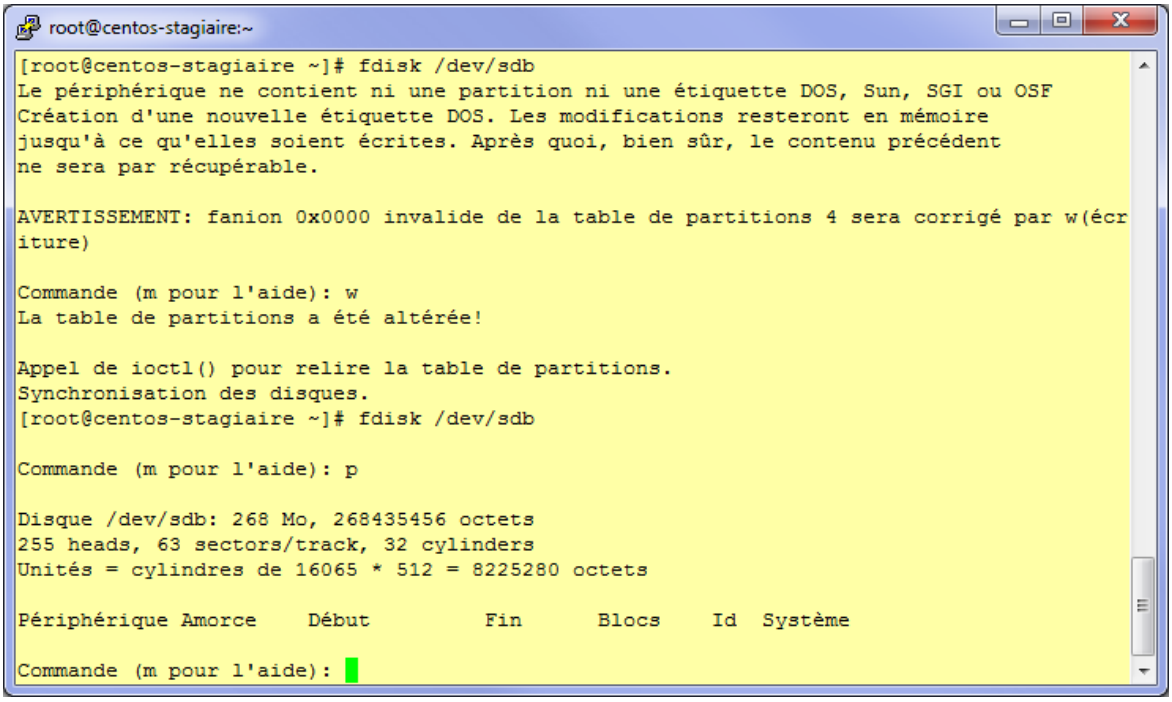
- **fdisk -l**



Ce mode non interactif est à utiliser une fois que vous avez validé votre ligne de commande car il agit immédiatement sur le périphérique (disque dur, Ex. : /dev/sdx) indiqué en paramètre.

Le mode interactif est vivement **conseillé**, pour ce faire procédez comme suit pour travailler sur le périphérique pointé par /dev/sdb :

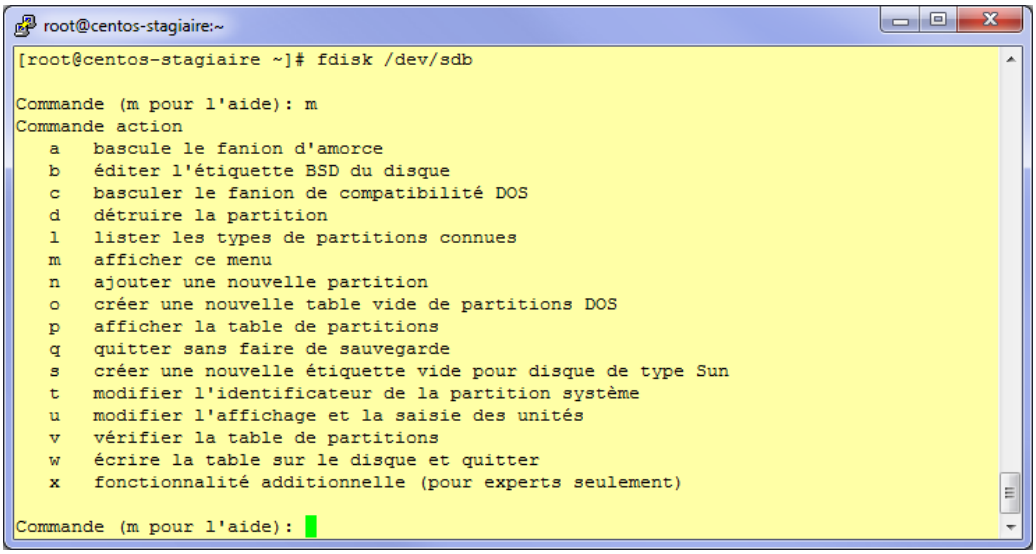
- **fdisk /dev/<périphérique>**



En général un nouveau disque dur (ici virtuel) ne contient pas de table de partition, **fdisk** le signale et vous invite à corriger ce « défaut ».
Pour se faire il suffit de saisir **« w »,** pour étiqueter au format DOS (MBR) un nouveau disque dur : attention car une fois saisie, cette **commande altère/crée irrémédiablement** la table de partition de ce disque dur.

Ensuite vous pouvez afficher les caractéristiques de votre disque sans erreur.
En saisissant **« p »**, vous pouvez afficher la table des partitions du disque. Vous disposez de toutes les informations utiles pour commencer le partitionnement.

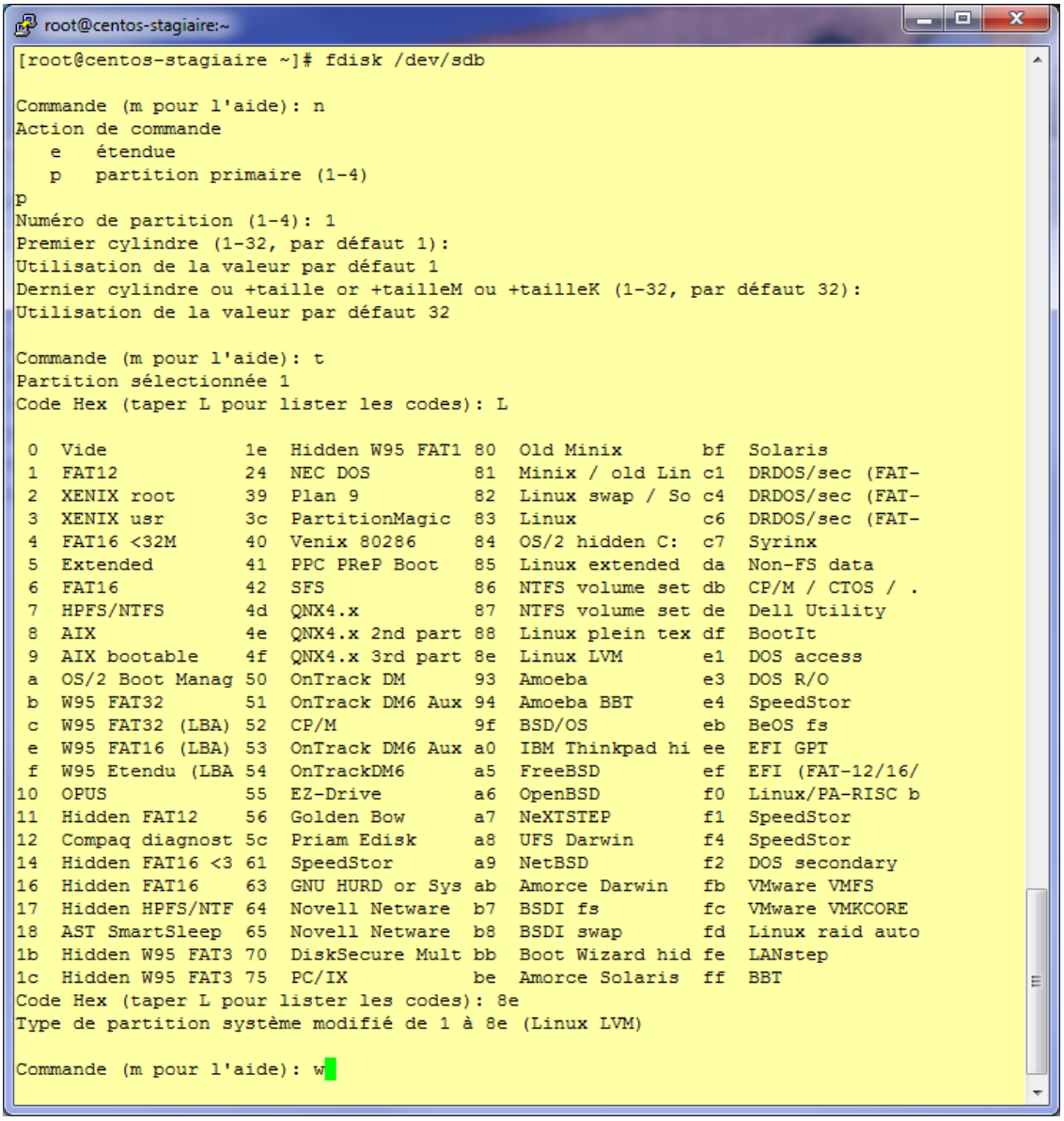
Voici les commandes disponibles avec **« fdisk », elles sont réellement appliquées lorsque vous ordonnez l'écriture de la table de partitions à l'aide de la commande « w » :**



Vous vous servirez essentiellement des commandes suivantes pour :

Commande	Fonctionnalité
A	Permet de rendre active une des 4 partitions présentes (Amorce pour le Boot Loader)
C	Permet de positionner l'étiquette DOS (MBR) d'une partition (vs techno GPT)
T	Permet de définir le type de partition (Linux, LVM, RAID, Mac, BSD, GPT etc.). Par défaut type Linux (83)
D	Permet de supprimer une partition
L	Lister les partitions du disque
M	Affichage du menu des commandes disponibles
N	Ajoute une nouvelle partition
P	Affiche la table des partitions telle quelle est ou lorsque elle sera écrite sur le disque dur avec **« w »**
Q	Quitter sans appliquer les changements effectués
v	Vérifie l'intégrité de la table de partition manipulée
w	Applique les changements sur la table de partition et sort de fdsik.

Exemple : Voici l'enchaînement de commande qu'il faut saisir pour créer une **partition primaire de type Linux LVM**.
Les types de partitions les plus utilisés sous GNU/Linux sont : **83, 82, fd et 8e**.



Note : Vous pouvez également visualiser les partitions en vous servant de l'arborescence virtuelle utilisée par le noyau Linux **/proc** :

- **cat /proc/partitions**

Note : Si votre nouvelle partition se situe sur le disque de démarrage de votre système il est nécessaire de demander au noyau Linux de relire la table de partition avec **« partprobe »**, outil contenu dans le paquetage **« parted »** ou bien avec **« blockdev »** :

- **partprobe /dev/<périphérique>**

Ou encore

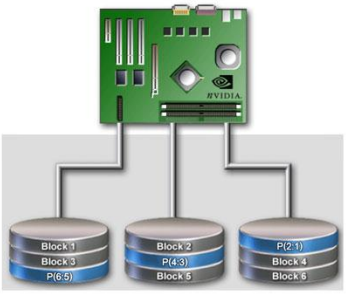
- **blockdev --rereadpt /dev/<périphérique>**

Veuillez noter qu'à ce stade nous définissons uniquement un **espace de stockage avec un type cependant** le **système de fichiersn'est pas encore créé**. Par exemple il est absurde, mais possible, de choisir le type **« e » (W95 FAT 16)** et de créer dans cette partition un système de fichiers**« ext3 » : Linux sera quand même capable d'exploiter les données** qui seront contenues dans cette partition.

Le partitionnement est un sujet largement débattu.
On conclue souvent qu'il n'existe **pas un partitionnement universel** mais **autant de stratégie de partitionnement qu'il existe de situations ou besoins donnés** : c'est en partie vrai.
Surtout quand on se cantonne au domaine des partitions telles qu'envisagées par la technologie Intel MBR.

Dans la section RAID et LVM nous verrons que l'intérêt de créer moult partitions à ce niveau devient obsolètes : les FAI et Datacenter ne travaillent plus de cette manière.

Le RAID logiciel de GNU/Linux



Pourquoi le RAID ? … la **tolérance de panne** !
(Redundant Array of Independent Disks)

Son but est **d'assurer la disponibilité des données en cas de défaillance d'un ou plusieurs disques durs** : donc la **continuité de service**. Donc, en aucun cas le RAID ne remplace la sauvegarde de vos données.

Le RAID n'est pas une sauvegarde, la sauvegarde est la première tâche que l'administrateur système doit faire/vérifier en arrivant le matin.

Par exemple deux disques en miroir ne constituent pas une sauvegarde : un disfonctionnement électrique dans le fonds de panier d'une baie de disques peut très bien griller tous les disques durs qui y sont logés.

Le RAID permet de laisser le temps aux administrateurs de changer le ou les disques durs défaillants (en échecs), encore faut-il avoir pris connaissance de la défaillance. D'où l'intérêt de disposer **d'outils de surveillance centralisée de vos matériels** (serveurs, baies de disques, San etc.)

Un des gros inconvénients du **RAID matériel** provient de la carte **contrôleur qui est propre à chaque constructeur**. En effet si jamais elle vient à tomber en panne votre seul salut, pour récupérer les données de votre grappe RAID, est de la **remplacer par une carte identique**. Si cette carte est introuvable … seule la sauvegarde la plus récente pourra vous sauver.

C'est là qu'intervient le **RAID logiciel proposé par GNU/Linux**.
Certes les performances ne sont pas forcément du même niveau qu'un contrôleur matériel dédié RAID mais votre **grappe RAID pourra être reconnue sur un autre matériel**, serveur, baie etc. en cas de défaillance du matériel existant : c'est donc un avantage indéniable.
En cas de **souci de budget pour l'acquisition d'un contrôleur RAID matériel** (achat, garantie, maintenance) cette **solution est intéressante**.

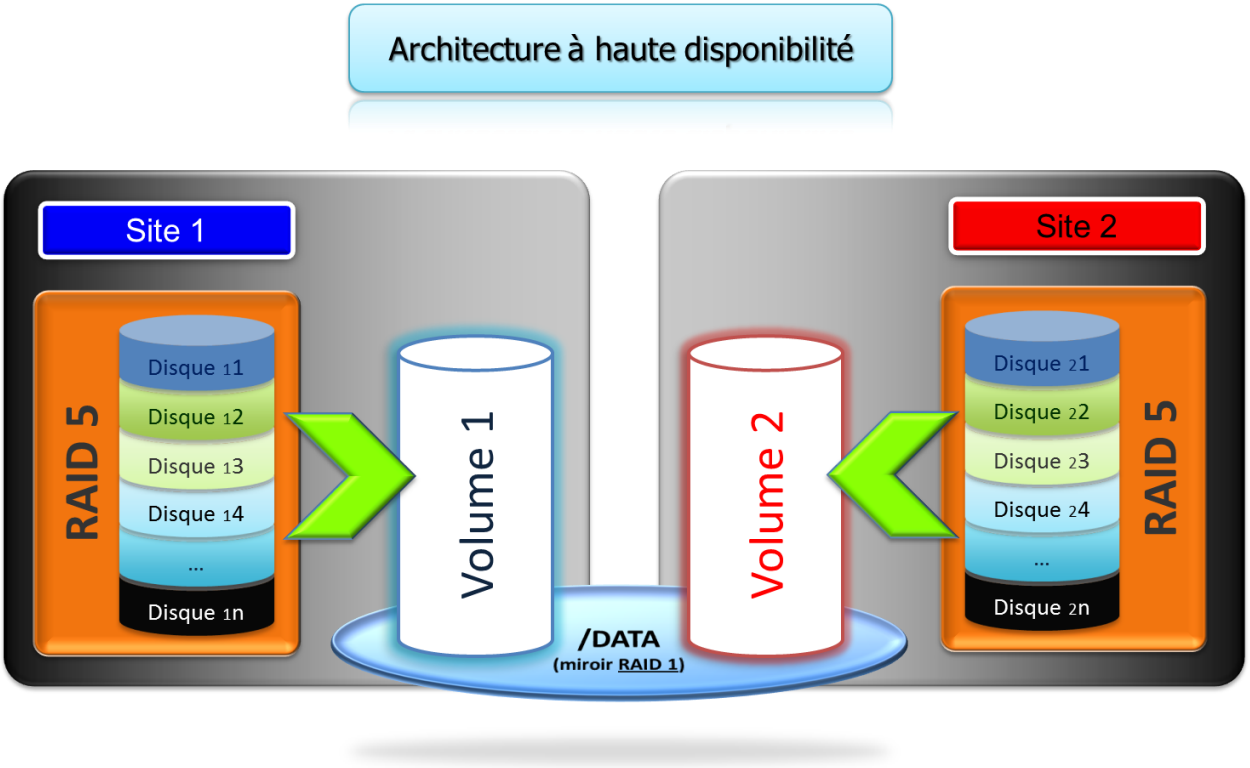
De plus cela permet comprendre le concept de fonctionnement d'une carte RAID matériel.

Ici le **but** va être de disposer d'un **périphérique** (Ex. : **/dev/md0)** qui soit une **grappe RAID de niveau 5.** Cette dernière sera présentée comme un disque dur simple au noyau, disposant d'une **tolérance de panne**.

Rappel RAID5 : Il faut minimum 3 disques dur pour faire un RAID 5, dans ce cas la continuité de service étant assurée si on perd 1 disque (n-1)

Pour plus de détail sur les différents RAID possibles veuillez consulter le site internet suivant : http://fr.wikipedia.org/wiki/RAID_(informatique)

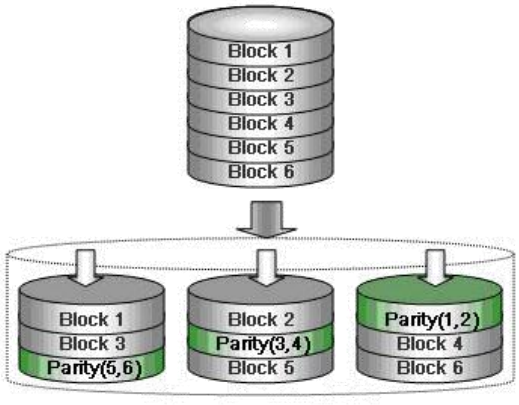
Sachez juste qu'en générale pour disposer d'une architecture à haute disponibilité on travaille en combinant plusieurs niveaux de RAID (0, 5, etc.) et technologie matériel (SAN, SAS, FC etc.)



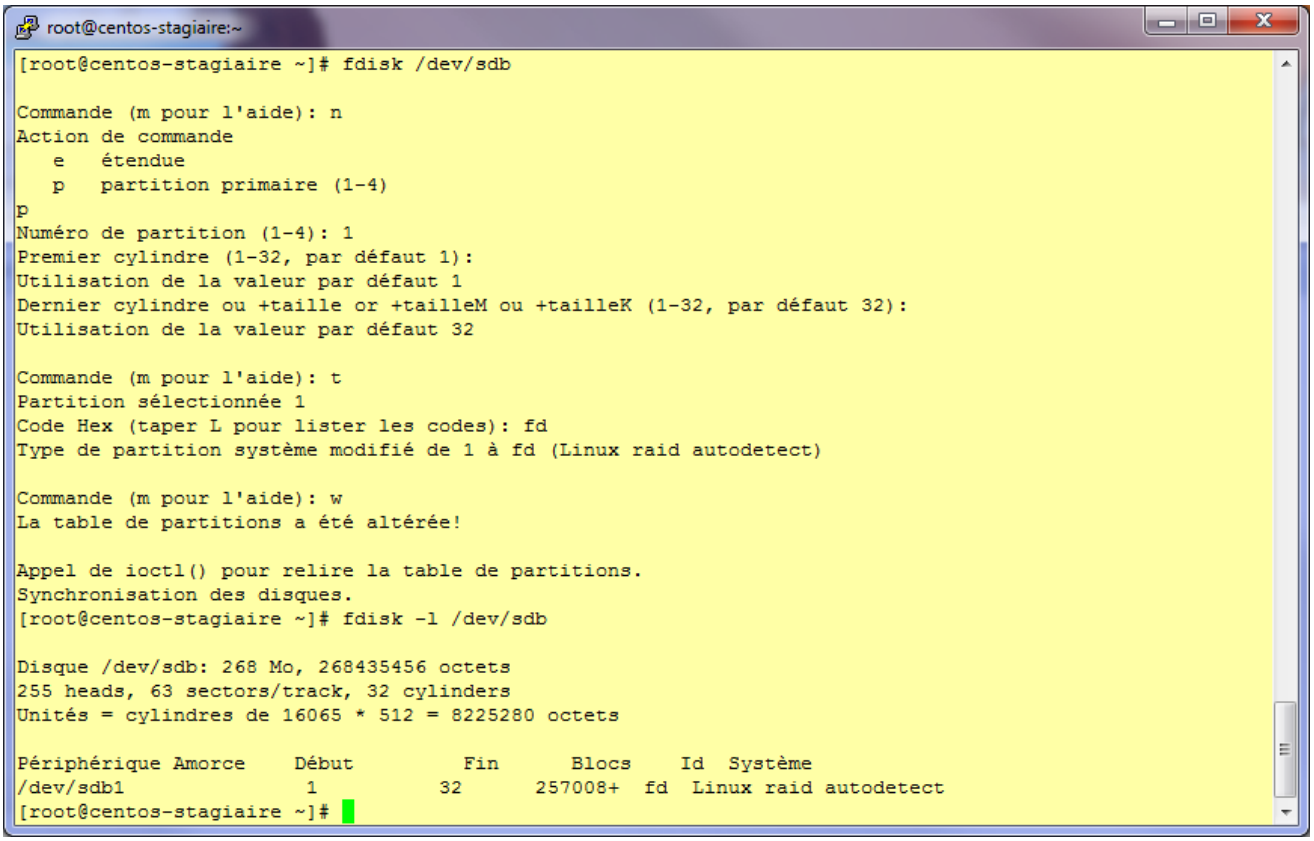
Pour créer un raid 5 logiciel vous devez :

- Disposer de **3 disques minimum** disposant de partition de **type « Linux raid »**
- Avoir l'outil **« mdadm »,**
- Mettre à jour le fichier de montage des systèmes de fichier : **»/etc/fstab ».**

Un peu de théorie sur le RAID 5 : On constate qu'en cas de perte d'un disque le système peut encore fonctionner, les blocs étant ingénieusement répartis.



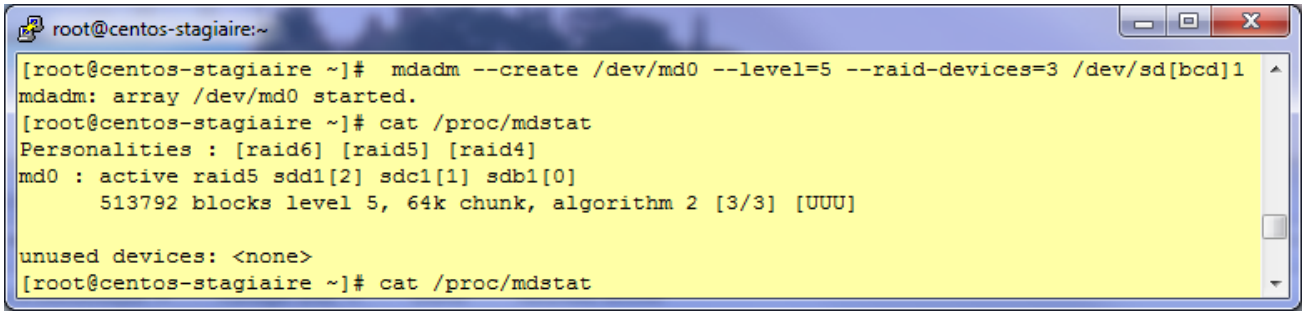
Passons à la pratique, tout d'abord vous devez **créer une partition « Linux raid »**



Après avoir réalisé cette opération sur 3 disques vous avez à votre disposition trois **partitions primaires** de **type « Linux raid** » nommées :

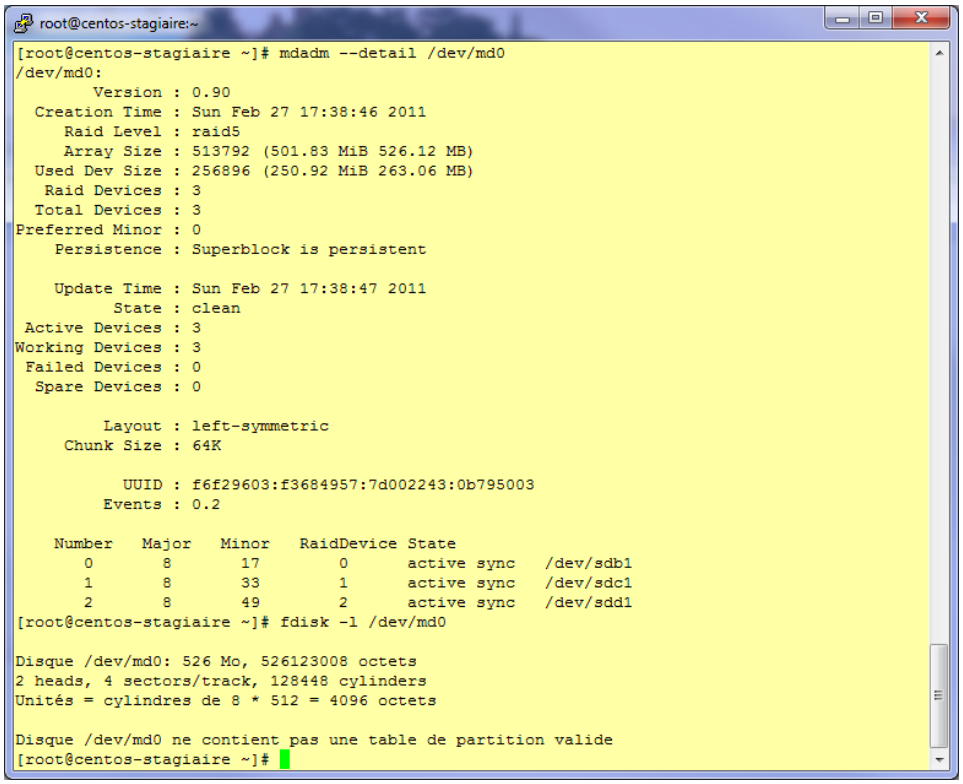
- /dev/sdb1,
- /dev/sdc1,
- /dev/sdd1.

Maintenant il faut créer le périphérique raid de niveau 5 **/dev/md0 à l'aide de commande « mdadm »** :



Vous disposez d'un nouveau périphérique **« /dev/md0 »** à tolérance de panne niveau 5 (sur lequel il n'y a aucune partition ni système de fichiers). La commande **« mdadm »** peut fournir beaucoup d'informations concernant l'état du système RAID.
Pour connaître l'état globale et détaillé de votre grappe RAID 5 :

- **mdadm --detail /dev/md0**



Pour simuler une panne vous pouvez mettre un disque constituant votre grappe RAID 5 en erreur :

- **mdadm /dev/md0 -f /dev/sdb1**

La commande **« watch »** permet, tout comme **« tail -f »**, de visualiser en temps réel (rafraîchissement à 2 secondes par défaut) les changements qui s'effectuent dans un fichier. Ici il est intéressant de saisir :

- **watch cat /proc/mdstat ou watch –d ls -l /proc/mdstat**

Vous constatez que l'un de vos disques est indiqué comme (F), Faulty, en erreur donc.

En saisissant :

- **mdadm --detail /dev/md0**

Vous obtenez une information plus détaillée indiquant que votre grappe **RAID 5 fonctionne mais en mode dégradée**.

Nous allons donc devoir changer ce disque dur défaillant en le **retirant** puis en le **remettant** en place (ici nous ferons que remettre le disque en place de façon logiciel). Cela se traduit donc par les manipulations suivantes :

- **mdadm /dev/md0 -r /dev/sdb1**

- **mdadm /dev/md0 -a /dev/sdb1**

Vous pouvez afficher les détails pour suivre en continue la reconstruction du RAID 5 sur l'ensemble de la grappe RAID 5.

Note: En formation, sur des disques de faibles capacités, les temps pris par les calculs XOR (calcul de parité) sont rapides, en production une **reconstruction suite à un sinistre** peut **prendre une nuit** entière. Cela **dépend de la volumétrie** hébergée sur votre périphérique RAID.

Lors d'une reconstruction RAID (logiciel comme matériel) les performances en terme d'entrée/sorties (I/O Bench) sont dégradées mais cela est tout à fait normal.

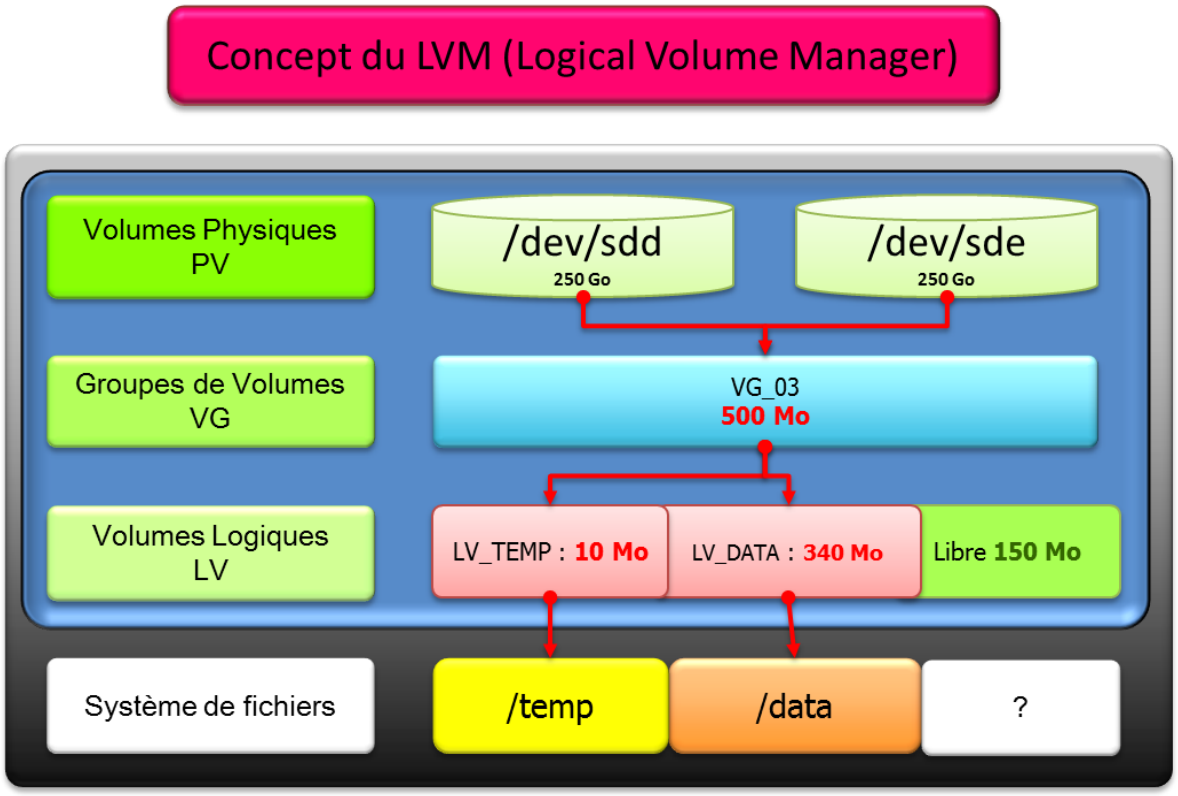
Pour aller plus loin :
De nos jours un des concurrents prometteur du RAID est le **ZFS** : **Z**eta **F**ile **S**ystem. Ce système de fichiers combine RAID et LV : c'est-à-dire redondance et souplesse de maintenance du système de fichier.

La technologie LVM (version 2)

Logical **V**olume **M**anager : gestionnaire de volume logique.

Les volumes logiques (LV) permettent de s'affranchir de la lourdeur et de la rigidité imposées par la gestion des partitions donc du sous-système disque de bas niveau (disque dur, RAID matériel ou logiciel, SAN etc.) : on bascule dans une **organisation logique et non plus physique des partitions**.

Il s'agit donc d'une **couche d'abstraction logicielle qui permet de réduire ou d'augmenter la taille d'un à la volée**.
Cela va même plus loin, car vous pouvez également ajouter des ressources de stockage supplémentaires et les intégrer après coup dans le(s) Groupe(s) de Volume.



Volume Physique **(PV)** : une représentation logique d'un support de stockage dit physique,

Groupe de Volume **(VG)** : un regroupement logique de 1 à n **PV**,

Volume Logique **(LV)** : un découpage logique (une partition) au sein d'un **VG**.

Ne perdez jamais de vu le schéma ci-dessus quand vous manipulerez les volumes logiques. Car lorsque vous souhaiterez réduire ou agrandir une partition vous devrez toujours penser comme suit :

- ***Réduction ou agrandissement de l'enveloppe : le volume logique,***
- ***Puis Réduction ou grandissement du contenant : le système de fichier.***

Une des questions récurrente est la suivante :
Que se passerait-il en cas de perte d'un disque dur (Ex. : /dev/sdd sur le schéma précédent) **?**

⇨ Réponse simple : vous perdriez toutes vos partitions donc « / », et plus grave « /home ».

Donc il faut **éviter de comparer les LV au RAID** ce sont deux technologies dont les finalités sont différentes.

- le **RAID** permet d'obtenir la **tolérance de panne**,
- le **LVM** permet de gagner en **souplesse de manipulation du système de fichier**.

Donc assurez la tolérance de panne au niveau du RAID (bas niveau) et, seulement ensuite, intégrez le périphérique fraîchement créé (ou étendu) dans le LVM.

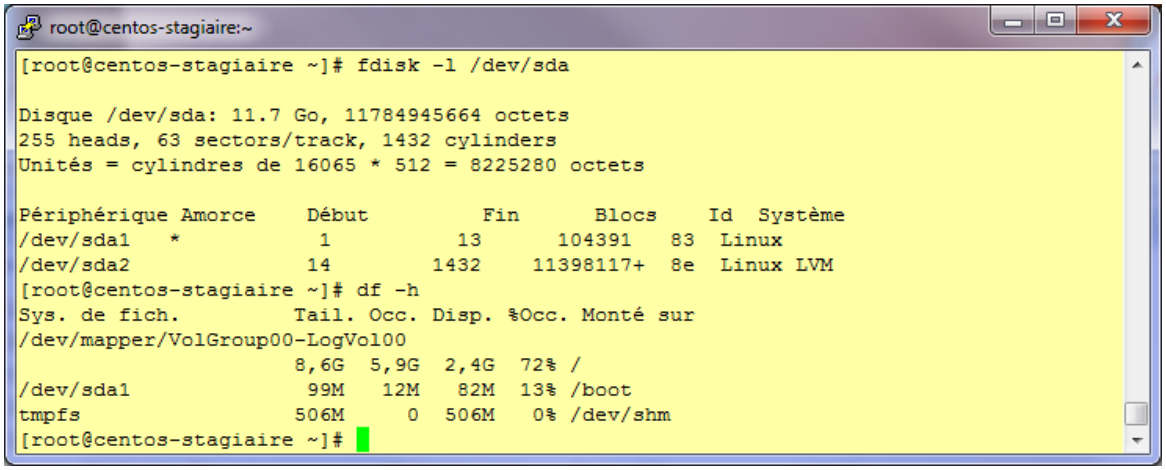
Et comme toujours : sauvegarde, sauvegarde … test de restauration, sinon vous travaillez sans filet.

Dès que l'on commence à travailler avec des solutions de stockages professionnels, où le **disque dur physique se situe à plusieurs niveaux d'abstractions** (SAN, RAID matériel ou logiciel etc.) du système de fichier, l'intérêt de créer plusieurs partitions du type :

- /boot = une partition,
- /var = une partition,
- /home = une partition,
- /usr = un partition,
- /tmp = une partition,
- Etc.

… devient plus contraignant et source d'erreur que gage de finesse et de souplesse.
Et comme dit précédemment : partitionner pour gagner en performance n'est généralement plus d'actualité sur des systèmes de stockage professionnel (SAN, RAID matériel etc.).

Par exemple observons le partitionnement que proposent, par défaut, les intégrateurs de RedHat© quelque soit le sous-système disque physique vu par le noyau lors de l'installation de GNU/Linux :



La stratégie retenue est **simple** mais en même temps permet un très **grande souplesse pour la gestion du système de fichiers actuel** et surtout **futur** :

- La zone d'amorce du système est placée sur une **partition primaire amorçable de type « Linux »**. On y trouve le point de montage **/boot** qui contient **GRUB**(Stage 2), le **noyau Linux**, et **l'image INITRD**. Il s'agit du **minimum pour démarrer GNU/Linux :** elle ne **dépend pas du LVM**,

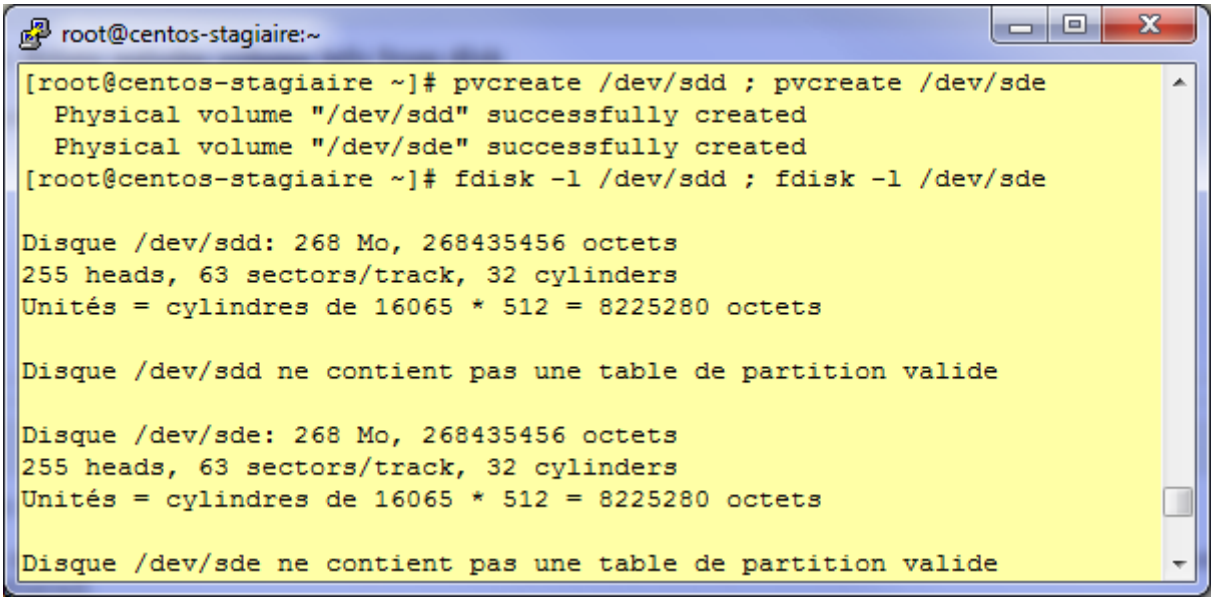
- Le **reste du système est placé** dans un ou des **volumes logiques appelés LVM** (Logical Volume Manager).
  Et c'est dans ces volumes logiques que nous allons retrouver des partitions avec les points de montages d'un système GNU/Linux usuels.

Voici maintenant venu le moment de la mise en œuvre. La configuration du LVM est stockée dans les fichiers et répertoires suivant : **« /etc/lvm/ »**. Cependant pour **manipuler les volumes logiques il faut passer par les commandes prévues** par la suite logiciel du LVM.
Note : Avec la technologie LVM, les unités s'expriment soit en **PE/LE** soit en **Go, Mo** etc.

Les volumes physiques

Vous pouvez créer un volume physique sur :
- Un disque dur complet (SAS, SATA, SCSCI, FC etc.),
- Un disque logique issu d'une grappe RAID logiciel ou matériel.

Voici comment créer 2 volumes physiques sur 2 disques durs (**« pvremove »** permet de les supprimer) :



Pour visualiser les volumes physiques présents sur tous les disques :

➢ **pvscan**

Pour visualiser les volumes physiques de façons détaillées :

➢ **pvdisplay**

Note : l'option -v, mode verbeux, est également disponible.

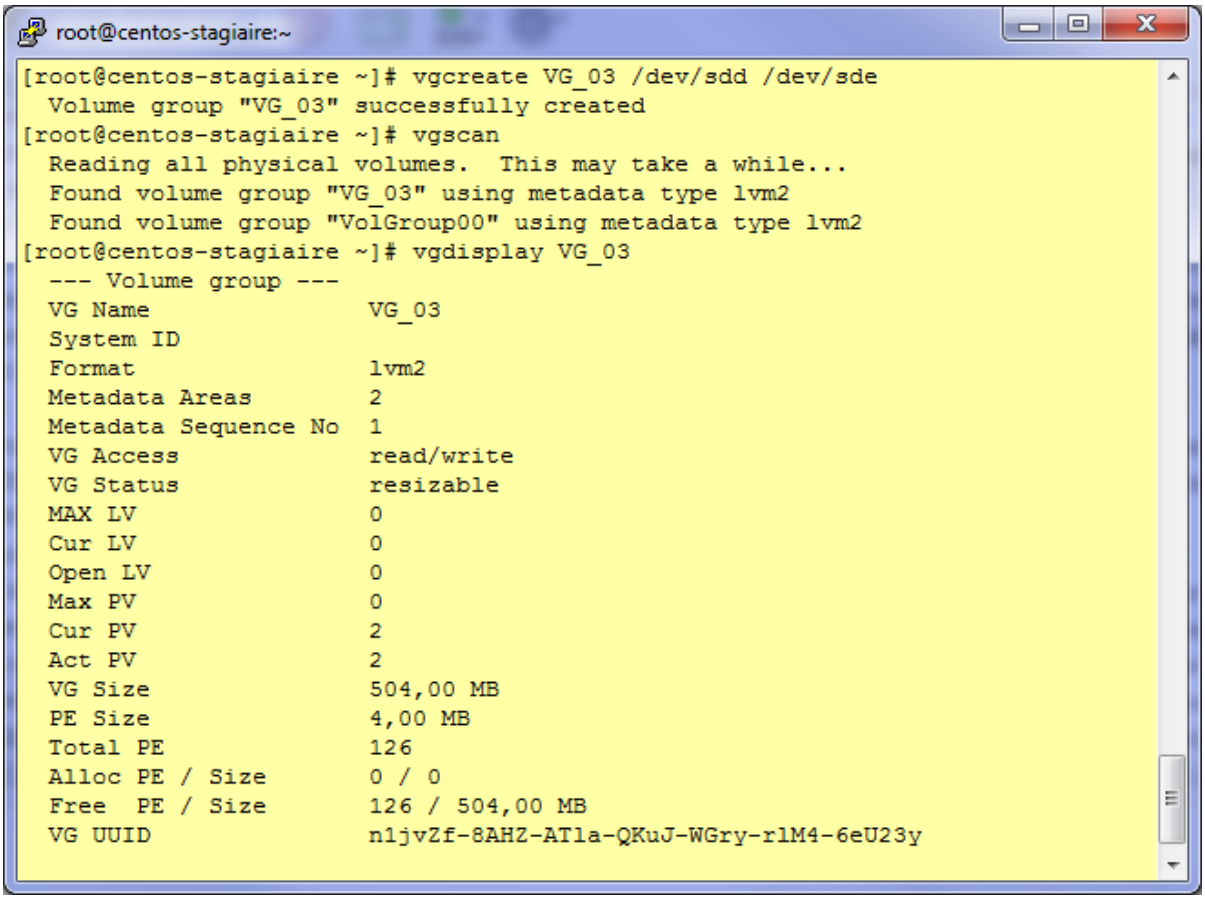
Le **PE** (**P**hysical **E**xtends) est l'unité de base de travail du LVM. Si un PE fait 4 Mo, cela signifie que l'espace pourra être découpé au sein du groupe de volume par tranches de 4 Mo. L'allocation se fait par PE : la création d'un volume logique de 500 PE de 4 Mo fait donc 2000 Mo.

Note : Dès que le PE sera dans un groupe de volume (VG) les champs dont la valeur est à zéro seront renseignés.

Les groupes de volumes

Pour pouvoir créer un groupe de volume vous devez disposer d'au moins un volume physique.

Voici comment créer un groupe de volume (VG_03) basé sur deux volumes physiques (/dev/sdd et /dev/sde) puis afficher sa description (**« vgremove »** permet de le supprimer) :



Pour visualiser les groupes de volumes présents sur tous les disques :

- **vgscan**

Pour visualiser les groupes de volumes de façon détaillée :

- **vgdisplay**

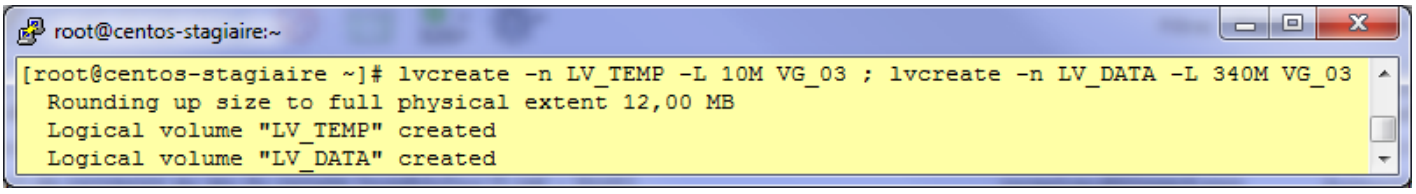
Vous pouvez supprimer ou ajouter des volumes physiques dans un groupe de volume avec les commandes respectives (voit TP pour manipulation) :

- **vgreduce**
- **vgextend**

Les volumes logiques

Un volume logique (LV) est un découpage d'un groupe de volume (VG). Chaque partie de ce découpage est vue par le noyau comme un périphérique dans lequel on peut créer une ou des partitions.
Dans ces partitions pourront ensuite être créés des systèmes de fichiers.
Ici on parlera de **LE** (**L**ogical **E**xtension) qui est la représentation des **PE** au sein d'un volume logique.

Voici comment créer deux volume logiques **« LV_TEMP » : 10 Mo** et **« LV_DATA » : 340 Mo** dans le groupe de volume **« VG_03 »**, dont la capacité est de **500Mo** (**« lvremove »** permet de les supprimer) :

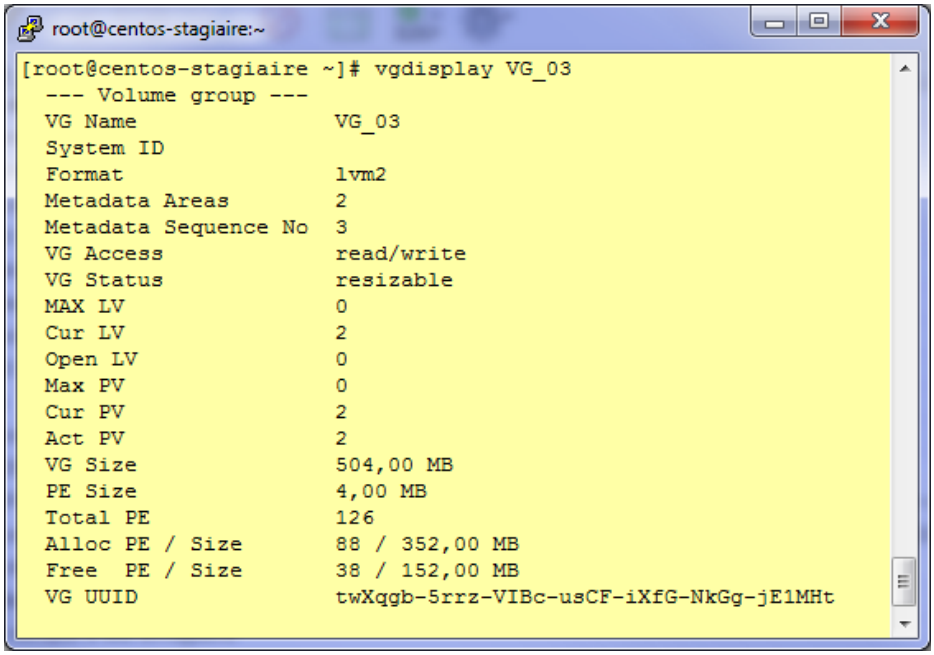


Une fois créé, un volume logique est un périphérique qui est référencé ainsi :

/dev/<nom_du_groupe_de_volume>/<nom_du_volume_logique>

Note : il s'agit d'un lien symbolique vers le « **device mapper** » (/dev/mapper/).

Voici l'espace disponible qu'il nous reste dans le groupe de volume « VG_03 » après création des 2 volumes logiques ci-dessus :



Pour visualiser les volumes logiques présents sur tous les disques :

- **lvscan**

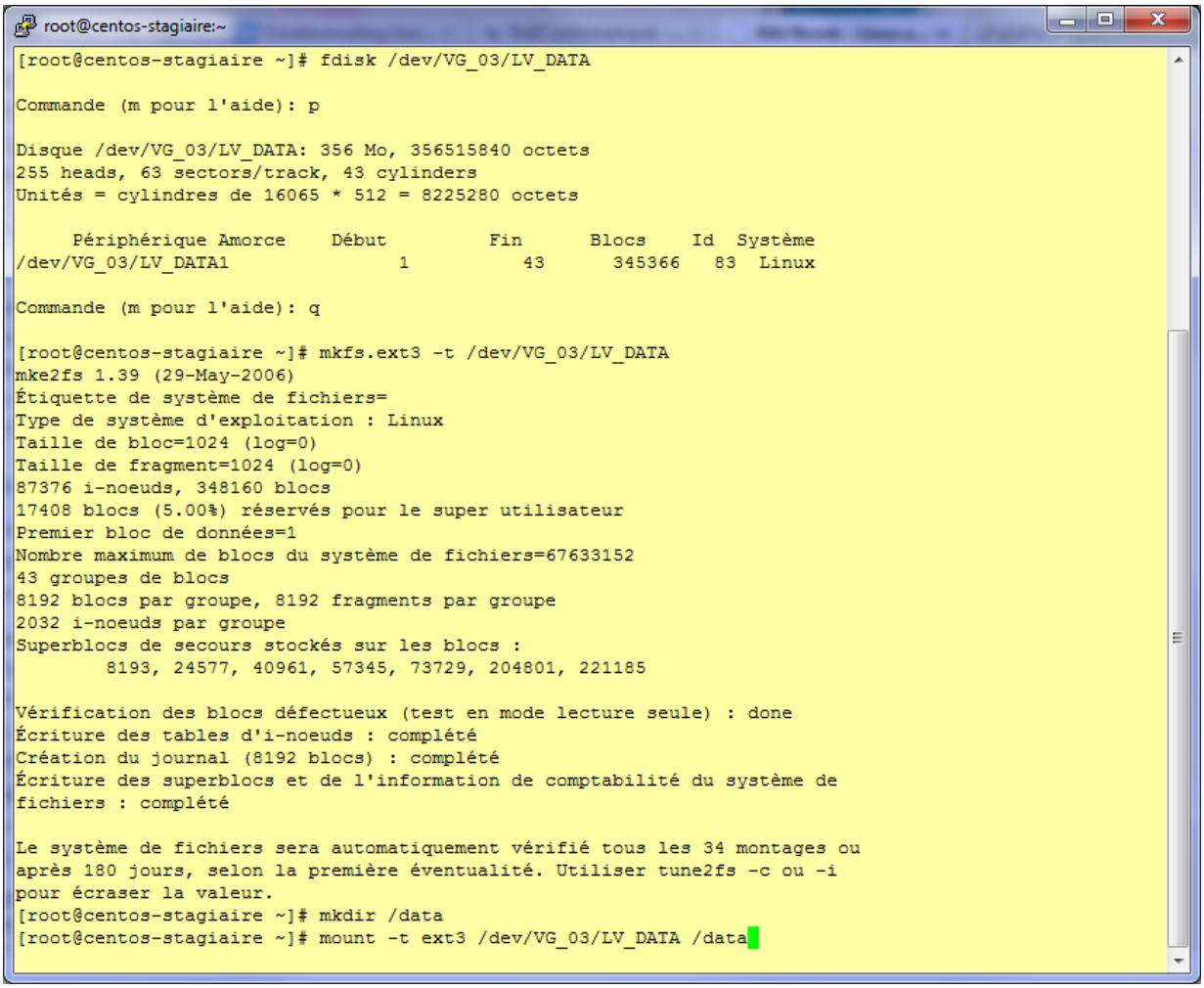
Pour visualiser les volumes logiques de façon détaillée :

- **lvdisplay**

Note : N'hésitez pas à utiliser le mode verbeux pour l'affichage des informations des LVM : **« -v »**

Les commandes « **pvs** », « **vgs** » **et** « **lvs** » permettent d'obtenir des informations sur LVM en choisissant les informations à afficher via une sélection par champs.

