Linux

Activer alias ll pour ls -l

Vous avez l'habitude de faire un « ll » (double L) pour lister le contenu d'un répertoire mais cette machine Linux ne comprend pas cette simple commande ? Que ce soit sur Ubuntu, Debian, Fedora, Red Hat ou autre CentOS, les alias de la commande universelle « ls » ne sont pas forcément interprétés de la même façon.

```
user@ubuntu:~$ ll
ll : commande introuvable
```

linux alias ll failed

Le simple « ls » liste en ligne alors qu'une présentation plus claire existe en colonne grâce à « ls -l » ainsi qu'en affichant les fichiers cachés par « ls -la » : facile à mémoriser mais long à taper lorsque l'on utilise le Terminal du matin au soir. Il existe une manière d'accélérer ça en utilisant un alias à configurer sur le profil de l'utilisateur Linux. Ce tutoriel explique comment **utiliser la commande ll à la place de ls -l**. Ce guide n'est pas qu'à destination des experts Linux, un développeur d'applications ou un simple utilisateur qui manipulent un peu le Terminal ont fréquemment besoin de liste le contenu de répertoires, que ce soit sur le disque dur local ou sur un partage réseau (serveur, NAS…).

Les versions récentes de la distribution Ubuntu activent par défaut les alias ll, la et l mais une Debian ne le fera par exemple pas. C'est ainsi que l'on peut activer le ou les alias qu'on a l'habitude d'utiliser sur d'autres distribs Linux.

Configurer l'alias ll pour ls -la

1. Ouvrir un Terminal ou une connexion distante par SSH sur la machine Linux.

2. Aller dans le dossier utilisateur :

```
cd /home/votreuser
```

3. Editer le fichier caché .bashrc avec vi ou un autre éditeur de texte (avec un *sudo*, si besoin) :

```
nano .bashrc
```

4. Localiser ce groupe :

```
# some more ls aliases
#alias ll='ls -l'
#alias la='ls -A'
```

```
#alias l='ls -CF'
```

5. Décommenter, c’est-à-dire supprimer le symbole dièse devant la ligne « alias ll=’ls -l’ :

```
# some more ls aliases
alias ll='ls -l'
#alias la='ls -A'
#alias l='ls -CF'
```

6. Enregistrer les modifications.

7. Lors de la prochaine connexion SSH ou avec un Terminal local, valider le fonctionnement de « ll » comme raccourci de « ls -l »

Ajouter un utilisateur sudo

La commande sudo est un programme conçu pour permettre aux utilisateurs d'exécuter des commandes avec les privilèges de sécurité d'un autre utilisateur. Par défaut, l'utilisateur root

Dans ce guide, nous allons vous montrer comment créer un nouvel utilisateur sur un système Debian et lui donner un accès sudo

Vous pouvez utiliser ce compte utilisateur pour exécuter des commandes admins sans vous connecter à votre serveur Debian en tant qu'utilisateur root.

Créer un utilisateur Sudo

Suivez les étapes ci-dessous pour créer un nouveau compte utilisateur et lui donner un accès sudo. Si vous souhaitez configurer sudo pour un utilisateur existant, passez à l'étape 3.

1. Se connecter à la machine Linux

Se connecter en premier lieu avec le compte root de la machine

2. Création d'un compte utilisateur

Créer un utilsateur en utilisant la commande `adduser`. Ne pas oublier de remplacer `username` par le nom d'utilisateur à créer :

```
adduser username
```

Le terminal vous demande de taper le mot de passe pour le nouvel utilisateur et de confirmer. Pensez à créer un mot de passe suffisamment sécurisé (combinaison de lettres, chiffres et caractères spéciaux).

```
Adding user `username' ...
Adding new group `username' (1001) ...
Adding new user `username' (1001) with group `username' ...
Creating home directory `/home/username' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
```

Une fois que vous avez défini le mot de passe pour l'utilisateur, la commande créée un répertoire de base pour l'utilisateur, copie plusieurs fichiers de configuration dans le répertoire de base et vous invite à définir les informations du nouvel utilisateur. Si vous souhaitez laisser toutes ces informations vides, appuyez simplement sur `ENTER` pour tout accepter sans rien remplir.

```
Changing the user information for username
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n]
```

3. Ajouter l'utilisateur au groupe sudo

Par défaut, sur les systèmes Debian, les membres du groupe sudo ont un accès sudo. Pour ajouter un utilisateur au groupe sudo, utilisez la commande `usermod` .
Remplacer username par le nom d'utilisateur créé plus haut :

```
usermod -aG sudo username
```

Tester l'accès sudo

Basculer sur le compte utilisateur créé :

```
su - username
```

Utiliser la commande sudo `whoami` pour savoir si vous êtes connecté avec le bon compte :

```
sudo whoami
```

Si l'utilisateur a un accès sudo, le résultat de la commande `whoami` sera `root`:

```
root
```

Le cas où sudo n'est pas installé.

Dans cette partie nous allons voir le cas ou sudo n'est pas installé par défaut. C'est le cas par exemple sur la version cd de Debian.

I L'installation de sudo en fonction de l'OS.

Dans le cas d'un debian ou ubuntu :

```
apt-get install sudo
```

Dans le cas d'un red hat ou CentOS :

```
yum install sudo
```

II Ajout de droits sur un utilisateur :

Dans ce cas, la création d'un utilisateur est nécessaire. Il faut donc faire en sorte d'avoir les droits root.

```
adduser <nomuser>
su #se mettre en root
```

On va donc ensuite modifier le fichier qui modifie les droits sudo via cette commande :

```
visudo
```

Conclusion

Vous avez appris à créer un utilisateur avec les privilèges sudo. Vous pouvez maintenant vous connecter à votre serveur Debian avec ce compte utilisateur et utiliser sudo pour exécuter des commandes admins

DDClient - DynDns Client

Automatiser la gestion des adresses IP dynamiques

ddclient permet d'automatiser la mise à jour d'une adresse IP dynamique reliée à un nom de domaine (de 213.95.41.11 à www.ubuntu-fr.org par exemple). Ce système est extrêmement pratique quand il s'agit de créer un serveur web personnel accessible rapidement *via* un nom de domaine, alors que votre adresse IP change régulièrement.

Pré-requis

- S'être inscrit sur l'un des services pris en charge. Pour les connaître, saisissez dans un terminal la commande suivante :

  ```
  ddclient --help | grep "o '"
  ```

- Disposer des droits d'administration.
- Disposer d'une connexion à Internet configurée et activée.
- Avoir activé l'accès aux dépôts Universe et Multiverse.

Installation

Installez le paquet **ddclient**.

```
sudo apt install -y ddclient
```

La touche espace sert à sélectionner la redirection "[] Nomduhost.dyndns.com"

Configuration

Commencez par générer le fichier de configuration en tapant :

```
sudo dpkg-reconfigure ddclient
```

Tout d'abord, vous devrez spécifier le service à utiliser. Le ou les noms de domaines qui pointeront vers l'adresse IP dynamique. L'identifiant et le mot de passe avec lesquels vous vous êtes inscrit au service.

ddclient peut prendre connaissance de l'adresse IP dynamique de diverses manières :

Paramètre	Signification
web	à partir d'une page web
if	par une interface réseau (ex : ppp0, eth0...)
fw	en interrogeant le routeur
cmd	en exécutant une commande

Si vous avez choisi une autre option que « web », il sera nécessaire de modifier le fichier « /etc/ddclient.conf » par la suite afin de spécifier l'interface, l'adresse IP du routeur ou la commande à employer.

