Notes Linux

1. Working with Shell (common commands)

- ***pwd*** (present working Directory)
- ***type <command>*** (permet de vérifier le type de la commande : built-in-shell ; interne ; externe ou fichier binaire)
- ***ls*** (List contents)
- ***mkdir -p*** (créer les répertoires parent s'ils n'existent pas)
- ***pushd*** (permet de se déplacer vers un répertoire tout en sauvegardant la position précédente)
- ***popd*** (permet de revenir à la position précédente sauvegardée par la dernière commande pushd)
- ***more <filename.txt>*** (affiche le contenu détaillé d'un fichier à l'écran)
- ***less <filename.txt>*** (affiche les 10 dernières lignes d'un fichier)
- ***whatis <command>*** (affiche une description d'une ligne de ladite commande)
- ***man <command>*** (affiche le manuel pour ladite command)
- ***apropos <mot-clé>*** (affiche la liste de toutes les commandes du système contenant ce mot clé)
- ***echo $SHELL*** (affiche le shell utilisé)
- ***chsh*** (changer le shell par défaut utilisé on distingue le sh, bash, ksh, zsh, csh, tcsh) ; le bash est plus utilisé que le sh parce qu'il permet l'auto-complétion de façon native
- Pour rendre une variable d'environnement persistante, il faut la rajouter dans le répertoire ***~/.profile ou ~/.pam_environment***
- ***Which <command>*** (permet de vérifier la source d'une commande d'où est ce qu'elle provient)
- ***Export PATH=$PATH:/opt/app/bin*** (permet d'ajouter un nouveau répertoire dans la variable d'environnement PATH su système : les fichiers se trouvant dans les répertoires du PATH peuvent être appelés comme commande n'importe où dans le système.
- ***Bash promt et $PS1*** (PS1 est la variable d'environnement dans laquelle se trouve les paramètres du bash promt exemple [\W]$: \W pour present working directory et $ user symbol promt
- Il est possible configurer le bash promt au travers de sa variable d'environnement en utilisant certains caractères spéciaux suivants :
    - \d : la date
    - \h : le nom réduit de l'hôte (HQDN)
    - \H : le nom complet de l'hôte
    - \s : le nom du Shell utilisé
    - \t : le temps actuel au format 24 heures
    - \T le temps actuel au format 12 heures
    - \u : le nom d'utilisateur
    - \w : le répertoire courant complet préfixé du $HOME (~)
    - \W : le nom de base du répertoire courant préfixé du $HOME

2. Linux cores concepts

2.1. Le Kernel Linux

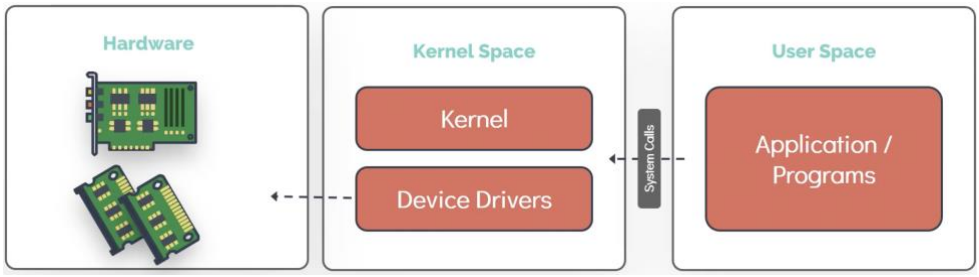
Linux Kernel c'est le gestionnaire ou l'intermédiaire entre le software et le hardware dans un système linux.

Le Kernel linux est responsables des 04 taches majeures suivantes :

- ***Memory management*** : que ce soit la mémoire vive (RAM) ou mémoire dure (ROM)
- ***Process Management*** : détermine quel processus peut utiliser le CPU, Quand et pour combien de temps
- ***Device drivers*** : il agit comme un interpréter (plugins) entre le hardware et le software
- ***Systems calls and security :*** il reçoit et traite les requêtes provenant des services ou processus

Le kernel sépare la mémoire (memory management) en deux espaces :

- ***Kernel space*** : espace mémoire dédié au processus systèmes du kernel et également aux drivers des périphériques (E/S). Un processus kernel a un accès illimité au hardware
- ***User Space*** : espace mémoire dédié aux processus ou applications utilisateurs ayant un accès plus ou moins restreint à la mémoire et au CPU. Pour accéder aux ressources mémoire ou CPU, un programme utilisateur doit réaliser une requête (system calls) au Kernel afin que ce dernier lui alloue les ressources nécessaires

