

Administration système GNU / Linux

Vincent LAINE



Table des matières

1. [Introduction](#)
2. [L'environnement de travail](#)
3. [Interpréteur de commande](#)
 1. [Root et sudo](#)
4. [VIM/VI](#)
5. [Workflow](#)
6. [Le Shell](#)
[Les redirections](#)
7. [Configuration réseau](#)
8. [Gestion des paquets](#)
9. [Gestion des services](#)
10. [Gestion des utilisateurs & groupes](#)
[Création des comptes](#)
11. [Gestions des droits](#)
12. [Gestion des médias](#)
13. [Gestion des disques & LVM](#)
14. [Gestion des processus](#)
15. [Crontab](#)
16. [Archivage et sauvegarde](#)
17. [Debug & Logs Debian](#)
18. [les fichiers hosts.allow et hosts.deny](#)
19. [xinetD](#)
20. [iptables](#)
21. [ufw \(Uncomplicated Firewall\)](#)
22. [Scripting BASH](#)
23. [Administration système](#)
- [Services GNU / Linux](#)
24. [NTP - Le serveur de temps](#)
25. [VSFTPD - Le serveur FTP Linux](#)
26. [Serveur WEB](#)
27. [Service Web - Apache](#)
 1. [Apache2 - Le serveur WEB](#)
[Sécuriser son site web](#)
28. [Partage de fichiers sous Linux](#)
 1. [Partage de fichier avec Samba](#)
[Administration - Partage privé](#)
 2. [Partage de fichier avec Samba \(standalone\)](#)
[Administration - Partage public](#)
29. [Samba4 - l'AD, le DC, le membre ?](#)
 1. [Samba4 - Serveur de fichier \(DM\)](#)
 2. [Samba4 - Le Serveur d'impression](#)
30. [DHCP](#)
31. [Serveur DNS & Bind](#)

Introduction

Qu'est ce que Linux ?

- logiciel libre multi utilisateur développé par Linus Torvald open source en 1991
- 1992 : Linux deviens Open source
- 1996 : Tux deviens la mascotte officiel

Linux est un noyau de type UNIX, les distribution elle sont GNU/Linux (/!\ A ne pas confondre !!!)

Le premier noyaux OS linux : Mimix.os (1991)

Par défaut, les utilisateurs ordinaires ne peuvent pas toucher aux fichiers d'autre utilisateurs

Un utilisateur ne peut pas modifier les fichiers des paramètres du système, ni supprimer des programmes; etc... il n'y a que root qui peut le faire.

L'utilisateur root = super administrateur, ayant tous les droits.

Os multi-utilisateurs/multi tâches :

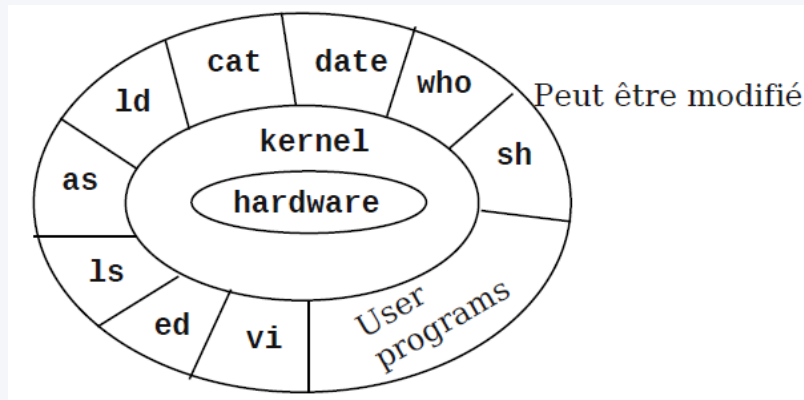
- Peut exécuter plusieurs tâches simultanés sur plusieurs utilisateurs simultanément

L'environnement de travail

Diverses couches dans un système UNIX :

3 couches principales :

- Hardware/matériel (proc, carte réseau...)
- Kernel (compilé en C et C++ et assembleur)
- Des modules (Bibliothèques d'application)
- Cat, ls, vi.. Ce sont des module GNU (couche 3)
- GNU : GnulsNot unix ->Marquer la différence entre linux et UNIX
- Les programmes utilisateurs sont au même niveaux que les modules GNU



L'environnement de travail

L'accès aux fonctionnalités et aux données dans l'environnement de travail Linux s'effectue de deux manières :

- Window Manager (KDE, gnome,) : Autrement dit une interface graphique
- Via un terminal (une ou plusieurs sessions) appelé shell (coquillage). Plusieurs types de shell : Bash, ksh, csh...

Window manager : Interface graphique entre les commandes de bases (ls, cd, etc..) et le système.

Quelques applications essentiels

Les éditeurs de textes sont de loin les applications les plus utilisés :

Editeurs de textes graphiques

- Emacs, Xemacs (l'un des trois principaux)
- Kate (sous KDE, peu utilisé)

Editeur de texte uniquement

- Vi, ViM (Vi improved) semi-graphique)
- nano (éditeur de texte de base pour débutant)

autre éditeur

- Quanta+ (uniquement pour le web)

Interpréteur de commande

Interpréteur de commande : outils pour exécuter des commandes tapées par un utilisateur dans un terminal

Un shell (coquillage) : tout passe à travers le shell, il est l'interface homme/machine en ligne de commande (CLI)

Les commandes sont tapés en mode texte dans le terminal.

Nous pouvons avoir des sortie graphique ou bien textuelle des commandes utilisés dans un shell

Les shells sont programmables afin de faciliter leur utilisation, amélioré leur performance, et surtout SCRIPTER.

Exemple d'un shell bash

```
gaetano@wolfix: /home/gaetano - local - Konsole

[wolfix@home/gaetano] ls
asd/  cnam/  Downloads/  imash/  lfa/  Mail/  monet/  projets/  TMP/  wolfix.old/
bin/  Desktop/  Essi/  isia/  lib/  Master/  music/  RI/  unix/
C/  docs/  images/  js/  lpmi/  media/  perso/  tmp/  web/

[wolfix@home/gaetano] ls asd/
0304/  colles/  cours/  index.html  java/  misc/  solutions/  sujets/  support/

[wolfix@home/gaetano] ls bin/
2pages*  latexps*  mycvs*  pub  syncport*  twopages*  websync*  xunison*
backup*  listing*  print  start-ssh-add.sh*  tidy*  unison*  xframe*
fromweb*  mkps*  prosper*  startXemacs*  toweb*  webclient*  xskey*

[wolfix@home/gaetano] ls Downloads/
12412-flatcolourscheme.tar.gz  CrystalColor.kth  linux.love-1600x1200.png
14.jpg  essential-20050412.tar.bz2  linux.love-800x600.png
24253-kubuntu-linear16-10.svgz.tar.gz  FairytaleWorld.tar.gz  linux.love.tar.gz
A DHTML Calendar.zip  gestaction3D.avi  README
ApplicationIN_kopplad0506.pdf  install_flash_player_7_linux-1.tar.gz  Thumbs.db
Cezanne_packaged.tar.bz2  knifty-0.4.2/  Torchlight_0_2_0_tar.bz2
CrystalClear.tar.gz  linux.love-1024x768.png

[wolfix@home/gaetano] ls projets/
0203/  0304/  0405/

[wolfix@home/gaetano] ls unix/
CoursZsh.ps  docs/  eg/  electrons/  essi/  harmo/

[wolfix@home/gaetano] date
Mon Aug  8 17:27:05 CEST 2005
[wolfix@home/gaetano]
```

Exemple de
commande
utilisé dans
un shell
Bash

Raccourcis shell bash

CTRL + **r** : recherche dans l'historique

CTRL + **a** : début de ligne

CTRL + **e** : fin de ligne

CTRL + **←** : mot précédent

CTRL + **→** : mot suivant

CTRL + **k** : coupe de la position courante jusqu'à la fin de la ligne

CTRL + **y** : colle ce qui a été précédemment coupé

CTRL + **d** : efface le caractère à droite du curseur

CTRL + **t** : transpose le caractère sous le curseur avec le caractère suivant

ESC + **d** : efface le mot à droite du curseur

ESC + **t** : transpose le mot précédent avec le mot suivant

Raccourcis shell bash

Complétion

TAB

: complète la saisie en cours

TAB

+

TAB

: affiche une liste des complétion possibles

Historique

CTRL

+

r

texte : recherche *texte* dans l'historique



: commande précédente



: commande suivante

!

nombre : exécute la commande numéro *nombre*

!

texte : exécute la dernière commande débutant par *texte*

\$HISTFILE : fichier historique (par défaut ~/.bash_history)

\$HISTFILESIZE : *taille du* fichier historique (par défaut 500)

Les interpréteurs les plus courant

sh : Bourne Shell. L'ancêtre de tous les shells.

bash : Bourne Again Shell. Une amélioration du Bourne Shell, disponible par défaut sous Linux et Mac OS X.

ksh : Korn Shell. Un shell puissant assez présent sur les Unix propriétaires, mais aussi disponible en version libre, compatible avec bash.

csh : C Shell. Un shell utilisant une syntaxe proche du langage C.

tcsh : Tenex C Shell. Amélioration du C Shell.

zsh : Z Shell. Shell assez récent reprenant les meilleures idées de bash, ksh et tcsh.

apt-get install ksh

#chsh /bin/ksh : Change Shell.

Shell

Entrée / Sorties

Par défaut tout processus possède :

- 1 entrée
- 2 sorties

Ces I/O sont distinctes pour chaque processus

Pour les programmes interactifs (comme les shells) :

- les entrées proviennent du clavier
- les sorties s'affichent sur l'écran

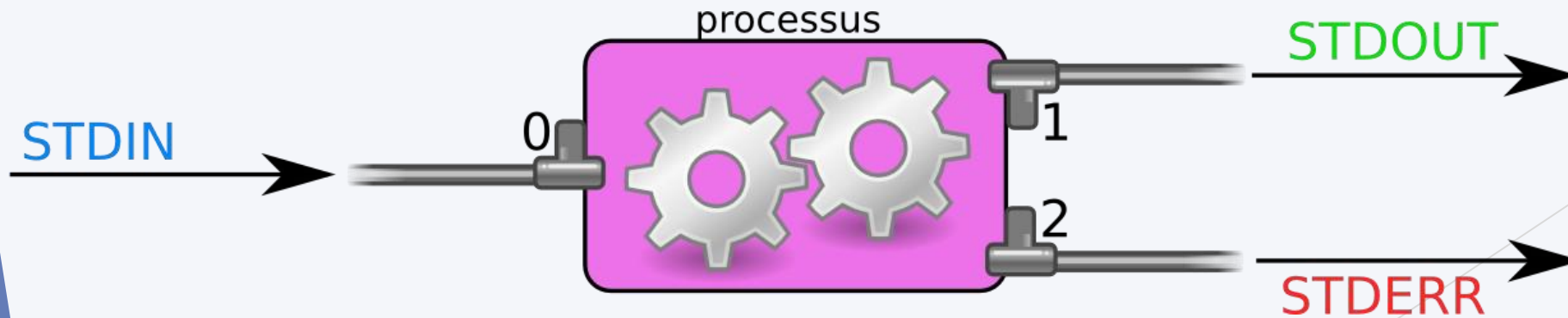
Shell

Entrée / Sorties

STDIN(entrée standard) : ce qui est envoyé vers le processus

STDOUT(sortie standard) : ce qui est envoyé par le processus

STDERR(sortie erreur standard) : les erreurs renvoyés par le processus



Shell

Entrée / Sorties

Ces entrées/sorties standard sont en fait des noms symboliques, correspondant à des «descripteurs de fichiers » :

- **STDIN**(INput) : descripteur de fichier 0 (clavier)
- **STDOUT**(OUTput) : descripteur de fichier 1 (écran)
- **STDERR**(ERRor) : descripteur de fichier 2 (écran)

```
user@host:~$ cat azerty
```

```
azerty
```

```
user@host:~$ cat /etc/hosts
```

```
127.0.0.1      localhost
```

```
192.168.1.1    box
```

```
user@host:~$ cat /etc/bidon
```

```
cat: /etc/bidon: Aucun fichier ou répertoire de ce type
```

Root et sudo

Connexion en root

~su

~sudo su (avec mot de passe user si autorisé)

Mot de passe (root défini à l'installation) : *****

#apt-get install...

Utilisation pour une seule commande :

~ sudo apt-get install

→sudo = SU DO (superuser doing)

- sudo est la commande qui permet d'utiliser des commandes normalement réservés à une certaine classe d'utilisateur
- La configuration de sudo est dans /etc/sudoers (fichier à éditer)

Utilisateur = Utilisation la commande
« sudo » pour effectuer une actions sur le
système :

- Exemple: sudo apt-get install vim

Pour que cela fonctionne, il faut autoriser
l'utilisateur dans le fichier sudoers

Super Administrateur = root

Il a le droit de tout faire sur la machine.

Différents type de commandes

- ★ Une commande Unix peut avoir 0, un nombre fixe ou un nombre variable d'arguments, plus un certain nombre d'options. Une option commence en général par un tiret (-) et deux tirets à la suite (--) indiquent la fin des options sur une ligne de commande
- ★ `[hal@home/bob] date`
Une commande sans paramètre qui affiche la date et l'heure courante
- ★ `[hal@home/bob] cal 2006`
Une commande avec un paramètre (une année) qui affiche le calendrier complet de l'année en question
- ★ `[hal@home/bob] cal 9 2006`
La même commande avec deux paramètres, le mois et l'année
- ★ `[hal@home/bob] echo`
Sans argument cette commande affiche une ligne vide
- ★ `[hal@home/bob] echo Bonjour tout le monde`
Affiche les quatre arguments `Bonjour`, `tout`, `le` et `monde`, et va à la ligne
- ★ `[hal@home/bob] echo -n Bonjour tout le monde`
L'option `-n` empêche le passage à la ligne

Quelques commandes simples

-**history** : Permet d'afficher les historiques des commandes que l'on a tapé.

Pour en ré exécuté une, on peu faire par exemple : !13

-**whoami** : Affiche l'utilisateur

```
[alex@localhost ~]$ whoami
```

```
alex
```

-**hostname** : Affiche le nom de la machine

```
[alex@localhost #]$ hostname
```

```
localhost.localdomain
```

-**pwd** (*Print Working Directory*) : Affiche le répertoire où l'on se trouve

```
[alex@localhost ~]$ pwd
```

```
/home/alex
```

-**man** : Affiche le Manuel (*man [commande]*)

- ~ : Répertoire de base utilisateur

- Find / -name NomDuFichier = Recherche le nomDuFichier dans le dossier racine

home/alex

-which : Indique où se trouve la commande (*which [commande]*)

```
[alex@localhost ~]$ which date
```

```
/bin/date
```

« Un espace » Le dossier est dans /root/

```
cd /root/Un_espace/
```

-/root/Uneespace/ : permet de ne pas traiter le caractère qui suit comme un caractère spécial. (*Utile lors d'un espace par exemple*)

-info : Elle permet d'avoir des infos complètes sur les commandes (*Info [commande]*)

-locate : Cela permet de localiser où est la commande, nécessite d'être en root. (*locate [commande]*)

-ls -la : Affiche les fichiers / dossier d'un répertoire . Par défaut ls liste de manière alphabétique

(-a affiche tout les fichiers, même les cachés)

(-l affiche les informations supplémentaires sur fichier, propriétaire, taille du fichier, date de dernière modif...)

(-h affiche les octets en valeur "humaine" Ko, Mo, Go...)

(-R affiche les répertoires, ainsi que les fichiers contenus dans les sous répertoires)

(-S pour trier selon la taille des fichiers)

(-t permet d'afficher les fichiers modifiés récemment en premier)

(-r permet d'inverser le tri)

Principes et commandes de bases

Sous UNIX/Linux, tout n'est que fichier. Un dossier est un fichier, un fichier texte est un fichier.

Une commande (type ls, cd..) fait appelle à un script qui n'est autre qu'un fichier.

Aucune limite du nombre de caractère dans un fichier (contrairement à windows : 255 caractères max)

Les extensions de fichier sont facultatif (monfichier.exe n'est pas un exécutable au même titre que sous windows)

```
/home/vincent# ls
```

```
test.txt
```

```
test2.txt
```

```
dossier
```

```
cd dossier/
```

```
cd /home/vincent/dossier/
```

★ Un *chemin* («path») est une séquence de répertoires imbriqués, avec un fichier ou un répertoire à la fin, séparés par le caractère /

★ Chemin relatif: `docs/cours/unix.html`
Relatif au répertoire courant

★ Chemin absolu: `/home/john/docs/cours/unix.html`
Chemin depuis le répertoire racine du système (/)

Système de fichier sous linux

/ (racine)

Contient l'ensemble des répertoires

/bin

Contient les exécutable spécifiques au mode utilisateur

/sbin

Contient les exécutable spécifiques au système

/dev/cdrom

Contient les fichiers représentant les périphériques système

/etc

Fichiers de configuration relatifs aux applications

/lib

Librairies partagées pour /bin et /sbin nécessaires pour démarrer le système et exécuter des commandes. On y trouve des modules de Kernel additionnels

/mnt/ ou /media (Suse)

Clique droit sur la VM (dans workstation) → Install Vmware tools

mkdir /mnt/cdrom

Montage : *mount /dev/cdrom /mnt/cdrom*

Démontage : *umount /dev/sdb1 /mnt/usb1*

ls -l /mnt/cdrom

Point de montage pour les systèmes de fichiers extérieurs

/proc

Informations sur le kernel et le système.

/boot

Contient les fichiers nécessaires pour le démarrage: noyau Linux, bootloader,...

Système de fichier sous linux

/home (optional)

Répertoire contenant les sous repertoires personnels pour les utilisateurs

/root (optional)

Répertoire equivalent au home pour l'utilisateur root

/tmp

Fichiers temporaires lies aux processus en cours d'execution

/usr

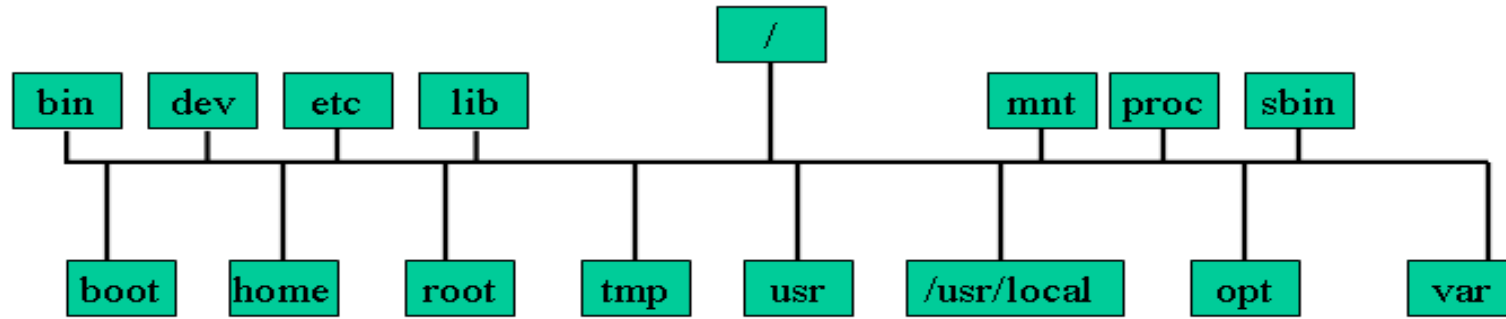
Ressources (données, programmes) pour les utilisateurs

/usr/local or /opt (optional)

Add-on des applications

/var

Données variables, cad les fichiers logs



The base directories

boot home root tmp usr /usr/local opt var

Directories that can be mount points for separate devices

Commandes de bases

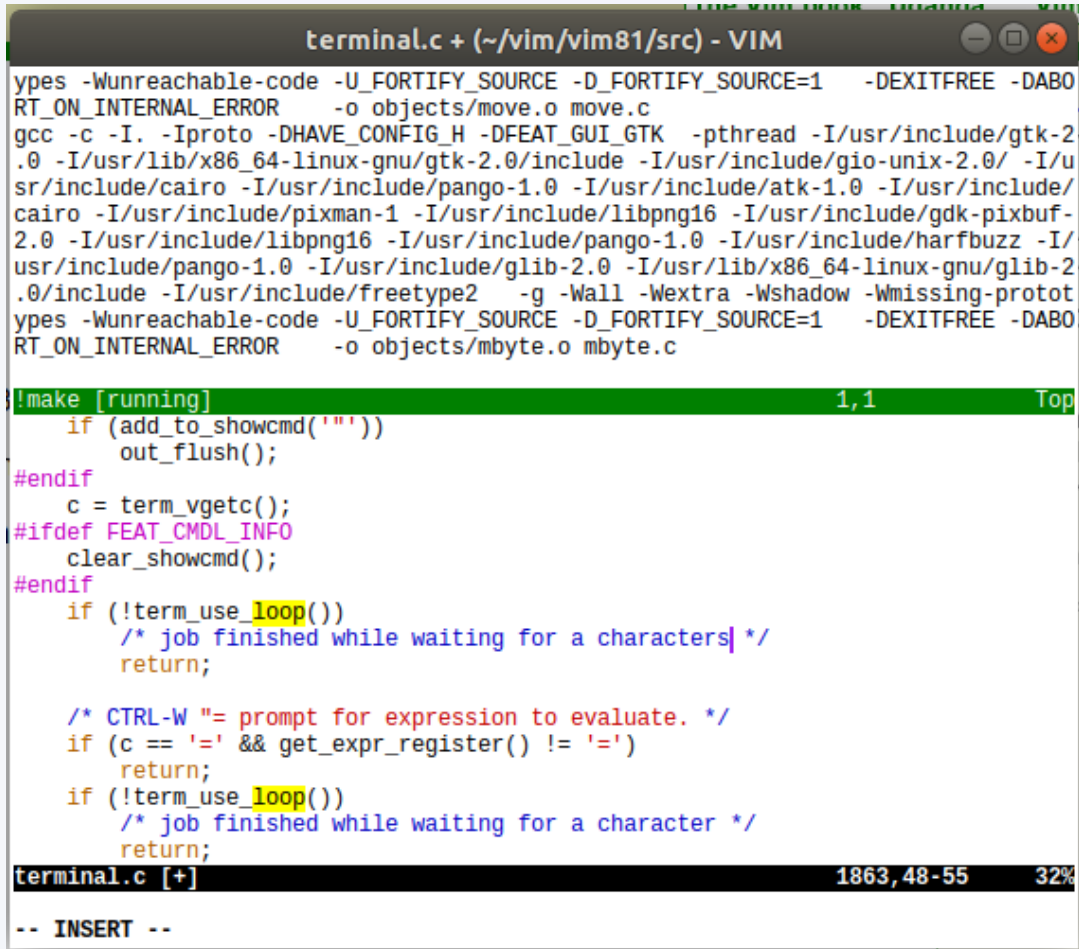
- `cd MonDossier`
 - Aller dans le répertoire `MonDossier`
- `ls`
 - Lister le contenu du répertoire
- `cp source destination`
 - Copier un fichier source vers un fichier destination. Option `-r` pour un répertoire
- `mv source destination`
 - Déplacer un fichier source vers une destination ou le renommer
- `mkdir NouveauDossier`
 - Crée un dossier `NouveauDossier`
- `rm`
 - Efface un fichier. Option `-rf` pour un répertoire
- `cat FichierAlire`
 - Lis le fichier texte `FichierAlire`
- `grep « toto » RechercheFichier`
 - Recherche le mot `toto` dans le fichier `RechercheFichier`

TP - les basiques

1. Allez dans le dossier /root
2. Créer un dossier « d_tp01 »
3. Dans « d_tp01 » créer un fichier « f_01 » avec touch
 1. Exemple : (root@srv-debian:/home/vlaine# cp /root/fichier1 .) = « . » = /home/vlaine/
4. Puis créer un autre fichier « f_02.txt » avec nano dans d_tp01
5. Editer le fichier pour y inclure
6. Nom-prénom-age-classe
7. Copier le fichier f_01 dans /home/votre_user
8. Copier le dossier d_tp01 dans /home/votre_user/copy_dossier/

Astuce : l'option « -r » pour la commande « cp » permet de copier « récursivement et donc de copier un dossier et son contenu, si vous ne mettez pas l'option, la copie échouera
9. Déplacer (donc couper) le fichier « f_02.txt » dans /home/votre_user
10. Créer 3 dossiers en même temps dans /home/votre_user
 1. /home/vlaine/votre/creation/de/dossier
 2. Vous pourrez accomplir ça grâce à l'option « -p » de la commande mkdir
11. Supprimer le dossier et ses sous-dossiers « creation »
12. Renommage de f_02.txt en f_03.txt (la commande est la même que pour le déplacement de fichier)

VIM/VI



```
terminal.c + (~vim/vim81/src) - VIM
types -Wunreachable-code -U_FORTIFY_SOURCE -D_FORTIFY_SOURCE=1 -DEXITFREE -DABO
RT_ON_INTERNAL_ERROR -o objects/move.o move.c
gcc -c -I. -Iproto -DHAVE_CONFIG_H -DPEAT_GUI GTK -pthread -I/usr/include/gtk-2
.0 -I/usr/lib/x86_64-linux-gnu/gtk-2.0/include -I/usr/include/gio-unix-2.0/ -I/u
sr/include/cairo -I/usr/include/pango-1.0 -I/usr/include/atk-1.0 -I/usr/include/
cairo -I/usr/include/pixman-1 -I/usr/include/libpng16 -I/usr/include/gdk-pixbuf-
2.0 -I/usr/include/libpng16 -I/usr/include/pango-1.0 -I/usr/include/harfbuzz -I/
usr/include/pango-1.0 -I/usr/include/glib-2.0 -I/usr/lib/x86_64-linux-gnu/glib-2
.0/include -I/usr/include/freetype2 -g -Wall -Wextra -Wshadow -Wmissing-protot
ypes -Wunreachable-code -U_FORTIFY_SOURCE -D_FORTIFY_SOURCE=1 -DEXITFREE -DABO
RT_ON_INTERNAL_ERROR -o objects/mbyte.o mbyte.c

!make [running] 1,1 Top
    if (add_to_showcmd(''))
        out_flush();
#endif
    c = term_vgetc();
#ifdef FEAT_CMDL_INFO
    clear_showcmd();
#endif
    if (!term_use_loop())
        /* job finished while waiting for a characters */
        return;

    /* CTRL-W "= prompt for expression to evaluate. */
    if (c == '=' && get_expr_register() != '=')
        return;
    if (!term_use_loop())
        /* job finished while waiting for a character */
        return;
terminal.c [+] 1863,48-55 32%
-- INSERT --
```

Photo : Editeur de texte VIM
(coloration syntaxique lié à
Vim)

```
root# cd /home/user
#sudo apt-get install vim
#touch test.txt
#vim test.txt
→ Vous arrivez ensuite dans
votre fichier
→ À savoir
    → i = insertion
    → Echap = sortir du
        mode insertion
    → :w = enregistrer
    → :q = quitter
    → :wq = enregistrer et
        quitter
```

Commande sous ViM

Editeur présent dans toutes les distributions

2 modes de fonctionnement

Commande (mode par défaut)

Insertion

Naviguer dans les modes

i pour passer en mode insertion

: pour passer en mode commande

Echap pour quitter n'importe quel mode

:w → sauvegarde

:q → quitte le document

:q! → quitte le document sans sauvegarder

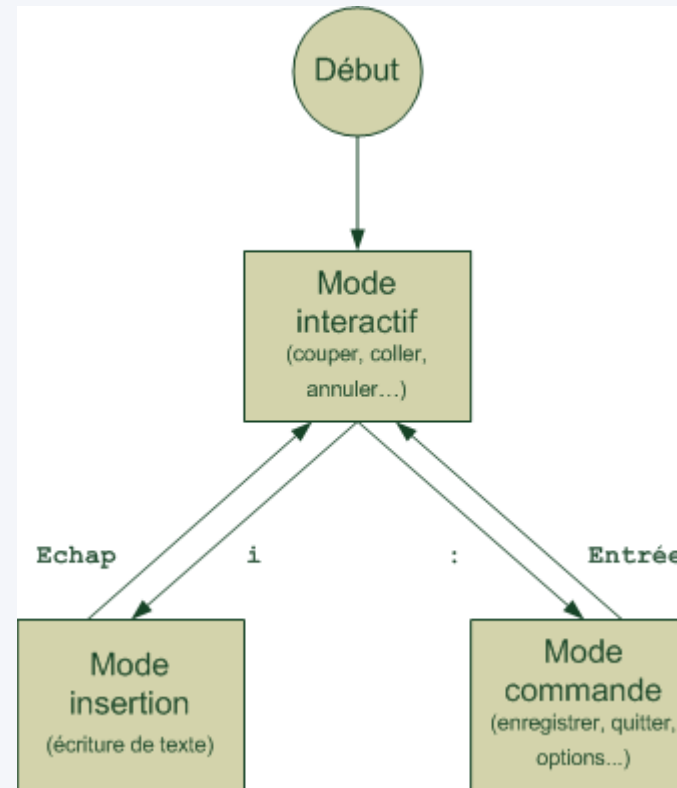
:wq → sauvegarde et quitte

:f toto → enregistre sous un autre fichier

:u → Annulations des dernières actions

/toto → recherche toto (n pour le prochain toto)

v → passe en mode sélection (visual)



x → Suppression d'un caractère
u → Annulation de la dernière action
d → coupe la sélection (mode visuel)
dd → Efface la ligne entière
2dd → Efface 2 lignes
dw → Efface le mot
y → copie la sélection
yy → copie la ligne du curseur
p → colle (après le curseur)
2p → Colle 2 fois la ligne
G → Va a la fin du fichier
gg → Va au début du fichier
r → Remplacement d'une lettre
10G → On va placer le curseur à la 10^{ème} ligne du fichier

Recherche :

/ = Rechercher une occurrence depuis la ligne ou est le curseur

? = Rechercher une occurrence depuis le début du fichier

Recherche et remplacement de texte

:s/ancien/nouveau : remplace la première occurrence de la ligne où se trouve le curseur ;

:s/ancien/nouveau/g : remplace toutes les occurrences de la ligne où se trouve le curseur ;

:#,#s/ancien/nouveau/g : remplace toutes les occurrences dans les lignes n° # à # du fichier ;

:%s/ancien/nouveau/g : remplace toutes les occurrences dans tout le fichier. C'est peut-être ce que vous utiliserez le plus fréquemment.

:1,15:s/supercour/supercours → De la ligne 1 à 15, le mot « supercour » sera remplacé (substitute) par « supercours »

:sp

:vsp

Split

vim :

:sp = Split horizontal

:vsp = Split vertical

tmux = Splitter de terminal

- ☞ `Ctrl + w` puis `Ctrl + w` : navigue de viewport en viewport. Répétez l'opération plusieurs fois pour accéder au viewport désiré.
- `Ctrl + w` puis `j` : déplace le curseur pour aller au viewport juste en dessous. La même chose fonctionne avec les touches `h`, `k` et `l` que l'on utilise traditionnellement pour se déplacer dans Vim.
 - `Ctrl + w` puis `+` : agrandit le viewport actuel.
 - `Ctrl + w` puis `-` : réduit le viewport actuel.
 - `Ctrl + w` puis `=` : égalise à nouveau la taille des viewports.
 - `Ctrl + w` puis `r` : échange la position des viewports. Fonctionne aussi avec « R » majuscule pour échanger en sens inverse.
 - `Ctrl + w` puis `q` : ferme le viewport actuel.

TP ViM - Bonus

1. Récupérer le fichier `apache2.conf` en installant le paquet « `apache2` »
2. Dans `/etc/apache2` → Copier le fichier `apache2.conf` dans `/home/votre_user`
3. Décommenter les lignes 30 à 50
4. Effacer les lignes 35 à 40
5. Effacer les 10 premiers mots de la ligne 45
6. Copier les lignes 200 à 202
7. Coller ses lignes à l'emplacement des lignes 35
8. Remplacer toutes les lettres de la ligne 5 par la lettre « `a` »
9. Annuler les modifications
10. Aller à la ligne 100 (grâce à `G`)
11. Rechercher les mots « `error` »
12. Remplacer tout les mots « `error` » par « `super_error` »
13. Splitter l'écran horizontalement
14. Lancer la commande externe « `ls -lisa` »
15. Modifier le fichier `/etc/vim/vimrc` pour configurer la numérotation des lignes automatiques

TP ViM - Bonus Correction

1. `apt-get install apache2`
2. `cp /etc/apache2/apache2.conf /home/vlaine/`
3. `:30,50s/#//g`
4. `5dd`
5. `45G - 10dw`
6. `200G - 3yy` (Ou mode visuel --> Selection puis y)
7. `35G - p`
8. `5G - Selection d'une ligne en visu puis : (r)a`
9. `u`
10. `100G`
11. `?error` (ou `/error`) (puis n pour aller à la prochaine occurrence)
12. `:%s/error/super_error/g`
13. `:sp` (ou `:vsp`)
14. `:!ls -lisa` (entrée pour revenir au fichier)
15. Ouverture du `/etc/vim/vimrc`
 1. Ajout de la directive "set number" à la ligne 50

Workflow

- Après l'installation d'une machine ?
- Configuration réseau (IP, DNS) et mot de passe root
- Configuration du SSH
- Ajout d'un utilisateur (ou pas ?)
- Installation des paquets essentiels et mise à jour
 - apt-get update
- Installation de votre éditeur favoris
- Installation des VMWare tools si machine virtuelle
- Mise en place de son environnement de travail (Putty, Moba...)