Si vous êtes derrière un routeur, ce n'est pas l'ordinateur qui initiera la connexion. Par conséquent, répondez « Non » à la question « Faut-il lancer ddclient lors de la connexion PPP ? ».

L'avant-dernière question vous propose de faire tourner ddclient en tâche de fond (daemon). Puisque le but est d'automatiser la mise à jour de l'adresse IP, répondez « Oui ».

Enfin, vous pouvez définir un délai entre les vérifications. Par défaut, celles-ci s'effectueront toutes les 300 secondes (5 minutes).

ddclient.conf

Maintenant que vous avez répondu aux questions, nous pouvons modifier le fichier « /etc/ddclient.conf » afin d'affiner les réglages.

Celui-ci se présente sous cette forme :

```
pid=/var/run/ddclient.pid
ssl=yes
protocol=dyndns2            #ou zoneedit1 si vous utilisez zoneedit
use=web
server=members.dyndns.org   #ou dynamic.zoneedit.com si vous utilisez zoneedit
```

```
login=identifiant
password='motdepasse'
nom.de.domaine              #si plusieurs nom de domaine, separez par ','
```

Exemple de configuration DynDns OVH :

```
protocol=dyndns2

use=web
server=www.ovh.com
login=my-domain.com-user
password='passworddemy-domain.com-user'
subdomain.my-domain.com

use=web
server=www.ovh.com
login=my-domain.com-user1
password='passworddemy-domain.com-user1'
subdomain1.my-domain.com

# ...
```

Si vous utilisez l'option « `ssl=yes` », veillez à ce que le paquet libio-socket-ssl-perl soit installé.

Votre version de ddclient doit être supérieure à 3.7.0 (cf. le site de DynDNS).

Option	Signification
ssl	connexion sécurisé lors de l'échange avec DynDNS
protocol	service utilisé pour effectuer la liaison
use	comment identifier l'adresse IP
server	serveur utilisé pour effectuer la mise à jour
login	l'identifiant pour le service
password	le mot de passe pour le service
wildcard	Définit si les adresses du type *.nom.de.domaine fonctionnerons

Intéressons nous à l'option **use**. Celle-ci accepte plusieurs paramètres :

Paramètre	Signification

web	donne l'adresse IP à partir d'une page web (par défaut)
cmd	donne l'adresse IP en exécutant une commande
fw	donne l'adresse IP en interrogeant un routeur
if	donne l'adresse IP à partir d'une interface réseau (ex. : ppp0, eth0, etc.)
ip	donne l'adresse IP à partir d'une adresse IP

Pour utiliser la plupart de ces paramètres, il vous faudra rajouter quelques lignes au fichier de configuration.

Exemple d'obtention à partir d'une interface réseau :

```
use=if
if=ppp0
```

Exemple d'interrogation d'un routeur :

```
use=fw
fw=ip_du_routeur
fw-login=identifiant_du_routeur
fw-password=motdepasse_du_routeur
```

Par défaut, ddclient récupère la première adresse IP qu'il trouve. Lorsque l'option **fw** est utilisée, il est possible que votre routeur ne fournisse pas la bonne. Pour palier ce problème, vous avez la possibilité de spécifier directement le modèle du routeur. Pour obtenir la liste des routeurs pris en charge par la version de ddclient installée, saisissez la commande suivante :

```
ddclient --help | grep use=
```

Pour vérifier que la configuration fonctionne:

```
sudo ddclient -daemon=0 -debug -verbose -noquiet
```

ddclient utilise un fichier cache en local afin d'éviter de trop solliciter le serveur (certains serveurs limitent la périodicité des mises à jour). On peut alors voir apparaître le message **skipped: IP address was already set to xxx.xxx.xxx.xxx**
Il est alors nécessaire de supprimer le cache local, afin de forcer ddclient à mettre à jour le serveur :

```
sudo rm /var/cache/ddclient/ddclient.cache
```

Exemple de configuration pour le service noip (fichier ddclient.conf) :

```
ssl=yes
protocol=noip
use=web, web=checkip.dyndns.com/, web-skip='Current IP Address'
server=dynupdate.no-ip.com
login=votrelogin
password='motdepasse'
votreserveur.no-ip.biz
```

Ajout d'un enregistrement MX à votredomaine.dyndns.org

→ Voir l'article Wikipédia pour la définition et son utilité.

L'astuce ici est de créer un second enregistrement sur votre compte *dyndns.org* du style **mxvotredomaine.dyndns.org** et qui sera également mis à jour par le démon.

Après avoir créé **mxvotredomaine.dyndns.org**, il vous faut modifier le fichier **/etc/ddclient.conf** de la sorte :

```
pid=/var/run/ddclient.pid
protocol=dyndns2
use=web
#use=if; if=web
server=members.dyndns.org
login=votre_login
password='votre_password'
mx=mxvotredomaine.dyndns.org    ← Adresse de l'enregistrement MX
backupmx=no                     ← Votre enregistrement MX est prioritaire
votredomaine.dyndns.org,mxvotredomaine.dyndns.org    ← Mise à jour de l'adresse MX
```

Rechargez ensuite **ddclient** :

```
sudo /etc/init.d/ddclient force-reload
```

Réglage complémentaire avec FreeDNS

Si vous utilisez un sous-domaine fourni par FreeDNS, il se peut que vous soyez confronté à ce message d'erreur :

> FATAL: Error loading the Perl module Digest::SHA1 needed for freedns update.
> FATAL: On Debian, the package libdigest-sha1-perl must be installed.

En attendant que le problème soit corrigé, il est possible de modifier le fichier **/usr/sbin/ddclient** et de remplacer « `require Digest::SHA1` » par « `require Digest::SHA` » et « `import Digest::SHA1` » par « `import Digest::SHA` ».

Source : *Ask Ubuntu : How can I get ddclient to work with FreeDNS?*

Désinstallation

Pour supprimer cette application, il suffit de supprimer son paquet. La configuration de l'application sera conservée ou supprimée selon la méthode de désinstallation que vous choisirez.

Grub

Apprenez à maitriser Grub !

Déjà, qu'est ce que Grub ?

Derrière ce nom barbare se cache un programme très utile aux utilisateurs de Linux.
En vérité Grub est ce qu'on appelle un chargeur d'amorçage, ou (pour les anglophones *bootloader*).
L'acronyme signifie **the GRand Unified Bootloader** ou littéralement **le Grand Chargeur Unifié**.

Je sais, ça ne veux rien dire ^^ mais ne vous inquiétez pas, nous allons clarifier ça tout au long de ce tutoriel.

- Note : le tuto était au départ destiné à la première version de Grub, mais une nouvelle version étant apparue entre temps, j'ai dû adapter le tuto pour les faire cohabiter.
  Ce tutoriel va donc vous proposer de configurer aussi bien Grub que Grub2 (Grub-pc).
  Ainsi, si vous voyez une mauvaise référence à l'ancienne version ou une manipulation qui n'est plus à jour (malgré le soin de mes petites mains et de celles de toute l'équipe), merci de me le signaler ! ;)

Histoire et fonctionnement de Grub

Parlons tout d'abords du bien connu projet GNU.

Sa création

Le projet GNU a été créé par Richard Stallman en 1984 alors qu'il travaillait au MIT (au laboratoire d'intelligence artificielle). Il souhaitait selon ses dires créer un système d'exploitation libre et complet pour "ramener l'esprit de coopération qui prévalait dans la communauté informatique dans les jours anciens".

La mascotte de GNU est un gnou (en raison de sa prononciation anglophone).