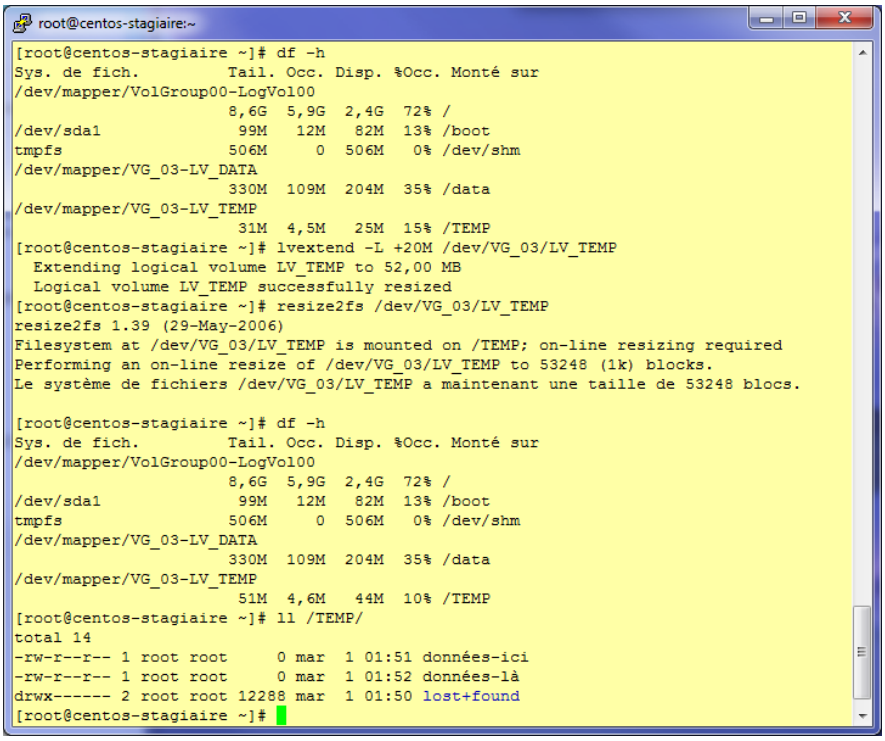
Maintenant nous allons créer une **partition de type 83 « Linux » dans le volume logique LV_DATA** du volume groupe **« VG_03 »**. Nous utilisons **« fdisk »** sur le périphérique **« /dev/VG_03/LV_DATA »** pour cela.
Puis nous allons créer et formater un **système de fichiers journalisé de type « ext3 »** et enfin le monter sous le répertoire **« /data »** :



Extension, réduction de volume logique

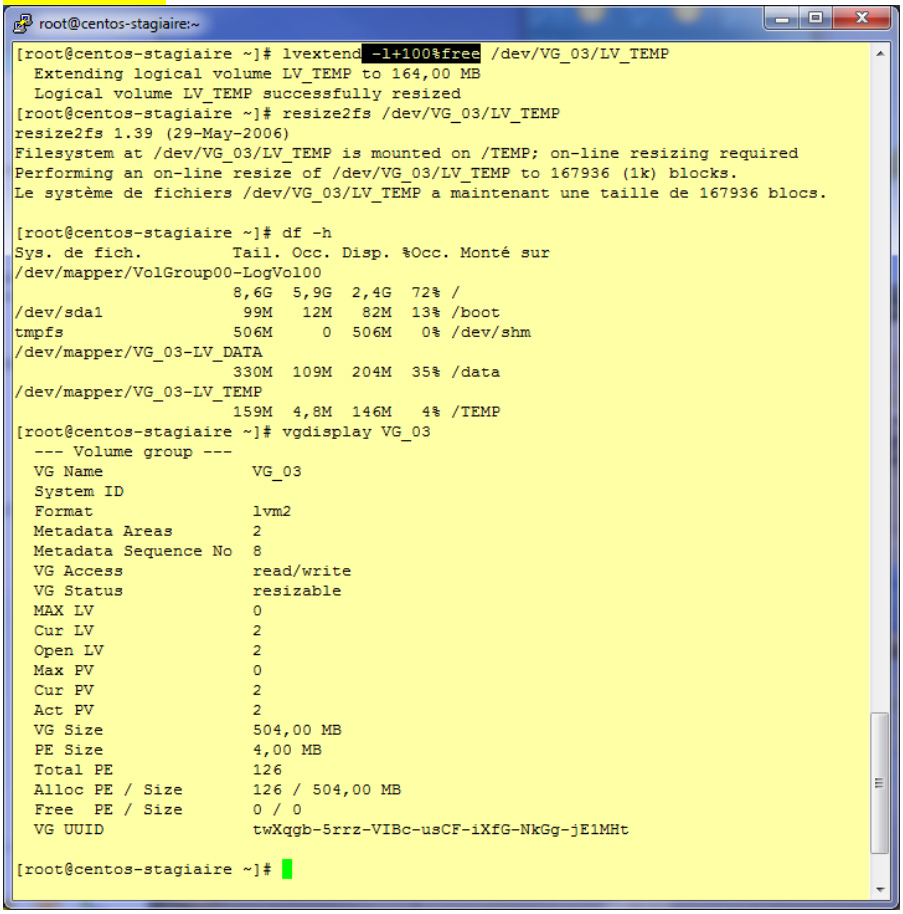
Voyons **comment agrandir un volume logique** en préservant bien entendu les données se trouvant dans la partition.
Voici les opérations à réaliser :

- Augmenter la taille de l'enveloppe, le volume logique (commande **« lxextend »**),
- Redimensionner le contenant, le système de fichiers (commande **« resize2fs »**).



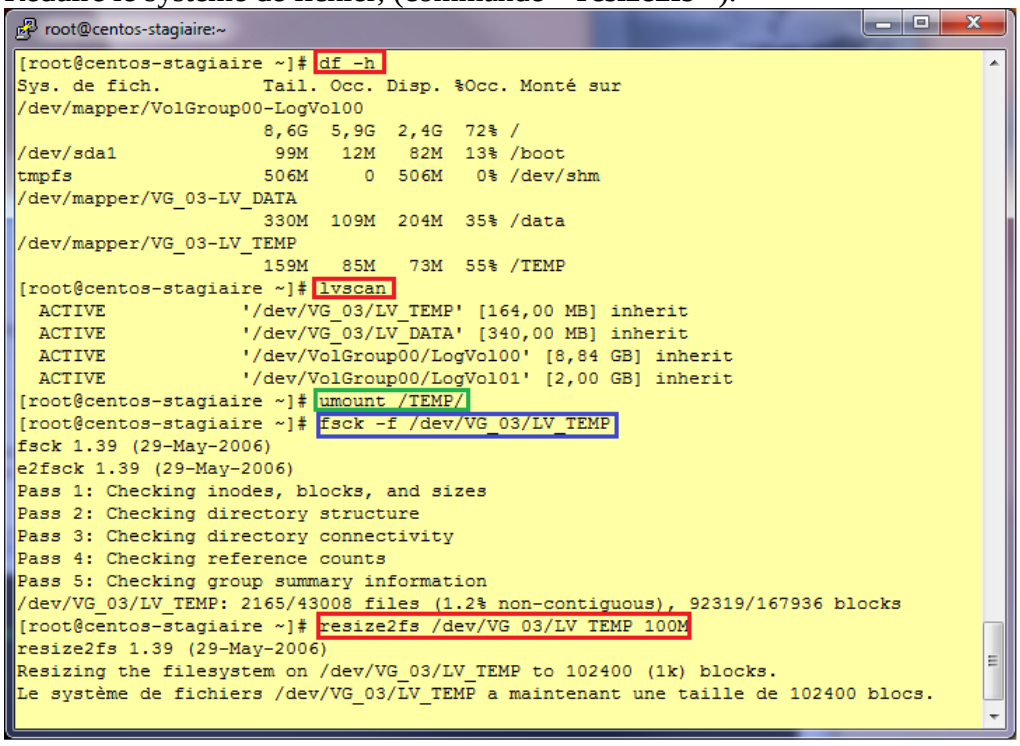
Ci-dessus on a utilisé l'option suivante :
-L+<taille_a_ajouter>[GMTPE] (avec la taille en Giga, Mega, Tera … byte)

On peut augmenter la taille en utilisant tout l'espace libre :
-l+100%free

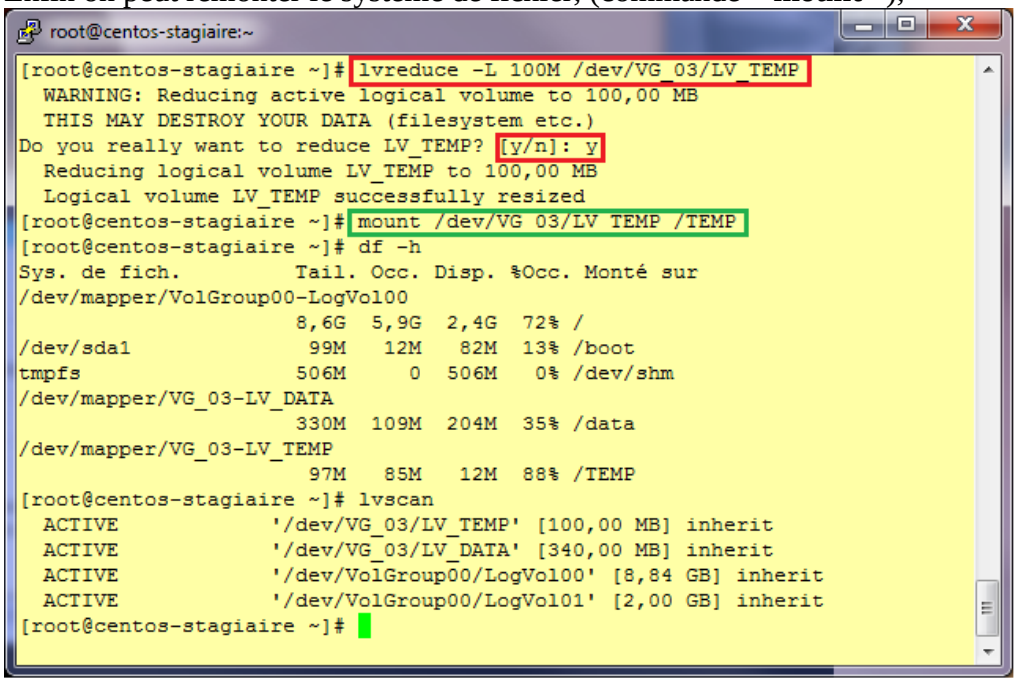


Voyons **comment réduire un volume logique** en préservant bien entendu les données se trouvant dans la partition. Veuillez noter qu'il est parfois obligatoire de passer en mode « Single User » pour pouvoir démonter la partition à réduire. **VOUS DEVEZ AVOIR UNE SAUVEGARDE EXPLOITABLE DU LV**
Voici les opérations à réaliser.

- Redimensionner le contenant, le système de fichiers :
  Démonter le système de fichier, (commande **« umount »**),
  Vérifier le système de fichier, (commande **« fsck »**),
  Réduire le système de fichier, (commande **« resize2fs »**).



- Réduire la taille de l'enveloppe, le volume logique (commande **« lvreduce »**),
  Enfin on peut remonter le système de fichier, (commande **« mount »**),



Astuce : Les commandes **« lsof »** et **« fuser »** vous permettent de connaître les processus qui verrouillent des fichiers présents sur une partition que vous voulez démonter.

Pour aller plus loin : Les **SnapShots**

Les *snapshots* sont des volumes logiques permettant d'effectuer une sauvegarde cohérente d'un autre volume logique du même groupe de volumes.

La création d'un *snapshot* consiste à prendre une « photo », un instantané du volume logique cible (ce qui est quasi-immédiat) et on commence alors à enregistrer les modifications apportées au volume logique cible.

Avantage des *snapshots*

Ils peuvent être utilisés comme une méthode de sauvegarde. Ils permettent de stocker une image statique d'un volume logique à un instant précis. On peut ensuite effectuer une sauvegarde sur cartouche du *snapshot*, qui contiendra les données présentes sur le volume cible au moment de la création du *snapshot*.

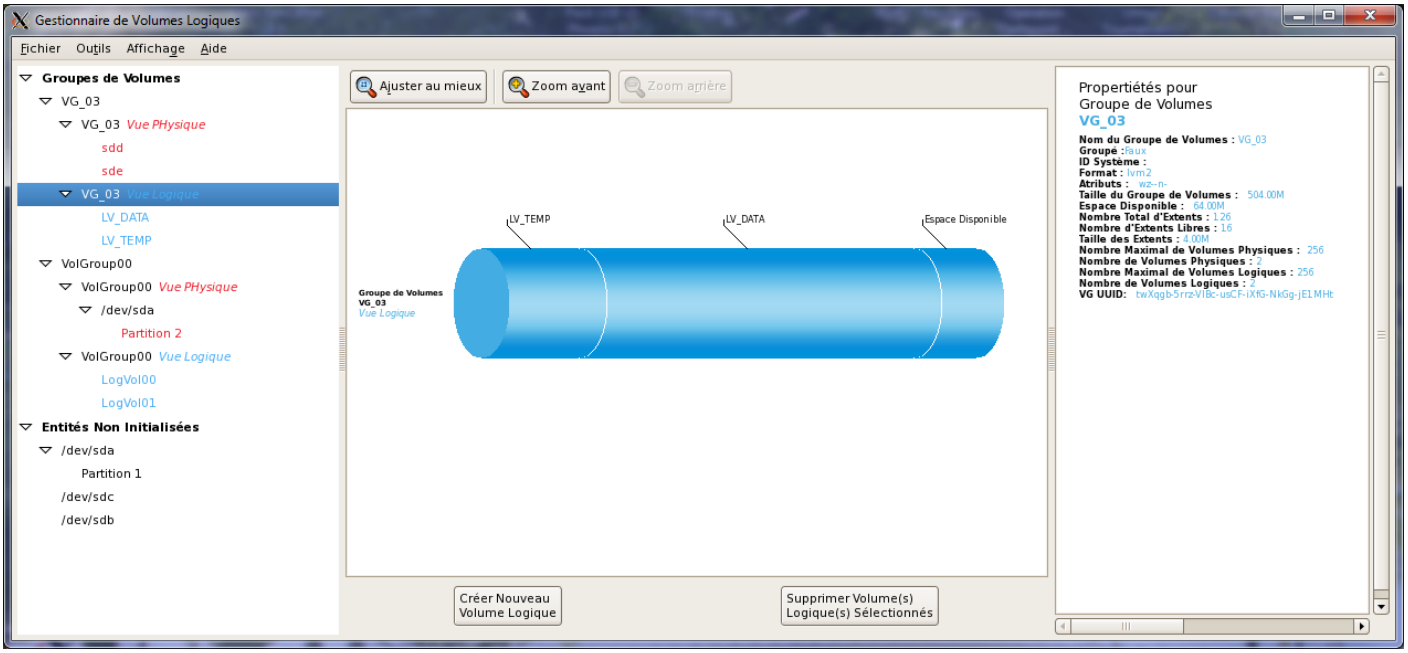
Limite des *snapshots*

Ils **ne sont pas** une sauvegarde complète d'un volume logique, ils enregistrent uniquement les modifications apportées au volume cible, ils ne contiennent pas les données de celui-ci ; de plus ils ne sont pas persistants, c'est-à-dire qu'ils disparaissent en cas de redémarrage de la machine.

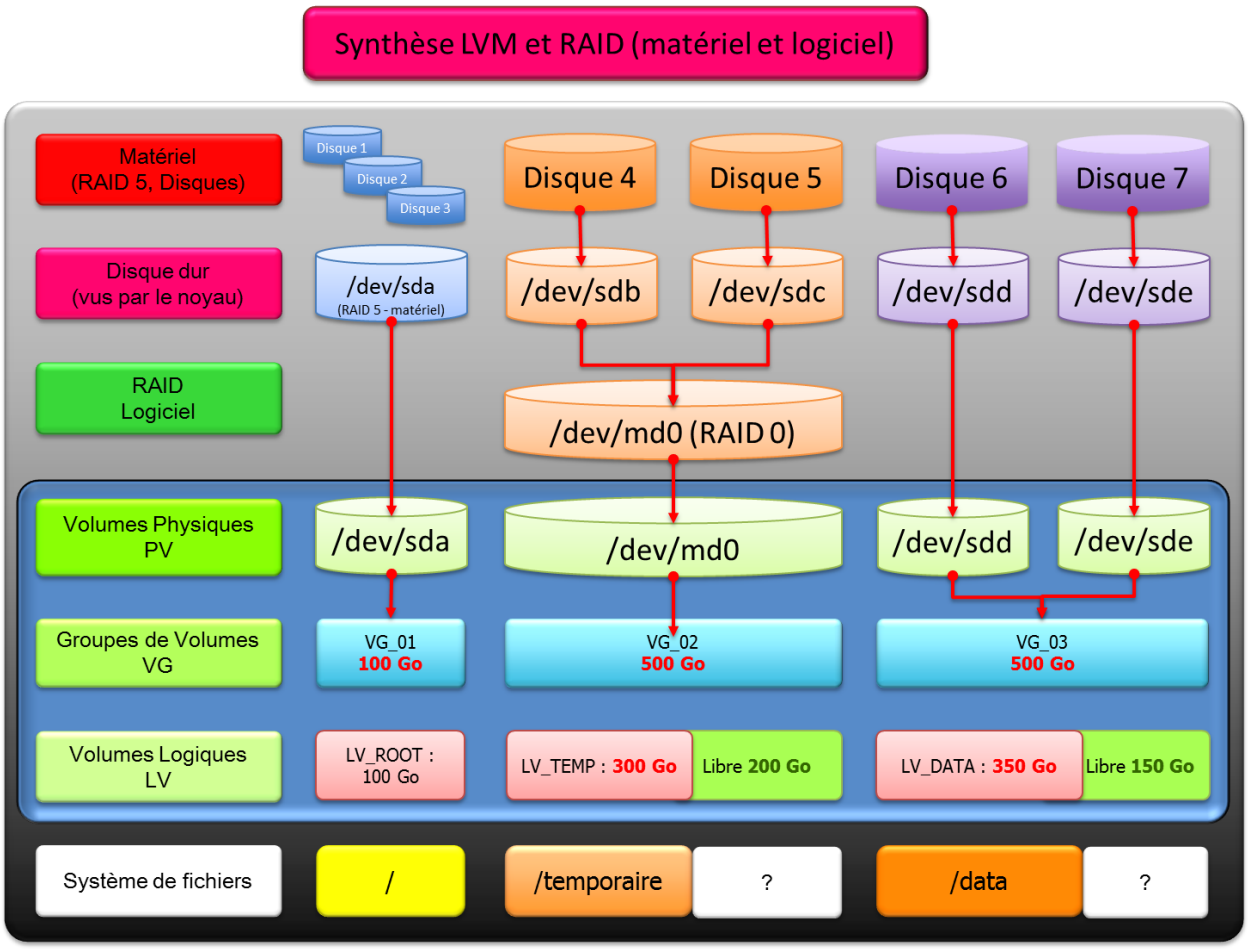
Taille des *snapshots*

Une idée très répandue veut que la taille d'un *snapshot* doit être égale à environ 15-20 % de la taille du volume logique cible. Ces approximations sont totalement infondées, la taille nécessaire à un *snapshot* dépendant de l'activité en écriture sur le volume logique cible pendant la durée de vie de ce *snapshot*. La suppression de toutes les données sur le volume logique cible demandera par exemple un *snapshot* d'une taille au moins égale à la taille du volume logique cible. La taille d'un *snapshot* doit donc être calculée suivant l'estimation du volume de données écrit sur le lecteur cible pendant la durée de votre sauvegarde. Dans la plupart des cas cela nécessitera bien moins de 15 %…

Vous pouvez également gérer les LVM avec l'outil graphique **« system-config-lvm »** :



Voici une synthèse de ce chapitre :



Pour un système utilisable en production il faudra utiliser le premier cas : celui du **RAID matériel (1 ou 5 au minimum)**.

L'exemple des **disques 6 et 7 (en VG_03)** ne propose aucune tolérance de panne et surtout provoquerait une perte de données irrémédiable en cas d'une défaillance sur l'un des deux disques physiques.

Dans le cas où vous vous situez sur un système virtualisé ce sera en amont que les administrateurs du SAN (qui contient en générale le stockage de données - Data Storage) devront assurer la tolérance de panne : vous n'aurez qu'à exploiter un espace de stockage, qui vous sera alloué.

Maintenant étudions le « système de fichiers » qui est la partie exploitable par les programmes et où nous pourrons stocker les données utiles.

ERRATA FDISK CENTOS

```
[root@vtst ~]# fdisk /dev/vg/test

The number of cylinders for this disk is set to 2610.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): d
Selected partition 1

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 22: Invalid argument.
The kernel still uses the old table.
The new table will be used at the next reboot.
Syncing disks.
```

Note the warning. This is expected due to the device not being real.

Le système de fichiers

Un **système de fichiers** (*file system* ou *filesystem* en anglais) est une façon de stocker les informations et de les organiser dans des fichiers sur ce que l'on appelle des mémoires secondaires (disque dur, disquette, CD-ROM, clé USB, SSD, etc.). Une telle gestion des fichiers permet de traiter, de conserver des quantités importantes de données ainsi que de les partager entre plusieurs programmes informatiques. Il offre à l'utilisateur une vue abstraite sur ses données et permet de les localiser à partir d'un chemin d'accès.

Arborescence du système de fichiers racine et le standard FHS

Le système de fichiers vient en surcouche d'une partition. C'est le système de fichiers qui héberge une arborescence de de répertoires et de fichiers.
Entre plusieurs UNIX ou GNU/Linux cette arborescence peut légèrement varier. Cependant pour éviter une trop grande diversité d'arborescence un standard a été adopté par les acteurs majeurs du marché : **« File Hierarchy Standard »**.

Ce standard définit **l'arborescence** et le **contenu des principaux répertoires des systèmes de fichiers** des systèmes d'exploitation GNU/Linux et de la plupart des systèmes de type Unix.

Schéma FHS.

Voici les principaux (source Wikipédia) :

Répertoire	Description	Exemple d'implémentation de la norme
/bin/	Commandes de base pour tous les utilisateurs (par exemple : cat, ls, cp) *(abréviation de binaries, en français : binaires)*	
/boot/	Chargeur d'amorçage Exemple de fichiers : • *initrd* (image mémoire du ramdisk utilisé par le processus init) • noyaux,	• Exemple d'implémentation : Si grub est le chargeur d'amorçage, il y aura le fichier de configuration correspondant *grub.conf*. Le noyau est généralement nommé ***vmlinuz*** *ou* ***vmlinux***
/dev/	Fichiers correspondant (directement ou non) avec un périphérique *(abréviation de device)* • Les fichiers de périphériques : ○ périphériques physiques (disque, réseau, bande, disquette) ○ périphériques virtuels ; ▪ */dev/null*	Exemple d'implémentation • Les fichiers de périphériques : ○ Périphériques physiques ▪ IDE (Exemple pour GNU/Linux : */dev/sda*, */dev/sdb*, */dev/sdc* : En effet, dans les récentes version du noyau Linux, les périphériques IDE ne se

	▪ */dev/zero*	nomment plus hdX, mais bel et bien sdX , comme les périphériques SCSI) ▪ SCSI ; sous GNU/Linux, cela inclut l'USB et le S-ATA ; exemple : */dev/sda*, */dev/sdb*, */dev/sdc* ▪ Les terminaux : ▪ */dev/tty*, tty0 à tty59 ▪ /dev/console Console initiale ▪ Carte réseau (sous GNU/Linux, cela correspond aux interfaces réseau *eth0*, *eth1*...etc. ▪ Cartes son ▪ Périphérique à bande ▪ Périphérique série : par exemple modem ▪ Disquette 3 pouces et demi : sous GNU/Linux, */dev/fd0* • Liens symboliques ○ Cas typique sous GNU/Linux : */dev/cdrom* est un lien symbolique vers le fichier de périphérique correspondant (par exemple : */dev/hdb1*)
/etc/	Fichiers de configuration *(abréviation de editing text configuration, en français : configuration éditable en mode texte)*	
/etc/opt/	Fichiers de configuration pour les applications installées dans /opt	
/etc/X11/	Ce répertoire contient les fichiers de configuration pour X11.	NB : Le client et le serveur X11 peuvent être sur deux ordinateurs différents. Le serveur peut ne pas avoir de carte graphique. • Pour les ordinateurs utilisant XFree86, le principal fichier de configuration est *XF86Config-4* ou *XF86Config* selon la distribution • Pour les ordinateurs utilisant Xorg, le principal fichier de configuration est *xorg.conf*
/etc/sgml/	Fichiers de configuration pour SGML.	Sans objet le plus souvent
/etc/xml/	Fichiers de configuration pour **XML**.	
/home/	Répertoires des utilisateurs (exemple : */home/dupont*)	
/lib/	Bibliothèques logicielles nécessaires pour les exécutables de */bin/* et */sbin/* *(abréviation de libraries)*	
/mnt/	Point de montage pour les fs (systèmes de fichiers) temporaires *(abréviation de*	

	mount)	
/media/	Point de montage pour les médias amovibles (apparu dans FHS-2.3)	Parmi les media amovibles, il y a notamment les CD-ROM et les clés USB
/opt/	Logiciels optionnels	Logiciels non inclus dans la distribution, installés manuellement
/proc/	Système de fichiers virtuel documentant le noyau et les différents processus *(abréviation de processes)*	Il évolue en temps réel, et n'occupe pas d'espace disque.
/root/	Répertoire de l'Utilisateur root	
/sbin/	Exécutables pour les administrateurs *(abréviation de system binaries, en français : binaires système)*	
/srv/	Données pour les **serv**ices hébergés par le système, comme du contenu http/ftp (l'arborescence complète d'un site web), une base de données...	
/tmp/	Fichiers temporaires (voir aussi */var/tmp*) *(abréviation de temporary)*. Est vidé à chaque démarrage et possède le sticky bit	
/usr/	Contient certains dossiers semblables à ceux présents à la racine mais qui ne sont pas nécessaires au fonctionnement minimal du système (*usr* comme **u**nix **s**ystem **r**esources)	
/usr/bin/	Binaires exécutables en complément de /bin	
/usr/include/	Entêtes des bibliothèques partagées	
/usr/lib/	Bibliothèques partagées	
/usr/sbin/	Binaires pour l'administrateur (complément de /sbin)	
/usr/share/	Fichiers indépendants de la plateforme (non binaires)	La documentation : • répertoire *man* pour les man (sous GNU/Linux : document au format roff compressé) • répertoire *doc* : documentation au format HTML ou autre
/usr/src/linux	Sources du noyau Linux	
/usr/X11R6/	X Window System, X11 version 6.	
/usr/local/	Hiérarchie tertiaire pour les données locales, spécifiques à l'ordinateur	
/var/	Fichiers variables, tels que o base de données o sites web o boîte aux lettres de messagerie	

	o journaux, voir historique	
/var/lock/	Fichiers de verrouillage, permettant de connaître quelles ressources sont en cours d'utilisation	
/var/log/	Fichiers de journalisation	Exemple sous GNU/Linux : syslog, *XFree86.0.log*, *kern.log*, *mysql*, *gdm/:0.log*
/var/mail/	Boîtes aux lettres utilisateurs	
/var/run/	Fichiers temporaires des logiciels en cours d'exécution	Exemples : PIDs ou statut des services
/var/spool/	Files d'attente des services	File d'attente de fichiers à imprimer par CUPS
/var/spool/mail/	Mails en cours de transit sur la machine	Mails en attente d'envoi vers d'autres serveurs, ou en attendant de délivrance aux utilisateurs locaux
/var/spool/cron	Stockage des tâches planifiées des utilisateurs	
/var/tmp/	Fichiers temporaires. Préféré à */tmp* lorsqu'on est au niveau d'init multiutilisateur	
/var/www/	Répertoire web par défaut d'Apache	

Ceci est donc un standard mais comme tout standard il n'est pas toujours respecté à la lettre. Il constitue néanmoins une bonne base pour comprendre l'arborescence d'un système de fichiers GNU/Linux.

Les types de File System

Introduction

Nous entrons dans le vif du sujet. En effet GNU/Linux peut utiliser un nombre impressionnant de systèmes de fichiers.

Sous linux vous avez remarqué que la notion de formatage au sens Windowsien du terme n'existe pas : sous GNU/Linux on **crée un système de fichiers avec un type** choisi **puis on le monte** pour le rendre accessible à l'utilisateur.

C'est grâce à la couche **VFS** (**V**irtual **F**ile **S**ystem) que le noyau peut prendre en charge et faire cohabiter ensemble plusieurs de type de systèmes de fichiers. Voici les principaux types :

- Locaux journalisés ou non (ext3, ext4, ext2, xfs, jfs, vfat, fatxx, ntfs, reiserfs, hfs+, nss etc.),
- Virtuels (/proc, /sys, ramfs etc.),
- Réseaux (nfs, cifs, afp, etc.),
- Clusterisés (gfs2, coda, etc.),
- Etc.

En cas de besoin le noyau Linux peut se voir ajouter la prise en charge de systèmes de fichiers supplémentaires via recompilation du noyau ou par ajout de module noyau (**K**ernel **O**bject, .ko).

Info : La **journalisation** d'un système de fichiers apporte une **couche transactionnelle au système de fichier**.
Les systèmes de fichiers journalisent les métadonnées. Les métadonnées sont les structures de contrôle d'un système de fichiers : inodes, tables d'allocation de l'espace libre, tables d'inodes, etc.

Inodes et blocs

Le bloc est l'unité de base de stockage, c'est-à-dire la plus petite unité d'allocation du système de fichiers. Un fichier occupe toujours un nombre entier de bloc. La taille d'un bloc est choisie à la création du système de fichiers.
Si par exemple la taille d'un bloc est de 512 octets, si vous créez un fichier de 1 octet cela gâchera un espace de 511 octets. Cette notion doit être bien assimilée car si par malchance vous avez une application qui produit une quantité impressionnante de fichier de 1 octet vous allez remplir votre espace disque très rapidement en étant surpris.
Certes ce cas est très rare, mais si vous planifiez d'utiliser une application qui va produire des fichiers de petites tailles : prenez bien le temps de choisir la taille de bloc de base de votre système de fichiers dès sa création.

Soyez également vigilants aux formats qui sont utilisés par certaines commandes pour renvoyer des tailles d'occupation disque : bloc ou octet ?

L**e super bloc est une zone de métadonnées qui contient plusieurs informations sur le système de fichiers** que vous manipulez. Un système de fichiers dispose d'un super bloc minimum et ce dernier nous donne des renseignements sur le système de fichiers :

- Son type,
- Sa taille,
- Son état,
- Des informations de positionnement sur les autres zones de métadonnées (autres super bloc, tables des inodes, etc.)

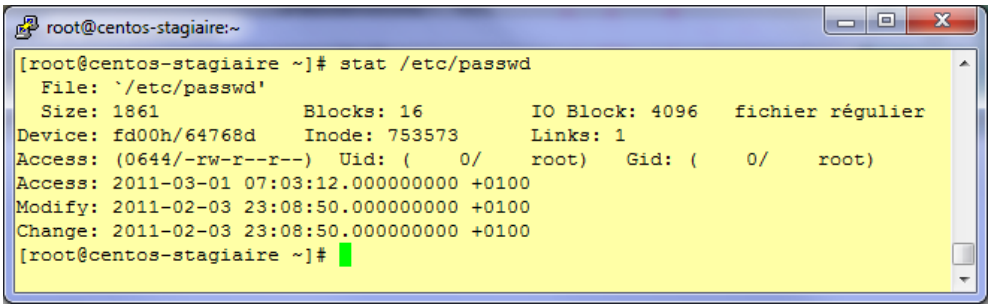
Un inode est la structure de données qui contient les attributs pour gérer un fichier.
Voici le contenu d'un inode :

- La taille du fichier en octets
- Identifiant du périphérique contenant le fichier
- L'identifiant du propriétaire du fichier
- L'identifiant du groupe auquel appartient le fichier
- **Le numéro d'inode** qui identifie le fichier dans le système de fichiers : **il est unique**
- Le *mode* du fichier qui détermine quel utilisateur peut lire, écrire et exécuter ce fichier
- horodatage (timestamp) pour
  - La date de dernière modification *ctime* de l'**inode** (affichée par la commande *stat* ou par *ls -lc*, modification des droits du fichier)
  - La date de dernière modification du fichier *mtime* (affichée par le classique *ls -l*)
  - La date de dernier accès *atime* (affichée par la commande *stat* ou par *ls -lu*)
- Un compteur indiquant le nombre de liens physiques sur cet inode
- Les adresses pointant sur les premiers blocs de données du fichier
- Les adresses pointant sur des blocs contenant d'autres champs d'adresses (bloc d'indirection simple, double ou triple)

Les inodes ne contiennent pas les noms de fichier. Cette fonction est assignée à la table des inodes.

Note : les liens physiques (Hard links) permettent de rajouter une référence sur un inode. Le lien physique rajoute une association dans la table des inodes

La commande **« stat »** donne quelques informations sur un inode :

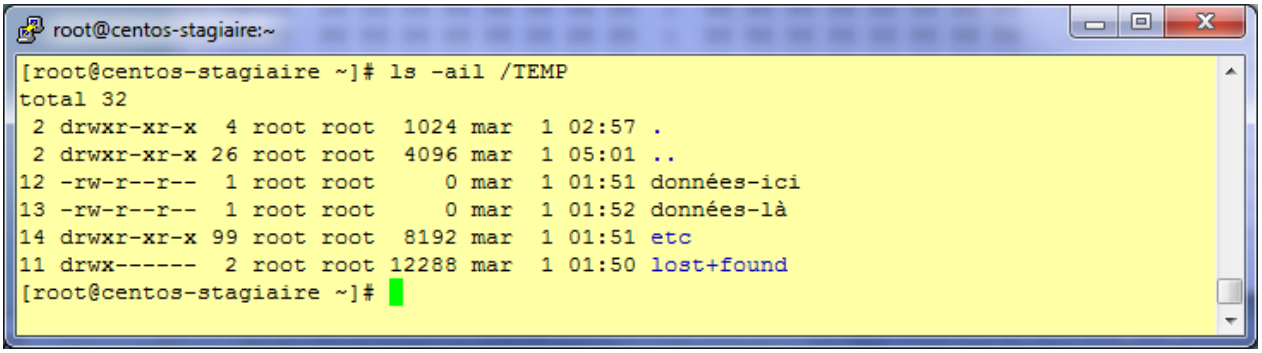


Chaque fichier dispose d'un numéro d'inode (i-number). Parfois plusieurs fichiers, ayant un nom distincts, peuvent pointer vers le même inode. Tous les inodes sont regroupés au sein **d'une table d'inodes** afin de répertorier chaque fichier d'un système de fichiers et ils sont uniques.

Note : Un nom de fichier ne peut dépasser **255** caractères.

Les noms des fichiers n'étant pas stockés dans l'inode lui-même, ils sont stockés dans la table des inodes. Voici une représentation simpliste d'une table des inodes :

Vue utilisateur avec **« ls -i »** :



L'inode ne contient pas le nom du fichier, cette correspondance inode/nom de fichier est stockée dans la **table des catalogues**. Cette table est un répertoire. Un répertoire contenant une liste de fichiers (ls -ail), et un fichier étant représenté par un inode, chaque nom de fichier est associé au sein du répertoire à son inode

Extrait de vue d'une table de catalogue du répertoire /TEMP :

Table du catalogue TEMP (répertoire /TEMP)	
Inode	**Nom**
12	***données-ici***
13	***données-là***
14	***etc***
11	***lost+found***

Les systèmes de fichier locaux courants : « ext2 », « ext3 »

« ext2 »

Il est considéré comme le système de fichiers historique de Linux, bien que celui-ci utilisait au tout début le MinixFS. La première mouture appelée « ext » (extended filesystem), bien que corrigeant les défauts de minix, avait quelques limites qui n'en faisaient pas un véritable système de fichiers Unix.
Ext2 est donc le premier système de fichiers développé spécifiquement pour Linux, d'un niveau de production et aux normes Unix (on parle de niveau de production pour indiquer un système quelconque répondant aux critères de mise en production (utilisation réelle) en entreprise). Prévu dès le début pour supporter les rajouts de fonctionnalités, il continue depuis 1993 à être utilisé et amélioré
Ext2 n'est pas journalisé.
Bien que disposant d'un successeur (ext3), il est toujours utilisé voire conseillé dans certains cas. Il est rapide et nécessite moins d'écritures que les autres, donc il occasionne moins d'usure des supports de stockage, notamment les disques SSD, les clés USB ou les cartes mémoire. Ces supports peuvent parfois ne supporter qu'un nombre restreint de cycles de lecture/écriture...
Les fichiers peuvent avoir jusqu'à une taille de 2To (2048 Go), tandis qu'une partition peut atteindre 32 To, voire 128 To , selon la taille des blocs et l'architecture.

« ext3 »

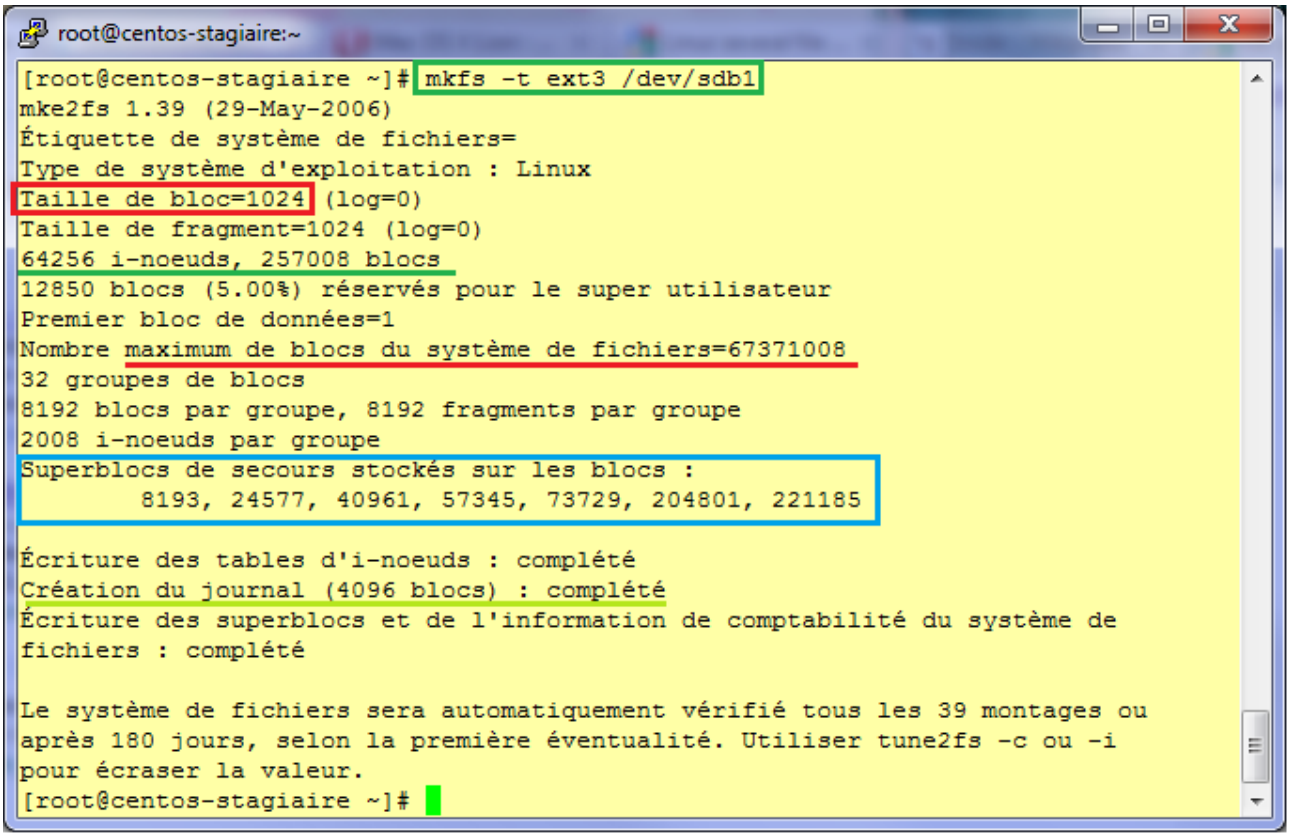
Il s'agit du successeur d'ext2 depuis 1999. Il est journalisé. Surtout, il est entièrement compatible avec ext2. Le journal est une extension d'ext2. Il est possible d'utiliser un système de fichiers ext3 comme étant ext2, avec les mêmes commandes, les mêmes manipulations. Il est possible de transformer en quelques secondes un système ext2 en ext3, et vice versa. C'est l'un des systèmes de fichiers de choix pour Linux, et le plus utilisé pour sa souplesse.
Comme pour ext2, la taille maximale des fichiers est de 2 To, et celle d'une partition de 32 To, suivant les mêmes restrictions.

Manipulations du système de fichiers

Création d'un système de fichiers

Commençons par créer un système de fichiers « ext3 ».

La commande qui permet de faire cette opération est **« mkfs »** :



Veuillez noter les précieuses informations fournies par cette commande concernant notre système de fichiers :

- **Taille de bloc,**
- **Nombre maximum de bloc,**
- **Nombre d'inodes et blocs disponibles,**
- **Superblocs primaire et de secours,**
- **Journalisation,**
- Périodicité de la maintenance.

Voici la syntaxe générale de « mkfs » (notez que **« mkfs » appelle** mkfs.<typefs>):

- mkfs -t typefs options périphérique

typefs :

- ext3,
- swap,
- vfat,
- ntfs,
- etc.

Mise en ligne du système de fichier

Un fois votre système de fichiers créé, il va falloir le mettre à disposition des utilisateurs.

Sous GNU/Linux on peut accéder à un périphérique (lecture, écriture) à travers un fichier spécial. Ce type de fichier permet d'accéder au matériel.

Voici les caractéristiques de ce type de fichier :

- Le mode :
  bloc (b), disques, partitions
  caractère (c), port série, carte son, lecteur de bande
- Un majeur, valeur désignant le pilote du périphérique,
- Un mineur, qui permet d'identifier le périphérique géré par le pilote désigné par le majeur.

Monter un système de fichier

Le montage est l'opération qui permet de positionner un point de l'arborescence à partir duquel les fichiers d'un système de fichiers (réseau, local etc.) seront visibles par un utilisateur. En général seul « root » peut effectuer cette opération.
Pour se faire on utilise la commande **« mount »**. Elle permet donc d'a**ttacher le répertoire racine d'un système de fichier**, présent sur une partition, à **un répertoire quelconque appelé point de montage**.

Voici la commande qui permet de monter des systèmes de fichiers manuellement :

mount -t typefs -o options <périphérique> <point_de_montage>

Il existe plusieurs manières pour monter un système de fichiers :

- Montage par périphérique,
  mount -t ext3 *//dev/sdb1* /GESTION_FS
- Montage par label,
  La liste des **labels existants** s'obtient avec **« ll /dev/disk/by-label »**
  mount -t ext3 -*L GESTION_FS* /GESTION_FS
- Montage par UUID,
  La liste UUID s'obtient avec **« ll /dev/disk/by-uuid »** ou **« dumpe2fs -h /<periph.> | grep UUID »**
  mount -t ext3 -*U 86d22660-d74d-4e05-aef3-38b30a2a3076* /GESTION_FS

Le **montage par UUID est maintenant le plus utilisé**. L'UUID est très intéressant par le fait que même si votre disque change de position logique ou physique il sera quand même retrouver par le noyau via cet identifiant unique.
Utilisé seul la commande **« mount » affiche tous les systèmes de fichier montés**. En fait elle affiche le **fichier /etc/mtab**. Vous pouvez aussi utiliser **le fichier /proc/mounts**.

Le système de fichiers « Swap ».
Sur un système GNU/linux il est toujours intéressant de disposer d'une partition dite de « swap » même si le serveur dispose de beaucoup de mémoire vive. En général vous n'aurez pas à créer cette partition car elle sera toujours créée par une installation par défaut et correctement dimensionnée.
Procéder comme pour toute autre partition puis créer le système de fichiers type « swap » : **« mkswap »** dessus, puis activez là avec «**swapon <partition> ».** Le dimensionnement est le suivant :
Si RAM < 512 Mo => Taille SWAP=2xRAM,
Si 1Go < RAM < 4Go => Taille SWAP = RAM enfin si RAM > 4Go => Taille SWAP = 4Go.

Voici les principales « options » de montage de la commande **« mount »** :

Option	Signification
defaults	Prends les options suivantes : « ***rw,suid,dev,exec,auto,nouser,async »***
sync/async	Active ou désactive les écritures synchrones. Avec async les écritures passent par un tampon qui diffère les écritures (plus performant) rendant la main plus vite. Il est préférable d'activer les écritures synchrones sur des supports externes (clés USB, disques USB/Firewire/eSATA, etc.).
exec/noexec	Permet l'exécution/ou non des fichiers binaires sur le support.
noatime	Évite la mise à jour de l'horodatage à chaque accès à un fichier (pertinent sur les supports externes, disques SSD, pages web, newsgroups, etc.).
auto/noauto	Le système de fichiers est automatiquement monté/ne peut être monté que explicitement (voir fstab).
user/nouser	N'importe quel utilisateur peut monter le système de fichiers (implique noexec, nosuid, et nodev)/seul root a le droit de monter le système de fichiers (voir fstab).
remount	Remontage du système de fichiers pour la prise en compte de nouvelles options.
ro/rw	Montage en lecture seule ou lecture et écriture.
dev/nodev	Interpréter/Ne pas interpréter les fichiers spéciaux.
noload	Pour ext3, ne charge pas le journal.
acl	Permet l'utilisation des Access Control Lists.
user_xattr	Pour ext3 et xfs, accepte les attributs étendus sur les fichiers, par exemple pour y coller des informations additionnelles (l'encodage du texte, etc.), des champs d'indexation (utilisés par Beagle par exemple), etc.
umask	Pour FAT/NTFS, applique un autre masque global que celui par défaut (ex 133).
dmask=/fmask=	FAT/NTFS, différencie les masques pour les répertoires et les fichiers.
uid=/gid=	FAT/NTFS, comme les droits et propriétaires ne sont pas gérés, applique un utilisateur ou un groupe par défaut sur les fichiers (ex gid=users).

Démontage

Il suffit de saisir la commande suivante :

umount <point_de_montage>

Astuce : en cas de périphérique occupé voir plus bas (« Outil de diagnostic du système de fichiers ») les commandes « **lsof** » ou « **fuser** ».

Montage automatique au démarrage

Par défaut, les montages effectués manuellement restent disponibles tant que votre système ne redémarre pas. En cas de redémarrage ces montages ne sont pas persistants et disparaitront.
Pour faire en sorte qu'un montage devienne persistant et donc présent au démarrage de GNU/Linux, il faut définir ce montage dans le fichier **« /etc/fstab »**.

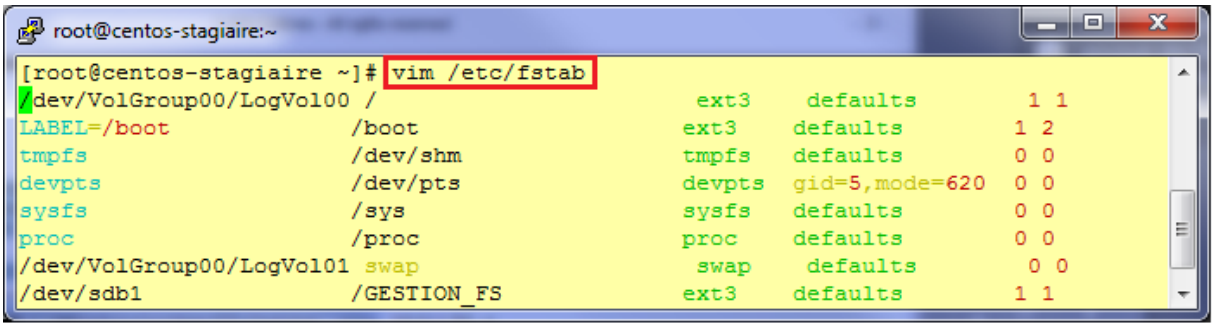
Le fichier **/etc/fstab** contient une configuration statique des différents points de montage des systèmes de fichiers. Il est appelé à chaque démarrage du système car c'est ici qu'on indique les périphériques et leurs points de montage. Il contient six champs.

<nom_du_périphérique> <point_de_montage> <typefs> <options> <dump> <fsck>

Les champs sont séparés par des espaces ou des tabulations.

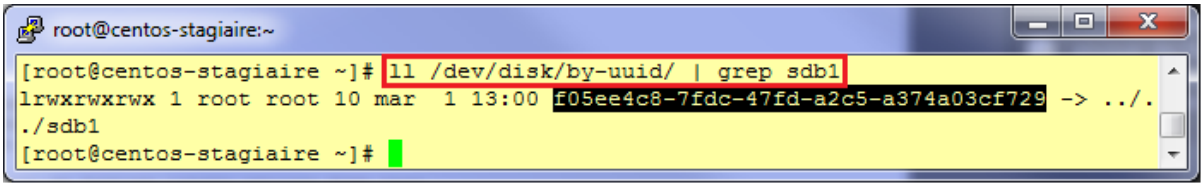
Champ	Description
nom_du_périphérique	Le périphérique à monter. Il peut être spécifié en tant que chemin de périphérique (/dev/hda1 par exemple), que label de système de fichiers s'il existe (LABEL=/home), ou encore en tant que UUID (UUID=xxxx).
point_de_montage	Le répertoire d'accès au système de fichiers monté.
typefs	Le type (ext2, etx3, reiser, vfat, etc.) du système de fichiers.
options	Les options, séparées par des virgules, vues précédemment.
dump	Fréquence de dump pour les outils de dump ou de sauvegarde.
fsck	Fréquence de vérification du système de fichiers.0=ignorer. 1=en premier, 2= en second, etc. Les systèmes ayant le même numéro sont vérifiés en parallèle.

Voici un exemple de fichier « **/etc/fstab** »:

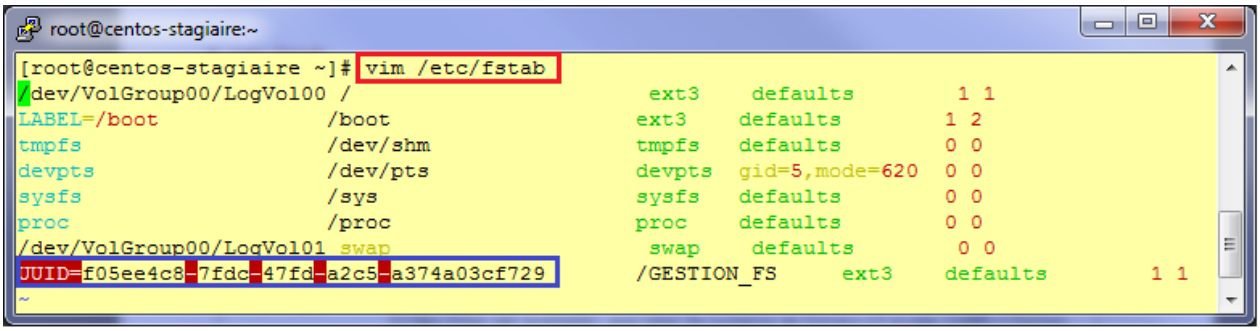


Une des améliorations à apporter à une installation de base Centos est l'utilisation du UUID dans /etc/fstab (il a déjà un exemple avec un LABEL pour /boot). Voici comment procéder pour le périphérique suivant: **« /dev/sdb1 »**.

- Recherche de l'UUID du périphérique :



- Edition du fichier /etc/fstab :

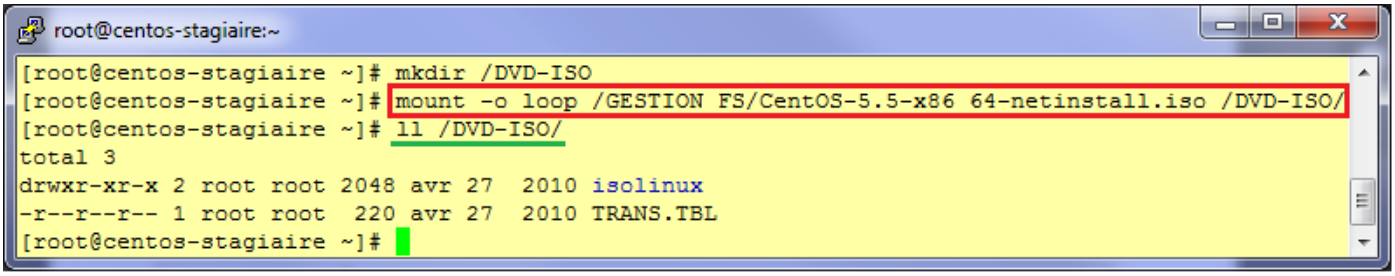


La commande **« mount -a »** permet de refaire les montages présents dans ce fichier /etc/fstab

Loopback device

L'option **« -o loop »** de la commande **« mount »** permet de considérer un fichier (.iso, .dmg, .img etc.) comme un système de fichier. Cela permet par exemple de monter une image iso fraîchement téléchargée sur un point de montage de votre choix.

Exemple :



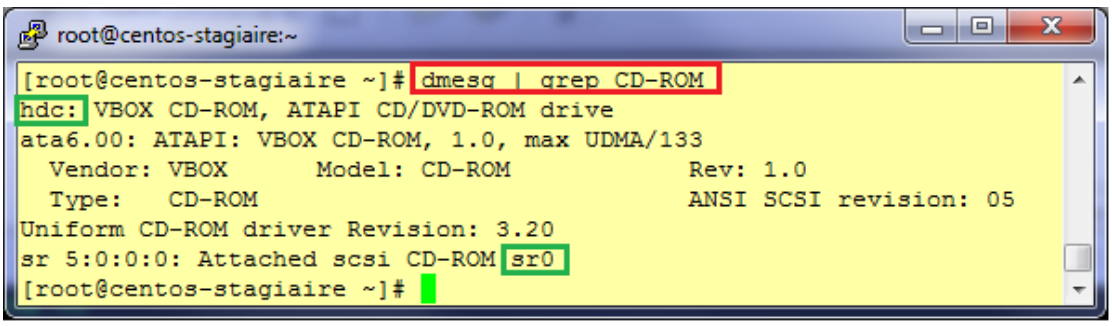
Les CD et DVD-ROM

D'abord vous devez créer un répertoire (mkdir) pour accueillir le point de montage : **/dev/cdrom**

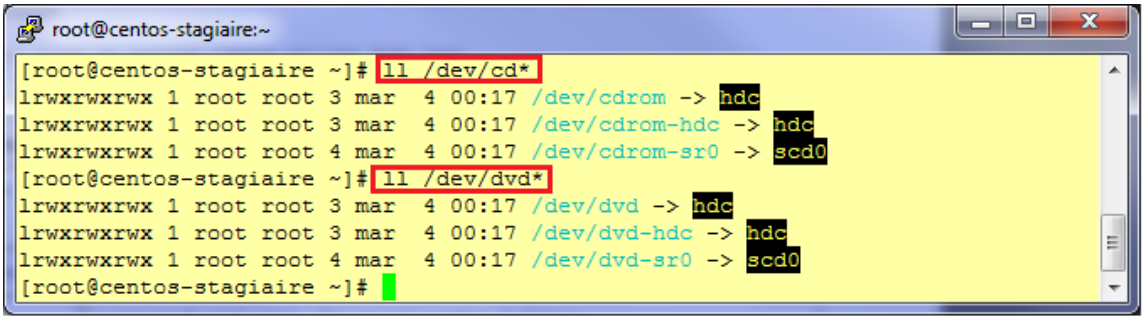
Puis pour monter le CD/DVD-ROM :

mount *dev/cdrom* /media/cdrom

Trouver les périphériques CD-ROM ou DVD-ROM :



Sous Centos les CD-ROM, DVD-ROM ne sont pas accessibles via les périphériques « **hdx** », « **scdx** » ou « **srx** », mais via des liens symboliques : **/dev/cdrom ou /dev/dvd**



La commande **« eject -n »** donne également des indications concernant le lecteur de CD ou DVD présents sur le système.