Workflow

- L'environnement de travail
- L'accès aux fonctionnalités et aux données dans l'environnement de travail Linux s'effectue de deux manières :
 - Window Manager (KDE, gnome,) : Autrement dit une interface graphique
 - Via un terminal (une ou plusieurs sessions) appelé shell (coquillage). Plusieurs types de shell : Bash, ksh, csh...
 - Window manager : Interface graphique entre les commandes de bases (ls, cd, etc..) et le systèmes.
- Les outils de connexion à distance
 - Putty
 - MobaXterm

Workflow

- Mot de passe root :
 - Le mot de passe est défini à l'installation, si ce n'est pas le cas nous pouvons le définir après l'installation :
 - `sudo su` - (connexion au compte root via le premier compte utilisateur)
 - `passwd` (commande pour définir un mot de passe)
- Configurer son réseau (IP fixe + DNS)
- SSH
 - L'administration d'un serveur linux se fait dans la majorité des cas via SSH.
 - Autoriser la connexion en root :
 - `/etc/ssh/sshd_config`
 - Trouver la ligne « `PermitRootLogin prohibit-password` »
 - Et la remplacer par « `PermitRootLogin yes` » pour autoriser la connexion en root via ssh
 - `ssh root@ip_du_server`

Workflow

- Root & Sudo (SU DO (superuser doing))
- sudo est la commande qui permet d'utiliser des commandes normalement réservées à une certaine classe d'utilisateur.
- Connexion en root
 - su -
 - sudo su - (avec mot de passe user si autorisé)
- # apt-get install sudo (si non sudoers non présent)
- Utilisation pour une seule commande :
 - vlaine@ubuntu:~\$ sudo apt-get install
- La configuration de sudo est dans /etc/sudoers (fichier à éditer)
 - Commande : visudo
 - root ALL=(ALL:ALL) ALL
 - L'utilisateur root à tout les droits sur la machine
 - %admin ALL=(ALL) ALL
 - Le groupe "admin" à tout les droits
 - jpierre ALL=(ALL) NOPASSWD: /usr/bin/su
 - l'utilisateur jpierre peut utiliser la commande "sudo su" sans utiliser de mot de passe
 - jpierre ALL=(ALL) /usr/sbin/reboot,!/usr/sbin/shutdown, NOPASSWD: /usr/bin/su
 - l'utilisateur jpierre peut :
 - utiliser la commande "sudo reboot" avec son mot de passe
 - utiliser la commande "sudo su" sans son mot de passe
 - Ne peut pas utiliser la commande "sudo shutdown"

Workflow

- Installation des VMWare tools
- Solution 1 : via le CDRom
 - Dans votre hyperviseur, monter le CD des vmware tools
 - Puis en console, monter le CD :
 - « mount /dev/cdrom /mnt »
 - Copier l'archive sur votre machine :
 - « cp /mnt/[Nom_Des_Tools].tar.gz /tmp »
 - Décompressez le
 - « tar xvf /tmp/[Nom_Des_Tools].tar.gz »
 - Puis installer :
 - « cd /tmp/vmware-tools-distrib/ »
 - « ./vmware-install.pl »
- Solution 2 : Via la ligne de commande :
 - Debian : apt-get install open-vm-tools
 - Ubuntu : apt-get install open-vm-tools (14.04)
 - et/ou apt-get install open-vm-tools-desktop (> 14.04)
 - RHEL, CENTOS, FEDORA, SLES, SUSE : apt-get install open-vm-tools-desktop puis apt-get install open-vm-tools

Workflow

- La complétion automatique :
 - Très utile, il est conseillé de se familiariser avec.
- Elle Permet de compléter une saisie utilisateur dans le shell
 - Affecté à la touche « tab »
 - La complétion affiche la plus grande correspondance unique
- S'applique aux commandes et à leurs arguments (selon configuration)
 - « tab-tab » affiche toutes les correspondances possibles.
- Raccourcis clavier du shell :

CTRL + **a** : début de ligne

CTRL + **e** : fin de ligne

CTRL + **k** : coupe de la position courante jusqu'à la fin de la ligne

CTRL + **y** : colle ce qui a été précédemment coupé

Workflow

Clavier FR + bashrc

- cat /etc/default/keyboard : Configuration du clavier par défaut
 - XKBLAYOUT=« fr » (ici clavier en Français)
- loadkeys fr : défini temporairement le clavier en français
- /root/.bashrc ou /home/user/.bashrc
 - .bashrc est le fichier de personnalisation du shell utilisateur
 - alias l='ls -lisahrt' = Exemple d'alias : Quand nous taperons « l » dans la console, la commande sera reconnu en tant que « ls -lisahrt »
 - # enable color support of ls and also add handy aliases
if [-x /usr/bin/dircolors]; then
 test -r ~/.dircolors && eval "\$(dircolors -b ~/.dircolors)" || eval "\$(dircolors -b)"

alias ls='ls --color=auto'

 ○ Permet d'afficher les couleurs automatiquement sur les dossiers/fichiers

```
68 # enable color support of ls and also add handy aliases
69 if [ -x /usr/bin/dircolors ]; then
70     test -r ~/.dircolors && eval "$(dircolors -b ~/.dircolors)" || eval "$(dircolors -b)"
71     alias ls='ls --color=auto'
72     #alias dir='dir --color=auto'
73     #alias vdir='vdir --color=auto'
74
75     alias grep='grep --color=auto'
76     alias fgrep='fgrep --color=auto'
77     alias egrep='egrep --color=auto'
78 fi
79
80 # some more ls aliases
81 alias ll='ls -aF'
82 alias la='ls -A'
83 alias l='ls -lisahrt'
```

Workflow

TP

- Installer une machine Debian ou Ubuntu (sans interface graphique)
- Préparer son environnement de travail

Le Shell

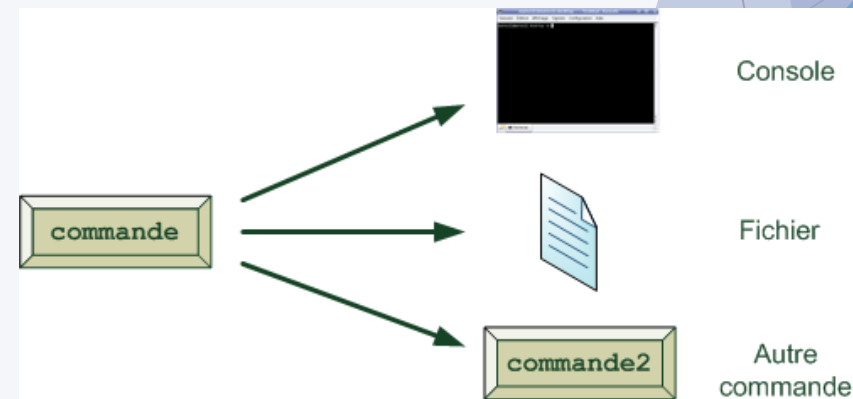
E/S

- Plusieurs "devices" (fichiers dans /dev) ont une vocation particulière :
- /dev/null
 - trou noir annihilant tout ce qui lui est envoyé
- /dev/zero
 - envoie des zéros ad-vitam
- /dev/random /dev/urandom
 - fournisseurs officiels de hasard
- /dev/full
 - dispositif hypochondriaque : se plaint toujours (d'être plein)

Le Shell

Les redirections

- Les E/S peuvent être redirigées de ou vers un fichier
- processus < fichier
 - provient du fichier
- processus > fichier
 - est écrit dans fichier
- processus >> fichier
 - est ajouté dans fichier (en plus des éléments existant)
- Processus > fichier 2> error.txt
 - Ecrit dans le fichier puis écrit les erreurs dans error.txt
- ls > test.txt 2>&1
 - la sortie stderr 2 est tout d'abord redirigée vers la sortie stdout 1, puis la sortie standard de ls s'inscrit dans le fichier test.
- echo bonjour > bonjour.txt
- cat aurevoir.txt >> bonjour.txt
- cat /etc/hostname > /dev/null
- Insérer du texte dans un fichier avec cat (EOF = end of file)
 - cat >> /etc/ntp.conf << "EOF"
 - « Contenu du fichier »
 - EOF



Le Shell

Les redirections

- commande `>/dev/null 2>&1`
- Dans l'ordre, est exécuté :
 - commande (dont les stdout et stderr sont redirigées)
 - stdout vers `/dev/null` (le trou noir)
 - stderr vers stdout ... donc vers `/dev/null`
- La commande exécutée ne renverra ni sa sortie, ni ses messages d'erreur. C'est équivalent à :
- commande `> /dev/null 2 > /dev/null`

Le Shell

TP

1. Copier le contenu de `/etc/passwd` dans le fichier `/tmp/users.txt` (grâce aux redirections)
2. Envoyer le contenu de `users.txt` dans `users2.txt`
3. Ecrire «linux» à la fin de `users.txt`
4. Rediriger le résultat standard de « `ls -lR /` » dans `/tmp/users.txt` (en gardant le contenu initial du fichier)
 1. Bonus : Rediriger les erreurs dans `erreur.txt`
5. Vider le fichier `users.txt`
 1. `/dev/null > users.txt 2> error.txt`

Astuce : `echo`, `cat`, `>`, `>>`

Le Shell

Pipes

1. Les «pipes» (pipelines) permettent d'envoyer la sortie d'une commande (STDOUT) à l'entrée d'une autre (STDIN).
2. On trouve très souvent la commande grep au milieu de pipelines
3. grep permet de n'afficher une ligne que si elle contient une chaîne de caractères donnée
4. Sa syntaxe est :
 1. grep chaîne fichier
 2. affiche les lignes de fichier contenant "chaîne"
5. cat text.txt | grep chaîne
 1. affiche les lignes lues sur l'entrée standard contenant "chaîne«
6. ip link | grep UP > uplinks.txt
 1. Les pipes et redirections peuvent être combinées

Le Shell

Les opérateurs &, &&

- « & » = Cet opérateur permet de lancer simultanément deux commandes dont la première sera lancée en arrière plan. Par exemple :
 - `ls & ls -la`
- L'opérateur "&&" permet de lancer une commande si et seulement si la première (celle à gauche de l'opérateur) s'est correctement terminée :
 - `ls && ls /`
 - Si la commande "ls" se termine correctement alors "ls /" sera exécutée. Un second exemple :
- N:B : Code retour 0. Lorsque qu'une commande est lancé avec succès elle envoie le code retour 0, l'opérateur && fonctionne sur ce principe.

Le Shell

Besoin d'aide ?

1. help commande
 1. affiche le manuel d'une commande interne (builtin)
2. apropos sujet
 1. affiche les pages de man correspondant au sujet
3. whatis commande
 1. affiche une information succincte sur la commande
4. W
 1. Affiche qui fait quoi sur le système
5. strace whoami 2>&1 | grep -E '/etc|/lib'
 1. tracer le cheminement et les actions d'une commande

Le Shell

Quelques commandes de bases

- `cat fichier1 fichier2 ...`
 - affiche le contenu de `fichier1 fichier2 ...` sur la sortie standard si `cat` est appelé sans arguments, la source est l'entrée standard.
 - `cat /dev/urandom`
- `less fichier1 fichier2`
 - comme `cat`, affiche le contenu de `fichier1 fichier2 ...` sur la sortie standard mais effectue un arrêt à chaque page si `less` est appelé sans arguments, la source est l'entrée standard (q pour quitter)
 - `less /etc/passwd`
- `tee fichier`
 - duplique l'entrée standard vers la sortie standard et dans un fichier.
 - `vmstat 1 | tee toto`
 - `vmstat 1 | tee -a toto`
- `wc option fichier`
 - compte le nombre de lignes (-l), bytes (-c), mots (-w) dans fichier.
 - `wc -l /etc/passwd`

Le Shell

Quelques commandes de bases

- `head [nX] fichier1 fichier2 ...`
 - affiche les X premières lignes de fichier1 fichier2 ... sur la sortie standard si tail est appelé sans arguments, la source est l'entrée standard
 - `head -n1 /etc/passwd`
- `tail [nX] [f] fichier1 fichier2 ...`
 - affiche les X dernières lignes de fichier1 fichier2 ... sur la sortie standard
 - si tail est appelé sans arguments, la source est l'entrée standard et le nombre de lignes est 10
 - l'option -f permet de faire un 'tail' continu sur un fichier qui croît
 - `tail -n5 /var/log/syslog`
 - `tail -f /var/log/syslog`
- Une combinaison des deux permet d'afficher la nième ligne d'un fichier :
 - `head -n10 /etc/passwd | tail -n1` : affiche la 10^{ème} ligne de /etc/passwd
 - `head -n10 /etc/passwd | tail -n3` : affiche les lignes 7 à 10 (10-3) lignes de /etc/passwd
- Commande AWK
 - `cat /etc/passwd | awk -F ":" '{ $2 = "" ; print $1 }'`
 - imprime chaque ligne du fichier /etc/passwd après avoir effacé les champs après le premier séparateur (\$2 correspond au deuxième champs, et print \$1 dis que nous afficherons uniquement le premier)

Le Shell

TP Quelques commandes de bases

1. Envoyer la commande « `ls -lR /` » dans le fichier `/tmp/users.txt`
2. Affiche les 50 premières lignes de `users.txt` et redirige les vers `/tmp/redirection.txt`
3. Affiche les 50 dernière lignes et redirige les vers `redirection.txt`
4. Affiche l'intervalle entre les lignes 1000 et 1500, puis redirige les vers `redirection.txt`
5. Affiche la ligne 3000 et redirige la vers `redirection.txt`
6. Affiche combien de fois le mot `root` est contenu dans `redirection.txt`

Utiliser les redirecteurs et la commande `tee`

Configuration réseau

- La commande « ip » du paquet iproute2
 - Par défaut sur les nouvelles versions
- La commande ifconfig du paquet net-tools
 - L'installation du paquet est nécessaire
- Commandes utiles
 - « ip addr » (ou « ip a ») = Voir les interfaces réseaux et leurs IPs
 - « ip link » = Affiche les interfaces réseau disponibles
 - « ifconfig » = Pareil que ip addr
 - « ifconfig -a » = Pareil que ip link
 - « route » = Voir la table de routage et passerelles
- Pour les DNS :
 - « nslookup » (paquet « net-tools »)
 - « dig » (paquet « dig »)
- Manuels :
 - « man ip »
 - « man ifconfig »

Configuration réseau

- Netplan pour Ubuntu:

Depuis la version 18.04 (Bionic Beaver) Ubuntu est passé à Netplan pour la configuration des interfaces réseau.

Il s'agit d'un système de configuration basé sur YAML, qui simplifie le processus de configuration.

- Ce nouvel outil remplace le fichier de configuration (`/etc/network/interfaces`) qui avait été précédemment utilisé pour configurer les interfaces réseau sur Ubuntu. Il remplace également `/etc/resolv.conf`.
- Les fichiers de configuration se trouvent maintenant sous la forme de fichiers YAML à l'emplacement `/etc/netplan/*.yaml`.
- Assurez-vous de respecter les normes YAML lorsque vous modifiez le fichier. Une erreur de syntaxe peut engendrer une mauvaise lecture du fichier de configuration.
- Un fichier `01-netcfg.yaml` ou `/etc/netplan/00-installer-config.yaml` est utilisé pour configurer la première interface.

Configuration réseau

- IP fixe ou DHCP pour Ubuntu :
 - Depuis les dernières version de Ubuntu, celui-ci utilise Netplan pour la configuration IP, sous la forme d'un fichier « yaml ».
 - D'abord vérifier le nom de l'interface :
 - « ip addr show »*

```
root@ubuntu01:/etc/netplan# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:0c:29:ff:a6:0c brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.223/24 brd 192.168.1.255 scope global dynamic ens160
        valid_lft 561sec preferred_lft 561sec
    inet6 fe80::20c:29ff:feff:a60c/64 scope link
        valid_lft forever preferred_lft forever
```

- Edition /etc/netplan/00-installer-config.yaml (fichier par défaut)
 - DHCP :

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    ens160:
      dhcp4: true
  version: 2
```

Configuration réseau

- Edition /etc/netplan/00-installer-config.yaml
 - IP Fixe :

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    ens160:
      addresses: [192.168.1.240/24]
      gateway4: 192.168.1.1
      nameservers:
        addresses: [192.168.1.140, 192.168.1.141]
  version: 2
```

N:b : A retenir que chaque option est séparée d'un saut de ligne.

L'indentation se fera automatiquement lors du saut de ligne.

Option	Exemple	Description
addresses	[192.168.1.157/24]	Une liste d'adresses IP à affecter à une interface. Le format utilise la notation CIDR.
Gateway4	192.168.1.1	L'adresse IP de votre passerelle IPv4 locale.
Dhcp4	True	Définir si DHCP est activé pour IPv4 - true ou false
Dhcp6	True	Définir si DHCP est activé pour IPv6 - true ou false
nameservers	nameservers: addresses: [4.2.2.2, 8.8.8.8]	Configuration des serveurs DNS

Configuration réseau

- Application des paramètres :
 - netplan apply

```
root@ubuntu01:/etc/netplan# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:0c:29:ff:a6:0c brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.157/24 brd 192.168.1.255 scope global ens160
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:feff:a60c/64 scope link
        valid_lft forever preferred_lft forever
```

- Vérification du DNS utilisé :
 - systemd-resolve --status | grep 'DNS Servers' -A2

```
root@ubuntu01:~# systemd-resolve --status | grep 'DNS Servers' -A2
DNS Servers: 192.168.1.140
root@ubuntu01:~# nslookup sbind01.vlne.fr
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   sbind01.vlne.fr
Address: 192.168.1.140
```

Configuration réseau

TP

1. Ajouter une carte réseau supplémentaire à votre VM
2. Modifier votre adresse IP (de la 1^{ère} carte) en IP Fixe
 1. Ne pas oublier les DNS
 2. Nom = 01-netcfg.yaml
3. Modifier en IP Fixe la 2^{ème} carte
 1. 02-netcfg.yaml

Configuration réseau

- Configuration pour Debian ou Ubuntu sans utiliser Netplan :
- Editer le fichier `/etc/network/interfaces` pour les paramètres IP

```
# The primary network interface
allow-hotplug eth0
iface eth0 inet static
address 192.168.1.116
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
gateway 192.168.1.254
#dns-domain example.com
dns-nameservers 192.168.1.101
```

- Editer le fichier `/etc/resolv.conf` pour les paramètres DNS :

```
domain vlne.fr
search vlne.fr
nameserver 192.168.1.1
nameserver 192.168.1.140
nameserver 8.8.8.8
```


Configuration réseau

/etc/hosts

- /etc/hosts peut contenir des associations IP/hostname, permettant :
 - de définir des noms personnalisés (plus courts, pour du test, ...)
 - de surcharger (recouvrir) des noms d'hôtes existants (interception de l'accès à un hostname, utilisation d'un miroir plus proche)
 - de résoudre plus rapidement des noms fréquemment demandés
 - de résoudre des noms en l'absence ou indisponibilité des DNS
 - a.b.c.d nom.domaine.org nom nombis ...

127.0.0.1 localhost

127.0.1.1 ubuntu01

192.168.157.201 srv-deb

192.168.1.150 ubuntu01

Configuration réseau

Connectivité

Test de connectivité réseau :

- netstat -paunt : Lister les ports ouverts
 - -a : Tous les ports
 - -t : Tous les ports TCP
 - -u : Tous les ports UDP
 - -l : Tous les ports en écoute
 - -n : Affiche directement les IP. Pas de résolution de nom.
 - -p : Affiche le nom du programme et le PID associé.
- nc -zvu 192.168.1.140 53
 - Tester le port 53 en UDP
- telnet 192.168.1.140 53

Configuration réseau

Nmap

Nmap :

- `nmap -F -n -sU 127.0.0.1`
 - scan tout les ports en UDP sans résolution de nom (-n) (plus rapide)
- `nmap -F -n -sS 127.0.0.1`
 - scan tout les ports en TCP
- `nmap -sSUS 127.0.0.1`
 - Scan tout les ports
- `nmap -v -A`
 - Scanner le système d'exploitation et les versions de paquets du système cible
- `nmap -O 127.0.0.1`
 - Scan la version du système d'exploitation
- `nmap -p 53 127.0.0.1`
 - Scan d'un port précis
- `nmap -p 0-80,50000 127.0.0.1`
 - Scan une plage de ports. Ici on scan du port 0 au 80 et tous ceux supérieurs à 50000)
- `nmap -p 80 192.168.1.0-255`
 - Scan les serveurs WEB (80) sur le réseau
- on scan 127.0.0.1, par l'interface réseau eth0, en se faisant passer pour 10.0.0.0 depuis le port 80
- `nmap -oN sortie.txt 127.0.0.1` OU `nmap -oX sortie.xml 127.0.0.1`
 - Choisir un fichier de sortie pour y écrire les résultats du scan

Configuration réseau

Nmap

- `nmap -v -sn 192.168.1.0-255`
- Scan toutes les machines UP du réseau
- Trier les informations : `nmap -v -sn 192.168.1.0-255 | grep up -B1 -A1 | grep "Nmap scan" | awk -F " " '{ $1 = "" ; print $5 " " $6 }' | tr -d \" | tr -d \"`

```
sbind01:~# nmap -v -sn 192.168.1.0-255 | grep up -B1 -A1 | grep "Nmap scan" | awk -F " " '{ $1 = "" ; print $5 " " $6 }' | tr -d \" | tr -d \"
adjust timeouts2: packet supposedly had rtt of -199776 microseconds. Ignoring time.
192.168.1.1
192.168.1.113
192.168.1.141
192.168.1.150
192.168.1.157
192.168.1.158
192.168.1.161
192.168.1.167
192.168.1.189
192.168.1.224
192.168.1.226
ubuntu01.vlne.fr 192.168.1.228
192.168.1.229
192.168.1.230
192.168.1.240
sbind01.vlne.fr 192.168.1.140
```

Scanlogd

- Permet la détection de scan de port via NMAP sur une machine cible.
- Les tentatives s'afficheront dans le fichier de log de la machine.
- `apt install scanlogd`
- `systemctl status scanlogd`

```
root@ubulvm:~# tail -f /var/log/syslog | grep scanlogd
Apr 14 17:06:40 ubulvm scanlogd: 192.168.1.240:36206 to 192.168.1.158 ports 143, 53, 3389
Apr 14 17:06:40 ubulvm scanlogd: message repeated 2 times: [ 192.168.1.240:36206 to 192.1
```

Configuration réseau

TP

- Configurer vos 2 machines en IP fixe avec vos serveur DNS
- Les 2 machines doivent se ping entre elles.
 - Les 2 machines doivent donc être sur la même carte et dans le même réseau.

Gestion des paquets

- Les packages (paquetages) ont pour objectif :
 - de permettre une installation simple
 - désinstallation ...
 - mise à jour
 - de gérer des dépendances entre packages
 - d'être re compilable facilement
- .RPM sous Fedora, Redhat, Mandriva, SuSE...
- .DEB sous Debian, Ubuntu
- .tar.gz binaires sous Slackware
- .scripts + tar.gz sources sous Gentoo
- ...sans compter les packages source

Gestion des paquets

- Les packages Debian/Ubuntu (APT)
- Fonctionne sur les distributions basées Debian
 - http://en.wikipedia.org/wiki/List_of_Linux_distributions#Debian-based_free_distributions
- Les systèmes Debian possèdent l'équivalent (approximatif) du monde RPM pour gérer les paquets :
 - dpkg : permet de manipuler les fichiers .deb et de gérer les paquets installés
 - apt-get: permet d'installer des paquets depuis différentes sources
- Les packages Debian ont une extension de fichier *.deb* (paquets binaires)
- La commande dpkg n'accepte pas d'URL

Gestion des paquets

- Les fichiers packages sont manipulés avec la commande dpkg
- **Installer** ou **mettre à jour** un package
 - dpkg -i
 - *package.deb*
- **Supprimer** un package
 - dpkg -r *package*
- **Supprimer** un package et sa configuration
 - dpkg -P *package*
- **Reconfigurer** un package
 - dpkg-reconfigure *package*
- **Rechercher** un fichier dans les paquets installés
 - dpkg -S *fichier*
- **Lister** les fichiers *actuellement* installés pour un paquetage.
 - dpkg -L *paquetage*
- **Lister** les fichiers installés par un paquetage.
 - dpkg -l *paquetage*
- Lister les paquets sur la machine
 - dpkg -l

Gestion des paquets

- La commande apt-get permet de rechercher ou d'installer un paquetage directement depuis un dépôt (Internet, CD/DVD, disque)
- apt-get est *très* performant par rapport a yum
- La liste des dépôts est définie dans /etc/apt/{sources.list,sources.list.d/}
- **Installer** le paquetage nom
 - apt-get install *nom*

les outils de la famille apt utilisent des *sources* pour obtenir les packages et les métadonnées associées

Ces sources, définies dans /etc/apt/{sources.list,sources.list.d/} ont le format suivant :

```
deb http://fr.archive.ubuntu.com/ubuntu/ edgy main restricted
    ↑                               ↑   ↑   ↑
    url du dépôt                 distribution sections...
```

- Attention en ajoutant des sources :
 - il est impératif d'utiliser des sources de confiance
 - il faut vérifier la signature des paquets (apt-key permet de gérer les identités)

Gestion des paquets

- Upgrader tous les paquetages installés
 - apt-get upgrade
- Mettre à jour l'information depuis les dépôts
 - apt-get update
- Supprimer le paquetage
 - apt-get remove package
 - apt-get autoremove package
- Passer à une distribution plus récente
 - apt-get dist-upgrade
- Vérifier l'état de la base de données
 - apt-get check
- Purger les fichiers restants :
 - apt-get purge paquet
- Lister les paquets :
 - apt-cache policy paquet
- apt est en fait une famille de commandes dédiées à la gestion de paquetages.
- Quelques membres de la famille :
 - Apt-cache : rechercher un package dont la description ou le nom contient une expression régulière
 - aptfile search `which bash`
 - aptcache search tcpdump
 - aptcache n search '.*dump.*'

Gestion des paquets

- Il est préférable d'installer des logiciels avec le gestionnaire de paquetage de la distribution
 - intégration plus fine dans la distribution
 - gestion des dépendances et des conflits
 - mises à jour automatiques
 - désinstallation aisée
- Tous les logiciels ne sont cependant pas disponible sous forme de paquetages. Il est donc parfois obligatoire d'installer un paquet depuis les sources.
- Les logiciels installés en dehors du gestionnaire de package doivent l'être dans `/usr/local/` - `/usr/share` - `/opt`
- La plupart des logiciels sont construits autour d'outils de développement communs :
- `automake/autoconf` : définition des paramètres de compilation en fonction de la plateforme
- `make` : automatisation de la construction du binaire à partir des sources et installation
- `build-essential` : Outils de compilations
- Des cas particuliers existent souvent; il n'y a pas de recette miracle...

Gestion des paquets

- Exemple d'installation d'un paquet depuis les sources
 - Ce n'est pas toujours exactement de cette manière, se référer au README.txt présent dans la plupart des sources pour avoir la procédure exacte.
- Décompresser les sources :
 - `tar xvf paquet_source.tar.gz`
 - `cd paquet_source`
- Configurer la compilation :
 - `./configure`
- Compiler :
 - `make`
- Installer :
 - `make install`
- Désinstaller
 - `make uninstall`
- Vous devriez ensuite trouver votre paquet installé dans `/usr/local`
- Attention cette méthode ne permet la désinstallation simple (avec APT par exemple)

Gestion des paquets

- Exemple d'installation d'un paquet depuis les sources méthode création du binaire
- Décompresser les sources :
 - `tar xvf paquet_source.tar.gz`
 - `cd paquet_source`
- Configurer la compilation :
 - `./configure`
- Compiler :
 - `make`
- Installer :
 - `[sudo] checkinstall`
- Après quelques questions, checkinstall va créer un paquet binaire à partir des sources, puis va l'installer.

Gestion des paquets

TP

- Télécharger vsftpd[...].deb
http://ftp.debian.org/debian/pool/main/v/vsftpd/vsftpd_3.0.3-12_amd64.deb
 - wget
 - transfert direct
- Télécharger les sources de htop et les mettre sur votre machine :
<https://sourceforge.net/projects/htop/>
- Installer le paquet libncursesw5-dev avec apt
- Installer le paquet .deb
- Installer htop via les sources
- Afficher la liste des fichiers de vsftpd
- Reconfigurer le paquet vsftpd (dpkg)
- Supprimer le paquet vsftpd
- Mettre à jour Ubuntu

Gestion des services

- Sur Debian, un service (ou daemon / démon) est un script d'initialisation de type "System V" qui va permettre de gérer un serveur (serveur openSSH, serveur FTP, ou autre serveur NTP, etc...) ou tout simplement un programme qui va exécuter des tâches.
- Le service peut être démarré, arrêté, en erreur... Ces services sous forme de script sont présent dans le dossier /etc/init.d
- Il y a plusieurs commandes qui permettent de gérer les services :
 - La commande « service [Nom_Du_Service] [option] » pour la gestion ponctuelle
 - options disponible :
 - start : pour le démarrer
 - stop : pour l'arrêter
 - restart : pour le redémarrer ou le démarrer s'il est arrêté
 - reload : pour recharger la configuration sans le redémarrer (donc sans couper les connexions actives)
 - status : Pour connaître l'état du service. Démarré ou arrêté.
 - enable : Pour activer un service au démarrage de la machine
 - disable : Pour désactiver un service au démarrage de la machine
- Il ne faut pas oublier que ces commandes sont ponctuelles.
Au prochain redémarrage du serveur les services récupéreront le statut qui est défini par défaut.
Si on souhaite modifier le comportement global du service, il faut passer à la configuration.

Gestion des services

- Les applications qui doivent s'exécuter au boot installent un script dans `/etc/init.d`
- Il est ensuite possible de contrôler cette application grâce à ce script
- Démarrer l'application
 - `/etc/init.d/application start`
- Stopper l'application
 - `/etc/init.d/application stop`
- Redémarrer l'applications
 - `/etc/init.d/application restart`
- Demander à l'application de recharger sa configuration
 - `/etc/init.d/application reload (force-reload)`

Gestion des services

Gestion des services Debian avec systemd

- Afficher le statut du système :
 - `$ systemctl status`
- Lister les unités échouées :
 - `$ systemctl --failed`
- Lister les fichiers unités installés :
 - `$ systemctl list-unit-files`
- Lister tout les services en cours d'exécution :
 - `$ systemctl`
- Activer le service « example1 » immédiatement :
 - `# systemctl start example1`
- Désactiver le service « example1 » immédiatement :
 - `# systemctl stop example1`
- Redémarrer le service « example1 » immédiatement :
 - `# systemctl restart example1`
- Voir le statut du service « example1 » :
 - `# systemctl status example1`
- Activer « example1 » pour être lancé au démarrage :
 - `# systemctl enable example1`
- Désactiver « example1 » pour ne pas être lancé au démarrage :
 - `# systemctl disable example1`

Gestion des services

Gestion des services Debian avec systemd

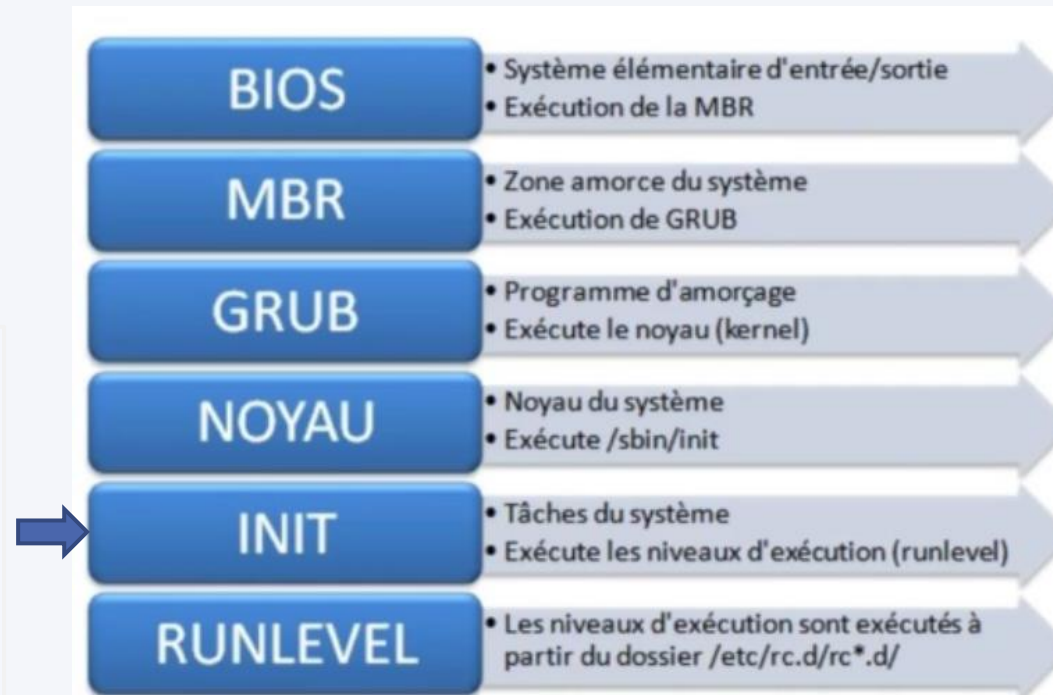
- Installer le paquet samba
 - Apt install samba
- Arrêter le service
 - /etc/init.d/smbd stop
- Démarrer le service
 - Service smbd start
- Activer au démarrage de la machine
 - Systemctl enable smbd
- Afficher son status
 - Systemctl status smbd
- Redémarrer le service
 - /etc/init.d/smbd restart
- Lister tout les services échoués
 - Systemctl --failed
- Désactivation du networkManager :
 - systemctl stop network-manager
 - systemctl disable network-manager
- Ou
 - Vim
/etc/NetworkManager/NetworkManager.conf
 - [ifupdown]
managed=false
 - systemctl restart network-manager

Gestion des services

Gestion des services Debian avec systemd

- systemd est un gestionnaire de système et de services pour Linux. Il est le système d'init par défaut dans Debian depuis Jessie. Systemd est compatible avec les scripts d'init SysV et LSB. Il peut fonctionner comme un remplaçant de sysvinit. Systemd :
 - Offre un démarrage à la demande des daemons
 - Supporte les snapshots et leurs restaurations
 - Maintient les points de montage (montage/démontage)
 - Systemd s'exécute comme démon en tant que PID 1.

- met à jour la liste des bibliothèques partagées ;
- met à jour le nom de la machine (HOSTNAME) ;
- démarre une série de démons : syslogd (service de journaux du système), acpid (gestion de l'énergie), hald (gestion du matériel), cups (file d'impression), etc. ;
- active la gestion du réseau ;
- monte les systèmes de fichiers ;
- etc.



Gestion des services

Gestion des services Debian avec systemd

- Les tâches de systemd sont organisées en tant qu'unités.
 - services (.services)
 - les points de montage (.mount)
 - les périphériques (.device)
 - les sockets (.socket)
 - les minuteurs (.timer).
- Par exemple, le démarrage du démon shell sécurisé est effectué par l'unité `ssh.service`.
- Systemd place chaque service dans un groupe de contrôle (cgroup) dédié.
- Les cibles (targets) sont des groupes d'unités. Les cibles appellent les unités pour assembler le système (`graphical.target`, `default.target`...)
- Systemd crée et gère les sockets utilisés pour la communication entre les composants du système. (ex : crée `/dev/log` puis démarre le démon `syslog`)
 - les processus communiquant avec `syslog` via `/dev/log` peuvent être démarrés en parallèle
 - les services écrasés peuvent être redémarrés sans que les processus qui communiquent via les sockets avec eux perdent leur connexion (mise en tampon par le noyau)

Gestion des services

- Les services Debian sont exécutés en fonctions de leurs niveaux d'attribution de "runlevel" : 0, 1, 2, 3, 4, 5, 6.
 - Le niveau 0 : extinction de l'ordinateur.
 - Le niveau 1 : le mode "single-user", ne contient que les services de base. C'est un peu le mode "sans échec" de Windows, pour les opérations de maintenance
 - Le niveau 2 : le niveau standard qui inclut... tout. Les services réseau, les serveurs, l'interface graphique, etc...
 - Le niveau 6 : identique au niveau 0 sauf qu'il est utilisé en cas de reboot et pas d'arrêt complet
 - Les niveaux 3, 4 et 5 : identiques au niveau 2. Sont présent pour donner plus de finesses à l'administrateur en cas de besoin.
- Un service est constitué au minimum :
 - d'un fichier de script répertorié dans le dossier /etc/init.d
 - d'un lien symbolique de démarrage comme "S10service", placé dans l'un des 6 répertoires /etc/rc?.d/. En général : 2, 3, 4, 5
 - d'un lien symbolique pour l'arrêt du service comme "K20service", placé dans l'un des 6 répertoires /etc/rc?.d/. En général : 0, 1, 6
- Pour les scripts de démarrage (start), le système place un "S" devant le nom du script.
- Pour les scripts d'arrêt (kill), le système place un "K" devant le nom du script.
 - Après chaque lettre "S" ou "K" il y a un nombre : 10, 20, etc... C'est ce chiffre qui donne l'ordre d'exécution du service au sein du runlevel.
- Cela permet de gérer les priorités. 20 s'exécute avant 30, qui s'exécute avant 40, etc... Donc si on veut faire démarrer un service après un autre qui a une priorité de 20, on lui mettra 30 par exemple.
 - A noter que si dans un même runlevel il y a des "K" et des "S", les "K" sont traités avant les "S".

Gestion des services

Runlevel	Systemd Target	Notes
0	runlevel0.target, poweroff.target	Arrête le système
1	runlevel1.target, rescue.target	Mode single user
3	runlevel3.target, multi-user.target	Mode multi-utilisateur, non graphique
2, 4	runlevel2.target, runlevel4.target, multi-user.target	Mode défini par l'utilisateur, identique au 3 par défaut.
5	runlevel5.target, graphical.target	Mode graphique multi-utilisateur
6	runlevel6.target, reboot.target	Redémarre
emergency	emergency.target	Shell d'urgence

Gestion des services

Dans Debian 10 / Debian 9, systemd utilise des targets au lieu de niveaux d'exécution. Le fichier /etc/inittab n'est plus utilisé par systemd pour modifier les niveaux d'exécution.