Image utilisateur

Le terme GNU est un acronyme récursif pour Gnu's Not Unix (Gnu N'est pas Unix).
Il faut savoir qu'au moment de la création du projet GNU, le système UNIX était déjà bien lancé et reconnu par les informaticiens. GNU fut donc créé dans le but d'être compatible avec ce système.

En 1985, Stallman fonde la Free Software Foundation, structure logistique, légale et financière du projet GNU.
La FSF finance le développement du projet GNU et ce sont pour la plupart des communautés étudiantes américaines qui ont rendu ce projet viable. Le projet GNU se développe rapidement si bien que plusieurs sociétés contribuent au projet GNU en revendant ses logiciels, ou en offrant un support technique.

En 1990, GNU dispose de son propre éditeur de texte (Emacs), de son compilateur (GCC) et de la plupart des bibliothèques dont dispose Unix. Mais il manque encore un noyau.

Les systèmes 100% GNU sont des utopies mais il existe certains systèmes s'en approchant (comme Debian).
La totalité des distributions de Linux sont plus ou moins apparentées au projet GNU au point que Richard Stallman défendait l'appellation *distribution GNU/Linux*.

A ce stade, il manquait encore une pièce du puzzle : le noyau, et c'est ici que Linux fait son apparition.

Hé ! Je suis là pour vous parler de Grub pas de GNU ;)

L'arrivée de Grub

Grub est un programme de multiboot libre et gratuit.

Ce genre de programme se nomme **bootloader***ou***chargeur d'amorçage**. Il permet de choisir entre plusieurs systèmes d'exploitation (*OS*) sur une seul machine.

Grub a été créé en 1995 par Erich Boleyn à l'université de l'Utah. Avec l'aide de Brian FORD, il a créé un lanceur de multiboot. Erich a compris qu'il aurait plus vite fait d'écrire son propre chargeur plutôt que d'en modifier un autre.

Grub était né !

Erich a ajouté de nombreuses fonctions à GRUB, mais d'autres priorités l'ont empêché de suivre les demandes du nombre croissant d'utilisateurs. En 1999, Gordon Matzigkeit et Yoshinori K. Okuji ont adopté GRUB comme un projet GNU officiel.

Sa principale fonction est de pouvoir supporter la spécification multiboot.
Ses concepteurs voulait qu'il soit en même temps :

- Simple d'utilisation ;
- Fonctionnel pour les experts et les concepteurs de noyau ;
- Qu'il soit compatible avec les systèmes du moment.

Une chose s'avère être intéressante : Grub dispose d'une interface menu lors de son chargement.
On peut y ajouter autant d'entrées que l'on souhaite.
Il n'y a pas de limite, et, de plus on n'est pas forcé de mettre des "liens" vers un système : rien n'empêche au programmeur mégalomane d'insérer une ligne comme "Bienvenue Monsieur."

Grub possède une interface de commande. Elle s'affiche automatiquement si il manque un fichier de configuration ou si il y a une erreur.

Contrairement à LILO, GRUB n'a pas besoin d'être réinstallé pour mettre à jour sa configuration.
Une simple commande permet sa mise à jour `sudo update-grub`

GRUB est par ailleurs très flexible : il permet de charger aussi bien des systèmes compatibles avec le multiboot que des systèmes non-compatibles avec cette fonction (comme Microsoft Windows). Il supporte en outre beaucoup de systèmes de fichiers comme ext3, VFAT ou NTFS. GRUB est également compatible avec le mode Logical Block Address (anglais et pas très utile ici ^^).

Le boot

Venons en au boot.

What ?

Héhé :p

Citation : Wikipédia

> En informatique, le mot boot (apocope du mot anglais bootstrap, nom qui désigne la languette des chaussures pour pouvoir les enfiler plus facilement, voir la partie Historique de l'article pour plus de précisions) désigne la procédure de démarrage d'un ordinateur. En français son usage est un anglicisme, il peut être employé de façon synonyme avec le mot amorçage (du verbe amorcer) de

> même que le mot bootable est synonyme d'amorçable. Initialement lorsque les ordinateurs n'avaient pas encore de disque dur, cette procédure nécessitait l'usage d'une disquette souple de 5"1/4 ; toutefois actuellement ce terme est également employé lorsqu'il s'agit d'un autre médium comme un CD, un DVD une clef USB, ou encore un accès réseau.
>
> On distingue :
>
> * le « boot à froid » (cold boot), obtenu en allumant la machine, ou en l'éteignant puis en la rallumant ;
> * du « boot à chaud » (warm boot), ou «re-boot», obtenu en redémarrant. L'option est présente au niveau du système d'exploitation.

Derrière ce pavé se cache une notion fondamentale de l'informatique. Tout d'abord, en anglais, TO BOOT signifie INITIALISER.
En informatique, la procédure de boot est le démarrage de l'ordinateur en "passant la main" à un système d'exploitation.

A savoir : on ne boot pas forcément sur un système d'exploitation, on peut aussi booter sur un CD ou un DVD *ou pour les ordinateurs les plus récents* le boot sur un support USB ou support SD.

Que se passe-t-il entre le moment où vous appuyez sur le bouton de mise sous tension de votre ordinateur et le moment où vous voyez le logo de votre OS avec sa barre de chargement ?

Image utilisateur

Pendant le moment ou vous voyez les écrans publicitaires avec les marques et logos de votre PC, un programme se charge : le BIOS.

Le BIOS (ou Basic Input Output System) va lire le MBR (Master Boot Record), premier secteur d'un disque dur ou se trouve la table des partitions et un programme : le bootloader.

Ce superbe programme, stocké sur la ROM (mémoire morte) ou sur la mémoire vive, va extraire le noyau du système d'exploitation (kernel en anglais) qui fera le lien entre logiciel et matériel et donne sa dernière instruction : l'exécuter.

Et ce bootloader cela peut être Grub.

Voila pour le début, la partie la plus ennuyeuse est passée. :p

Installation de Grub

Vous allez ici apprendre à installer Grub2 sur votre machine. Depuis la version 2, c'est devenu très facile ! ;)

ATTENTION !
La partie qui suit comporte certains risques, minimes certes mais pas inexistants.
Les opération décrites ici ont été effectuées par des millions de gens mais je ne saurais être tenu responsable des éventuels dommages rencontrés si vous vous écartez hors des sentiers battus.

Installation

Installer Grub2 n'est pas difficile, il vous suffit d'installer le paquet *grub-pc*, puis de lancer la configuration automatique :

```
$ sudo apt-get install grub-pc
$ sudo grub-install /dev/sdX // Remplacez le X par la lettre de votre disque-
dur dans /dev/ (Exemple: /dev/sda)
$ sudo update-grub
```

Grub-install copiera tous les fichiers dont grub-pc a besoin dans le dossier /boot et la commande update-grub génèrera un nouveau fichier grub.cfg (voir plus loin).

Upgrade depuis la version précédente

La première version de Grub est peut-être déjà installée sur votre machine. Il est alors possible que Grub2 soit automatiquement mis à jour si la version de votre système est récente, mais pour les versions antérieures, vous serez probablement amenés à effectuer les manipulations suivantes :

```
$ sudo apt-get install grub-pc
$ sudo upgrade-from-grub-legacy
```

Durant l'installation, vous devrez configurer le disque sur lequel se trouve la partition /boot.

Image utilisateur

Attention, n'appuyez surtout pas directement sur ENTRER sans avoir confirmé la sélection du disque, sous peine de corrompre Grub legacy sans installer correctement Grub2. Durant le prochain démarrage, vous allez obtenir une erreur "GRUB Error 15" (il faudra alors utiliser un Live-CD afin de réparer Grub).