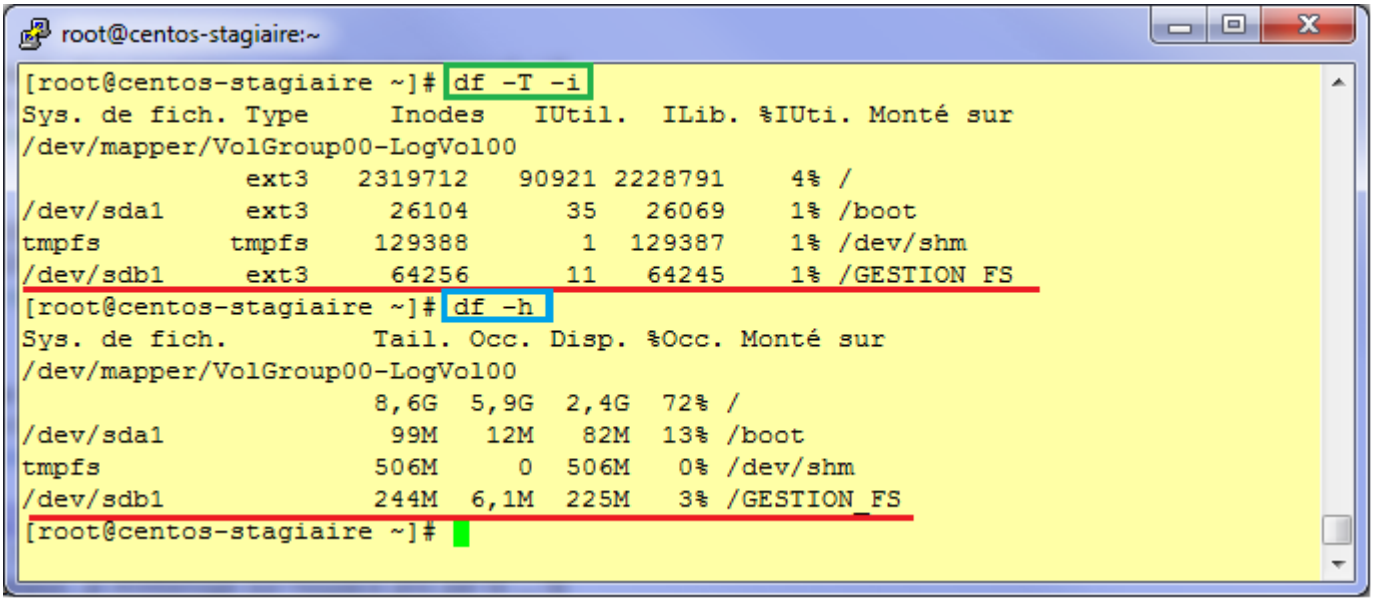
Pour aller plus loin :

C'est **« udev »** qui se charge de monter automatiquement les CD, DVD, clé USB, disque externe sur une distribution GNU/Linux moderne.

Outil de diagnostic du système de fichiers

Vous pouvez à tout moment visualiser le taux d'occupation de votre système de fichiers que ce soit en visualisant :

- Les inodes libres, utilisés, totaux => **« df -T -i**, »
- Ou les tailles en Go, Mo, Ko (Octets) => **« df -h »**,

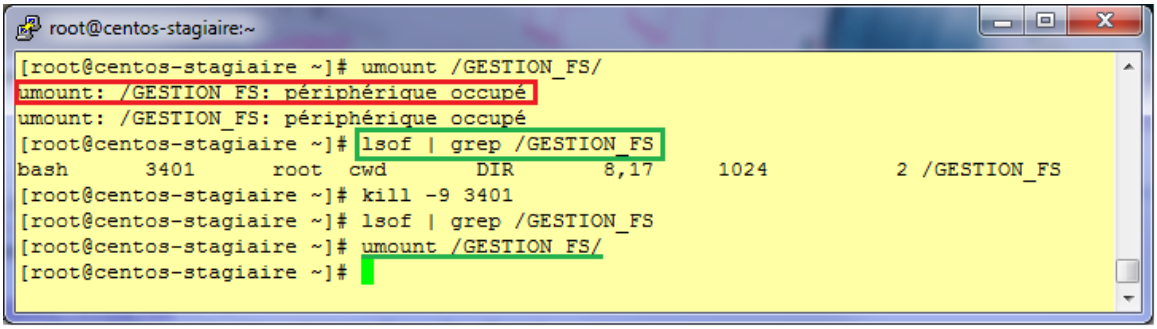


Parfois il peut arriver que vous ne parveniez pas à **démonter un système de fichier**, pour effectuer des opérations de maintenance, car ce dernier s'avère être **occupé** (Busy).

Il existe deux outils indispensables pour trouver le **fichier**/**processus**/**utilisateur** coupable du verrouillage :

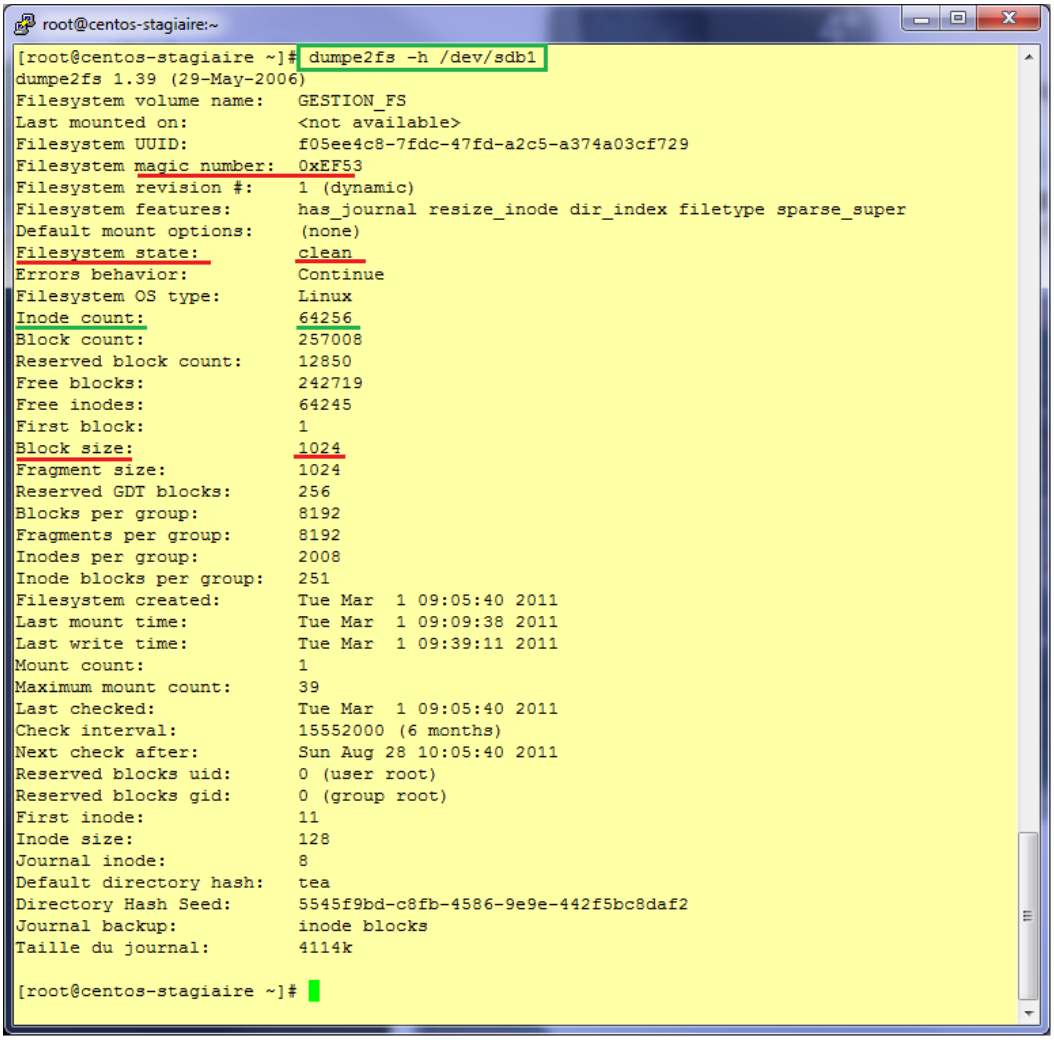
- **« lsof »**,
- **« fuser »**.

Voyons le fonctionnement de « lsof » :



Comme vous pouvez le constater « lsof » vous indique quel est le processus qui verrouille le fichier, ici le **processus « bash » avec le PID 3401**.

Une autre commande est assez intéressante : **« dumpe2fs »**. Elle permet d'afficher un ensemble d'informations assez complètes sur un système de fichier.



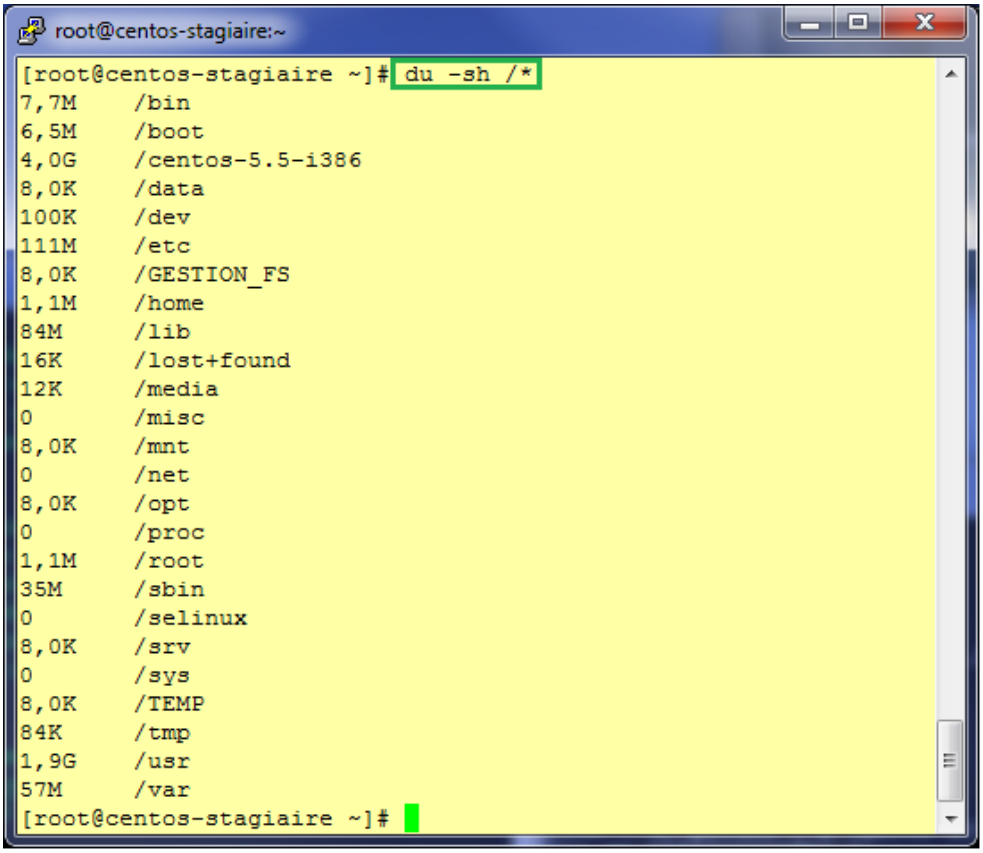
En saisissant la commande suivante vous avez une information beaucoup plus complète :

- **dumpe2fs -h /dev/sdb1**

Consultation du « superbloc » :

- **tune2fs -l /dev/sdb1 | grep Label**

« du » est une commande incontournable. Elle permet de calculer l'espace occupé par un répertoire, une arborescence (récursivement) :



Note : l'option -x permet de se limiter au système de fichiers courant.

Comme GNU/Linux est un système se basant sur les fichiers, vous pouvez également surveiller l'évolution de la mémoire via le système de fichiers virtuel /proc. Pour se faire :

- **watch -d cat /proc/meminfo ou la commande « free »**

Opérations de maintenance

Avant toute opération de maintenance sur un système de fichiers il faut en interdire l'accès à tous utilisateur ou processus. Pour ce faire **démontez simplement le système de fichiers que vous voulez manipuler**.

La commande suivante **« tune2fs »** permet de modifier certains paramètres d'un système de fichiers ext2 ou ext3 :
Modification du « label »

- **tune2fs -L *label* /dev/sdb1**

Modification du nombre de montage avant vérification automatique du système de fichiers :

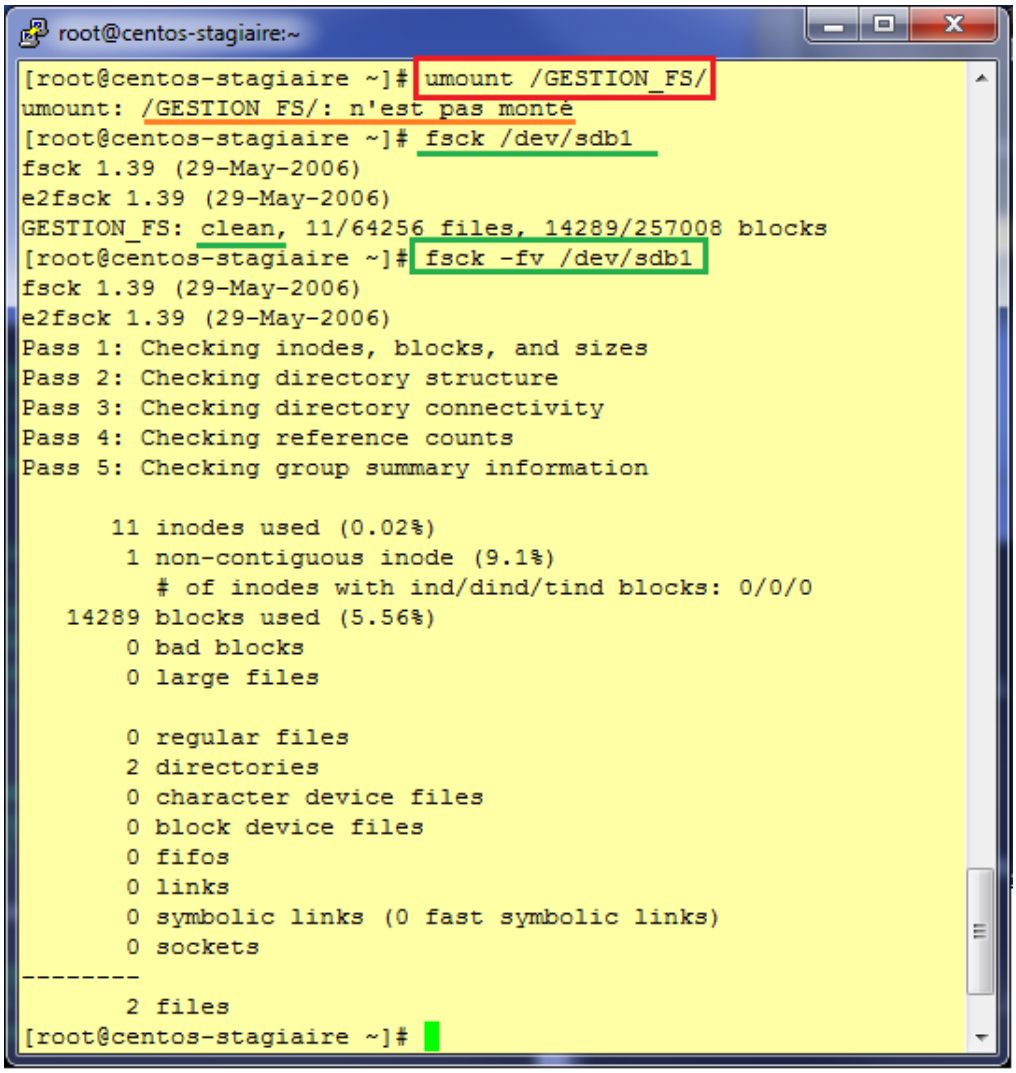
- **tune2fs -c 10 /dev/sdb1**

Changement du label d'un système de fichiers (ne pas dépasser plus de 16 caractères) :

- **e2label /dev/sdb1 *label***

Enfin voici des commandes très importantes qui vont vous permettre de **vérifier et réparer votre système de fichiers** :

- **fsck,**
- **e2fsck**



Une fois de plus avant l'utilisation de telles commandes vous devez avoir votre **système de fichiers démonté**.

Note : **fsck -t ext2** appelle **e2fsck**

La commande e2fsck fait des contrôles de cohérence.
Voici les options les plus courantes de **« fsck »** :

Options	Fonctionnalité
-y	Répond automatiquement « yes » à toutes les réponses du mode interactif
-c	Recherche les blocs défectueux et les place dans un inode prévu à cet effet
-b	Permet de spécifier un bloc qui contient une copie du super bloc afin de réparer un super bloc primaire endommagé

Les droits d'accès (POSIX et ACL Etendues)

Tout système d'exploitation se doit de proposer une sécurisation du système de fichiers qu'il gère. GNU/Linux est un système multi-utilisateurs où l'accès à chaque fichier ou répertoire est contrôlé par des droits.
Voici la commande pour afficher les droits POSIX **« ls –al »**, et celle permettant d'obtenir des informations concernant l'utilisateur actuellement connecté **« id »** :

```
root@centos-stagiaire:~
[root@centos-stagiaire ~]# ls -al /var/www/
total 64
drwxr-xr-x  8 root      root 4096 fév  3 22:46 .
drwxr-xr-x 25 root      root 4096 fév  3 22:48 ..
drwxr-xr-x  2 root      root 4096 avr  4  2010 cgi-bin
drwxr-xr-x  3 root      root 4096 fév  3 22:46 error
drwxr-xr-x  2 root      root 4096 avr  4  2010 html
drwxr-xr-x  3 root      root 4096 fév  3 22:46 icons
drwxr-xr-x 14 root      root 4096 fév  3 22:46 manual
drwxr-xr-x  2 webalizer root 4096 fév  3 22:46 usage
[root@centos-stagiaire ~]# id
uid=0(root) gid=0(root) groupes=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
```

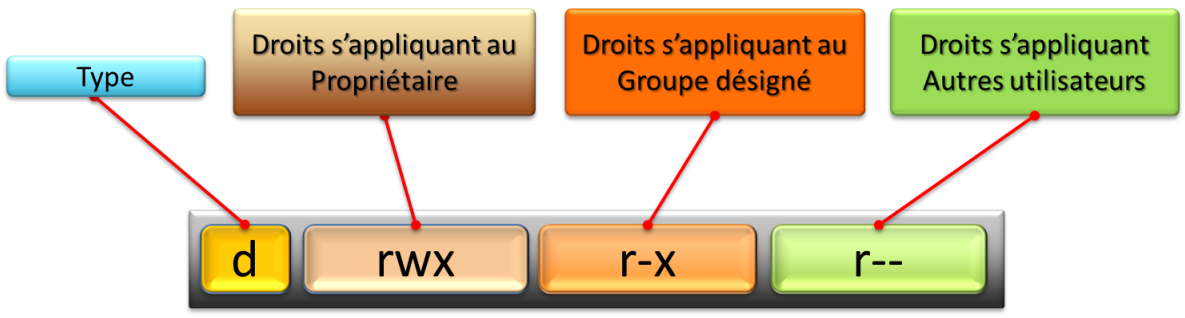
Signification des champs :

- **Droits d'accès POSIX,**
- **Propriétaire du répertoire (en interne UID),**
- **Groupe du répertoire (en interne GID),**
- **UID de l'utilisateur,**
- **GID du groupe principale de l'utilisateur,**
- **Groupes, groupes secondaires de l'utilisateur**

À sa création par l'administrateur, un utilisateur se voit affecter un **UID** (*User Identification*) unique. Les utilisateurs sont définis dans le fichier « **/etc/passwd** ». De même chaque utilisateur est rattaché au moins à un groupe (groupe principal), chaque groupe possédant un identifiant unique, le **GID** (*Group Identification*). Les groupes sont définis dans « **/etc/group** ».

À **chaque fichier (inode)** sont **associés un UID et un GID** définissant son propriétaire et son groupe d'appartenance.
Vous affectez des droits pour le propriétaire, pour le groupe d'appartenance et pour le reste du monde.
On distingue trois cas de figure, dit « **UGO** » (**U**ser**, G**roup**, O**thers) :

- UID de l'utilisateur identique à l'UID défini pour le fichier. Cet utilisateur est propriétaire du fichier.
- Les UID sont différents : le système vérifie si le GID de l'utilisateur est identique au GID du fichier. Si oui l'utilisateur appartient au groupe associé au fichier.
- Dans les autres cas (aucune correspondance) : il s'agit du reste du monde (**others**), ni le propriétaire, ni un membre du groupe. Ils doivent être des utilisateurs présents dans le fichier **« /etc/passwd »**

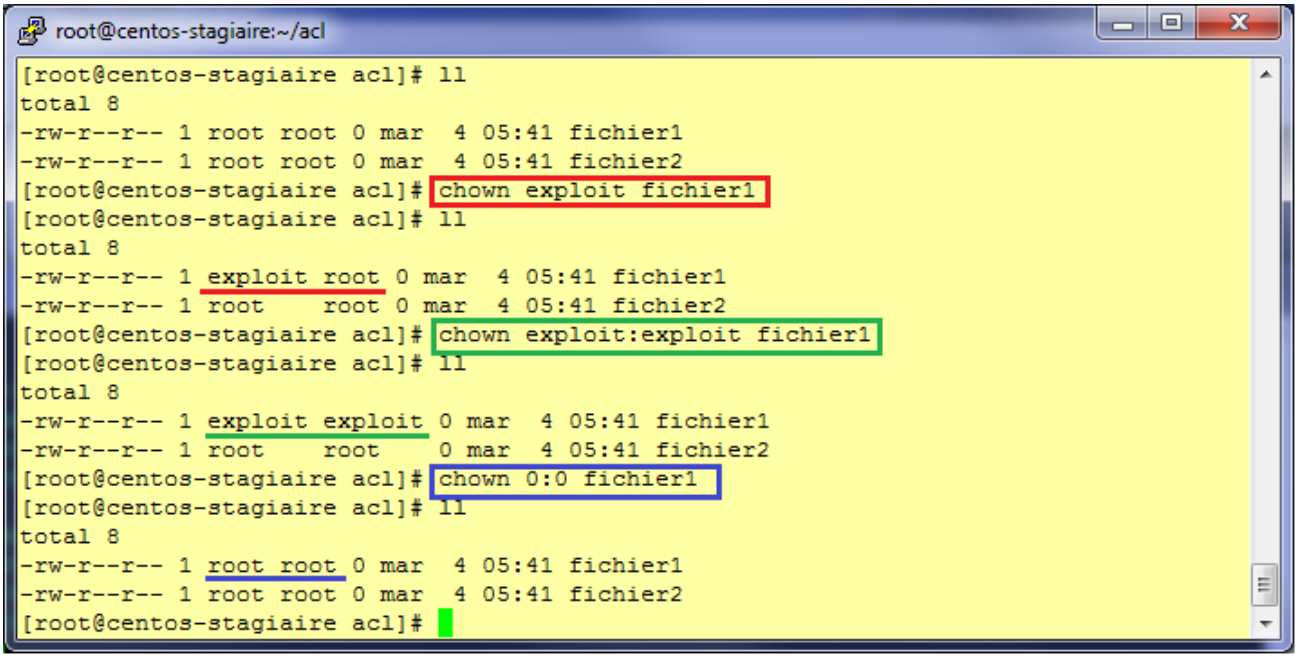


Voici la signification des droits d'accès :

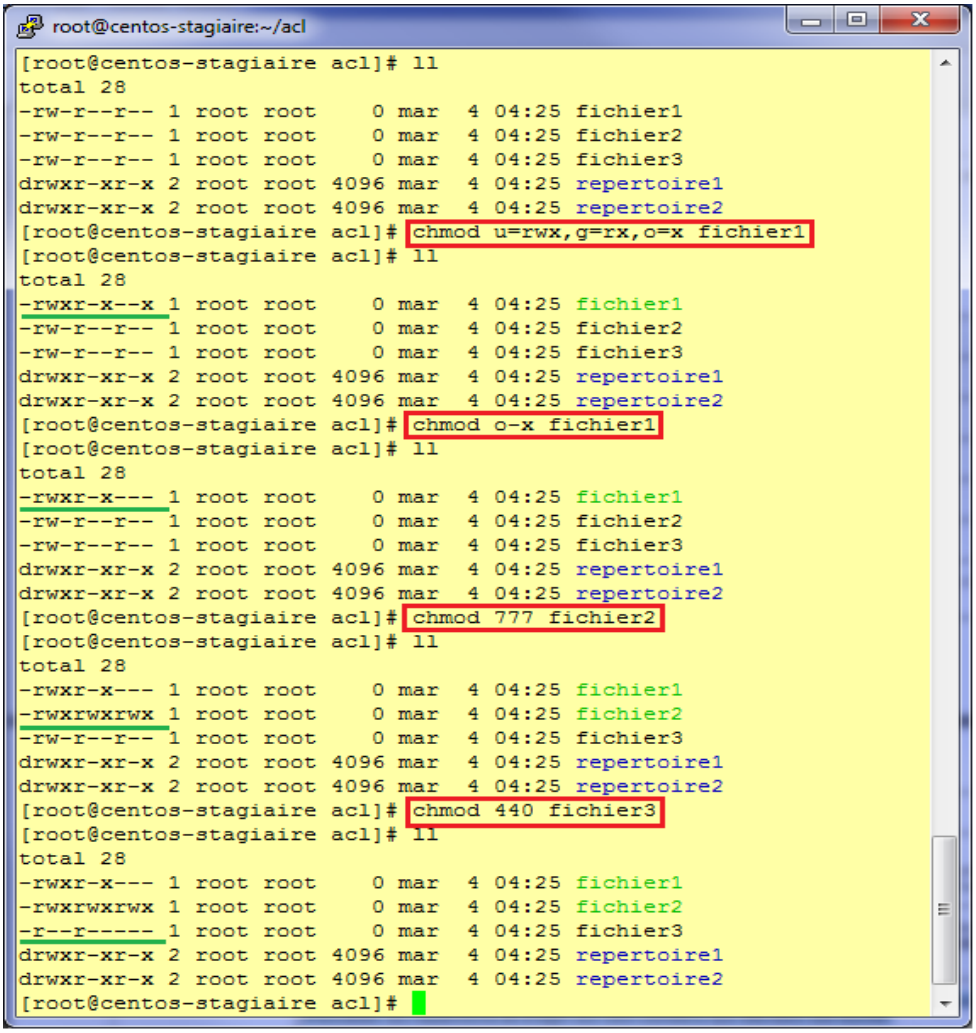
Droit	Signification
Général	
r	Readable (lecture)
w	Writable (écriture)
x	Executable (executable)
Fichier	
r	Le contenu du fichier peut être lu, chargé en mémoire, visualisé, recopié.
w	Le contenu du fichier peut être modifié, on peut écrire dedans. La suppression n'est pas forcément liée à ce droit (voir droits sur répertoire).
x	Le fichier peut être exécuté depuis la ligne de commande, s'il s'agit soit d'un programme binaire (compilé), soit d'un script (shell, perl...).
Répertoire	
r	Les éléments du répertoire (catalogue) sont accessibles en lecture. Sans cette autorisation, ls et les critères de filtre sur le répertoire et son contenu ne sont pas possibles. L'accès individuel à un fichier reste possible si vous connaissez son chemin.
w	Les éléments du répertoire (catalogue) sont modifiables et il est possible de créer, renommer et supprimer des fichiers dans ce répertoire. **C'est ce droit qui contrôle l'autorisation de suppression d'un fichier.**
x	Le catalogue peut être accédé par CD et listé. Sans cette autorisation il est impossible d'accéder au répertoire et d'agir sur son contenu qui devient verrouillé.

La commande qui vous permet de **changer les droits d'accès POSIX d'un fichier/répertoire** est **« chmod »**. Le paramètre **« -R »** permet de le faire de manière récursive.
La commande qui vous permet de **changer le propriétaire** (nom ou UID **ou le groupe** (groupe ou GID)) **du fichier/répertoire** est **« chown »**. Le paramètre **« -R »** permet de le faire de manière récursive. Cette commande **s'appuie sur « /etc/passwd » et « /etc/group »** pour travailler.

Exemples (chown)



Exemples (chmod) :



Vous pouvez travailler avec les **caractères** (r,w ou x) ou en **octal**.
Un **administrateur** travaille en **général en octal** car cela est beaucoup plus rapide, c'est une question d'habitude en fait.

Faites très attention lorsque vous vous servez des options de récursivités, vous pouvez mettre hors d'usage votre système GNU/Linux si vous appliquez des droits de manière inconsidérée.

GNU/linux est un savant assemblage de fichiers, répertoires qui sont sécurisés de façon minutieuse avec les droits POSIX : soyez donc vigilants avec les commandes « chmod » et « chown » elles sont très puissantes.

Lors de l'intégration sur votre serveur d'un logiciel installé via une copie de fichier soyez vigilant quant aux réglages des droits sur le(s) répertoire(s) d'installation, bien souvent il s'agit de la cause principale des soucis d'installation.

Lorsque vous créez un nouveau répertoire ou un nouveau fichier vous observez que les droits sont positionnés par défaut. On appelle cette caractéristique le **« umask », il est positionné à 002 sous Centos** (022 pour les utilisateurs avec un uid < 100, donc les comptes de services). La commande **« umask »** vous permet de prendre connaissance du « umask » en cours.

Sur une Centos par défaut :

- Les **fichiers** sont créés avec un « **umask »** à 002 (cf. **/etc/bashrc**) soit le complément **664** (pour 666)
- Les **répertoires** sont créés en **775**.

Le **« umask »** se retrouve sur tout système de fichiers, qu'il soit :

- Local,
- Réseaux,
- Virtuel,
- Etc.

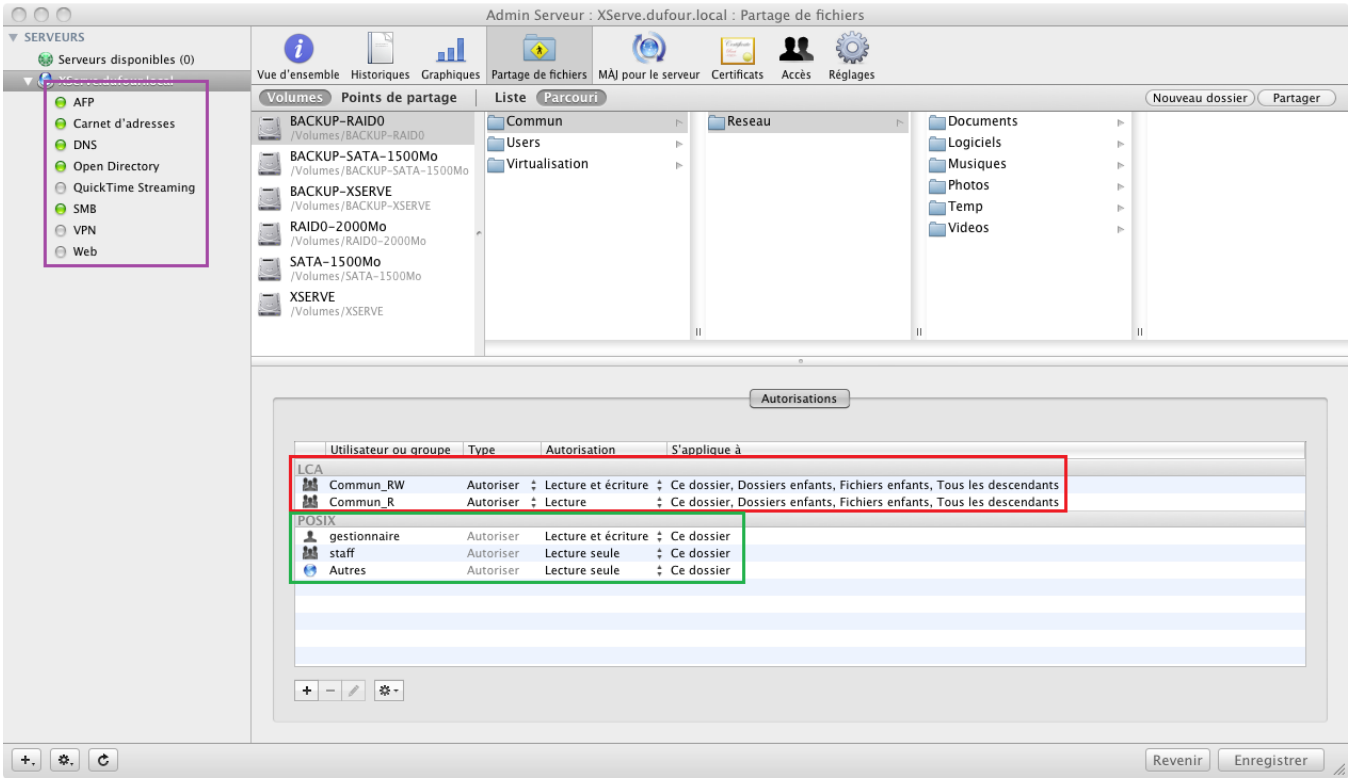
Il faut savoir qu'il existe 3 d'autres types de droits :

- **Les droits étendus SUID, SGID** (s) : appliqués à une commande ils permettent à cette commande de s'exécuter avec les droits du propriétaire ou du groupe d'appartenance de la commande, et non plus avec les droits de l'utilisateur l'ayant lancée.
- Le **« Sticky bit » (t)** : il permet d'affecter une protection contre l'effacement du contenu d'un répertoire
- **Le droit « s » appliqué au groupe sur un répertoire** : tous les fichiers créés au sein de ce répertoire, et quel que soit le groupe de la personne créant ce fichier, seront du même groupe que ce répertoire.

Pour aller plus loin vous : les **ACL étendues**

Elles proposent la même granularité que celles du système de fichiers NTFS.
Sachez qu'elles fonctionnent parfaitement avec le couple SAMBA/LDAP.
Elles se manipulent avec les commandes **« setfacl »** et **« getfacl »**.
Cependant attention : tous les services réseau etc. ne les supportent pas forcément

Un exemple de parfaite intégration des ACL étendues et POSIX se trouve chez Mac OS X serveur.



Les quotas disques

Les **quotas** permettent de poser des limites à l'utilisation de systèmes de fichiers. Ces limites sont de deux types :

- **inodes** : limite le nombre de fichiers.
- **blocs** : limite la taille disque.

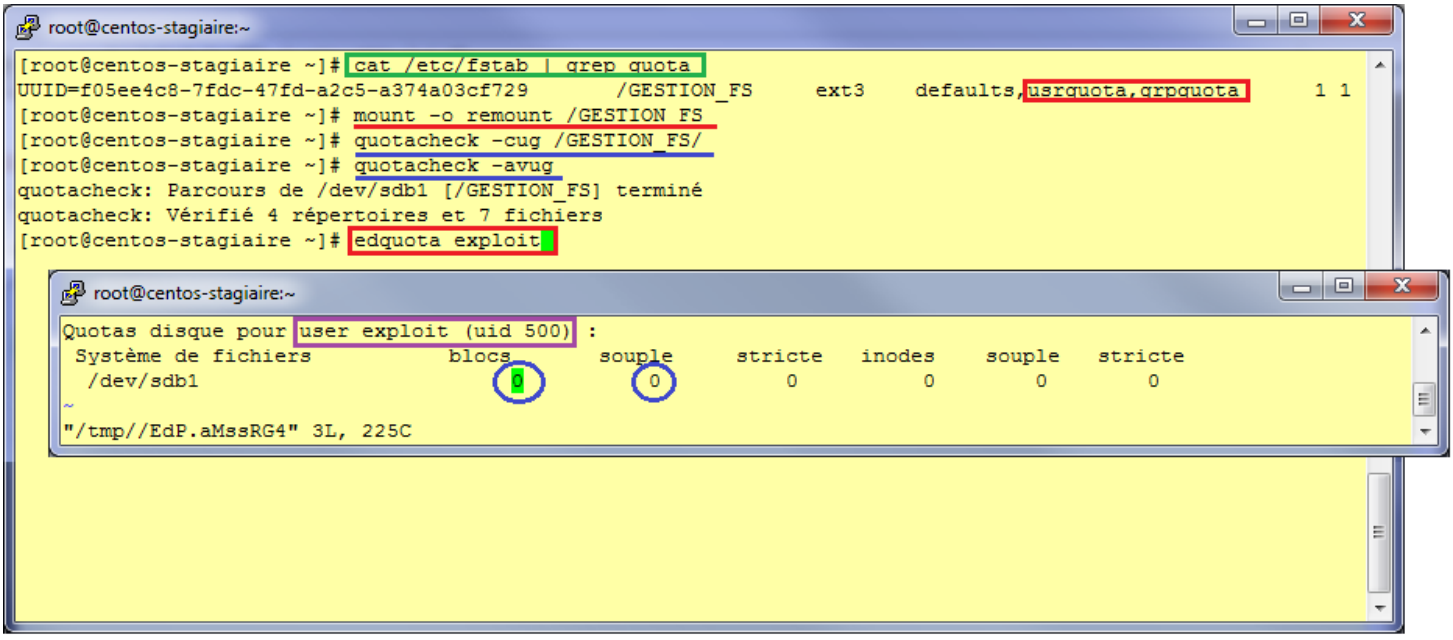
Les quotas sont implémentés par système de fichiers individuel et pas pour l'ensemble des systèmes de fichiers. Chaque utilisateur peut être géré de manière totalement indépendante. Il en est de même pour les groupes.

Pour chaque utilisation (**inode** ou **bloc**), vous pouvez mettre en place deux limites dans le temps :

- **Limite stricte** (hard) : quantité maximale d'inodes ou de blocs utilisés que l'utilisateur ou le groupe ne peuvent absolument pas dépasser. Dans ce cas, plus rien ne sera possible (création de fichier ou fichier dont la taille dépasse la limite).
- **Limite souple** (soft) : quantité maximale d'inodes ou de blocs utilisés que l'utilisateur ou le groupe peuvent temporairement dépasser. Dans ce cas, les créations et modifications seront possibles jusqu'à un certain point : limite dure et délai de grâce.
- **Un délai de grâce** est mis en place. Durant ce temps, l'utilisateur peut continuer à travailler sur le système de fichiers. Le but est qu'il revienne à terme sous la limite douce. Le délai dépassé, la limite douce devient la limite dure. Quoi qu'il arrive, l'utilisateur ne pourra jamais dépasser la limite dure.

Les quotas sont implémentés dans le noyau Linux et au sein des systèmes de fichiers. Pour les utiliser, les outils de quotas (packages quota) doivent être installés, ceux-ci sont déjà installés par défaut sur Centos.

Voici un exemple de mise en place de quota (simple) :



La sauvegarde et la restauration des données

Le travail numéro un de l'administrateur système est de s'assurer chaque matin qu'il va pouvoir travailler en toute sécurité durant la journée : **la sauvegarde des données est sa première priorité.**

En effet on peut se permettre de perdre une machine virtuelle, un serveur, un disque dur, une imprimante, une application etc. mais les **données des utilisateurs**, qu'elles soient sous forme de fichiers ou de bases de données ne peuvent pas être régénérées en partant de rien.

Seule la sauvegarde exploitable permet de sortir intact d'un sinistre informatique. Alors … sauvegardez parfaitement vos données sensibles.

Le but n'est pas de faire un cours sur la sauvegarde mais sachez toutefois que la sauvegarde est un projet à part entière, faisant partie intégrante de la sécurité informatique d'une entreprise.
Vous devez élaborer une stratégie globale et centralisée de sauvegardes de vos données.
L'audit est forcément de rigueur.

Voici un aperçu des questions incontournables que vous devez vous posez dès le début de l'étude :

Questions	Réponses
Quelles sont les **données à sauvegarder** ?	
Quelle sera le **niveau de confidentialité** des données à sauvegarder ?	
Quelle sera la **volumétrie** des sauvegardes ?	
Quelle sera la **fréquence et le type** des sauvegardes envisagées ?	
Le **temps de conservation et nombre de copies** des sauvegardes souhaités?	
Quels seront les **lieux de stockage**, deux étant un minimum ?	
Quel sera le **support de sauvegarde** choisi ?	
Quelle sera la **plage horaire autorisée pour les sauvegardes** ?	
Contrainte concernant la disponibilité du SI (H24, 7/7 etc.) ?	
Mise en place d'un **réseau dédié de sauvegarde** ?	
Quel sera l'**outil de sauvegarde** ?	
Budgétisation achat, formation, maintenance, garantie, support, consommable … ?	
Elaboration d'un **plan de reprise après sinistre** ?	
Temps de rétablissement désiré après sinistre (GTR) ?	
Etc.	

Sur des petits sites les solutions de sauvegardes s'appuyant sur les outils de base intégrés à GNU/Linux et sur des scripts centralisés sont possibles.
Néanmoins dès que vous allez dépasser un certain nombre de serveurs à sauvegarder vous vous apercevrez rapidement qu'il sera obligatoire de passer par des solutions de stockage conséquentes qui dépassent le cadre cet ouvrage. La plupart de ce solutions sont propriétaires et onéreuses : les données n'ont pas de prix !

Avec GNU/Linux vous pouvez commencer par envisager les possibilités suivantes :

- Si vous désirez sauvegarder des fichiers ou une arborescence vous pouvez utiliser « **tar** » et « **cpio** »,
- Si vous désirez sauvegarder des partitions ou disque dur complets vous pouvez utiliser « **dd** »,

Couplés aux commandes suivantes vous allez pouvoir sauvegarder l'intégralité ou une portion de votre système GNU/Linux tout en compressant les données :

- **mt** : permet le contrôle d'une bande magnétique.
- **find** : commande générique de sélection des fichiers à sauvegarder.
- **compress** et **uncompress** : pour gagner de l'espace en compressant et décompressant les fichiers.
- **gzip**, **gunzip**, **zcat**, compression et décompression au format GnuZip.

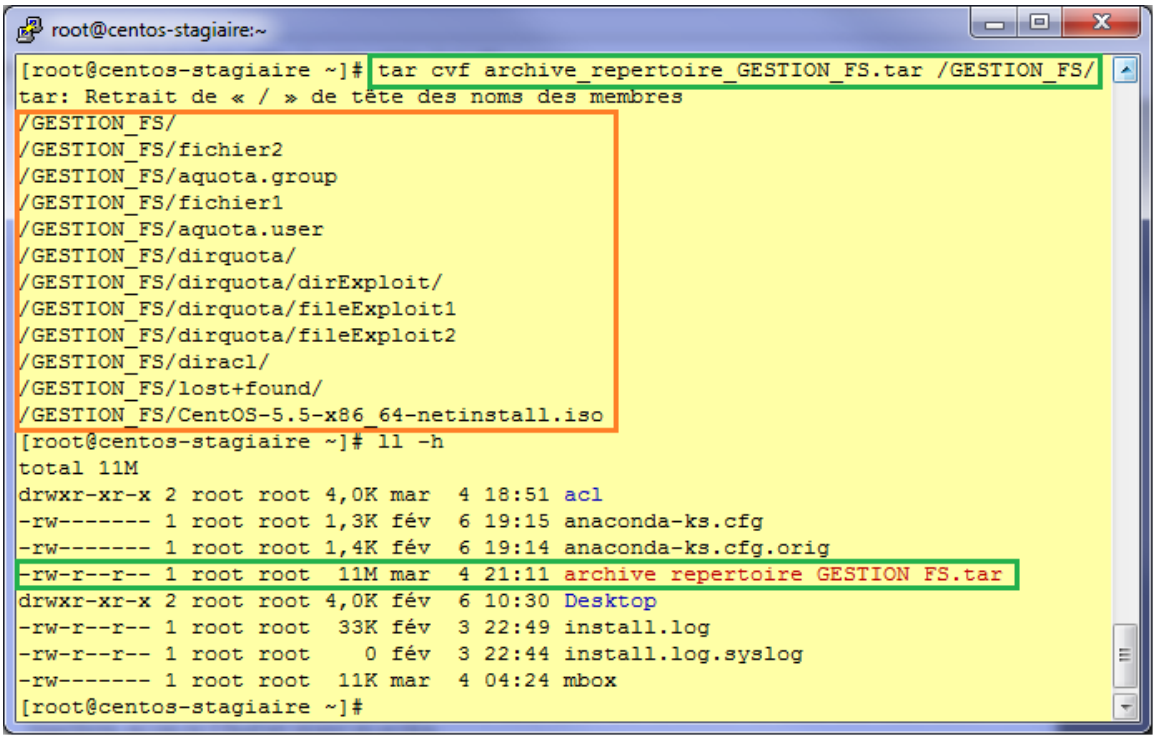
Tar

tar options fichier_d'archive [fichier ou répertoire à archiver]

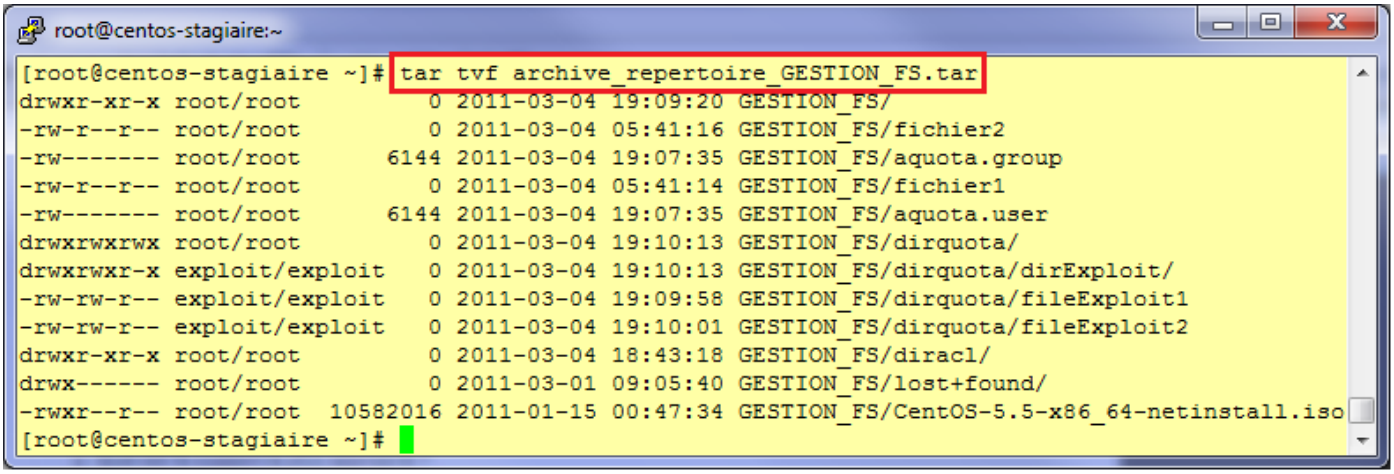
Cette commande permet **d'archiver le contenu partiel ou complet d'un système de fichiers**. Elle crée des archives des fichiers, y compris des arborescences de fichiers, sur tout type de support y compris dans un autre fichier (extensions **.tar**).