- Vérifier le runlevel actuel
 - `systemctl get-default` (runlevel par défaut)
 - `who -r`
 - `runlevel`
- Vérifier la liste des targets :
 - `systemctl list-units --type=target`
- Définir un runlevel par défaut
 - `systemctl set-default [NOM_TARGET].target`
- Paquet de gestion des runlevels
 - `sysv-rc-conf`
- Changer de runlevel à chaud
 - `systemctl isolate [NOM_TARGET].target`
 - `Init [0-6]`
 - `telinit [0-6]`

Gestion des services

- Les targets ont leur propres dossier de gestion des services.
- Quand une target est défini par défaut, le dossier contenant les services associés sera utilisé :
- Service par default de multi-user.target :
 - `/etc/systemd/system/multi-user.target.wants/[SERVICE].service`
- Service par default de default.target :
 - `/etc/systemd/system/default.wants/[SERVICE].service.`
- Service par default de graphical.target :
 - `/etc/systemd/system/graphical.wants/[SERVICE].service.`

```
drwxr-xr-x 2 root root 4096 févr. 16 14:34 bluetooth.target.wants
drwxr-xr-x 2 root root 4096 févr. 16 14:33 default.target.wants
drwxr-xr-x 2 root root 4096 févr. 16 14:13 getty.target.wants
drwxr-xr-x 2 root root 4096 févr. 16 14:34 graphical.target.wants
drwxr-xr-x 2 root root 4096 févr. 25 15:49 multi-user.target.wants
drwxr-xr-x 2 root root 4096 févr. 16 14:34 network-online.target.wants
drwxr-xr-x 2 root root 4096 févr. 16 14:34 sockets.target.wants
drwxr-xr-x 2 root root 4096 févr. 16 14:14 sysinit.target.wants
drwxr-xr-x 2 root root 4096 févr. 16 14:34 timers.target.wants
```


Gestion des services

- Changer le runlevel au démarrage
- Lors du lancement de grub appuyer la touche « e » pour éditer le grub menu.
- A la fin de la ligne « linux /vmlinuz-4.19.0[...]ro quiet » entre le « ro » et le « quiet », ajoutez le numéro du runlevel voulu.
- Puis appuyer sur Ctrl + X pour démarrer.
- Ici on démarre avec le runlevel 1 (single user) :

```
GNU GRUB  version 2.02+dfsg1-20+deb10u3

set root='hd0,msdos1'
if [ x$feature_platform_search_hint = xy ]; then
  search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1\
--hint-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1  9666c8de-4c41-4f03\
-b704-639b3e0cd960
else
  search --no-floppy --fs-uuid --set=root 9666c8de-4c41-4f03-b70\
4-639b3e0cd960
fi
echo          'Chargement de Linux 4.19.0-14-amd64 '
linux        /vmlinuz-4.19.0-14-amd64 root=/dev/mapper/srv--deb0\
1--vg-root ro 1 quiet
echo          'Chargement du disque mémoire initial...'
initrd       /initrd.img-4.19.0-14-amd64
```

Gestion des services

TP

1. Afficher votre runlevel par défaut
 1. (normalement le 5)
2. Afficher le runlevel actuel
3. Changer le runlevel
 1. Du 5 au 3
4. Afficher les targets disponibles
5. Supprimer le service smbd dans le runlevel 3
 1. `/etc/rc3.d/S04smbd -> ../init.d/smbd`
 2. `Rm /etc/rc3.d/smbd`
 3. `Systemctl disable smbd`
6. Redémarrer la machine grâce à la commande init
7. Ajouter le service smbd au démarrage (rc3) grâce à un lien symbolique
 1. Depuis `/etc/init.d` jusqu'à `rc3.d/...`
 2. Exemple `ln -s /etc/init.d/ssh /etc/rc3.d/S25ssh`
8. Eteindre la machine
9. Lancer la machine avec le runlevel 1 en éditant le grub menu
10. Revenir au runlevel 3 à chaud après avoir démarré sur le 1
11. Changer le runlevel à 5 à chaud (systemctl)
12. Afficher le runlevel actuel (avec une autre commande)
13. Installer sysv-rc-conf et retirer vsftpd du runlevel 3

Gestion des services

Gestion des services Debian avec update-rc.d et les en-têtes de script LSB

- La commande « update-rc.d » Installer ou supprimer les liens vers les scripts d'initialisation de type Système V.
- update-rc.d met à jour automatiquement les liens vers les scripts d'initialisation de type Système V dont le nom est /etc/rc[runlevel].d/[NN]nom vers les scripts /etc/init.d/[name].
- Exemple d'un en-tête LSB nécessaire au fonctionnement de update-rc.d :

```
### BEGIN INIT INFO
# Provides:          exim4
# Required-Start:    $remote_fs $syslog $named $network $time
# Required-Stop:     $remote_fs $syslog $named $network
# Should-Start:      postgresql mysql clamav-daemon greylist spamassassin
# Should-Stop:       postgresql mysql clamav-daemon greylist spamassassin
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: exim Mail Transport Agent
# Description:       exim is a Mail Transport agent
### END INIT INFO
```

- Required-Start : les services qui doivent être lancés AVANT qu'Exim démarre
- Required-Stop: Même chose pour l'arrêt
- Should-Start : les services qui DEVRAIENT démarrer en même temps qu'Exim
- Default-Start : les runlevels de démarrage
- Default-Stop : les runlevels d'arrêt

Gestion des services

Gestion des services Debian avec update-rc.d et les en-têtes de script LSB

- Donc pour chaque script on indique dans son en-tête LSB ses contraintes de démarrage et d'arrêt.
 - Plus d'instructions à passer en paramètre à update-rc.d tout est dans le fichier.
 - Pour tout savoir sur les en-têtes LSB, « man insserv ».
- Pour appliquer les règles, il suffit d'invoquer update-rc.d avec les options par défaut. Il ira lire la configuration LSB et appliquera les paramètres en conséquences.

```
update-rc.d [-n] nom_du_service defaults
```

- L'option -n test la commande sans la jouer.

```
update-rc.d [-n] [-f] nom_du_service defaults
```

- L'option -f permet de forcer la suppression des liens même si /etc/init.d/nom_du_service existe encore.
- Puis si je souhaite réactiver les services ultérieurement je vais réutiliser la commande :

```
update-rc.d [-n] nom_du_service defaults
```

Gestion des services

Gestion des services Debian avec update-rc.d et les en-têtes de script LSB

- update-rc.d [-n] [-f] nom remove
 - Supprimer un lien
- update-rc.d [-n] nom defaults
 - Créer le lien symbolique vers les runlevels
- update-rc.d [-n] nom
- update-rc.d [-n] nom disable|enable [S|2|3|4|5]

```
1 #!/bin/sh
2 ### BEGIN INIT INFO
3 # Provides:          helloworld
4 # Required-Start:    $remote_fs $syslog $named $network $time
5 # Required-Stop:     $remote_fs $syslog $named $network
6 # Should-Start:
7 # Should-Stop:
8 # Default-Start:     2 3 4 5
9 # Default-Stop:      0 1 6
10 # Short-Description: Script hello test
11 # Description:       Script helloworld de test
12 ### END INIT INFO
13
14 # Placez ici des commandes à exécuter à chaque appel du script
15 echo 'Le script a été appelé !' >> /root/helloworld.txt
16
17 # Le switch case ci-dessous permet de savoir si le système souhaite lancer ou arrêter le script (on le
18 )
19 case "$1" in
```

Gestion des utilisateurs & groupes

Création des comptes

- La commande `/usr/sbin/useradd`
 - Ajoute un utilisateur
 - Commande `useradd`
 - Exemple `useradd toto`
 - Valeurs par défaut avec la commande `useradd -D`
 - `GROUP=100`
 - `HOME=/home`
 - `INACTIVE=-1`
 - `EXPIRE=`
 - `SHELL=/bin/bash`
 - `SKEL=/etc/skel`
 - Aussi visible dans `/etc/default/useradd`
 - Ligne typique dans `/etc/passwd` :
- `vlaine:x:1000:1000:Vincent Laine:/home/vlaine:/bin/bash`
- **Nom d'utilisateur:x:UID:GID:description:home/directory:interpreteur**
- UID : utilisateur ID - unique à chaque user
- GID : Groupe ID

Gestion des utilisateurs & groupes

Commandes de création de compte

- ❖ -c → commentaire (Full Name)
 - ❖ -d → chemin vers le répertoire home
 - ❖ -g → (GID). Le GID doit déjà exister
 - ❖ -G → liste des groupes supplémentaires (séparés par une virgule)
 - ❖ -u → UID
 - ❖ -s → définit le shell par défaut
 - ❖ -e → date d'expiration du compte
 - ❖ -k → répertoire skel-e
 - ❖ -n → désactive le schéma UPG
 - ❖ -m crée en même temps le répertoire personnel
 - ❖ -f Définir la date d'expiration mot de passe
 - ❖ -p définir mot de passe
 - ❖ /etc/passwd - /etc/shadow
- `useradd -c « Yvan Dupont » -d /maison/yvandupont -g 1000 -s /bin/sh -e 2020-12-12 -f 45 -m -p doranco yvandupont2`
 - `adduser --home /home/toto/ --shell /bin/bash --uid 2003 toto`
 - → Ici nous ajoutons un utilisateur avec comme description « Yvan Dupont' », avec son dossier personnel dans /home/yvandupont, avec comme groupe ID 1000, avec comme interpréteur /bin/sh, le compte expire le 12 dec 2020, le mot de passe expire tout les 45 jours, le dossier personnel sera créé automatiquement (-m), avec comme mot de passe « doranco » et comme nom d'utilisateur « yvandupont2 »

Gestion des utilisateurs & groupes

Commandes de création de compte

- useradd - Ajouter un utilisateur avec comme description « Yvan Dupont », avec son dossier personnel dans /home/yvandupont, avec comme uid 2003, avec comme interpréteur /bin/sh, le compte expire le 12 dec 2021, le mot de passe expire tout les 45 jours, le dossier personnel sera créé automatiquement (-m), avec comme mot de passe « azerty » et comme nom d'utilisateur « yvandupont »
 - `useradd -c "Yvan Dupont" -d /home/yvan -u 2003 -s /bin/sh -m -e 2021-12-12 -f 45 -p azerty yvandupont`
- adduser - Ajouter un utilisateur avec comme description « Gerard », avec son dossier personnel dans /maison/gerard, avec comme uid 2006, avec comme interpréteur /bin/bash, le dossier personnel sera créé automatiquement, avec comme mot de passe « azerty » et comme nom d'utilisateur « gerard ». Une description complète devra être ajoutée.

Gestion des utilisateurs & groupes

Fonctionnement des comptes

- Avec un nouveau mot de passe
 - `passwd login-name mot-de-passe`
- Les noms de tous les utilisateurs du système sont stockés dans
 - `/etc/passwd`
 - Structure
 - Login name
 - Password (or x if using a shadow file)
 - The UID
 - The GID
 - Text description for the user
 - The user's home directory
 - The user's shell
- Les mots de passe sont cryptés dans `/etc/shadow`
 - Seul le root peut les voir
 - Le `etc/passwd` doit être lisible par tout le monde → `chmod 644`
 - Le `etc/shadow` doit être restreint → `chmod 600` ou `chmod 400`

Gestion des utilisateurs & groupes

Date d'expiration des comptes

- Par défaut un mot de passe est valide 9999 jours (MAX_DAYS par défaut)
 - L'utilisateur est averti 7 jours avant l'expiration
 - MIN_DAY
 - Nombre minimum de jours avant qu'un utilisateur puisse modifier son mot de passe (par défaut 0)
- Modification → outil chage
 - `chage [-l] [-m min_days] [-M max_days] [-W warn] [-I inactive] [-E expire] [-d last_day] user`
 - Format date : YYYY-MM-DD
 - `chage -m 45 -M 90 -d 2021-12-12 -W 7 -I 15 -E 2021-31-12 gerard`

Gestion des utilisateurs & groupes

Modification des comptes

- usermod
 - -d → répertoire utilisateur
 - -g → GID initial de l'utilisateur
 - -l → nom de connexion de l'utilisateur
 - -u → UID
 - -s → Shell par défaut
- Bloquer/débloquer un utilisateur
 - passwd -l / passwd -u
 - usermod -L / usermod -U
 - Avec shadow → Remplacer le x par *
 - Supprimer un password avec « passwd -d »
 - Assigner /bin/false au shell par défaut dans /etc/passwd de l'utilisateur
- Supprimer un utilisateur :
 - userdel -r user1
 - L'option -r supprime aussi le dossier home de l'utilisateur
- Visualiser les utilisateurs connectés
 - Commande users
 - Commande who

Gestion des utilisateurs & groupes

Gestion des groupes

- Chaque nouvel utilisateur est assigné à un groupe primaire
- Il existe 2 conventions
 - Traditionnellement le groupe primaire est le même pour tous les utilisateurs et est appelé users (GID = 100)
 - User Private Group Scheme (UPG) introduit par RedHat □ chaque nouvel utilisateur appartient à son propre groupe primaire identique à son nom d'utilisateur (GID entre 500 et 60 000)
- Un utilisateur peut appartenir à plusieurs groupes
 - A chaque fois un seul groupe est effectif (par exemple lors de la création d'un fichier)
 - On peut obtenir la liste des groupes par utilisateur
 - Commande groups
 - Command id
- Créer un groupe
 - Commande groupadd <nomdugroupe>
- Supprimer un groupe
 - Commande groupdel <nomdugroupe>

Gestion des utilisateurs & groupes

Gestion des groupes

- Ajouter un utilisateur à un groupe
 - `gpasswd -a <user> <nom_group>`
 - `adduser user nom_group`
 - `usermod -aG group user`
- Enlever un utilisateur d'un groupe
 - `gpasswd -d <user> <group>`
- Les informations sur les groupes
 - `etc/group`
- Structure
 - Group name
 - The group password (or x if gshadow file exists)
 - The GID
 - A comma separated list of members
- gshadow
 - `/usr/sbin/grpconv` → crée le gshadow
 - `/usr/sbin/grpunconv` → supprime le gshadow
- Groupmod (modification d'un groupe)
 - `-g` → GID
 - `-n` → nom du groupe
- Le fait de changer le groupe par défaut
 - change le groupe effectif
 - Commande `newgrp <group>`
 - Le commutateur `-` démarre une nouvelle session

Gestion des utilisateurs & groupes

TP

1. Ajout d'un utilisateur pdupont (ID 3001)
 1. `adduser --gid 3001 pdupont`
 2. `useradd -g 3001 pdupont`
2. Ajout d'un utilisateur gparmesa (ID 3002)
 1. `adduser --gid 3002 gparmesa`
 2. `useradd -g 3002 gparmesa`
3. Ajout d'un groupe informatique (ID 4000)
 1. `addgroup --gid 4000 informatique`
 2. `groupadd -g 4000 informatique`
4. Ajout d'un groupe rh (ID 5000)
 1. `addgroup --gid 5000 rh`
 2. `groupadd -g 5000 rh`
5. Ajout des utilisateurs gparmesa et pdupont au groupe Informatique
 1. `adduser gparmesa informatique`
 2. `adduser pdupont informatique`
 3. `gpasswd -a gparmesa informatique`
 4. `gpasswd -a pdupont informatique`
6. Ajout de pdupont au groupe RH
 1. `adduser pdupont rh`
 2. `gpasswd -a pdupont rh`
7. Créer le fichier gshadow
 1. `grpconv`
8. Définir un mot de passe sur le groupe Informatique
 1. `groupmod -p azerty informatique`
9. Ajouter un groupe Error
 1. `addgroup error`
 2. `groupadd error`
10. Supprimer le groupe Error
 1. `delgroup error`
 2. `groupdel error`
11. Modifier l'ID du groupe Informatique en 4500
 1. `groupmod -g 4500 informatique`
12. Définir l'expiration de mot de passe avec chage avec un minimum, un maximum et un warning
 1. `chage -m 40 -M 80 -W 10 pdupont`

Gestions des droits

- Droits d'accès inspirés de la gestion des droits UNIX
- Pour chaque fichier, on attribue des droits à:
 - Propriétaire noté u (user)
 - Droits du groupe auquel le propriétaire appartient noté g (group)
 - Les autres noté o (others)
 - Tout le monde, noté a (all)
- Pour chaque type d'accès, on doit affecter des droits:
 - Droit de lecture (r)
 - Droit d'écriture (w)
 - Droit d'exécution (x)
 - Un « mode » correspond à un droit affecté a un type d'utilisateurs
- Les droits d'accès accordés aux types d'utilisateurs sont notés linéairement de gauche à droite:
- rwx|rwx|rwx : lecture / écriture /exécution pour
U G O

Gestion des droits

Modification des droits

- La modification de droits est effectuée via la commande `chmod [ugoa] [+ -=] nom_fichier`
- Les types d'utilisateurs
 - Le propriétaire du fichier (user)
 - Le groupe du propriétaire du fichier (group)
 - Les autres utilisateurs, ou encore le reste du monde (others)
- Les types de droits
 - r : droit de lecture (read) - Valeur octale 4
 - w : droit d'écriture (write) - Valeur octale 2
 - x : droit d'exécution (eXecute) - Valeur octale 1

Position binaire	Valeur octale	Droits	Signification
000	0	- - -	Aucun droit
001	1	- -x	Exécutable
010	2	- w -	Ecriture
011	3	- w x	Ecrire et exécuter
100	4	r - -	Lire
101	5	r - x	Lire et exécuter
110	6	r w -	Lire et écrire
111	7	r w x	Lire écrire et exécuter

Gestion des droits

- A la gauche des permissions, un caractère supplémentaire est présent:
- Via la commande `ls -l` :
 - `-` : indique qu'il s'agit d'un fichier
 - `d` : indique qu'il s'agit d'un répertoire
 - `l` : indique qu'il s'agit d'un lien (Hardlink ou Softlink)

```
527654 4 -rw-r--r-- 1 andn andn 807 Feb 25 2020 .profile
526377 4 -rwxr-xr-x 1 root root 3939 Feb 15 17:12 S01ssh
526381 0 lrwxrwxrwx 1 root root 6 Feb 15 17:14 S01ssh.link -> S01ssh
527734 0 -rw-r--r-- 1 andn andn 0 Feb 10 15:50 .sudo_as_admin_successful
526379 4 drwxr-xr-x 2 root root 4096 Feb 15 17:13 test
```

Type d'utilisateurs	Propriétaire	Groupe	Les autres
Droits	r w x	r - x	- - x
Position Binaire	111	101	001
Valeur Octale	7	5	1

Gestion des droits

- La modification de droits d'accès peut revêtir trois aspects:
 - L'ajout de droit (+)
 - La suppression de droits (-)
 - La fixation de droits (=)
- Changer les droits :
 - Commande « chmod »
 - Exemple:
 - `chmod u+x fichier.txt` ajoute les droits d'exécution au propriétaire sans toucher aux autres droits
 - `chmod u=rwx,g=rw;o=r fichier.txt`
- Changer le propriétaire d'un fichier:
 - `chown <utilisateur> <fichier>`
 - `chown :<groupe> <fichier>`
- Changer à la fois le propriétaire et le groupe:
 - `chown utilisateur:groupe fichier`
- Changer le propriétaire d'un dossier et des sous dossiers et fichiers
 - `chown -R <utilisateur> <dossier>`
- Changer le groupe propriétaire:
 - `chgrp <groupe> <fichier>`

Gestion des droits

TP

1. Aller votre dossier utilisateur (root ou user)
2. Créer un dossier « dr_test »
3. Créer un fichier « fr_test »
4. Dans le dossier « dr_test », créer un fichier « fr2_test »
5. En utilisant uniquement les droits « ugoa »
 1. Ajout des droits en lecture/exécution pour u (dr_test)
 1. `chmod -R u=rx dr_test/`
 2. Ajout des droits lecture/écriture pour g (fr2_test)
 1. `chmod g=rw dr_test/fr2_test`
 3. Ajout des droits pour u et g en lecture/écriture (fr_test)
 1. `chmod ug=rw fr_test`
 4. Ajout des droits lecture pour o (fr_test)
 1. `chmod o=r fr_test` (ou `o+r`)
 5. Ajout des droits lecture pour o (dr_test)
 1. `chmod -R o=r dr_test/`
 6. Changer le propriétaire pour le compte utilisateur (dr_test)
 1. `chown -R vlaine dr_test/`
 7. Changer le groupe pour le groupe root (fr2_test)
 1. `chown -R :vlaine dr_test/`
 2. `chgrp -R vlaine dr_test/`

Gestion des droits

TP

En utilisant les valeurs octales :

1. Lecture, écriture pour le propriétaire / Lecture groupe / Lecture pour les autres
2. Lecture, écriture pour tout le monde
3. Lecture, écriture, exécution juste pour le propriétaire
4. Le propriétaire à tous les droits / Le groupe aucun / Les autres lire et exécuter
5. Le propriétaire à tous les droits / Les autres lire et exécuter
6. Tous droits pour le propriétaire / Lecture, écriture pour le groupe / Lecture seule pour les autres
7. Tous les droits pour tous

Gestions des droits

TP

En utilisant les valeurs octales :

1. Lecture, écriture pour le propriétaire / Lecture groupe / Lecture pour les autres
 1. 644
2. Lecture, écriture pour tout le monde
 1. 666
3. Lecture, écriture, exécution juste pour le propriétaire
 1. 700
4. Le propriétaire à tous les droits / Le groupe aucun / Les autres lire et exécuter
 1. 705
5. Le propriétaire à tous les droits / Les autres lire et exécuter
 1. 755
6. Tous droits pour le propriétaire / Lecture, écriture pour le groupe / Lecture seule pour les autres
 1. 764
7. Tous les droits pour tous
 1. 777

Droits spéciaux

Exécuter un fichier avec les droits de son propriétaire:

Chmod u+s fichier (s correspond au setuid bit)

En octal : chmod 4777 (4 correspond au setuid bit, le 777 est un exemple de droits)

Exécuter un fichier avec les droits du groupe:

Chmod g+s fichier (g correspond au setgid bit)

En octal : chmod 2777 (2 correspond au setgid bit, le 777 est un exemple de droits)

Le sticky bit rend un fichier « sticky », c'est-à-dire qu'il persiste en mémoire après son exécution ou après son ouverture. De plus il peut-être accessible en 777 mais ne peut être supprimé que par son propriétaire.

chmod 1777 (u+t) fichier (1 correspond au sticky bit, le 777 est un exemple de droits)

Droit UMASK

Le UMASK (user file creation mode mask, masque de création de fichier par l'utilisateur) définit les permissions par défaut affectées aux fichiers et dossiers.

Par défaut il est égal a 022 ou 0022

Pour le définir: `umask « valeur »`

`umask -S` : voir les permissions par défaut

Vérifier sa valeur dans `/etc/login.defs` (éventuellement dans `/etc/profiles`)

Gestion des médias

- Les périphériques de stockage sont placés dans le répertoire /dev
 - Lecteur de disquettes 1: fd0
 - Lecteur de disquettes 2: fd1
- Disque dur maître: sda/hda
 - Partition 1: sda1, Partition 2: sda2,....
- Disque dur esclave: sdb
 - Lettre suivante pour les autres périphériques
 - Utilisez dmesg pour identifier le nom de votre clé usb
- Partitionnement
 - fdisk
 - fdisk /dev/sdb
 - cfdisk
 - Cfdisk /dev/sdc

Gestion des médias

- Un filesystem journalisé note toutes les transactions à venir avant de les exécuter
- En cas de crash, le système peut savoir ce qui a été fait et ce qui ne l'a pas été
- La journalisation procure donc quelques avantages :
 - la durée d'un fsck au boot ne dépend plus de la taille du filesystem (on sait où chercher)
 - intégrité des données accrue (et paramétrables)
- ext3/ext4/jfs/xfs/ReiserFS/Reiser4 sont journalisés

Gestion des médias

- minix
 - fs original
 - simple, encore utilisé sur les disquettes
 - limité (64 Mb)
- ext2
 - développé en 1993
 - limité (2 Tb)
- ext3
 - ext2 journalisé
 - Conserve un espace libre (5%)
- ext4
 - supporte jusqu'au Exbioect (2^{60})
 - en cours d'intégration au kernel
- jfs/xfs
 - journalisés
 - respectivement IBM/SGI
 - Donne la totalité de l'espace
- ReiserFS/Reiser4
 - journalisé
 - 10 fois plus performant sur les petits fichiers
 - problème de pérennité ?
- iso9660/udf
 - respectivement CDRoms/DVDRoms
- fat/vfat/ntfs
 - fat/vfat bien gérés
 - support ntfs plus délicat (ro)
- unionfs
 - agrégat de filesystems différents
 - mélange possible entre ro/rw
- nfs/smbfs/coda
 - filesystems réseau

Gestion des médias

Le fichier `/etc/fstab` contient les points de montage systèmes.

- Pour chaque filesystem à monter, `/etc/fstab` contient :
 1. la partition à monter (`/dev/sda1`)
 2. le point de montage (`/var`)
 3. le type de filesystem (`reiserfs`, `ext3`, ...)
 4. les options de montage (lecteur seule, propriétaire, etc...)
 5. un booléen à 1 si le fs doit être sauvegardé par dump (archivage)
 6. un numéro d'ordre pour la vérification de fs au boot

```
# Nom du périphérique point de montage du fs type options dump-freq pass-num
LABEL=/ / ext3 defaults 1 1
none /dev/pts devpts gid=5,mode=620 0 0
none /proc proc defaults 0 0
none /dev/shm tmpfs defaults 0 0

# disques amovibles
/dev/cdrom /mnt/cdrom udf,iso9660 noauto,owner,kudzu,ro 0 0
/dev/fd0 /mnt/floppy auto noauto,owner,kudzu 0 0

# partition NTFS de Windows (version Vista ou autre) sur un multiboot
/dev/hda1 /mnt/WinVista ntfs-3g defaults 0 0

# Le swap de linux
/dev/sda1 swap swap defaults 0 0

# Une partition FAT que linux et Windows peuvent lire et écrire
/dev/hda5 /mnt/shared vfat umask=000 0 0
```

Gestion des médias

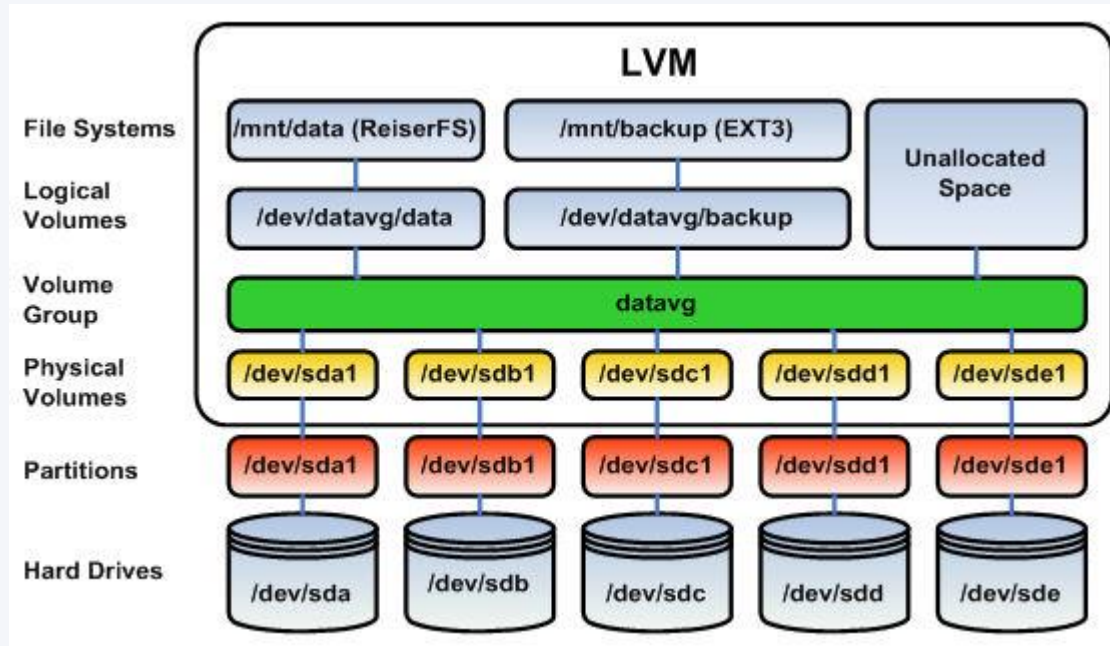
- {u,}mount : {dé,}montage des partitions sur le filesystem
- mount <device> <mountpoint>
- umount <device|mountpoint>

Les options communes à tous les types de systèmes de fichiers sont :

ro / rw	Montage en lecture seulement/lecture-écriture
suid / nosuid	Autorise ou interdit les opérations sur les bits suid et <i>sgid</i>
dev / nodev	Interprète/n'interprète pas les périphériques caractères ou les périphériques blocs spéciaux sur le système de fichiers
exec / noexec	Autorise ou interdit l'exécution de fichiers binaires sur ce système de fichiers
auto / noauto	Le système de fichiers est (c'est l'option par défaut) / n'est pas monté automatiquement
user / nouser	Permet à tout utilisateur / seulement à root (C'est le paramétrage par défaut) de monter le système de fichiers correspondant
sync / async	Selon cette valeur, toutes les entrées/sorties se feront en mode synchrone ou asynchrone
defaults	Utilise le paramétrage par défaut (c'est équivalent à <i>rw, suid, dev, exec, auto, nouser, async</i>)

Gestion des disques & LVM

- LVM (Local Volume Management) Fournit une abstraction du matériel de stockage :
 - on travaille sur des volumes logiques et non plus sur des partitions
 - on travaille sur des volume groups au lieu de disques
 - on peut ajouter des partitions dans des volume groups
 - on peut agrandir les volumes logiques si nécessaire
- LVM permet au final de construire des systèmes de fichiers sur des devices ayant des tailles modulables
- Le système de fichier doit supporter le redimensionnement pour en profiter !



Les blocs

- Lors d'un formatage → Division des secteurs en petit groupes appelés blocs.
- Une taille de bloc peut être spécifié.
 - `mkfs -t ext3 -b 4096 /dev/sda1`
- Les tailles :
 - Ext2 peut être de 1 Ko, 2 Ko, 4 Ko, 8 Ko
 - Ext3 peut être de 1 Ko, 2 Ko, 4 Ko, 8 Ko
 - Ext4 peut être comprise entre 1 Ko et 64 Ko
- Impact :
 - Taille maximal d'un fichier
 - Taille maximal du système de fichier
 - Performances
- A savoir:
 - Plus les blocs sont petits plus la lecture d'un gros fichier prendra du temps
 - Avantageux d'utiliser des petits blocs pour de petits fichiers
 - Moins d'IOPS sera exécuté si vous avez une plus grande taille de bloc pour votre système de fichiers.

Les blocs

Constitué de blocs de 1024 octets

3 types de blocs

- Superblocks (stock les métadonnées du système)

 - Répétés tous les 8193 blocs, informations sur la taille des blocs, les inodes libres, dernier moment de montage...

- Inodes (stock les métadonnées des fichiers)

 - Pointeurs vers des blocs de données.

 - Chaque inode fait 256 octets et contient les « user, group, permissions et time-stamp » associés aux données

Les blocs de données

- Fichiers et répertoire contenant les données

Gestion des disques & LVM

- Lister les partitions d'un disque `/dev/sda`
 - `fdisk -l /dev/sda`
 - `fdisk -l` (voir toutes les partitions)
- Partitionner un disque `/dev/sda`
 - `fdisk /dev/sda`
- Déclarer des PV (Physical Volume)
 - `pvcreate /dev/sda /dev/sdb /dev/sdc`
 - Visualiser : `pvdisplay` ou `pvs`
- Créer un VG (Volume Group)
 - `vgcreate VGDATA /dev/sda1 /dev/sdb1`
 - Visualiser : `vgdisplay` ou `vgs`
- Créer un LV (Logical Volume)
 - `lvcreate -n [Nom_LV] -L [Taille] [nom_VG]`
 - Visualiser : `lvdisplay` ou `lvs`
- Création de partition
- Formater en ext4
 - `mkfs.ext4 /dev/ext00-vg/lv01` (chemin)
- Formater en xfs
 - `mkfs.xfs /dev/sda1`
- Montage du LV
 - `mkdir /mnt/lv01`
 - `mount /dev/ext00-vg/lv01 /mnt/lv01`

Gestion des disques & LVM

- Augmenter la taille du VG après ajout d'un disque :
 - `pvcreate [nouveau_disque]`
 - `vgextend [nom_VG] [Nom_du_nouveau_disque]`
 - `vgextend ext00-vg /dev/sdb`
- Ensuite vérifier l'état de son LV
 - `e2fsck -f [chemin de votre lv]` (visible avec `fdisk -l`)
 - `e2fsck -f /dev/mapper/ext00--vg-lv01`
- Puis redimensionner le LV
 - `lvresize -L [nouvelletaille] [chemin de votre LV]`
 - `lvresize -L 12G /dev/ext00-vg/lv01`
- Et Redimensionner la partition
 - `resize2fs [chemin du LV]`
 - `resize2fs /dev/mapper/ext00--vg-lv01`
- Vérifier avec `df -h`

Gestion des disques & LVM

- Diminuer la taille d'un LV

Attention : Si vous diminuez votre LV avant la partition contenu à l'intérieur vous pouvez la détruire!

- D'abord démonter la partition
 - `umount [votre_partition]`
 - `umount /mnt/lv01`
- Vérifier l'état de la partition :
 - `e2fsck -f /dev/mapper/ext00--vg-lv01`
- Réduire la taille de la partition (Avant de diminuer le LV !)
 - `resize2fs /dev/mapper/ext00--vg-lv01 2G`
- Puis réduire le LV
 - `lvreduce -L <la taille voulue> <le chemin>`
 - `lvreduce -L 2G /dev/ext00-vg/lv01`
 - Confirmer avec « y » à la question
- Puis remonter le volume
 - `mount /dev/ext00-vg/lv01 /mnt/lv1/`
 - `df -h` pour vérifier : (ici nous avons descendu la partition à 2GO.)