Voila, GRUB est enfin installé.
Si tout s'est bien passé, félicitations à vous. Si ce n'est pas le cas, reportez-vous à la partie "Réparation et Restauration" (en cours de rédaction :-°) ou en attendant, aux forums ou bien envoyez moi un MP ;)

Configurez Grub2

Ce chapitre va vous permettre de vous faire la main avec Grub2 et de le configurer à vos envies. A vos claviers !

Suite à un commentaire, je vous signale qu'il y a une différence importante entre Grub et Grub2 dans la numérotation des partitions :

- Sur Grub, la numérotation démarre à 0 pour les disques physiques ET pour les partitions.
  Exemple : /dev/hda1 ou /dev/sda1 sera nommé hd(0,0) par grub.
- Sur Grub2, elle démarre à 0 pour les disques physiques et à 1 pour le numéro de la partition.
  Exemple : /dev/hda1 ou /dev/sda1 sera nommé hd(0,1) par grub-pc.

Mise au point

Voici d'abord ce que la plupart d'entre vous ont (je dis la plupart car cela peut varier selon vos OS installés) :

Image utilisateur

Un Menu classique de GRUB

Ce n'est pas très accueillant, je vous l'accorde...

Les fichiers à modifier

Le fichier /etc/default/grub (debian)

On commence par ouvrir le fameux fichier (en root bien sur) :

```
# gedit /etc/default/grub // Pour Gnome
# kate /etc/default/grub // Pour KDE
```

Citation

> # If you change this file, run 'update-grub' afterwards to update
> # /boot/grub/grub.cfg.
>
> GRUB_DEFAULT=0
> GRUB_HIDDEN_TIMEOUT=0
> GRUB_HIDDEN_TIMEOUT_QUIET=true
> GRUB_TIMEOUT="3"
> GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
> GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
> GRUB_CMDLINE_LINUX=""
>
> # Uncomment to disable graphical terminal (grub-pc only)
> #GRUB_TERMINAL=console
>
> # The resolution used on graphical terminal
> # note that you can use only modes which your graphic card supports via VBE
> # you can see them in real GRUB with the command `vbeinfo'
> #GRUB_GFXMODE=640x480
>
> # Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to
> Linux
> #GRUB_DISABLE_LINUX_UUID=true
>
> # Uncomment to disable generation of recovery mode menu entrys
> #GRUB_DISABLE_LINUX_RECOVERY="true"

Dans cette couleur bleue, ce sont les commentaires.
Dans cette couleur rouge, ce sont les parties actives qui seront traitées.

Le répertoire /etc/grub.d/ (debian)

La commande `ls -al` nous renvoie :

```
total 52
drwxr-xr-x   2 root root  4096 2009-11-09 23:45 .
drwxr-xr-x 169 root root 12288 2009-12-07 17:40 ..
-rwxr-xr-x   1 root root  3296 2009-10-24 02:44 00_header
-rwxr-xr-x   1 root root  1154 2009-10-24 02:31 05_debian_theme
-rwxr-xr-x   1 root root   783 2008-03-04 19:20 06_dell_theme
-rwxr-xr-x   1 root root  3778 2009-10-24 02:44 10_linux
-rwxr-xr-x   1 root root   772 2009-10-23 18:11 20_memtest86+
-rwxr-xr-x   1 root root  5467 2009-10-29 17:21 30_os-prober
-rwxr-xr-x   1 root root   214 2009-10-24 02:44 40_custom
-rw-r--r--   1 root root   483 2009-10-24 02:44 README
```

A l'exception du README, les autres sont des script bash. La génération du fichier grub.cfg va se passer tout simplement en exécutant l'ensemble de ces scripts dans un ordre croissant, l'ordre étant appliqué sur les numéros XX précédent le caractère _ dans le nom de ces scripts.

- **00_header :** script permettant la génération de l'en-tête du fichier grub.cfg. Cet en-tête est généré principalement à l'aide des informations que vous avez paramétrées dans le fichier /etc/default/grub.
- **05_debian_theme :** script permettant la génération des informations sur le thème graphique du menu de démarrage. À l'heure où j'écris ces lignes, sur Ubuntu 9.10, le thème est minimal : du texte blanc sur un fond noir.
- **10_linux :** script permettant de générer les entrées correspondant à votre système GNU/Linux hôte.
- **20_memtest86+ :** script permettant de générer les entrées memtest.
- **30_os-prober :** script permettant de détecter des nouveaux systèmes lors de la génération du fichier grub.cfg et de générer les entrées correspondantes. Il est normalement capable de détecter les systèmes avec un noyaux Linux, un noyau Hurd, les systèmes Windows et les systèmes Mac OS X.
- **40_custom :** script permettant de générer des entrées introduites manuellement dans ce fichier.

Les vérifications memtest86+ sont parfois très utiles, mais sont rarement employées et sont surtout très longues ...
Il est donc possible de les désactiver, mais comment ?

Rappelez-vous :

Citation : Moi

> A l'exception du README, les autres sont des script bash. La génération du fichier grub.cfg va se passer tout simplement en exécutant l'ensemble de ces scripts dans un ordre croissant, l'ordre étant appliqué sur les numéros XX précédent le caractère _ dans le nom de ces scripts.

Il suffit donc d'empêcher le script memtest86+ de s'exécuter.
Or, pour empêcher ça, nous avons les chmods, facile : `sudo chmod -x 20_memtest86+`

Le fichier /boot/grub/grub.cfg

Alors là, pas touche ! Si vous êtes sur Debian (Ubuntu par exemple), ce fichier est généré automatiquement par Grub après chaque mise à jour de votre part (n'oubliez pas de le faire après les modifications) : `sudo update-grub`

Sinon, vous pouvez le modifier ; le principe reste le même que plus haut (la syntaxe est un peu différente):

- **(hdn,m)** est la partition m du disque n, le numéro du disque commence par zéro (0), les partitions par un (1).
- **set default=n** est l'entrée par défaut, qui démarrera automatiquement après le time-out.
- **set timeout=m** le temps m m est le temps d'attente avant que le choix par défaut ne démarre.
- **menuentry "str"{options} title string 'str'** pour préciser manuellement une entrée du menu et la façon donc elle sera affichée.
- s**et root=(hdn,m)** définit la partition root (/), là où est installé le noyau.
- **linux /path ro root=/dev/device initrd /initrd.img** options pour la partition root, si le noyau n'est pas installé sur la partition / .
- **chainloader +1** permet de passer le relai à un autre bootloader (exemple celui de Windows, nécessaire à son démarrage)

Modification des entrées

Ici, ça ce complique un peu (non, ne partez pas, j'ai dit un peu ! :p)

La ligne de base

Nous allons éditer la ou les lignes commençant par **GRUB_DISTRIBUTOR** (je n'en ai qu'une, mais je n'exclus pas la possibilité de systèmes un peu plus exotiques ^^)

Ces lignes vont définir ce qui sera affiché à l'écran suivant les options choisies.

Choisissez parmi les options suivantes :

Citation : Doc

> GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian` # donne : **Ubuntu, Linux 2.6...**
> #GRUB_DISTRIBUTOR=`lsb_release -d -s 2> /dev/null || echo Debian` # donne : **Ubuntu 9.10, Linux 2.6...**
> #GRUB_DISTRIBUTOR=`echo -n $(lsb_release -cds 2> /dev/null || echo Debian)` # donne : **Ubuntu 9.10 karmic, Linux 2.6....**

Les autres

Il est aussi possible d'ajouter des entrées "manuellement". Pour cela, éditez le fichier **/ect/grub.d/40_custom** et ajoutez vos entrées. En effet, tout ce qui est présent dans ce fichier sera recopié tel quel dans le fichier **grub.cfg**.