Options	Signification
c	Création d'un archive
x	Extraction d'un archive
l	Archivage du système de fichiers local
t	Consultation de l'archive
v	Mode détaillé (verbeux)
z	L'archive est compressée au format **gzip**
j	L'archive est compressée au format **bzip2**
f fichier	Spécification du fichier de l'archive

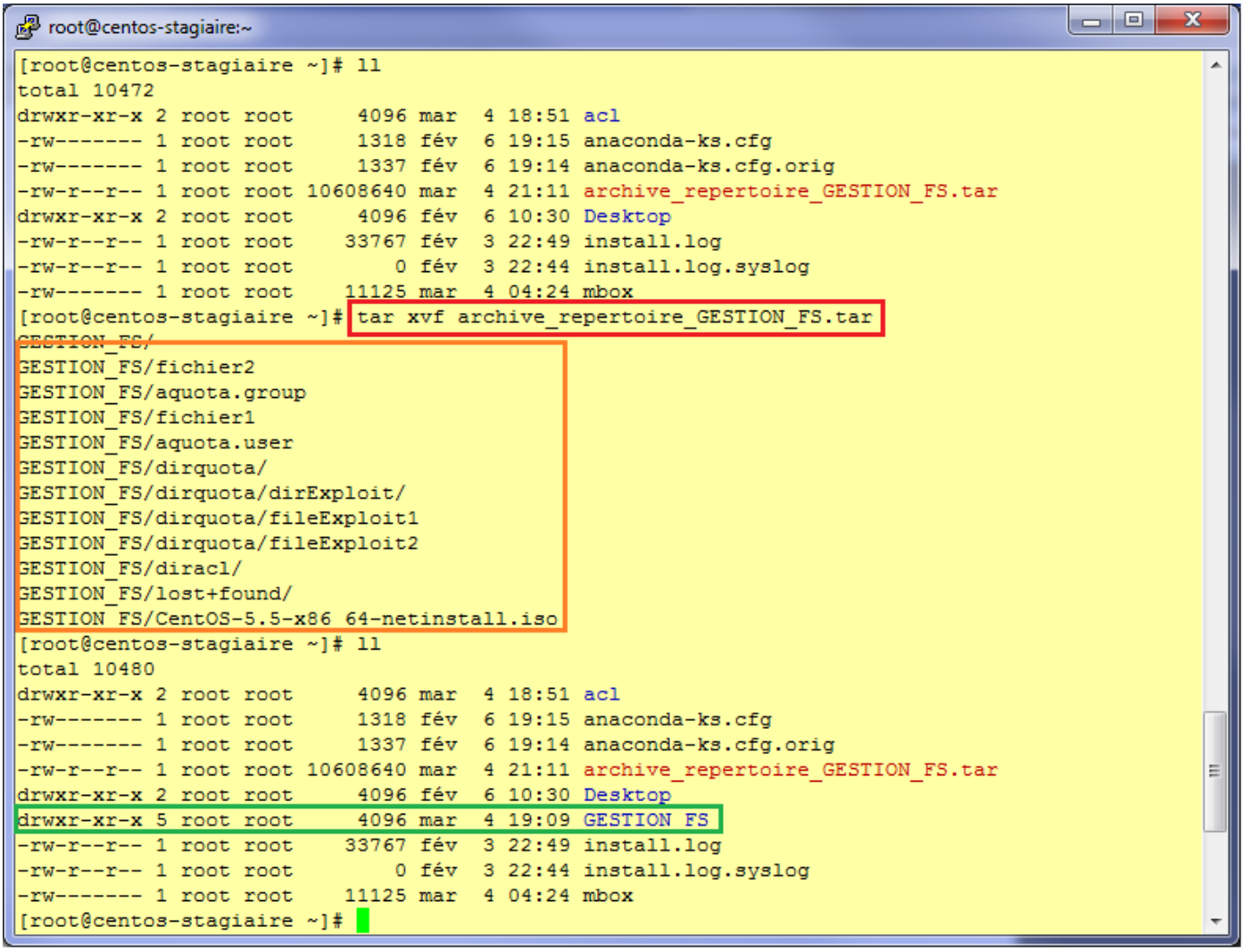
Archiver une arborescence de fichiers (ou un fichier) :



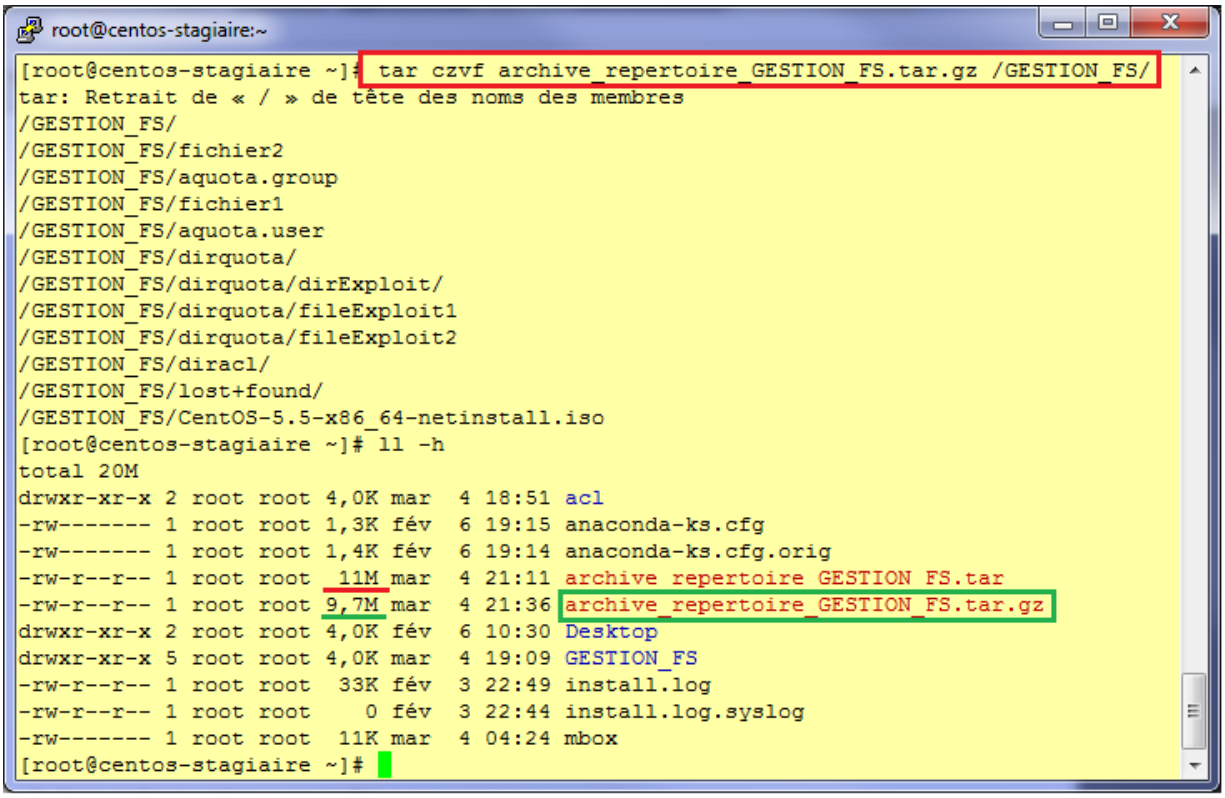
Lister le contenu d'une archive existante :



Restaurer le contenu d'un archive à l'endroit où vous trouvez (voir pwd) :


root@centos-stagiaire:~
[root@centos-stagiaire ~]# ll
total 10472
drwxr-xr-x 2 root root   4096 mar 4 18:51 acl
-rw------- 1 root root   1318 fév 6 19:15 anaconda-ks.cfg
-rw------- 1 root root   1337 fév 6 19:14 anaconda-ks.cfg.orig
-rw-r--r-- 1 root root 10608640 mar 4 21:11 archive_repertoire_GESTION_FS.tar
drwxr-xr-x 2 root root   4096 fév 6 10:30 Desktop
-rw-r--r-- 1 root root   33767 fév 3 22:49 install.log
-rw-r--r-- 1 root root   0 fév 3 22:44 install.log.syslog
-rw------- 1 root root   11125 mar 4 04:24 mbox
[root@centos-stagiaire ~]# tar xvf archive_repertoire_GESTION_FS.tar
GESTION_FS/
GESTION_FS/fichier2
GESTION_FS/aquota.group
GESTION_FS/fichier1
GESTION_FS/aquota.user
GESTION_FS/dirquota/
GESTION_FS/dirquota/dirExploit/
GESTION_FS/dirquota/fileExploit1
GESTION_FS/dirquota/fileExploit2
GESTION_FS/diracl/
GESTION_FS/lost+found/
GESTION_FS/CentOS-5.5-x86_64-netinstall.iso
[root@centos-stagiaire ~]# ll
total 10480
drwxr-xr-x 2 root root   4096 mar 4 18:51 acl
-rw------- 1 root root   1318 fév 6 19:15 anaconda-ks.cfg
-rw------- 1 root root   1337 fév 6 19:14 anaconda-ks.cfg.orig
-rw-r--r-- 1 root root 10608640 mar 4 21:11 archive_repertoire_GESTION_FS.tar
drwxr-xr-x 2 root root   4096 fév 6 10:30 Desktop
drwxr-xr-x 5 root root   4096 mar 4 19:09 GESTION_FS
-rw-r--r-- 1 root root   33767 fév 3 22:49 install.log
-rw-r--r-- 1 root root   0 fév 3 22:44 install.log.syslog
-rw------- 1 root root   11125 mar 4 04:24 mbox
[root@centos-stagiaire ~]#

Archiver **et** compresser les fichiers (ici format gzip, donc option : c**z**vf) :



Cpio

cpio -options < fichiers_à_sauvegarder

Cette commande lit sur l'entrée standard les fichiers à sauvegarder puis écrit sur la sortie standard la sauvegarde.
Archiver et compresser le répertoire /GESTION_FS :

- **find /GESTION_FS -print | cpio -ocv | gzip > archive_GESTION_FP.cpio.gz**

Lister l'archive :

- **cat archive_GESTION_FP.cpio.gz | gzip -cd | cpio -it**

Restaurer l'archive :

- **cat archive_GESTION_FP.cpio.gz | gzip -cd | cpio -iuvd**

Voici les options usuelles:

Options	Signification
o	Sauvegarde de fichiers, création de la sauvegarde en sortie
i	Restauration de fichiers (lecture de l'archive en entrée)
t	Consultation d'une archive
c	Mémorisation des attributs des fichiers
m	Préservation de la date de dernière modification des fichiers restaurés
d	Création des répertoires et sous-répertoires lors de la restauration
u	Restauration en écrasant les fichiers existants
B	Augmentation de la vitesse de cpio (tampon passe de 512 à 5120 octets)

Dd

dd options

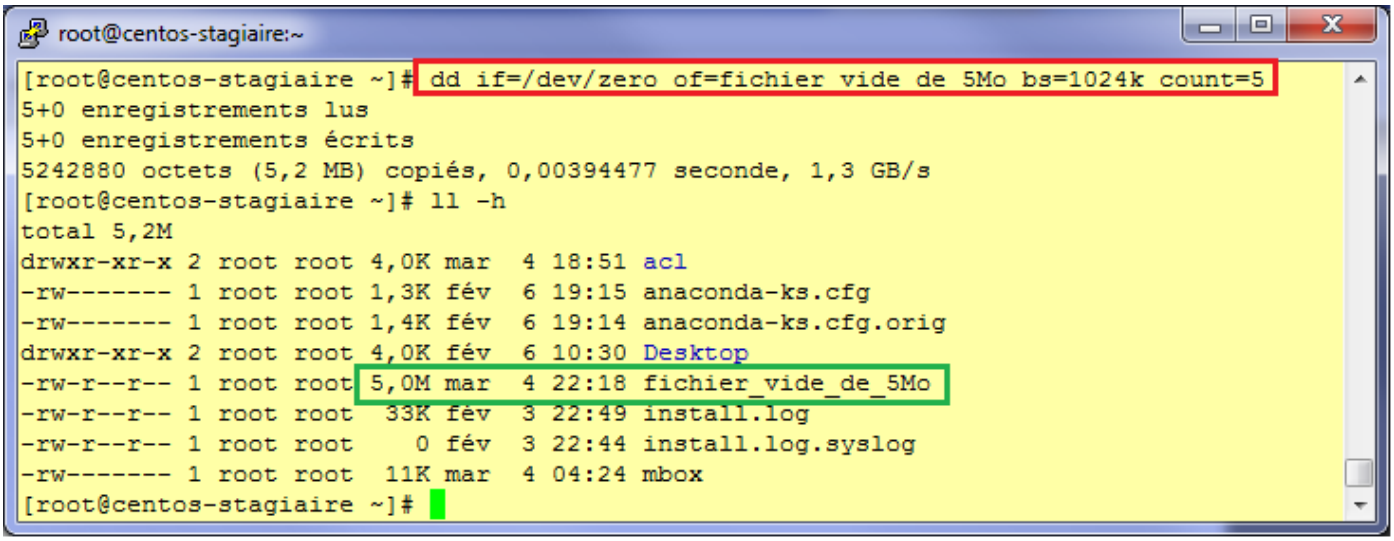
Littéralement « Device to Device », cette commande permet de faire une **sauvegarde bloc à bloc**. Elle travaille au niveau des blocs et n'a pas de vue concernant les systèmes de fichiers présents, encore moins des données présentes.
Elle permet de faire des copies physiques de disques et de systèmes de fichiers.

Options usuelles:

Options	Signification
if=fichier	Nom du fichier en entrée à utiliser (/dev/sdx, /dev/zero, /dev/random …)
of=fichier	Nom du fichier en sortie à utiliser
bs=n	Taille de bloc, en octets, utilisé par dd (défaut à 512 octets)
count=n	Nombre de blocs à copier
skip=n	Nombre de blocs à sauter au début du fichier d'entrée
seek=	Nombre de blocs à sauter au début du fichier de sortie
ibs=	Taille de bloc en entrée
obs=	Taille de bloc en sortie

Vous avez déjà vu la copie du MBR via **« dd »**, c'est-à-dire l'extraction des premiers 512 octets du disque de démarrage.

Voici comment créer un fichier de 5Mo (rempli de 0) :



Dump et Restore

Ces commandes permettent respectivement de sauvegarder puis de restaurer un système de fichiers de type ext2/ext3.
Avec on peut effectuer des sauvegardes incrémentales, c'est-à-dire une sauvegarde des fichiers modifiés par rapport à un niveau de sauvegarde précédent.

Rsync

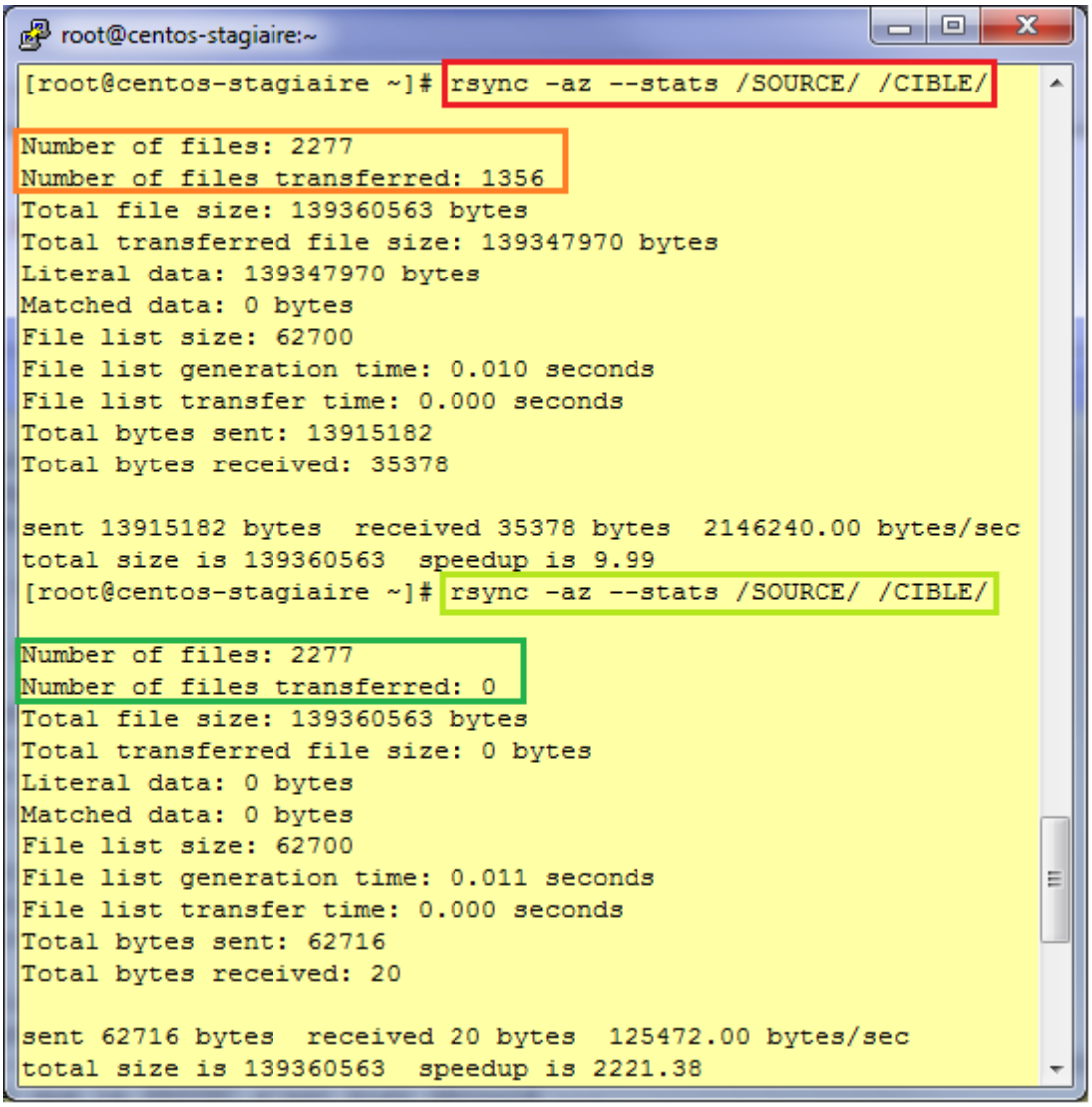
Rsync (*remote synchronization*, en français : « synchronisation distante ») est un logiciel libre de synchronisation de fichiers, distribué sous GNU GPL. La synchronisation est unidirectionnelle, c'est-à-dire qu'elle copie les fichiers de la source en direction de la destination. Rsync est donc utilisé pour réaliser des **sauvegardes incrémentales** ou pour diffuser le contenu d'un répertoire de référence.
Rsync fonctionne sur un large spectre de systèmes d'exploitation (Microsoft Windows, Linux, Mac OS X), permettant ainsi de synchroniser des fichiers de différents systèmes d'exploitation.

Cette commande est intéressante dans le sens où elle permet de **synchroniser :**

- **Des To de données en quelques minutes**,
- En **mode crypté** à fort chiffrement (OpenSSh),
- **A distance des arborescences** partielles ou complètes.

Dans le cas de dépôts Linux elle est très efficace.
Rsync est par exemple grandement utilisé pour **synchroniser les dépôts des distributions GNU/Linux** sur les miroirs des fournisseurs d'accès à internet (FAI), université, école etc.

Exemple d'utilisation en deux passes, la première longue, la deuxième ultra rapide (sans changements) :



Pour clôturer ce chapitre sur la sauvegarde ne perdez pas de vue qu'il est préférable de vous servir systématiquement de **l'outil de sauvegarde de l'application que vous utilisez**, si celle-ci en est dotée.
Exemple :

- Base MySql => mysqldump,
- Annuaire OpenLdap => slapcat,
- Base Oracle => export rman,
- Etc.

Ensuite vous pourrez vous servir de votre outil de sauvegarde favori pour **stocker l'export sur votre support de sauvegarde quotidienne**.

Note : parfois l'outil de sauvegarde intègre **un agent de sauvegarde prévu** pour le produit que vous souhaitez sauvegarder, servez-vous en !

Astuce : Les sauvegardes incrémentales impliquent d'avoir une sauvegarde complète opérationnelle etc. Alors **préférez les sauvegardes complètes** au quotidien. Cela reste la manière la plus simple et sure de travailler

Bien sûr il existe des solutions plus coûteuses qui faciliterons et surtout centraliserons grandement la tâche de sauvegarde de votre SI.
A vous de **mettre dans la balance le coût de l'acquisition d'un tel outil versus le temps**, donc l'argent, dépensé à exploiter, maintenir vos scripts etc.

Ne perdez pas de vue non plus que la **sauvegarde sur bande n'est pas une finalité**, certaines multinationales **sauvegardent sur disque dur via des SAN** redondants localisés sur plusieurs sites géographiques.

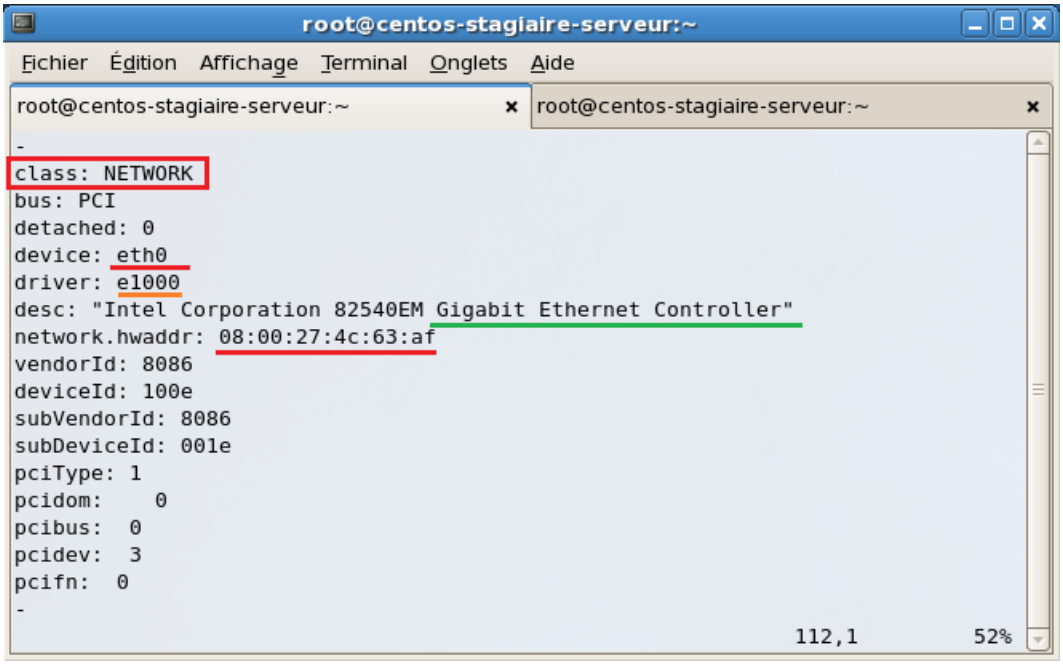
Configuration du réseau

Centos propose des outils de configuration graphique et en ligne de commande pour la partie réseau entre autre.
Mais il est nécessaire de mettre en garde le lecteur que les outils graphiques ne couvrent pas toujours toutes les options disponibles du mode ligne commande. Pire : parfois l'utilisation de l'outil graphique va supprimer certaines de vos modifications que vous avez apportées en ligne de commande.

Conclusion : utilisez au maximum la ligne de commande, elle assure le pilotage complet de votre serveur, à distance, à travers des liens réseaux à très faibles débits sans la moindre surprise. En plus vous savez exactement ce que vous apportez comme modification aux divers fichiers de configuration.

Les interfaces matérielles

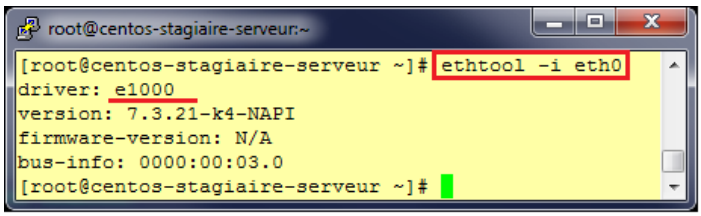
Commençons par recenser les interfaces réseaux disponibles sur votre système.
Elle(s) se trouve(nt) dans le fichier qui recense le matériel de votre serveur : **« /etc/sysconfig/hwconf »** :



La commande **« lspci »** donne aussi ce genre d'informations car la plupart des cartes réseaux sont connectées sur le bus **PCI**(e).

Vous pouvez également afficher et configurez vos périphériques réseaux avec **« ethtool »** ou **« mii-tool »** et **« mii-diag »**. Ceci de manière non persistante, pour figez les paramètres faites comme ci-dessous.
Normalement l'auto-négociation fera correctement son travail mais si vous désirez brider la vitesse de votre carte réseau, à 100Mbits par exemple, vous devez éditer le fichier **« /etc/sysconfig/network-scripts/ifcfg-eth0 »** et ajouter l'option suivante :

ETHTOOL_OPTS="autoneg off speed 100 duplex full"

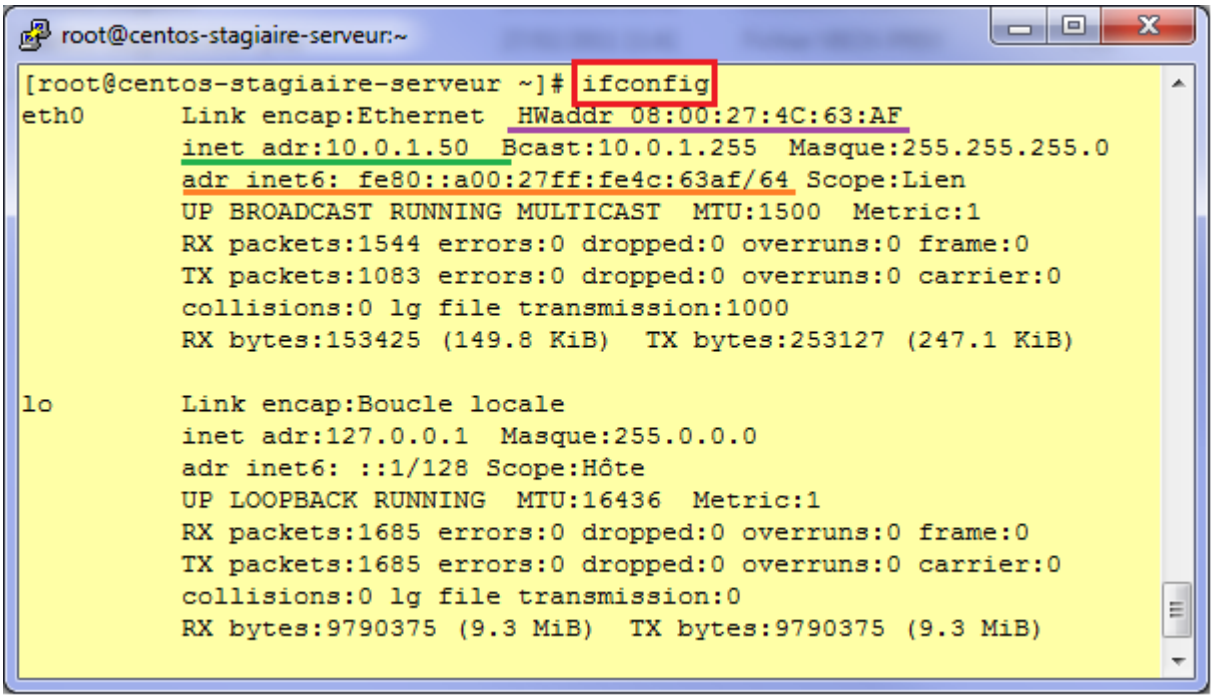


Configuration de base

Vous pouvez configurer vos paramètres réseau de deux manières différentes :

- L'une immédiate mais qui ne restera pas active après un redémarrage,
- L'autre persistante, c'est cette dernière que nous allons privilégier.

Quelques mots sur la première.
La commande **« ifconfig »** permet de visualiser les paramètres de votre interface réseau mais aussi de la **paramétrer** de façon **non persistante**.



Voici quelques exemples d'utilisation de **« ifconfig »** :

Options	Signification
ifconfig -a	Liste toutes les interfaces réseaux reconnues par le noyau et dont le pilote (module) est actif(carte active ou non)
ifconfig <interface_réseau> {up|down}	Active ou désactive l'interface réseau
ifconfig <interface> <adresse_ip> <masque_ss_réseau>	Paramétrage de base d'une interface réseau

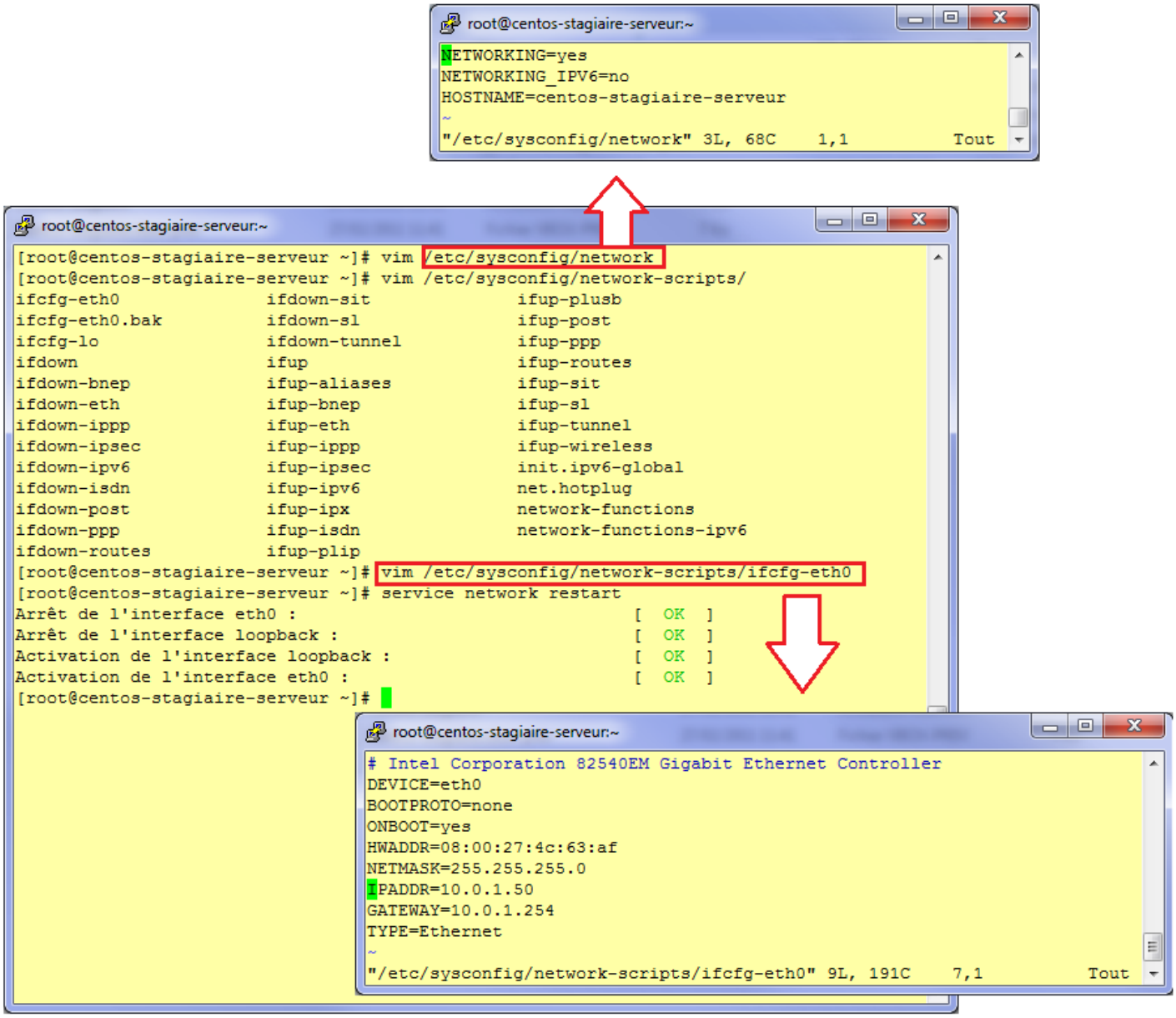
Maintenant nous allons passer à la **méthode persistante**.

Voici une liste des fichiers à modifier pour changer vos paramètres d'interfaces réseau (exemple : **eth0**) :

- **/etc/sysconfig/network,**
- **/etc/sysconfig/network-scripts/ifcfg-ethxxx.**

Seul le deuxième fichier « /etc/sysconfig/network-scripts/ifcfg-ethxxx » est à modifier pour changer votre adresse IP.

Pour information :
Le fichier **« /etc/sysconfig/nework »** contient les paramètres communs à toutes les interfaces réseaux.
Alors attention de ne pas y glisser n'importe quel paramètre, vous auriez des surprises !



Pour changer votre adresse IP il suffit d'éditer le fichier **« /etc/sysconfig/network-scripts/ifcfg-eth0 »**.
En renseignant correctement les champs suivants :

- **IPADDR** (l'adresse IP),
- **NETMASK** (le masque de sous-réseau),
- **GATEWAY** (l'adresse IP de la passerelle de votre réseau).
- **BOOTPROTO** (type d'assignation des paramètres réseaux : static, dhcp ou none),
- **ONBOOT** (démarrage de l'interface au boot, yes)

Ensuite vous devez valider votre configuration en **relançant (restart) le service (démon) réseau** :

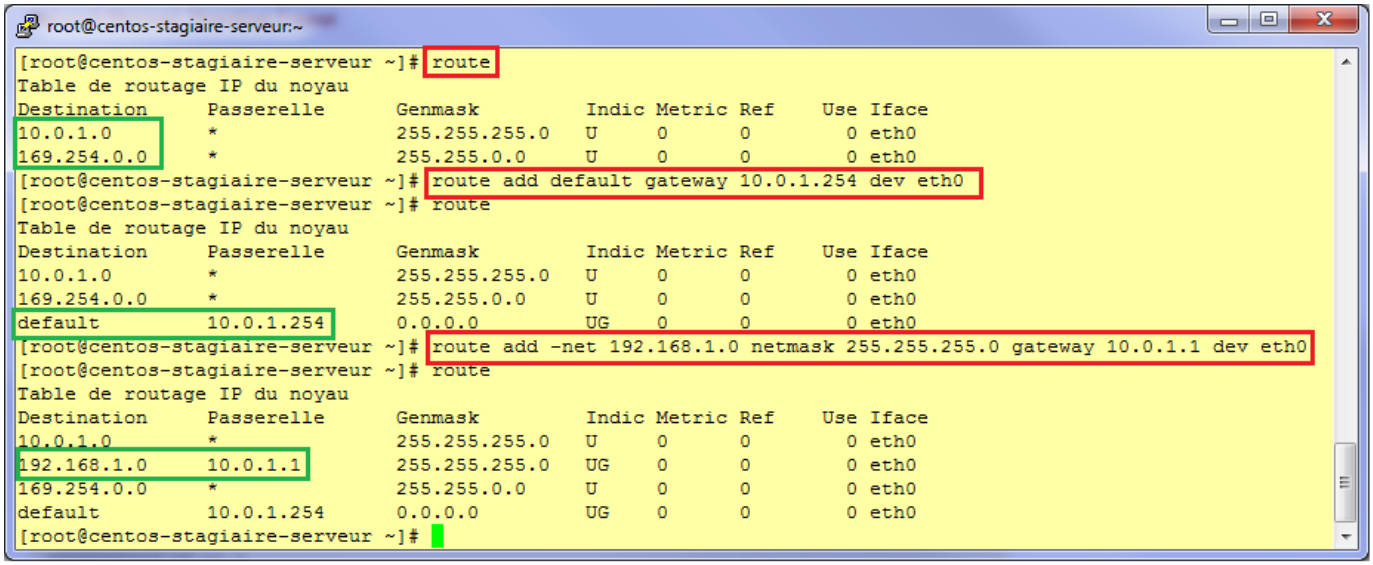
service network restart

Evitez « ifdown ethxxx » car si vous êtes en administration à distance (99% des cas) vous allez perdre votre session et serez obligé de vous rendre en salle machine pour lancer localement **« ifup ethxxx »**.

Vous pouvez également passé par un GUI en mode texte, en lançant l'outil de configuration de base de votre distribution Centos : **« setup », puis « Configuration du réseau »**
Note : faire « service network restart » après être sorti de l'outil.

Le routage

Le routage permet de déterminer si une machine destinataire est sur le même réseau que vous ou non.
Les paramètres de routage sont situés dans une table de routage.
Sous GNU/Linux vous pouvez manipuler cette table de routage avec la commande **« route »**.
Attention car là encore vous vous trouvez en mode non persistant.



Dans l'exemple précèdent nous avons :

- **Affiché la table de routage**,
- Ajouté une **passerelle par défaut**,
- Ajouté une **route statique.**

Pour paramétrer de **façon persistante une route par défaut** vous devez éditer le fichier ci-dessous.

- **« etc/sysconfig/network-scripts/ifcfg-ethxxx »**

Pour paramétrer plus en détail votre table de routage, avec **plusieurs routes par interface de manière persistante**, vous devez éditer le fichier de configuration suivant :

- **/etc/sysconfig/network-scripts/route-ethx**

Pour plus d'information concernant les **routes statiques** n'hésitez pas à consulter la **documentation officielle Centos**/RedHat©.
http://www.centos.org/docs/5/html/5.1/Deployment_Guide/s1-networkscripts-static-routes.html

La résolution de nom

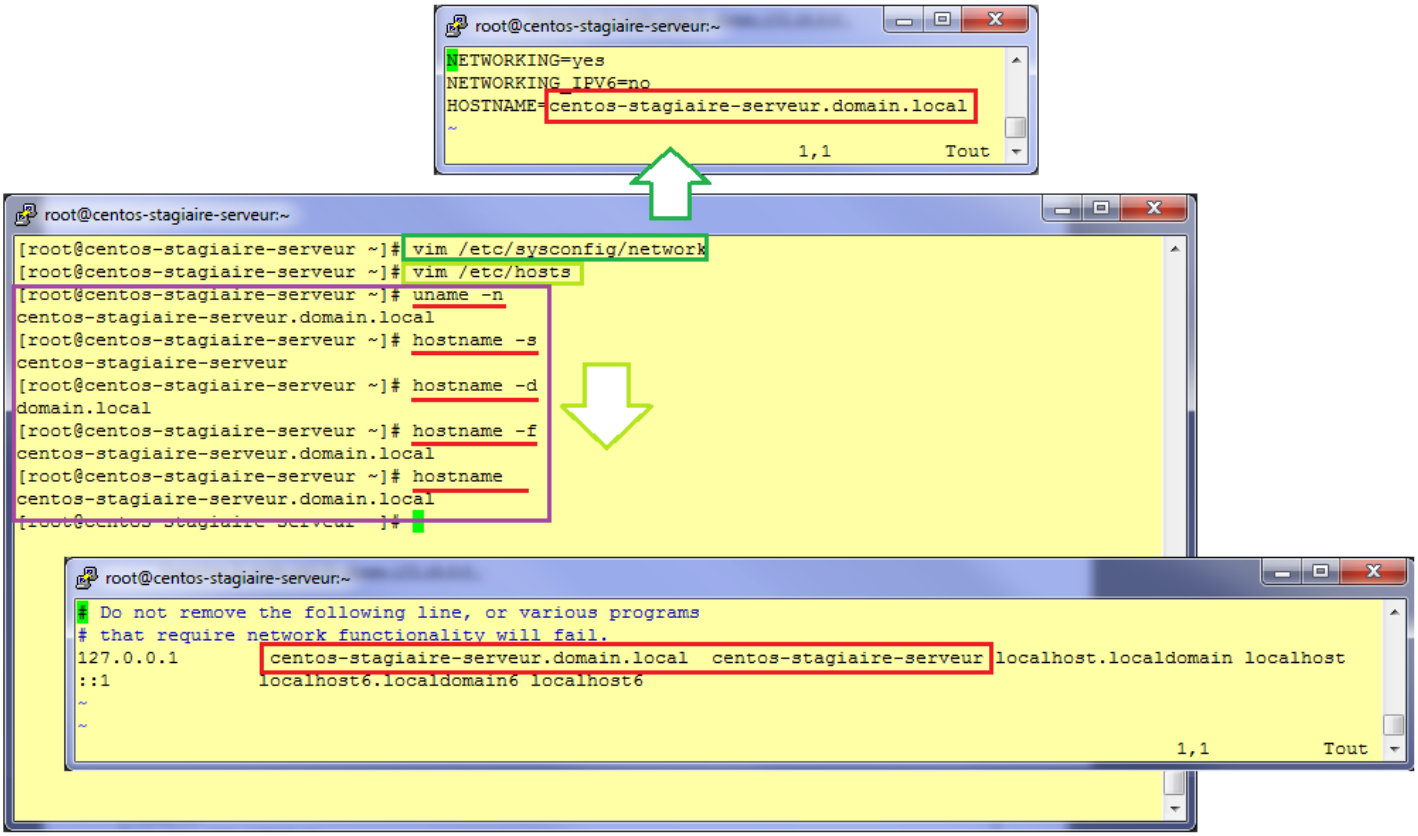
Elle permet de faire correspondre à une adresse IP un nom FQDN (**F**ully **Q**ualify **D**omaine **N**ame) et inversement (reverse).
Ex. : 10.0.1.50 ⇔ centos-stagiaire-serveur.domain.local

Sur Centos quatre fichiers contrôlent la résolution des noms parti client DNS, nous ne parlons pas ici de serveur BIND/DNS :

- **/etc/sysconfig/network,**
- **/etc/resolv.conf,**
- **/etc/nsswitch,**
- **/etc/hosts.**

Nous allons **changer proprement le nom du serveur** puis vérifierons la configuration. Voici les fichiers qui rentrent en jeux pour cette opération, il va nous falloir les éditer :

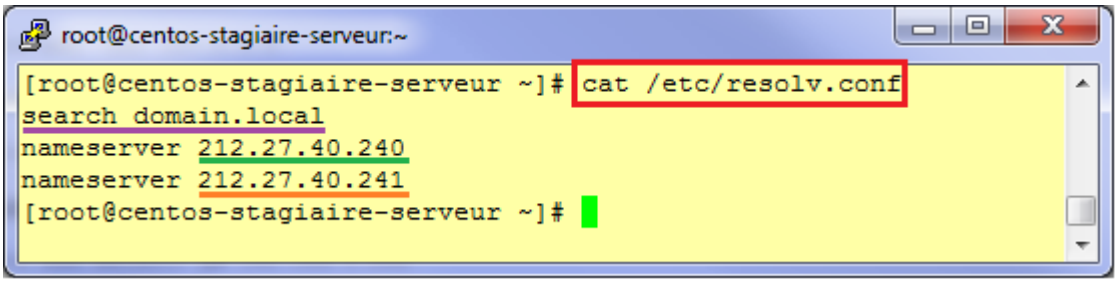
- **/etc/sysconfig/network,**
- **/etc/hosts.**



C'est dans le fichier **« /etc/hosts »** que vous pouvez inscrire en dur des entrées IP⇔NOM_HÔTE. Cependant il est préférable de s'appuyer sur **un (à trois) serveur(s) DNS** pour assurer les résolutions de nom (via le résolveur) dont aurait besoin votre serveur.

Pour ce faire nous allons utiliser le fichier **« /etc/resolv/conf »**.

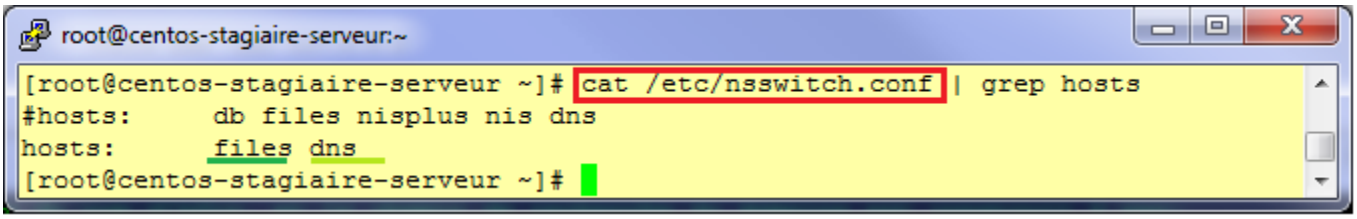
En général vous devez connaître le ou les **serveurs DNS** de votre entreprise (vous ne pourrez en saisir que **trois dans ce fichier**). Le DNS est un serveur qui centralise toutes les entrées IP⇔NOM_HÔTE (et inverse).
Il vous permet d'atteindre un autre serveur sans avoir à renseigner son adresse IP mais juste son nom d'hôte FQDN (dans un navigateur internet par exemple).



- **domain** : nom du domaine local. Les requêtes sont généralement réduites à des raccourcis relatifs au domaine local. S'il est absent le nom du domaine doit être déterminé à partir du nom d'hôte complet : c'est la partie située après le premier « **.** ».
- **search** : liste des domaines de recherche. Par défaut lors de l'utilisation de raccourcis (noms d'hôtes courts) le « résolveur » lance une recherche sur le domaine défini par la ligne domain, mais on peut spécifier ici une liste de domaines séparés par des espaces ou des virgules.
- **nameserver** : adresse IP du serveur de noms (le serveur DNS). On peut en placer au maximum trois. Le « résolveur » essaie d'utiliser le premier. En cas d'échec (timeout), il passe au second, et ainsi de suite.
- **options** : des options peuvent être précisées. Par exemple **timeout:n** où n (en secondes) indique le délai d'attente de réponse d'un serveur de noms avant de passer au suivant.

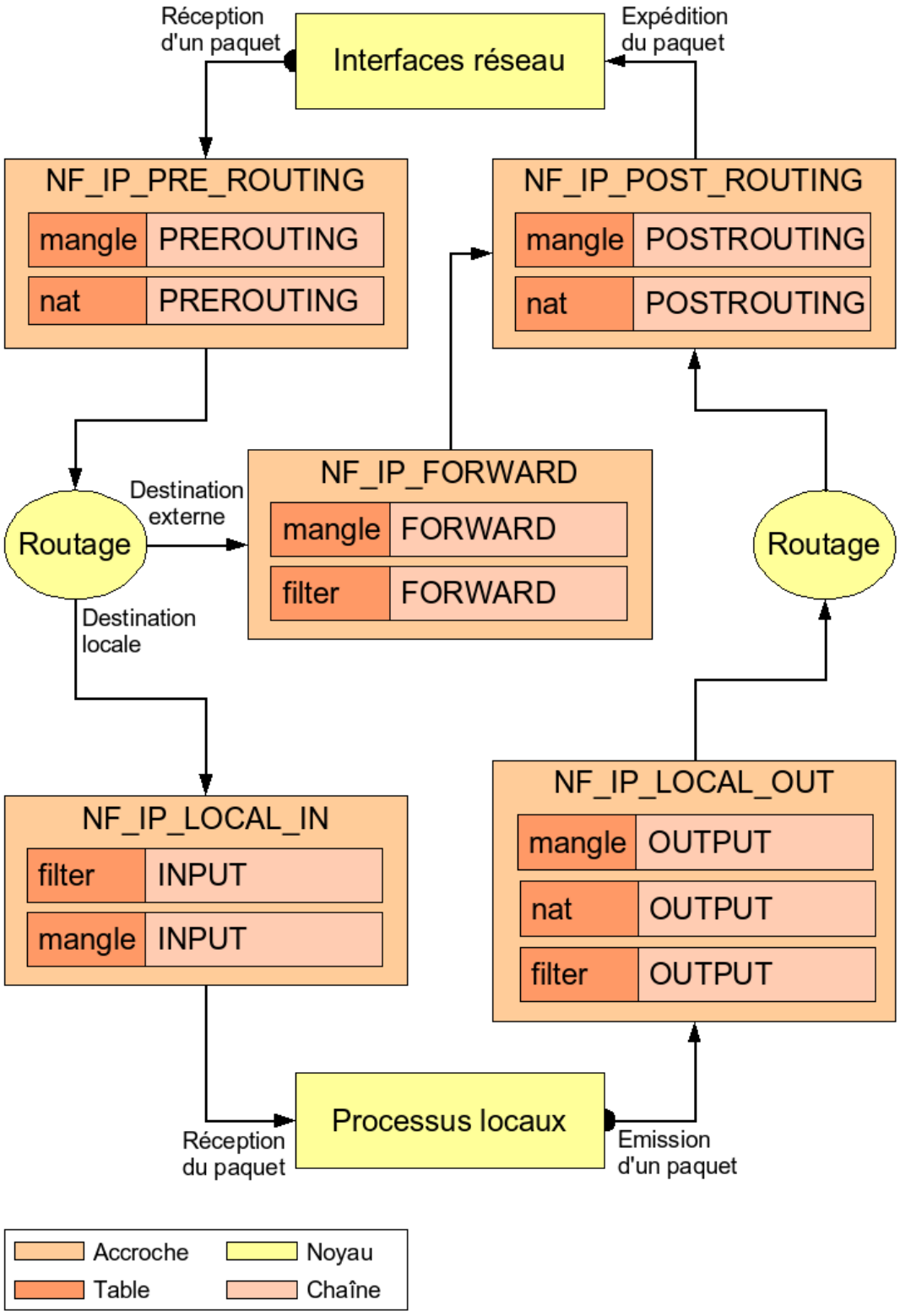
Le fichier est lu par le « résolveur » (librairies en C) de **résolution de nom de haut en bas**, donc on s'arrête au premier DNS qui répond.

Le **mécanisme de résolution de nom qui est prioritaire** pour le « résolveur » (fichier /etc/hosts ou /etc/resolv.conf) est **définit** dans le fichier « **/etc/nsswitch.conf »** à la ligne « hosts ».



La ligne « hosts » de l'exemple indique que lors d'une requête de résolution de nom les fichiers locaux sont prioritaires sur les serveurs de noms mentionnés.
Donc le fichier « **/etc/hosts »** est d'abord lu, puis, si le « résolveur » ne trouve pas l'information il passe par une résolution vi le serveur DNS mentionné dans « **/etc/resolv.conf »**.

~~*La sécurité : le NetFilter (IpTables)*~~



http://www.centos.org/docs/5/html/5.2/Deployment_Guide/ch-fw.html

Diagnostiques et statistiques réseaux

Voici quelques outils pour diagnostiquer la partie réseau.

Afficher les routes statiques :

- netstat -rn

Afficher les connexions réseaux TCP(t) sans résolution de nom DNS(n) avec tout (a) leurs processus associés(p) :

- netstat -tpan

Afficher la route que suit un paquet sans résolution de nom (n) afin de déterminer le nœud réseau qui est défaillant :

- traceroute -n

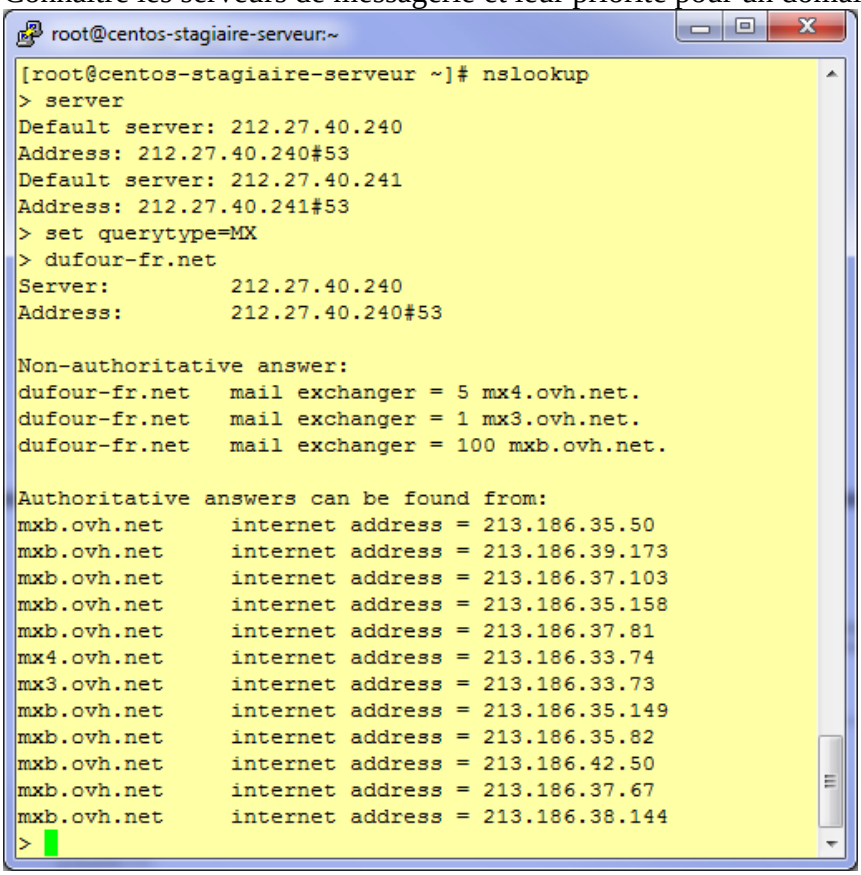
Tester la disponibilité d'un service réseau d'un serveur cible (**attention tout passe en clair**).
Il faut bien entendu connaître les commandes du protocole (service) que l'on souhaite vérifier.
Exemple : POP :110, SMTP :25, HTTP :80, SSH :22 etc.

- telnet <serveur> <service_réseau_testé>
- telnet ip 22

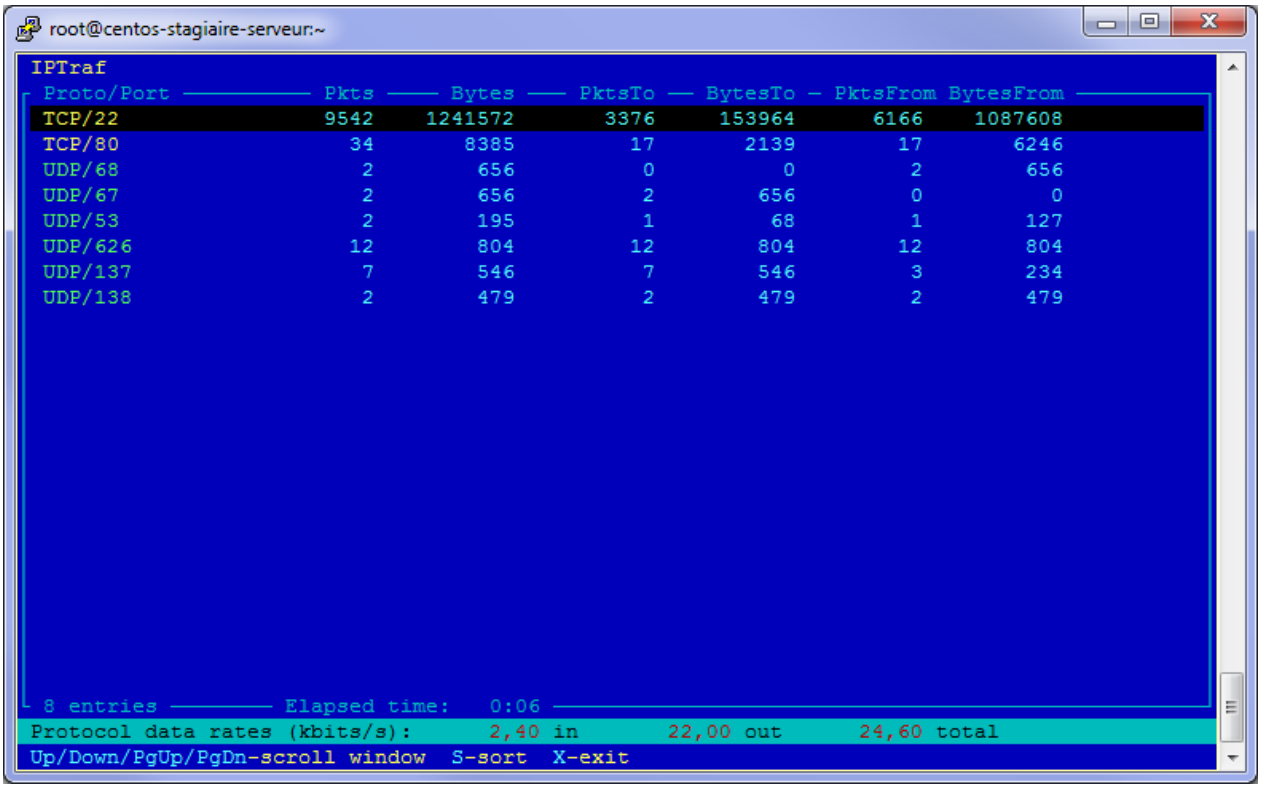
Tester la résolution de nom :

- nslookup <nom_hote>

Connaître les serveurs de messagerie et leur priorité pour un domaine particulier :



Pour visualiser en détail votre trafic réseau en temps réel utilisez **« iptraf » (yum install iptraf)** :



Il faut faire un **« miroring »** de port pour écouter le trafic dans un environnement commuté, une sonde d'écoute est en générale mieux adaptée :

- tcpdump

Si vous désirez connaitre les services réseaux connus par Centos :

- cat /etc/services | more

De même pour les protocoles usuels associés à leur port réseau :

- cat /etc/protocols | more

La gestion des utilisateurs

Nativement, et ce depuis sa genèse, GNU/Linux est un système multi-utilisateur là ou d'autres systèmes d'exploitation en sont encore à batailler pour faire en sorte qu'un programme puisse s'exécuter avec de simple droits utilisateur.
Sous GNU/Linux un compte utilisateur est représenté par un **UID (U**ser **IDentifier)** et un **GID (G**roup **IDentifier).**

Ce compte utilisateur permet d'effectuer des tâches sur le système avec des droits biens structurés.

On distingue deux types de comptes utilisateurs :

- Les comptes système : compris entre 0 et 500,
- Les comptes utilisateurs : 500 et au-delà.

Les **UID et les GID sont en principe uniques**. Le login est unique. Il est cependant envisageable d'associer plusieurs logins au même UID, le système travaillant parfois avec le login.
L'UID identifie l'utilisateur (ou le compte applicatif) tout au long de sa connexion. Il est utilisé pour le contrôle de ses droits et de ceux des processus qu'il a lancé. Ce sont les UID et GID qui sont stockés au sein de la table des inodes, dans la table des processus, etc., et non les logins.

L'utilisateur dispose des attributs de base suivants :

- un nom de connexion appelé le login ;
- un mot de passe ;
- un UID ;
- un GID correspondant à son groupe principal ;
- un descriptif ;
- un répertoire de connexion ;
- une commande de connexion (en général le BAsh) ;

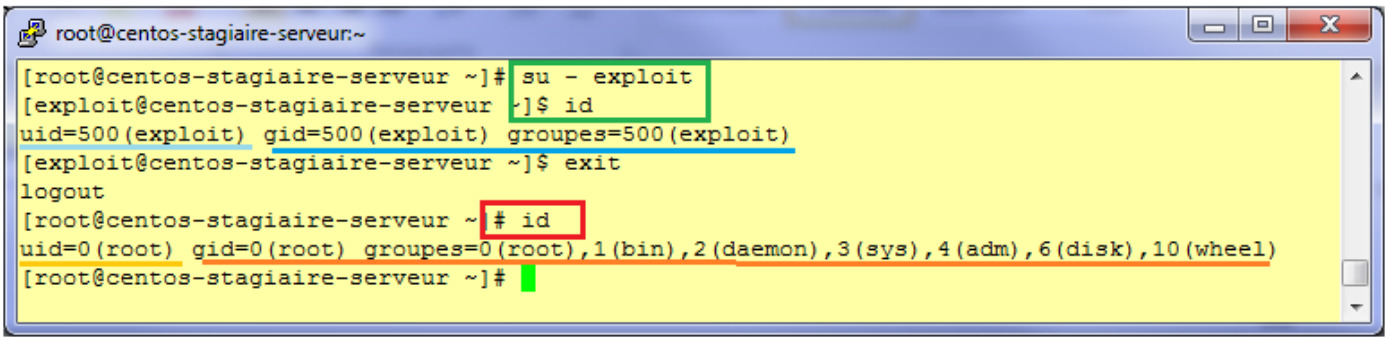
D'autres attributs sont disponibles via l'utilisation de la sécurité des mots de passe du fichier **/etc/shadow** (voir la section fichiers de configuration).
Les **UID d'une valeur inférieure à 100** sont en principe associés à **des comptes spéciaux** avec des droits étendus. Ainsi **l'UID de root, l'administrateur, est 0**. **A partir de 500**, et ce jusqu'à environ **60000**, ce sont les UID des **utilisateurs sans droits particuliers**.