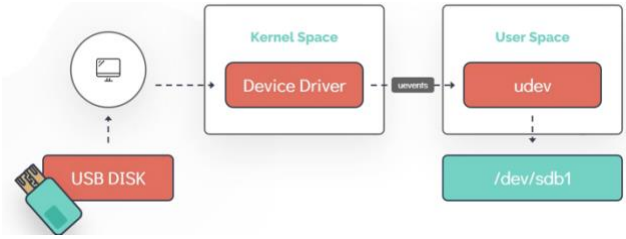


2.1.1. Les commandes

- ***uname*** : (affiche le nom du kernel)
- ***uname -r*** : (affiche la version du kernel)

2.2. Gestion du Hardware dans Linux

Lorsqu'un nouvel disque comme une clé USB est ajouté à la machine linux, il est détecté par la fonction « device drivers » du kernel ; cette fonction fait le nécessaire concernant les plugins et envoi un évenement (uevents) au service de gestion des disques du user space (udev) qui est responsable de créer de façon dynamique les liens permettant d'attacher le nouveau disque au répertoire /dev



2.2.1. Les commandes

- ***dmesg*** : affiche à l'écran l'ensemble des messages générés par le kernel (msg d'erreurs ou des logs du kernel) ; peut être utilisé avec less ou grep pour faire une recherche filtrée
- ***udevadm***: outil de management du service udev (user devices)
- ***udevadm info --querry=path --name=/dev/sda5*** : permet d'avoir les infos liées au path d'un device monté dans le /dev/sda5
- ***udevadm monitor*** : affiche l'ensembles logs du service udev et également l'ensemble des uvents (messages provenant du kernel pour le udev concernant la gestion des disques).
- ***lspci*** : affiche l'ensemble des informations sur les cartes pci (carte Ethernet, carte graphique …) connectées à la carte mère de la machine.
- ***lsblk*** : (list bloc device) affiche l'ensemble des disques montés sur la machine
- ***lscpu*** : affiche les informations sur l'architecture du cpu, le nombre de cœurs, le model … bref un ensemble d'informations sur le processeur
- ***lsmem ou lsmem –summary*** : affiche le résumé sur la mémoire disponible du système
- ***free -m*** : affiche la quantité de mémoire utilisée et restantes du système (le -m pour afficher le résultat en méga ; le -k en kilo et le -g en giga)
- ***lshw*** : affiche les détails sur l'ensemble du hardware configuré sur la machine

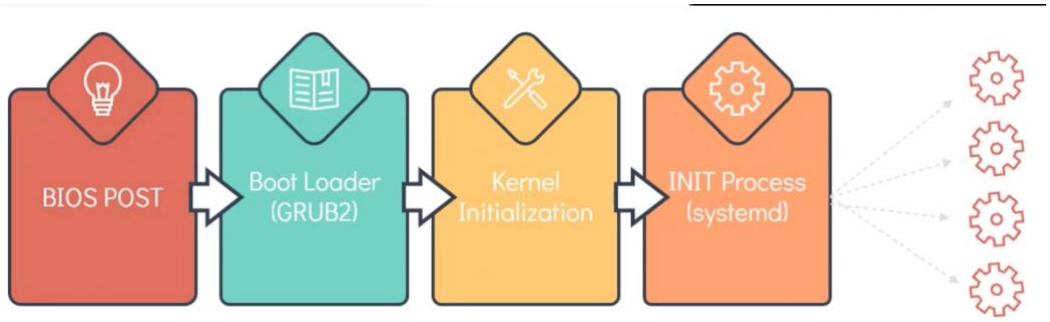
NB : ces différentes commandes nécessitent généralement une élévation de privilèges (sudo)

2.3. Linux Boot Sequence

La procédure de démarrage ou de boot de linux peut être décomposée en 04 grandes parties :

- ***BIOS POST*** : dans cette étape, le bios lance une série de tests afin de s'assurer que le hardware est bien attaché aux devices et fonctionnent correctement
- ***BOOT LOADER*** : après avoir fait les premières vérifications, le bios lance la séquence de boot définie dans le boot loader (hardware initialisation et mise en place des fonctions kernel)
- ***KERNEL Initialisation*** : ici on initialise le kernel et enfin
- ***INIT PROCESS*** : On démarre les services à l'aide du Systemd ou sysfile (pour connaitre le gestionnaire de services du système, on exécute ***ls -l /sbin/init*** et le résultat pointera vers le gestionnaire utilisé

```
[~]$ ls -l /sbin/init
lrwxrwxrwx /sbin/init -> /lib/systemd/systemd
```



2.4. Runlevels ou systemd targets

Il s'agit ici du niveau caractérisant le mode de fonctionnement de linux.