Exemple:

Citation : 40_custom

> ### BEGIN /etc/grub.d/40_custom ###
> # This file provides an easy way to add custom menu entries. Simply type the
> # menu entries you want to add after this comment. Be careful not to change
> # the 'exec tail' line above.
>
> menuentry "Windows 7" {
> set root=(hd0,1)
> chainloader +1
> }

Protégez Grub2 avec un mot de passe

Bon, ici je ne suis pas un pro et le web n'est franchement pas riche en informations, donc malgré tous mes tests, je vous conseille une émulation avant de modifier véritablement ce qui suit.

Protéger automatiquement tous les recovery modes

Le recovery mode est un mode de Linux vous permettant de réparer votre OS en cas de problème et vous offre notamment un shell normal ainsi que plusieurs autres options intéressantes.

Pour le protéger, il vous suffit de dé-commenter cette ligne du fichier **/ect/default/grub** :

Citation : menu.lst

> #GRUB_DISABLE_LINUX_RECOVERY="true" // Enlevez le #

On n'oublie pas de mettre à jour :

```
$ sudo update-grub
```

Ainsi Grub ne génèrera plus d'entrée recovery dans son menu. Notez cependant que le jour où vous avez un problème, vous risquez fort de devoir booter sur un Live CD pour la re-commenter.

Protéger entièrement le menu

Vous pouvez également protéger avec un mot de passe tout le menu.

Ce How To m'a été proposé dans les commentaires. Je ne traiterai pas cette partie plus en détail car :

- je n'aime pas vraiment le concept (devoir sans arrêt taper son mot de passe à chaque démarrage...) ;
- ces explications, bien qu'en anglais, sont très bien rédigées.

Le démarrage

Eh oui, la charrue avant les bœufs...
Reprenons le début du fichier /etc/default/grub :

Citation

> GRUB_DEFAULT=0

Surement la plus intéressante : elle vous permet de modifier l'entrée de votre menu sur laquelle vous allez booter automatiquement. La valeur peut être un nombre naturel (définissant la position) ou *saved*. Ce dernier indique que l'entrée choisie au dernier démarrage sera la prochaine entrée par défaut. Lorsque cette ligne est commentée, la valeur par défaut est 0.

Citation

>

> GRUB_TIMEOUT="5"

Cette ligne précise le nombre de **secondes** à attendre avant de booter sur le *default*. La valeur est soit un nombre naturel (secondes), soit -1. Ce dernier indiquera que le menu doit attendre indéfiniment. Précisons que si la valeur est 0, l'entrée par défaut sera bootée instantanément. Lorsque cette ligne est commentée, la valeur par défaut est 5.

Citation

> #GRUB_HIDDEN_TIMEOUT=0

Permet de définir si le menu de démarrage doit être affiché ou non. Si ce paramètre est commenté, il sera affiché, sinon, sa valeur est un nombre naturel (secondes). Lequel définit le nombre de secondes à attendre avant de lancer le boot de l'entrée par défaut. Si cette valeur est supérieure à 0, pendant le nombre de secondes définies, vous aurez la possibilité de faire afficher le menu manuellement en appuyant sur Esc par exemple. Notez bien la différence avec le précédent : ici, vous ne verrez pas le menu.

Citation

> #GRUB_HIDDEN_TIMEOUT_QUIET=true

Permet de définir si un chronomètre doit être affiché durant les GRUB_HIDDEN_TIMEOUT secondes définies dans le paramètre précédent. Sa valeur est true ou false. Lorsque cette ligne est commentée, la valeur par défaut est false.

Fond d'écran

Plutôt que de faire un vieux plagia, je préfère vous rediriger vers la doc, bien plus complète sur le sujet.

Le lien est le suivant :

http://doc.ubuntu-fr.org/grub-pc#exemple_fond_d_ecran

Notez cependant que ce n'est pas quelque chose d'essentiel, et que la plupart d'entre vous choisiront sûrement de ne pas afficher leur menu du tout.

Pas vraiment de conclusion, les choses vues dans ce chapitres sont assez diverses et variées.

Encore une fois, si vous avez rencontré un problème, vous pourrez vous reporter au chapitre sur les erreurs fréquentes (en cours de rédaction) ou bien poster un message sur les forums (pensez à faire une recherche avant, les forums grouillent de sujets sur les erreurs de Grub) ou encore m'envoyez un MP, je me ferai un plaisir de vous aider ! ;)

Configurez Grub-legacy

Dans ce chapitre, vous allez apprendre à configurer Grub (ou encore "Grub legacy", l'ancienne version) pour le rendre plus convivial et à terme plus sécurisé !

Déterminer l'ordre de boot

Prenez votre menu et définissez le **numéro** auquel va correspondre le système qui vous intéresse de cette façon :

```
Ubuntu, kernel 2.6.17-10-generic

Ubuntu, kernel 2.6.17-10-generic (recovery mod)

Ubuntu, memtest86+

Other operating systems:

Windows XP Media Center Edition

Windows NT/2000/XP

Microsoft Windows XP Embedded
```

Sur mon code :

- **Ubuntu, kernel 2.6.17-10-generic = 0**
- Ubuntu, kernel 2.6.17-10-generic (recovery mode) = 1
- Ubuntu, memtest86+ = 2

- **Other operating systems = 3**
- Windows XP Media Center Edition = 4
- Windows NT/2000/XP = 5
- Microsoft Windows XP Embedded = 6

La première ligne compte pour 0.
Il est important de prendre en compte les lignes d'information telles que "Other operating systems".

Notez bien le nombre qui vous intéresse : celui sur lequel vous désirez booter par défaut.

Méthode 1 : Le default

Une fois revenu sur votre bureau, ouvrez un terminal et entrez-y :

- pour KDE :
  ```
  kdesu kate /boot/grub/menu.lst
  ```
- pour Gnome :
  ```
  gksudo gedit /boot/grub/menu.lst
  ```
- pour XFCE :
  ```
  sudo mousepad /boot/grub/menu.lst
  ```

Si vous n'avez pas compris, on va ouvrir le fichier cible de l'adresse **/boot/grub/menu.lst**.

Effectivement, ce n'est pas très digeste... :'(
Le mien est résumé ainsi :

```
default 0


timeout 15


hiddenmenu



title Ubuntu, kernel 2.6.17-10-generic
```

```
root (hd0,4)
kernel /boot/vmlinuz-2.6.17-10-generic root=/dev/sda5 ro quiet splash
initrd /boot/initrd.img-2.6.17-10-generic
quiet
savedefault
boot



title Ubuntu, kernel 2.6.17-10-generic (recovery mode)
root (hd0,4)
kernel /boot/vmlinuz-2.6.17-10-generic root=/dev/sda5 ro single
initrd /boot/initrd.img-2.6.17-10-generic
boot



title Ubuntu, memtest86+
root (hd0,4)
kernel /boot/memtest86+.bin
quiet
boot


title                Other operating systems:
root



title                Windows XP Media Center Edition
root               (hd0,0)
savedefault
makeactive
chainloader        +1
```

Remplacez tout simplement le chiffre suivant *default* par celui qui vous intéresse.

```
default X
```

Où X est le chiffre que vous avez choisi en première partie.