Un login accepte la plupart des caractères. Il ne doit pas commencer par un chiffre. Il est possible de modifier la liste des caractères autorisés et de forcer la longueur et la complexité via les mécanismes d'authentification **PAM** du système ainsi que le fichier **/etc/login.defs**.

Chaque utilisateur fait partie d'au moins un groupe. Un **groupe regroupe des utilisateurs**. Comme pour les logins, le **GID du groupe accompagne toujours l'utilisateur pour le contrôle de ses droits**. Un **utilisateur** peut **faire partie de plusieurs groupes**, auquel cas il faut distinguer son groupe primaire des groupes secondaires.
Les groupes sont aussi des numéros. Il existe des groupes spécifiques pour la gestion de certaines propriétés du système et notamment l'accès à certains périphériques.

Il est impossible d'inclure les groupes entre eux.
Le groupe primaire est celui qui est toujours appliqué à la création d'un fichier

La commande **id** permet de connaître les informations essentielles sur un utilisateur : uid, gid, groupes secondaires.



A propos des mots de passe.

Dans un monde parfait il faudrait avoir recours exclusivement au **SSO** (**S**ingle **S**ign **O**n) : un seul mot de passe pour tous vos accès sécurisés.

Dans la pratique cela est difficilement réalisable de par la complexité des SI modernes et les multiples briques applicatives qui le constituent.
Toutefois dès que vous le pouvez, tentez de centraliser la gestion des mots de passe : l'annuaire est une des réponses à cette problématique.
De plus veillez à utiliser systématiquement des **technologies qui cryptent les mots de passe**.
Donc

- **telnet**,
- **ftp**,
- **lanman v1**,
- etc.

… ne doivent plus faire partie des options retenues : elles sont totalement vulnérables même en environnement commuté, sur internet n'en parlons même pas.

La généralisation d'accès par **carte à puce, biométrie** etc. est en passe de se généraliser. Le télétravail propose des accès par VPN, n'hésitez pas à vous doter de solutions sûres et efficaces, pas de bricolage.

Ne laissez aucun fichier de mot de passe accessible à tous (pensez droits d'accès), et surtout cryptez-les dès que cela est possible (md5, des, rsa, dsa etc.).

Vous devez également changer régulièrement de mot de passe, sur tous vos serveurs. Dans le cas où vous ouvrez vos sessions à l'aide d'**OpenLDAP** ou tous autres annuaires cela est relativement simple.

Enfin basez votre gestion des mots de passe sur une politique de sécurité mûrement réfléchie.

Vous pouvez vous aider d'un **outil de gestion de mot de passe robuste**. En effet, cela est moins pire que de tout noter dans un calepin, sur des post-it ou feuilles volantes : **KeyPass (GPLV3)**

Les fichiers de configuration

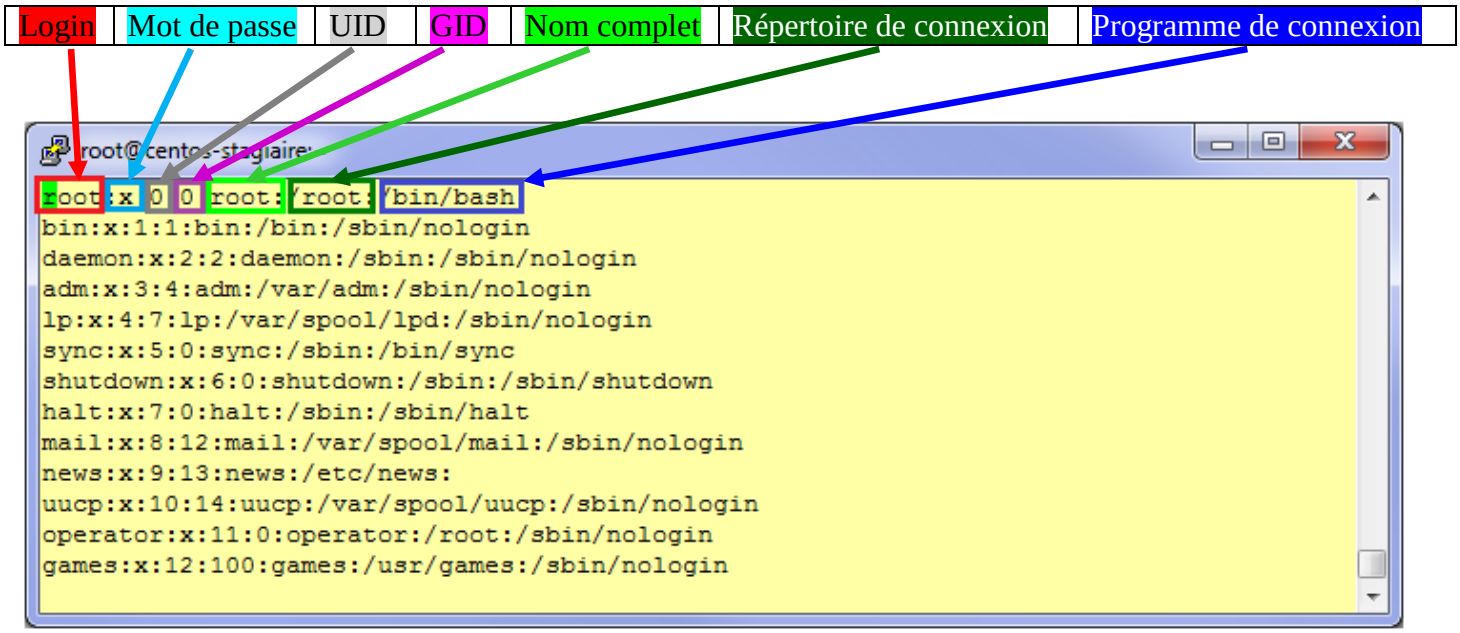
Les 3 fichiers dans lesquels sont stockés les informations des utilisateurs, groupes et mots de passe sont :

- /etc/password,
- /etc/group,
- **/etc/shadow/ qui est très sensible et doit avoir les droits suivant : r-- --- --- (400).**

Le fichier « /etc/password »

Ce fichier contient la liste **des comptes utilisateurs** de votre système GNU/Linux (comptes locaux).
Ce fichier peut être laissé lisible pour les utilisateurs standards.

Voici son format, le séparateur de champ est « : » :



Vous pouvez modifier ce fichier avec l'outil **« vipw » ou « vi »**.

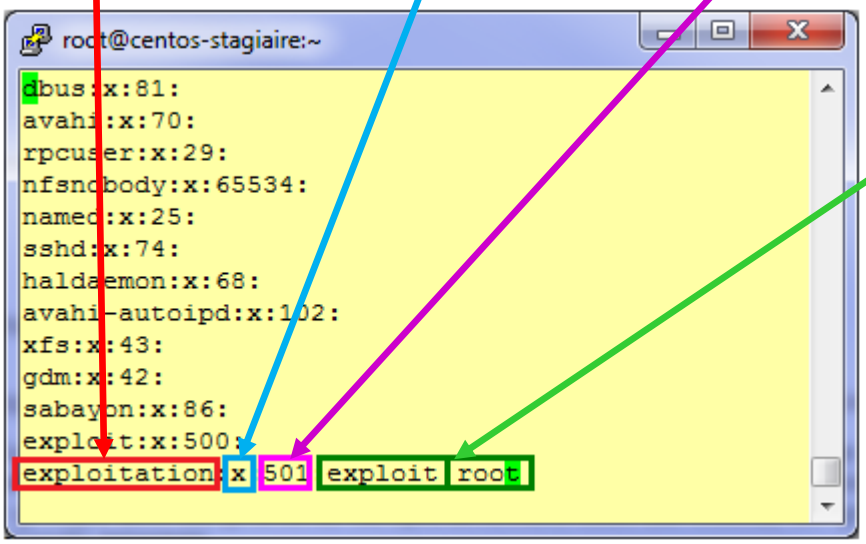
- Champ 1 : le login ou nom de compte de l'utilisateur,
- Champ 2 : sur les vieilles versions, le mot de passe crypté. Si un « **x** » est présent, le mot de passe est placé dans « **/etc/shadow** ». Si c'est un point d'exclamation **« ! »** le compte est verrouillé,
- Champ 3 : le User ID,
- Champ 4 : le Group ID, c'est à dire le groupe principal,
- Champ 5 : un commentaire ou descriptif. C'est un champ d'information, Ex. nom complet
- Champ 6 : le répertoire de travail, personnel, de l'utilisateur. C'est le répertoire dans lequel il arrive lorsqu'il se connecte. On peut choisir un autre répertoire également.
- Champ 7 : le Shell par défaut de l'utilisateur. Mais ce peut être toute autre commande, y compris une commande interdisant la connexion, Ex. **« /sbin/nologin »**.

Le fichier « /etc/group »

Ce fichier contient la **liste des groupes** de votre système GNU/Linux (**groupes locaux**) et pour chacun une liste des membres.
Ce fichier peut être laissé lisible pour les utilisateurs standards.

Voici son format, le séparateur de champ est « : » :

Nom du groupe	Mot de passe	GID	Membres du groupe



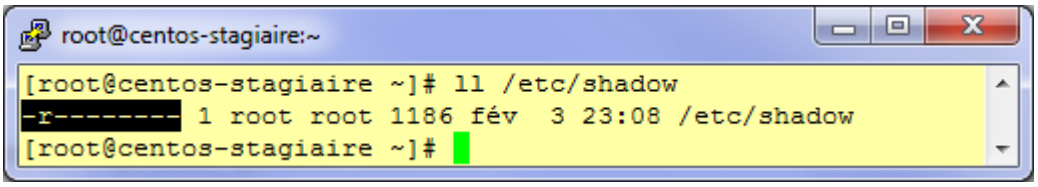
Vous pouvez modifier ce fichier avec l'outil **« vigr » ou « vi »**.

- Champ 1 : le nom du groupe.
- Champ 2 : le mot de passe associé
- Champ 3 : le Group Id.
- Champ 4 : la liste des membres appartenant à ce groupe.

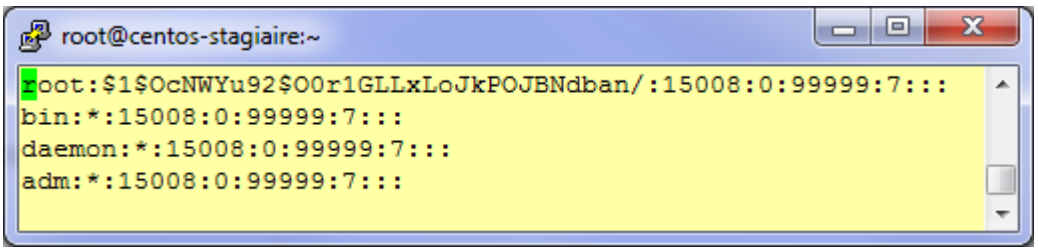
Enfin le fichier « /etc/shadow/ »

Le fichier **/etc/shadow** accompagne le fichier **/etc/passwd**. C'est là que sont **stockés**, entre autres, **les mots de passe crypté de chaque utilisateur**. Pour être plus précis il contient toutes les informations sur le mot de passe et sa validité dans le temps. Ce fichier peut compromettre tout votre système s'il est dérobé.

Ce fichier doit **OBLIGATOIREMENT** avoir les **droits POSIX** positionné comme suit (en **400**) :



Chaque ligne est composée de 9 champs séparés par des « : » « : »



- Champ 1 : le login.
- Champ 2 : le mot de passé crypté. Le xx initial indique le type de cryptage (voir ci-dessous).
- Champ 3 : nombre de jours depuis le 1e r janvier 1970 du dernier changement de mot de passe.
- Champ 4 : nombre de jours avant lesquels le mot de passe ne peut pas être changé (0 : il peut être changé n'importe quand).
- Champ 5 : nombre de jours après lesquels le mot de passe doit être changé.
- Champ 6 : nombre de jours avant l'expiration du mot de passe durant lesquels l'utilisateur doit être prévenu.
- Champ 7 : nombre de jours après l'expiration du mot de passe après lesquels le compte est désactivé.
- Champs 8 : nombre de jours depuis le 1e r janvier 1970 à partir du moment où le compte a été désactivé.
- Champ 9 : réservé.

Le champ 2 est assez précieux :

- ***** signale un compte prédéfini,
- **!!** indique un compte verrouillé,
- **Vide** pas de mot de passe.

Le mot de passe peut être crypté en :

- md5 (valeur 1),
- sha256 (valeur 5),
- sha512 (valeur 6).

Dans les lignes données en exemple, les mots de passe sont cryptés en md5. Ils ne sont pas cryptés seul, pour renforcer la résistance aux attaques. Ils sont cryptés en utilisant un **"salt"**, valeur ajoutée au mot de passe qui permet de ne jamais obtenir la même empreinte même si le mot de passe est le même

Les algorithmes utilisés pour créer les empreintes donnent des résultats à taille fixe. Ainsi, quelle que soit la taille du mot de passe, l'empreinte fait la même taille et ne donne pas d'indication quant à la longueur du mot de passe.

Vous pouvez vérifier la cohérence de vos fichiers **/etc/passwd** et **/etc/shadow** avec la commande suivante :

- pwck

Note : **grck** fait la même chose sur les groupes.

La Gestion des comptes

Etudions maintenant les commandes qui vont vous permettre de gérer facilement les utilisateurs et groupes contenus dans les fichiers de configurations vus précédemment.

La première commande à connaître et celle qui vous permettra de positionner un mot de passe après avoir créé un compte :

- passwd exploit
- New UNIX password:
- Retype new UNIX password:

Note : l'option « -l », verrouille le compte. L'option « -u » déverrouille le compte.

Voici la commande pour connaître l'UID, GID et les groupes secondaires auxquels appartient un utilisateur, saisissez la commande suivante :

- id
- uid=0(root) gid=0(root) groupes=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)

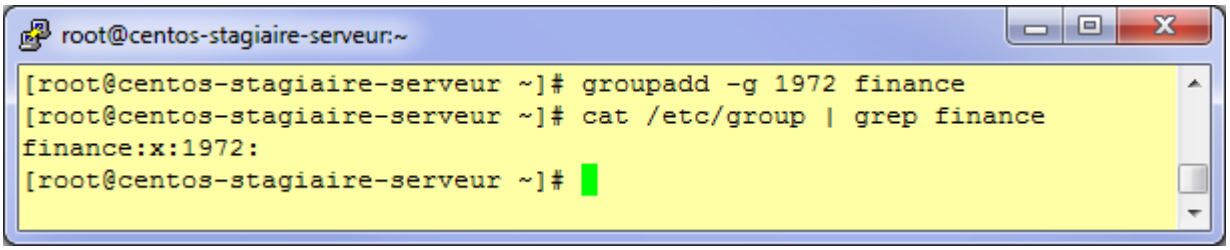
Ou encore

- id exploit
- uid=500(exploit) gid=500(exploit) groupes=500(exploit)

Pour connaître les groupes auxquels un utilisateur appartient:

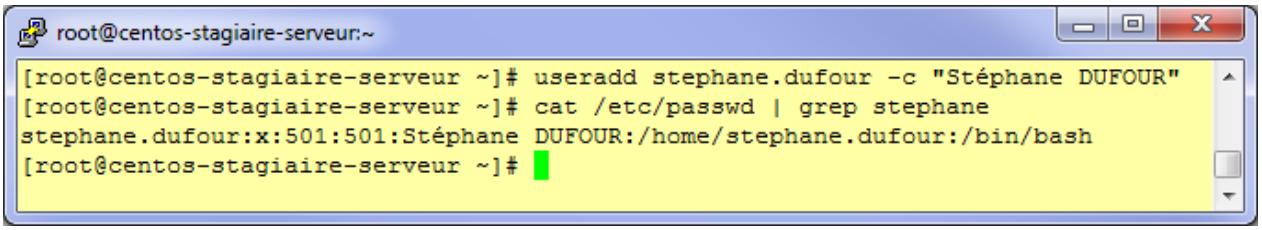
- groups root
- root : root bin daemon sys adm disk wheel

Pour ajouter un groupe au fichier **« /etc/group »**, utilisez la commande « **groupadd »** :



Note : **groupmod** = modification d'un groupe
groupdel = suppression d'un groupe.

La commande **« useradd »** permet l'ajout d'un compte dans le fichier **« /etc/passwd »** :



Options	Signification
-d	Spécifie le répertoire de connexion
-u UID	UID du compte
-g groupe primaire	Spécifie le groupe principal du compte
-G groupes	Listes des groupes secondaires
-m	Création du répertoire de connexion s'il n'existe pas
-k squelette	Copie des fichiers du squelette qui sont dans **/etc/skel/**
-s	Shell ou programme de connexion par défaut
-c	Commentaire (Ex. : Nom et prénom complet)

Note : **usermod** = modification d'un utilisateur (même option que useradd)
userdel = suppression d'un utilisateur (« **-r** » permet de supprimer également sa home directory).

La commande **« last »**, permet de dresser un listing des derniers utilisateurs connectés. **« lastlog »** fait à peu près la même chose.

Et toujours les journaux de connexions :

- tail -f /var/log/secure

Plus loin avec OpenLDAP

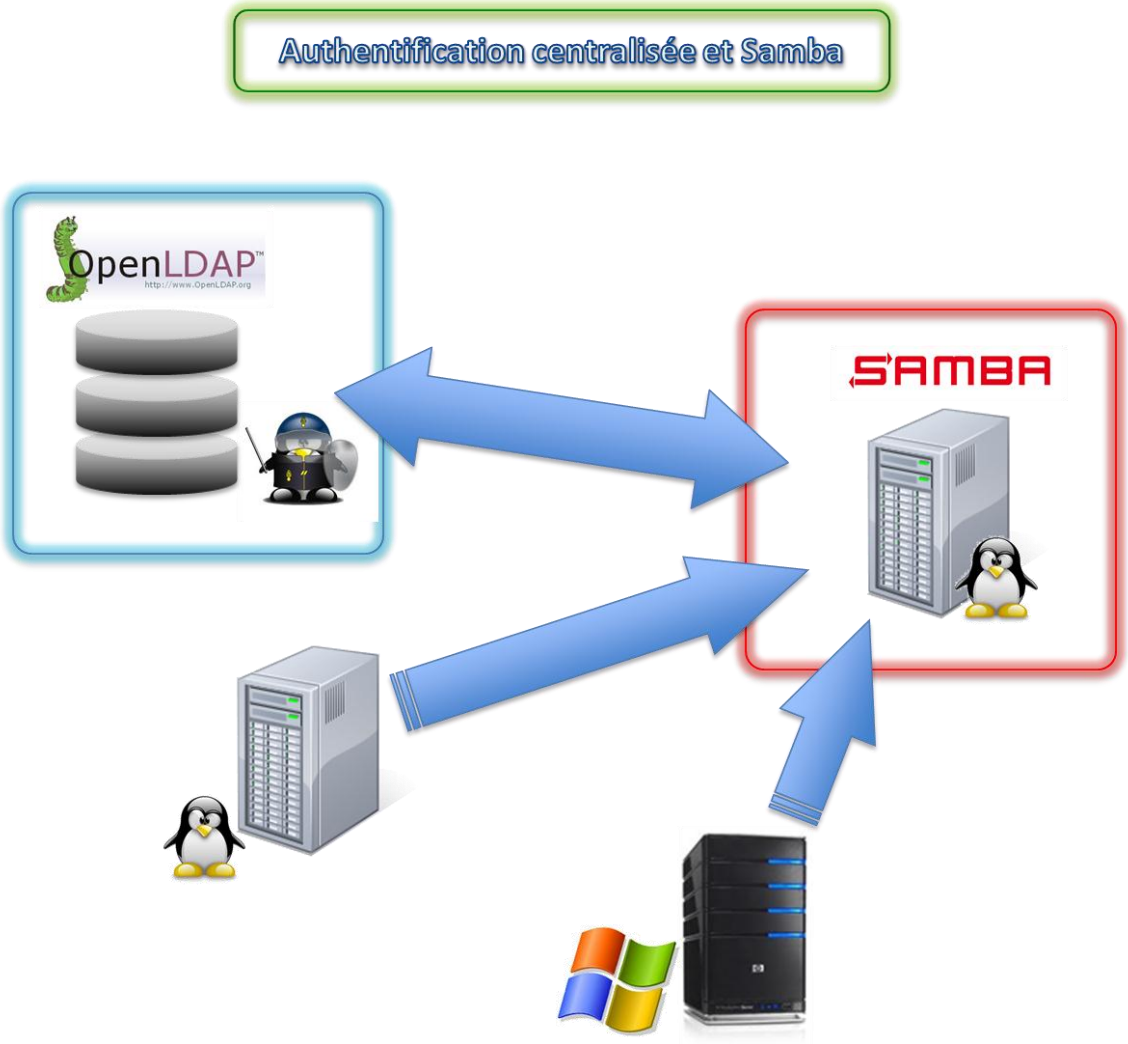
Rappel : toutes ces documentations, en HTML, sont intégrées dans le répertoire :

/usr/share/doc/Deployment_Guide-fr-FR-5.2/index.html

Installation et configuration d'un annuaire LDAP :
http://www.centos.org/docs/5/html/5.2/Deployment_Guide/s1-ldap-quickstart.html

Pour mettre en place une authentification PAM s'appuyant sur LDAP :
http://www.centos.org/docs/5/html/5.2/Deployment_Guide/s1-ldap-pam.html

Faire travailler SAMBA avec LDAP :
http://www.centos.org/docs/4/4.5/Reference_Guide/samba-PDC-LDAP.html


Authentification centralisée et Samba
OpenLDAP
SAMBA

Les services réseaux

Durant l'étude du SYSTEM V nous avons vu que tous les services réseaux peuvent être démarrés via la commande **« service ».** Dans ce chapitre cette commande sera constamment utilisée sur un terminal sécurisé (via OpenSSH) couplé à un traçage systématique du journal adéquat sur un autre terminal sécurisé.

La découverte de soucis éventuels en sera ainsi simplifiée.

OpenSSH : un vpn

Petit rappel historique : OpenSSH est issu du projet OpenBSD initié par ***Theo de Raadt*** qui est le système Unix le plus robuste en termes de sécurité de la planète.

OpenSSH est un ensemble d'outils (ssh, sftp, sshd, scp, ssh-keygen etc.) informatiques libres permettant des communications sécurisées sur un réseau informatique en utilisant le protocole SSH.
Il permet aussi le déport d'affichage sécurisé du protocole X-Windows (option -X). En fait, il est capable d'encapsuler des protocoles non sécurisés en faisant des redirections de ports.

Il peut rediriger tout le trafic d'une machine vers une autre en le cryptant : c'est un VPN.

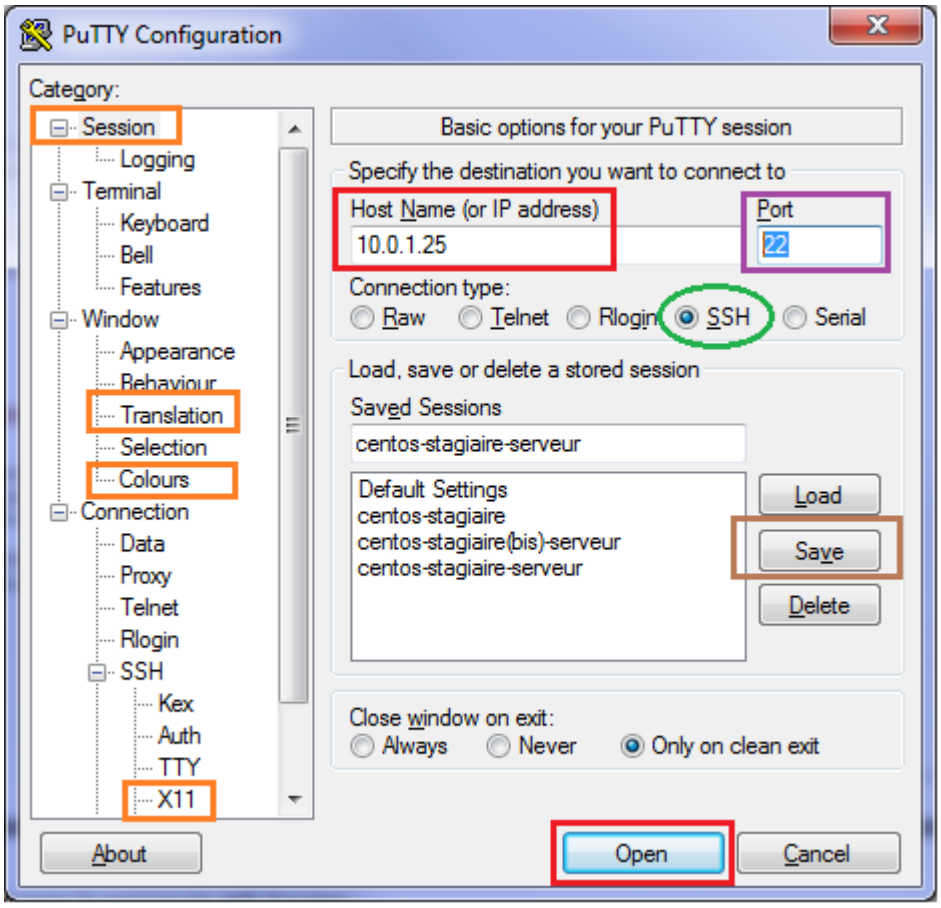
Par défaut OpenSSH client ou serveur est installé et activé par défaut sur le plupart des distributions GNU/Linux.
L'utilisation la plus commune reste l'accès distant sécurisé à une machine via le client « ssh », en vue d'administrer le serveur.
Comme tout service réseau régit par SYSTEM V, le serveur OpenSSH **se stoppe, s'arrête ou se redémarre avec la commande « service ».**

Ce service se configure avec le fichier **« etc/ssh/sshd_config »**.
Voici quelques options intéressantes :

Options	Signification
Port	Le numéro de port d'écoute du serveur OpenSSH
Protocol	Par défaut à 2, il permet de fixer la version de SSH utilisé
ListenAddress	Adresse IP sur laquelle écoute OpenSSH
PermitRootLogin	Définit si OpenSSH permet une connexion directe avec l'utilisateur « root ». Sur un serveur exposé il faut mettre à « no »
Banner	Permet d'afficher une bannière d'accueil à la connexion
ChrootDirectory	Permet de créer un environnement Chrooté (cloisonné)
AllowUsers/Groups	Utilisateurs/Groupes autorisés à se connecter
DenyUsers/Groups	Utilisateurs/Groupes non autorisés à se connecter
LogLevel	Très important pour débugué un souci de connexion
ServerKeysBits	Taille des clés de chiffrage publique, mini 512Bits (défaut 768)

Note : Dans le fichier de configuration les options commentées sont celles mise par défaut.
#PermitRootLogin yes, le comportement par défaut autorise une connexion avec « root »

Pour vous connecter à un serveur OpenSSH vous devez vous servir d'un client SSH comme PuTTY.



Avec OpenSSH, pour les besoins de la production, vous pouvez autoriser **les connexions automatiques** de certains utilisateurs (typiquement l'exploitation) **sans avoir besoin de saisir un mot de passe**. Le tout étant **sécurisé par clés privées/publiques**.
Aucune « passphrase » ne doit être saisie.
Du côté du serveur OpenSSH, la clé publique du client doit être placée dans un fichier contenant les clés autorisées à se connecter dans le compte de destination (**authorized_keys2**).

Pour la mise en œuvre faites attention aux droits sur les fichiers de clés et activez le mode debug du serveur (LogLevel). Le fichier de journaux de OpenSSH est **« /etc/var/log/secure »**, utilisez-le avec **« tail -f »**.

Pour aller plus loin :
http://www.centos.org/docs/5/html/5.2/Deployment_Guide/ch-openssh.html
http://wiki.centos.org/HowTos/Network/SecuringSSH

NFS

Le service NFS

Le partage de fichier **NFS** (*Network File System*) ou système de fichiers réseau permet de partager tout ou partie de son système de fichiers à destination de clients NFS, bien souvent d'autres Unix. Dans sa version de base c'est un système simple et efficace.
NFS s'appuie sur le **portmapper** (**portmap**), le support **nfs** du noyau et les services **rpc.nfsd** et **rpc.mountd**.

Les prérequis

Pour lancer le service NFS, portmap, nfslock et nfs doivent être lancés, sur une Centos **vous n'avez qu'à lancer le service « nfs »** avec le SYSTEM V est tout sera en place.
Vous pouvez vérifier le statut comme suit :

```
root@centos-stagiaire-serveur:~
[root@centos-stagiaire-serveur ~]# service portmap status
portmap (pid 1679) en cours d'exécution...
[root@centos-stagiaire-serveur ~]# service nfslock status
rpc.statd (pid  1711) en cours d'exécution...
[root@centos-stagiaire-serveur ~]# service nfs status
rpc.mountd (pid 3384) en cours d'exécution...
nfsd (pid 3381 3380 3379 3378 3377 3376 3375 3374) en cours d'exécution...
rpc.rquotad (pid 3369) en cours d'exécution...
[root@centos-stagiaire-serveur ~]#
```

Monter un **partage côté serveur** reste assez simple. Avec NFS on parle d'export.

Le partage NFS se paramètre dans le fichier **« /etc/exports »**. Chaque ligne est composée de deux parties :

- Le chemin du répertoire partagé (exporté),
- Les autorisations et modes d'accès.

La première ne requiert pas d'attention particulière, hormis le fait que vous devez veiller à la mise en place des droits POSIX adéquates pour le partage, au niveau de l'arborescence serveur.

Les autorisations d'accès sont composées de paires hôtes/permissions selon le format suivant :
host(permissions_mode_accès)

Si l'hôte n'est pas défini, c'est tout le réseau (portée dite mondiale) qui sera concerné par les permissions. Si les permissions ne sont pas définies, l'export sera en lecture seule. Il ne faut surtout pas mettre d'espaces entre l'hôte et les permissions.

L'hôte peut être :

- un nom d'hôte unique,
- un domaine,
- un réseau ou un sous-réseau,
- une combinaison de l'ensemble, avec des caractères de substitution (*, ?).

Les permissions peuvent être :

- **ro** : lecture seule,
- **rw** : lecture écriture,
- **no_root_squash** : le root distant équivaut au root local,
- **root_squash** : si root se connecte au partage, son uid sera remplacé par celui d'un utilisateur anonyme. Ainsi il n'y a pas de risques que l'utilisateur root d'un poste local puisse être root sur un partage distant ;
- **all_squash** : étend la règle précédente à tous les utilisateurs,
- **anonuid / anongid** : uid et gid pour l'utilisateur anonyme.

Une fois ce **fichier** correctement **paramétré** vous pouvez **lancer le service « nfs »,** si ce n'est déjà fait, ou bien **exécuter « exportfs -a »** pour **valider les changements opérés dans « /etc/exports ».**

Attention **NFS travaille**, pour les droits d'accès, avec **les UID et GID, et non pas les noms affichés**. Par conséquent les tests de mise en œuvre avec l'accès « root » sont ici triviaux, l'UID est « 0 ».

Avec une flopée d'utilisateurs les UID stockés sur toutes vos machines clientes auront intérêts à être identiques sur les serveurs NFS. Il peut donc y avoir un lourd travail d'**homogénéisation et de normalisation des UID/GID à mener sur tout votre parc**.
Il est donc intéressant d'avoir recours à NIS ou LDAP si vous envisagez de travailler massivement avec NFS.

La commande « **exportfs »** permet de contrôler les partages NFS.

- **exportfs -r** : rafraîchit la liste des partages après modification de /etc/exports,
- **exportfs -v** : liste des partages,
- **exportfs -a** : exporte (ou recharge) tous les partages de « **/etc/exports »** ou un partage donné,
- **exportfs -u** : stoppe le partage donné. -a pour tous.

La commande **showmount** montre les partages d'un hôte donné.
« showmount -e host »

Pour importer un partage NFS, du côté du client, vous devez éditez votre fichier de définition des pointd emontage : **« /etc/fstab »** comme suit

```
serveur_nfs:/chemin_partage_distant        /point_montage_local    nfs     default        0    0
```

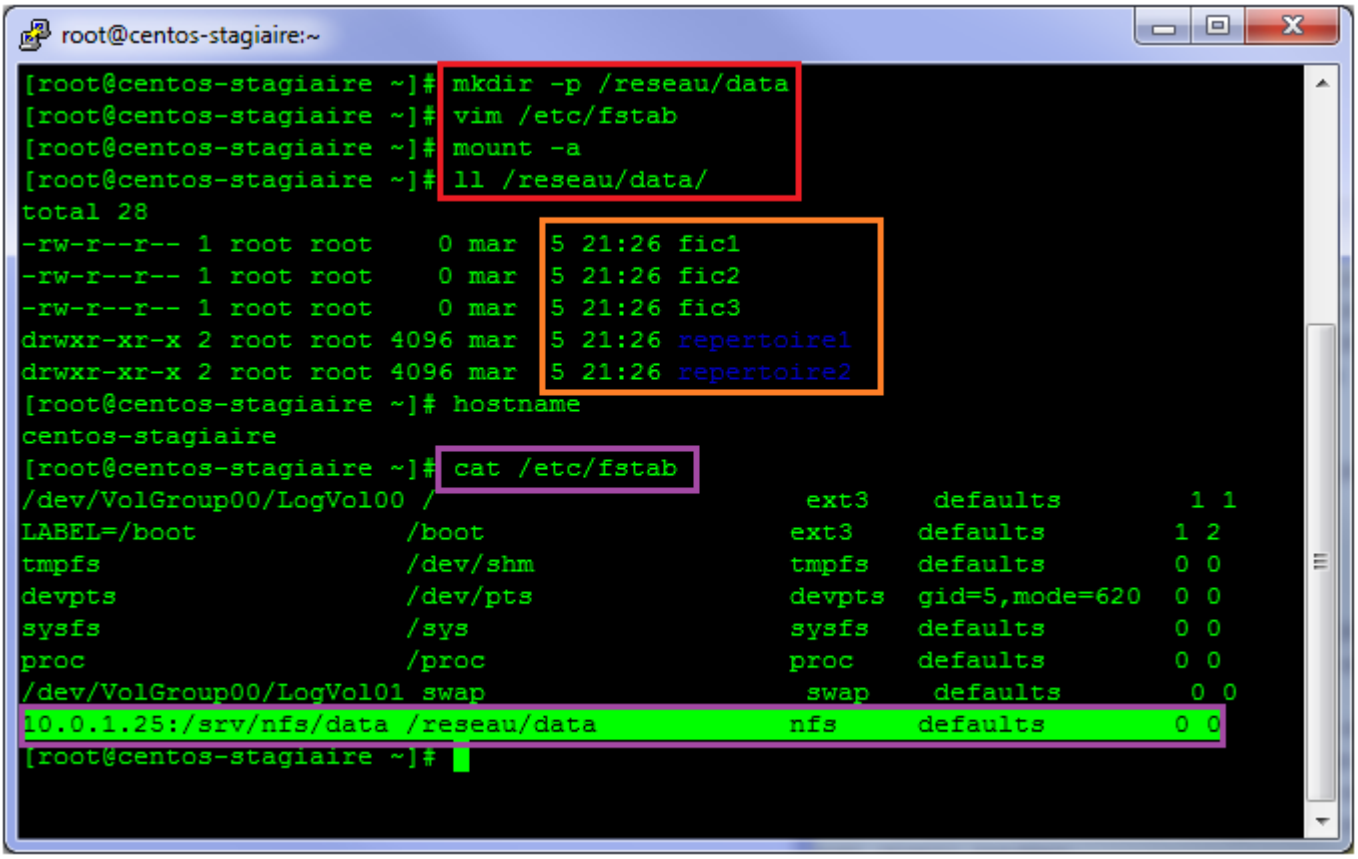
Astuce : Si souci de permission d'accès côté client faire un **« exportfs -v »** sur le serveur pour voir les paramètres d'exports effectifs (erreur style : **root_squash**).

Mise en œuvre client/serveur

Côté serveur NFS :

```
root@centos-stagiaire-serveur:~
[root@centos-stagiaire-serveur ~]# mkdir -p /srv/nfs/data
[root@centos-stagiaire-serveur ~]# vim /etc/exports
[root@centos-stagiaire-serveur ~]# cat /etc/exports
# Serveur NFS partage des données
/srv/nfs/data   *(rw,no_root_squash)

[root@centos-stagiaire-serveur ~]# service nfs start
Démarrage des services NFS :                               [  OK  ]
Démarrage du quota NFS :                                   [  OK  ]
Démarrage du démon NFS :                                   [  OK  ]
Démarrage de NFS mountd :                                  [  OK  ]
[root@centos-stagiaire-serveur ~]# tail -f /var/log/messages
Mar  5 21:14:37 centos-stagiaire-serveur avahi-daemon[2082]: Invalid legacy unic
ast query packet.
Mar  5 21:14:37 centos-stagiaire-serveur avahi-daemon[2082]: Received response f
rom host 10.0.1.10 with invalid source port 58962 on interface 'eth0.0'
Mar  5 21:14:40 centos-stagiaire-serveur last message repeated 5 times
Mar  5 21:17:10 centos-stagiaire-serveur avahi-daemon[2082]: Invalid legacy unic
ast query packet.
Mar  5 21:17:10 centos-stagiaire-serveur avahi-daemon[2082]: Received response f
rom host 10.0.1.10 with invalid source port 58962 on interface 'eth0.0'
Mar  5 21:17:11 centos-stagiaire-serveur avahi-daemon[2082]: Invalid legacy unic
ast query packet.
Mar  5 21:17:11 centos-stagiaire-serveur avahi-daemon[2082]: Invalid legacy unic
ast query packet.
Mar  5 21:17:11 centos-stagiaire-serveur avahi-daemon[2082]: Received response f
rom host 10.0.1.10 with invalid source port 58962 on interface 'eth0.0'
Mar  5 21:17:15 centos-stagiaire-serveur last message repeated 4 times
Mar  5 21:25:37 centos-stagiaire-serveur mountd[3384]: authenticated mount reque
st from 10.0.1.26:948 for /srv/nfs/data (/srv/nfs/data)

[root@centos-stagiaire-serveur ~]# hostname
centos-stagiaire-serveur.domain.local
[root@centos-stagiaire-serveur ~]#
```

Côté client NFS, méthode de montage automatisé par fichier **« /etc/fstab »** :



NFS (sur Centos 5.5) supporte les ACL étendues.

Pour aller plus loin avec NFS :

Guide avancé du déploiement de NFS
http://www.centos.org/docs/5/html/Deployment_Guide-en-US/ch-nfs.html

Alternative au montage via « /etc/fstab » : l'automonteur(**autofs)**
http://www.centos.org/docs/5/html/5.2/Deployment_Guide/s1-nfs-client-config-autofs.html

SAMBA

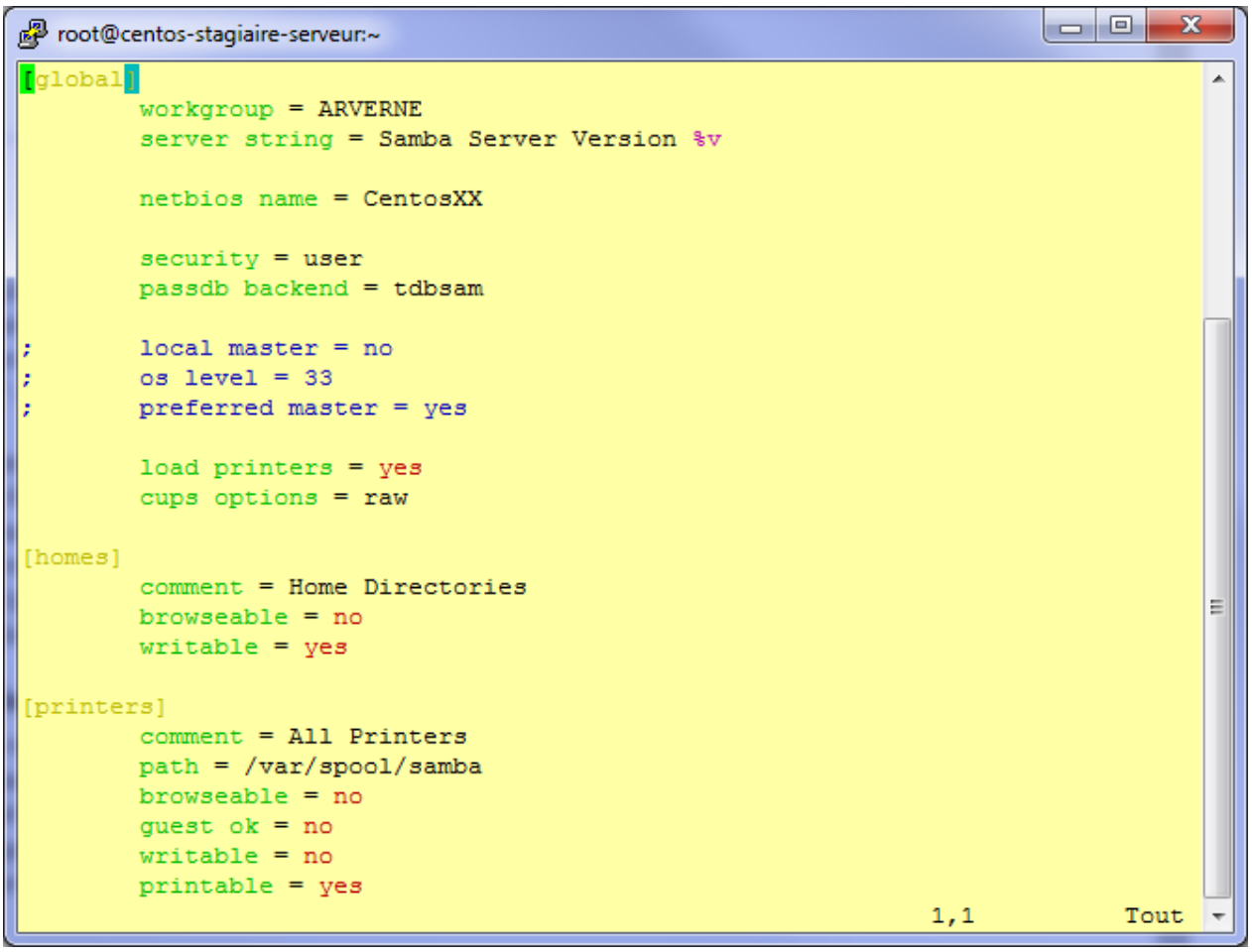
Samba est un logiciel libre et une mise en œuvre du protocole SMB/CIFS sous GNU/Linux, initialement développée par l'australien Andrew Tridgell. Il est sous licence GNU GPL 3[1]. Son nom provient du protocole SMB (Server message block), le nom du protocole standard de Microsoft, auquel ont été ajoutées les deux voyelles *a* : « SaMBa ».

À partir de la version 3, Samba fournit des fichiers et services d'impression pour divers clients Windows et peut s'intégrer à un domaine Windows Server, soit en tant que contrôleur de domaine principal (PDC) ou en tant que membre d'un domaine. Il peut également faire partie d'un domaine Active Directory. Il fonctionne sur la plupart des systèmes Unix, comme GNU/Linux, Solaris, AIX et les variantes BSD, y compris Apple, Mac OS X Server (qui a été ajoutée au client Mac OS X en version 10.2). Samba fait partie intégrante de presque toutes les distributions GNU/Linux. (source WikiPedia)

Configuration de base

Le service SAMBA se configure dans le fichier **« /etc/samba.smb.conf »**.
Beaucoup de personnes ont souvent l'impression que ce fichier est complexe et énorme : en gros que configurer Samba est complexe. C'est faux, le **fichier non modifié comporte 90% de commentaires** qui sont autant d'options disponibles. Expurgeons ces commentaires et voyons ci-dessous le fichier de configuration (faites avant tout une copie de ce fichier, les commentaires seront utiles) :

Voici les 19 lignes de bases, sans commentaires :



Par défaut trois sections sont présentes :

- **[global] :** réglages génériques et globaux du serveur, nom, commentaires, méthode d'authentification, réglages par défaut, etc.
- **[homes] :** partage des répertoires personnels des utilisateurs (le partage de fichiers par défaut).
- **[printers] :** partage des imprimantes.

Les paramètres sont de la forme :

nom = valeur

Faites attention à la directive **« os level » dans la section [global]** vous pouvez provoquer des soucis de maîtres d'élections sur les contrôleurs de domaines Windows PDC, BDC.

Sachez que **le fichier « smb.conf »** contient la plupart des options de configurations avec en sus les explications pour les exploiter. Cerise sur le gâteau il y a même des exemples pour la plupart des fonctionnalités : **vous n'aurez qu'à dé commenter ces exemples** !
Note : Les commentaires commencent par un point-virgule **« ; »** ou un dièse **« # »**.

En plus du partage par défaut [homes] vous pouvez ajouter de nouveaux partages Samba. Pour ce faire, ce **partage doit disposer de sa propre section** et d'une série d'options d'accès et de présentation.
Exemple :

```
[Partage_stagiaireXX]
comment = Répertoire de partage du Stagiaire XX
path = /srv/samba/partage_stagaireXX
browseable = yes
public = no
writable = yes
printable = no
group = partage_smb
```

Quelques une des options :

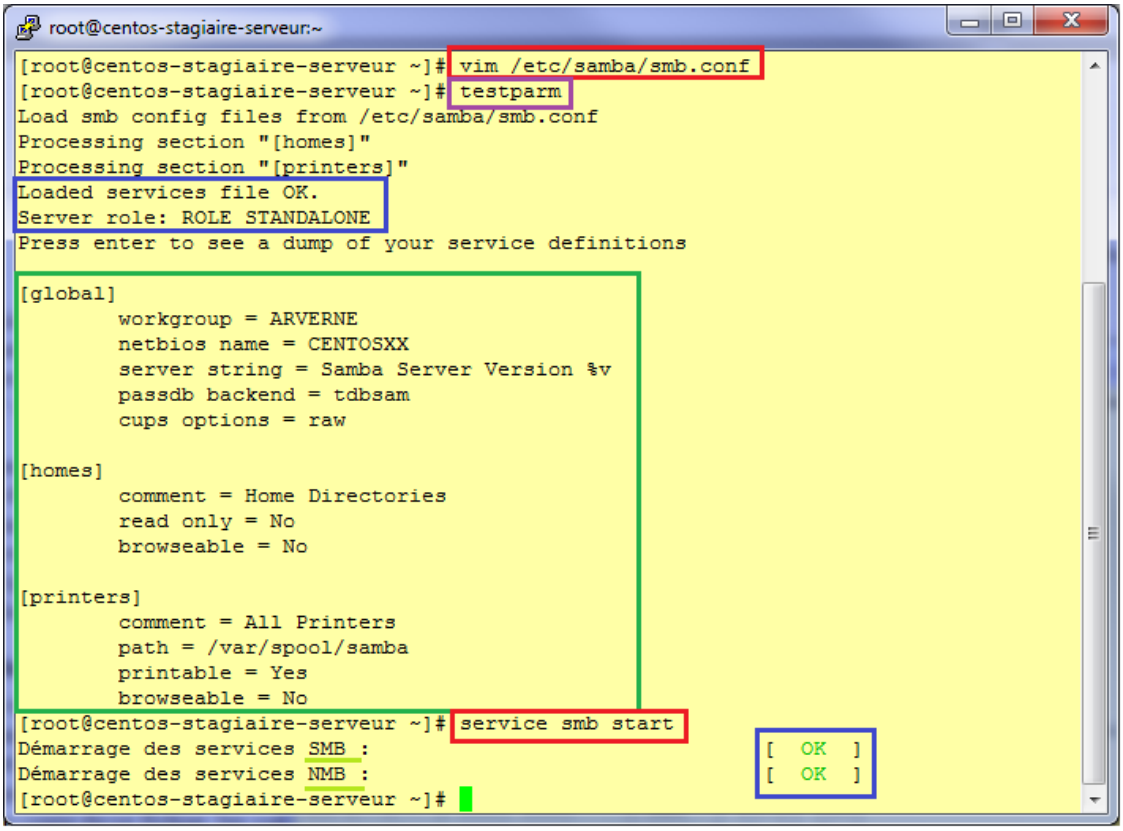
Options	Signification
comment	Description du partage qui sera visible par le réseau
path	Chemin du répertoire local que vous désirez partager
public	Le partage sera accessible à l'utilisateur par défaut « **»guest »** (invité)
browseable	Le partage apparaîtra dans le « Voisinage réseau »
writable	Le partage est accessible en lecture et écriture
printable	Le partage est une imprimante
group	Nom du groupe par défaut pour la connexion à ce partage
valid users	Nom des utilisateurs autorisés à accéder à ce partage
read only	Le partage est en lecture seule pour tout le monde
guest ok	Aucun mot de passe n'est nécessaire pour accéder au partage. Ce sera le compte « invité qui sera utilisé dans ce cas
guest only	Le partage est accessible uniquement aux invités
create mask	Permet de définir avec quels droits POSIX les fichiers seront créer sur ce partage

Tout comme les partages, l'ajout d'imprimantes se fait en créant une section du nom de votre imprimante **[Nom_imprimante]** telle quelle sera identifiée sur le réseau.
Exemple :

```
[HP3210_Bat01]
comment = HP Photosmart 3210 tout-en-un
printer = hp3210
valid users = exploit
path = /var/spool/hp3210
printable = yes
browseable = yes
public = no
writable = no
```

Options	Signification
comment	Description de l'imprimante qui sera visible par le réseau
path	Chemin du spool d'impression de cette impression
printer	Nom de l'imprimante sous GNU/Linux
valid users	Nom des utilisateurs autorisés à accéder à cette imprimante (@ devant un nom indique un groupe)

Dès à présent vous pouvez vérifier (**tesparm**) votre fichier de configuration et lancer le service « samba ».



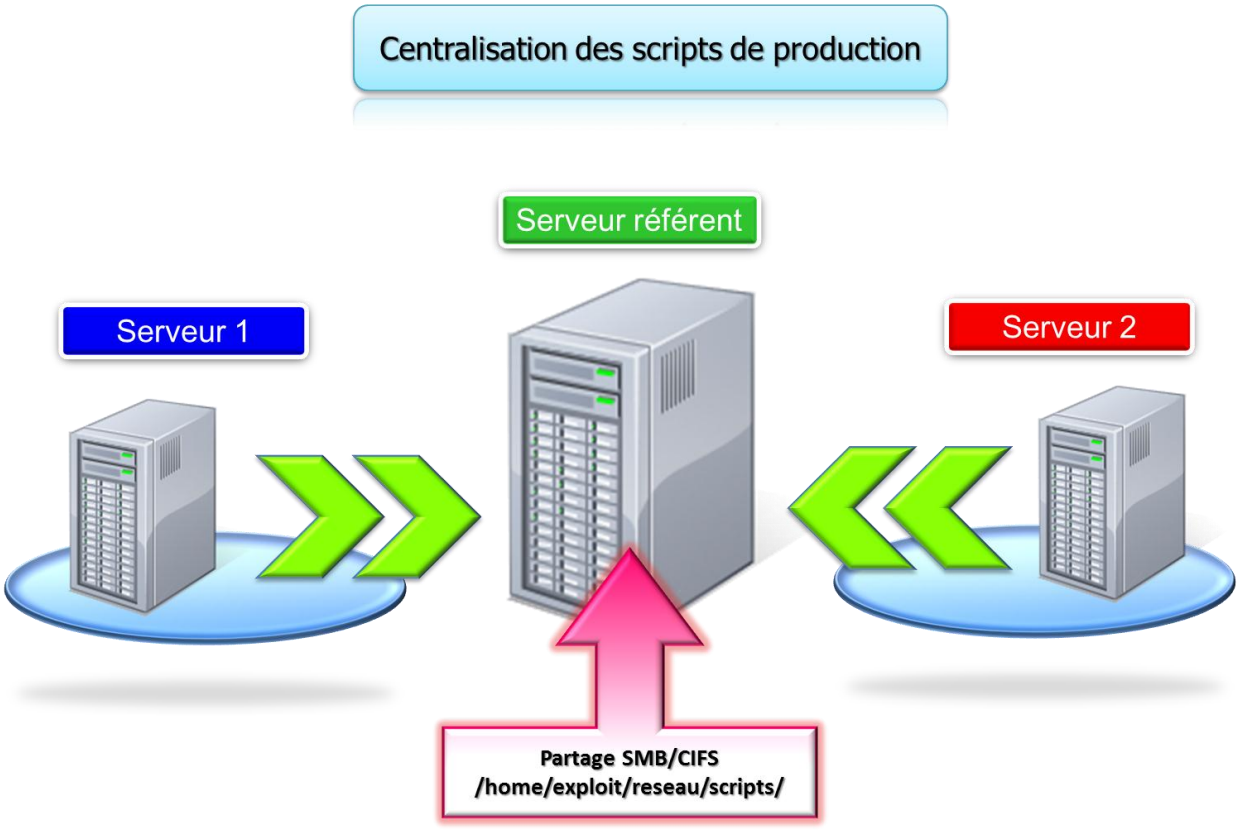
Lors du démarrage du **service « samba »** vous vous apercevez que deux serveurs sont lancés :

- **smbd** : le serveur SMB/CIFS qui authentifie et partage fichiers et imprimantes,
- **nmbd** : le serveur de nom NetBIOS qui permet le parcours des ressources et propose WINS.

Un troisième service non visible sur cette capture écran, **« winbindd »**, permet d'utiliser les comptes utilisateur d'un domaine Microsoft. Les dernières versions de Samba (3 et suivantes) permettent également de se **raccorder à Active Directory.**

Nous allons rajouter un partage supplémentaire.