Gestion des disques & LVM

- df : donne l'occupation des filesystems montés
 - `df -h | grep "^/dev/"`
- hdparm : tuning du disque (DMA, energie, ...)
 - `sudo hdparm /dev/sda`
- tune2fs : tuning du filesystem (max-mount, UUID)
 - `sudo tune2fs l /dev/sda2`
- fsck : vérification du filesystem (boot single-user)
 - `fsck a /dev/hda6`

Gestion des disques & LVM

- Filesystems spéciaux
- procfs
 - monté dans /proc
 - contient des informations sur les processus
 - zone fourre-tout pour les variables exportées du kernel
- sysfs
 - monté dans /sys
 - contient des informations sur les devices présents

Gestion des disques & LVM

- Le SWAP
- L'espace de swap permet de décharger temporairement la mémoire physique (RAM)
- La taille empirique recommandée est $2 * \text{RAM}$
- Lorsqu'un process n'utilise pas une zone de sa mémoire («page»), le kernel peut décider de la mettre dans le swap («swap out»)
- Lorsque la mémoire physique se fait rare, les zones nécessaires aux process :
 - doivent être stockées sur le swap («swap out»)
 - puis relues depuis le swap lorsque le process s'exécute («swap in»)

Gestion des disques & LVM

Exemple de partitionnement

```
Storage configuration [ Help ]

[ /var          1.000G  new xfs  new LVM logical volume  ▶ ]
[ SWAP          2.000G  new swap new LVM logical volume  ▶ ]

DISQUES DISPONIBLES

PÉRIPHÉRIQUE          TYPE          TAILLE
[ ubuntu-vg (new)     LVM volume group  18.996G ▶ ]
espace libre          4.000G

[ Create software RAID (md) ▶ ]
[ Create volume group (LVM) ▶ ]

USED DEVICES

PÉRIPHÉRIQUE          TYPE          TAILLE
[ ubuntu-vg (new)     LVM volume group  18.996G ▶ ]
root-lv               new, to be formatted as ext4, mounted at /  8.996G ▶ ]
swap                 new, to be formatted as swap                2.000G ▶ ]
home-lv               new, to be formatted as xfs, mounted at /home 2.000G ▶ ]
var-lv                new, to be formatted as xfs, mounted at /var 1.000G ▶ ]
tmp-lv                new, to be formatted as xfs, mounted at /tmp 1.000G ▶ ]

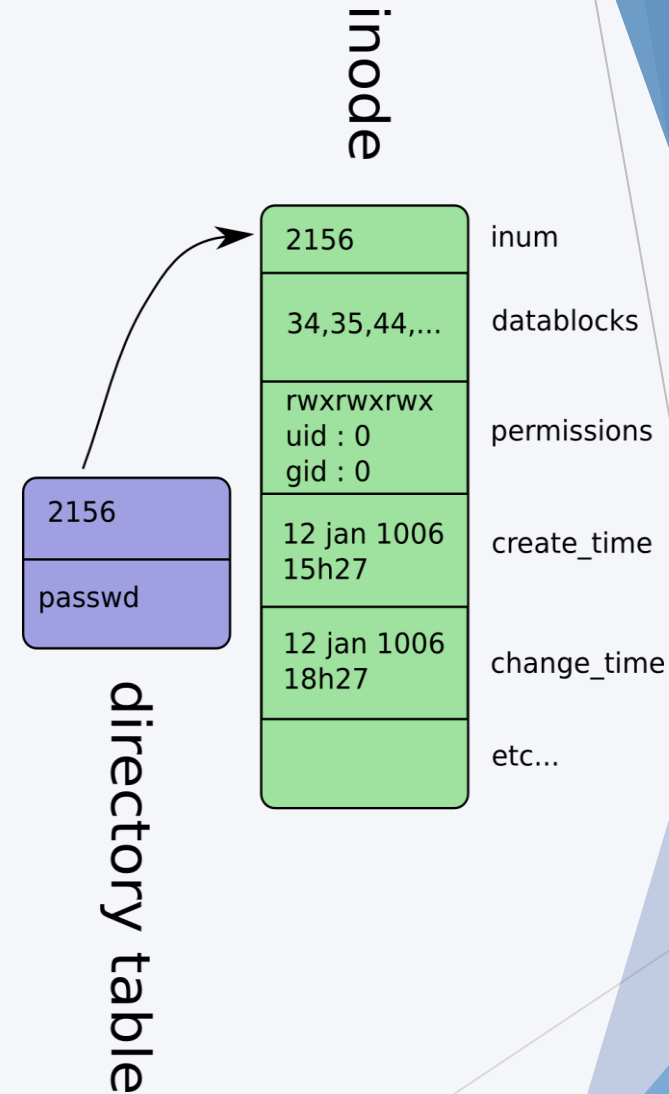
[ /dev/sda            disque local          20.000G ▶ ]
partition 1           new, bios_grub                            1.000M ▶ ]
partition 2           new, to be formatted as ext4, mounted at /boot 1.000G ▶ ]
partition 3           new, PV of LVM volume group ubuntu-vg 18.997G ▶ ]

[ Terminé ]
[ Rétablir ]
[ Retour ]
```

Gestion des disques & LVM

Les inodes

- Dans un fs unix, chaque fichier est représenté par un inode
- Le fs contient une table d'association
 - "nom de fichier" \Leftrightarrow "inode"
- L'inode contient toutes informations nécessaires concernant le fichier :
 - numéro d'inode
 - permissions, propriétaire
 - dates (création, accès, modification)
 - références vers les blocs de données



Inodes

Inodes (Index nodes)

Pointeurs vers des blocs de données.

Chaque inode fait 256 octets et contient:

L'identifiant du propriétaire (UID)

L'identifiant du groupe propriétaire (GID)

Les droits (rwxrwxrwx)

La taille du fichier

La date de dernier accès au fichier

La date de dernière modification du fichier

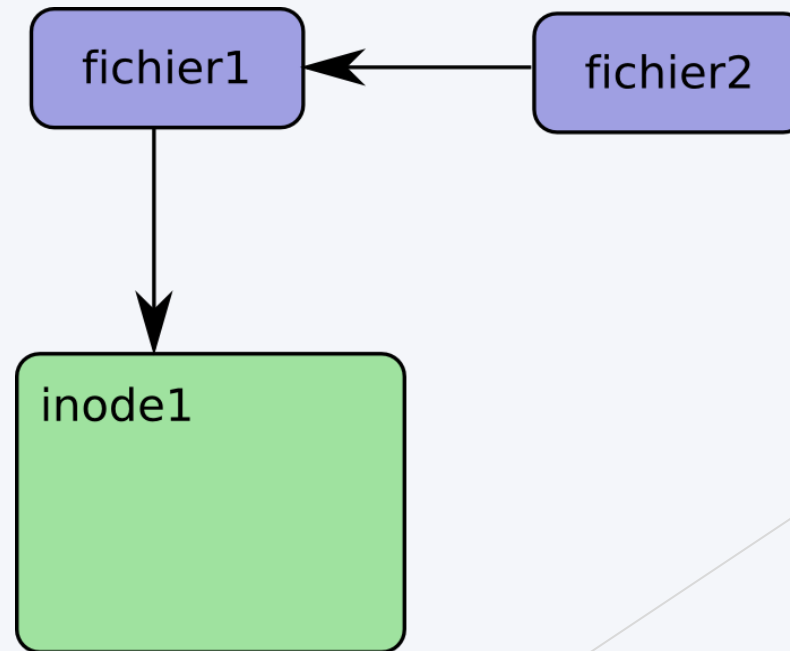
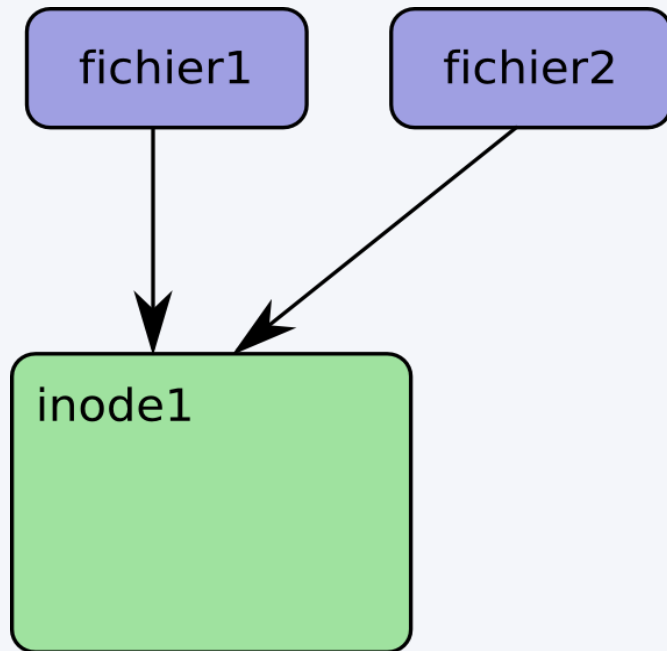
Le nombre de liens

La table des blocs

- Retrouver un fichier avec son inode :
 - `find / -inum <inode>`
- Recherche un fichier par son nom
 - `find / -name <nom du fichier>`

Gestion des disques & LVM

- Un hard link est simplement une entrée de la DT qui pointe vers un inode
- Il peut y avoir plusieurs hard-links sur un inode (et donc plusieurs « noms » pour un même contenu)
- Les inodes existent tant qu'il y a au moins un hard-link pointant dessus
- Les soft links (symbolic links / symlinks) pointent vers un autre « fichier » de la DT
- Le lien symbolique n'existe que par l'objet qu'il pointe : sans lui, plus de fichier



Gestion des disques & LVM

Utilisation des liens

- - hard links
 - ne peuvent fonctionner que sur un même filesystem (la tables des inodes est spécifique au filesystem)
 - performances maximum (rien ne distingue un hard link d'un autre)
- soft links
 - peuvent fonctionner entre les filesystems
 - impact sur les performances (indirection)
 - un symlink n'est rien sans son hard-link !
- « ln » permet de créer des liens
 - ln original destination
 - crée un hard link (destination) pointant sur l'inode de original
- « ln -s » original destination
 - crée un lien symbolique destination pointant sur original
- « stat » permet de connaître toutes les infos d'un inode
 - stat fichier
 - affiche le contenu de l'inode pointé par fichier

Périphérique de stockage

Nommage

Les périphériques de stockage sont placés dans le répertoire /dev

Lecteur de disquettes 1: fd0

Lecteur de disquettes 2: fd1

Disque dur maître: sda

Partition 1: sda1, Partition 2: sda2,....

Disque dur esclave: sdb

Lettre suivante pour les autres périphériques

Utilisez dmesg pour identifier le nom de votre clé usb

Gestion des processus

- Chaque exécutable, script, démon, apparaît comme un processus sur le système lorsqu'il est lancé
- Lorsqu'un processus est lancé, le kernel lui attribue un numéro (PID)
- La commande jobs ne permet de voir que les processus exécutés dans le terminal courant
- Pour visualiser les autres processus, il faut utiliser « ps »
- Voir tous les processus
 - ps -ef
- Voir tous les processus, un peu plus de détail
 - ps -awux
- Voir les processus de l'utilisateur courant
 - ps -wux

Mnémoniques

PID	: ID du process
PPID	: ID du process parent
VSZ	: consommation totale du process (RAM+SWAP)
RSS,RES	: taille occupé sur la mémoire physique (RAM)
TTY	: terminal associé
STAT	: état (R=runnable, S=sleep, T=stopped, Z=zombie)
START	: heure de lancement
TIME	: cumul temps CPU consommé

Gestion des processus

- Voir tous les processus en arbre
 - `ps -ejH`
- Voir tous les processus lancé par root
 - `ps -u root`
- Liste dynamique des processus (contrairement à `ps` qui est statique)
 - `top`
 - `q` = ferme `top`
 - `h` = aide
 - `f` = ajoute ou supprime des colonnes
 - `u` = filtre par user

Gestion des processus

Gérer les processus : `kill`, `killall`

- *kill* permet d'envoyer un signal à un processus
- ce processus peut intercepter ce signal et agir en conséquence
- *kill* n'est pas forcément malfaisant !
- *man kill* donne la liste de signaux possibles
- *killall* permet d'envoyer un signal à des processus par leur nom
- un utilisateur ne peut signaler que ses processus
- attention : *kill* est souvent un built-in (et non `/bin/kill`), donc `help kill` et non `man kill` pour de l'aide !

`kill signal pid`

`killall signal nom`

- `pid` : numéro du processus
- `signal` : numéro ou nom du signal (1 envoie le signal à tous les processus)

Gestion des processus

Gérer les processus : `kill`, `all`

- Signaux les plus courants
 - STOP/CONT : équivalents à <Ctrl>-z et fg/bg pour les process du terminal
 - INT (2) : arrêt demandé par l'utilisateur (généralement via <Ctrl-C>)
 - TERM (15) : kill demande gentiment au process de se terminer
 - KILL (9) : le processus est tué sans sommation
 - HUP (1) : signal de déconnexion du terminal, maintenant surtout utilisé pour demander une reconfiguration
 - USR1 : généralement utilisé pour demander une reconfiguration
- `kill PiD`
- `pkill Name_of_process`
- `jobs`

Gestion des processus

Gérer les processus : kill, all

- kill PID = Arrête proprement le processus
- kill -9 PID = Force l'arrêt d'un processus
- killall = Arrête tous les processus qui ont le même nom
 - killall -KILL [process name]
 - killall -SIGKILL [process name]
 - killall -9 [process name]
 - killall -l : Affiche la liste des signaux disponibles

Gestion des processus

Le job control

- Fonctionnalité interne au shell
- Permet de gérer des tâches (processus) multiples dans un même shell :
 - suspension temporaire : le processus est arrêté jusqu'à nouvel ordre
 - arrêt définitif : le processus est terminé
 - mise en arrière/avant plan : le processus reçoit l'entrée clavier
- Un processus peut ignorer l'arrêt définitif, mais pas la suspension
- Suspendre (stopper) un processus : `<Ctrl>z`
- Arrêter (terminer) un processus : `<Ctrl>c`
 - (si le processus est à l'avant plan)
- Arrêter (terminer) un processus : `kill %n`
 - (si le processus est à l'arrière plan)
- Voir la liste des processus du terminal (jobs) : `jobs`
 - '+' est le job « courant »
 - '-' est le job « précédent » (%)
- Mettre un job en arrière plan : `bg %n`
- Mettre un job en avant plan : `fg %n` ou `%n`
- Lancer un job en arrière plan : commande `&`

Gestion des processus

TP job control

1. Depuis un interpréteur de commande, faire : `man kill`
2. Puis interrompre ce processus dans l'interpréteur de commande par `Control-Z`
3. Recommencer avec `man cat`
 1. Que retourne la commande `jobs` ?
4. Taper `fg` qui remet le processus courant interrompu en premier plan, et lire cette page. Interrompre la lecture avec `Ctrl-Z`
5. Faire revenir la page manuel de `kill` au premier plan en utilisant `fg %n` où `n` est le numéro entre crochet de ce processus. Interrompre la lecture avec `Ctrl-Z`.
6. Tapez `sleep 200 &` pour lancer le processus en arrière plan
7. Remettre le processus en premier (`fg`) plan puis le suspendre (`ctrl + ?`)
8. Puis relancer le processus en arrière plan (`bg`)

Crontab

- cron permet de programmer des tâches récurrentes sur le système
- ces tâches sont listées dans des «crontabs»
- chaque utilisateur possède sa propre crontab
- il y a une crontab système
- ces fichiers sont scrutés par le démon cron/crond/anacron chaque minute (pas besoin de redémarrer le démon)
- cron exécute une tâche si son heure est venue
- grâce à cron, on peut automatiser la rotation des logs, la mise à jour de bases de données, les backups, la génération d'index... à une heure/date fixe

Crontab

cron peut être configuré de multiples manières

- `/etc/crontab`
 - crontab système; contient les programmations globales du système
- `/var/spool/cron/crontabs/$USER`
 - contient la crontab de \$USER
- `/etc/cron.d/`
 - contient des 'mini-crontabs' ajoutées par les packages à l'installation
- `/etc/cron.{hourly,daily,weekly,monthly}`
 - contient des scripts exécutés respectivement toutes les heures/jours/semaines/mois

Crontab

- min heure jourmois mois jour_semaine user command
 - (le champ user n'existe que dans la crontab système)
 - min : à quelle minute de l'heure [0-59]
 - heure : à quelle heure [0-23]
 - jourmois : quel jour du mois [1-31] mois : quel mois de l'année [1-12]
 - joursemaine : quel jour de la semaine [0-7], 0=7=dimanche
 - user : sous quel utilisateur
 - command : commande à exécuter

```
mm hh jj MMM JJJ [user] tâche > log
```

- Pour chaque unité, on peut utiliser les notations suivantes :
 - 1-5 : les unités de temps de 1 à 5.
 - */6 : toutes les 6 unités de temps (toutes les 6 heures par exemple).
 - 2,7 : les unités de temps 2 et 7.

Crontab

Utiliser la commande crontab

- Pour afficher le contenu du fichier crontab :
 - `crontab -l`
- Pour supprimer toutes les actions du fichier crontab :
 - `crontab -r`
- Pour éditer les actions du fichier crontab :
 - `crontab -e`
- Le crontab s'ouvre avec un éditeur par défaut que vous pouvez sélectionner à la première ouverture. Si on veut le changer ultérieurement :
 - `export EDITOR=vim`
 - `crontab -e`

Crontab

- crontab -e = Edition du crontab système

```
# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .---- day of week (0 -
6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * user command to be executed
```

- Chaque ligne correspond à une action.
- `mm hh jj MMM JJJ [user] tâche > log`
- mm : minutes (00-59).
- hh : heures (00-23) .
- jj : jour du mois (01-31).
- MMM : mois (01-12 ou abréviation anglaise sur trois lettres : jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec).
- JJJ : jour de la semaine (1-7 ou abréviation anglaise sur trois lettres : mon, tue, wed, thu, fri, sat, sun).
- user (facultatif) : nom d'utilisateur avec lequel exécuter la tâche.
- tâche : commande à exécuter.
- > log (facultatif) : redirection de la sortie vers un fichier de log. Si un fichier de log n'est pas spécifié, un mail sera envoyé à l'utilisateur local.

Crontab

Exemples :

- Exécution tous les jours à 22h00 d'une commande et rediriger les infos dans sauvegarde.log :
 - `00 22 * * * /root/scripts/sauvegarde.sh >> sauvegarde.log`
- Exécution d'une commande toutes les 6 heures :
 - `00 */6 * * * /root/scripts/synchronisation-ftp.sh`
- Exécution d'une commande toute les heures :
 - `00 */1 * * * /usr/sbin/ntpdate fr.pool.ntp.org`
- Exécution d'une commande toutes les minutes uniquement les lundis :
 - `* * * * 1 /root/script/commandes-du-lundi.sh`
- Exécution d'une commande une fois par an à une heure précise (ici le 25 décembre à 00h15) :
 - `15 00 25 12 * echo "Le père Noël est passé !"`
- Exécuter chaque jour, de chaque mois à 2:15 la commande eix-sync
 - `15 02 * * * /usr/bin/eix-sync`
- Exécuter une commande au redémarrage de l'ordinateur, pour un utilisateur spécifique :
 - `@reboot root /root/scripts/synchronisation-dns.sh`

Archivage et sauvegarde

Définition

- Compression : Réduire la taille d'un fichier par algorithme de compression.
- Archivage : Placer un ensemble de fichiers et/ou de dossiers dans un seul fichier.
- Compression sans archivage : gzip/gunzip, bzip2/bunzip2
- Archivage avec ou sans compression : tar, star

-
- Rappel sur « for »: Bouclage sur une liste de valeurs
 - La boucle for permet de parcourir une liste de valeurs et de boucler autant de fois qu'il y a de valeurs
 - POUR variable PRENANT valeur1 valeur2 valeur3
 - FAIRE
 - -----> effectuer_une_action
 - VALEUR_SUIVANTE

```
#!/bin/bash
for variable in 'valeur1' 'valeur2' 'valeur3'
do
    echo "La variable vaut $variable"
done
```

La variable vaut valeur1
La variable vaut valeur2
La variable vaut valeur3

Archivage et sauvegarde

tar

- Le tar, {g,b} & zip
- tar permet de créer un fichier contenant d'autres fichiers
- Il est souvent utilisé en combinaison avec gzip ou bzip2
- tar est parfois utilisé comme système de gestion de paquetage dans certaines distributions (Slackware)
- Créer une archive archive.tar contenant *fichiers*
 - tar cvf archive.tar fichiers
- Créer une archive tar compressée gzip archive.tar.gz contenant *fichiers*
 - tar cvzf archive.tar.gz fichiers
- Créer une archive tar compressée bzip2 archive.tar.bz2 contenant *fichiers*
 - tar cvjf archive.tar.bz2 fichiers
- Lister le contenu de l'archive
 - tar tvf archive.tar.gz
 - tar tvzf archive.tar.gz
 - tar tvjf archive.tar.bz2
- Extraire une archive tar
 - tar xvf archive.tar.gz -C /dossier_de_destination
 - tar xvzf archive.tar.gz
 - tar xvjf archive.tar.bz2

Archivage et sauvegarde

gzip

- L'utilitaire gzip est basé sur l'algorithme Deflate (combinaison des algorithmes LZ77 et Huffman). C'est la méthode de compression la plus populaire sous GNU/Linux.
- Compresser un fichier (le fichier est remplacé par son format compressé) :
 - `gzip mon_fichier`
- Décompresser un fichier gzippé :
 - `gunzip mon_fichier_compresse.gz`
 - `gzip -d mon_fichier_compresse.gz`
- Compresser un fichier de façon optimisée :
 - `gzip -9 mon_fichier`
- Compresser plusieurs fichiers en un :
 - `gzip -c fichier1 fichier2 > fichier_compress.gz`

Archivage et sauvegarde

bzip2

- bzip2 est une alternative à gzip, plus efficace mais moins rapide.
- Compresser un fichier :
 - `bzip2 fichier`
- Décompresser un fichier bzippé :
 - `bunzip2 fichier_compress.bz2`

Archivage et sauvegarde

zip

- On utilise les commandes zip et unzip.
- Création ZIP
 - `zip votre_archive.zip [liste des fichiers]`
 - `zip -r votre_archive.zip [dossier]`
- Afin de compresser plusieurs sous-dossiers séparément (bash) :
 - `for f in * ; do zip "$f.zip" "$f"/* ; done`
- Compresser en plusieurs fichiers :
 - `zip -s 2m wordpress.zip --out dvd-split.zip`
- **`zip -e votre_archive.zip [liste des fichiers]` chiffre le zip et demande un mot de passe.**
- Extraction unzip
 - `unzip votre_archive.zip -d mon_repertoire`
- Extraction de plusieurs fichiers .zip d'un même dossier :
 - `for f in *.zip ; do unzip "$f" ; done`
- Archives zip découpées
 - Les archives zip sont parfois découpées : `archive.z01`, `archive.z02`, `archive.zip`. Il faut rassembler les fichiers dans une seule archive, puis extraire cette dernière :
 - `cat archive.z* > archive_globale.zip`
 - `unzip archive_globale.zip`

Archivage et sauvegarde

Sauvegarde avec dd

- Attention: Lorsque vous utilisez la commande dd, soyez prudent vous pouvez perdre des données en cas d'erreur !
- Pour sauvegarder une copie complète d'un disque dur vers un autre disque dur connecté au même système, exécutez la commande dd.
- `dd if=<source> of=<cible> bs=<taille des blocs> skip= seek= conv=<conversion>`
- S'il y a une erreur, la commande dd s'arrête (abandon).
 - source (if) = D'où proviennent les données à copier;
 - cible (of) = où seront copiées les données traitées par la commande;
 - bs = taille des blocs, de puissance 2, par défaut égale à 512 octets;
 - skip = Ignorer le nombre indiqué de blocs (dont la taille est fournie par ibs) au début de la lecture;
 - seek = Ignorer le nombre indiqué de blocs (dont la taille est fournie par ibs) au début de l'écriture.
 - conv = Modifier le fichier comme indiqué par l'argument conversion, qui peut prendre les valeurs suivantes (pas d'espace autour des virgules lorsque plusieurs arguments sont fournis) :
 - ascii, ebcdic, ibm, block, unblock, lcase, ucase, swab, noerror, notrunc, sync.

Archivage et sauvegarde

Sauvegarde avec dd

- `dd if=/dev/sda1 of=/dev/sdb1 conv=noerror,sync`
- Si vous donnez le paramètre « conv = noerror », alors dd continue à copier même lorsque des erreurs de lecture sont rencontrées.
- Le fichier d'entrée et de sortie doivent être mentionnés très soigneusement, si vous mentionnez un périphérique source à la place de la cible et vice-versa, vous risquez de perdre toutes vos données.
- Lors d'une copie partition à partition avec la commande dd, l'option sync permet de synchroniser les Entrée/Sorties.
- Copier une partition autre vers une partition copie aussi l'UUID puisqu'il est inscrit dans la partition même.
 - blkid : voir les UUID des partitions
- Modifier l'UUID
 - `tune2fs -U random /dev/sdb1`

Archivage et sauvegarde

Sauvegarde avec dd

- Copie de l'intégralité d'un disque
- `dd if=/dev/sda of=/dev/sdb conv=noerror`
 - source est `/dev/sda`, et le nom du disque cible est `/dev/sdb`.
- Créer une l'image d'un disque dur
 - `dd if=/dev/sda of=~ /hdadisk.img`
- Restaurer à partir d'une image disque
 - `dd if=hdadisk.img of=/dev/sdb`
- Sauvegarder une partition dans un fichier image
 - `dd if=/dev/sda1 of=~ /partition1.img`
- Créer un iso à partir d'un Cdrom
 - `dd if=/dev/sr0 of=/cdrom.iso bs=2048`
 - Monter l'ISO
 - `mount -o loop -t iso9660 cdrom.iso /mnt/cd`
 - FSTAB : `/chemin/cd.iso /mnt/cd iso9660 rw,user,noauto 0 0`

Archivage et sauvegarde

Sauvegarde avec dd

- Sauvegarder le chargeur de boot (ne copiera que les 440 premiers octets)
 - `dd if=/dev/sda of=ChrgBoot.dd bs=440 count=1`
- Sauvegarder le MBR (ne copiera que les 512 premiers octets)
 - `dd if=/dev/sda of=MBR.dd bs=512 count=1`
- Sauvegarder un disque ou une partition dans un fichier compressé
 - `dd if=/dev/sda | gzip -c > /image.gz`
- Restaurer l'image compressée :
 - `gzip -cd /image.gz | dd of=/dev/sda`

Archivage et sauvegarde

cpio

- cpio est un utilitaire d'archivage ainsi qu'un format de fichier utilisé sur UNIX.
- À l'origine comme un moyen de sauvegarder des données sur bande magnétique sur les premières versions d'UNIX.
- Remplacé par tar mais utilisé par le RPM PM et initrd.
- Une archive cpio est un assemblage de fichiers et de répertoires à l'intérieur d'un seul fichier d'archive.(.cpio). Tout les fichiers doivent lus sur l'entrée standard, et l'archive est envoyé sur la sortie.
- Création d'une archive :
 - `find . | cpio -o [-cv] > archive`
- Vérification d'une archive :
 - `cpio -it [-cv] < archive [fichiers_a_verifier ...]`
- Extraction d'une archive :
 - `cpio -i [-cdv] < archive [fichiers_a_extraire ...]`
- Principales options :
 - `-c` : Création/vérification/extraction d'une archive ayant un format d'en-tête d'archive portable entre les systèmes UNIX
 - `-d` : Extraction avec création des répertoires s'ils n'existent pas
 - `-i` : Extraction d'une archive
 - `-it` : Vérification d'une archive
 - `-o` : Création d'une archive
 - `-v` : Mode verbeux

Archivage et sauvegarde

cpio

- Utilisation de cpio
 - Afficher les fichiers à sauvegarder dans la sortie puis sauvegarder
 - `find dossier/`
 - `find dossier/ | cpio -ocv > archive.cpio`
 - Vérifier l'archive
 - `cpio -icvt < archive.cpio`
 - Décompresser l'archive :
 - `cpio -icdv < archive.cpio`
- Création d'une archive cpio compressée avec bzip2
 - `find dossier/ | cpio -ocv | bzip2 -c > archive.cpio.bz2`
- Décompression :
 - `bzip2 -dc monArchive.cpio.bz2 | cpio -icvd`

Archivage et sauvegarde

rsync

- rsync (remote synchronization ou synchronisation à distance), est un logiciel de synchronisation de fichiers.
- Fréquemment utilisé pour mettre en place des systèmes de sauvegarde distante.
- rsync travaille de manière unidirectionnelle :
 - il synchronise, copie ou actualise les données d'une source (locale ou distante) vers une destination (locale ou distante) en ne transférant que les octets des fichiers qui ont été modifiés.
- Il utilise des fonctions de compression.
- Il utilise SSH par défaut pour les synchronisations distantes. Il est aussi utile pour des copies locales.

Archivage et sauvegarde

rsync

- apt-get install rsync
- sync source/ destination/
- rsync -az source/ login@serveur.org:/destination/
- rsync -e ssh -avz --delete-after /home/source
user@ip_du_serveur:/dossier/destination/
- rsync -e ssh -avzn --delete-after /home/mondossier_source
user@ip_du_serveur:/dossier/destination/

Archivage et sauvegarde

rsync

- Synchroniser deux répertoires au sein d'une même machine
 - `rsync -zvr /home/andn/dossiers/ /var/opt/dossiers/`
 - `-z` est pour activer la compression
 - `-v` pour verbeux
 - `-r` pour récursif
 - Attention ; Ne préserve pas l'horodatage
- l'option `-a` de `rsync` indique l'utilisation du mode archive.
 - Mode récursif
 - Préserver les liens symboliques
 - Préserver les permissions
 - Préserver l'horodate
 - Préserver les propriétaires et groupes
- `rsync -zva /home/andn/dossiers/ /var/opt/dossiers/`

Archivage et sauvegarde

rsync

- Synchroniser des fichiers vers un serveur distant
 - `rsync -avz /home/dossiers/files/ root@192.168.1.115:/opt/save`
- Si vous voulez utiliser la commande sans mot de passe, configurer l'échange de clef SSH.
- Synchroniser des fichiers depuis un serveur distant
 - `rsync -avz root@192.168.1.115:/opt/save /home/dossiers/files`
- Ne pas écraser les fichiers de destination modifiés (-u)
 - `rsync -avzu root@192.168.1.115:/opt/save/test.txt /home/dossiers/files/`
- Supprimer les fichiers créés sur la cible
 - `rsync -avz --delete root@192.168.1.115:/opt/save/ /home/dossiers/files/`
 - L'option `--delete` de `rsync` supprime les fichiers qui ne sont plus présent dans le répertoire source.

Archivage et sauvegarde

rsync

- Voir les modifications entre la source et la destination
 - L'option -i affiche les éléments modifiés.
 - `rsync -avzi root@192.168.1.115:/opt/save/test.txt /home/dossiers/files/`
- Ne pas transférer les gros fichiers
 - Vous pouvez demander à rsync de ne pas transférer des fichiers qui sont supérieures à une taille spécifique en utilisant l'option --max-size.
 - `rsync -avz --max-size='300K' root@192.168.1.115:/opt/save/test.txt /home/dossiers/files/`
- Inclure et exclure des fichiers à l'aide de pattern
 - rsync vous permet de fournir un pattern pour les fichiers ou répertoires que vous souhaitez inclure ou exclure durant la synchronisation.
 - `rsync -avz --include 'h*' --exclude '*' root@192.168.1.115:/opt/save/test.txt /home/dossiers/files/`
 - Seulement les fichiers commençant par « h » sont inclus et tous les autres sont exclus.
- Transférez les fichiers en entier
 - Au lieu d'envoyer des fichiers en entier, il n'envoie que les blocs modifiés. (+rapide mais contrôle source/cible passé)
 - Si limité en CPU, vous pouvez demander à transférer les fichiers complets, en utilisant `rsync -W`. (nécessite bonne bande passante)
 - `rsync -avzW root@192.168.1.115:/opt/save/test.txt /home/dossiers/files/`

Archivage et sauvegarde

dar

- Disk Archive (DAR) est utilisé pour faire des sauvegardes de données et des sauvegardes différentielles.
- Les principales fonctions sont :
 - archivage différentiel
 - paramétrage de la taille des archives
 - création d'un fichier catalogue contenant la liste des fichiers de l'archive (utile si on veut des archives différentielles sans garder l'archive de référence complète)
 - compression des données avec gzip ou bzip
 - extraction partielle de l'archive
 - DAR existe aussi pour windows
- `apt-get install dar`

Archivage et sauvegarde

dar

- Sauvegarde complète
- `dar -v -c "/home/jean/backups/backup" -R "/home" -w -s 734003200 -m 150 -P « jean/backups" -P jean/mp3 -P jean/photo -P jean/.Trash -X "*.iso" -Z "*.jpg"`
 - `-v` : actionne le mode verbose,
 - `-c "/home/jean/backups/full_backup"` : création de l'archive
 - le paramètre qui suit `-c` indique la localisation de l'archive et son nom générique (backup)
 - le (ou les) fichier(s) archive(s) portera le nom générique avec l'extension `.1.dar` (dans le cas ou plusieurs fichiers sont créés les extensions seront nom-de-la-trame.n.dar)
 - `-R "/home"` : indique l'arborescence à sauvegarder, ici on sauvegarde le répertoire `/home` et tous les sous répertoires.
 - `-w` : DAR écrase le fichier archive s'il existe déjà sans vous prévenir.
 - `-s 734003200` : taille des fichiers archive en octet (byte), dans ce cas la taille des fichiers archives est limité à 730 M.
 - `-m 150` : signifie que les fichiers de moins de 150 octets ne sont pas compressés.
 - `-P chemin` : défini les chemins des répertoires à ne pas prendre en compte dans l'archive (attention pas de chemin absolu ici)
 - `-X *.iso` : défini les type de fichiers à exclure, ici les fichiers avec l'extension `.iso`.
 - `-Z *.jpg` : défini les type de fichiers à ne pas compresser.

Archivage et sauvegarde

dar

- Exemple de commande :
- `dar -v -c "/root/dar_backups/full_backup" -R "/home" -w -m 10 -P "dar_backups" -X "*.iso"`
 - `-v` : actionne le mode verbose,
 - `-c "/root/dar_backups/full_backup"` : création de l'archive
 - `-R "/home"` : On sauvegarde home et ses sous répertoires
 - `-w` : DAR écrase le fichier archive s'il existe déjà sans vous prévenir.
 - `-m 150` : signifie que les fichiers de moins de 150 octets ne sont pas compressés.
 - `-P` : Ne prend pas en compte le répertoire dar_backups
 - `-X *.iso` : Exclue les iso

Archivage et sauvegarde

dar

- Sauvegarde différentielle
- `dar -v -c "/root/dar_backups/diff_backup_`date -l`" -R "/home" -A "dar_backups/full_backup" -w -s 734003200 -m 150 -X "*.iso"`
 - `-c "/root/dar_backups/diff_backup_`date -l`"` : définit le nom de l'archive différentielle.
 - La "commande" ``date -l`` permet d'ajouter la date dans le nom du fichier. Le nom des fichiers produits est du type suivant : `diff_backup_2021-04-19.1.dar`
 - `-A "/home/famille/dar_backups/full_backup"` : ce paramètre est important, il permet d'indiquer le nom de l'archive de référence.