En effet linux peut être lancé dans plusieurs modes différents (Ex. graphique ou non).

Pour voir le mode dans lequel est exécuté un système linux, on tape la commande « ***runlevel*** ». Cette commande retourne un nombre qui représente le mode dans lequel est exécuté le système linux (N=5 correspond au mode graphique tandis que N=3 au mode non graphique)

Runlevel	Systemd Targets	Function
5	graphical.target	Boots into a Graphical Interface
3	multiuser.target	Boots into a Command Line Interface

Pour afficher le systemd target lié au mode dans lequel fonctionne un système, on utilise les commandes suivantes :

- ***systemctl get-default*** : affiche la valeur du systemd target
- ***ls -ltr /etc/systemd/system/default.target*** : affiche également le target utilisé
- ***systemctl set-default multi-user.target*** : permet de changer le systemd target et ainsi le mode d'exécution du système.

NB : le terme runlevel précédemment utilisé dans les sysV init systems a été remplacé par systemd target dans les systèmes utilisant systemd

2.5. Files types in Linux

Tout dans linux est un fichier ; il existe 03 types de fichiers dans Linux

- ***regular file*** : fichiers standards ou communément utilisés, pouvant contenir du texte, des scripts, des images, fichier de configuration ou autre
- ***Directory*** : type de fichier dans linux pouvant contenir d'autres fichiers, ils sont appelés répertoires
- ***Specials files*** : les fichiers spéciaux, ils peuvent être à leur tour repartis en 05 catégories

- ***Fichiers caractères*** : il s'agit ici des fichiers spéciaux représentant les périphériques de la machine se trouvant dans le répertoire /dev, par exemple la souris, le clavier …
- ***Les block files*** : représentent les devices de type bloc comme les disques ou la ram également situé dans le /dev ex. /dev/sda
- ***Links files***: hard links ou soft link (représentent les liens)
- ***Les sockets files :*** fichiers spéciaux qui permettent ou activent la communication entre deux processus
- ***Piped files :*** fichiers spéciaux permettant la connexion d'un processus à un autre

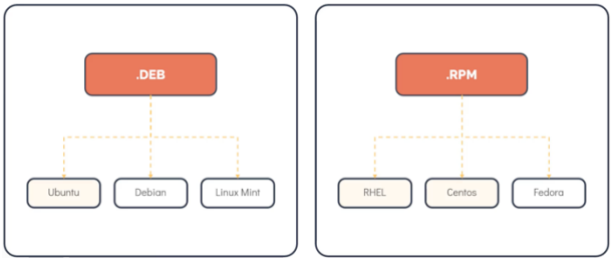
File Type	Identifier
DIRECTORY	d
REGULAR FILE	-
CHARACTER DEVICE	c
LINK	l
SOCKET FILE	s
PIPE	p
BLOCK DEVICE	b

2.5.1. Commandes

- ***file <arborescence>*** : permet de connaitre le type de fichier
- ***ls -ld <arborescence> :*** affiche le détail d'un fichier ou des fichiers d'un répertoire (le premier caractère de la ligne permet d'identifier le type de fichier)

NB: IDE (Integrated Development Environent)

3. Package management



Il existe à ce jour plus de 100 distributions Linux en cours d'utilisation ; un des éléments permettant de les catégoriser est le gestionnaire de paquets utilisé
En effet, Un paquet est un archive compressée contenant tous les fichiers nécessaires à l'installation d'une application particulière.

Un gestionnaire de paquet dans linux est une application qui fournit des processus complets et automatisés permettant d'installer, configurer, mettre à jour et supprimer les paquets dans le système.
Les fonctions d'un gestionnaire de paquets sont les suivantes :

- Vérifier l'intégrité et l'authenticité des paquets
- Simplifier la gestion des paquets
- Regrouper les paquets en catégories facilitant ainsi leur gestion
- Gérer les dépendances

Dépendant de la distribution Linux utilisée, il existe plusieurs types de gestionnaire de paquet :

- ***DPKG*** : gestionnaire de paquets de base pour les distributions Debian
- ***APT*** : le new front-end du gestionnaire DPKG également pour les distributions Debian
- ***APT-GET*** : front-end traditionnel pour DPKG
- ***RPM*** : gestionnaire de paquets de base pour les distributions RedHat, CentOS et Fedora
- ***YUM*** : front-end pour le gestionnaire RPM
- ***DNF*** : front-end plus riche pour le gestionnaire RPM