Enregistrez votre fichier, fermez et redémarrez.
Magique :magicien: , le curseur est déjà placé sur le choix que vous vouliez. Laissez passer le temps et Grub bootera sur votre choix par défaut.
C'est tout.

Méthode 2 : savedefault

Il existe une autre méthode afin de déterminer le système par défaut.

Grub vous propose de déterminer lui-même le système sur lequel vous allez booter, en ne choisissant pas n'importe lequel, mais le dernier état sauvegardé.

Question logique : comment sauvegarde-t-on ?

Regardez vos entrées, avec un peu de chance, vous devriez y voir au moins un *savedefault*. Cela signifie que vous allez automatiquement enregistrer ce choix comme choix par défaut pour le boot.

Placez donc *savedefault* après votre choix le plus important et prenez soin de le retirer aux autres. Ensuite, remontez en haut du fichier et éditez encore une fois la ligne *default X* et changez le X par *saved*.

Grub sélectionnera automatiquement le système qu'il aura sauvegardé au préalable ;)

Annexes

Le timeout

Une chose qui peut être intéressante est de pouvoir modifier la ligne :

```
timeout 15
```

Cette ligne permet de régler le temps limite pour faire votre choix.

Il est par défaut de 10 et comme vous pouvez le voir, j'ai mis 15 secondes.
Autre chose : vous pouvez désactiver cette fonction en commentant la ligne (mettre un # devant), ainsi :

Citation : menu.lst

> # timeout 10

Le gris signifie que la ligne est commentée, GRUB ne bootera plus après un certain temps, ce sera donc à vous de taper Entrée.

ATTENTION !, Ne mettez SURTOUT PAS

```
timeout 0
```

Vous n'auriez plus aucun temps de battement avant le démarrage. C'est une erreur très fréquente et qui peut être particulièrement désagréable à réparer.
Imaginons que vous choisissez de faire booter votre système sur Windows. Vous n'auriez alors plus accès à Linux, et sachant que vous ne pouvez modifier votre menu.lst que depuis ce système, cela vous pose donc un gros problème.

Trop tard ? Ne vous inquiétez pas, ce n'est pas dramatique du tout ;) Faites une recherche sur les forums où envoyez moi un petit MP, ou reportez-vous à la partie concernant les erreurs fréquentes (en cours de rédaction actuellement). Un chapitre fera en effet bientôt la liste des erreurs les plus courantes et expliquera comment les rattraper ;)

Les noms des systèmes

Autre chose : il est possible et facile de modifier les noms des systèmes.

Une partie de mon menu.lst :

```
title Ubuntu, kernel 2.6.17-10-generic
root (hd0,4)
kernel /boot/vmlinuz-2.6.17-10-generic root=/dev/sda5 ro quiet splash
initrd /boot/initrd.img-2.6.17-10-generic
quiet
savedefault
boot
```

Il suffit de modifier cette ligne :

title Ubuntu, kernel 2.6.17-10-generic

En :

title Ubuntu

Pour changer le nom et avoir un menu de boot plus simple.

La protection des entrées

Et oui ! C'est possible ! Grub vous permet de protéger votre système avec un mot de passe supplémentaire.

Et alors ? J'ai déjà un Mdp pour me connecter sur ma machine, moi !

C'est vrai. Néanmoins, ce que vous ne savez peut être pas, c'est qu'en mode recovery, vous avez accès à beaucoup d'éléments et notamment une console en root sans mot de passe ! C'est plutôt risqué !

Pour protéger Grub par un mot de passe, il vous faut déjà ... un mot de passe.
Grub vous propose par ailleurs de le crypter lui-même en md5 :

```
grub> md5crypt
md5crypt
Password: votre mot de passe
Encrypted: $1$gLhU0/$aW78kHK1QfV3P2b2znUoe/
grub> quit
```

Décommentez ensuite la ligne correspondant dans le menu.lst (**en n'oubliant pas de changer le mot de passe encrypté par le vôtre !!!**)

Citation

> ## password ['--md5'] passwd
> # If used in the first section of a menu file, disable all interactive editing
> # control (menu entry editor and command-line) and entries protected by the
> # command 'lock'
> # e.g. password topsecret
> password --md5 1gLhU0/$aW78kHK1QfV3P2b2znUoe/
> # password topsecret

Pour entrer dans le système, vous devrez entrer votre mot de passe avant de sélectionner l'entrée. Pour cela, tapez simplement sur **p**.

Maintenant vous avez deux possibilités :

Protéger une seule entrée

C'est assez facile, ajoutez simplement lock à la suite de vos options sur le système voulu :

Citation

```
title Ubuntu 9.04, kernel 2.6.28-14-generic (recovery mode)
lock
uuid e77d8ae8-36d8-42d1-891d-2cf89dbf0b0c
kernel /boot/vmlinuz-2.6.28-14-generic root=UUID=e77d8ae8-36d8-42d1-891d-2cf89dbf0b0c ro single
initrd /boot/initrd.img-2.6.28-14-generic
```

Pour être sûr de ne pas vous planter, testez sur un ancien noyau ou un vieux truc histoire de ne pas rester coincé au menu.

Protéger tous les recovery mode

Encore une fois, cela va être assez rapide.
Remplacez :

Citation

> # lockalternative=false

Par :

Citation

> # lockalternative=true

Ainsi, tous vos recovery modes seront protégés et cela même après mise à jour du noyau ;)

Il est vrai que la première version est plus facile à configurer que la dernière, tout tient en un seul fichier (un peu "mastoc" certes :p).

La structure des répertoires

Contrairement à MS-DOS, Unix voit ses disques comme une unique arborescence. Une partition contient la racine du système de fichier, qu'on note `/` (et non `c:\` comme sous MS-DOS). D'autres partitions, des disquettes, des CD-ROM, etc., peuvent être "montés" dans des répertoires. Par exemple, sur la machine que j'utilise en ce moment, le contenu du CD-ROM est accessible dans le répertoire `/mnt/cdrom`. Une fois les différents systèmes de fichiers montés, leur utilisation est transparente, sauf dans quelques cas particuliers: impossibilité d'écrire sur un CD-ROM... Bien souvent, sur les systèmes importants, les utilisateurs ne savent même pas dans quelle machine se trouve le disque dur qu'ils utilisent.

La structure "standard" des répertoires est décrite par le FHS (Filesystem Hierarchy Standard) auquel la plupart des distributions Linux essaient de se conformer.

Les principaux répertoires sont:

/bin :

contient les commandes de base (sous Unix, presque toutes les commandes sont "externes", et non intégrées au shell comme sous DOS).

/boot :

contient les informations nécessaires au démarrage de la machine.

/dev :

contient les fichiers spéciaux correspondant aux périphériques.

/etc :

la plupart des fichiers de configuration.

/home :

contient les répertoires personnels des utilisateurs. Par exemple, l'utilisateur toto a généralement pour répertoire `/home/toto`.

/lib :

contient les principales bibliothèques partagées (équivalent des DLL de Windows).

/lost+found :

quand un disque crashe pour une raison ou pour une autre, on utilise le programme fsck pour réparer (équivalent de scandisk); c'est là qu'il dépose les fragments de fichiers perdus.

/mnt :

les répertoires utilisés pour monter temporairement un système de fichiers (disquette, CD-ROM...).

/opt :

c'est la qu'on installe les logiciels commerciaux.

/proc :

un répertoire factice, dont les fichiers contiennent des infos sur l'état du système et des processus en cours d'exécution.

/root :

le répertoire de l'administrateur système. Il n'est pas sous `/home` pour des raisons que je détaillerai plus loin.