Archivage et sauvegarde

dar

- Restauration d'une sauvegarde dar
- Nous supposons que vous êtes dans le dossier où est stocké « full_backup.1.dar »
- `dar -v -R /testdar -x "full_backup.1.dar"`
 - `-v` : actionne le mode verbose.
 - `-x` : indique que l'on veut extraire l'archive « full_backup.1.dar »
 - `-R /testdar` : indique l'arborescence où restaurer, ici on restaure dans le répertoire /testdar et tous les sous répertoires.
- D'autres options :
 - `-n` : permet de répondre automatiquement et négativement à une action utilisateur de réécriture.
 - `-w` : permet de répondre automatiquement et positivement à une action utilisateur de réécriture.
 - `-r` : ne restaure que les fichiers absents ou plus récents.
 - `-f` : ne restaure pas la structure des dossiers (intéressant pour la récupération de fichiers indépendants)

Archivage et sauvegarde

dar

- Examiner une archive :
 - `dar -v -t full_backup.1.dar`
- Restaurer un fichier uniquement
 - `dar -v -R /root -x "full_backup.1.dar" -g andn/passeport3.jpg`
 - Cette commande restaurera le passeport3.jpg dans root (dans le dossier andn)

Debug & Logs Debian

- Les logs systèmes se trouvent dans `/var/log/`
- En général, les logs d'autres applications se trouvent dans `/var/log/`, à moins que vous ne l'ayez configuré autrement.
- Pour voir "en direct" des logs (`tail -f`)
 - `# tail -f /var/log/auth.log`
- Avoir les 20 dernières lignes d'un fichier log
 - `# tail -n 20 /var/log/messages`
- Rechercher dans le texte facilement "à la vim" - `/recherche` avec `less`
 - `# tail -n 20 /var/log/messages | less`
- Rechercher un paquet en particulier
 - `# grep -R "nom_du_paquet" /var/log/*`

Debug & Logs Debian

- systemd possède son propre mécanisme de journalisation, syslog n'est plus requis par défaut.
- Voir les logs :
 - journalctl
- De manière "tail -f", continue sur la console jusqu'à ctrl+C :
 - journalctl -f
- Filtrer par service :
 - journalctl -u crond
- Filtrer par PID :
 - journalctl _PID=1
- Filtrer par programme :
 - journalctl /usr/sbin/sshd
- Filtrer par niveau de log (ici que les erreurs) :
 - journalctl -p err
- Vous pouvez combiner plusieurs options :
 - journalctl -f /usr/sbin/urpmi -p info

Debug & Logs Debian

- Filtrer les logs par date avec --since et --until.
 - Mettre la date au format YYYY-MM-DD HH:MM:SS
- depuis 21h le 16/06/2021
 - `journalctl --since "2021-06-16 21:00:00"`
- Entre le 16 et 17/06/21
 - `journalctl --since "2021-06-16 21:00:00" --until "2021-06-17 22:00:00"`
- Gestion de la taille des logs dans `/etc/systemd/journald.conf` :
 - `SystemMaxUse=500M` (Maximum 500Mo)
- Journaux non conservés sur le disque au redémarrage (machine non critique) :
 - `[Journal]`
`Storage=volatile`

les fichiers hosts.allow et hosts.deny

- Afin de déterminer si un ordinateur client est autorisé à se connecter à un service, les enveloppeurs TCP référencent les deux fichiers suivants, couramment appelés fichiers d'accès des hôtes :
- /etc/hosts.allow
- /etc/hosts.deny
- Lorsqu'une requête client est reçue par un service enveloppé avec TCP, ce dernier suit les étapes élémentaires ci-dessous :
- Le service référence /etc/hosts.allow — Le service enveloppé avec TCP analyse le fichier /etc/hosts.allow de manière séquentielle et applique la première règle spécifiée pour ce service. Si une règle correspond au service, il autorise la connexion. Sinon, il passe à l'étape suivante.
- Le service référence /etc/hosts.deny — Le service enveloppé avec TCP analyse le fichier /etc/hosts.deny de manière séquentielle. Si une règle correspond au service, il refuse la connexion. Sinon, il autorise l'accès au service.

les fichiers hosts.allow et hosts.deny

- Parce que les règles d'accès contenues dans le fichier hosts.allow sont appliquées en premier, elles ont priorité par rapport aux règles spécifiées dans le fichier hosts.deny. Par conséquent, si l'accès à un service est autorisé dans hosts.allow mais qu'une règle refusant l'accès à ce même service est contenue dans le fichier hosts.deny, cette dernière ne sera pas prise en compte.
- Étant donné que les règles dans chaque fichier sont lues de haut en bas et que la première règle appliquée à un service donné est la seule règle prise en compte, l'ordre de ces dernières est extrêmement important.
- Si aucune règle contenue dans l'un ou l'autre des fichiers ne s'appliquent au service ou si aucun de ces fichiers n'existe, l'accès au service est autorisé.
- Des services enveloppés avec TCP ne mettent pas en cache les règles des fichiers d'accès d'hôtes, ainsi, tout changement apporté à hosts.allow ou hosts.deny prend effet immédiatement sans devoir redémarrer les services réseau.

les fichiers hosts.allow et hosts.deny

- Comme dans toute politique de défense correcte, il est préférable de tout interdire puis d'autoriser service par service.
- hosts.deny
 - ALL: ALL
- hosts.allow
 - ALL: 192.168.0.3 On autorise tous les services pour le poste 1.3
 - ALL: 192.168.0.35 On autorise tous les services pour le poste 1.35
 - smtp: ALL accès depuis tous les postes
 - sshd: *.example.com Des astérisques peuvent être utilisés pour autoriser des groupes entiers de noms d'hôtes ou d'adresses IP, à condition qu'ils ne fassent pas aussi partie d'une liste de clients contenant d'autres types de filtres
 - http: 192.168. En plaçant un point à la fin d'une adresse IP, tous les hôtes partageant les premiers groupes numériques d'une adresse IP seront autorisés
 - sendmail : ALL
 - domain: ALL

xinetD

- Les fonctionnalités non exhaustives :
 - Contrôler l'origine des accès
 - Contrôler le moment des accès
 - Contrôler l'exposition du système pendant l'accès : chroot
 - Autoriser / interdire un service
 - Limiter les attaques de type Deny of Service
 - Redirection de ports
 - Attribution d'un service à une adresse IP
-
- <https://linux.die.net/man/5/xinetd.conf>
 - <http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-fr-4/s1-tcpwrappers-xinetd-config.html>

xinetD

- xinetd, qui signifie eXtended InterNET Daemon, est un démon open source qui tourne sur la plupart des systèmes Unix et qui gère les connexions basées sur l'internet. C'est une réécriture plus sécurisée de inetd.
- Fichier de configuration :
 - /etc/xinetd
 - /etc/xinetd.d/service1
- Exemple :

```
cat /etc/xinetd.d/telnet
service telnet
{
    ...
    server = /usr/sbin/in.telnetd
    ...
}
```

- attribut : server
- assignement : "="
- valeur : /usr/sbin/in.telnetd

xinetD

- Contrôler l'origine des accès
- Filtrer des adresses : `only_from = valeur [valeurs...]` ou `no_access = valeur [valeurs...]`
 - La connexion au service ne sera possible qu'à partir de la liste fournie à cet attribut. Elle peut contenir :
 - des adresses IP
 - des adresses réseau : 192.168.0.0
 - des hostnames : Ce sont des noms de machine. A utiliser si et seulement si le fichier `/etc/hosts` est correctement renseigné.
 - des noms de domaine

service vsftpd

```
{  
  ...  
  only_from = ubuntu01 192.168.1.0 syslab.lan  
  ...  
}
```

xinetD

- Contrôler le moment des accès
- Vous pouvez choisir le moment auquel vous autoriserez les accès à tout ou partie de vos services réseaux.
- On utilisera l'attribut `access_times`.
 - Il vous permet de définir une ou plusieurs plages horaires pendant lesquelles la connexion sera possible.
- Syntaxe : `access_times = interval [interval...]`
- L'intervalle de temps s'écrit : heures:minutes-heures:minutes
- Limiter l'accès de mon ftp de 9h à 12h et de 14h à 16h.
- `service vsftpd`
- `{`
- `...`
- `access_time = 9:00-12:00 14:00-16:00`
- `...`
- `{`

xinetD

- Contrôler l'exposition du système pendant l'accès : chroot
- xinetd vous permet de "chrooter" un service.
- Rappel : la commande chroot permet de lancer un programme en restreignant ses accès disques à une sous arborescence. En fait pour le processus, la racine du disque est la racine de l'arborescence dans laquelle il a été restreint.
- L'attribut `server_args` va nous permettre d'automatiser le chroot : la commande chroot est considérée comme le serveur et le service est passé en argument.
- Chrooter mon serveur ftp.
- `service vsftpd`
- `{`
- `...`
- `server = /usr/sbin/chroot`
- `server_args = /opt/vsftpd/vsftpd`
- `...`
- `}`
- Lorsqu'un client tente un accès ftp, chroot est exécuté en tant que serveur et vsftpd en tant qu'argument.

xinetD

- Autoriser / interdire un service
 - Les services installés et gérés par xinetd sont désactivés d'office, pour des raisons de sécurité.
 - Nous pouvons choisir de désactiver complètement un service.
 - Syntaxe : disable = yes|no
-
- Désactiver openssh server.
 - service sshd
 - {
 - disable = yes
 - ...
 - }

xinetD

- Limiter les attaques de type Deny of Service
- **Contrôle de la charge CPU** : `rlimit_cpu = seconds`. Cet attribut vous permet de limiter le temps CPU utilisé par un ou plusieurs services.
- **Priorité accordée au processus serveur** : `nice = level`. Fixer une priorité d'ordonnancement pour le serveur. Le level peut prendre les valeurs de -20 (le plus prioritaire) à 19 (le moins prioritaire).
- **Limite du nombre de connexions par service** : `instances = value`. L'attribut détermine le nombre d'instances simultanées du serveur qui seront autorisées. Préciser un nombre. Sans précision, le nombre d'instances pourra être illimité.
- **Limite du nombre de connexions ayant la même origine** : `per_source = value`.
 - Value = un nombre ou UNLIMITED
 - Filtrer les IP clientes, le nombre d'instances, limiter les connexion à un serveur donné (venant d'une même IP)
- **Blacklister des adresses IP** : Il vous est possible blacklister des adresses IP qui tenteraient des connexions sur des services que vous avez désactivés mais qui constituent la cible préférée des hackers (exemple : telnet). On utilisera 2 attributs de manière combinée :
 - `flags = SENSOR`
 - `deny_time = minutes`
 - `SENSOR` empêche toute connexion au service concerné et stocke l'adresse IP pendant un temps déterminé par l'attribut `deny_time`.
 - Si cette même adresse tente de se connecter, à n'importe quel service géré par xinetd, il sera automatiquement bloqué. Le temps est déterminé en minutes, mais vous pouvez utiliser également la valeur **FOREVER** : l'IP restera blacklistée jusqu'au prochain redémarrage de xinetd.

xinetD

- Attribution d'un service à une adresse IP
- xinetd va vous permettre de lier un service à une adresse IP grâce à l'attribut bind.
- Construire 2 serveurs ftp bien différenciés. L'un sera réservé à aux machines en local (serveur de fichiers interne) l'autre mettra à disposition d'autres fichiers sur un serveur ftp réservé aux connexions externes.
- 2 interfaces réseau avec les adresses IP respectives : 192.168.1.115 et 1.1.1.1.

```
root@routeur# cat /etc/xinetd.d/vsftpd
```

```
service vsftpd
```

```
{
```

```
  id = ftp_public
```

```
  ...
```

```
  server = /opt/vsftpd/vsftpd
```

```
  bind = 1.1.1.1
```

```
}
```

```
service vsftpd
```

```
{
```

```
  id = ftp_privé
```

```
  ...
```

```
  server = /opt/vsftpd/vsftpd
```

```
  bind = 192.168.1.115
```

```
}
```

L'attribut id sert uniquement à différencier les 2 configurations du service.

iptables

- Une table est constituée de chaînes
 - Une chaîne est constituée d'une suite de règles
 - Une règle est constituée de motifs (patterns)
- destiné à reconnaître des paquets selon des
- critères (matches)
 - En cas de reconnaissance du paquet, une
- décision, appelée cible (target), est prise.

iptables

- iptables est l'un des firewall de Linux.
- Il existe également « ufw » (Uncomplicated Firewall) intégré à Ubuntu.
 - En graphique il y a également « Gufw »
- Iptable contient trois tables, FILTER, NAT et MANGLE
 - iptables -L
 - Afficher les règles actuelles sur la machine
 - Chain INPUT : correspond aux règles manipulant le trafic entrant ;
 - Chain FORWARD : correspond aux règles manipulant la redirection du trafic ;
 - Chain OUTPUT : correspond aux règles manipulant le trafic sortant.
 - (policy ACCEPT)
 - Signifie que tout le trafic est accepté
 - iptables -F chain
 - Réinitialiser toutes les règles de firewall
 - Si on rajoute la « chain », supprime la chain correspondante
 - Les règles sont lus dans l'ordre (de haut en bas)
 - iptables -L --line-numbers
 - Vérifier l'ordre des règles

iptables

- NOTE IMPORTANTE :

Les règles iptables auront disparues au redémarrage de la machine. Il est donc nécessaire de créer un script qui s'exécute au démarrage et qui applique les règles.

- Iptables est un firewall logiciel, capable de tout ce que peut faire un firewall réseau/transport.
- Iptables ne peut pas faire proxy.
- Iptables à trois chain :
 - pour les paquets qui entrent dans le serveur INPUT,
 - pour les paquets qui sortent du serveur OUTPUT,
 - pour les paquets qui transitent par le serveur FORWARD.
- Il faut commencer par définir la politique par défaut à DROP, pour chaque chain :
 - iptables --policy chain DROP

iptables

- Exemple de règle :

Chain INPUT (policy DROP)

num	target	prot	opt	source	destination	
1	ACCEPT	tcp	--	anywhere	anywhere	tcp dpt:ssh
2	ACCEPT	tcp	--	anywhere	anywhere	tcp dpt:www

- Chaque ligne correspond à une règle différente qui permet de filtrer ou non une IP ou un port.
 - **target** : ce que fait la règle. Ici c'est ACCEPT, c'est-à-dire que cette ligne autorise un port et / ou une IP
 - **prot** : le protocole utilisé (tcp,udp,icmp).
 - **source** : l'IP de source. Pour INPUT, la source est l'ordinateur distant qui se connecte à vous ;
 - **destination** : l'IP de destination. Pour OUTPUT, c'est l'ordinateur auquel on se connecte
 - **la dernière colonne** : elle indique le port après les deux points « : ». Ce port est affiché en toutes lettres, mais avec « -n » vous pouvez obtenir le numéro correspondant.
- Ici, ssh et www sont autorisé en entrée. Personne ne peut se connecter à la machine par un autre moyen (port)
- (policy DROP) : ignore tous les autres paquets

iptables

```
iptables --append INPUT --protocol tcp --dport 80 -m conntrack --ctstate  
NEW,ESTABLISHED -j ACCEPT
```

- --append ou -A ajoute une règle,
- --protocol ou -p permet de spécifier le protocole,
- --dport permet de spécifier le port de destination,
- --sport le port source,
- --match ou -m conntrack permet d'affiner la recherche sur le paquet,
- --ctstate NEW, ESTABLISHED permet de spécifier si l'état de connexion est nouveau ou établi,
- --jump ou -j ACCEPT permet de dire ce qu'il faut faire de ce paquet. (ACCEPT, REJECT ou DROP).

iptables

- Quelques options :
- **-D chain rulenum** : supprime la règle n° **rulenum** pour la **chain** indiquée.
- **-I chain rulenum** : insère une règle au milieu de la liste à la position indiquée par **rulenum**. Si vous n'indiquez pas de position **rulenum**, la règle sera insérée en premier, tout en haut dans la liste.
- **-R chain rulenum** : remplace la règle n° **rulenum** dans la **chain** indiquée.
- **-P chain regle** : modifie la règle par défaut pour la **chain**. Cela permet de dire, par exemple, que par défaut tous les ports sont fermés, sauf ceux que l'on a indiqués dans les règles.
- **-j (décision)** : ACCEPT OU REJECT ou DROP (ignorer)

iptables

- Ajouter ou supprimer une règle

```
iptables -A (chain) -p (protocole) --dport (port) -j (décision)
```

```
iptables -A INPUT -p tcp --dport ssh -j ACCEPT
```

- Ici on ajoute dans le trafic entrant une règle qui autorise le SSH (22)
 - Lorsque que la machine reçoit une trame sur le port 22, elles seront acceptées.

```
iptables -A INPUT -p tcp --dport www -j ACCEPT
```

```
iptables -A INPUT -p icmp -j ACCEPT
```

- Pareil mais pour le web (80) et le ping et l'ICMP (ping)
- /!\ Si on ne précise pas de port, tous les ports seront acceptés, donc ici, la règle ICMP permet d'autoriser tous les ports mais uniquement pour le ping.

iptables

```
iptables -A INPUT -i lo -j ACCEPT
```

- Ici, le “-i lo” autorise le trafic interne (sur l’interface loopback)

```
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

- Cette règle autorise toutes les connexions qui sont déjà à l’état ESTABLISHED ou RELATED (netstat).
- Elle autorise toutes les connexions qui ont été demandées par votre PC. Cela permet d'assouplir le pare-feu et de le rendre fonctionnel pour une utilisation quotidienne.

```
iptables -P INPUT DROP
```

- Toutes les autres données (qui ne sont pas autorisées), vont être refusées. (policy DROP)

ufw (Uncomplicated Firewall)

- Uncomplicated Firewall est pré-installé sous Ubuntu.
 - Sinon paquet « ufw »
- Vérifier le statut actuel :
 - `ufw status`
- État : actif ou inactif
- Activer UFW : appliquer les règles
 - `ufw enable`
- Désactiver UFW : ne plus appliquer les règles définies
 - `ufw disable`
- Vérifier le statut actuel en détail :
 - `ufw status verbose`
- Afficher les règles numérotés
 - `ufw status numbered`

ufw (Uncomplicated Firewall)

- Lorsque UFW est activé, par défaut le trafic entrant est refusé et le trafic sortant est autorisé.
- Autoriser le trafic entrant suivant les règles par défaut :
 - `ufw default allow`
- Refuser le trafic entrant suivant les règles par défaut :
 - `ufw default deny`
- Autoriser le trafic sortant suivant les règles par défaut :
 - `ufw default allow outgoing`
- Refuser le trafic sortant suivant les règles par défaut :
 - `ufw default deny outgoing`

ufw (Uncomplicated Firewall)

- Activer la journalisation :
 - ufw logging on
- Désactiver la journalisation :
 - ufw logging off
- Autoriser une connexion entrante :
 - ufw allow [règle]
- Refuser une connexion entrante :
 - ufw deny [règle]
- Refuser une IP entrante :
- Si vous voulez bloquer une IP sur tous vos services, il faut le faire avant les autorisations existantes avec "insert 1" qui met ce "deny" avant tous les "allow".
 - ufw insert 1 deny from [ip]

ufw (Uncomplicated Firewall)

- Refuser une connexion entrante, uniquement en TCP :
 - `ufw deny [port]/tcp`
- Refuser une connexion sortante :
 - `ufw deny out [règle]`
- Supprimer une règle :
 - `ufw delete allow [règle]`
 - `ufw delete deny [règle]`
- Supprimer simplement une règle d'après son numéro :
 - `ufw delete [numéro]`
- `[port]` est à remplacer par le numéro du port désiré.
- `[règle]` est à remplacer par le numéro du port ou le nom du service désiré.
- `[numéro]` est à remplacer par le numéro de la règle désiré.

ufw (Uncomplicated Firewall)

- Ouverture du port 53 en TCP et UDP :
 - `ufw allow 53`
- Ouverture du port 25 en TCP uniquement :
 - `ufw allow 25/tcp`
- Utilisation des services
- UFW regarde dans sa liste de services connus pour appliquer les règles standards. Ces règles sont automatiquement converties en ports.
- Lister des services :
 - `cat /etc/services`
- Autoriser le service SMTP :
 - `ufw allow smtp`
- Autoriser le port de Gnome-Dictionary (2628/tcp) :
 - `ufw allow out 2628/tcp`
- Autoriser le protocole pop3 sécurisé (réception de courriers)
 - `ufw allow out pop3s`

ufw (Uncomplicated Firewall)

- Refuser le protocole (proto) TCP à (to) tout le monde (any) sur le port (port) 80 :
 - `ufw deny proto tcp to any port 80`
- Refuser à (to) l'adresse 192.168.1.1 de recevoir sur le port (port) 25 les données provenant (from) du réseau de classe A et utilisant le protocole (proto) TCP :
 - `ufw deny proto tcp from 10.0.0.0/8 to 192.168.1.1 port 25`
- Refuser les données utilisant le protocole (proto) UDP provenant (from) de 1.2.3.4 sur le port (port) 514 :
 - `ufw deny proto udp from 1.2.3.4 to any port 514`

ufw (Uncomplicated Firewall)

- Insérer une règle à une position précise en utilisant le numéro
 - `ufw insert NUM RULE`
- Insérer en numéro 2 une règle refusant le trafic entrant utilisant le protocole (proto) UDP (to) en direction de (any) toute les adresses en écoute sur votre machine sur le port (port) 514 en provenance (from) de 1.2.3.4
 - `ufw insert 2 deny proto udp to any port 514 from 1.2.3.4`
- Réinitialiser UFW
 - `ufw reset`
- Forcer le reset sans demander d'autorisation :
 - `ufw reset --force`

Scripting BASH

- La programmation en Shell
 - Mini langage de programmation intégré à Linux
 - Très utile pour l'automatisation de tâches
- A noter qu'un script en sh sera fonctionnel sur toutes les distributions, car le bash n'est pas forcément installé sur les distributions Unix (Solaris, MacOS...)
- Créer un script bash ou sh :
 - vim script.sh
 - L'extension « .sh » est ici une convention de nommage pour les scripts et n'est pas obligatoire
 - Ajouter les droits d'exécutions à votre script.
- A la toute première ligne d'un script, vous devez indiquer quel Shell sera utilisé pour son exécution. Donc dans « script.sh » ajoutez :

```
#!/bin/bash
```

Le #! est appelé le sha-bang.

- /bin/bash peut être remplacé par /bin/sh si vous souhaitez coder pour sh, /bin/ksh pour ksh, etc.

Scripting BASH

- Principe d'un script :
- Après le sha-bang, vous pouvez commencer à ajouter vos différentes commandes.
- Un script utilisera les même commandes en BASH que vous utilisez dans votre console.

```
#!/bin/bash  
ls -lisa
```

- Voici un script qui affiche le contenu d'un dossier.
- Vous pouvez ajouter des commentaires à votre script.

```
#!/bin/bash  
ls -lisa  
#Ceci est un commentaire
```

- Les commentaires ne seront pas pris en compte dans votre script, mais il est conseillé de les utiliser pour détailler votre script.

Scripting BASH

- Exécution de votre script :
- Après avoir ajouter les droits (chmod +x) , placez vous dans le dossier contenant votre script, puis lancez le en tapant « ./ » devant.

```
~#./script.sh
```

- Vous pouvez lancer le mode debug d'un script avec la commande :

```
~#bash -x script.sh
```

- Vous pourrez ainsi voir le comportement du script lors de son exécution.

Scripting BASH

- Créer sa propre commande :
- Pour créer votre commande, vous devrez ajouter votre script dans l'un des dossier d'accès au binaire de votre machine. Ceux-ci sont référencer dans la variable d'environnement \$PATH
 - echo \$PATH
 - env | grep PATH
- Donc par exemple, si je place « script.sh » dans « /usr/local/bin », je peux ensuite taper directement dans ma console (depuis n'importe ou) « script.sh », et mon script s'exécutera.

Scripting BASH

Les variables

- Les variables permettent de stocker temporairement des informations en mémoire.
- Une variable est un **NOM** associé à une zone mémoire pour conserver une chaîne de caractères ou une valeur, permettant la mémorisation d'informations et l'échange d'informations entre processus.
- Une variable est identifiée par un **NOM**.
- Ce nom peut comporter des chiffres, des lettres ainsi que le caractère « _ » .
- **Le nom de la variable ne doit jamais commencer par un chiffre.**
- Une variable n'est accessible que par le Shell qui l'a créée, c'est à dire votre Shell de connexion par exemple.
- Rappel sur les types de quotes :
 - les apostrophes ' ' (simples quotes) ;
 - les guillemets " " (doubles quotes) ;
 - les accents graves ` ` (back quotes), qui s'insèrent avec Alt Gr + 7.

Scripting BASH

Les variables

- Définir une variable :
 - Pour définir une variable, on écrit son nom, puis sa valeur après le « = ». Attention il n'y a pas d'espace.

```
#!/bin/bash
variable1=' bonjour '
variable2=' bonjour c\ 'est moi '
variable3=`pwd`

echo $variable1
echo " La variable 2 est $variable2 "
echo -e " Vous êtes dans le dossier\n$variable3 «
echo " J'affiche $variable1 et $variable2 "
```

- variable1 contient une chaîne de caractère simple, entre simple quote.
 - echo \$variable 1 affiche le contenu de la variable.
- variable2 contient une chaîne de caractère contenant une apostrophe, il faut donc y ajouter « \ » avant l'apostrophe pour celle-ci soit reconnue en tant que caractère.
 - Dans echo, pour afficher des caractères, en plus d'une variable, tout est entre double quote.
- variable3 contient une commande, elle est définie entre 2 « accents grave » (alt gr + 7).
 - le « -e » du echo permet d'activer le retour à la ligne grâce au symbole « \n »

Scripting BASH

Les variables

- Demander une saisie de l'utilisateur avec « read ».

```
read nom  
echo "bonjour $nom"
```

- Option « -p » pour afficher du texte avant la saisie :

```
read -p 'Entrez votre nom : ' nom  
echo "bonjour $nom"
```

- Limiter le nombre de caractère avec « -n »

```
read -p 'Entrez votre nom (5 caractère max) : ' -n 5 nom  
echo "bonjour $nom"
```

- « -s » pour saisir une chaîne sans qu'elle s'affiche en console.

```
read -p 'Entrez votre mot de passe: ' -s pass  
echo -e « \nMerci, j'affiche maintenant votre mot de passe : $pass"
```

Scripting BASH

Les variables d'environnement (env)

- Quand vous créez une variable dans un script, celle-ci n'existe QUE dans le script.
- Les variables d'environnement ou variables globales sont par contre utilisables dans n'importe quel programme.
- Quelques exemples :
 - SHELL : indique quel type de shell est en cours d'utilisation (sh, bash, ksh...) ;
 - PATH : une liste des répertoires qui contiennent des exécutables que vous souhaitez pouvoir lancer sans indiquer leur répertoire.
 - EDITOR : l'éditeur de texte par défaut qui s'ouvre lorsque cela est nécessaire ;
 - PWD : le dossier dans lequel vous vous trouvez ;
- Vous pouvez les appeler simplement par leur nom dans votre script.

```
#!/bin/bash  
echo « Mon shell est le $SHELL »
```

Scripting BASH

Les variables des paramètres

- Comme les commandes en bash, vous pouvez ajouter des paramètres à votre votre script.
 - `./script.sh param1 param2`
- Les variables des paramètres seront automatiquement créées :
 - « `$#` » : Contient le nombre de paramètres
 - « `$0` » : Contient le nom du script exécuté (« `./script.sh` »)
 - « `$1` » : Premier paramètre
 - « `$2` » : 2nd paramètre
 - [...]
 - « `$9` » : 9^{ème} paramètre

```
#!/bin/bash
```

```
echo « Execution de $0, il y a $# paramètres »  
echo "Le paramètre 1 est $1"
```

```
~ ./script.sh param1 param2  
Vous avez lancé ./script.sh, il y a 2 paramètres  
Le paramètre 1 est param1
```

Scripting BASH

Les variables des paramètres

- Si il y a plus de 9 paramètres, vous pouvez les décaler dans votre script en utilisant « shift ».

```
#!/bin/bash  
  
echo "Le paramètre 1 est $1"  
shift  
echo "Le paramètre 1 est maintenant $1 "
```

```
~ ./script.sh param1 param2  
Le paramètre 1 est param1  
Le paramètre 1 est maintenant param2
```

- shift est généralement utilisé dans des boucles.

Scripting BASH

Les variables tableaux

- Le bash gère également les variables « tableaux ».
- Ce sont des variables qui contiennent plusieurs cases, comme un tableau.
- Définir un tableau :

```
tableau=('valeur0' 'valeur1' 'valeur2')
```

Ma variable « tableau » contient ici trois valeurs.

Si je veux afficher une case du tableau dans un echo :

```
echo ${tableau[1]}
```

Les cases sont numérotés à partir de 0.

Scripting BASH

Les variables tableaux

- Définir manuellement une case d'un tableau
 - `tableau[10]='valeur10'`
- Ici j'ai défini la 10^{ème} case du tableau. Avoir 9 autres cases n'est pas indispensable, nous pouvons définir autant de cases que voulu, avec la numération voulu et sauter des cases est autorisé.

```
#!/bin/bash
```

```
tableau=('valeur0' 'valeur1' 'valeur2')  
tableau[10]='valeur10'  
echo ${tableau[1]}
```

- Pour afficher toutes les valeurs du tableau
 - `${tableau[*]}`

```
tableau=('valeur0' 'valeur1' 'valeur2')  
tableau[10]='valeur10'  
echo ${tableau[*]}
```

Scripting BASH

Opérations mathématiques

- #Opérations basique possible grâce à "let«

```
#!/bin/bash  
let "a = 5"  
let "b = 2"  
let "c = a + b"  
echo $c
```

```
#!/bin/bash  
let "a = 5 * 3" # $a = 15  
let "a = 4 ** 2" # $a = 16 (4 au carré)  
let "a = 8 / 2" # $a = 4  
let "a = 10 / 3" # $a = 3  
let "a = 10 % 3" # $a = 1
```

- l'addition : + ;
- la soustraction : - ;
- la multiplication : * ;
- la division : / ;
- la puissance : ** ;
- le modulo (renvoie le reste de la division entière) : %.

Scripting BASH

Les conditions - if

Les branchements conditionnels (conditions) constituent un moyen de dire dans notre script

« SI (if) cette variable vaut tant, ALORS (then) fais ceci, SINON (else) fais cela ».

Une boucle se termine par FIN SI (fi).

```
SI test_de_variable
ALORS
-----> effectuer_une_action
FIN SI
```

```
nom= "Vincent"

if [ $nom = "Vincent" ]
then
    echo "Salut Vincent !"
else
    echo "Who are you ?!"
fi
```

N: B: Le test de la boucle (if) est entre crochet et doit TOUJOURS avoir des espaces à l'intérieur.
[test] OK
[test] PAS OK

Scripting BASH

Les conditions - if

Ici le test n'est pas valide donc c'est le else qui sera affiché en console.

```
nom="Vincent"  
nom2="Leon"  
  
if [ $nom = $nom2 ]  
then  
    echo "Salut Vincent !"  
else  
    echo "Who are you ?!"  
fi
```

Scripting BASH

Les conditions - else

Quand la condition n'est pas remplie, vous pouvez rajouter un else qui signifie « sinon ».

```
SI test_de_variable  
ALORS  
-----> effectuer_une_action  
SINON  
-----> fait ça  
FIN SI
```

```
nom= "Vincent"  
  
if [ $nom = "Vincent" ]  
then  
    echo "Salut Vincent !"  
else  
    echo "Who are you ?!"  
fi
```

Scripting BASH

Les conditions - else

Ici, si le paramètre (\$1) n'est pas égal à Leon, nous afficherons le else :

```
nom="Vincent"
nom2="Leon"

if [ $1 = « Leon » ]
then
    echo "Salut Léon !"
else
    echo "Who are you ?!"
fi
```

Scripting BASH

Les conditions - elif

Quand la condition n'est pas remplie, vous pouvez rajouter un elif qui signifie « sinon si » . Vous pouvez en ajouter autant que vous voulez.

```
SI test_de_variable  
ALORS  
-----> effectuer_une_action  
SINON  
-----> fait ça  
SINON SI  
-----> fait ça  
FIN SI
```

```
if [ $1 = "Vincent" ]  
then  
    echo "Salut Vincent !"  
  
elif [ $1 = "Leon" ]  
then  
    echo "Salut Leon !"  
  
elif [ $1 = "Jean" ]  
then  
    echo "Salut Jean !"  
  
else  
    echo "Who are you ?!"  
fi
```

Scripting BASH

Les tests

On appelle un test les éléments entre les crochet au « if » ou « elif ».

```
if [ test ]
then
    echo "Salut Vincent !"

elif [ test ]
then
    echo "Salut Leon !"

elif [ $1 = "Jean" ] #ceci est test
then
    echo "Salut Jean !"

else
    echo "Who are you ?!"
fi
```

Les différents types de tests :

- des tests sur des chaînes de caractères ;
- des tests sur des nombres ;
- des tests sur des fichiers.


Scripting BASH

Les tests

- Les opérateurs permettent de faire plusieurs tests à la fois.
- Ils s'incluent dans le if ou elif.
- ET = &&
- OU = ||
- ! -[option] = Inverser le test
 - if [! -e fichier] = Vérifier que le fichier n'existe pas.

```
#!/bin/bash
```

```
if [ $# -ge 1 ] && [ $1 = 'vincent' ]  
then  
    echo « Test OK »  
else  
    echo « Nope »  
fi
```



Vérifie qu'il y a au moins un paramètre (\$#)
Vérifie que ce paramètre est égal à la chaîne
'vincent'

Scripting BASH

Les tests

Tests sur des chaînes de caractères.

Les variables en bash sont toutes considérés comme des chaînes de caractères.
(Bash est sensible à la casse).

```
if [ $1 != $2 ]
then
    echo " Les 2 paramètres sont différents !"
else
    echo "Les 2 paramètres sont identiques"
fi
```

```
#!/bin/bash

if [ -z $1 ]
then
    echo "Pas de paramètre"
else
    echo "Paramètre présent"
fi
```

Condition	Signification
\$chaine1 = \$chaine2	Vérifie si les deux chaînes sont identiques.
\$chaine1 != \$chaine2	Vérifie si les deux chaînes sont différentes.
-z \$chaine	Vérifie si la chaîne est vide.
-n \$chaine	Vérifie si la chaîne est non vide.

Scripting BASH

Les tests

Test sur les nombres.

Condition	Signification
<code>\$num1 -eq \$num2</code>	Vérifie si les nombres sont égaux (e qual). À ne pas confondre avec le « = » qui, lui, compare deux chaînes de caractères.
<code>\$num1 -ne \$num2</code>	Vérifie si les nombres sont différents (n onequal). Ne pas confondre pas avec « != » qui est censé être utilisé sur des chaînes de caractères.
<code>\$num1 -lt \$num2</code>	Vérifie si num1 est inférieur (<) à num2 (l ower t han).
<code>\$num1 -le \$num2</code>	Vérifie si num1 est inférieur ou égal (<=) à num2 (l ower o re q ual).
<code>\$num1 -gt \$num2</code>	Vérifie si num1 est supérieur (>) à num2 (g reater t han).
<code>\$num1 -ge \$num2</code>	Vérifie si num1 est supérieur ou égal (>=) à num2 (g reater o re q ual).

Scripting BASH

Les tests

Test pour vérifier un nombre supérieur ou égal.