3.1. RPM and YUM

L'extension des paquets ou fichiers gérés par RPM ont l'extension .rpm
Nous avons vu plus haut que le gestionnaire RPM permet d'installer, désinstaller, mettre à jour, faire des requêtes et vérifier les paquets

3.1.1. Les commandes RPM

- ***rpm -ivh <telnet.rpm>*** : permet d'installer un paquet (i pour install, v pour la verbosité et h pour l'aide sur la commande)
- ***rpm -e <package_name>*** : désinstaller un paquet
- ***rpm -Uvh <package_name>*** : permet de mettre à jour un paquet avec (U majuscule)
- ***rpm -q <package_name>***: permet de faire une requête sur un paquet installés. Les paquets sont sauvegardés dans le répertoire /var/lib/rpm
- ***rpm -Vf <package_name>***: permet de vérifier un paquet en comparant les fichiers du paquet installés à ceux du fichier original du paquet (path du paquet original)

NB : Il est également important de noter que rpm ne gère pas l'installation des dépendances nécessaires au fonctionnement du paquet ; c'est la raison pour laquelle on a généralement besoin de YUM ou DNF

3.2. YUM package Management

Yum est un sigle qui signifie « ***Yellowdog Updater, Modified*** » ; c'est le gestionnaire de paquet amélioré de RPM. Yum fonctionne à l'aide des repos applicatifs qui peuvent être assimilés à une collection centralisée de paquets. Les repos de yum sont stockés dans le repertoire ***/etc/yum.repo.d*** et les fichiers correspondant aux repos ont l'extension ****.repo***

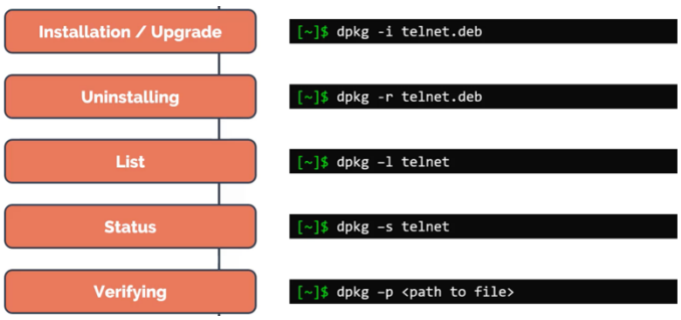
NB : YUM a toujours besoin de RPM afin d'installer les paquets sur le système à la différence de RPM qu'il permet de gérer les dépendances.

3.2.1. Commandes YUM

- ***yum repolist*** : affiche la liste des repos présents ou ajoutés à votre système
- ***yum provides <command>*** : permet d'afficher les paquets qui doivent être installés afin qu'une commande fonctionne
- ***yum install <package> ; yum remove <package> ; yum update <package> :*** permet d'installer, supprimer et mettre à jour un paquet
- ***yum update :*** pour mettre à jour tous les paquets du système

3.3. DPKG and APT

Dpkg est identique à RPM mais est utilisé pour dans les distributions Debian. Ci-dessous le commandes DPG



Les repos nécessaires à l'installation des paquets via APT se trouvent dans le répertoire ***/etc/apt/sources.list*** Ou alors peuvent avoir des sources distantes accessibles via HTTP, HTTPS ou FTP.

3.4. APT VS APT-GET

APT est meilleur que APT-GET et est également plus utilisé que ce dernier. Les différences entre ces deux outils sont les suivantes :

- L'installation d'un paquet avec APT est plus conviviale qu'une installation à partir de APT-GET ; en effet APT affiche à l'écran une sortie conviviale pour l'utilisateur avec pourcentage d'avancement, ce que APT-GET n'a pas.
- La recherche d'un paquet à l'aide de APT est plus facile et la sortie est meilleure qu'avec APT-GET. Pour APT, on utilise juste la commande ***apt search <package_name>*** tandis qu'avec APT-GET on utilise un autre outil ***apt-cache search <package_name>***

4. Linux Core Concepts II

4.1. File compression and Archival

4.1.1. Les commandes de manipulation de fichiers

- ***du -sk <file.img>*** : Affiche à l'écran la taille d'un fichier en Kilobits
- ***du -sh <file.img>*** : Affiche la taille dans un format facilement lisible par l'homme
- ***ls -lh <file.img> :*** affiche les propriétés d'un fichier