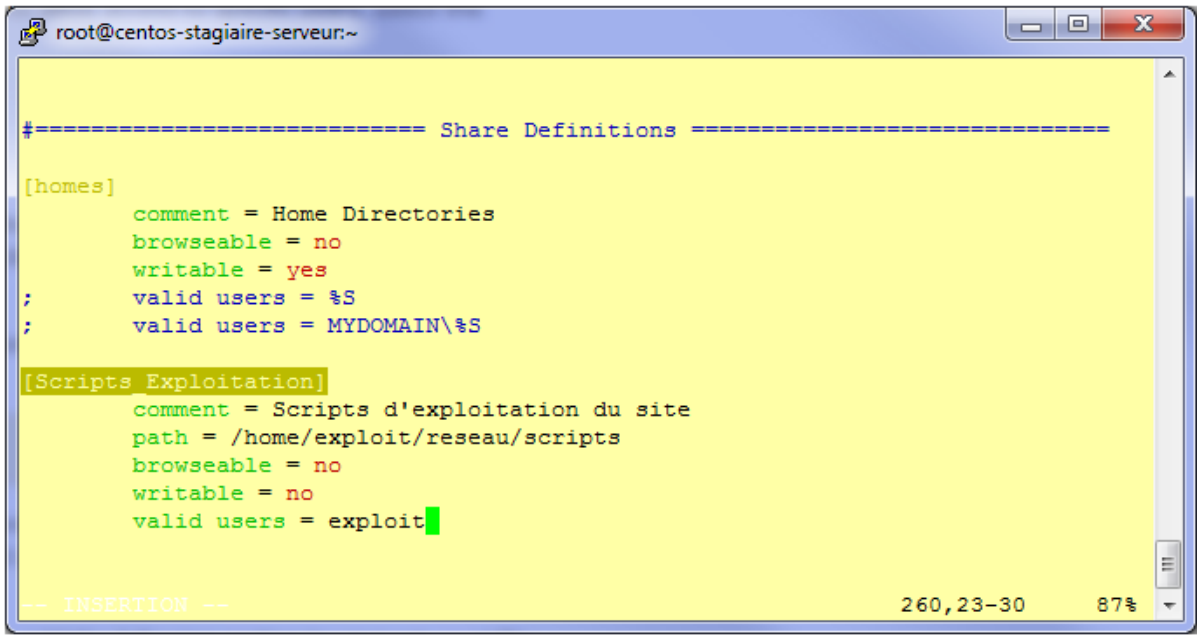
Un peu d'architecture système :



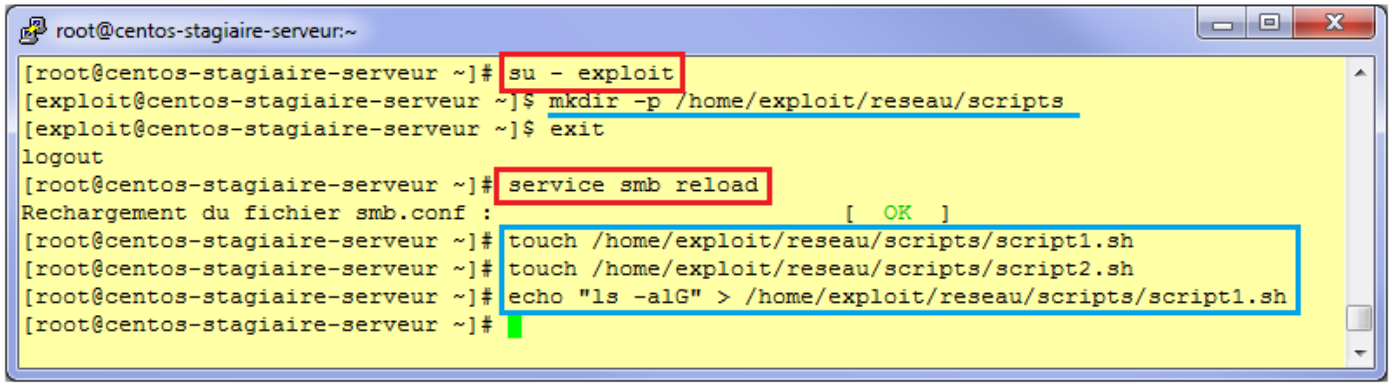
Proposons à l'utilisateur de l'exploitation informatique (**« exploit »)** un partage dédié aux scripts d'exploitation (sauvegarde de base, tâches quotidiennes, arrêt/démarrage pour maintenance etc.).
Ces derniers seront accessibles pour chaque serveur du site en lecture seul via en montage automatique (/etc/fstab ou automountage)
Seul l'administrateur du site pourra modifier les scripts sur un des serveurs référents qui héberge physiquement les scripts

Voyons comment mettre en place ce partage.

Ajoutons le partage :



Puis créons le répertoire **« /home/exploit/reseau/scripts »** qui hébergera les scripts (on crée dans ce répertoire un **script1.sh** de test) qui seront communs à tous vos serveurs de production (via un montage réseau SMB/CIFS).
Enfin on met en production notre nouveau partage **[Scripts_Exploitation]** avec le rechargement du fichier de configuration via **« service smb reload »** :



Note : Si vous êtes sur un serveur de production (90% de cas) il ne faut pas redémarrer le service avec « **restart** », vous risquez de déconnecter des utilisateurs, vous devez juste rafraichir la configuration de **« smbd »**.
Pour ce faire :

- [root@centos-stagiaire-serveur ~]# service smb
- Syntaxe : /etc/init.d/smb *{start|stop|restart|reload|status|condrestart}*
- [root@centos-stagiaire-serveur ~]# **service smb reload**

Note: Ce principe s'applique à d'autres services réseaux.

Gestion des comptes

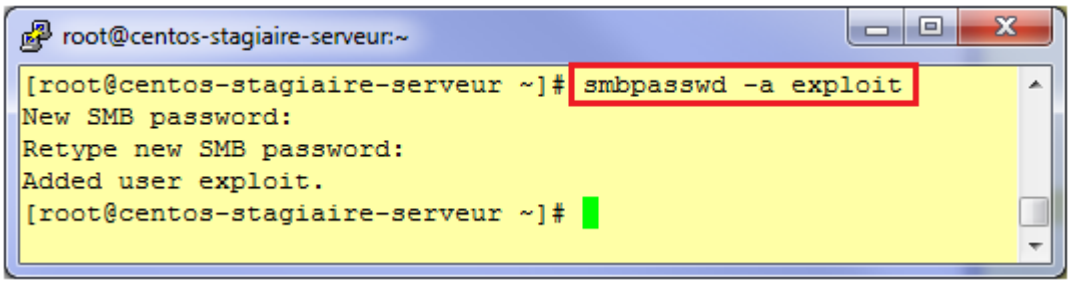
A ce stade vous avez remarqué que malgré la configuration précédente il vous est impossible d'accéder aux partages du serveur via l'explorateur Windows ou d'un montage sous GNU/Linux : quelques explications s'imposent.

Samba propose plusieurs méthodes d'authentification définies dans la section [global] :

- **user** : **méthode par défaut**; l'accès à l'ensemble des partages d'un serveur se fait par la validation d'un nom d'utilisateur et d'un mot de passe uniques.
- **share** : méthode de validation des identifiants partage par partage. Dans ce cas, tous les accès aux partages, même publics, nécessitent des identifiants.
- **domain** : utilisation d'un groupe de travail avec authentification.
- **ads** : utilisation d'Active Directory.

D'autres types d'authentification sont possibles comme un couplage à un annuaire LDAP (OpenLDAP). Ici nous allons utiliser la première méthode de fonctionnement de Samba, les autres nécessitant un ouvrage complet pour être correctement mise en œuvre et en expliquer le fonctionnement précis.

Vous devez créer un utilisateur pour que le **service « samba » en prenne connaissance**. Il s'agit ni plus ni moins d'un **mappage** entre **/etc/passwd** et **Samba**. Par conséquent **l'utilisateur doit aussi exister sous « /etc/passwd », donc sous GNU/Linux**.
Une fois votre compte créé sous /etc/passwd (**adduser <utilisateur>**) vous devez utiliser la commande qui permet d'ajouter un utilisateur Samba : **« smbpasswd »**.



Lors de l'invocation de cette commande une entrée est ajoutée dans le fichier crypté suivant :

- **« /etc/samba/passdb.tdb ».**

Ce fichier est votre base SAM local (stockage des comptes locaux Windows). Pour en voir le contenu utilisez **« pdbedit -L »**.

Voici quelques options de **« smbpasswd »** pour manipuler vos utilisateurs Samba :

Options	Signification
-a	Ajouter et définir le mot de passe d'un utilisateur Linux existant dans Samba.
-x	Cette option indique que le nom d'utilisateur suivant doit être supprimé du fichier local
-d	Cette option indique que le nom d'utilisateur suivant doit être désactivé dans le fichier local
-e	Cette option indique que le nom d'utilisateur suivant doit être activé dans le fichier local

Accéder aux partages

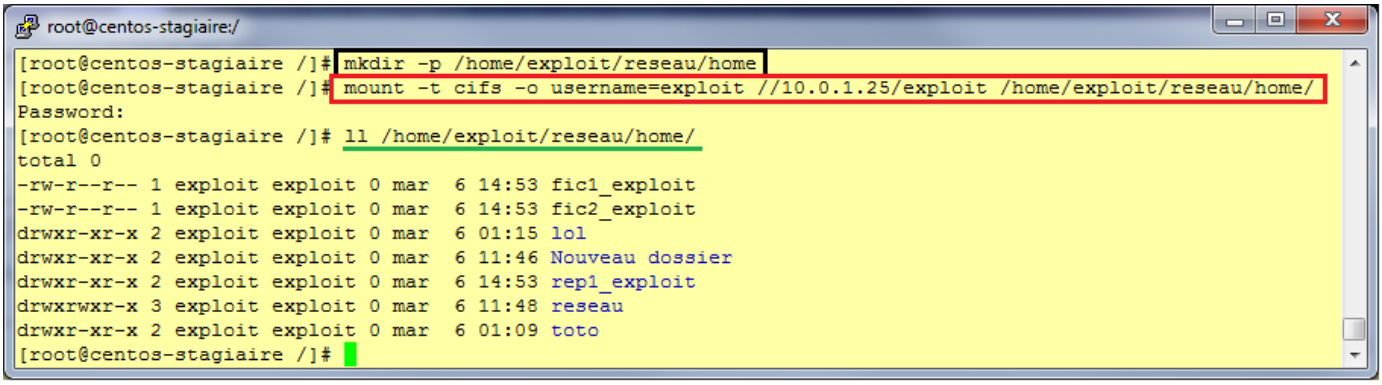
Accéder à un partage Samba sous **GNU/Linux**.

Il existe 4 manières :

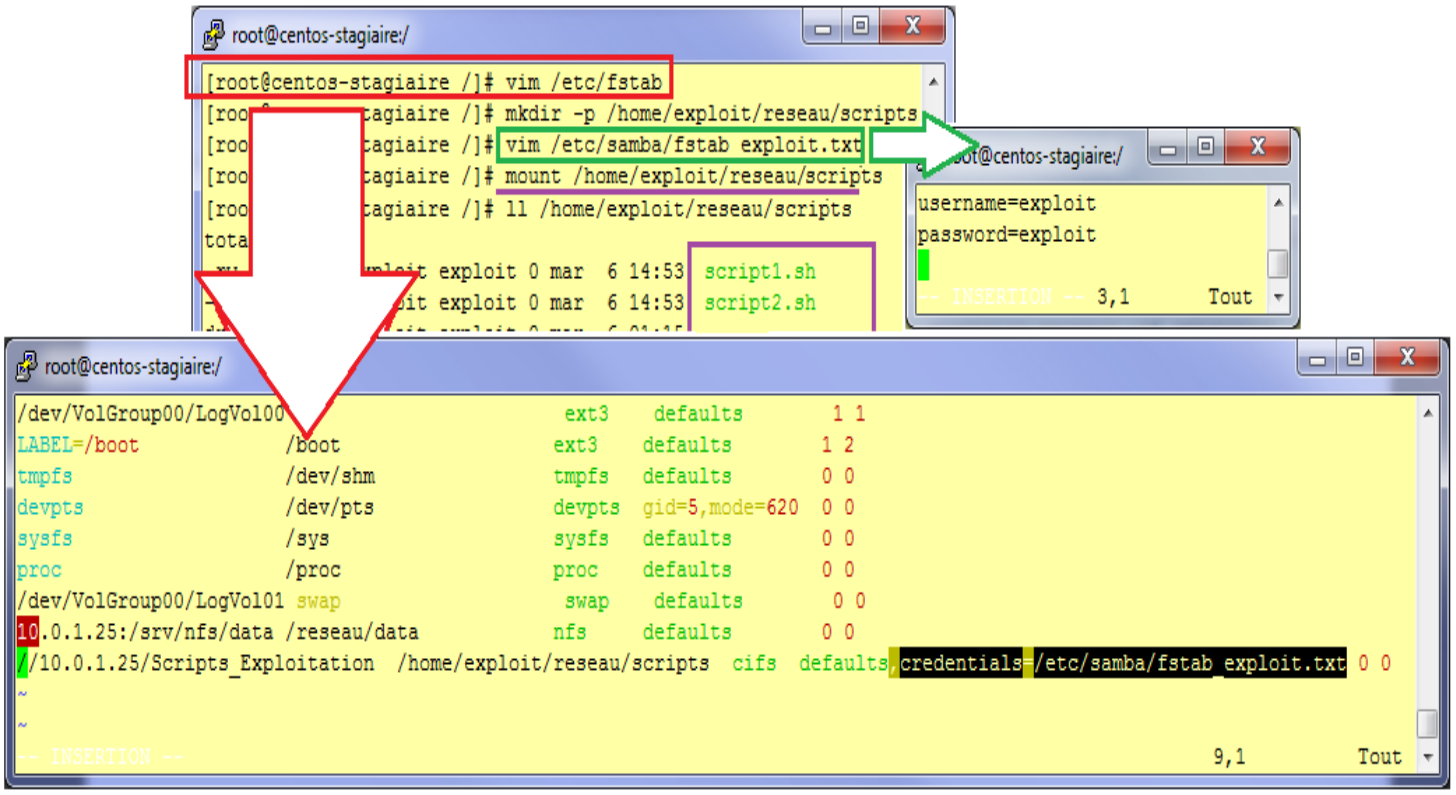
- mount,
- smbclient,
- /etc/fstab,
- automounter.

Pour ces exemples 10.0.1.25 étant notre serveur référent.
Se connecter à un partage d'un serveur avec la commande « **mount** » depuis un autre serveur (10.0.1.26) **:**

Note : n'utilisez plus le type de FS **« smb »** il n'est plus à jour depuis longtemps, prenez bien **« cifs ».**



Montage automatique depuis un autre serveur (10.0.1.26) via **« /etc/fstab »**, avec en sus une protection du mot de passe dans fichier seulement accessible par « root » (**chmod 400 /etc/samba/fstab_exploit.txt**):



Se connecter à un partage d'un serveur avec **« smblicent »** et obtenir une invite de commande depuis un autre serveur (10.0.1.26) :

- smbclient //<serveur>/partage -U <utilisateur_smb>

Pour mettre en œuvre **Samba avec automounter** :
http://wiki.centos.org/TipsAndTricks/WindowsShares

Visualiser les partages d'un serveur (-N : pour l'utilisateur invité) :

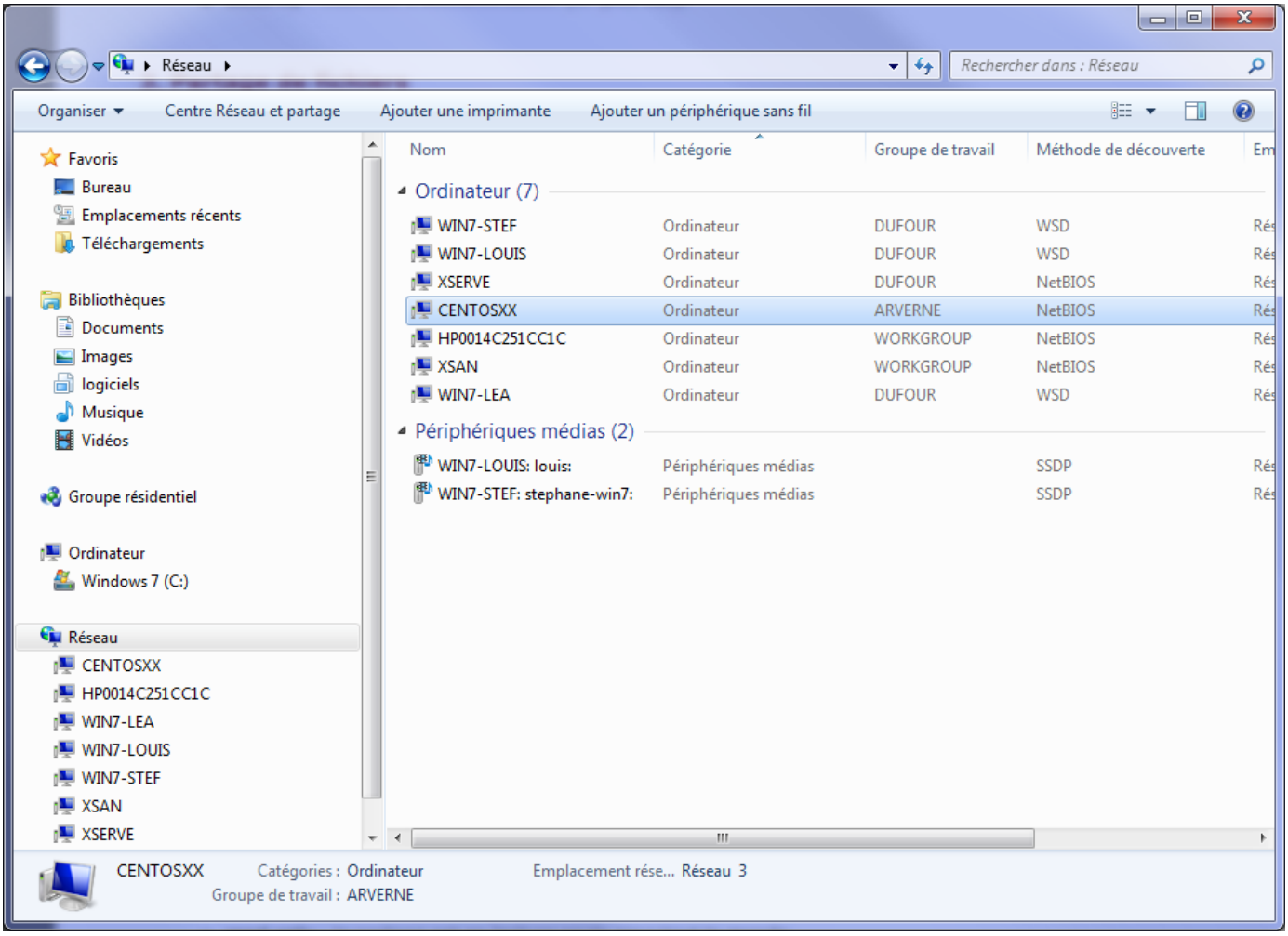
- smbclient -L <serveur> -U <utilisateur_smb>

Sur un serveur ayant le service Samba lancé, vous pouvez obtenir des informations sur le service, les utilisateurs connectés, les fichiers ouverts etc.:

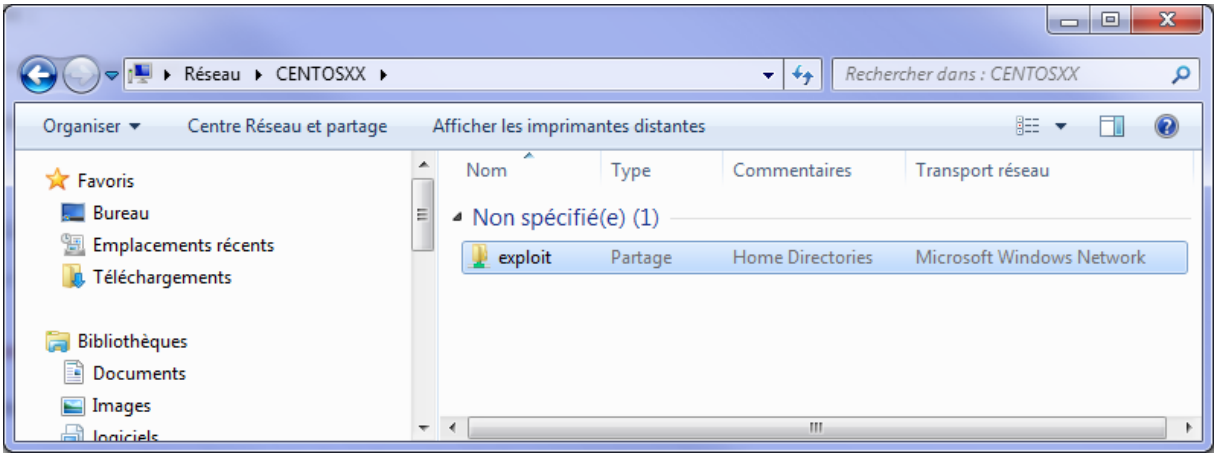
- smbstatus -d

Accès aux partages Samba à partir de **Windows** :

Ouvrez l'explorateur Windows, cliquez sur le serveur **CENTOSXX**.

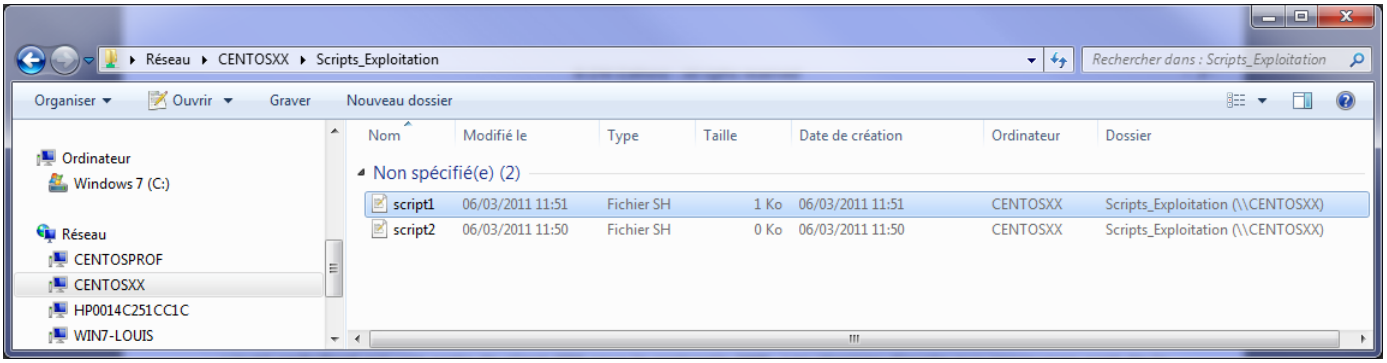


Cliquez sur votre partage Samba personnel (Home Directories) : **« exploit »**,
Puis à l'invite saisissez le **nom d'utilisateur/mot de passe** (mentionné avec « **smbpasswd -a** ») :

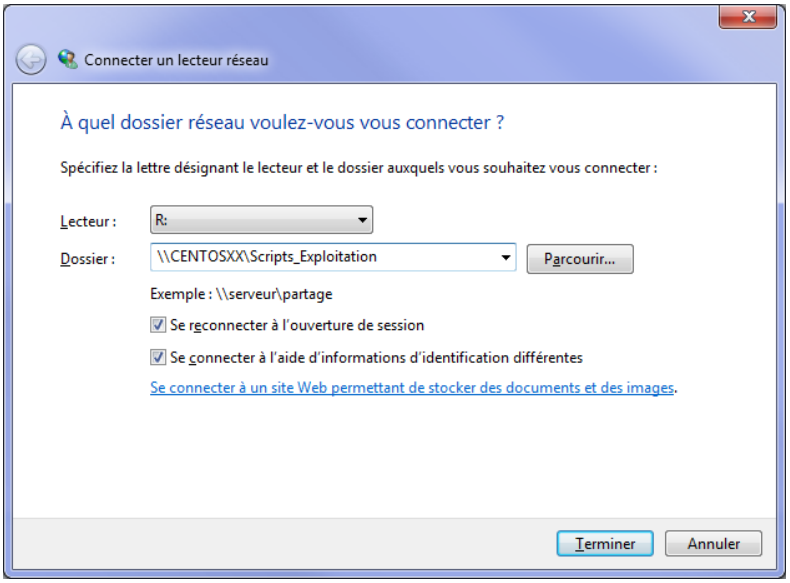


Vous constatez toutefois que le partage des scripts de l'exploitation n'est pas visible. Cela est normal, car il n'est pas **« browseable ».** Dans la barre d'adresse de l'explorateur saisissez :

« \\CENTOSXX\Scripts_Exploitation »



Vous pouvez également parvenir au même résultat avec la **« connexion à un lecteur réseau »** :



Pour plus de détails :
http://www.centos.org/docs/5/html/Deployment_Guide-en-US/s1-samba-configuring.html
(Préférez la partie « 19.4.2. Command Line Configuration », le mode graphique ne supportant pas forcément toutes les options de Samba)

Pour débuguer les accès etc., toujours la même méthode : « **tail -f /var/log/samba/smbd.log »**

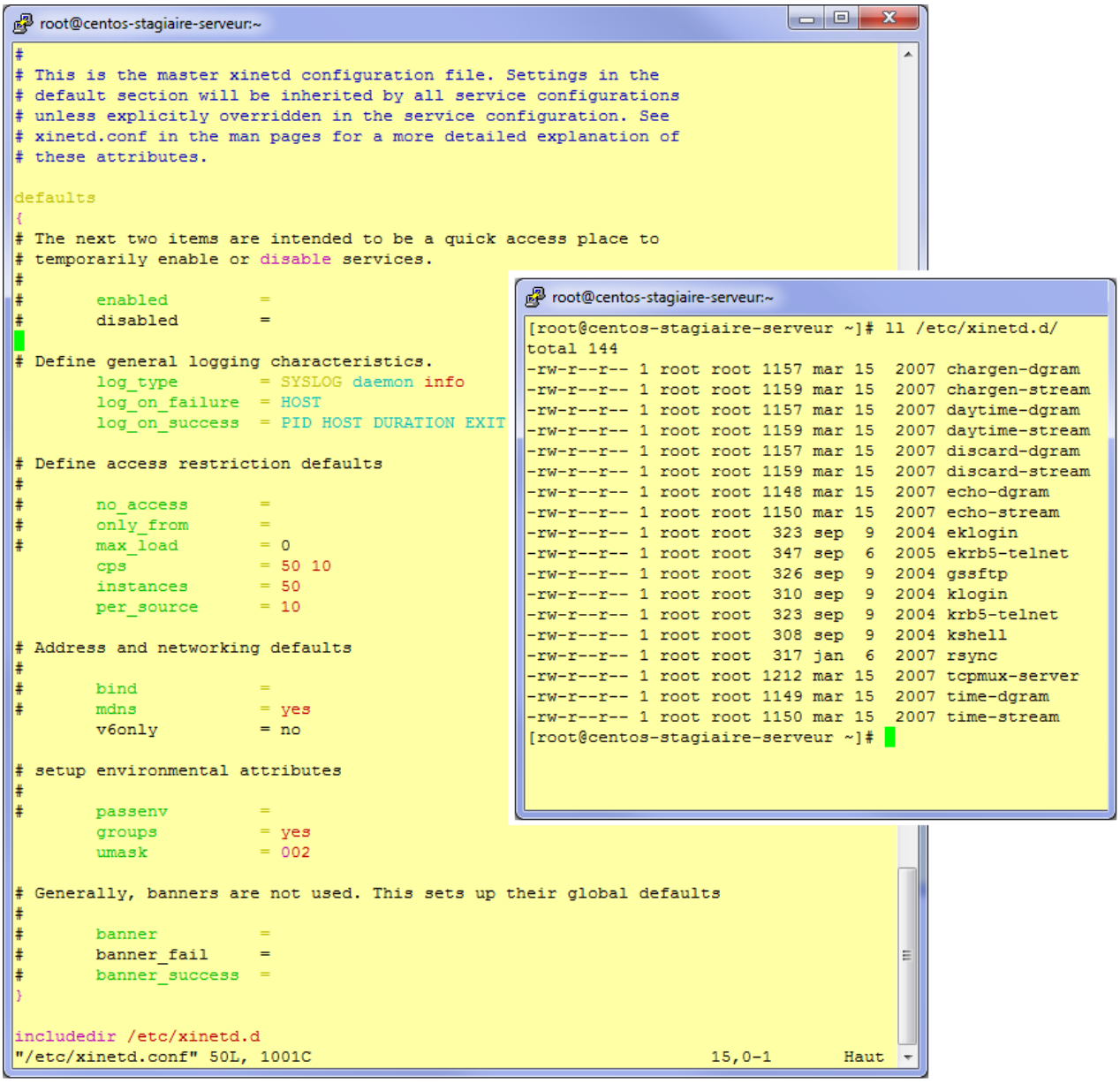
Aller plus loin, intégration LDAP avec SAMBA : le duo gagnant.

XINETD

Sur un serveur les services sont démarrés ou non. Pour service démarré, même si le service est très peu utilisé il utilise un espace en mémoire etc.
Le super serveur « **xinetd** » a été créé dans l'optique de **faire du service à la demande (Service On Demand)** :

- Si **personne n'utilise** un service réseau donné alors « **xinetd** » **stoppe ce service**.
- Si un **utilisateur distant désire utiliser** un service donné alors « **xinetd** » **démarre ce service**.

« xinetd » fonctionne en permanence et surveille tous les ports des services qu'il gère. Lors de la réception d'une requête de connexion à l'un des services qu'il gère, **« xinetd »** démarre le serveur adapté pour ce service.
Voici les fichiers de configuration de xinetd :



Description :

- **/etc/xinetd.conf** : configuration globale
- **/etc/xinetd.d/*** : répertoire contenant les fichiers spécifiques aux services. Il peut exister un fichier par service, du même nom que celui précisé dans **/etc/services**.

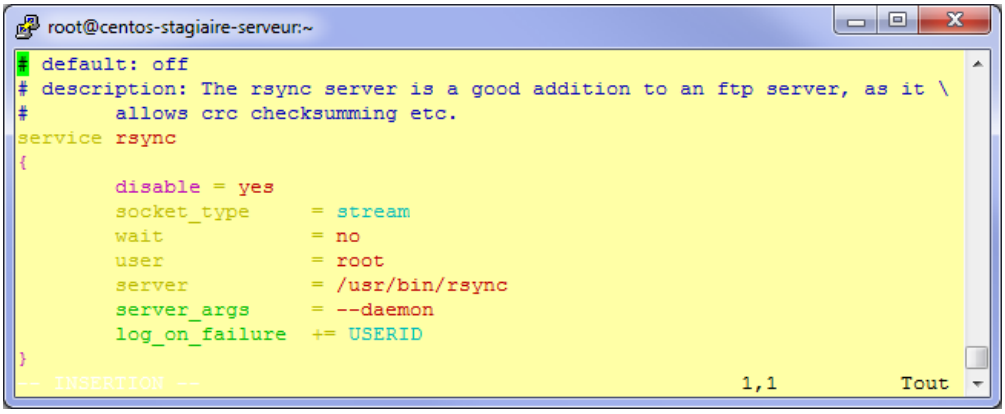
Voyons ce qui se trouve dans le fichier de configuration principale :

- **instances** : nombre maximal de requêtes qu'un service xinetd peut gérer à un instant donné.
- **log_type** : dans notre cas, les traces sont gérées par le démon **syslog** via **authpriv** et les traces sont placées dans /var/log/secure. **FILE /var/log/xinetd** aurait placé les traces dans /var/log/xinetd.
- **log_on_success** : xinetd va journaliser l'événement si la connexion au service réussit. Les informations tracées sont l'hôte (**HOST**) et le **PID** du processus serveur traitant la connexion.
- **log_on_failure** : idem mais pour les échecs. Il devient simple de savoir quels hôtes ont tenté de se connecter si par exemple la connexion n'est pas autorisée.
- **cps** : xinetd n'autorise que 25 connexions par secondes à un service. Si la limite est atteinte, xinetd attendra 30 secondes avant d'autoriser à nouveau les connexions.
- **includedir** : inclut les options des fichiers présents dans le répertoire indiqué.

Comme vous pouvez le constater sur une Centos, dans le répertoire **« /etc/xinetd.d/ »,** il est très peu fait appel à **« xinetd » par défaut**.
« xinetd » s'arrête ou se démarre comme n'importe quel autre service de SYSTEM V, avec la commande suivante :

- **service** xinetd start|stop|restart

Voici le cas de « rsync » qui par défaut est désactivé :



Vous pouvez consultez les options de xinetd ici :
http://www.centos.org/docs/5/html/Deployment_Guide-en-US/ch-tcpwrappers.html

FTP

Le serveur **FTP** (*File Transfer Protocol*) le plus courant est **vsftpd** (***V**ery **S**ecure **FTP D**aemon*).
Sa devise « Beat me, Break me ».
Il a l'avantage d'être très petit, performant et rapide tout en étant tout de même très configurable néanmoins en deçà de Proftpd ou d'autres. C'est un service qui peut aussi bien être lancé par **xinetd** en tant que service seul.
Deux niveaux de sécurité sont utilisables :

- **Anonyme :** tout le monde peut se connecter au serveur FTP en tant que utilisateur **ftp** ou **anonymous**.
  L'environnement FTP est chrooté, l'utilisateur connecté ne peut voir la racine du système.

- **Utilisateur :** les utilisateurs qui existent sur le serveur peuvent se connecter avec leur mot de passe et ont un accès complet à leurs données dans leur répertoire personnel.

Les utilisateurs anonymes étant considérés comme l'utilisateur ftp, c'est le répertoire personnel de ce compte qui est la racine du ftp.

Le fichier de configuration est présent dans **« /etc/vsftpd/vsftpd.conf »**.

La racine du ftp par défaut est **chrooté sous dans /var/ftp en anonymous**.
Sinon dans le répertoire personnel de chaque utilisateur en non chrooté.

Le script de lancement est **/etc/init.d/vsftpd** (service vsftpd start).
Pour activer ou non l'accès anonyme on modifie le fichier de configuration.
Dans ce cas, l'utilisateur peut se connecter en tant que anonymous ou ftp.
Dans tous les cas, il sera reconnu comme utilisateur « ftp » du serveur une fois connecté :

anonymous_enable=YES/NO

Pour activer ou non l'envoi de fichiers sur le serveur par des anonymes. Dans ce cas, l'autorisation d'écriture dans un répertoire est fonction des droits du répertoire sur le serveur (notamment si l'utilisateur ftp a le droit d'écrire ou non dans un répertoire) :

anon_upload_enable=YES/NO

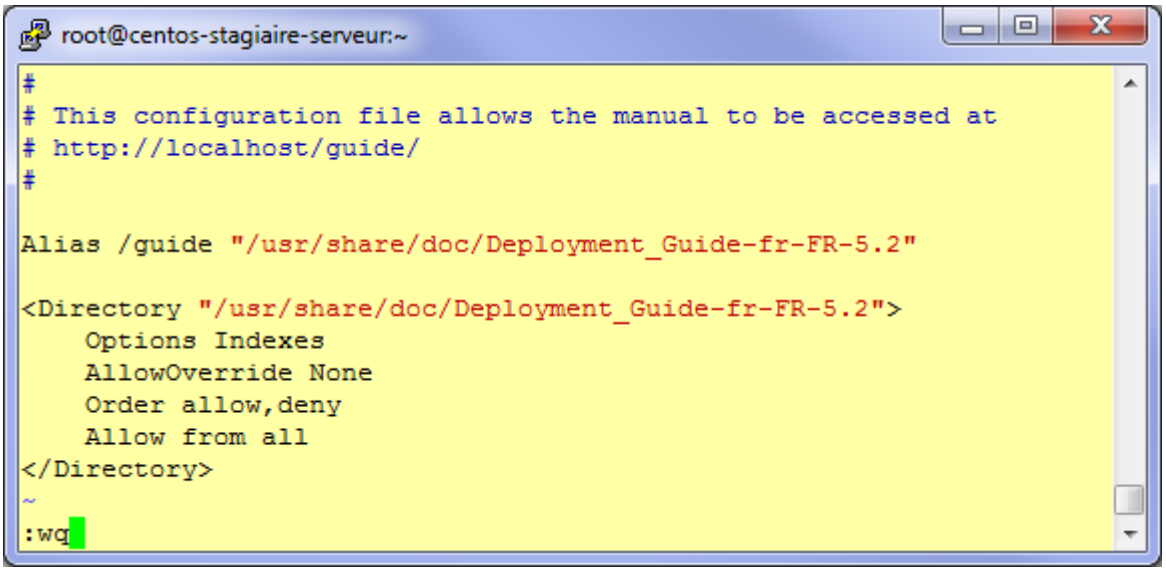
Vous pouvez interdire à des utilisateurs de se connecter en plaçant leurs noms dans **« /etc/vsftpd.ftpusers ».**
Vous pouvez ajouter des utilisateurs dans « **/etc/vsftpd.user_list »** si « **userlist_enable=YES »**. Dans ce cas, c'est la valeur de **userlist_deny (YES/NO)** qui déterminera si le fichier contient les utilisateurs interdits ou autorisés.
On peut créer dans chaque répertoire du serveur un fichier **.message**. Dans ce cas, son contenu sera affiché lors de l'accès au répertoire.

Toutefois attention, FTP n'est pas un protocole sûr, les mots de passe circulent en clair. Il donc est préférable de nos jours d'utiliser SFTP qui couvre la plupart des fonctionnalités de FTP et propose un cryptage fort. (SFTP est un dérivé d'OpenSSH)

HTTP

Créer un fichier comme suit :

« vim /etc/httpd/conf.d/guide.conf »



Puis lancez le serveur Apache (httpd)

service httpd start (ou reload si il est déjà lancé)

Pointer votre navigateur Web sur :
http://<ip_votre_serveur>/guide

Vous avez accès à la documentation française de Centos.

Donc maintenant deux manières d'en savoir plus sur Apache intégré à Centos :

- Sur votre serveur, directement ici
  http://<ip_votre_serveur>/guide/#ch-httpd

- Ou sur internet là
  http://www.centos.org/docs/5/html/5.2/Deployment_Guide/ch-httpd.html

Ordonnancement et automatisation

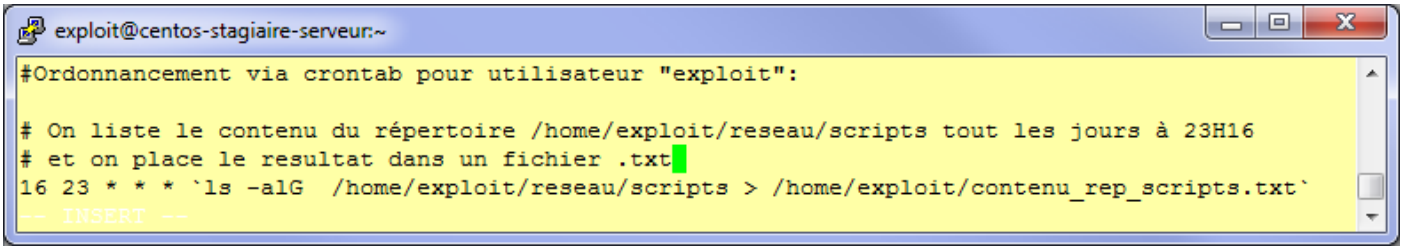
L'ordonnancement consiste à faire **exécuter automatiquement des tâches à des heures et fréquences définies**.
Sur des serveurs en production cela est particulièrement utile pour lancer des tâches qui doivent être effectuées durant les heures de fermetures de votre entreprise, sans bien sur solliciter votre présence.

Ex. : sauvegarde, export de base de données, nettoyage des journaux systèmes, mise en production de nouvelles versions de logiciels, maintenance applicative etc.

Crontab

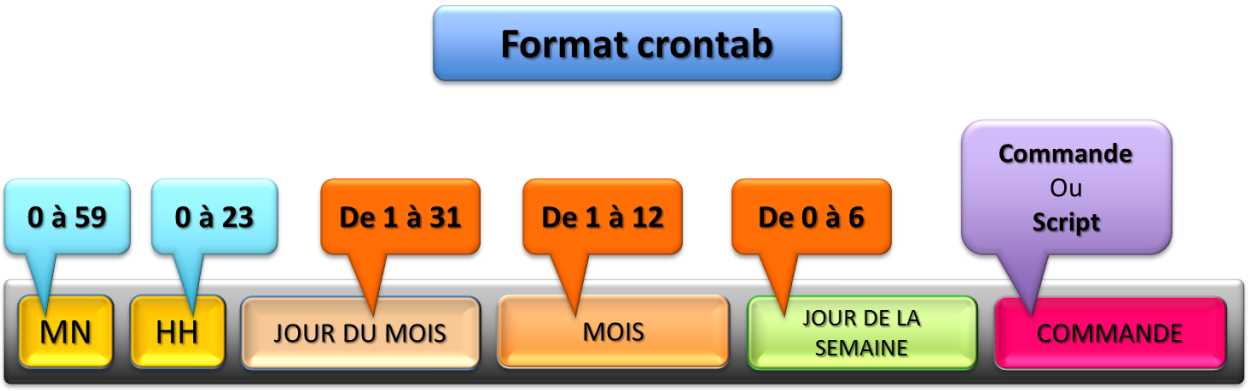
Le système GNU/Linux propose donc un service (démon) permettant de réaliser ce type de service, il s'agit du démon **crond (service crond start|stop|status)**. Par défaut il doit être lancé dès le démarrage.
Il se base sur une table des tâches à lancer pour chaque utilisateur. Cette table est stockée dans le fichier « **/var/spool/cron/<utilisateur>** ».

Cette table (un fichier au format texte) est éditable avec l'outil suivant **« crontab -e »**.
Exemple pour l'utilisateur « exploit » :



Note : on peut **consulter la table des tâches** à effectuer avec **« crontab -u user -l »,** la **supprimer** avec **« crontab -u user -r »,** et **« crontab -u user -e »** pour editer celle d'un utilisateur particulier.

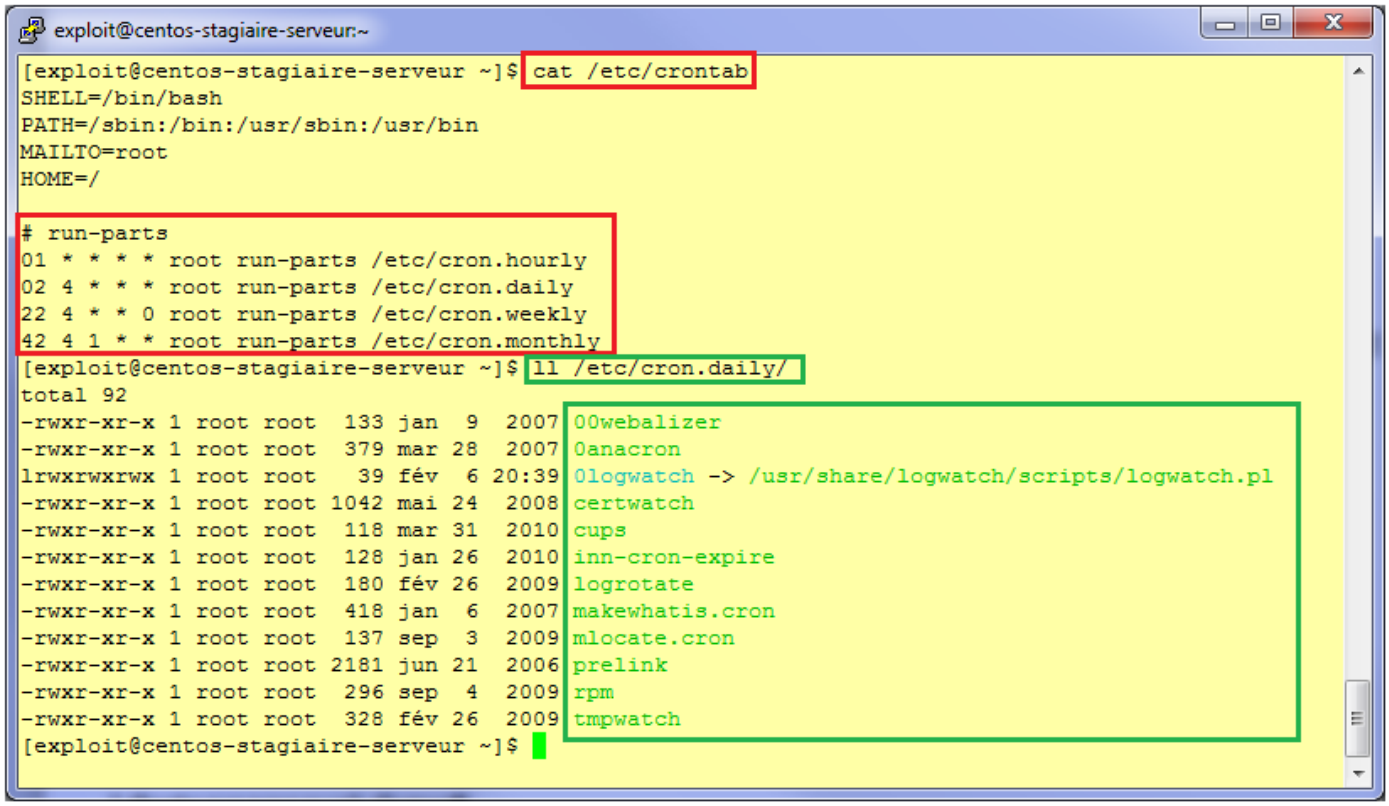
Voici le format d'une ligne de cette table (**crontab**) :



- Une valeur pour indiquer quand (MN, HH, JDM, M, JDS, COMMANDE) il faut exécuter la commande.
- Une liste de valeurs séparées par des virgules. Ex : 1,4,7 dans le champ mois pour janvier, avril, juillet.

- Un intervalle de valeurs. Ex : 1-4dans le champ jour de la semaine indique du lundi (1) au jeudi (4). Le 0 est le dimanche et le 6 le samedi.
- Le caractère * pour toutes les valeurs possibles. Ex : * dans le champ jour du mois indique tous les jours du des mois.

Il existe également une **crontab** système qui planifie (par heure, par jour, par semaine, par mois) les tâches de maintenance du système voici son contenu :



Dans l'exemple ci-dessus vous pouvez observer entre autre :

- la **rotation des journaux systèmes (logrotate)** s'effectue **tous les jours** (cron.daily ⇔ logrotate),
- le **nettoyage des fichiers temporaire** (**tmpwatch**).

Vous pouvez contrôler l'accès à la commande **crontab** par utilisateur avec les fichiers **« /etc/cron.allow »**et **« /etc/cron.deny »**.

- Si **cron.allow est présent**, seuls les utilisateurs qui y sont explicitement indiqués peuvent utiliser at (cf. Automatisation Avec la commande **« at »**).
- Si **cron.allow est absent**, cron vérifie la présence d'un fichier **cron.deny**. Tous les utilisateurs n'y étant pas sont autorisés à utiliser cron. S'il est vide la commande cron est autorisée pour tout le monde.
- Si les deux fichiers sont absents, seul root peut utiliser cron.

Sachez qu'il existe une autre commande pour lancer des tâches : **« at »**. Contrairement à **Crontab** les modifications sont volatiles.

Ordonnanceur : Ortro

Un des gros inconvénients de l'ordonnancement proposé par **crontab** provient du fait que cet outil **agit sur un serveur à la fois**. Même en disposant d'une batterie de scripts bien conçus : cela devient vite fastidieux.

C'est pour cela qu'il existe des ordonnanceurs professionnels de plus ou moins bonne qualité.
Le but ici n'est pas d'étudier ou de faire la promotion d'un ordonnanceur commerciale particulier, mais :

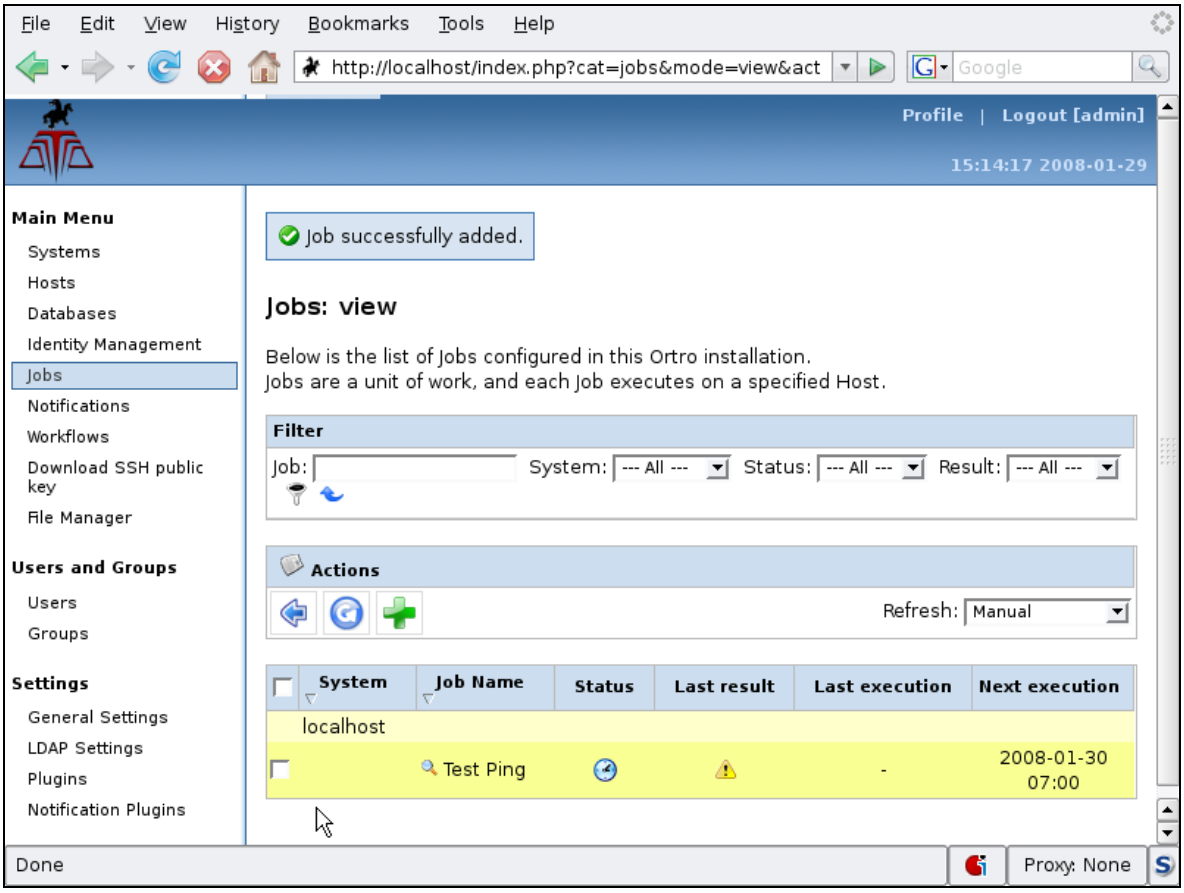
- De un : vous alerter sur la nécessité de superviser votre production efficacement,
- De deux : avoir un outil visuel pour gérer de façon rapide et efficace la bonne exécution de vos scripts etc.

De plus dans le cadre du « best effort » et de la mutualisation des moyens, il est judicieux de vous doter d'un ordonnanceur évolué pour la gestion de votre production quotidienne qui peut comprendre :

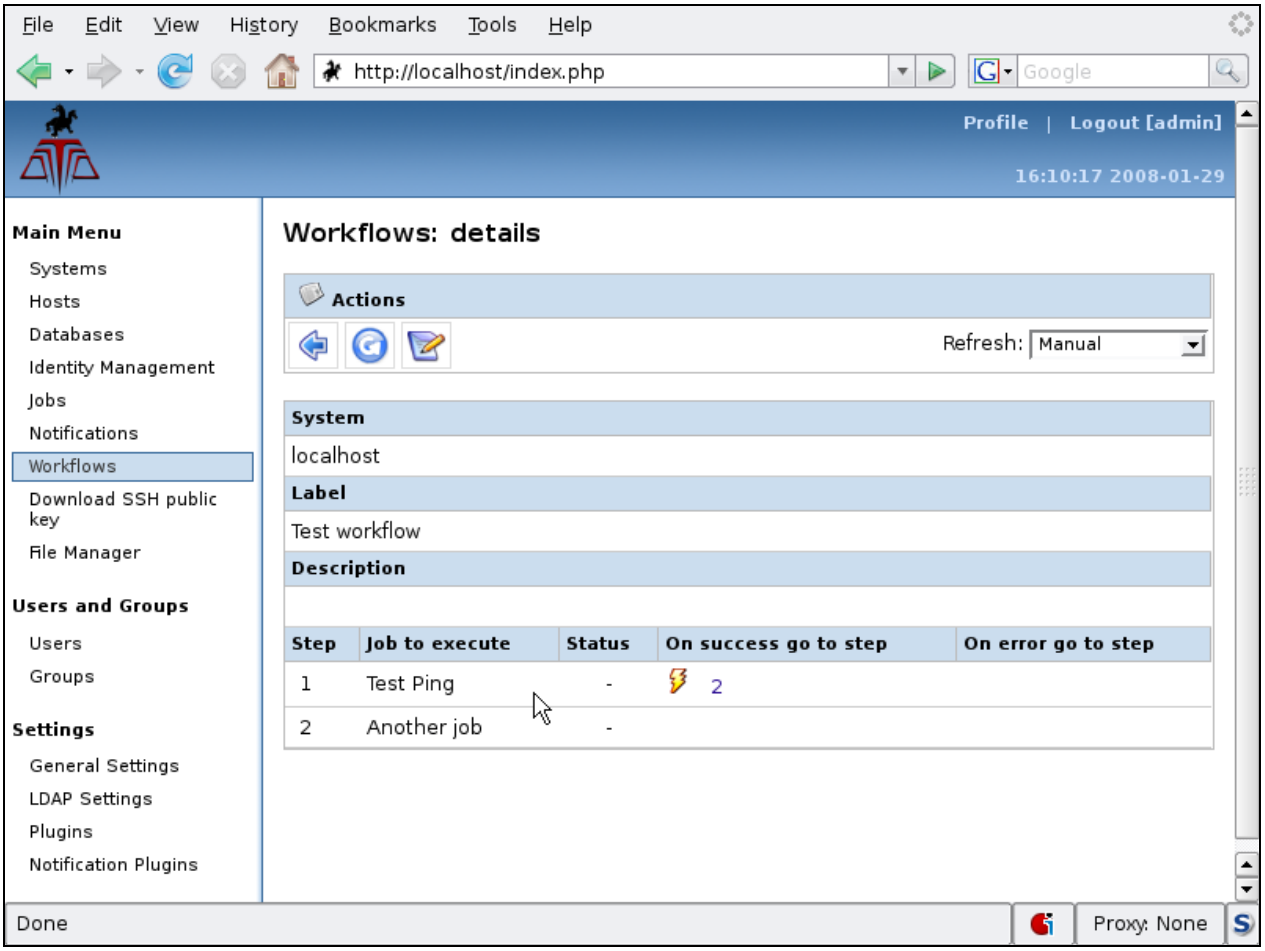
- Les sauvegardes,
- L'export de base de données,
- Le nettoyage des journaux applicatifs,
- La mise en production de nouvelles versions logicielles,
- La maintenance applicative,
- etc.