```
#!/bin/bash

if [ $1 -ge 26 ]
then
    echo "Vous avez plus de 26 ans (vous avez $1 ans)"
else
    echo "Vous avez $1 ans, un peu jeune non ?"
fi
```

Scripting BASH

Les tests

Tests sur les fichiers

Condition	Signification
-e \$nomfichier	Vérifie si le fichier existe.
-d \$nomfichier	Vérifie si le fichier est un répertoire. (Un dossier est aussi considéré comme un fichier)
-f \$nomfichier	Vérifie si le fichier est un fichier. Un vrai fichier cette fois, pas un dossier.
-L \$nomfichier	Vérifie si le fichier est un lien symbolique (raccourci).
-r \$nomfichier	Vérifie si le fichier est lisible (r).
-w \$nomfichier	Vérifie si le fichier est modifiable (w).
-x \$nomfichier	Vérifie si le fichier est exécutable (x).
\$fichier1 -nt \$fichier2	Vérifie si fichier1 est plus récent que fichier2 (<i>newerthan</i>).
\$fichier1 -ot \$fichier2	Vérifie si fichier1 est plus vieux que fichier2 (<i>olderthan</i>).

Scripting BASH

Les tests

```
#!/bin/bash

read -p 'Entrez un répertoire : ' repertoire
read -p 'Entrer un fichier : ' fichier

if [ -e $repertoire ] && [ -e $fichier ]
then
    echo "l'élément $repertoire existe et l'élément $fichier aussi"
    if [ -d $repertoire ] && [ -f $fichier ]
    then
        echo "$repertoire est un dossier ! et $fichier est un fichier"
    elif [ -d $repertoire ] && [ ! -f $fichier ]
    then
        echo "$repertoire est un dossier mais $fichier n'est pas un fichier"
    elif [ ! -d $repertoire ] && [ -f $fichier ]
    then
        echo "$repertoire n'est pas un dossier mais $fichier est un fichier"
    else
        echo "Ce ne sont ni des dossier ni des fichiers"
    fi
elif [ -e $repertoire ] && [ ! -e $fichier ]
then
    echo "l'élément $repertoire existe mais l'élément $fichier n'existe pas"
    if [ -d $repertoire ]
    then
        echo "$repertoire est bien un dossier"
    else
        echo "$repertoire n'est pas un dossier"
    fi
elif [ ! -e $repertoire ] && [ -e $fichier ]
then
    echo "l'élément $fichier existe mais l'élément $repertoire n'existe pas"
    if [ -f $fichier ]
    then
        echo "$fichier est bien un fichier"
    else
        echo "$fichier n'est pas un fichier"
    fi
else
    echo "Le dossier $repertoire et le fichier $fichier n'existe pas!"
fi
```

Ce script permet de tester si le répertoire et le fichier donné en argument (read) existent sur la machine, et vérifie également si ce sont bien des dossiers et/ou des fichiers

Vous pouvez sans soucis associer tous les opérateurs, mettre des conditions dans des conditions etc...

Scripting BASH

Les conditions - case

Le rôle de case est de tester la même variable mais plus simplement.

Utilisez case pour les tests d'une même variable et elif quand vous avez d'autres variables à tester. A noter que cela revient au même, c'est simplement plus lisible avec case.

La variable évaluée et les valeurs proposées peuvent être des chaînes de caractères ou des résultats de sous-exécutions de commandes.

Lorsque la valeur est sujette à variation, il est conseillé d'utiliser les caractères jokers [] pour spécifier les possibilités :

```
[Oo][Uu][l])  
  echo "oui"  
;;
```

Le caractère | permet aussi de spécifier une valeur ou une autre :

```
"oui" | "OUI")  
  echo "oui"  
;;
```

Scripting BASH

Les conditions - case

```
#!/bin/bash

case $1 in
    "Vincent")                #Nous testons la variable $1
                              #Test de la valeur Vincent
        echo "Salut Vincent !"
        ;;                   # Si la valeur Vincent est vérifié, tout ce qui suit est exécuté jusqu'au double point virgule
    J*)                       # Test de la valeur J*, tout ce qui commencera par J sera pris en compte

        echo "Salut toi qui commence par J"
        ;;
    "Paul" | "Jacques")       # Test de la valeur Paul
        echo "Salut Paul OU Jacques"
        ;;
    *)                       #Correspond au "else" du case
        echo "Salut personne !"
        ;;
esac                          # Marque la fin du case
```

Scripting BASH

Les boucles

- Les boucles permettent de répéter autant de fois que nécessaire une partie du code.
- Il faut être vigilant sur la syntaxe. Une espace de trop ou de moins, l'oubli d'un caractère spécial et plus rien ne fonctionne.
- Il existe différents types de boucle :
- La boucle « While » TANT QUE
- La boucle « for » qui permet de boucler sur une liste de valeurs

Rappel des tests de chaîne de caractère.	Condition	Signification
	\$chaîne1 = \$chaîne2	Vérifie si les deux chaînes sont identiques.
	\$chaîne1 != \$chaîne2	Vérifie si les deux chaînes sont différentes.
	-z \$chaîne	Vérifie si la chaîne est vide.
	-n \$chaîne	Vérifie si la chaîne est non vide.

Scripting BASH

Les boucles - while

- Structure de la boucle while

```
TANT QUE test  
FAIRE  
    [Actions]  
RECOMMENCER
```

- En bash cela donne :

```
while [ test ]  
do  
    echo 'Que pasa ?'  
done
```


Scripting BASH

Les boucles - while

- Ici nous pouvons tester une chaîne de caractère (à noter qu'il est possible de faire les mêmes tests qu'avec les conditions)
- Cette boucle testera si l'utilisateur rentre bien « oui » dans la console. Associé à une condition qui envoie un message en fonction de la réponse.

```
#!/bin/bash

while [ -z $reponse ] || [ $reponse != 'oui' ]
do
    read -p 'Dites oui : ' reponse
    if [ $reponse = oui ]
    then
        echo "Cool tu as dis oui !"
    else
        echo "Je t'ai dis de dire oui!"
    fi
done
```

Nous testons :

Est-ce \$réponse est vide ? (-z) OU
\$réponse est différent de « oui » ?

L'utilisateur entre une chaîne avec
« read », qui est envoyée ensuite
dans la variable « reponse »

Si la réponse est « oui », ALORS dit
« Cool... » SINON dire « Je t'ai dis... »

FIN DE LA CONDITION
FIN DE LA BOUCLE

Scripting BASH

Les boucles - while

- While peut boucler en fonction du nombre de ligne dans un fichier :

```
#!/bin/bash
```

```
num=0
```

```
while read ligne  
do
```

```
    num=$((num+1))
```

```
    echo "$num $ligne"
```

```
done < /etc/passwd
```



```
root@sweb01:~/scripts# ./boucle2.sh
1 root:x:0:0:root:/root:/bin/bash
2 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
3 bin:x:2:2:bin:/bin:/usr/sbin/nologin
4 sys:x:3:3:sys:/dev:/usr/sbin/nologin
5 sync:x:4:65534:sync:/bin:/bin/sync
6 games:x:5:60:games:/usr/games:/usr/sbin/nologin
7 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
8 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
9 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
10 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
11 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
12 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
13 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
14 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
15 list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
16 irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
17 gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
18 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
19 systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
20 systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
21 systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
22 messagebus:x:103:106:/:/nonexistent:/usr/sbin/nologin
23 syslog:x:104:110:/:/home/syslog:/usr/sbin/nologin
24 _apt:x:105:65534:/:/nonexistent:/usr/sbin/nologin
25 tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
26 uidd:x:107:112:/:/run/uidd:/usr/sbin/nologin
27 tcpdump:x:108:113:/:/nonexistent:/usr/sbin/nologin
28 landscape:x:109:115:/:/var/lib/landscape:/usr/sbin/nologin
29 pollinate:x:110:1:/:/var/cache/pollinate:/bin/false
30 sshd:x:111:65534:/:/run/sshd:/usr/sbin/nologin
31 systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
32 vlaine:x:1000:1000:vlaine:/home/vlaine:/bin/bash
33 lxd:x:998:100:/:/var/snap/lxd/common/lxd:/bin/false
```

Scripting BASH

Les boucles - for

- Rappel sur « for »: Bouclage sur une liste de valeurs
 - La boucle for permet de parcourir une liste de valeurs et de boucler autant de fois qu'il y a de valeurs

```
POUR variable PRENANT valeur1 valeur2 valeur3  
FAIRE  
-----> effectuer_une_action  
VALEUR_SUIVANTE
```

```
#!/bin/bash  
for variable in 'valeur1' 'valeur2' 'valeur3'  
do  
    echo "La variable vaut $variable"  
done
```

Scripting BASH

Les boucles - for

- Les valeurs de for n'ont forcément à toutes être dans le code, nous pouvons boucler sur la sortie d'une commande, ou un fichier.
- Ici nous avons inclus des commandes dans des variables pour faire une recherche.

```
#!/bin/bash
```

```
command=$(find /root -name "*.sh"); #Ajouter une commande à  
une variable comme ça exécute la commande dans le script !!!!
```

```
var=`pwd` #Entre accent grave est le moyen d'envoyer STDOUT  
dans la variable.
```

```
echo "Ce script va chercher tout les scripts avec l'extension .sh,  
vous êtes dans le dossier $var"
```

```
for files in $command  
do
```

```
    echo "J'ai trouvé : $files" | awk -F ":" '{ $1 = ""; print $2 }'
```

```
done
```

Scripting BASH

Les boucles - for

- Vous pouvez sans problème y ajouter des commandes utilisant les variables, ici pour l'exemple nous copions les fichiers du résultats de la recherche, puis les compressons à un endroit voulu.

```
#!/bin/bash
var=`pwd`

echo "Ce script va chercher tout les scripts avec l'extension .sh, vous êtes dans le dossier $var"

for files in `find /root -name "*.sh"`
do
    echo "J'ai trouvé : $files" | awk -F ":" '{ $1 = ""; print $2 }'

    cp -r $files /root/scripts/saves/ > /dev/null 2>&1
    tar -czf /root/scripts/saves/saves_`date -l`.tar.gz /root/scripts/saves/*.sh > /dev/null 2>&1

    echo "Le fichier $files a été sauvgardé"
done
```

Scripting BASH

Les boucles - for

- For peut parcourir une liste de valeur, il est donc possible de rediriger une STDOUT pour boucler dessus, précédemment nous l'avons fait sur les fichiers. Mais cela est possible avec n'importe quelle commande.
- Exemple : la commande « seq 1 10 » compte de 1 à 10 et affiche les numéros en sortie, il est donc possible de boucler autant de fois qu'il y a de nombre en sortie.

```
root@sweb01:~/scripts# seq 1 10
1
2
3
4
5
6
7
8
9
10
root@sweb01:~/scripts#
```

```
#!/bin/bash

for i in `seq 1 2 10`;
do
    echo $i
done
```

```
root@sweb01:~/scripts# ./for_i.sh
1
3
5
7
9
root@sweb01:~/scripts#
```

Scripting BASH

Les boucles - for

```
# syntaxe basique
for i in 1 2 3 4 5 6 7 8 9 10; do
    # mon travail
    echo $i
done
```

```
# syntaxe avancée
for ((a=1; a<=10 ; a++)); do
    # mon travail
    echo $a
done
```

```
# autre syntaxe avancée
for a in {1..10}; do
    # mon travail
    echo $a
done
```

```
# syntaxe avec seq
for b in $(seq 1 10); do
    # mon travail
    echo $b
done
```

Scripting BASH

Les fonctions

- Une fonction est un ensemble d'instructions permettant d'effectuer plusieurs tâches, avec des paramètres d'entrée différents.
- Nous pouvons appeler une fonction autant de fois que nécessaire dans un script.
- Le nombre de fonction est illimité mais doivent avoir un nom différent.
- Nous pouvons utiliser nos variables locales ou globales dans une fonction.
- Une fonction est un élément qui gèrera un élément spécifique et récurrent dans notre script.

```
# déclaration de fonction méthode 1  
maFonction ()
```

```
{  
  bloc d'instructions  
}
```

```
#appel de ma fonction
```

```
maFonction
```

```
# déclaration de fonction méthode 2  
function maFonction
```

```
{  
  bloc d'instructions  
}
```

```
#appel de la fonction
```

```
maFonction
```


Scripting BASH

Les fonctions

- Une fonction doit être le plus générique possible.
- Une fonction comprend plusieurs instructions.
- Vous pouvez l'appeler autant de fois que vous voulez dans votre code : laissez donc votre fonction assez générique et faites-la varier grâce aux paramètres que vous pouvez préciser en l'appelant.
- Vous pouvez déclarer une fonction Bash avec `maFonction ()` ou `function maFonction`.
- Vous faites ensuite appel à votre fonction `maFonction()` en tapant simplement dans votre code `maFonction`.
- Il est tout à fait possible de passer des paramètres à une fonction.


Scripting BASH

Les fonctions

- Exemple d'une fonction simple :

```
#!/bin/bash

# déclaration d'une fonction
function maFonction()
{ local varlocal="je suis la fonction"
  echo "$varlocal"
  echo "Nombres de paramètres : $#"
```



```
  echo $1
  echo $2
}

# appel de ma fonction
maFonction "Hello" "World!"
```

```
je suis la fonction
Nombres de paramètres : 2
Hello
World!
```



Administration système - Services GNU / Linux

Vincent LAINE



NTP - Le serveur de temps

- Il est important d'avoir l'heure synchroniser entre toutes les machines d'un parc.
- Certains service nécessite d'avoir l'heure à + ou - 5min pour fonctionner (kerberos)
- apt-get install ntp
- Pour la dernière version nous pouvons compiler le paquet
 - <http://www.ntp.org/downloads.html>
- ntpd - Démon NTP (Network Time Protocol)
- ntpq - programme de requête NTP standard
- ntpdc - programme spécial de requête NTP
- ntpdate -q 127.0.0.1- régler la date et l'heure via NTP
- ntptrace - trace une chaîne de serveurs NTP jusqu'à la source principale
- L'option de configuration (./configure) --enable-ntp-signd est nécessaire si on veut configurer l'heure sur un samba-ad-dc
- Dépendances : libevent-core-2.1-7 libevent-pthreads-2.1-7 libnss-systemd libopts25 libpam-systemd libsystemd0 sntp systemd systemd-sysv
- Installation : <http://www.linuxfromscratch.org/blfs/view/svn/basicnet/ntp.html>

NTP - Le serveur de temps

- Fichier de configuration
 - `/etc/ntp.conf`
- `ntpd -q`
 - Démarrer le démon et synchro (peut être ajouté au cron)
- `ntpq -p`
 - Vérifier les serveur ntp
- `/etc/timezone`
- `timedatectl`
 - Vérifier les paramètres de temps (UTC)
- `timedatectl list-timezones`
 - Lister les timezone disponibles
- `timedatectl set-timezone Europe/Paris`
 - Changer la timezone

NTP - Le serveur de temps

- ntpq -p

```
root@SAMBA:~# ntpq -p
      remote               refid              st t when poll reach  delay  offset  jitter
=====
LOCAL(0)      .LOCL.             10 l 629   64    0   0.000   0.000   0.000
*portunus.boudot 195.176.26.206      2 u  18   64   377   9.775   1.035   8.974
+routerzw.connec 131.176.107.13      2 u 224   64   110  20.932   5.719   4.209
+s206-75-147-25. 192.168.10.254      2 u  22   64   377 147.014  -3.128   9.149
```

- Sur la sortie ci-dessus on voit que mon client possède 3 références de temps,
- Si le premier caractère d'une ligne de référence n'est pas vide (*, + ou) alors il y a un qualifcateur pour cette référence. Au démarrage du daemon ntpd aucun qualifcateur n'est présent car le daemon a besoin de pooler les références avant de décider laquelle est la plus stable.
- Une * représente la référence choisie pour se mettre à jour, un + représente une référence de bonne qualité qui pourra être choisie par ntpd si la référence choisie ne répond plus.
- **remote** : c'est normalement l'adresse IP de référence.
- **refid** : indique le type de ma référence. .LOCL. représente l'horloge local
- **st** : c'est le stratum de la référence. Mon horloge local a un stratum de 10 tandis que la référence choisie a un stratum de 2. Plus le stratum est petit, plus la chaîne de serveur est courte.
- **when** et **pool**: à chaque fois que when atteint la valeur de pool, le daemon ntpd interroge la référence et réinitialise le compteur when à 0
- **reach** : la colonne reach indique si les dernières requêtes vers la référence ont été effectuées avec succès, c'est à dire que l'on a bien reçu des données et que le temps était synchronisé.
 - On commence à la valeur 0 et pour chaque requête réussie on décale de 1 vers la gauche. La séquence depuis le début est donc 0, 1, 3, 7, 17, 37, 77, 177, 377
- Lorsque la valeur est à 377, cela veut dire que les 8 dernières requêtes ont été effectuées avec succès.
- **delay**, **offset** (compensation) et **jitter** (gigue, variation anormale) : ce sont les valeurs de temps qui résultent des requêtes effectuées. Les valeurs sont en millisecondes. Le delay est dérivé du roundtrip time des requêtes. L'offset représente la différence de temps entre l'horloge système et le temps de référence et le jitter indique la magnitude de jitter entre plusieurs requêtes de temps.

VSFTPD - Le serveur FTP Linux

- Lors de la création et la gestion d'un site internet, vous utiliserez constamment un serveur FTP.
- FTP (File Transfer Protocol) permet de stocker ou récupérer des fichiers sur le serveur. L'avantage est que vous pourrez ainsi manipuler vos fichiers depuis n'importe quel ordinateur à travers le monde.
- Il permet donc le transfert de fichiers entre un client et un serveur.
- Un client FTP est une application qui s'utilise depuis un ordinateur. Elle est utilisée pour importer ou exporter des fichiers d'un serveur FTP.
- Logiciel client FTP : Filezilla.

VSFTPD - Le serveur FTP Linux

TP

1. Installer le paquet vsftpd + IP fixe
2. Copier le fichier de configuration original pour l'avoir en « backup »
3. Créer un utilisateur « ftp » et définissez lui un mot de passe
 1. Si vous avez déjà un utilisateur « ftp » supprimez le puis recréez le.
4. Créer un dossier « ftp » dans /opt/
5. Lier le dossier à l'utilisateur « ftp »
6. Redémarrer le serveur FTP
7. Créer un dossier dans /opt/ftp/ que vous nommerez avec votre nom d'utilisateur
8. Fichier de configuration du FTP : /etc/vsftpd.conf
9. Vérifier si les paramètres suivants sont configurés :
 1. « local_enable=YES »
 2. « write_enable=YES »
 3. « chroot_local_user=YES »
 4. « allow_writeable_chroot=YES »
10. Puis redémarrer le service
11. Tester la connexion
 1. Depuis un navigateur : [ftp://\[IP_SERVEUR\]](ftp://[IP_SERVEUR])
 2. Et depuis Filezilla
12. En ligne de commande
 1. Paquet vsftpd ou ftp
 2. ftp -p [IP] : Se connecter à un serveur ftp
 3. bye : Se déconnecter
 4. Nom d'utilisateur anonyme : anonymous (exemple)
 5. cd : Se déplacer dans les dossiers
 6. get test.txt : Transférer le fichier test.txt à notre machine
 7. put test.txt upload.txt : Télécharger un fichier dans le serveur avec un nouveau nom
13. Envoyer un fichier dans le dossier créé à l'étape 6.
 1. Changer les droits d'accès /opt/ftp/user
 1. chown -R ftp:ftp [...]

VSFTPD - Le serveur FTP Linux

vsftpd.conf

Options	Description	Commentaire
listen	Permet de définir si le démon est en standalone (YES) ou dirigé par (x)inetd (NO)	Partisan du (x)inetd, Partisan du standalone, chacun son choix. Personnellement, je préfère le standalone...
anonymous_enable anon_root=/var/ftp/ anon_upload_enable=YES anon_mkdir_write_enable=YES	Permet d'accepter les connexions anonymes Défini le dossier par défaut Autoriser upload par Anonymous Autoriser Anonymous à créer des dossier	Tout dépend du but de votre serveur. Par défaut, je conseillerai de rejeter les connexions anonymes. Mais si votre serveur est au sein d'un réseau et que tout le monde doit y accéder, alors mettez YES, sinon NO.
local_enable	Oblige les personnes à s'identifier avec un compte utilisateur	Dans tous les cas, je dis YES. Si une personne a un compte, le serveur ftp est présent pour elle. Sauf si vous ne voulez pas les laisser exporter ou importer des fichiers
write_enable	Permission d'écriture	Comme les deux précédents, tout dépend de vos besoins et de la fonction de votre service ftp.
xferlog_file	Écriture d'un log des fichiers	Obligatoire selon moi pour tout administrateur digne de ce nom. Il faut savoir ce qu'il se passe surtout sur ces protocoles qui permettent les entrées/sorties de données.
ftpd_banner	Bannière d'affichage à la connexion FTP	Étrangement, je trouve très importante cette bannière qui peut sembler superflue. Pourquoi ? Parce que vous pouvez l'utiliser pour communiquer : dire sur quel serveur l'utilisateur se connecte (pratique quand on doit se connecter à divers serveurs), donner des informations sur les mises à jour, les maintenances, etc. Indispensable si vous voulez envoyer des informations.
chroot_local_user	Permet de chrooter la connexion de l'utilisateur	Quand l'utilisateur se connecte en ftp, il arrive dans son répertoire home(défini dans /etc/passwd). Cette option active vous permet de l'obliger à rester dans ce répertoire (ou tout du moins de ne pas redescendre dans l'arborescence). Il reste compartimenté dans son répertoire home. Très intéressant, si vous ne voulez pas qu'il se balade partout et télécharge des fichiers système.

VSFTPD - Le serveur FTP Linux

vstftp.conf

1. Accorder un accès en écriture à l'ensemble du dossier de base :
 1. `allow_writeable_chroot=YES`
2. Utiliser un répertoire différent pour les download et upload. (Vous pourrez upload dans le dossier upload, et uniquement télécharger depuis le dossier « ftp »)
 1. Créer un répertoire « ftp » dans `/home/[USER]`
 1. Droits : 755
 2. Accès : root:root
 2. Créer un répertoire « upload » dans le dossier « ftp »
 1. Droits : 755
 2. Accès : user:user (utiliser l'utilisateur voulu)
 3. Ajouter à `vsftpd.conf`
 1. `user_sub_token=$USER`
 2. `local_root=/home/$USER/ftp`
3. Autoriser seulement certains utilisateurs à se connecter :
 1. `userlist_enable=YES`
 2. `userlist_file=/etc/vsftpd.userlist`
 3. `userlist_deny=NO`
 4. Puis ajouter les utilisateurs dans le fichier `vsftpd.userlist`
5. Documentation : <https://www.howtoforge.com/tutorial/ubuntu-vsftpd/>
 - <https://doc.ubuntu-fr.org/vsftpd>

VSFTPD - Le serveur FTP Linux

vstftp.conf

vsftpd over SSL

- Générer la clef :
 - `openssl req -new -x509 -days 365 -nodes -out /etc/ssl/private/vsftpd.cert.pem -keyout /etc/ssl/private/vsftpd.key.pem`
- Editer le fichier vsftpd.conf

```
ssl_enable=YES
rsa_cert_file=/etc/ssl/private/vsftpd.cert.pem
rsa_private_key_file=/etc/ssl/private/vsftpd.key.pem

allow_anon_ssl=NO

force_local_data_ssl=YES
force_local_logins_ssl=YES

ssl_tlsv1=YES
ssl_sslv2=YES
ssl_sslv3=YES

require_ssl_reuse=YES
#ssl_ciphers=HIGH
```

Serveur WEB

- Qu'est ce qu'un serveur Web ?
 - Un serveur web, est l'intermédiaire entre le site web (l'application) et les machines clientes
 - Il extrait le contenu du serveur sur chaque requête d'utilisateur et le transmet au web
 - Le plus grand défi d'un serveur web est de servir simultanément plusieurs et différents utilisateurs web chacun demandant des pages différentes.
 - Les serveurs web traitent les fichiers écrits dans différents langages de programmation tels que PHP, Python, Java et autres.
 - Ils les transforment en fichiers HTML statiques et diffusent ces fichiers dans le navigateur des utilisateurs web.
 - Quand vous entendez le mot serveur web, considérez-le comme l'outil responsable de la communication serveur-client.

Service Web - Apache

- Apache
 - Apache fait tourner presque 35% des sites web à travers le monde
 - En réalité apache se nomme sous sa forme longue : Serveur Apache HTTP
 - Développé par Apache software Foundation
 - Première version d'apache date de 1995 (soit près de 25 ans !!)
 - Nginx (1^{er} serveur WEB)



Serveur Web - Apache

- Qu'est ce qu'un serveur Web Apache ?
 - Bien que nous appelions Apache un serveur web, ce n'est pas un serveur physique mais plutôt un logiciel qui s'exécute sur un serveur.
 - Son travail consiste à établir une connexion entre un serveur et les navigateurs des visiteurs du site web (Firefox, Google Chrome, Safari, etc.)
 - Tout en délivrant des fichiers entre eux (structure client-serveur). Apache est un logiciel multiplateforme
 - Il fonctionne donc à la fois sur les serveurs Unix et Windows.

Serveur Web - Apache

- Comment fonctionne Apache ?
 - Le travail de base de tous les serveurs Web est d'accepter les requêtes des clients (p. ex. le navigateur Web d'un visiteur)
 - Puis d'envoyer la réponse à cette requête (p. ex. les composantes de la page qu'un visiteur souhaite voir).
 - Le serveur web Apache dispose de modules qui ajoutent plus de fonctions à son logiciel, tels que MPM (pour la gestion des modes multiprocesseurs) ou mod_ssl pour activer le support SSL v3 et TLS (lecture suggérée: TLS vs SSL).
 - Quelques caractéristiques communes vues dans Apache incluent :
 - .htaccess
 - IPv6
 - FTP
 - HTTP/2
 - Perl, Lua et PHP
 - Limitation de la bande passante
 - Équilibrage de charge
 - Réécriture d'URL
 - Suivi des sessions
 - Géolocalisation basée sur l'adresse IP

Apache2 - Le serveur WEB Administration

- Pour installer apache2
 - apt-get install apache2
 - Utilisateur par défaut apache2 : www-data
- Les configurations dans apache2 : Comprendre
- Vous trouverez la configuration d'Apache dans le répertoire :
 - /etc/apache2/
- La configuration est ensuite découpée en un grand nombre de fichiers. Le fichier principal se nomme :
 - apache2.conf
- Dossier contenant vos sites web de base :
 - /var/www/html
- Les lignes commençant par un # sont des commentaires.

Apache2 - Le serveur WEB

/etc/apache2/apache2.conf

- Ils définissent l'utilisateur et le groupe sous lesquels tourne Apache (sous forme de variable)
 - User \${APACHE_RUN_USER}
 - Group \${APACHE_RUN_GROUP}
- Par défaut l'utilisateur apache est « www-data »
- Vous trouverez ensuite la configuration des logs dans apache2.conf :

```
ErrorLog ${APACHE_LOG_DIR}/error.log

#
# LogLevel: Control the severity of messages logged to the error_log.
# Available values: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the log level for particular modules, e.g.
# "LogLevel info ssl:warn"
#
LogLevel warn
```

Apache2 - Le serveur WEB

/etc/apache2/[conf-mod-site]-[available-enabled]

- Puis le dossier /etc/apache2/ inclut aussi tous les fichiers de configuration placés eux même dans des dossiers :
 - conf-enabled/
 - mods-enabled/
 - sites-enabled/
- Vous avez dans ces dossier (enabled) les configurations que VOUS AVEZ « ACTIVE » grâce à la commande « a2en... ».
- A savoir, le « conf » correspond aux configurations additionnelles d'apache. Le « mods » correspond aux modules apaches disponible/active. Et le « sites » correspond à la configuration de nos sites web.
 - conf-available/
 - mods-available/
 - sites-available/www.vlne.fr.conf
- Puis dans les dossiers available, vous avez vos fichiers de configuration « disponible ».

/etc/apache2/	Fonction	Commande
conf-enabled/	CONFIGURATIONS ACTIVEES	a2disconf
mods-enabled/	MODULES ACTIVES	a2dismod
sites-enabled/	SITES ACTIVES	a2dissite
conf-available/	CONFIGURATION DISPONIBLE/DESACTIVEE	a2enconf
mods-available/	MODULES DISPONIBLE/DESACTIVEE	a2enmod
sites-available/	SITES DISPONIBLE/DESACTIVEE	a2ensite

Apache2 - Le serveur WEB

Les commandes a2[en|dis][site|mod|conf]

- Les commandes a2[...] nous permettent de gérer différents paramètres liés à l'activation/désactivation d'un site web, d'un mod ou d'une configuration additionnelle.
- Elle se décompose en 3 parties :
- La première partie « a2 » veut dire apache2, c'est le préfixe.
- Ensuite nous la complétons avec « en » ou « dis ». Si on met « en », c'est pour activer, si on met « dis », c'est pour désactiver.
- Puis la dernière partie correspond au type de configuration que l'on veut activer. « site » pour une site web, « mod » pour un module et « conf » pour une configuration additionnelle.

a2	en dis	site mod conf
Apache2	Enable/disable	Site web, Modules, configurations

Apache2 - Le serveur WEB

TP

1. Mise en service d'un serveur WEB avec apache2
 1. Installation et vérification avec la page par défaut
2. Désactiver le site web par défaut (000-default.conf)
3. Créer un nouveau fichier de configuration de site web
 1. En se basant sur 000-default.conf
4. Activer ce site web
5. Vérifier que vous y avez accès
6. Changer l'emplacement initial du site web (/var/www/html) vers une autre destination (/opt/www/html)
 1. Changer la directive « DocumentRoot » dans le fichier de configuration du site
 2. Et activer l'accès à /opt/www dans apache2.conf
7. A chaque modifications, ne pas oublier systemctl reload apache2

```
<VirtualHost *:80>
# The ServerName directive sets the request scheme
# the server uses to identify itself. This is used
# redirection URLs. In the context of virtual hosts
# specifies what hostname must appear in the request
# match this virtual host. For the default virtual host
# value is not decisive as it is used as a last resort
# However, you must set it for any further virtual hosts
#ServerName www.example.com

ServerAdmin webmaster@localhost
DocumentRoot /var/www/html

# Available loglevels: trace8, ..., trace1, debug,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

# For most configuration files from conf-available/
# enabled or disabled at a global level, it is possible
# include a line for only one particular virtual host
# following line enables the CGI configuration for this
# after it has been globally disabled with "a2disca
#Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

```
<Directory /var/www/>
Options Indexes FollowSymLinks
AllowOverride None
Require all granted
</Directory>
<Directory /opt/www/>
Options Indexes FollowSymLinks
AllowOverride None
Require all granted
</Directory>
```

Apache2 - Le serveur WEB

Sécuriser son site web

- Sécuriser la configuration de son site web
- Afin que votre site donne le moins d'informations possible sur son système (exemple la version, le nom etc...) lors d'une requête HTTP, apache regroupe différentes directives de sécurité dans le fichier :
 - `/etc/apache2/conf-available/security.conf`
- Pour donner le moins d'infos possible, vérifier les directives suivantes :

```
ServerTokens Prod
ServerSignature Off
```

- En plus de cela, il est recommandé de désactiver les modules installés qui ne vous servent pas.
 - <https://httpd.apache.org/docs/current/fr/mod/>

Apache2 - Le serveur WEB

Sécuriser son site web

- Restreindre l'accès à certaines ressources en fonction :
 - de l'IP
 - par login/mot de passe
 - ...
- La directive pour sécuriser un élément de votre est « Require ». Cette directive s'ajoute dans le fichier de configuration de votre site web.
- Les directives dans un fichier de configuration se définissent comme suit (/etc/apache2/sites-available/www.votresite.conf).
- Les directives sont lus dans l'ordre.
- Ici l'IP 1.10 pourra accéder au dossier secret sans mot de passe, les autres devront s'authentifier.
- Les directives Auth* précisent le fonctionnement de cette autorisation par mot de passe :
 - AuthType Basic : indique une authentification par login/mot de passe simple
 - AuthName : permet de définir le titre de la fenêtre d'authentification qui sera affiché aux clients
 - AuthBasicProvider file : indique que les comptes utilisateurs sont définis dans un fichier
 - AuthUserFile : précise le fichier où sont enregistrés les comptes utilisateurs

```
<Directory /var/www/www.abalone.fr/secret>
AuthType Basic
AuthName "Accès restreint aux utilisateurs authentifiés"
AuthBasicProvider file
AuthUserFile "/etc/apache2/secret_users"
Require ip 192.168.1.10
Require valid-user
</Directory>
```

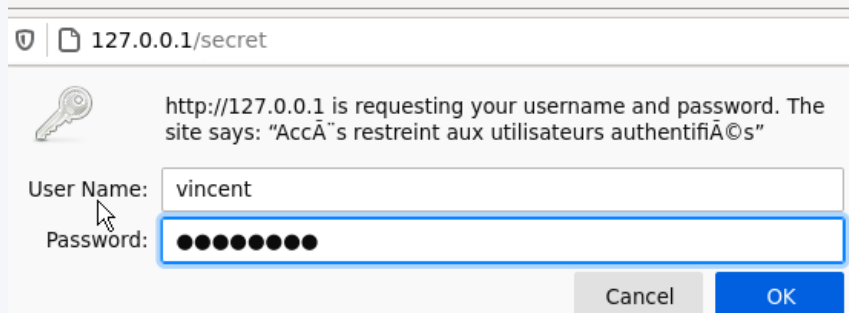
Apache2 - Le serveur WEB

Sécuriser son site web

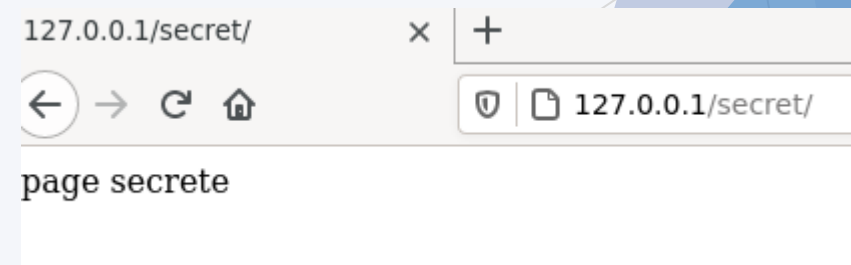
- Le fichier d'authentification « secret_users » va contenir les comptes/mot de passes autorisés grâce au htpasswd.
- htpasswd est une commande permettant de générer des comptes sécurisés pour apache2.
- Il faut donc créer le fichier en premier lieu, en y ajoutant un premier utilisateur.

```
htpasswd -c /etc/apache2/secret_users vincent
```

- Ici je crée le premier utilisateur du htpasswd « vincent ».
- A noter que l'option « -c » ne s'utilise que la première fois, pour ajouter d'autres users dans le même fichier vous devrez enlever cette option.



A screenshot of a web browser security warning dialog box. The address bar shows "127.0.0.1/secret". The message says: "http://127.0.0.1 is requesting your username and password. The site says: 'Accès restreint aux utilisateurs authentifiés'". There are input fields for "User Name:" (containing "vincent") and "Password:" (containing masked characters). At the bottom are "Cancel" and "OK" buttons.