/sbin :

les commandes de base nécessaires a l'administration système (vérification et réparation des disques, mise en place du réseau...).

/tmp :

les fichiers temporaires.

/usr :

les logiciels installés avec le système.

/usr/X11R6 :

X-Window.

/usr/bin :

les exécutables.

/usr/dict :

les dictionnaires (pour les correcteurs d'orthographe et les craqueurs de mots de passe).

/usr/doc :

la doc.

/usr/etc :

des fichiers de config.

/usr/games :

les jeux.

/usr/include :

les fichiers d'en-tête pour la programmation.

/usr/info :

la doc au format GNU info.

/usr/lib :

les DLL non vitales.

/usr/local :

une sous-hiérarchie qui contient des logiciels compilés sur place à partir des sources. Organisation similaire à `/usr`.

/usr/man :

le manuel en ligne. Les fichiers sont compressés.

/usr/sbin :

principalement les serveurs réseau.

/usr/share :

des fichiers de données.

/usr/spool :

généralement un lien symbolique vers `/var/spool`.

/usr/src :

les sources de certains logiciels, principalement le noyau de Linux.

/var :

des données fréquemment réécrites.

/var/catman :

les pages du manuel décompressées (ça se fait à la demande).

/var/lib :

des bases de données, des fichiers de config...

/var/local :

complète `/usr/local` de la même façon que `/var` complète `/usr`.

/var/lock :

des fichiers qui servent à marquer l'utilisation de certaines ressources. Par exemple, quand un logiciel se sert du modem, il crée un fichier ici pour le signaler.

/var/log :

le journal du système.

/var/run :

principalement des infos sur les serveurs en fonctionnement.

/var/spool :

les spools: tout ce qui est "de passage" en attendant d'être utilisé par un logiciel. Ca inclut entre autres le mail, les news, les files d'attente des imprimantes...

/var/tmp :

des fichiers temporaires.

Une petite explication des structures assez similaires de `/`, `/usr` et `/usr/local`. Il est assez fréquent que `/usr` soit stocké sur un disque distinct de `/`, voire sur une autre machine. Donc `/` contient de quoi monter `/usr`, y compris à travers le réseau, et de quoi réparer si quelque chose foire. La même remarque s'applique a `/home`; c'est pourquoi le répertoire de l'administrateur est `/root` et non `/home/root`. De plus, comme `/usr` peut être monté en réseau, il peut être partagé entre plusieurs machines; c'est pourquoi les fichiers de config et de données sont dans `/var`.

Quant à `/usr/local`, il est séparé de `/usr` car si `/usr` est facilement récupérable en cas de pépin (au pire, on réinstalle tout), `/usr/local` est beaucoup plus difficile à reconstruire: il faut récupérer les sources des logiciels et les recompiler.

Enfin, `/opt` a une structure totalement différente: chaque logiciel y a son propre répertoire. On pourra par exemple avoir une hiérarchie du style:

> ```
> /opt ---+
>   +- /opt/Office40
>   +- /opt/netscape
>   +- /opt/kde
>   +- /opt/gnome
>   |
> ```

Généralement, chaque répertoire de `/opt` possède au moins les sous-répertoires `bin` et `etc`, qui contiennent respectivement les exécutables et les fichiers de config:

> ```
> /opt ---+- /opt/gnome --+- /opt/gnome/bin
>   |   +- /opt/gnome/etc
>   |   |
>   |
> ```

Étendre la partition et le système de fichiers (Côté Linux)

Une fois l'espace alloué par Proxmox, le système Linux à l'intérieur du LXC doit être informé qu'il peut utiliser ce nouvel espace.

Étape 1 : Vérification de l'état actuel

Connectez-vous en console ou SSH **à l'intérieur** de votre LXC et vérifiez l'espace disponible :

```
df -h
```

Si la taille affichée pour `/` est déjà la nouvelle taille, Proxmox a fait le travail automatiquement (courant en LXC). Sinon, passez à l'étape suivante.

Étape 2 : Redimensionner le système de fichiers (ext4)

La majorité des LXC utilisent le système de fichiers **ext4**. Pour l'étendre sans redémarrer :

```
# On demande à resize2fs d'occuper tout l'espace libre sur le disque racine
resize2fs /dev/sda1
```

Note : Si `/dev/sda1` ne fonctionne pas, utilisez la commande `lsblk` pour identifier le nom exact de votre partition racine.

Étape 3 : Cas particulier - Gestion par LVM

Si votre LXC utilise LVM (très rare à l'intérieur d'un conteneur, mais possible), suivez ces étapes :

```
# 1. Étendre le volume physique
pvresize /dev/sda

# 2. Étendre le volume logique (remplacez par votre chemin)
lvextend -l +100%FREE /dev/mapper/ubuntu--vg-ubuntu--lv

# 3. Étendre le système de fichiers
resize2fs /dev/mapper/ubuntu--vg-ubuntu--lv
```

> **Astuce :** Si après ces commandes `df -h` ne montre toujours pas de changement, essayez simplement de redémarrer le conteneur depuis l'interface Proxmox (Stop puis Start).

? Guide Complet : Authentification SSH par Clés

L'authentification par clés SSH remplace avantageusement les mots de passe. Elle est plus sécurisée, insensible aux attaques par force brute, et indispensable pour l'automatisation avec des outils comme Ansible (AWX).

---

1. ? Le Concept : Clé Publique vs Clé Privée

L'authentification SSH repose sur la cryptographie asymétrique. Au lieu d'avoir un mot de passe unique, on utilise un couple de fichiers indissociables :

◫◫ La Clé Publique (Le Cadenas)	◫◫ La Clé Privée (La Clé physique)
• **Extension :** Finit toujours par `.pub` (ex: `id_ed25519.pub`). • **Rôle :** C'est le cadenas que vous installez sur toutes les portes (serveurs) où vous souhaitez entrer. • **Sécurité :** Elle n'est **pas secrète**. Vous pouvez la distribuer partout, l'envoyer par mail, la mettre sur GitHub. Sans la clé privée associée, elle ne sert à rien.	• **Extension :** Pas d'extension (ex: `id_ed25519`). • **Rôle :** C'est la clé unique que vous gardez sur vous (votre PC ou votre gestionnaire de mots de passe / AWX). Elle ouvre les cadenas. • **Sécurité : ULTRA SECRÈTE**. Elle ne doit jamais quitter votre machine ou votre coffre-fort. Si quelqu'un la vole, il a accès à tous vos serveurs.

---

2. ?? Générer sa paire de clés

On utilise aujourd'hui l'algorithme **ED25519**, qui est plus moderne, plus rapide et plus sécurisé que l'ancien RSA.

Ouvrez un terminal sur votre machine locale et tapez :

```
ssh-keygen -t ed25519 -C "cle-admin-cluster"
```

- `-t ed25519` : Spécifie l'algorithme cryptographique.

- `-C "..."` : Un commentaire pour vous rappeler à quoi sert cette clé (très utile quand on en a plusieurs).

Note : L'outil vous demandera où sauvegarder (laissez par défaut) et si vous voulez une "passphrase". Pour l'automatisation (Ansible/AWX), laissez la passphrase vide en appuyant deux fois sur Entrée.

---

3. ? Déployer la clé (Méthode classique)

Pour autoriser votre clé sur un serveur distant, il faut copier le contenu de la clé publique (le cadenas) dans le fichier `~/.ssh/authorized_keys` du serveur. La commande `ssh-copy-id` fait cela automatiquement :

```
# Remplacer 'utilisateur' et 'ip_du_serveur'
ssh-copy-id -i ~/.ssh/id_ed25519.pub utilisateur@10.151.x.x
```

Il vous demandera le mot de passe du serveur une dernière fois pour installer le cadenas. Ensuite, vous pourrez vous connecter sans mot de passe !