Voici à quoi ressemble un ordonnanceur.
Ici ORTRO, qui est un logiciel simple à mettre en œuvre et sous licence GPL :

Ici, une vue d'un job (tâche) simple qui consistait à effectuer un **« ping » le 30 janvier 2008**.



Le but d'un ordonnanceur est de pouvoir enchainer les jobs (tâches) de façon logique. Voici un exemple d'enchaînement simple de tâches (jobs) :



Pour plus d'informations ou de captures d'écran c'est ici (le logiciel est francisé une fois installé) : http://www.ortro.net/screenshots

Bien entendu si vous avez le budget, n'hésitez pas à vous doter d'un outil plus professionnel. Cependant attention à bien calibrer votre choix par rapport à vos besoins donc la taille de votre production.

Scripts centralisés : une nécessité

Discussions/notes

La supervision du serveur

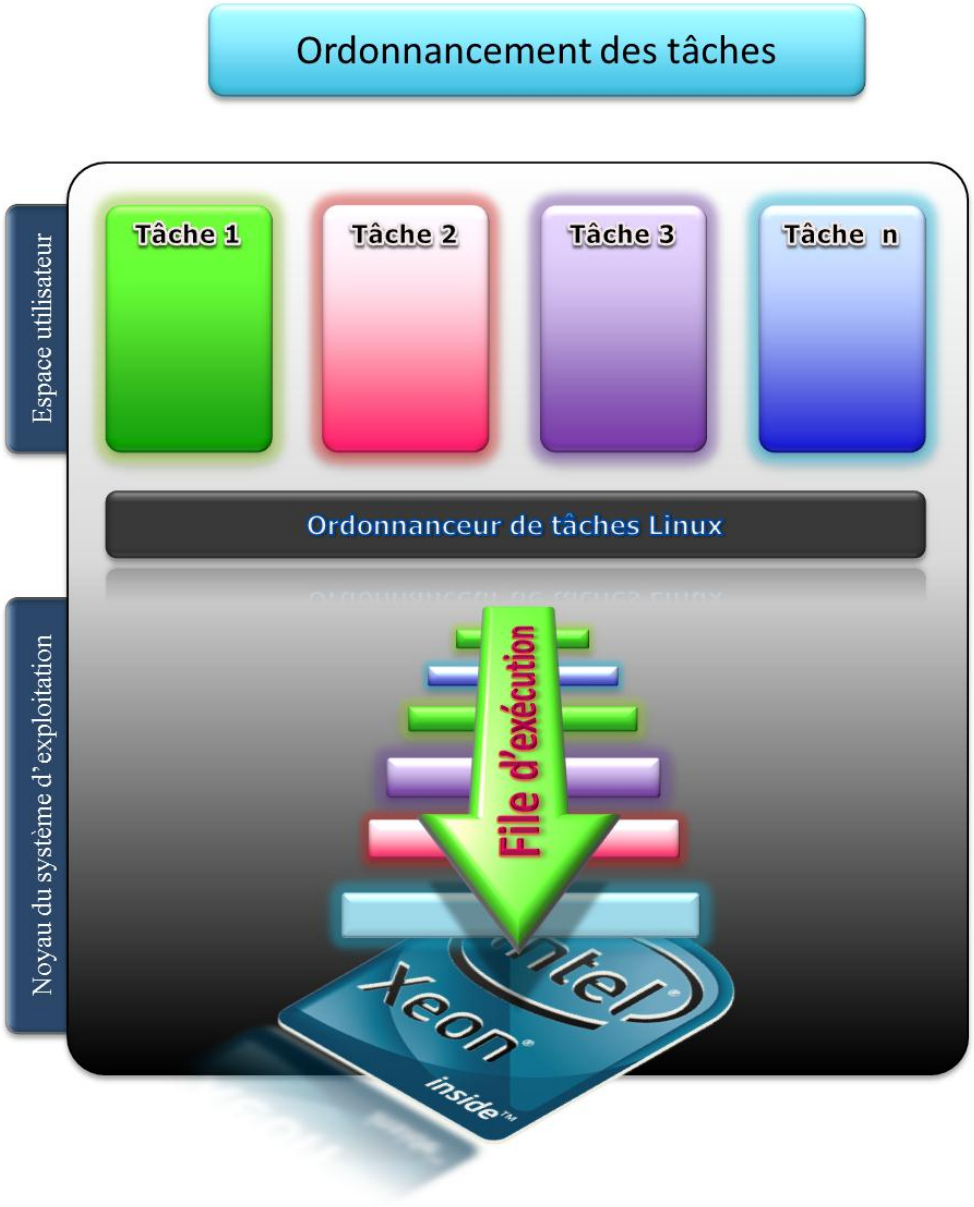
Bien qu'un serveur tournant sous GNU/Linux soit très stable et qu'il soit recordman des « **uptime** » sans la moindre intervention humaine : vous serez certainement amené à agir sur les processus, la mémoire et superviser vos serveurs pour en avoir un état précis à un instant donné.

Gestion des processus

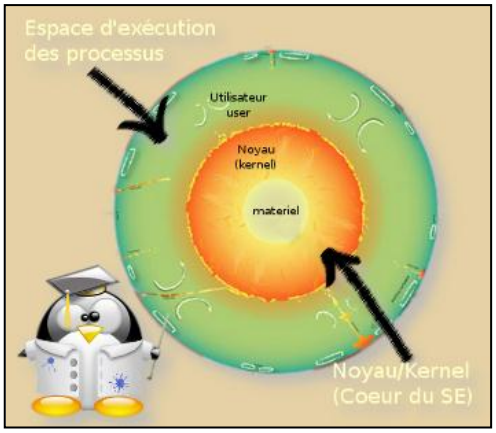
Tout d'abord il va vous falloir maîtriser la gestion des processus (tâches).
Une tâche (processus) représente :

- Un programme,
- Son environnement d'exécution (état du processeur, espace mémoire, identification, père, contexte de sécurité etc.).

Le noyau Linux travaille via son ordonnanceur de tâches en respectant le principe du « time sharing ».
Voici un schéma (simplifié) de l'ordonnancement des tâches :



Voici quelques données d'identification d'un processus :

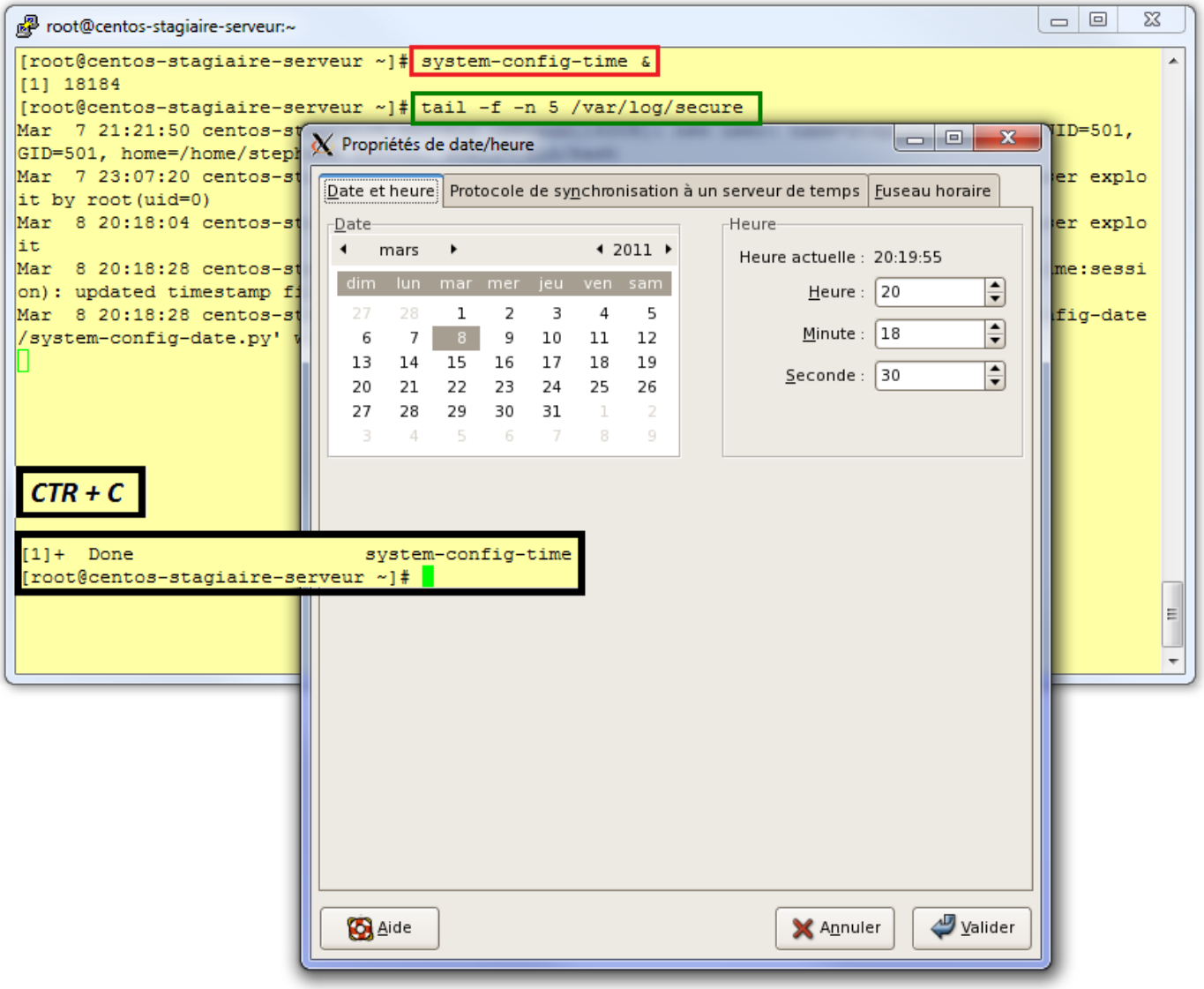


- **Un numéro de processus unique PID (Process ID)** : chaque processus Unix est numéroté afin de pouvoir être différencié des autres. Le premier processus lancé par le système est 1 et il s'agit d'un processus appelé généralement init. On utilise le PID quand on travaille avec un processus. Lancer 10 fois le même programme (même nom) produit 10 PID différents.
- **Un numéro de processus parent PPID (Parent Process ID)** : chaque processus peut lui-même lancer d'autres processus, des processus enfants (child process). Chaque enfant reçoit parmi les informations le PID du processus père qui l'a lancé. Tous les processus ont un PPID sauf le processus 0 qui est un pseudo-processus représentant le démarrage du système (crée le 1 init).
- **Un numéro d'utilisateur et un numéro de groupe : correspond à l'UID et au GID** de l'utilisateur qui a lancé le processus. C'est nécessaire pour que le système sache si le processus a le droit d'accéder à certaines ressources ou non. Les processus enfants héritent de ces informations. Dans certains cas (que nous verrons plus tard) on peut modifier cet état.
- **Durée de traitement et priorité** : la durée de traitement correspond au temps d'exécution écoulé depuis le dernier réveil du processus. Dans un environnement multitâche, le temps d'exécution est partagé entre les divers processus, et tous ne possèdent pas la même priorité. Les processus de plus haute priorité sont traités en premier. Lorsqu'un processus est inactif, sa priorité augmente afin d'avoir une chance d'être exécuté. Lorsqu'il est actif, sa priorité baisse afin de laisser sa place à un autre. C'est l'ordonnanceur de tâches du système qui gère les priorités et les temps d'exécution.
- **Répertoire de travail actif** : à son lancement, le répertoire courant (celui depuis lequel le processus a été lancé) est transmis au processus. C'est ce répertoire qui servira de base pour les chemins relatifs.
- **Fichiers ouverts** : table des descripteurs des fichiers ouverts. Par défaut au début seuls trois sont présents : 0, 1 et 2 (les canaux standards). À chaque ouverture de fichier ou de nouveau canal, la table se remplit. À la fermeture du processus, les descripteurs sont fermés (en principe).
- On trouve d'autres informations comme la **taille de la mémoire allouée**, la **date de lancement** du processus, **le terminal d'attachement, l'état du processus**, les UID effectif et réel ainsi que les GID effectif et réel.

Durant sa durée de vie (temps entre le lancement et la sortie) un processus peut passer par divers états ou « process state » :

- exécution en mode utilisateur (user mode) ;
- exécution en mode noyau (kernel mode) ;
- en attente E/S (waiting) ;
- endormi (sleeping) ;
- prêt à l'exécution (runnable) ;
- endormi dans le swap (mémoire virtuelle) ;
- nouveau processus;
- fin de processus sans rattachement à un processus père (zombie).

Lancer un processus en tâche de fonds.
Pour illustrer cela on peut par exemple lancer un outil graphique via le déport d'affichage X de OpenSSH (et Xming ouvert sur Windows) tout en continuant de lancer une autre commande (**tail -f**). Pour cela vous devez **faire suivre votre commande de « & »** :



Pour connaitre la liste de tous les processus **« ps »**, cette commande est souvent combinée avec **« | grep »** :

- **ps -ef**

Voici l'explication des champs en sortie :

Champs	Signification
UID	User ID, propriétaire du processus
PID	Process ID, numéro du processus
PPID	Parent Process ID, numéro du processus père (pour beaucoup kthread)
C	Ordre de priorité pour l'ordonnanceur, plus la valeur est grande plus la priorité est élevée : très utile pour calmer un processus qui mange toutes les ressources
STIME	Heure de lancement du processus
TTY	Nom du terminal depuis lequel a été lancé le processus

TIME	Durée de traitement du processus
CMD	Commande exécutée (nom du programme ou script)
S	Etat du processus R (Running) Z (Zombie) S (Sleeping)
PRI	Priorité du processus
NI	Nice, incrément pour l'ordonnanceur.

Pour tuer un processus
Vous devez prendre connaissance avec « ps -ef » du PID du processus puis lancer la commande :

- **kill -9 >PID_Processus>**

Cette commande envoie un signal au processus indiqué en paramètre. Il existe plusieurs type de signaux, ici pour tuer, sans attendre, le processus nous avons envoyé un signal SIGKILL(9).

La commande suivante vous donne une liste de tous les signaux existants :

- **kill -l**

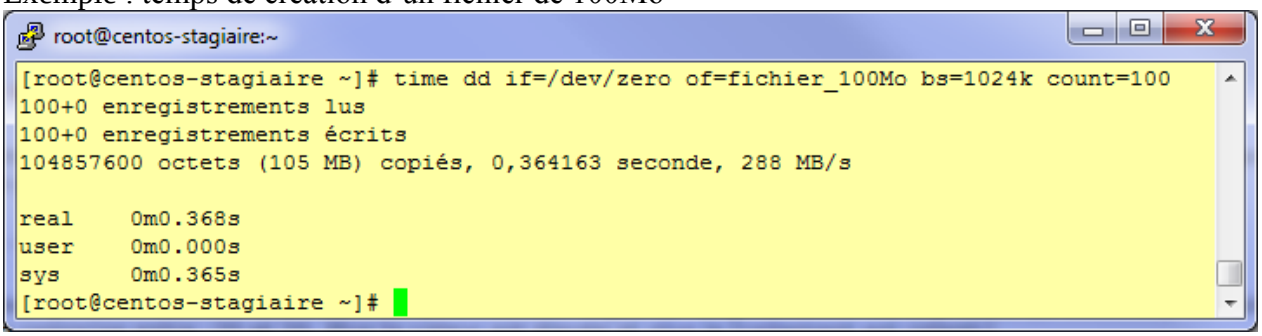
Quand le Shell est quitté (exit, [Ctrl] D...) le signal 1 SIGHUP est envoyé aux enfants pour qu'ils se terminent aussi. Lorsqu'un traitement long est lancé en tâche de fond et que l'utilisateur veut quitter le Shell, ce traitement sera alors arrêté et il faudra tout recommencer. Le moyen d'éviter cela est de lancer le traitement (processus) avec la commande **« nohup »**.
Dans ce cas le processus lancé ne réagira plus au signal SIGHUP, et donc le Shell pourra être quitté, la commande continuera son exécution.
Par défaut les canaux de sortie et d'erreur standards sont redirigés vers un fichier **nohup.out**, sauf si la redirection est explicitement précisée.

Uns commande assez intéressante pour gérer finement vos processus est « **nice ».**
Elle permet de lancer une commande avec une priorité pus faible, afin de permettre éventuellement à d'autres processus de tourner plus rapidement.

nice [-valeur] commande [arguments]

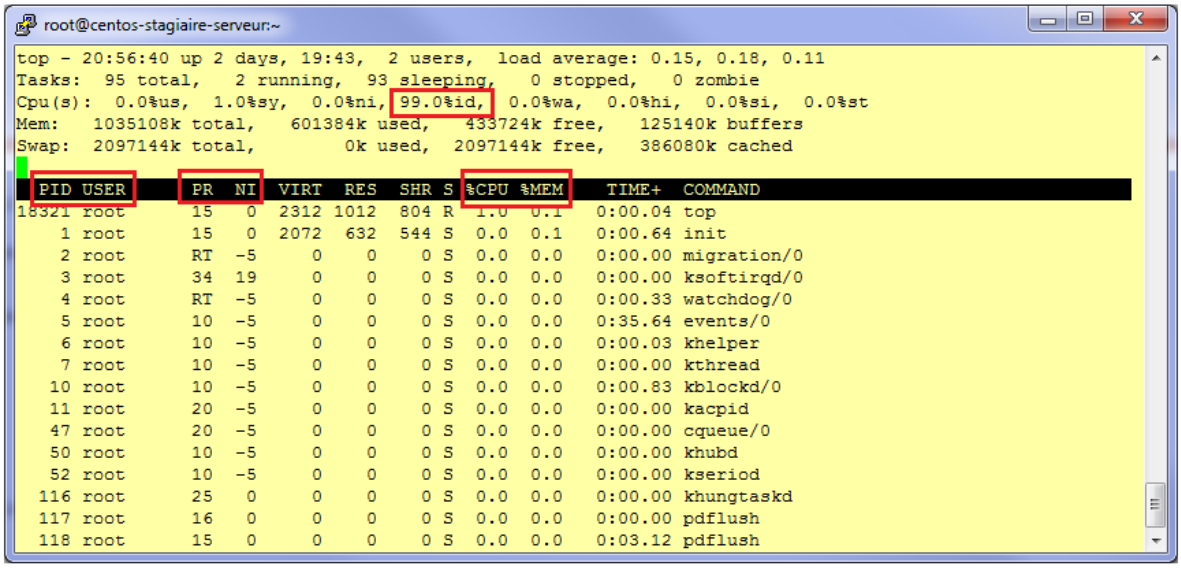
Une valeur positive causera une baisse de priorité, une négative l'augmentation de la priorité (si autorisé). La valeur doit être comprise entre 20 et 20. Plus la valeur est élevée et plus le traitement est ralenti.
« renice » permet de la faire pour d'autres utilisateurs.

Si vous désirez évaluer le temps que prends une commande pour s'exécuter vous pouvez utiliser la commande **« time ».**
Exemple : temps de création d'un fichier de 100Mo

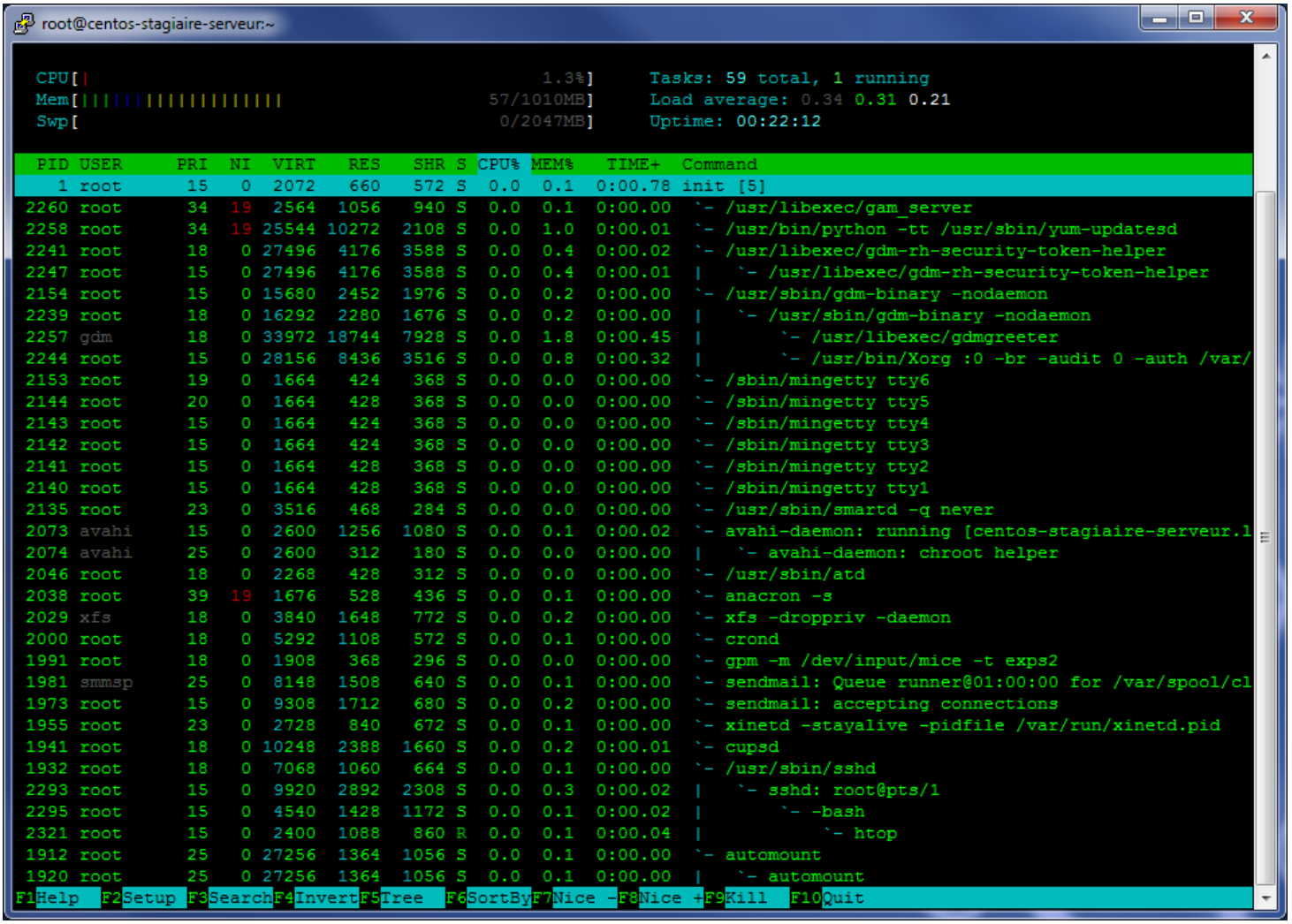


- **real** : durée totale d'exécution de la commande ;
- **user** : durée du temps CPU nécessaire pour exécuter le programme ;
- **system** : durée du temps CPU nécessaire pour exécuter les commandes liées au système d'exploitation.

Pour visualiser en temps réel vos processus vous pouvez utiliser **« top » mieux « htop »** :

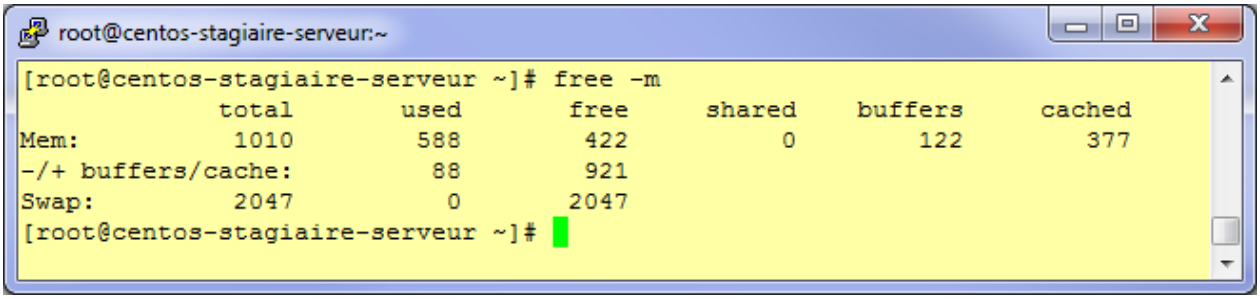


La version indispensable **« htop »,** plus visuelle, et dotée de plus de fonctionnalités :



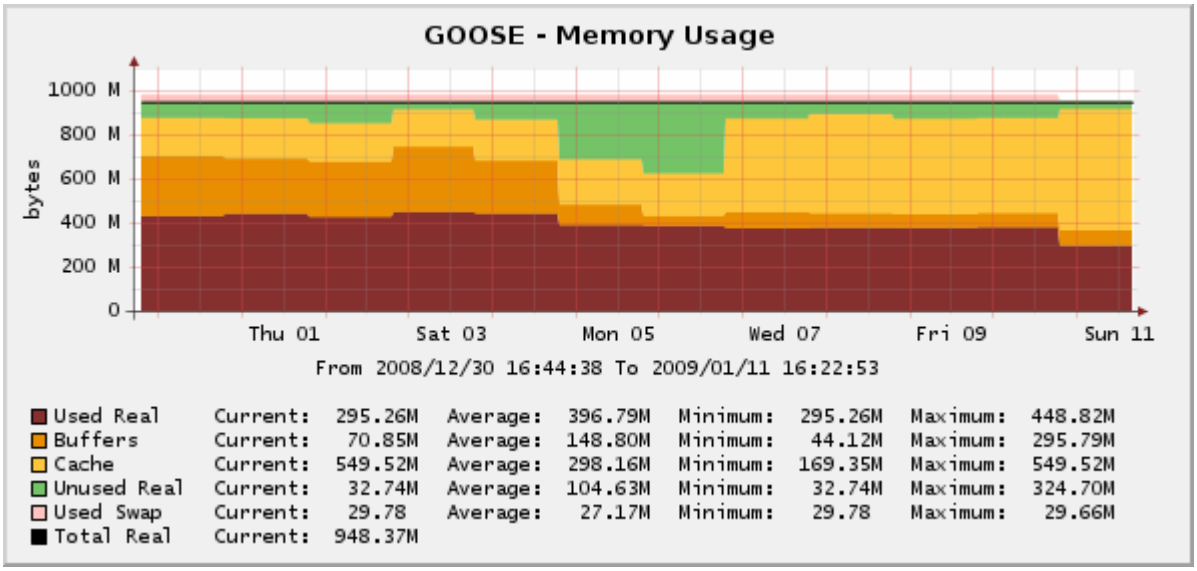
Gestion mémoire

Sous GNU/Linux vous disposez de l'outil **« free »** pour observer la consommation mémoire de votre système.



Cependant le noyau Linux fait ce qu'on appelle du « **provisioning** », c'est à dire qu'il **réserve presque toute la mémoire disponible** pour pouvoir la **redistribuer au moment voulu**, lorsque les processus en seront demandeurs.
De ce fait n'allez pas croire que votre système n'a plus de mémoire vive : c'est juste le comportement typique du noyau Linux.

Voici un audit mémoire pour étayer ces propos (voir Cacti) :



Par contre si vous constatez une **utilisation anormale de votre fichier de swap** : là oui, il y'a un sous-dimensionnement possible de la mémoire vive de votre serveur.
C'est pour cela qu'il est important de dimensionner correctement la taille de sa partition de **swap**.

Rappel (Page 86)

Si RAM < 512 Mo => Taille SWAP=2xRAM,

Si 1Go < RAM < 4Go => Taille SWAP = RAM enfin si RAM > 4Go => Taille SWAP = 4Go.

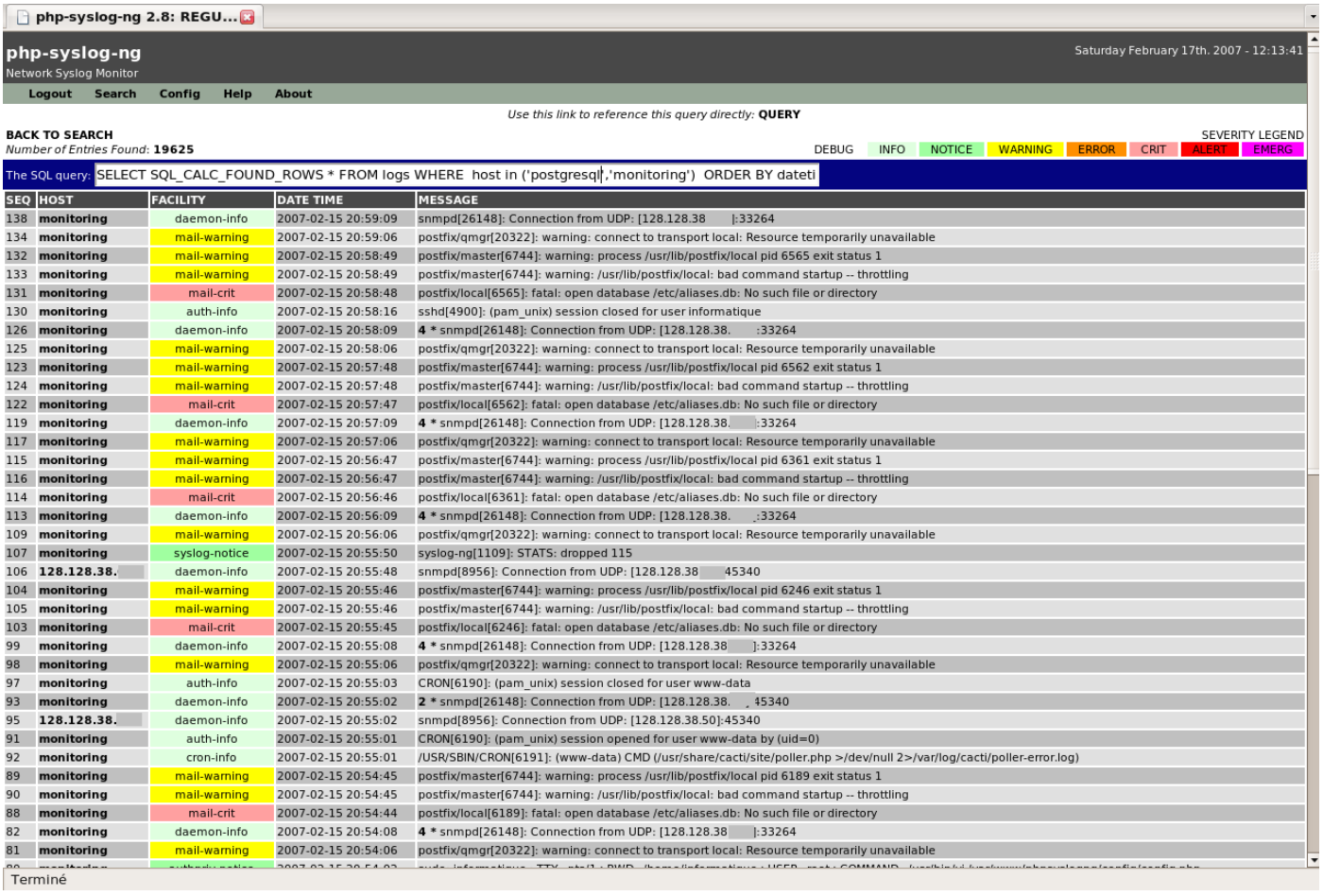
Gestion des journaux

Syslog(-ng)

Syslog est un protocole définissant un service de journaux d'événements d'un système informatique. C'est aussi le nom du format qui permet ces échanges.

En tant que protocole, **Syslog** se compose d'une partie cliente et d'une partie serveur. La partie cliente émet les informations sur le réseau, via le **port UDP 514**. Les serveurs collectent l'information et se chargent de créer les journaux.
L'intérêt de **Syslog** est donc de centraliser les journaux d'événements, permettant de repérer plus rapidement et efficacement les défaillances des serveurs du réseau que vous supervisez.

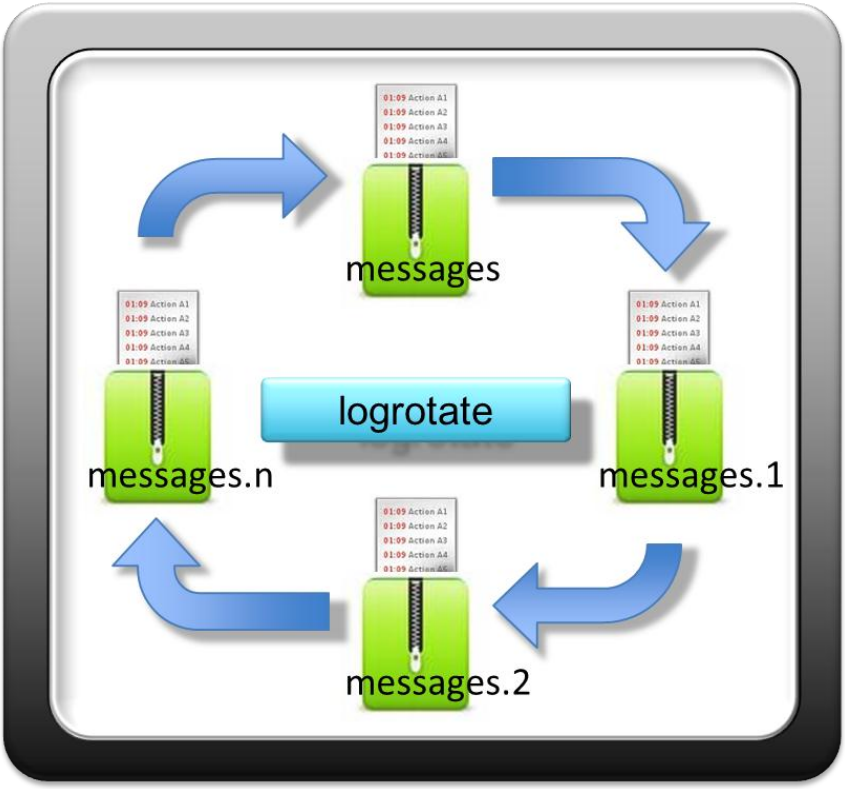
Il existe des frontaux graphique plus ou moins évolués pour syslog(ng).
« php-syslog-ng » écrit en HTML/PHP en est un :



Logrotate

Cet outil permet de stocker vos journaux dans un format **compressé (gzip) ou pas,** puis de les **effacer cycliquement**.
Il est donc important de régler vos rotations de journaux afin de ne pas saturer vos disques. Pour ce faire vous devez modifier le fichier **« /etc/logrotate.conf ».**

Schéma conceptuel :



Voici en pratique ce que cela donne :

root@centos-stagiaire:~

```
[root@centos-stagiaire ~]# ll /var/log/
total 1760
-rw-r----- 1 root  root    4080 mar 11 09:04 acpid
-rw------- 1 root  root  434440 fév  3 22:49 anaconda.log
-rw------- 1 root  root   20492 fév  3 22:49 anaconda.syslog
-rw------- 1 root  root   42027 fév  3 22:49 anaconda.xlog
drwxr-x--- 2 root  root    4096 fév  3 22:51 audit
-rw------- 1 root  root       0 mar  6 04:05 boot.log
-rw------- 1 root  root       0 mar  5 22:20 boot.log.1
-rw------- 1 root  root      93 mar  5 21:14 boot.log.2
-rw------- 1 root  utmp    1152 fév 10 09:16 btmp
drwxr-xr-x 2 root  root    4096 nov 11  2007 conman
drwxr-xr-x 2 root  root    4096 nov 11  2007 conman.old
-rw------- 1 root  root    9257 mar 11 09:04 cron
-rw------- 1 root  root    1010 mar  6 04:02 cron.1
-rw------- 1 root  root    7035 mar  5 22:17 cron.2
drwxr-xr-x 2 lp    sys     4096 mar  5 22:20 cups
-rw-r--r-- 1 root  root   13468 mar 11 09:03 dmesg
-rw------- 1 root  root    2424 fév  3 22:49 faillog
drwxr-xr-x 2 root  root    4096 mar 11 09:04 gdm
drwx------ 2 root  root    4096 avr  4  2010 httpd
-rw-r--r-- 1 root  root  146292 mar 11 09:05 lastlog
drwxr-xr-x 2 root  root    4096 fév  3 22:46 mail
-rw------- 1 root  root  201089 mar 11 09:04 maillog
-rw------- 1 root  root    2970 mar  6 04:05 maillog.1
-rw------- 1 root  root    6110 mar  5 22:20 maillog.2
-rw------- 1 root  root   26505 mar 11 09:04 messages
-rw------- 1 root  root      57 mar  5 22:20 messages.1
-rw------- 1 root  root  275959 mar  5 21:37 messages.2
```

Supervision centralisée

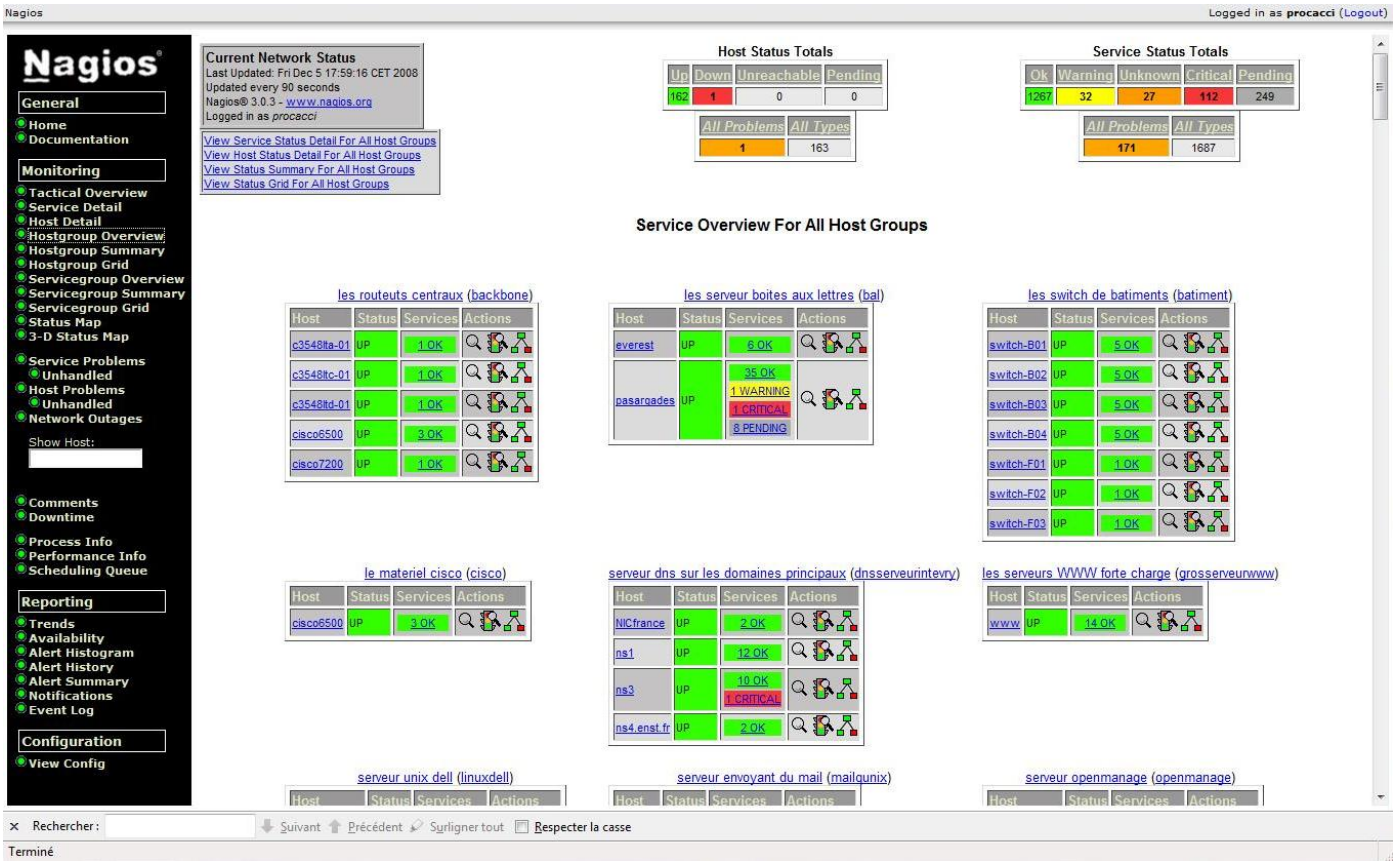
Au-delà de la dizaine de serveurs administrés il devient indispensable de disposer d'un outil qui puisse vous donner l'état de votre SI en temps réel. Afin d'être proactif il est intéressant de disposer sur un seul écran de la vision global de votre SI.

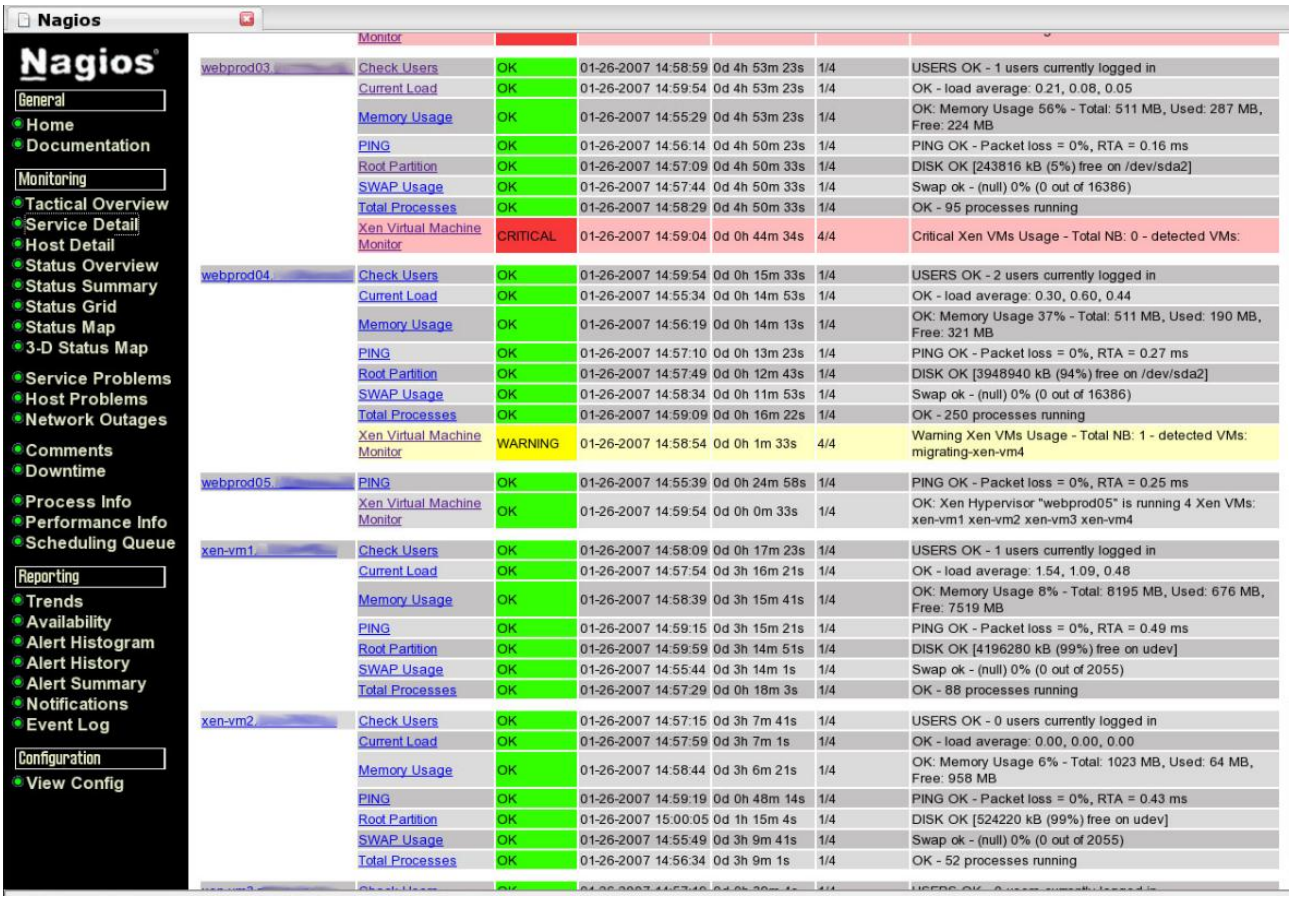
Nagios (anciennement appelé Net saint) est une application permettant la **surveillance système et réseau de façon visuelle et centralisée**. Nagios surveille les **hôtes et services spécifiés, alertant lorsque les systèmes présentes des disfonctionnement**.
C'est un logiciel libre sous licence GPL.
Nagios vous donne **l'état de vos serveurs, réseaux, switchs, services etc. en quasi temps réel**. Il est le compagnon idéal des administrateurs systèmes qui peuvent afficher son écran principal sur un grand écran visible de tous.

Nagios permet entre autre (source Wikipédia) :

- De superviser des services réseaux : (SMTP, POP3, HTTP, NNTP, ICMP, SNMP, LDAP, etc.)
- De superviser les ressources des serveurs (charge du processeur, occupation des disques durs, utilisation de la mémoire paginée) et ceci sur les systèmes d'exploitations les plus répandus.
- L'interfaçage avec le protocole SNMP.
- La supervision à distance peut utiliser SSH ou un tunnel SSL (notamment via un agent NRPE ;).
- L'utilisation des plugins qui sont écrits dans les langages de programmation les plus adaptés à leur tâche : scripts Shell (BAsh, ksh, etc.), C++, Perl, Python, Ruby, PHP, C#, etc.
- La vérification des services qui se fait en parallèle.
- La possibilité de définir une hiérarchie dans le réseau pour pouvoir faire la différence entre un serveur en panne et un serveur injoignable.
- La remontée des alertes entièrement paramétrable grâce à l'utilisation de plugins (alerte par courrier électronique, SMS, etc.).
- L'acquittement des alertes par les administrateurs.
- La gestion des escalades pour les alertes (une alerte non acquittée est envoyée à un groupe différent).
- La limitation de la visibilité, les utilisateurs peuvent avoir un accès limité à quelques éléments.
- De créer ses propres plugins, dans le langage désiré. Il suffit de respecter la norme Nagios des Codes retour.

Voici Nagios en action :





Audit de parc

Parfois il peut être intéressant de connaître avec précision et de manière suivi sur un laps de temps conséquent (jour, semaine, mois, année) :

- La charge de votre réseau,
- La charge de vos serveurs,
- Les flux réseaux,
- Etc.

Pour cela il existe des **outils d'audit**. Ne les confondez pas avec les outils de supervision comme vu précédemment leur finalités d'utilisation ne sont pas du tout les mêmes.
Voici les ténors qui sont gratuits (en partie pour certains) :

- **Cacti**,
- **Zabbix**,
- Centreon (Oreon) est un peu à part dans le sens où il intègre Cacti et Nagios.

Présentation de **Cacti** (source Wikipédia).

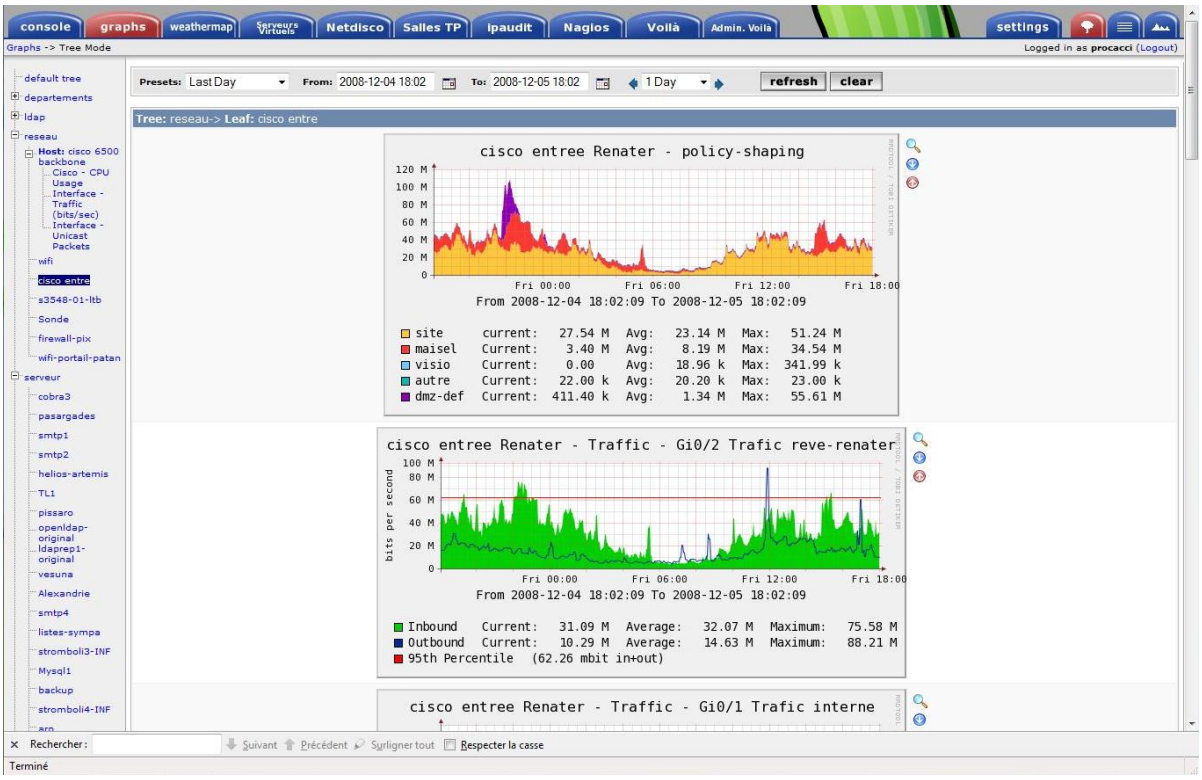
Cacti est un logiciel libre de mesure de performances réseau et serveur basé sur la puissance de stockage de données de RRDTool.
Il est bien souvent utilisé avec des logiciels de supervision (par exemple nagios), mais il ne fait pas de supervision en tant que tel.
Il ne fait pas de corrélation d'incidents ni d'alerte en cas d'incident (bien que des plugins existent, ce n'est pas son but premier).
Par ailleurs, il permet de faire l'étude d'indicateurs sur une période donnée (moyenne sur le mois par exemple, ou maximum de la semaine, etc ...) contrairement à la supervision qui permet de connaître l'état de l'indicateur en temps réel. Il fonctionne grâce à un serveur web équipé d'une base de données et du langage PHP. Il peut être considéré comme le successeur de MRTG et également comme une interface d'utilisation de RRDTool.

Il permet de représenter graphiquement divers statuts de périphériques et équipements réseau utilisant SNMP pour connaître la charge CPU, le débit des interfaces réseau, utilisation de la QOS sur une ligne, la qualité d'une liaison (CRC/s) ou encore la latence réseau. Cacti utilise aussi un système de scripts (Bash, PHP, Perl, VBs...) pour effectuer des mesures plus complexes, par exemple l'espace disque restant, la charge processeur pour un processus donné ou le temps de réponse applicatif.

L'attrait de ce logiciel réside principalement dans son principe de *modèles* (*Templates*) qui permet de créer de manière générique les graphiques afin de pouvoir les réutiliser.
Ce système peux sembler déroutant pour les nouveaux utilisateurs, mais montre vite bien des avantages lorsqu'il s'agit de superviser un grand nombre d'indicateurs et/ou d'équipements. Les possibilités d'import et d'export de ces *templates* permettent de les partager avec toute la communauté des utilisateurs.
Contrairement à MRTG qui régénère l'ensemble des graphiques toutes les 5 minutes, Cacti génère les images dynamiquement à l'affichage à partir des fichiers de données RRDTool. Cela permet par exemple de pouvoir zoomer sur une période ou changer dynamiquement la période du graphique.
Il est également possible d'effectuer des opérations simples (et des combinaisons d'opérations) avec les différentes données, avant leur affichage, grâce une interface graphique qui permet l'utilisation simplifiée de la commande CDEF de RRDTool. On peut ainsi convertir les octets en bits ou visualiser facilement un graphique en pourcentage.

Cacti en image :

Vue globale



Graphique par éléments (serveur, switch etc.)