Apache2 - Le serveur WEB

Sécuriser son site web

- Exemple d'un fichier de configuration de site en HTTP Apache2.

```
<VirtualHost *:80>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
#ServerName www.example.com

ServerName www.abalone.fr
ServerAlias abalone.fr
ServerAdmin webmaster@abalone.fr
DocumentRoot /var/www/abalone.fr
CustomLog ${APACHE_LOG_DIR}/abalone.fr-access.log combined
ErrorLog ${APACHE_LOG_DIR}/abalone.fr-error.log

<Directory /var/www/abalone.fr>
Options All
AllowOverride None
</Directory>
```



```
<Directory /var/www/abalone.fr/secret>
AuthType Basic
AuthName "Accès restreint aux utilisateurs authentifiés"
AuthBasicProvider file
AuthUserFile "/etc/apache2/secret_users"
Require ip 192.168.1.10
Require valid-user
</Directory>
# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```


Apache2 - Le serveur WEB

Sécuriser son site web

- Il est possible de sécuriser son site web en HTTPS de plusieurs manières. Soit un utilisant un certificat auto-signé, soit un certificat fourni par une autorité de certification.
- Les intérêts du HTTPS :
 - Connexions chiffrées
 - Empêche l'interception de mots de passe, informations de session, cookies, etc.
 - le serveur est identifié par un certificat ce qui permet au client de pouvoir s'assurer que le serveur auquel il se connecte est bien celui qu'il prétend être
 - HTTPS est maintenant la norme et les moteurs de recherche, Google en tête, référencent mieux les sites HTTPS que les sites HTTP.
- Pour pouvoir configurer votre site en HTTPS, vous devrez générer un certificat pour votre serveur.
- Tout le monde peut générer un certificat.
- Pour que les clients soient sûrs que le certificat qui leur est présenté à la connexion correspond au serveur qu'ils veulent joindre, ils font confiance à une autorité de certification pour valider le certificat du serveur.

Apache2 - Le serveur WEB

Sécuriser son site web

- Le sécurisation d'un serveur WEB par certificat auto-signé se fait avec le paquet openssl.
- Il est également possible de générer un certificat avec Let's Encrypt, qui est une autorité de certification gratuite, à but non lucratif, et reconnue comme officiel par les « client », elle fourni des certificats SSL.
 - `apt install certbot python3-certbot-apache`
 - <https://www.digitalocean.com/community/tutorials/how-to-secure-apache-with-let-s-encrypt-on-ubuntu-20-04-fr>
- Nous verrons dans ce cours la sécurité avec un certificat SSL auto-signé pour Apache. Nous ne passerons pas let's encrypt.

Apache2 - Le serveur WEB

Sécuriser son site web - HTTPS

- TLS, ou “transport layer security” - et son prédécesseur SSL - sont des protocoles utilisés pour envelopper le trafic normal dans une enveloppe protégée et cryptée. Grâce à cette technologie, les serveurs peuvent envoyer en toute sécurité des informations à leurs clients sans que leurs messages soient interceptés ou lus par une partie extérieure.
- Après installé apache2, si vous avez activé ufw, autorisez le.
 - `ufw allow "Apache Full"`
- Il nous faut ensuite activer le module SSL (`mod_ssl`) d'Apache.
 - `a2enmod ssl`
 - `systemctl restart apache2`
- Maintenant qu'Apache est prêt nous devons générer le certificat.

Apache2 - Le serveur WEB

Sécuriser son site web - HTTPS

- Le certificat stockera quelques informations de base sur votre site, et sera accompagné d'un fichier clé qui permet au serveur de traiter les données cryptées en toute sécurité.
- Nous pouvons créer les fichiers de clés et de certificats SSL avec la commande openssl :

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/auto-sign.key -out /etc/ssl/certs/auto-sign.crt
```

- **openssl** : c'est l'outil de ligne de commande utilisé pour la création et la gestion des certificats, clés et autres fichiers OpenSSL.
- **req -x509** : cela spécifie que nous voulons utiliser la gestion des demandes de signature de certificats (CSR) X.509. X.509 qui est une norme d'infrastructure de clé publique à laquelle SSL et TLS adhèrent pour la gestion des clés et des certificats.
- **-nodes** : cela indique à OpenSSL de ne pas utiliser l'option de sécurisation de notre certificat par une phrase de passe. Nous avons besoin qu'Apache soit capable de lire le fichier, **sans intervention de l'utilisateur**. Une phrase de passe empêcherait que cela se produise, puisque nous devrions la saisir après chaque redémarrage.
- **-days 365** : cette option fixe la durée pendant laquelle le certificat sera considéré comme valide. Ici, nous l'avons fixée pour un an. De nombreux navigateurs modernes refusent les certificats dont la durée de validité dépasse un an.
- **-newkey rsa:2048** : cette option précise que nous voulons générer un nouveau certificat et une nouvelle clé en même temps. Nous n'avons pas créé la clé nécessaire pour signer le certificat lors d'une étape précédente, nous devons donc la créer en même temps que le certificat. La partie rsa:2048 lui demande de fabriquer une clé RSA de 2048 bits.
- **-keyout** : cette ligne indique à OpenSSL où placer le fichier de clé privée généré que nous créons.
- **-out** : cela indique à OpenSSL où placer le certificat que nous créons.

Apache2 - Le serveur WEB

Sécuriser son site web - HTTPS

- Quand vous aurez lancé la précédente commande, vous devrez répondre correctement aux questions. C'est surtout la direction **Common Name** qui est importante !!
- Il est important que **Common Name** (IP ou nom de domaine) corresponde à ce que vous taperez dans la barre d'adresse du navigateur.

```
root@swb01:/etc/apache2# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
/etc/ssl/private/auto-sign.key -out /etc/ssl/certs/auto-sign.crt
auto-sign.key -out /etc/ssl/certs/auto-sign.crt
Generating a RSA private key
.....+++++
.....+++++
writing new private key to '/etc/ssl/private/auto-sign.key'
-----
[...]
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:FR
State or Province Name (full name) [Some-State]:IDF
Locality Name (eg, city) []:PARIS
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Abalone Inc
Organizational Unit Name (eg, section) []:IT Dept
Common Name (e.g. server FQDN or YOUR name) []:abalone.fr
Email Address []:webmaster@abalone.fr
```

Apache2 - Le serveur WEB

Sécuriser son site web - HTTPS

- Une fois le certificat généré, remplacer le port du virtual host par 443 (au lieu de 80) et ajouter ces lignes pour déclarer le SSL.
 - `SSLEngine on`
 - `SSLCertificateFile /etc/ssl/certs/auto-sign.crt`
 - `SSLCertificateKeyFile /etc/ssl/private/auto-sign.key`
- Puis si ce n'est pas déjà fait, activez votre site web et redémarrez apache2.
 - `apache2ctl configtest`


```
~# apache2ctl configtest
AH00558: apache2: Could not reliably determine the server's fully qualified
domain name, using sweb01.abalone.fr. Set the 'ServerName' directive globally
to suppress this message
Syntax OK
```

- La première ligne est un message vous indiquant que la directive `ServerName` n'est pas définie globalement. Si vous ne voulez plus voir ce message s'afficher, vous pouvez définir `ServerName` en spécifiant le nom de domaine ou l'adresse IP de votre serveur dans `/etc/apache2/apache2.conf`. Ceci est facultatif car le message ne fera pas de mal.

Apache2 - Le serveur WEB

Sécuriser son site web - HTTPS

- Connectez vous ensuite sur votre siteweb en https pour vérifier, acceptez le risque et poursuivre :



Warning: Potential Security Risk Ahead

Firefox detected a potential security threat and did not continue to abalone.fr. If you visit this site, attackers could try to steal information like your passwords, emails, or credit card details.

[Learn more...](#)

[Go Back \(Recommended\)](#) [Advanced...](#)

abalone.fr uses an invalid security certificate.

The certificate is not trusted because it is self-signed.

Error code: [MOZILLA_PKIX_ERROR_SELF_SIGNED_CERT](#)

[View Certificate](#)

[Go Back \(Recommended\)](#) [Accept the Risk and Continue](#)

Apache2 - Le serveur WEB

Sécuriser son site web - HTTPS

- Pour terminer il faut forcer la redirection d'Apache du HTTP vers HTTPS.
- Ouvrez votre fichier de configuration de site, et créez un virtualHost sur le port 80 en plus de celui en 443.
- Ensuite, utilisez Redirect (Rediriger) pour faire correspondre les requêtes et les envoyer au VirtualHost SSL. Veillez à inclure la barre oblique :

```
<VirtualHost *:80>  
    ServerName www.abalone.fr  
    ServerAlias abalone.fr  
    Redirect / https://abalone.fr  
</VirtualHost>
```

- Testez la configuration et redémarrez apache. Pour tester connectez-vous en HTTP à votre site et vérifiez la redirection vers HTTPS.



Partage de fichiers sous Linux

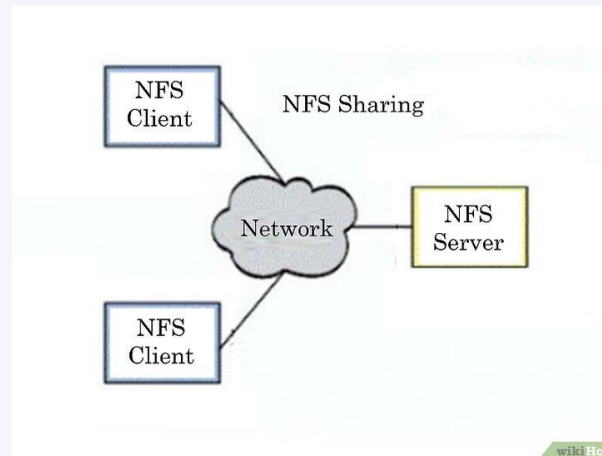
- Le partage NFS (Network File system)
 - Inventé par SUN MICROSYSTEM en 1984
 - Le partage NFS est un partage de fichier existant sous Windows ou linux
 - Il utilise les protocoles TCP/IP, RPC et XDR
 - NFS est le standard des fichier partagé sur un réseaux hétérogène (c'est-à-dire avec des OS différents)
 - Le principal intérêt de NFS réside donc dans le fait qu'il fonctionne correctement et est performant sur la plupart des systèmes d'exploitation, sans avoir besoin de beaucoup de configuration.

Partage de fichiers sous Linux

- Serveur NFS Désigne le système qui possède physiquement les ressources (fichiers, répertoires) et les partages sur le réseau avec d'autres systèmes.
- Client NFS. Désigne un système qui monte les ressources partagées sur le réseau. Une fois montées, les ressources apparaissent comme si elles étaient locales.
- NFS ne s'occupe pas des droits d'accès aux fichiers et répertoires, il laisse le système d'exploitation s'occuper de la sécurité.
- Celle-ci est donc gérée de la même façon que pour les fichiers locaux, avec les droits UNIX standards. C'est-à-dire les droits UGO

Partage de fichiers sous Linux

- Que permet NFS ?
 - Une gestion totale de la sécurité avec Kerberos
 - Meilleur support du trafic
 - Système de maintenance simplifié
 - Compatible avec les systèmes Unix, Windows et Mac
 - Protocole de transfert TCP (et non plus UDP)
 - Fonctionnement avec seul port (le 2049), pour simplifier la configuration des pare-feu
 - Utilisation d'opérations « stateful »
 - Le travail collaboratif



Partage de fichiers avec Samba

- Qu'est ce que Samba ?
 - Samba a été créé par Andrew Tridgell en 1991. Au commencement il développe un programme de gestion de fichiers basé sur le protocole propriétaire SMB.
 - Samba est géré et développé par la SAMBA TEAM
 - Logiciel Open source distribué sous licence GPL GNU
 - Samba est un ensemble d'outils permettant de partager des ressources telles que des imprimantes et des fichiers sur un réseau hétérogène.

Partage de fichiers avec Samba

- Que pouvons nous faire avec Samba ?
 - définir qui a accès au dossier au fichier en "lecture" ou en "écriture«
 - définir si le mot de passe est obligatoire ou pas
 - définir les utilisateurs
 - définir les droits que l'on souhaite lors de la création d'un dossier ou fichier
 - décider de cacher ou pas le chemin d'un dossier
 - décider ou pas de le partager.
 - Fonctionne particulièrement bien avec LDAP

Partage de fichiers avec Samba

- Samba peut aussi faire office de Contrôleur de domaine sur un réseau (depuis 2012)
- Fonctionne aussi bien avec la version HEIMDAL Kerberos que MIT Kerberos
- Plus communément utilisé avec OpenLDAP
- Samba peut distribuer des tokens d'identifications (clé) Kerberos

Partage de fichier avec Samba

Administration - Partage privé

- Les directives :

path	indique le chemin du répertoire partagé. Dans toutes les sections du fichier de configuration, il est possible d'utiliser certaines variables. Ici, vous utilisez %u qui sera automatiquement remplacé dans le chemin par le nom de l'utilisateur. C'est très pratique car sans ça, il aurait fallu créer un partage pour chaque utilisateur.
read only	un no indique que le partage sera en lecture-écriture, un yes indique que le partage sera en lecture seule. Il est possible de configurer un partage en lecture seule et ensuite de configurer des exceptions pour certains utilisateurs en ajoutant ensuite le paramètre write list = user1 user2 .
guest ok	indique si les comptes invités sont autorisés à accéder au partage
force create mode et force directory mode	indiquent les droits donnés respectivement aux nouveaux fichiers et répertoires créés sur le partage
valid users	indique les noms des utilisateurs autorisés à accéder au partage. Vous pouvez également indiquer des noms de groupes qui commencent par @ . Sur Ubuntu, le groupe sambashare est créé automatiquement à l'installation de samba. Ici seuls les utilisateurs appartenant à ce groupe pourront accéder au partage [home]
force user et force group	indiquent l'utilisateur et le groupe système auquel sera associé l'utilisateur distant qui se connecte. Comme pour le partage NFS, ici tous les utilisateurs qui accèdent à /export/shared seront reconnus comme nobody:nogroup .

Partage de fichier avec Samba (standalone)

Administration - Partage public

- Installer le paquet « samba »
- Nom du service Samba : smbd
 - `systemctl status smbd`
- Fichier de configuration samba :
 - `/etc/samba/smb.conf`
- Tester le fichier de configuration samba :
 - `Testparm`
- Droits du dossier public
 - `Nobody:sambashare`
 - `775`
 - OU
 - `root:sambashare`
 - `777`
- Exemple ; Définition d'un partage public accessible à tous (dans `smb.conf`)

```
[public]
path = /samba/public
browseable = yes
guest ok = yes
guest only = yes
writeable = yes
read only = no
create mode = 0777
directory mode = 0777
```

[public] = Nom du partage

Path = Dossier à partager

Guest ok = Active le partage invité

Guest only = Protéger en connexion invité uniquement

Writeable = Accessible en écriture | Browseable = Navigation accessible

Create mode & directory mode = Autoriser l'accès avec tout les droits

Partage de fichier avec Samba (standalone)

Administration - Partage privé

- Il est possible de créer des partages privés et sécurisés avec Samba.
- Seuls les utilisateurs membres du groupe approuvé pourront accéder à l'emplacement sécurisé avec des mots de passe.
- Créez un groupe samba appelé smbgroup pour le partage.
 - Seuls les membres y auront accès. Pour créer un groupe dans Ubuntu, exécutez les commandes ci-dessous.
- Ajouter l'utilisateur voulu au groupe
- Définir le mot de passe Samba associé à l'utilisateur
 - `smbpasswd -a user`
- Activer le compte Samba
 - `smbpasswd -e user`
- Créer le dossier à partager
 - Droits : 770
 - Accès : root:smbgroup

Exemple de définition d'un partage privé

```
[andn]  
path = /samba/andn_private  
valid users = @smbgroup  
guest ok = no  
writable = yes  
browsable = yes
```

Samba4 - l'AD, le DC, le membre ?

- Pour la mise en place d'un serveur Samba4, il nous faut déterminer à l'avance quel sera sa fonction. Nous avons plusieurs possibilités :
 1. La mise en place d'un Samba comme Active Directory Domain Controller (ADDC)
 1. Dans ce cas la, Samba sera votre ADDC Principal
 2. Joindre un serveur Samba comme second contrôleur de domaine (DC)
 1. Samba sera joint à l'ADDC Samba et agira comme second DC
 2. Il est à noter qu'il est possible de joindre un Samba DC à un ADDC Windows, mais ce n'est pas recommandé, la réplication est compliqué à mettre en place (DFS-R non pris en charge), de plus la documentation est plutôt axé sur l'ajout d'un nouveau DC à un Samba ADDC.
 3. La mise en place d'un serveur Samba comme membre du domaine. (DM)
 1. Cette méthode vous permettra d'utiliser les fonctions de Samba comme le serveur de fichier ou d'impression avec un service d'annuaire, vous pouvez l'y ajouter à un Samba ADDC ou Windows ADDC sans soucis.
 2. Cependant dans une infrastructure hybride, il faudra être consciencieux pour l'infra soit adapté aux postes Windows & Linux.
 4. La mise en place de Samba en « standalone ».

Documentation officielle : https://wiki.samba.org/index.php/Main_Page

Samba4 - Serveur de fichier (DM)

- « Un membre de domaine Samba est une machine Linux jointe à un domaine qui exécute Samba et ne fournit pas de services de domaine, tels qu'un contrôleur de domaine principal NT4 (PDC) ou un contrôleur de domaine (DC) Active Directory (AD). »
- Sur un membre du domaine Samba, vous pouvez:
 - Utilisez des utilisateurs et des groupes de domaine dans des ACL locales sur des fichiers et des répertoires.
 - Configurez les partages pour qu'ils agissent comme un serveur de fichiers.
 - Configurez les services d'impression pour qu'ils agissent en tant que serveur d'impression.
 - Configurez PAM pour permettre aux utilisateurs du domaine de se connecter localement ou de s'authentifier auprès des services installés localement.
- Il est possible de joindre Samba à un ADDC existant, et d'en faire un serveur de fichier lié à un Annuaire.
- Pour mettre samba en tant que membre du domaine, suivre la documentation officielle :
 - https://wiki.samba.org/index.php/Setting_up_Samba_as_a_Domain_Member#Choose_backend_for_id_mapping_in_winbindd
- En tant que serveur de fichier, vous pourrez définir des partages, avec des droits associés aux utilisateurs de votre ADDC (Windows ou Linux)
- Il est également possible de créer des homes directories automatiquement.
- Une fois en place, il est intéressant de noter que la gestion se fait en grande partie sur Windows (si votre ADDC est sous Windows). Si votre ADDC est un SambaV4, il est possible d'installer les RSAT sur un Windows pour l'administrer.
- «

Samba4 - Serveur de fichier (DM)

Pré-requis, étapes d'installation et choix importants lors de la mise en place d'un serveur de fichier Samba. Nous partons du principe que nous allons créer un serveur de fichier Samba en membre du domaine, associé à un ADDC Windows.

- Avoir un Active Directory Domain Controller fonctionnel, avec des niveaux fonctionnels de domaine ET de forêt Windows Server 2008/2008R2 ou 2012/2012R2.
- Un DNS résolveur fonctionnel, des enregistrements A et PTR doivent être créés manuellement.
- Le serveur Samba doit résoudre correctement l'ADDC et lui-même. Ajout du DNS résolveur et du domain search à notre Samba.
- Une fois la machine prête, installer les dépendances et vérifier la compatibilité du système de fichier.
 - Dépendances : https://wiki.samba.org/index.php/Package_Dependencies_Required_to_Build_Samba
 - Fichiers : https://wiki.samba.org/index.php/File_System_Support
- Puis configurer Kerberos et NTP (du moins un moyen d'avoir l'heure synchro avec votre ADDC) sur votre serveur Samba.

Samba4 - Serveur de fichier (DM)

- Vous devrez ensuite installer Samba. Vous avez 2 possibilités :
- L'installation par les sources :
 - https://wiki.samba.org/index.php/Build_Samba_from_Source
- L'installation par les paquets :
 - https://wiki.samba.org/index.php/Distribution-specific_Package_Installation
- L'installation par les sources nécessite plus de connaissances spécifiques, sur le système de fichier Linux, la création d'inits, l'ajout de nouveaux services, les configurations de variables d'environnement etc...
- En revanche, c'est ce type d'installation qui nous permet de bénéficier de la dernière version en date, car ce n'est pas forcément le cas avec les packages. C'est quasiment indispensable lors de la mise en place d'un ADDC ou d'un DC, et recommandé pour un DM.

Samba4 - Serveur de fichier (DM)

- Maintenant que Samba est installé nous pouvons passer à la configuration en tant que DM.
 - https://wiki.samba.org/index.php/Setting_up_Samba_as_a_Domain_Member#Configuring_Samba
- Dans la configurations, vous aurez des choix importants à faire. Il vous faudra dans un premier temps choisir votre « backend » :
 - winbind 'ad' (https://wiki.samba.org/index.php/ldmap_config_ad)
 - winbind 'rid' (https://wiki.samba.org/index.php/ldmap_config_rid)
- Les 2 backends ne fonctionnent simplement pas de la même manière.
- Winbind 'ad' nécessitera les attributs RFC2037, et notamment l'ajout d'un gid/uid manuellement à vos users dans l'ADDC.
- Winbind 'rid' quant à lui aura pour contrainte que tout les users devront utiliser le même shell et répertoire de base Unix; il est plus adaptés à un environnement « Windows ADDC, Samba DM ». De plus les uid/gid seront « converties » depuis l'annuaire initial pour être fonctionnels sur une machine linux.
- Rendez-vous dans la documentation pour plus d'informations.

Samba4 - Serveur de fichier (DM)

- Mapper l'utilisateur root en tant qu'Administrateur du domaine vous évitera une erreur (bien que non critique) lors de la jonction à l'AD, il est donc conseillé de le faire même si ce n'est pas obligatoire.
- Vous pouvez ensuite joindre le domaine :
 - `net ads join -U administrator`
- Si tout est ok, vous devrez éditer le `nsswitch.conf` pour y ajouter l'authentification via winbind.
- Puis lancer les différents services `smbd`, `nmbd` et `winbindd`.
 - Attention, si vous avez installé Samba par les sources, les scripts d'init ne seront pas forcément dans `/etc/init.d/`. Ce sera à vous d'ajouter les scripts init, de les activer au démarrage, et de les faire pointer correctement sur les binaires présents dans le dossier d'installation par les sources (`/usr/local/samba/...`)
- Vous pourrez ensuite vérifier la connexion au domaine, si cela fonctionne, vous êtes prêt à configurer le serveur de fichier.

```
root@sweb01:~# wbinfo --ping-dc
checking the NETLOGON for domain[ABALONE] dc connection to "SRV-ADDC01.abalone.fr" succeeded
root@sweb01:~# █
```

Samba4 - Serveur de fichier (DM)

- Vous pourrez voir directement depuis votre machine, les utilisateurs et groupe de votre AD, et définir des permissions avec ceux-ci.

```
root@sweb01:~# getent passwd | tail -20
tcpdump:x:108:113::/nonexistent:/usr/sbin/nologin
landscape:x:109:115::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:110:1::/var/cache/pollinate:/bin/false
sshd:x:111:65534::/run/sshd:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
vlaine:x:1000:1000:vlaine:/home/vlaine:/bin/bash
lxd:x:998:100::/var/snap/lxd/common/lxd:/bin/false
gluster:x:112:119::/var/lib/glusterd:/usr/sbin/nologin
ntp:x:113:120::/nonexistent:/usr/sbin/nologin
administrator*:10500:10513::/home/administrator:/bin/bash
guest*:10501:10513::/home/guest:/bin/bash
defaultaccount*:10503:10513::/home/defaultaccount:/bin/bash
krbtgt*:10502:10513::/home/krbtgt:/bin/bash
vlaine*:11105:10513::/home/vlaine:/bin/bash
jpierre*:11107:10513:jean pierre:/home/jpierre:/bin/bash
unixadmin*:11109:10513::/home/unixadmin:/bin/bash
abernard*:11110:10513::/home/abernard:/bin/bash
ybayer*:11111:10513::/home/ybayer:/bin/bash
```

N:B: les uid/gid
supérieur à 10000
sont des éléments de
notre annuaire.

```
root@sweb01:~# chown root:"Domain Admins" /srv/samba/users/
root@sweb01:~# l /srv/samba/
total 32K
659031 4.0K drwxr-xr-x  5 root root      4.0K Apr 25 20:17 .
524289 4.0K drwxr-xr-x  3 root root      4.0K Apr 25 18:37 ..
659041 8.0K drwxrwxr-x+  3 root domain users 4.0K Apr 25 20:29 profiles
659032 8.0K drwxrwxr-x+  3 root domain admins 4.0K Apr 25 18:25 public
659036 8.0K drwxr-s---+  6 root domain admins 4.0K Apr 25 19:59 users
root@sweb01:~#
```


Samba4 - Serveur de fichier (DM)

- Pour terminer, une fois la jonction au domaine fonctionnel, vous pouvez maintenant configurer Samba en tant que serveur de fichier.
 - https://wiki.samba.org/index.php/Samba_File_Serving
- A savoir lors de la mise en œuvre :
- Vous pouvez utiliser 2 types d'ACL pour gérer vos fichiers, soit les ACL Windows soit les ACL POSIX.
- Samba prend en charge les partages avec des listes de contrôle d'accès (ACL) POSIX sur les membres du domaine Unix, ils vous permettent de gérer les permissions localement sur l'hôte Samba en utilisant des utilitaires UNIX.
 - Il est conseillé par Samba d'utiliser les ACL de Windows, qui vous permettront de définir des ACL à granularité fine.
 - Si vous définissez des permissions de partage avec des ACL POSIX, alors vous ne devez pas utiliser les ACL Windows, ne définissez jamais les permissions à partir de Windows.
- Samba prend en charge les partages avec des ACL POSIX sur :
 - les membres du domaine
 - PDC et BDC NT4
 - Les hôtes autonomes
- Sur un contrôleur de domaine (DC) Samba Active Directory (AD), la prise en charge des ACL Windows est activée globalement, et donc les partages avec des ACL POSIX ne sont pas pris en charge. Vous devez utiliser les ACL Windows.
- Les listes de contrôle d'accès Windows (ACL) étendues vous permettent de définir des permissions sur des partages, des fichiers et des répertoires en utilisant des ACL et des applications Windows. Samba prend en charge les partages utilisant des ACL étendues sur :
 - les membres du domaine
 - Contrôleurs de domaine (DC) Active Directory (AD)
 - Contrôleur de domaine primaire NT4 (PDC)
 - Contrôleurs de domaine de secours NT4 (BDC)
 - Hôtes autonomes

Samba4 - Serveur de fichier (DM)

- Avec Samba en tant que serveur de fichier vous pourrez :
- Créer un partage avec des droits spécifiques
 - https://wiki.samba.org/index.php/Setting_up_a_Share_Using_Windows_ACLs
- Créer les répertoires personnels de vos utilisateurs
 - https://wiki.samba.org/index.php/Windows_User_Home_Folders
- Créer les profils inhérent de vos utilisateurs
 - https://wiki.samba.org/index.php/Roaming_Windows_User_Profiles
- Mettre en place un serveur d'impression
 - https://wiki.samba.org/index.php/Setting_up_Samba_as_a_Print_Server

Samba4 - Le Serveur de fichier (DM)

- Exemple d'un fichier de configuration Samba (en tant que serveur de fichier, lié à un ADDC Windows).
 - Ici nous utilisons le backend « rid » et les WINDOWS ACL.
- 2 partages sont déclarés, un public et celui qui servira aux dossiers utilisateurs, le reste des permissions se font se fait sur Windows.

```
[global]
workgroup = ABALONE
security = ADS
realm = ABALONE.FR

winbind refresh tickets = Yes
vfs objects = acl_xattr
map acl inherit = Yes
store dos attributes = Yes

username map = /usr/local/samba/etc/user.map
dedicated keytab file = /etc/krb5.keytab
kerberos method = secrets and keytab

winbind enum users = yes
winbind enum groups = yes
winbind use default domain = yes
log file = /var/log/samba/%m.log
log level = 1

# Default ID mapping configuration for local BUILTIN accounts
# and groups on a domain member. The default (*) domain:
# - must not overlap with any domain ID mapping configuration!
# - must use a read-write-enabled back end, such as tdb.
idmap config * : backend = tdb
idmap config * : range = 3000-7999
# - You must set a DOMAIN backend configuration
# idmap config for the ABALONE domain
idmap config ABALONE : backend = rid
idmap config ABALONE : range = 10000-999999
```

```
template shell = /bin/bash
template homedir = /home/%U

vfs objects = acl_xattr
map acl inherit = yes
# the next line is only required on Samba versions less than 4.9.0
store dos attributes = yes

[public]
    path = /srv/samba/public/
    read only = no
    acl_xattr:ignore system acl = yes

[users]
    path = /srv/samba/users/
    read only = no
    acl_xattr:ignore system acl = yes

[profiles]
    path = /srv/samba/profiles/
    read only = no
    acl_xattr:ignore system acl = yes
```

Samba4 - Le Serveur de fichier (DM)

- Il est expliqué dans la documentation comment faire un montage automatique du répertoire sur un membre du domaine Windows.
- Pour un montage automatique sur une machine Linux;
 - Vous pouvez créer un script qui montera automatiquement le partage sur la machine en fonction de l'utilisateur qui s'y connecte.
 - Monter un partage depuis un client
 - https://wiki.samba.org/index.php/Mounting_samba_shares_from_a_unix_client
 - Monter un partage automatique avec le module pam_mount
 - http://manpages.ubuntu.com/manpages/trusty/man5/pam_mount.conf.5.html
 - apt install libpam-mount
 - Ajouter à /etc/sssd/sssd.conf
 - override_homedir = /home/%u@%d
 - override_shell = /bin/bash
 - use_fully_qualified_names = False (cette option est adaptée pour certains cas, ici le False est nécessaire pour que les points de montage se montent correctement. En effet, nous avons créé les dossier partagé sous la forme "uname", par exemple "vlaine". Or libpam-mount essaie de monter [users/vlaine@abalone.fr](https://users.vlaine@abalone.fr) ce qui ne nous arrange pas, donc on met False.
 - Puis éditer la pam session (/etc/pam.d/common-session)
 - session required pam_mkhomedir.so skel=/etc/skel/ umask=0022
 - vérifier aussi session optional pam_mount.so
 - et auth optional pam_mount.so

Samba4 - Le Serveur de fichier (DM)

- Puis éditer le fichier /etc/security/pam_mount.conf.xml (fichier de configuration de pam_mount), pour le faire correspondre avec votre infra.

```
<?xml version "1.0" encoding "utf-8" ?>
<!DOCTYPE SYSTEM "pam_mount.conf.xml.dtd">
<!--
    See pam_mount.conf(5) for a description.
-->

<pam_mount>

    <!-- debug should come before everything else,
    since this file is still processed in a single pass
    from top-to-bottom -->

<debug enable "1" />

    <!-- Volume definitions -->

<volume fsgrp "domain users@abalone.fr" fstype "cifs" server "sweb01.abalone.fr" path "users/%(USER)"
mountpoint "/home/%(USER)@abalone.fr/Documents" user "*" options "username=%(USER)@abalone.fr,nosuid,nodev,rw,auto,icharset=utf8" />

    <!-- pam_mount parameters: General tunables -->

<!-- Note that commenting out mntoptions will give you the defaults.
    You will need to explicitly initialize it with the empty string
    to reset the defaults to nothing. -->
<mntoptions allow "nosuid,nodev,loop,encryption,fsck,nonempty,allow_root,allow_other,rw,auto,icharset,username" />

<!-- requires ofl from hxttools to be present -->
<logout wait "0" hup "no" term "no" kill "no" />

<mntoptions require "rw" />

<!-- pam_mount parameters: Volume-related -->

<mkmountpoint enable "1" remove "true" />

</pam_mount>
```

Samba4 - Le Serveur d'impression

- Si vous configurez Samba en tant que serveur d'impression, les clients de votre réseau peuvent envoyer des travaux d'impression à l'hôte Samba en utilisant le protocole SMB (Server Message Block). Les exemples présentés dans cette documentation utilisent une imprimante brute dans le back-end. Cette configuration exige que le travail d'impression soit formaté par un pilote sur le client et puisse donc être traité par l'imprimante sans autre traitement ou filtrage.
- Activer le serveur d'impression et le spooler sous Samba
 - https://wiki.samba.org/index.php/Setting_up_Samba_as_a_Print_Server#Enabling_the_spoolssd_Service
- Samba supporte CUPS et LPRng comme serveur d'impression.
- `apt-get install cups`
- `systemctl enable cups`
- Config file : `/etc/cups/cupsd.conf`
 - `# Show shared printers on the local network.`
 - `Browsing On`
 - `# Only listen for connections from the local machine.`
 - `Port 631`
 - `Listen /run/cups/cups.sock`
 - `# Restrict access to the server...`
 - `<Location />`
 - `Order allow,deny`
 - `Allow @LOCAL`
 - `</Location>`
 - `# Restrict access to the admin pages...`
 - `<Location /admin>`
 - `AuthType Default`
 - `Require valid-user`
 - `Order allow,deny`
 - `Allow @LOCAL`
 - `</Location>`

Samba4 - Le Serveur d'impression

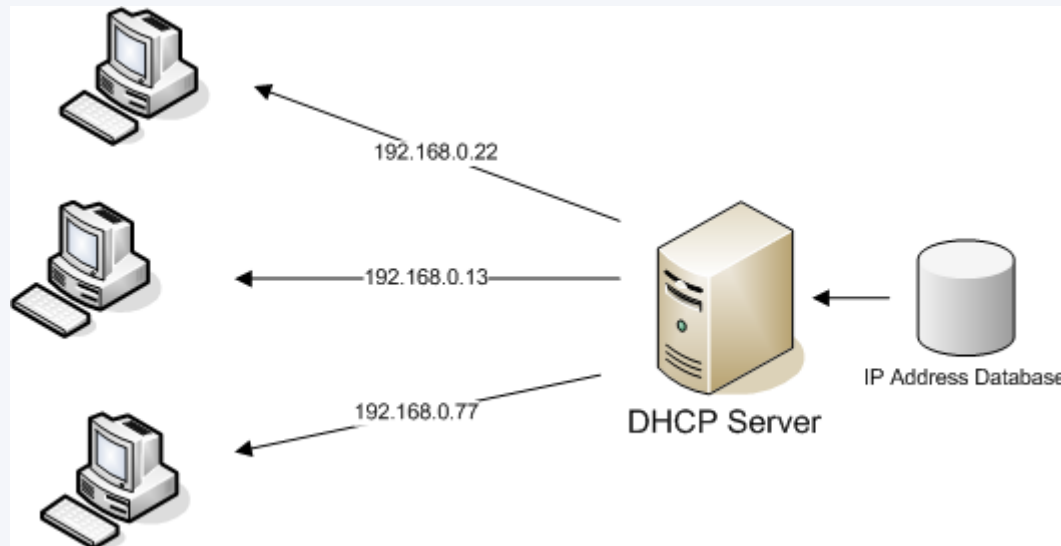
- Puis à vous d'ajouter votre imprimante depuis l'interface d'administration (login root..)
- Ouvrez l'interface web d'administration de CUPS dans votre navigateur.
`https://servername:631/admin`
 - Sélectionnez l'onglet Administration et cliquez sur Add Printers
- Sélectionnez le type de connexion et saisissez l'URL correspondant à la file d'attente de l'imprimante ou à la file d'attente du serveur d'impression distant. Par exemple :
 - Imprimantes basées sur LPD : `lpd://nom_imprimante/queue`
 - Imprimantes basées sur IPP (Internet Printing Protocol) : `ipp://printer_name/ipp/port`
 - Imprimantes basées sur SMB (Server Message Block) : `smb://nom d'utilisateur:mot de passe@domaine/windows_print_server_host_name/printer_name`
- Notez que le transfert d'une tâche vers un serveur d'impression fonctionnant sous Windows Vista ou plus récent, ou Windows Server 2008 ou plus récent, nécessite une authentification.
- Entrez un nom pour l'imprimante. Ce nom est utilisé dans le fichier `smb.conf` lors du partage de l'imprimante à l'aide de Samba.
- Sélectionner le fournisseur et le modèle de l'imprimante Raw.
- Enregistrez les paramètres.

Samba4 - Le Serveur d'impression

- Quelques documentations avec CUPS
- https://dev.tranquil.it/wiki/SAMBA_-_Samba4_et_CUPS
- https://tldp.org/HOWTO/Debian-and-Windows-Shared-Printing/sharing_with_windows.html
- https://wiki.samba.org/index.php/Setting_up_Samba_as_a_Print_Server
- https://wiki.samba.org/index.php/Setting_up_Automatic_Printer_Driver_Downloads_for_Windows_Clients

DHCP

- Qu'est ce qu'un DHCP ?
 - Dynamic Host Contrôle Protocol
- Le DHCP permet d'allouer dynamiquement des adresses IP aux clients d'un réseau
- A l'origine il était censé épauler BOOTP (Bootstrap protocol) pour l'installation d'une machine à travers le réseau



DHCP

- Fonctionnement d'un serveur DHCP
- Dans un réseau, il ne peut pas y avoir deux fois la même adresse IP
- Voici comment une machines récupère une adresse IP dynamiquement étape par étape :
 - **DHCPDISCOVER** from 00:0c:29:d7:40:46 via ens192
 - **DHCPOFFER** on 192.168.1.225 to 00:0c:29:d7:40:46 (client-linux01) via ens192
 - **DHCPREQUEST** for 192.168.1.225 (192.168.1.140) from 00:0c:29:d7:40:46 (client-linux01) via ens192
 - **DHCPACK** on 192.168.1.225 to 00:0c:29:d7:40:46 (client-linux01) via ens192

DHCP

- DHCPDISCOVER
- Au premier démarrage du client, DHCP cherche une adresse IP. Il initialise une version limitée du protocole TCP/IP puis diffuse un message (de type broadcast) DHCPDISCOVER qui correspond à une demande d'attribution IP reçue par tous les serveurs DHCP (l'adresse du destinataire est 255.255.255.255, ce qui concerne tous les nœuds du réseau).
 - Ce message diffusé contient le nom de l'hôte client (qui est également en général le nom NetBIOS de ce client) et l'adresse matérielle MAC (Media Access Control), celle qui est gravée une fois pour toutes dans la carte réseau de la machine.