---

4. ? Déploiement Massif (La boucle du Sysadmin)

Si vous avez 15 serveurs ou LXC à configurer d'un coup, faire la commande précédente manuellement est fastidieux (taper "yes" pour l'empreinte, taper 15 fois le mot de passe...).

Voici comment automatiser le déploiement de la clé publique avec une boucle Bash et l'outil `sshpass`.

Prérequis :

Installez l'utilitaire `sshpass` sur votre machine d'administration :

```
sudo apt update && sudo apt install sshpass -y
```

Le Script de boucle :

Modifiez ce bloc avec vos IPs, votre utilisateur cible, et votre mot de passe, puis collez-le dans le terminal.

```
# 1. On stocke le mot de passe dans une variable d'environnement
export SSHPASS='MonMotDePasseSuperSecret'

# 2. On lance la boucle sur notre liste d'IPs
for ip in 10.151.151.230 10.151.151.244 10.151.151.100; do
    echo -e "\n=== Copie de la clé sur root@$ip ==="

    # sshpass : injecte le mot de passe automatiquement
    # StrictHostKeyChecking=no : accepte automatiquement la nouvelle empreinte (le fameux 'yes')
    sshpass -e ssh-copy-id -o StrictHostKeyChecking=no -i ~/.ssh/id_ed25519.pub root@$ip
done

# 3. On détruit la variable pour des raisons de sécurité
unset SSHPASS
```

Résultat : En 5 secondes, votre clé SSH est poussée sur l'ensemble de votre parc. Vous êtes prêt pour Ansible !

Environnement Python pour Ansible & Molecule (venv)

Objectif : Standardiser un environnement Python isolé pour Ansible et Molecule afin de garantir stabilité, reproductibilité et compatibilité en environnement LXC / serveur / homelab.

---

1. Architecture recommandée

Principe :
Un environnement Python isolé (venv) par stack.

- `/opt/ansible-venv` → Ansible + Molecule
- Pas d'installation pip globale
- Pas de mélange apt + pip pour Ansible

---

2. Pré-requis système

```
sudo apt update
sudo apt install -y python3 python3-venv python3-pip \
gcc git build-essential
```

ATTENTION :
Ne jamais utiliser `sudo pip install` sur le système hôte.

---

3. Création du venv Ansible

```
python3 -m venv /opt/ansible-venv
```

Correction des permissions (important en LXC)

```
sudo chown -R $USER:$USER /opt/ansible-venv
```

4. Activation du venv

```
source /opt/ansible-venv/bin/activate
```

Alias recommandé

```
echo 'alias avenv="source /opt/ansible-venv/bin/activate"' >> ~/.bashrc
echo 'alias dvenv="deactivate"' >> ~/.bashrc
source ~/.bashrc
```

Usage : `avenv` pour activer l'environnement.

---

5. Installation Ansible (core)

```
pip install --upgrade pip setuptools wheel
pip install ansible
```

Validation

```
ansible --version
ansible-config --version
```

---

6. Installation Molecule

Installation standard

```
pip install molecule
```

Drivers recommandés

```
# Docker
pip install molecule molecule-docker

# Podman
```

```
pip install molecule molecule-podman
```

7. Vérification globale

```
which ansible-config
which molecule
ansible --version
molecule --version
```

Résultat attendu :
Tous les binaires doivent pointer vers `/opt/ansible-venv/bin/`

8. Maintenance et mises à jour

Voir les paquets obsolètes

```
pip list --outdated
```

Mettre à jour pip

```
pip install --upgrade pip
```

Mise à jour globale du venv

```
pip install pip-review
pip-review --auto
```

ATTENTION :
Les mises à jour automatiques peuvent casser la compatibilité Ansible/Molecule. À utiliser uniquement en environnement non production.

9. Sauvegarde / reproductibilité

Exporter l’environnement

```
pip freeze > requirements.txt
```

Restaurer

```
pip install -r requirements.txt
```

---

10. Dépannage (RUNBOOK)

? ansible-config introuvable

```
pip install ansible
```

? Permission denied dans /opt/ansible-venv

```
sudo chown -R $USER:$USER /opt/ansible-venv
```

? Molecule ne trouve pas Ansible

```
pip install ansible molecule
```

? Mauvais PATH (pipx / system conflict)

```
which ansible
which molecule
echo $PATH
```

---

11. Commandes opérationnelles

```
avenv               # activer venv
dvenv               # désactiver venv
ansible-playbook    # exécuter playbooks
molecule test       # tests infra
pip list            # paquets installés
pip list --outdated # audit
```

12. Recommandations d’architecture

- 1 venv = 1 stack (Ansible / CI / dev)
- éviter pip global système
- préférer /opt pour environnements partagés serveur
- préférer ~/venvs pour dev utilisateur

Conclusion :
Un environnement venv propre garantit stabilité, reproductibilité et compatibilité avec Ansible & Molecule en environnement LXC / homelab.

Extraire le contenu d'un coup d'un contexte

Voici comment extraire tout le contenu de plusieurs fichiers dans une sous-arborescence :

```
.venv) [nancmdzk@TOLX302130067 ~/workspace/techdata-ansible-collection]
(docs/roles_documentation)$ tree techdata/monitoring/roles/kibana/
techdata/monitoring/roles/kibana/
├── defaults
│   └── main.yml
├── files
│   └── node.options
├── handlers
│   └── main.yml
├── meta
│   ├── argument_specs.yml
│   └── main.yml
├── README.md
├── tasks
│   ├── cleanup_old_facts.yml
│   ├── configure.yml
│   └── main.yml
├── templates
│   └── kibana.yml.j2
├── tests
│   └── main.yml
└── vars
    └── main.yml

8 directories, 12 files
```

La commande magique :

```
find techdata/monitoring/roles/kibana/ -type f -exec echo -e "\n=== {} ===" \; -exec cat {} \;
```

La même commande magique extraite dans un fichier :

```
find techdata/monitoring/roles/kibana/ -type f -exec echo -e "\n=== {} ===" \; -exec cat {} \;
> kibana_full_export.txt
```

Notice explicative des options de la commande :

- **`find techdata/monitoring/roles/kibana/`** : Outil de recherche et chemin du dossier de départ où fouiller.
- **`-type f`** : Restreint la recherche aux fichiers uniquement (*f* pour *file*). Cela permet d'ignorer les dossiers et d'éviter les erreurs de lecture.
- **`-exec ... \;`** : Indique à la commande `find` d'exécuter une action spécifique sur chaque élément trouvé. Le `\;` sert à marquer la fin de l'instruction à exécuter.
- **`echo -e "\n=== {} ==="`** : La première action exécutée. Elle affiche le nom du fichier comme un titre.
  - **`-e`** : Active l'interprétation des caractères spéciaux, comme le `\n` qui crée un saut de ligne pour aérer la lecture.
  - **`{}`** : C'est la variable de `find`. Elle est remplacée dynamiquement par le chemin du fichier actuellement traité.
- **`-exec cat {} \;`** : La deuxième action exécutée immédiatement après. La commande `cat` lit et affiche l'intégralité du contenu du fichier (représenté par `{}`).
- **`> kibana_full_export.txt`** : Redirige la sortie standard (tout ce qui se serait affiché dans le terminal) pour l'écrire directement dans un fichier texte.