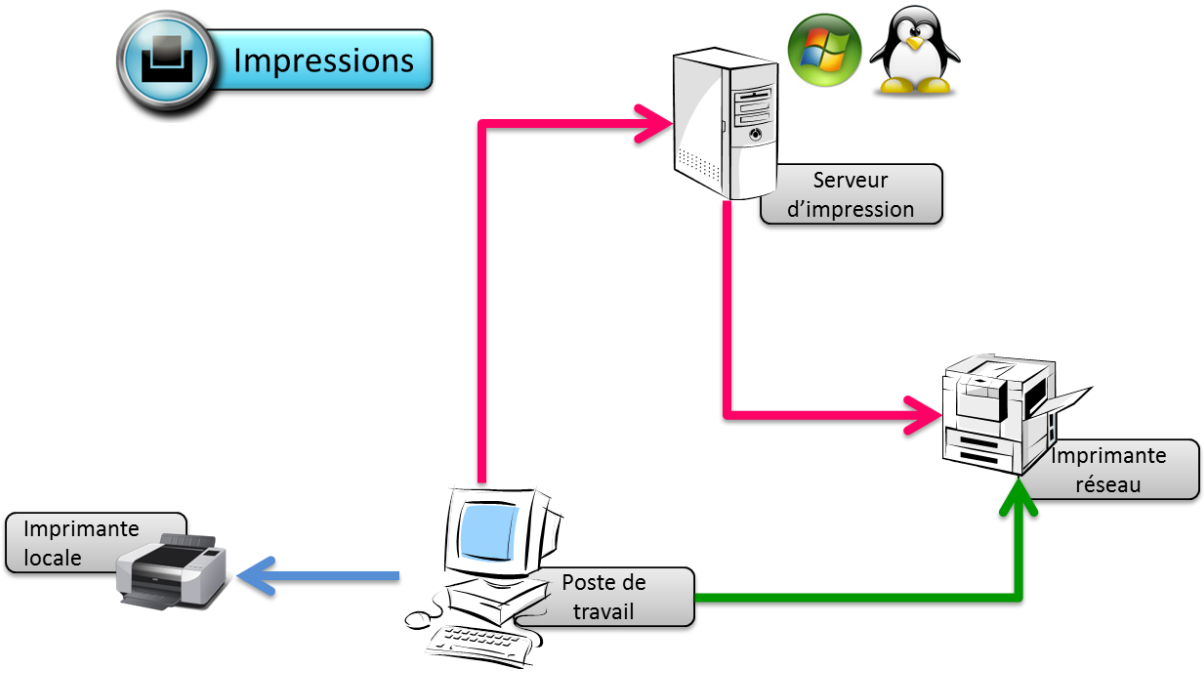
L'impression sous GNU/Linux

L'impression à partir de GNU/Linux (et autres Unix) peut se faire avec :

- SYSTEM V,
- BSD (LPR) ;
- CUPS.

Nous allons étudier l'impression via **CUPS**.

Rappel : les types d'impressions :



Architecture CUPS

CUPS (**C**ommon **U**nix **P**rinting System) est majoritairement développé par Apple Inc.

Il s'agit d'un système d'impression Unix orientée réseau.
Il s'appuie sur le protocole **IPP** (**I**nternet **P**rinting **P**rotocol).
Voici ses principaux avantages :

- Simple d'utilisation, notamment grâce à une configuration et une administration centralisée depuis une interface HTTP, des règles de conversion basées sur les types MIME, et des fichiers de description d'imprimante standards (PPD, PostScript Printer Description),
- CUPS reprend les commandes System V et BSD pour plus de simplicité,
- Les traces des impressions sont disponibles au format CLF (Common Log Format) de serveur Web et exploitables par les mêmes outils,
- CUPS est capable d'interagir avec les serveurs d'impression LPD pour garder une compatibilité ascendante,
- CUPS dispose de sa propre API permettant de créer des interfaces utilisateur pouvant s'intégrer dans des environnements graphiques ou des interfaces d'administration,
- Les pools d'impression permettent la redirection automatique des tâches,
- L'authentification est possible par utilisateur, hôte ou certificat numérique.

Configuration de CUPS

Le service d'impression se démarre comme suit :

- **service cups start**

L'administration de CUPS peut se faire de trois manières différentes sous Centos 5.x :

- En éditant les fichiers de configurations contenus sous **« /etc/cups/ »**,
- Via un navigateur internet en pointant sur l'URL : **http://localhost:631,**
- A l'aide de l'outil graphique Centos contenu dans le paquetage **« system-config-printer »**.

CUPSD est le spooler d'impression, il reçoit les requêtes d'impression des commandes utilisateurs (lp…) ainsi que les requêtes de gestion des commandes d'administration.
Quand CUPSD traite une requête d'impression, il :

- transmet les données à imprimer à un filtre (filter) qui dépend du modèle d'imprimante,
- transmet le résultat à un module terminal (backend) qui dialogue avec l'imprimante pour l'impression.

Les **échanges entre CUPSD**, le **filter**, le « **backend »** se font par des **répertoires de spools** (**« /var/spool/cups/ »**) et des **tubes** (pipe).
Les **filtres** transforment les données (documents, images, textes etc.) dans un format compréhensible par l'imprimante.
Les « **Backends »** réalisent directement l'impression. Il existe un « **backend** » par type d'impression : que ce soit une liaison local d'imprimante (USB, parallèle, série etc.) ou via un type de protocole réseau jouant le rôle de client réseau (le serveur étant l'imprimante elle-même) :

IPP : Client IPP,
LPD : Client LPD,
SMB : Client SMB via le service Samba,
Socket : Client JetDirect qui réalise directement une impression sur le port 9100 sans dialogue protocolaire,
Pap/cap : Client AppleTalk via le service **« netatalk »**.

CUPS se configurer avec les fichiers présents dans **« /etc/cups/ »**, voici les principaux :

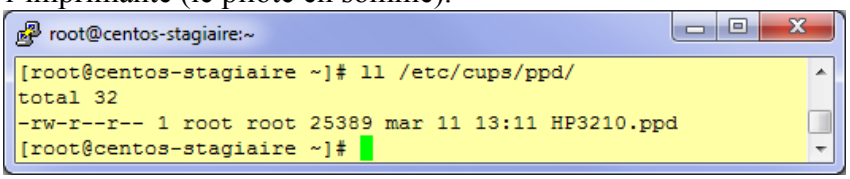
- **« /etc/cups/cupsd.conf »,** se configure à la manière du démon Apache (httpd)
  Ce fichier permet de configurer le démon CUPSD, sous la forme **« directive=valeur »,**
  Le nombre de directives possibles est important et tout comme Apache il ne sera pas détaillé ici,

- **« /etc/cups/client.conf »,** il permet de configurer le nom de la machine qui est cliente du démon **cupsd** (par défaut localhost),

- **« /etc/cups/printers.conf »,** définit les imprimantes locales disponibles,

- **« /etc/cups/classes.conf »,** permet de configurer des groupes (classes) d'imprimantes,

- Le répertoire **« /etc/cups/ppd »** contient les **fichiers .ppd** qui sont la définition complète de l'imprimante (le pilote en somme).

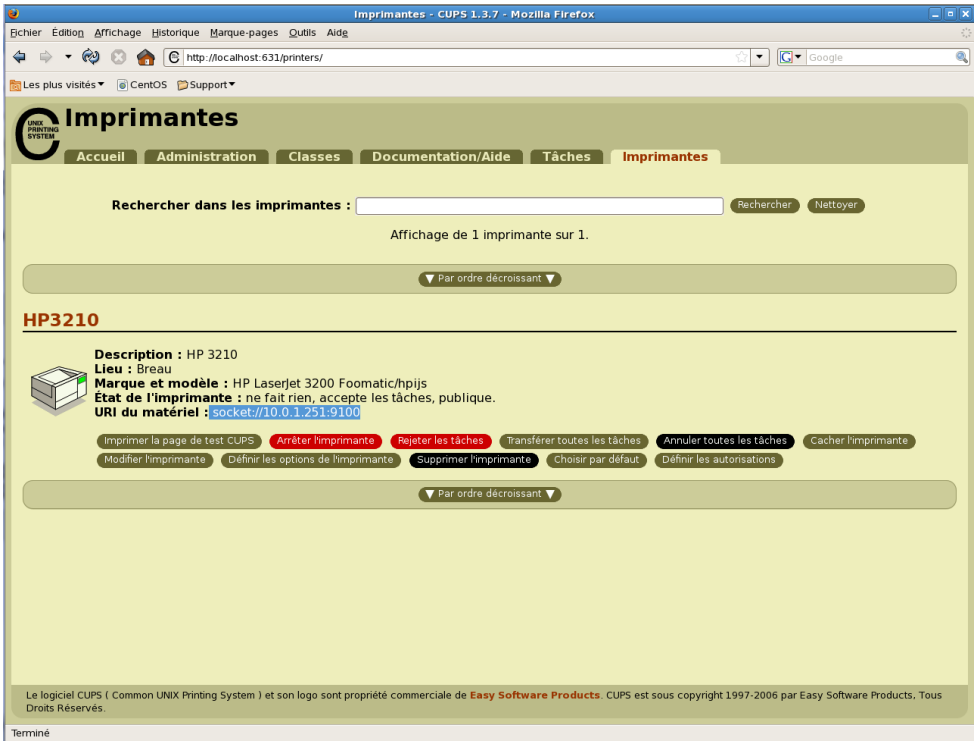


Vous pouvez également configurer votre serveur CUPS avec un navigateur internet. Cela est possible depuis une console d'administration à distance (Exemple : **10.0.1.10**), mais vous devrez éditer le fichier **« /etc/cups/cupsd.conf »** comme ci-dessous :

```
root@centos-stagiaire-serveur:~
#Listen localhost:631
Port 631 # Ecoutes sur n'importe quelle IP (localhost comme ip_serveur)
Listen /var/run/cups/cups.sock

# Show shared printers on the local network.
Browsing On
BrowseOrder allow,deny
# (Change '@LOCAL' to 'ALL' if using directed broadcasts from another subnet.)
BrowseAllow @LOCAL

# Default authentication type, when authentication is required...
DefaultAuthType Basic

# Restrict access to the server...
<Location />
  Order allow,deny
  Allow From 10.0.1.* # Gestion des tâches d'impression pour tous
</Location>

# Restrict access to the admin pages...
<Location /admin>
  Encryption Required
  Order allow,deny
  Allow From 10.0.1.10  # Console admin CUPS
</Location>

# Restrict access to configuration files...
<Location /admin/conf>
  AuthType Default
  Require user @SYSTEM
  Order allow,deny
  Allow From 10.0.1.10 # Console admin CUPS
-- INSERTION --                                          19,22         32%
```

➢ **service cups reload**

https://localhost:631/admin (sur le serveur) ou https://<ip_serveur_cups>:631/admin (à distance)



Commandes CUPS

Vous pouvez parfaitement gérer votre spooler d'impression en ligne de commandes.

Voici quelques commandes pour administrer CUPS en mode **CLI** (**C**ommand **L**ine **I**nterface) :

Commandes	Signification
lpinfo -m	Permet de lister les pilotes d'imprimantes disponibles..
lpinfo -v	Permet de lister les backends d'imprimantes disponibles
lpstat -r	Permet de savoir si le spooler d'impression (CUPSD) est actif
lpstat -p [printers]	Affiche la ou les queues d'impression gérées par CUPSD
lpstat -d	Affiche la queue d'impression par défaut
lpadmin	Configure les queues d'impression (printers) et groupes (class) (man lpadmin pour avoir toutes les options)
lpadmin -E	Active la queue d'impression et accepte les tâches d'impression dessus
lpadmin -d	Positionne la queue d'impression mentionnée comme destination d'impression par défaut
lpr <fichier>	Imprime le fichier passé en paramètre sur la queue d'impression par défaut
lpr -P <print_queue> <fichier>	Imprime le fichier passé en paramètre sur la queue d'impression mentionnée par <destination>
lpq -P <print_queue>	Liste les tâches d'impression de cette queue d'impression
lprm -P <print_queue> <tâche_impression>	Permet d'annuler une tâche d'impression de la queue d'impression, en précisant le numéro de tâche.

Astuce : Si vous disposez d'une imprimante HP, faites d'abord **« yum install hpijs »**
De plus si votre pilote d'imprimante n'est pas disponible nativement vous pouvez toujours **importer le « .ppd »** correspondant fourni par le **constructeur** de votre imprimante.

Le noyau linux

Le noyau est le cœur du système d'exploitation GNU/Linux. Linux en tant que tel est uniquement le nom du noyau lequel a été développé à l'origine par Linus Torvalds.

Le système d'exploitation GNU/Linux est composé du noyau et des outils d'exploitation de base.

Le noyau de Linux est libre. Ses sources sont disponibles. Il est donc possible de le recompiler pour l'adapter finement à ses besoins, de le modifier, d'y rajouter des extensions.
Le noyau Linux fait partie de la famille des noyaux monolithiques. C'est-à-dire que toutes ses fonctionnalités et composants sont regroupés dans un programme unique. Cependant depuis la version 2.0 (ou plutôt la version de développement 1.3 pour être plus précis) le noyau est devenu modulaire.

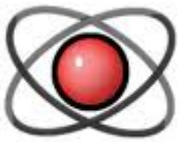
Le noyau est appelé Kernel. Il est situé dans le répertoire « **/boot »** et son nom, par convention, commence souvent par **vmlinuz-X.Y.Z.p-Vtxt.**
On obtient la version du noyau avec la commande « **uname** » (-a : affiche toutes les options possibles) :

- **uname -r**
  2.6.18-194.el5

Voici la signification des lettres du numéro de version du noyau :

- **X** : version majeure du noyau. Entre la version 1 et la version 2, le passage au fonctionnement modulaire a été déterminant, ainsi que la ré implémentation de la couche réseau.
- **Y** : une valeur paire représente une branche stable du noyau. Une version impaire représente une branche de développement (attention !). Chaque incrément pair (0,2,4,6) représente une évolution importante du noyau.
- **Z** : version mineure du noyau. Quand un lot de modifications par rapport à une version précédente nécessite la diffusion d'un nouveau noyau, alors on incrémente ce chiffre. Par exemple, un lot regroupant une modification du système son (Alsa qui passe de 1.0.8 à 1.0.9), du système de fichiers (ajout de ReiserFS 4), et ainsi de suite...
- **p** : version corrigée ou intermédiaire présente depuis la version 2.6. Quand le noyau nécessite une mise à jour mineure (correction d'un ou deux bugs, etc.) mais pas ou peu d'ajouts de fonctionnalités, on incrémente cette valeur.
- **V** : comme pour les packages, version propre à l'éditeur de la distribution.
- **txt** : on rajoute parfois un texte pour donner des précisions sur le noyau. Par exemple, **smp** indique un noyau multiprocesseur.

Enfin sachez qu'il existe de très nombreux ouvrages traitant du noyau Linux, vu sa complexité et sa richesse Il nous sera impossible d'explorer cette partie passionnante de GNU/Linux.



Gestion du noyau

Précédemment (Chapitre système de fichier) nous avons vu que **« /proc » et « /sys » sont des systèmes de fichiers virtuels** contenant des informations sur le **noyau en cours d'exécution**.
La version 2.4 du noyau ne connaît que /proc où toutes les informations sont regroupées.
La version 2.6 du noyau a modifié la fonction **de « /proc » pour déléguer une partie de ses tâches à « /sys »**.
S'agissant de systèmes de fichiers virtuels, ils ne prennent aucune place ni en mémoire, ni sur un disque quelconque. Il ne faut pas se laisser leurrer par la taille des pseudos fichiers ou arborescence contenus dedans. Ne tentez pas de supprimer « **/proc/kcore »** pour gagner de la place !
Tous ces fichiers (ou presque) peuvent être lus et affichés directement.
Les fichiers de **« /proc »** donnent énormément d'informations sur le système :

- interrupts : les paramètres IRQ.
- cpuinfo : détails sur vos processeurs.
- dma : les paramètres DMA.
- ioports : les ports mémoires E/S.
- devices : les périphériques présents
- meminfo : l'état global de la mémoire.
- loadavg : la charge du système.
- uptime : uptime du système, attente.
- version : détails de la version de Linux.
- modules : identique au résultat de lsmod.
- swaps : liste et état des partitions d'échange.
- partitions : liste et état des partitions connues du système.
- mounts : montages des systèmes de fichiers.
- pci : détails du bus PCI.

« /proc » contient également des sous-répertoires qui regroupent des informations par thème.

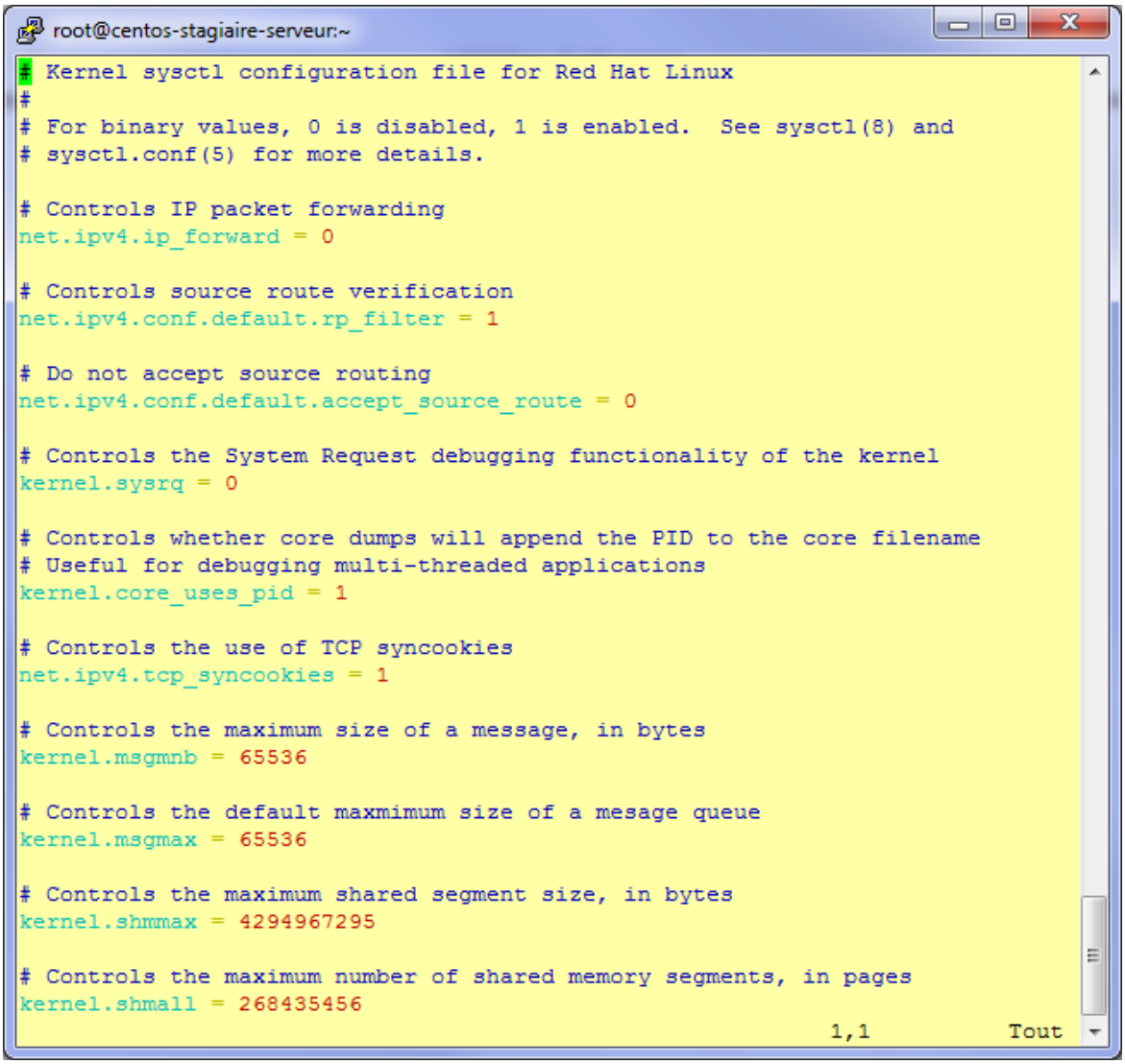
- /proc/scsi : informations sur le bus SCSI.
- /proc/ide : informations sur le bus IDE.
- /proc/net : informations sur le réseau.
- /proc/sys : paramètres et configuration dynamique du noyau.
- /proc/<PID> : informations sur le processus PID.

Certaines entrées des systèmes de fichiers « **/proc/sys » et « /sys »** sont différentes des autres car leur contenu peut être modifié et les modifications sont prises en compte directement par le noyau sans avoir à redémarrer la machine.
Exemple : /proc/sys/net/ipv4/ip_default_ttl

Avec cette méthode les valeurs modifiées ne sont pas enregistrées. En cas de redémarrage il faut recommencer.$

Le fichier « **rc.sysinit »** appelle la commande **« sysctl »** qui agit sur ces paramètres.
Pour que les valeurs de ces systèmes de fichiers virtuels restent permanentes (remises en place à chaque démarrage) il faut modifier le fichier **« /etc/sysctl.conf »**.

Voici ce fichier en version original :



La compilation du noyau

Il s'agit d'une des seules opérations qui nécessite le redémarrage de votre système.

Mise en garde :

N'oubliez pas que personnaliser son noyau (via recompilation) vous sort du circuit officiellement supporté par votre distribution.
A chaque mise à jour du noyau via YUM (ou autre) vous perdrez votre paramétrage personnel … vous devrez recompiler votre noyau avec les nouvelles sources.

Nous déconseillons, dans 99% des cas, de recompiler votre noyau. Les gains de performances, fonctionnalités sont vraiment anecdotiques par rapport à toute la souplesse que vous avez avec un noyau officiellement supporté !

Cela étant, voici les étapes à suivre pour une distribution Centos :

http://wiki.centos.org/HowTos/Custom_Kernel

Gestion des modules

Le noyau est composé d'une **partie statique** à laquelle on peut **dynamiquement greffer des modules**.
La partie statique est utilisée lors du démarrage du système et sera toujours chargée en mémoire vive, tandis que les **modules peuvent être chargés à la demande**.
Les modules peuvent assurer la gestion de périphériques, de système de fichiers etc., on peut faire une analogie avec les pilotes matériels.
Ce sont des fichiers dont l'extension est « **.ko » ils sont stockés dans « /lib/modules/$(uname -r) ».**

Vous pouvez lister les modules actuellement chargés avec la commande **« lsmod »** :

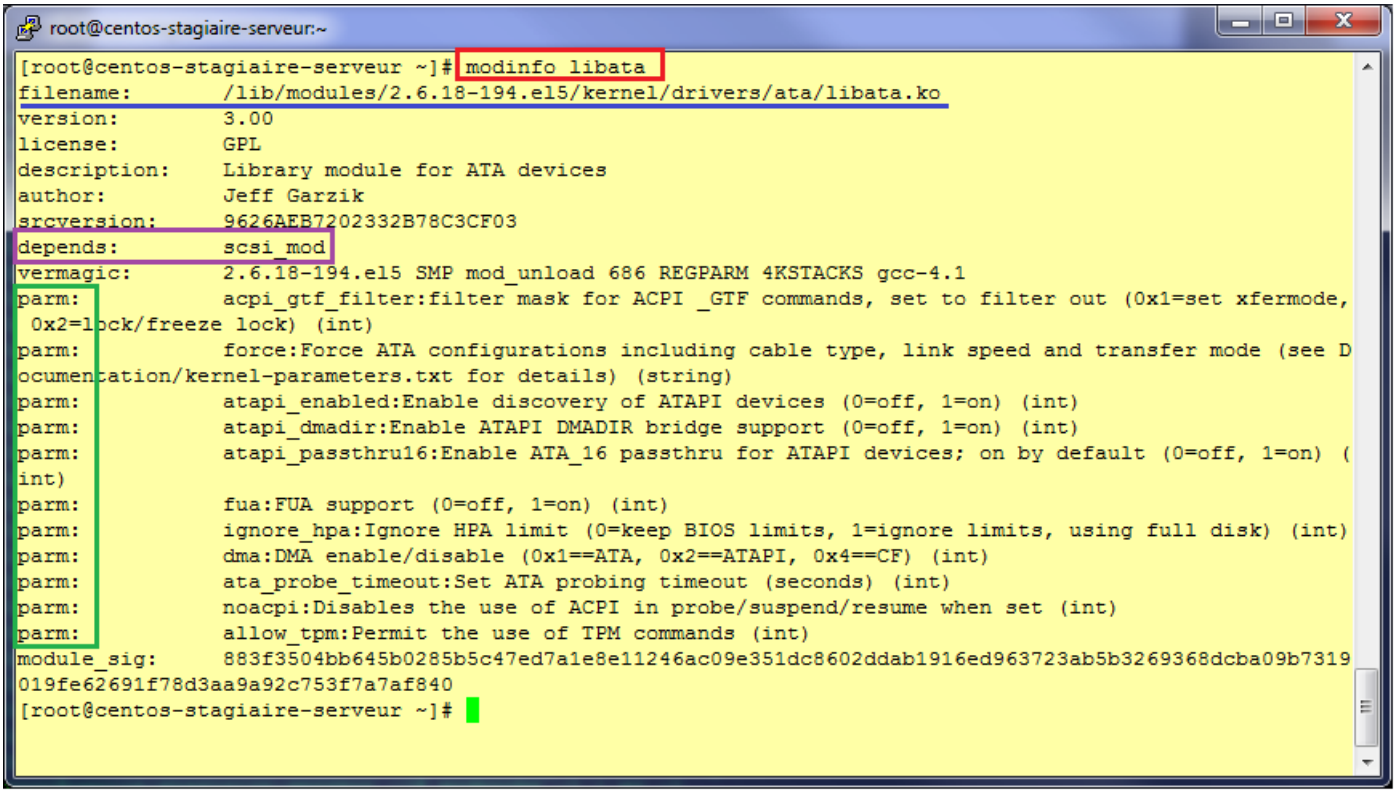
```
root@centos-stagiaire-serveur:~
[root@centos-stagiaire-serveur ~]# lsmod
Module                  Size  Used by
autofs4                 29253  3
hidp                    23105  2
rfcomm                  42457  0
l2cap                   29505  10 hidp,rfcomm
bluetooth               53925  5 hidp,rfcomm,l2cap
lockd                   63337  0
sunrpc                 146685  2 lockd
loop                    18761  0
dm_multipath            25421  0
scsi_dh                 12097  1 dm_multipath
video                   21192  0
backlight               10049  1 video
sbs                     18533  0
power_meter             16461  0
hwmon                    7365  1 power_meter
i2c_ec                   9025  1 sbs
dell_wmi                 8401  0
wmi                     12137  1 dell_wmi
button                  10705  0
battery                 13637  0
asus_acpi               19289  0
ac                       9157  0
ipv6                   270305  28
xfrm_nalgo              13381  1 ipv6
crypto_api              12609  1 xfrm_nalgo
lp                      15849  0
sg                      36573  0
snd_intel8x0            35421  0
snd_ac97_codec          93025  1 snd_intel8x0
ac97_bus                 6337  1 snd_ac97_codec
snd_seq_dummy            7877  0
snd_seq_oss             32577  0
snd_seq_midi_event      11073  1 snd_seq_oss
snd_seq                 49585  5 snd_seq_dummy,snd_seq_oss,snd_seq_midi_event
snd_seq_device          11725  3 snd_seq_dummy,snd_seq_oss,snd_seq
snd_pcm_oss             42817  0
snd_mixer_oss           19009  1 snd_pcm_oss
snd_pcm                 72133  3 snd_intel8x0,snd_ac97_codec,snd_pcm_oss
e1000                  118997  0
snd_timer               24517  2 snd_seq,snd_pcm
parport_pc              29157  0
parport                 37513  2 lp,parport_pc
ide_cd                  40161  0
snd                     55749  9 snd_intel8x0,snd_ac97_codec,snd_seq_oss,snd_seq,snd_seq_device,snd_pcm_oss,s
cdrom                   36577  1 ide_cd
pcspkr                   7104  0
i2c_piix4               13133  0
serio_raw               10693  0
i2c_core                24001  2 i2c_ec,i2c_piix4
soundcore               11553  1 snd
snd_page_alloc          14281  2 snd_intel8x0,snd_pcm
dm_raid45               67145  0
dm_message               6977  1 dm_raid45
dm_region_hash          15681  1 dm_raid45
```

Décodage de **« lsmod »**

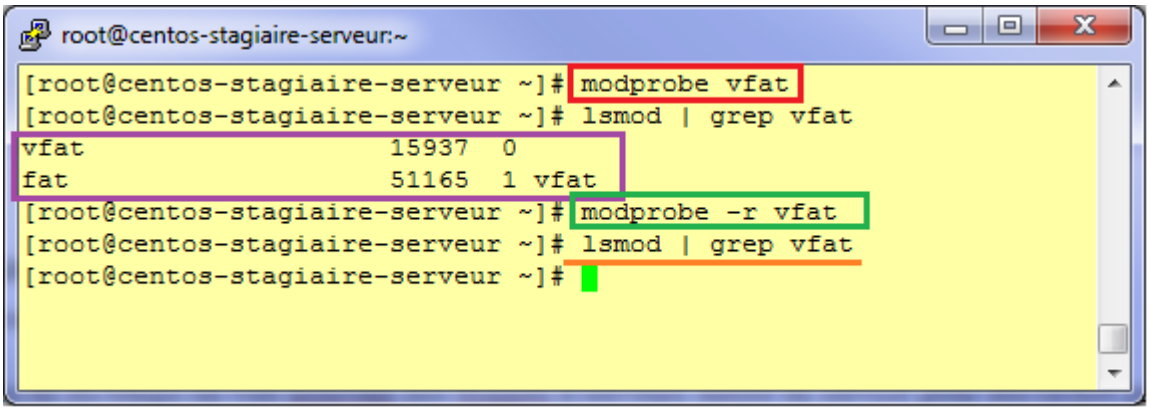
Champ	Signification
Module	Nom du module (.ko) chargé
Size	Taille du module en octets
Used	Nombre d'élément du système accédant au module
By	Liste les autres modules utilisant ce module

Une autre commande intéressante est **« depmod »**, elle permet de mettre à jour l'arbre des dépendances entre les modules en modifiant le **fichier « /lib/modules/$(uname -r)/modules.dep »**.

La commande **« modinfo »** fournit des informations (dépendance, fichier .ko, paramètre etc.) concernant un module particulier :

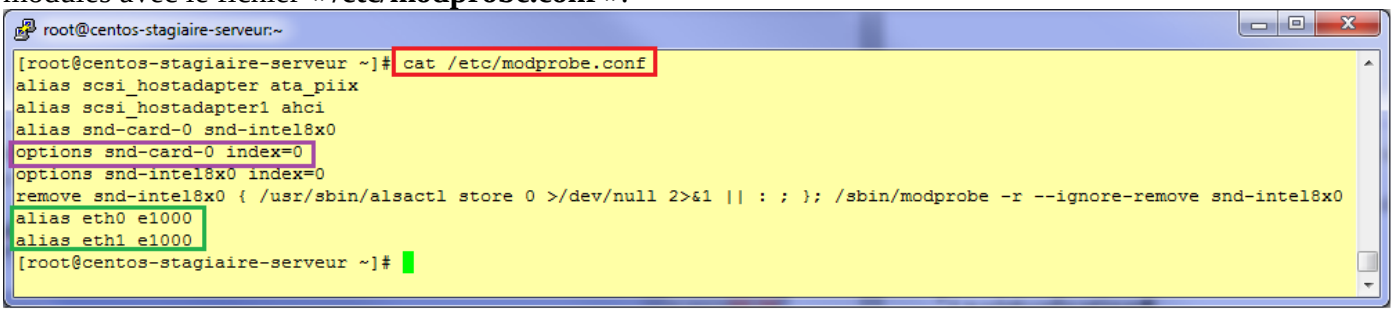


La commande **« modprobe »**, permet de charger ou décharger un module intelligemment. Cela veut dire que tout comme **YUM** elle va gérer les dépendances entre modules et prendre en compte les paramètres du fichier **« /etc/modprobe.conf »** etc,
A l'opposé, les commandes « **insmod/rmmod** » ne se concentrent que sur le module que vous souhaitez manipuler et donc ne gèrent pas les dépendances entres modules.
Préférez **« modprobe ».**



Pour comparer amusez-vous à charger et décharger le même module (**vfat**) avec « insmod » et « rmmod ».

Pour finir ce chapitre sur la gestion des modules signalons à titre indicatif que vous pouvez configurer vos modules avec le fichier **« /etc/modprobe.conf »**.



Mais attention **« udev »** gère automatiquement votre matériel sans intervention spéciale de votre part.

Néanmoins si vous avez besoin de gérer vous-même un module, dès le démarrage.
Centos vérifie l'existence du fichier **« /etc/rc.modules »** au démarrage, celui-ci contient différentes commandes pour charger les modules. **« rc.modules »**. Ce fichier est exécuté avant le processus de démarrage.

Par exemple, les commandes suivantes configurent le chargement du module « **foo** » au démarrage (en tant que root) :

```
➢ echo modprobe foo >> /etc/rc.modules
  chmod +x /etc/rc.modules
```

Pour avoir plus de détail concernant la gestion des modules sous Centos :
http://www.centos.org/docs/5/html/5.2/Deployment_Guide/ch-modules.html sur internet

Cette même documentation est disponible au **chapitre 32, en français**, sur votre installation Centos :
/usr/share/doc/Deployment_Guide-fr-FR-5.2/index.html

La virtualisation

Virtualiser une mode ?

Au fil du temps on voit souvent des flopés de serveurs entrer en service pour des tâches parfois très peu consommatrice de ressources.
Ou alors des acquisitions de matériels surdimensionnés par rapport aux services attendus.

En plus de couter cher à l'achat ce genre de stratégie fini par miner les budgets en ce qui concerne :

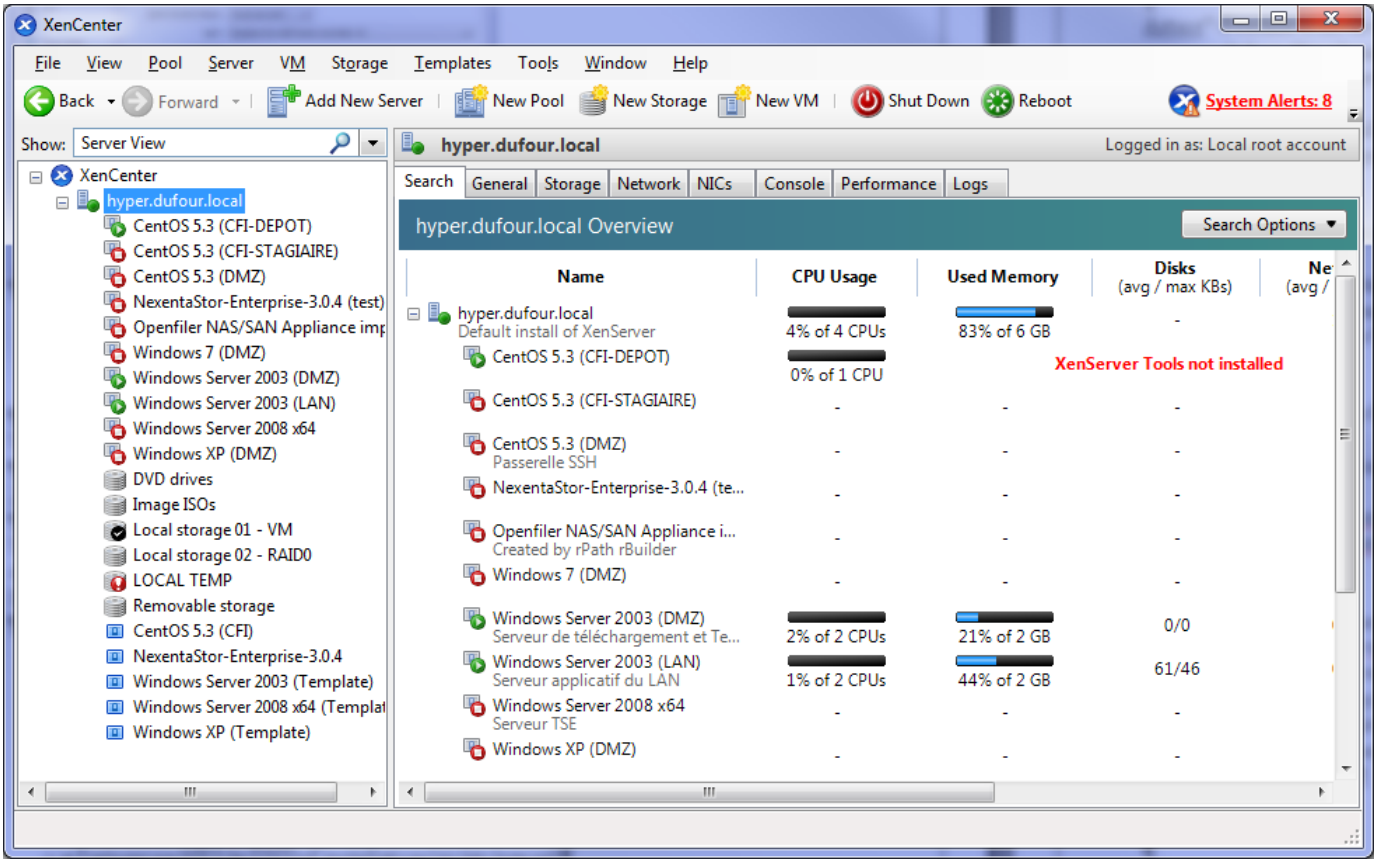
- la maintenance,
- l'administration.

De plus certaines problématiques se trouvent écartées avec la virtualisation.

Par exemple la plupart des grosses structures informatique qui travaillent en environnement virtualisé déploient de moins en moins leurs systèmes d'exploitations par :

- Installation manuelle,
- Scripting,
- Clonage comme nous l'avons vu précédemment.
- Etc.

Elles préfèrent plutôt s'appuyer sur des modèles de machines virtuelles (VM Template) déjà tout fait.
En un clic on déploie des serveurs ou des postes client.
Parfois cela va même plus loin : les déploiements de VM sont automatisés en fonction de la charge système demandés par les clients.



Le marché

Voici les principaux acteurs actuels pour la partie IT :

- Citrix : **XenServer**
- VMware : **ESXi, Infrastructure**
- Microsoft : **Hyper-V**
- Xen : **XenSource**

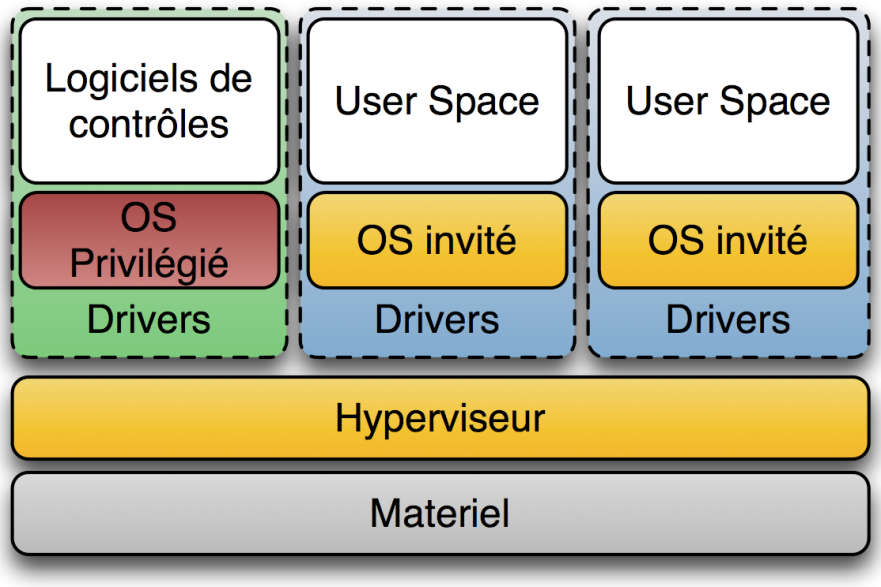
Qu'est-ce qu'un hyperviseur (Wikipédia)

L'hyperviseur de type 1

Un hyperviseur de **Type 1** (ou **natif**) est un logiciel qui s'exécute directement sur une plateforme matérielle donnée (comme *outil de contrôle* de système d'exploitation).
Un **système d'exploitation secondaire** peut de ce fait être exécuté au-dessus du hardware.
L'hyperviseur type 1 est un noyau hôte allégé et optimisé pour ne faire tourner initialement que des noyaux d'OS invités adaptés et optimisés pour tourner sur cette architecture spécifique, les OS invités ayant conscience d'être virtualisés.
Sur des processeurs ayant les instructions de virtualisation matérielle (AMD-V et Intel-VT), l'OS invité n'a plus besoin d'être modifié pour pouvoir être exécuté dans un hyperviseur de type 1.

Un hyperviseur de type 1 classique est CP/CMS, développé par IBM dans les années 60 et ancêtre de z/VM.

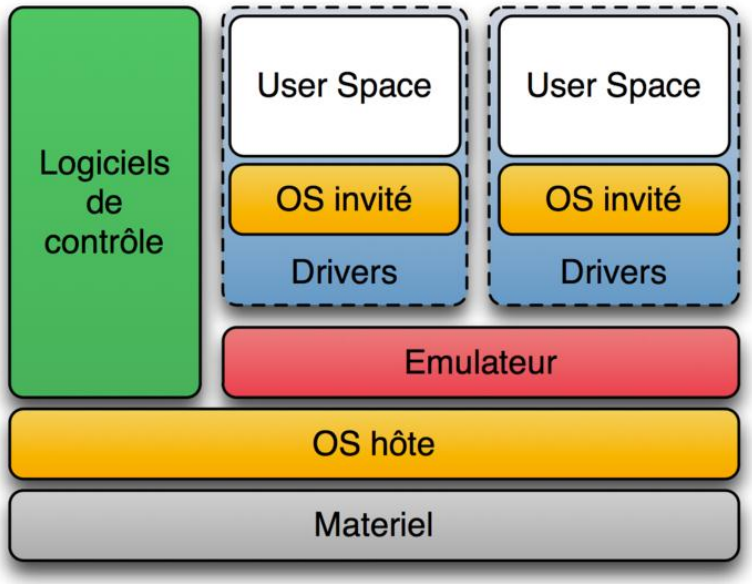
Des exemples d'hyperviseurs plus récents sont Xen, Oracle VM, ESX Server de VMware, TRANGO, l'hyperviseur LPAR de IBM (PR/SM), Hyper-V de Microsoft, Polyxene de Bertin, l'hyperviseur Logical Domains de SUN (sorti en 2005), Proxmox... Une légère variation consiste à intégrer l'hyperviseur dans le firmware de la plateforme. C'est ce qui a été fait dans le cas de l'hyperviseur Virtage d'Hitachi. Les machines virtuelles de noyau, qui transforment un noyau Linux complet en hyperviseur, sont également considérées comme hyperviseurs de type 1.



L'hyperviseur de type 2

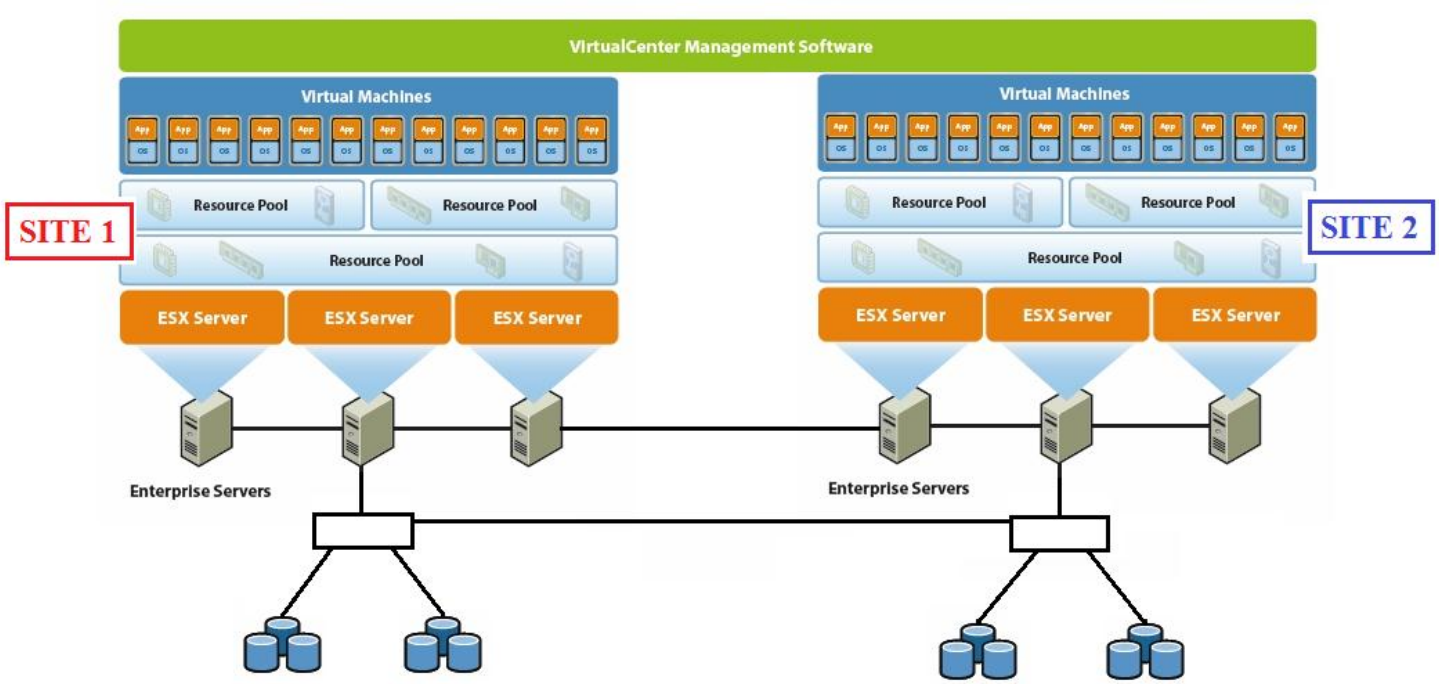
Un hyperviseur de **Type 2** est un logiciel qui s'exécute à l'intérieur d'un autre système d'exploitation.
Un système d'exploitation invité s'exécutera donc en troisième niveau au-dessus du hardware (matériel). Les OS invités n'ont pas conscience d'être virtualisés, il n'a donc pas besoin d'être adapté.

Quelques exemples de tels hyperviseurs sont VMware Server (anciennement connu sous le nom gsx), VMware Workstation, VMware Fusion, l'hyperviseur open source QEMU, Microsoft Virtual PC et les produits Microsoft VirtualServer , VirtualBox d'Oracle, de même que Parallels Workstation de SWsoft et Parallels Desktop.
Le terme hyperviseur prend sa source dans la ré implémentation par IBM de CP-67 pour le système d'exploitation System/370 sorti en 1972 sous le nom VM/370. Le terme **appel hyperviseur** ou **hypervisor call**, ou **hypercall**, fait référence à l'interface de paravirtualisation , par laquelle un système d'exploitation "invité" accède directement à des services à travers le logiciel de contrôle de niveau élévé – (Le terme "superviseur" fait référence au Noyau du système d'exploitation qui sur les mainframes IBM s'exécute en *mode Superviseur*.)



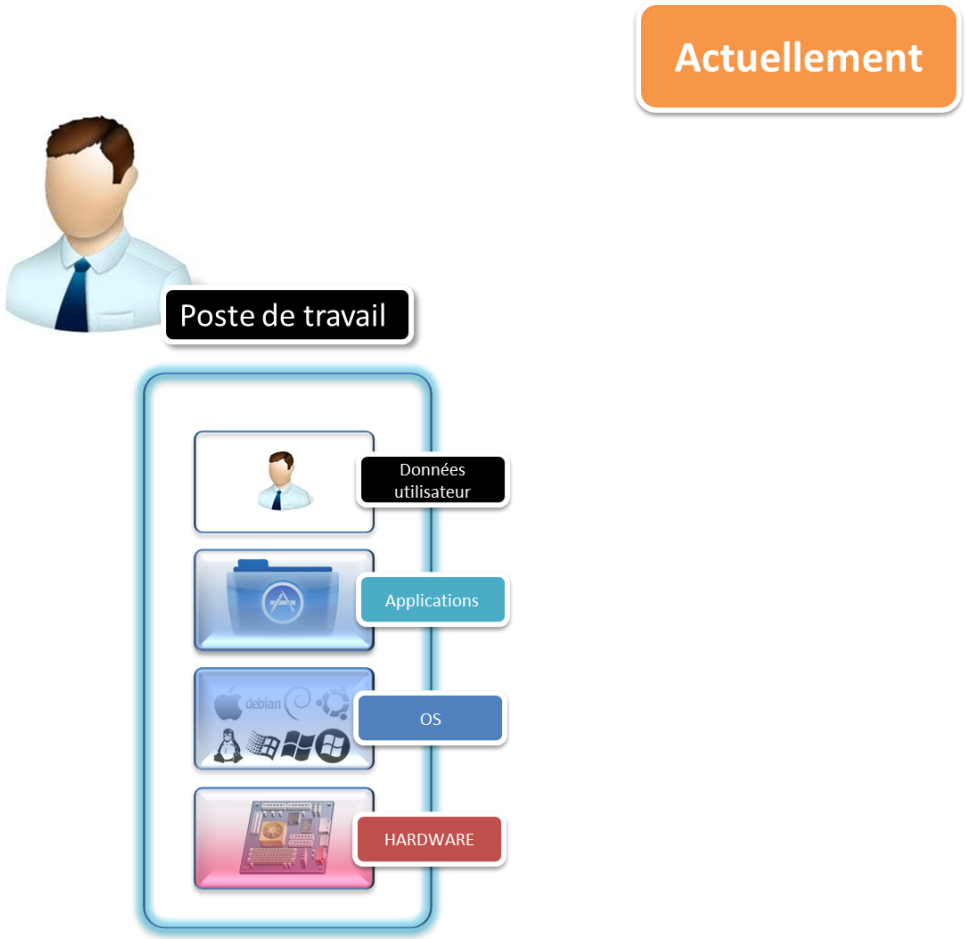
Architecture virtualisée

Exemple (schéma VMware)



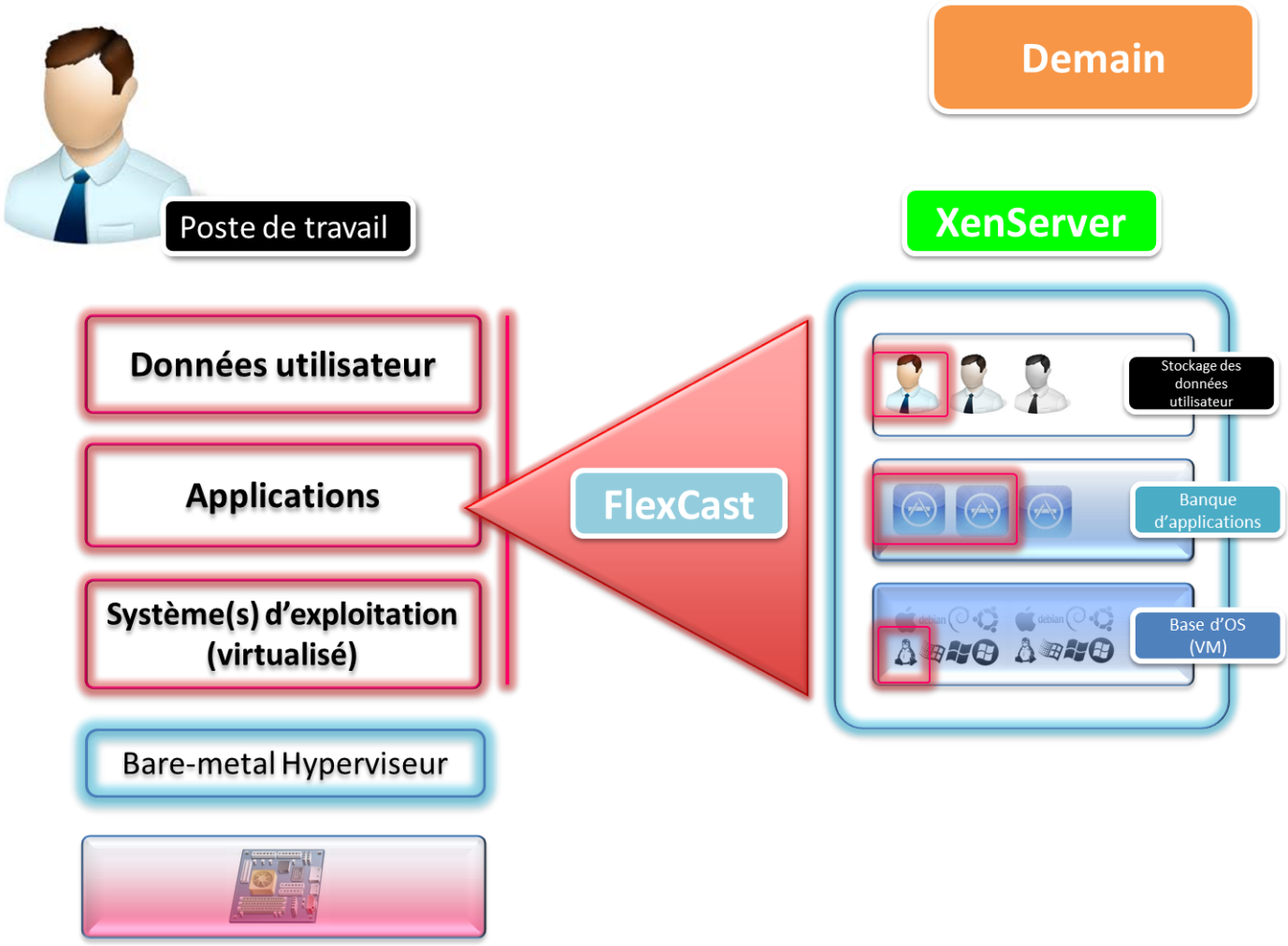
Virtualisation du poste de travail

Voici comment actuellement nous concevons le poste de travail …



Avec bien sûr une connexion au réseau, des serveurs de fichiers, d'applications etc.

Voici ce à quoi il pourrait ressembler demain, avec les techniques de virtualisation du poste de travail :



Quelques explications s'imposent. A ce niveau nous n'avons plus de système d'exploitation à proprement parlé installé sur le poste de travail mais une ou plusieurs machines.
Ces dernières sont stockées sous forme de fichier de machine(s) virtuelle(s)s sur le poste de travail (Virtual Hard Drive : vhd pour XenClient).
En cas de déconnexion du réseau ou d'un poste de travail mobile l'utilisateur est complétement autonome.

Si l'administrateur système décide de déployer un nouveau système d'exploitation, une mise à jour, un correctif etc … en quelques opérations il peut redéployer la ou les images de machine(s) virtuelle(s) sur le poste de travail.
Mieux la technologie FlexCast mise au point par Citrix permet de ne descendre sur le poste de travail que le delta de l'image qui a été modifiée !

Mais plus fort, imaginez une attaque virale de grande ampleur. Il est tout à fait possible d'imaginer un scénario ou vous pouvez massivement, sans vous déplacez, redéployer la partie de votre parc de station de travail qui a été infecté.

Les parties OS, Données et applications étant parfaitement séparées par les technologies de virtualisation.

VMware propose également ce concept « VMView » mais est beaucoup moins en avance que le produit « XenClient » (couplé à FlexCast) le produit de Citrix