DHCP

- DHCPOFFER
- Suite à cette requête, le serveur DHCP du sous-réseau mentionné répond par un message DHCPPOFFER qui contient une adresse IP candidate, un masque de sous-réseau et une durée d'attribution. Le message, qui contient en outre l'adresse IP du serveur DHCP émetteur, est diffusé à toutes les stations puisque le client IP demandeur n'a pas encore d'adresse IP pour être identifié.

DHCP

- DHCPREQUEST
- Dès que le client concerné reçoit le premier message DHCPPOFFER (il peut en recevoir plusieurs, mais il va s'arrêter au premier), il diffuse un message DHCPREQUEST à tous les serveurs DHCP du réseau pour dire qu'il accepte la proposition.
- Le message contient l'adresse IP octroyée par le serveur DHCP. Les autres serveurs du réseau annulent leur proposition et récupèrent l'adresse IP qu'ils avaient suggérée pour le prochain client qui en fait la demande.

DHCP

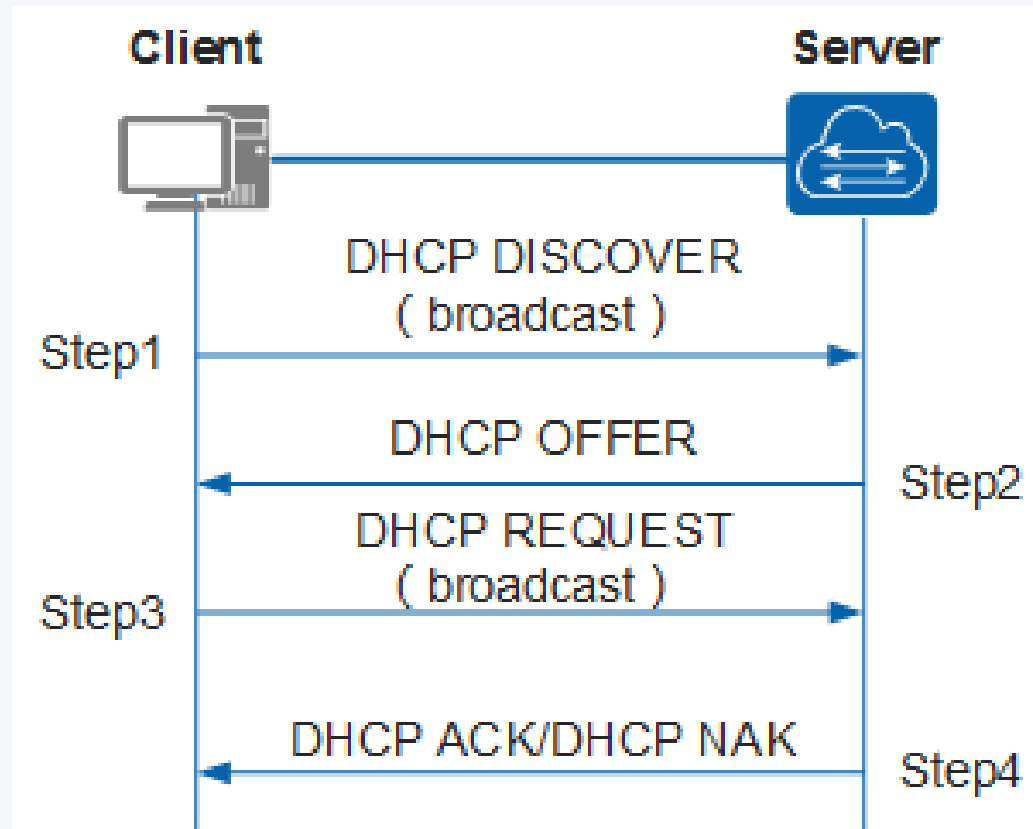
- DHCPACK/NOACK
- Pour finir, le serveur DHCP choisi envoie une confirmation au client sous la forme d'un message DHCPACK, qui contient l'adresse IP attribuée et d'autres informations de configuration TCP/IP. Le client mémorise toutes ces informations (sous Windows, elles sont stockées dans la base de registre).

DHCP

- Pour résumer l'échange de requête entre un client et le serveur DHCP :
- **DHCPDISCOVER** (pour localiser les serveurs DHCP disponibles)
- **DHCPOFFER** (réponse du serveur à un paquet DHCPDISCOVER, qui contient les premiers paramètres)
- **DHCPREQUEST** (requête diverse du client pour par exemple prolonger son bail)
- **DHCPACK** (réponse du serveur qui contient des paramètres et l'adresse IP du client)
- **DHCPNAK** (réponse du serveur pour signaler au client que son bail est échu ou si le client annonce une mauvaise configuration réseau)
- **DHCPDECLINE** (le client annonce au serveur que l'adresse est déjà utilisée)
- **DHCPRELEASE** (le client libère son adresse IP)
- **DHCPINFORM** (le client demande des paramètres locaux, il a déjà son adresse IP)

DHCP

- Et pour réellement simplifier, rien de mieux qu'un schéma !!



Le vocabulaire DHCP

- Etendue : Une étendue est la plage d'IP probable d'un réseau.
- Etendu globale : Rassemble différentes étendues, sur plusieurs sous réseau, contient d'autres étendues membres et enfants
- Plage d'exclusion : IP exclue de l'étendue
- Pool d'adresses : Une fois que vous avez défini une étendue DHCP et appliqué des plages d'exclusion, les adresses restantes forment le pool d'adresses disponible dans l'étendue.
- Réserveation : : Utilisez une réservation pour créer une affectation de bail d'adresse permanente par le serveur DHCP.
- Bail : Un bail est un intervalle de temps, spécifié par un serveur DHCP, pendant lequel un ordinateur client peut utiliser une adresse IP affectée

DHCP

- Le bail DHCP
- Pour des raisons d'optimisation des ressources réseau, les adresses IP sont délivrées avec une date de début et une date de fin de validité.
- Un client qui voit son bail arriver à terme peut demander au serveur une prolongation du bail par un DHCPREQUEST. De même, lorsque le serveur verra un bail arriver à terme, il émettra un paquet DHCPNAK.
- On peut optimiser l'attribution des adresses IP en jouant sur la durée des baux.
- Risques : aucune adresse n'est libérée au bout d'un certain temps, plus aucune requête DHCP ne pourra être satisfaite.
- Sur un réseau où beaucoup d'ordinateurs se branchent et se débranchent souvent il est intéressant de proposer des baux de courte durée.
- A l'inverse, sur un réseau constitué en majorité de machines fixes, très peu souvent rebootées, des baux de longues durées suffisent.

DHCP TP

- Mettre en place un serveur DHCP
- Le TP est fourni par l'examineur

Serveur DNS

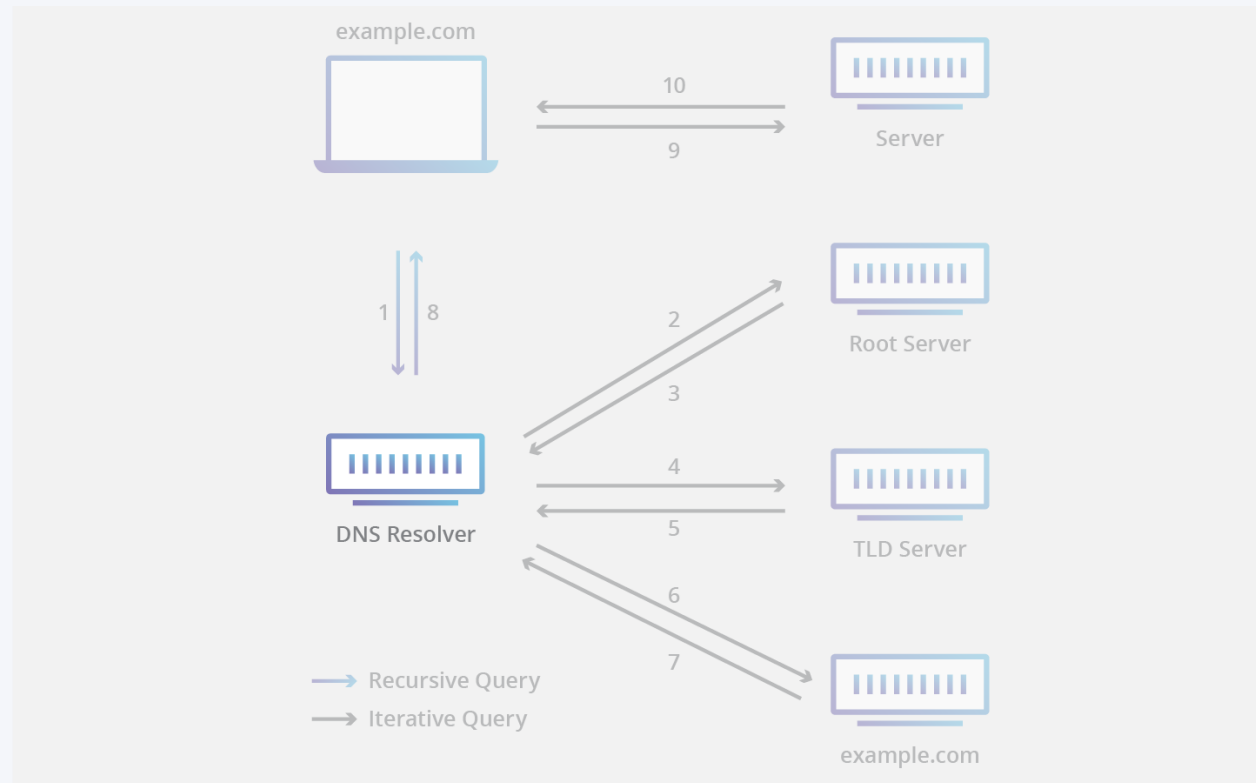
- Fonctionnement d'un serveur DNS
 - Qu'importe le l'OS (Windows ou Linux) le fonctionnement d'un serveur DNS reste le même
- Mais qu'est ce que c'est ?
 - DNS = Domain Name System
 - C'est l'organe le plus important des réseaux d'INTERNET
 - C'est un annuaire qui permet de traduire une adresse IP en nom et vice versa.
 - Les serveurs sont communément appelé serveur DNS ou serveurs de noms
- Il existe 13 DNS racines d'internet (allant de A à M) (+redondance)
- La NTIA délègue la gestion des zones racines à l'ICANN

Serveur DNS

- Il est impensable que les serveurs DNS racines connaissent toutes les machines se connectant à internet.
- Il existe donc une hiérarchie des serveurs DNS
- Concernant les pages WEB il existe 4 types de serveurs DNS
 - Les serveurs DNS récursifs (Resolving Name server)
 - Serveur racine du DNS (Root Name Server)
 - Serveur DNS TLD (Top Level Domain, ou Domaine de Premier Niveau)
 - Serveur de nom faisant autorité (Authoritative Name Server)

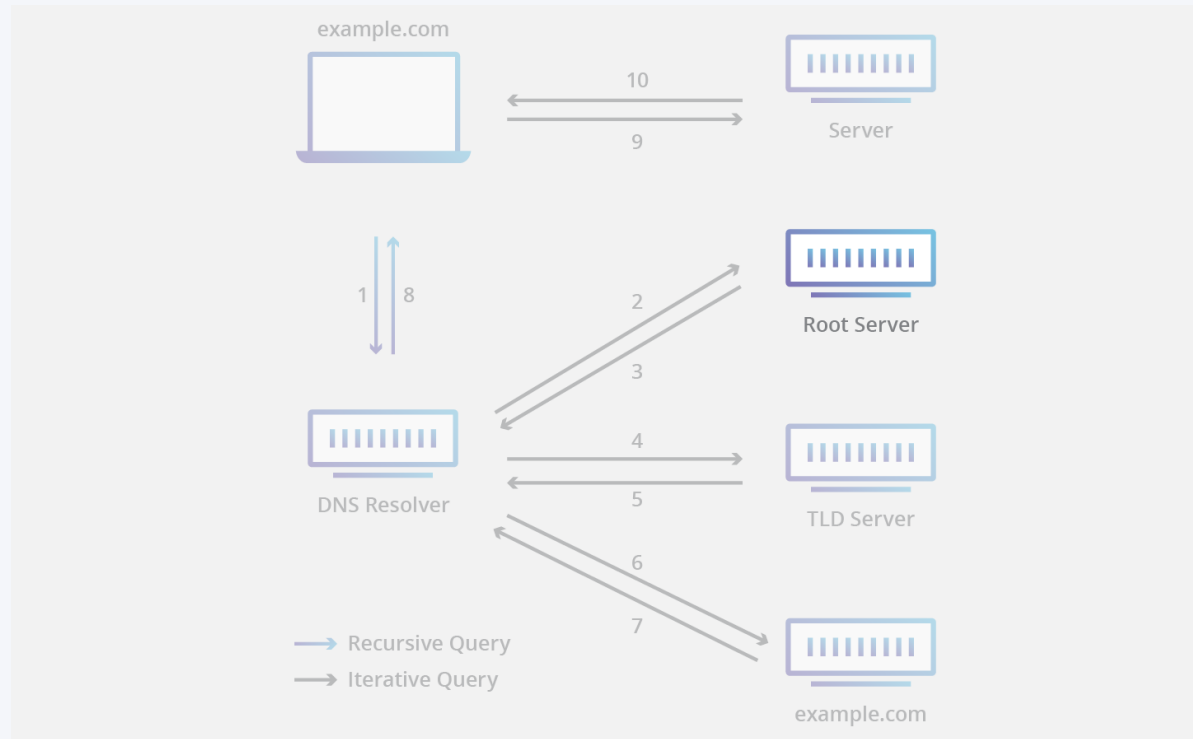
Serveur DNS

- Le serveur DNS récursif (Resolving Name Server) :
- Un résolveur récursif (ou récursueur DNS) est le premier arrêt d'une requête DNS. Il répondra à la requête en interrogeant d'autres serveurs ou aura déjà un enregistrement de l'adresse IP du site en cache.



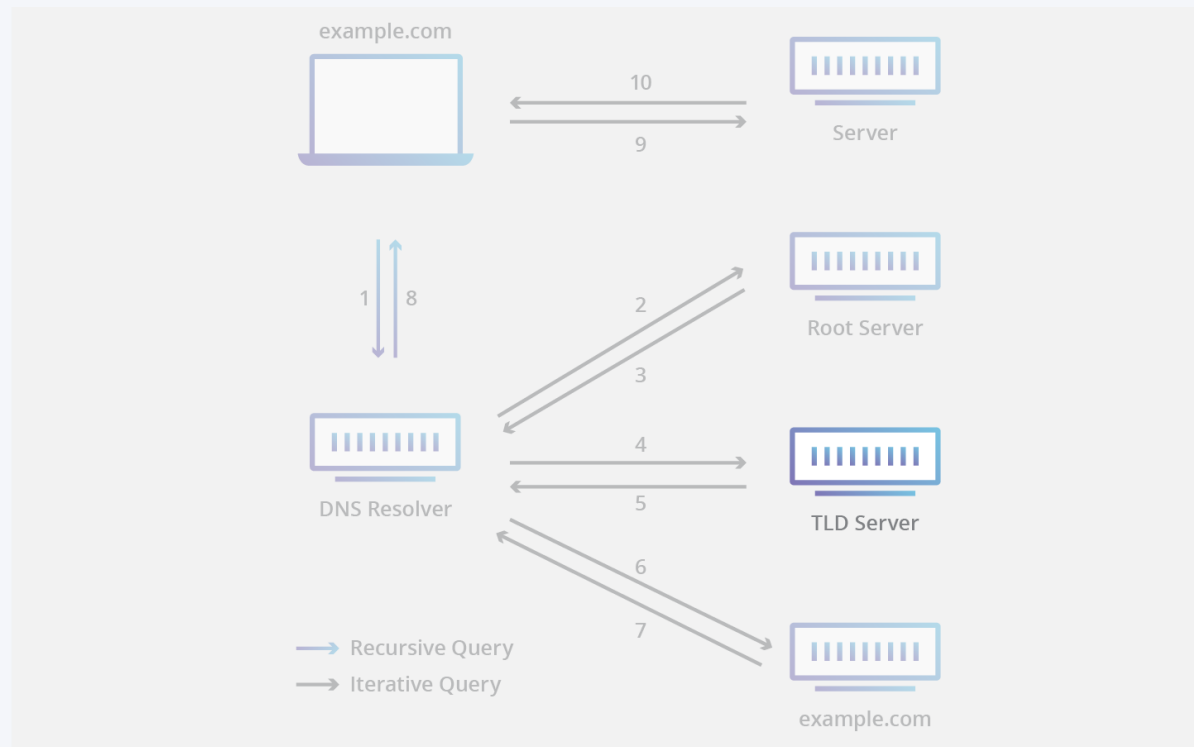
Serveur DNS

- **Serveur racine du DNS (Root Name Server) [ICANN]:** Il s'agit du serveur de noms pour la zone racine. Il répond à des requêtes directes et peut renvoyer une liste de noms de serveurs faisant autorité pour le domaine de haut niveau correspondant.
- Les 13 serveurs de noms racine DNS sont connus de tous les résolveurs récur­sifs.
- Interrogé en premier par les serveurs récur­sifs puis redirige ensuite celui-ci vers les TLD.



Serveur DNS

- **Serveur DNS TLD : le serveur TLD (top-level domain : domaine de premier niveau) :** Conserve les informations des noms de domaines ayant une extension commune (.net .com...) TLD.com contient des informations des sites qui se terminent par .com.
- Interrogé par le serveur récursif après que celui-ci ait interrogé un serveur de nom racine, qui répond en pointant vers le serveur faisant autorité.

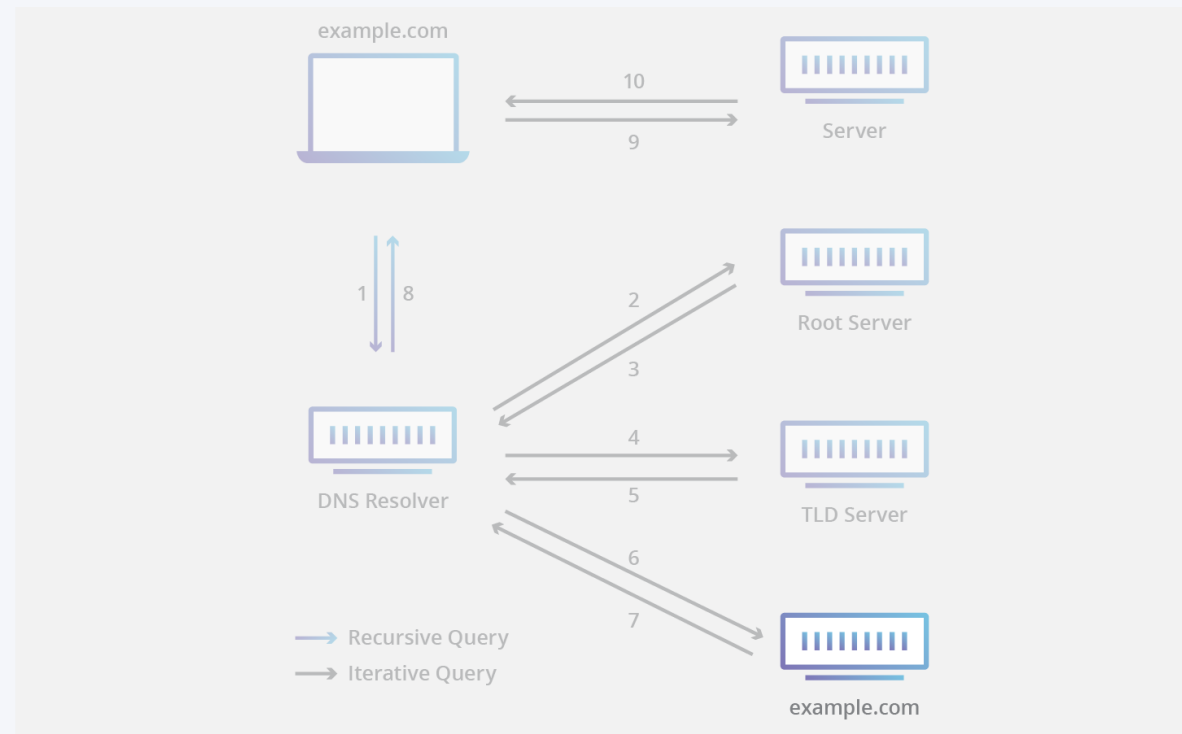


GESTION DES TLD

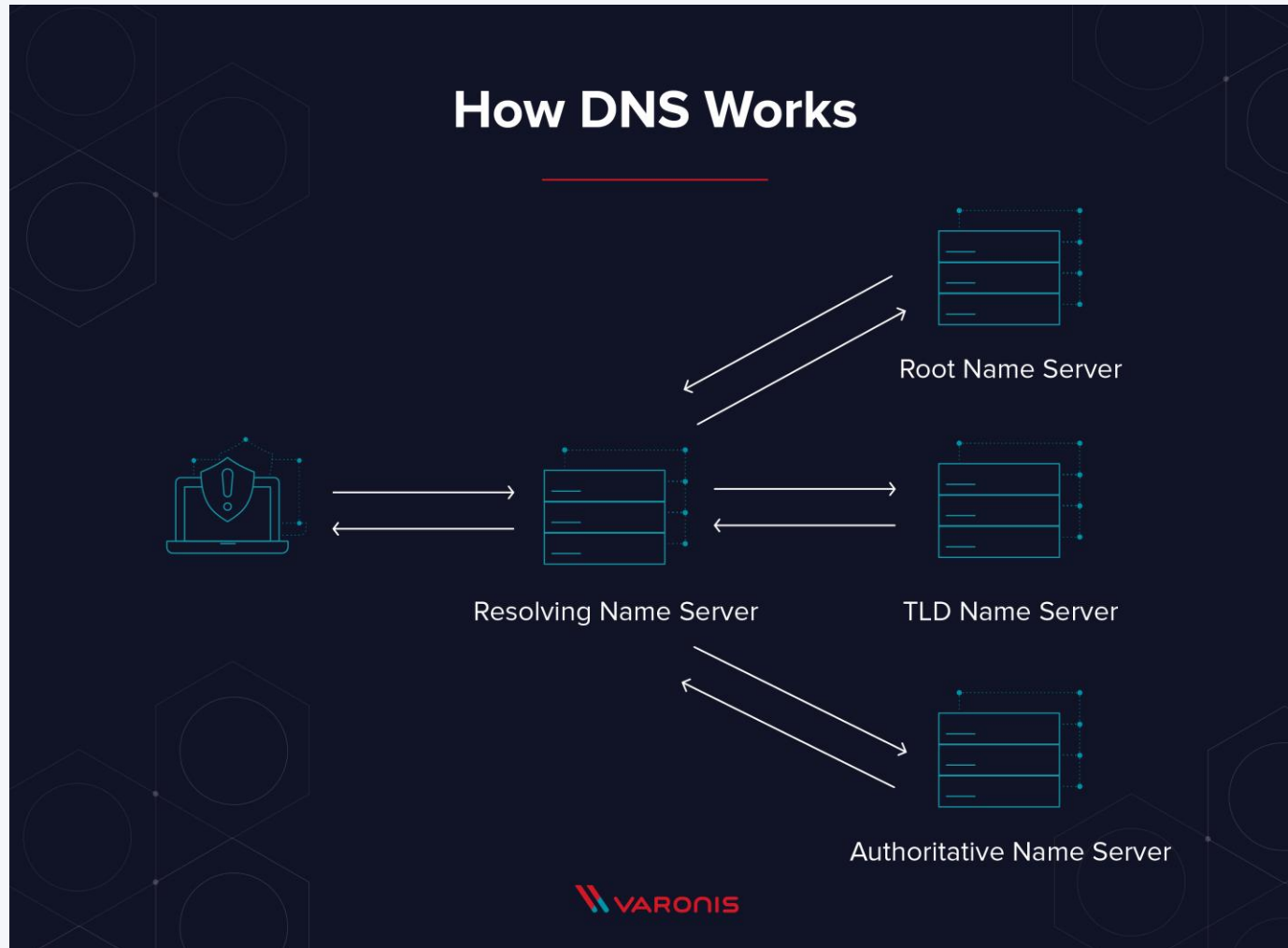
- La gestion des serveurs de noms TLD est assurée par l'IANA (Internet Assigned Numbers Authority), qui est un département de l'ICANN. L'IANA divise les serveurs TLD en deux groupes principaux :
- Domaines génériques de premier niveau : ce sont des domaines qui ne sont pas spécifiques à un pays, certains des TLD génériques les plus connus incluent .com, .org, .net, .edu et .gov.
- Domaines de premier niveau de code de pays : ces domaines incluent tous les domaines spécifiques à un pays ou à un état. Citons à titre d'exemple .uk, .us, .ru et .jp.
- Une troisième catégorie historique existe pour les .arpa, mais n'est presque plus utilisée car elle a surtout servi à la transition vers les DNS modernes.

Serveur DNS

- **Serveur de noms faisant autorité (Authoritative Name Server) :** Interrogé par le serveur récursif après que celui-ci ai récupéré une réponse du serveur TLD.
- le serveur de noms faisant autorité constitue le terminus d'une requête DNS. Le serveur de noms faisant autorité contient l'enregistrement DNS répondant à la requête (A, CNAME..)



Serveur DNS



Services DNS

- Types de services DNS
 - Il y a deux types de services DNS
- Résolveur DNS récursif
 - un résolveur DNS récursif est un serveur DNS qui répond à la requête et recherche le serveur de noms faisant autorité ou un cache de DNS contenant le résultat de la requête.
- Serveur DNS faisant autorité
 - un serveur DNS faisant autorité contient le résultat de la requête DNS. De ce fait, si vous demandez à un serveur DNS faisant autorité l'une de ses adresses IP, celui-ci n'a pas besoin d'interroger un autre serveur. Le serveur de noms faisant autorité est l'autorité finale en ce qui concerne les noms et adresses IP.

Services DNS

- Il existe des DNS privé et publique
- DNS public :
 - pour qu'un serveur soit accessible sur l'Internet public, il doit avoir un enregistrement DNS public, et son adresse IP doit être accessible sur Internet.
- DNS privé :
 - les ordinateurs qui sont derrière un pare-feu ou dans un réseau interne utilisent un enregistrement DNS privé qui permet aux ordinateurs locaux de les identifier par leur nom. Les utilisateurs extérieurs, sur Internet, ne pourront pas accéder directement à ces ordinateurs.

Les requêtes DNS

- Les étapes d'une recherche DNS
 - Une requête DNS débute lorsque vous essayez d'accéder à un ordinateur sur Internet. Par exemple, vous tapez `www.google.com` dans la barre d'adresse de votre navigateur.
 - La première étape de la requête DNS est le cache du DNS local. Lorsque vous accédez à différents ordinateurs, leur adresse IP est stockée dans un référentiel local. Si vous avez déjà visité `www.google.com`, l'adresse IP se trouve dans votre cache.
 - Si l'adresse IP ne se trouve pas dans le cache de votre DNS local, le DNS va consulter un serveur DNS récursif. Pour cela, votre équipe informatique ou votre fournisseur d'accès Internet (ISP) fournit en général un serveur DNS récursif.
 - Le serveur DNS récursif a son propre cache et, s'il contient l'adresse IP, il vous l'enverra en retour. Si ce n'est pas le cas, il en fera la demande au serveur DNS racine.
 - L'étape suivante est constituée des serveurs de noms TLD, dans ce cas le serveur de noms TLD pour les adresses en `.com`. Ces serveurs ne contiennent pas l'adresse IP dont nous avons besoin, mais ils peuvent envoyer la requête DNS dans la bonne direction.

Les requêtes DNS

- Les serveurs de noms TLD connaissent l'emplacement du serveur de noms ayant autorité pour le site recherché.
- Le serveur de noms ayant autorité répond avec l'adresse IP de `www.google.com` et le serveur DNS récursif la stocke dans le cache DNS local, puis l'envoie en retour vers votre ordinateur.
- Votre service DNS local obtient l'adresse IP et se connecte à `www.google.com` pour télécharger tout son (merveilleux) contenu. Le DNS enregistre ensuite l'adresse IP dans le cache local, en lui associant une durée de vie (TTL pour time-to-live). Le TTL est la durée de validité de l'enregistrement DNS local. Lorsque ce délai sera expiré, et la prochaine fois que vous voudrez accéder à `varonis.com`, le DNS suivra à nouveau le parcours que nous venons de décrire.

Les requêtes DNS

- Les différents types de requêtes DNS
- Requête récursive
 - dans une requête récursive, l'ordinateur demande une adresse IP ou la confirmation que le serveur DNS ne connaît pas cette adresse IP.
- Requête itérative
 - avec une requête itérative, on demande au serveur DNS la meilleure réponse en sa possession. Si le serveur DNS n'a pas l'adresse IP, il renverra le serveur de noms ayant autorité ou le serveur de noms TLD. Le demandeur poursuivra son processus itératif jusqu'à ce qu'il obtienne une réponse ou que le délai imparti expire.
- Requête non récursive
 - Un résolveur DNS utilisera cette requête pour trouver une adresse IP qu'il ne détient pas dans son propre cache. Pour limiter l'utilisation de la bande passante réseau, celles-ci sont limitées à une seule requête.

Les requêtes DNS

- Le cache DNS et la mise en cache du DNS permettent de :
 - Accélérer les requêtes DNS
 - Réduire la bande passante des requêtes DNS sur Internet
- Cependant la mise en cache peut être parfois problématique
 - Les modifications de DNS nécessitent du temps pour se propager, ce qui fait qu'il peut s'écouler un certain temps avant que le cache de l'ensemble des serveurs DNS soit mis à jour avec les données IP les plus récentes.
 - Le cache DNS est un vecteur d'attaque possible pour les pirates

Cache

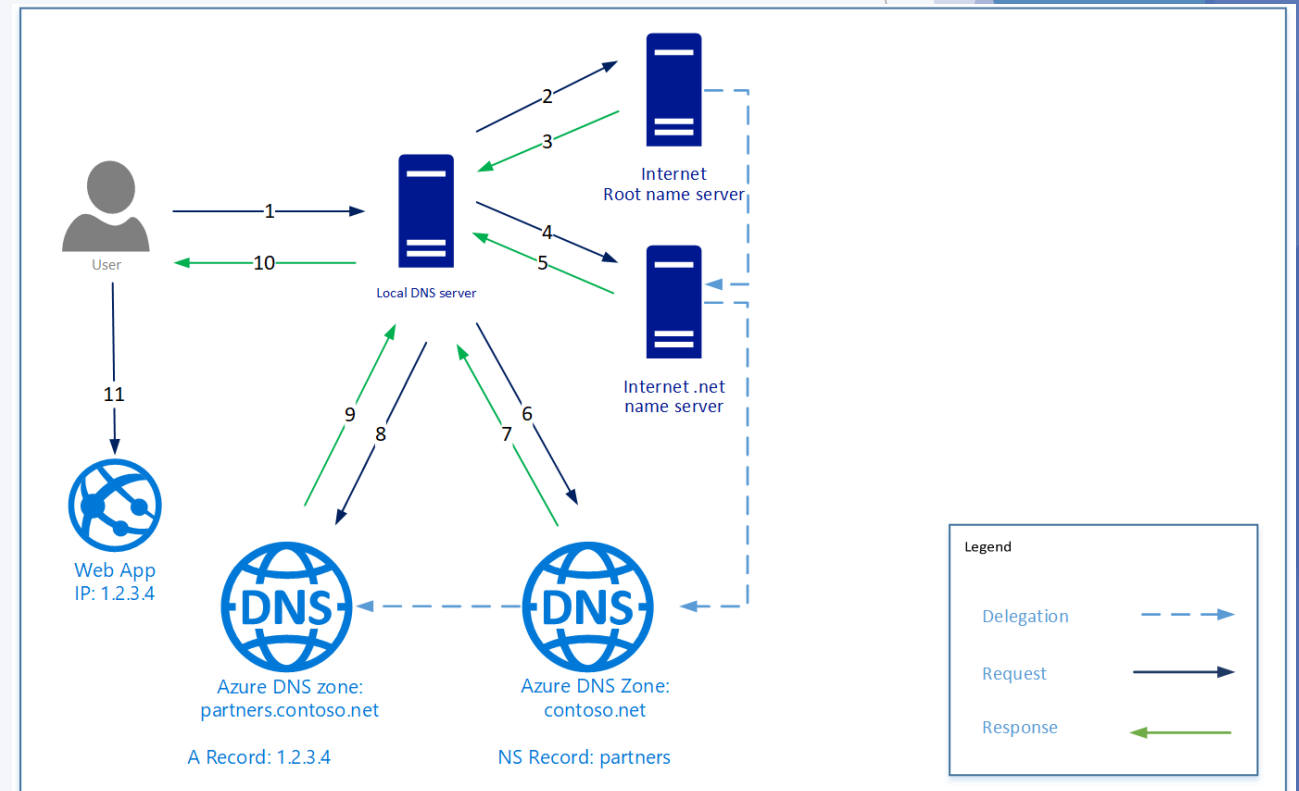
- Les différents types de mises en caches
- **La mise en cache DNS du navigateur :**
 - les navigateurs modernes (2018) intègrent une fonctionnalité de mise en cache. La résolution d'une requête DNS avec le cache local est rapide et efficace.
- **La mise en cache DNS du système d'exploitation :**
 - votre ordinateur est un client DNS, et il contient un service qui gère les requêtes et la résolution DNS. Ce cache DNS est également local : il est donc rapide et n'exige aucune bande passante.
- **La mise en cache DNS par résolution récursive :**
 - chaque DNS récursif a son propre cache DNS, et il y stocke toutes les adresses IP dont il a connaissance en vue de traiter les prochaines requêtes.

Zone DNS

- **Qu'est-ce qu'une Zone DNS?**
 - Une zone DNS se réfère à une certaine partie ou à l'espace administratif dans le Domain Name System mondiale (DNS).
 - Toutes les entrées dans la zone d'un domaine, tels que les sous-domaines (sub.domaine.com) ou un enregistrement MX (qui spécifie l'emplacement du serveur de courriels) sont dans cette zone.
 - Où est-ce situé? Cela dépend des DNS que vous utilisez.
- **Les différents champs d'une zone DNS**
 - **Domaine ou sous-domaine**
 - **TTL (Time to Live):** le temps de rafraichissement en seconde de l'entrée
 - **Classe:** "IN" veut dire "Internet"
 - **Type d'enregistrement:** A, MX, TXT ou CNAME
 - **Valeur:** nom web, IP ou une autre valeur pertinente

Record DNS

- Les différents records (enregistrements) dans un serveurs DNS
 - **A**: l'entrée réfère à une IPv4
 - **AAAA** : L'entrée réfère à une Ipv6
 - **CNAME**: l'entrée réfère un nom web
 - **MX**: nom web du serveur de courriels
 - **TXT**: Entrée-machine lisible utilisées souvent pour les enregistrements SPF ou DKIM
 - **NS**: Nom web du nom de serveur où le domaine est situé



Configuration zone DNS

- Configuration de bases d'un serveur DNS
 - Sous linux, il faut installer les paquets Bind9n et les net-tools
- Pour configurer une zone DNS principale il faut modifier le fichier named.conf

```
zone "mondomaine.org" {  
    type master;  
    file "mondomaine.org.zone";  
};
```

```
zone "domaine-ami.org" {  
    type slave;  
    file "domaine-ami.org.zone";  
    masters { 12.42.112.242; };  
};
```

Serveur DNS Bind9

TP

1. Monter un résolveur local (DNS) avec Bind9
2. Monter un DNS Dynamique
 1. Quand le DHCP distribue une adresse IP, elle s'ajoutera automatiquement dans le DNS.
3. Mettre en place DDNS pour sécuriser nos échanges

Le TP est fourni par l'examineur.