4.1.2. Les commandes d'archivage des fichiers

La commande par défaut e linux permettant d'archiver des fichiers (regrouper plusieurs fichiers dans un seul) est « ***tar :*** pour t archive ». Les options d'utilisation de cette commande sont les suivantes :

- ***tar -cf test.tar <file1> <file2> <file3>*** : compresser plusieurs fichiers dans une archive test.tar ; l'option -c permet de créer une archive et l'option -f est utilisée afin de spécifier le nom du fichier archive à créer
- ***tar -tf test.tar*** : utilisé pour voir le contenu de l'archive tar
- ***tar -xf test.tar*** : utilisé pour extraire un fichier archive
- ***tar -zcf test.tar file1 file2 file3*** : utilisé pour une compression optimisée (réduire un max la taille de l'archive)

On distingue d'autres outils de compression dans Linux parmi lesquelles les plus utilisés sont :

4.2. Searching for files and Patterns

Il existe plusieurs commandes de recherche de fichiers parmi lesquelles :

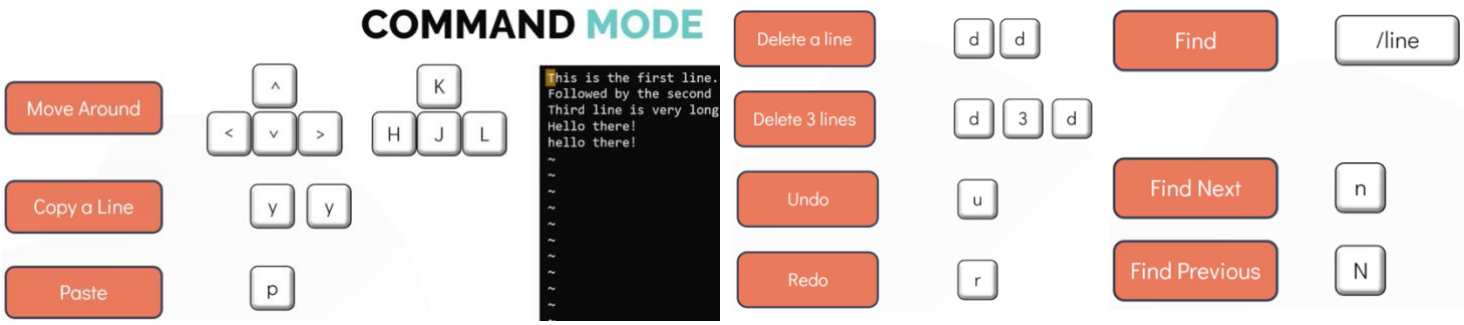
Les commandes de recherche de fichiers

- ***locate <filename.txt> :*** affiche en retour le ou les chemins de fichiers absolus où se trouve le fichier passé en paramètre.
- ***updatedb*** : commande permettant de mettre à jour la base de données de recherche pour la commande locate

- ***find /home/frazer -name <filename.txt>*** : recherche un fichier dans un répertoire particulier
- ***grep <mot-clé> filename.txt*** : permet de rechercher un mot clé dans u fichier (Il faut veiller à respecter la casse)
- ***grep -ir <mot-clé> /home/frazer :*** l'option -r permet de rechercher dans le répertoire passé en paramètre le fichier contenant le mot clé. Le -i permet d'éviter de gérer la casse par rapport au mot clé.
- ***grep -w <mot-clé> frazer.txt*** : l'option -w permet de rechercher dans le fichier les lignes contenant le exactement le mot clé passé en paramètre et non toutes les lignes ayant les mots composés en partie du mot clé (Ex si le mot clé est test et qu'on a dans le fichier une ligne ayant tester, cette dernière sera sélectionnée si pas d'option -w ; mais pas sélectionnée avec l'option -w)
- ***grep -vw <mot-clé> frazer.txt*** : le -v couplé au -w permet d'inverser le choix du -w donc seules les lignes ayant des mots contenant en partie le mot clé seront sélectionnées
- ***grep -A1 <mot clé> frazer.txt*** : le -A signifie After et permet d'afficher un certain nombre de lignes après la ligne contenant le mot clé (dans cet exemple on a A1 donc il affichera 1 ligne après)
- ***grep -B1 <mot> frazer.txt*** : ici on affiche un certain nombre de ligne avant la ligne de match (B pour before). Il est possible de combiner le -A et -B afin d'afficher des lignes avant et après la ligne de match.

4.3. IO Redirection

4.4. VI EDITOR



5. Networking

5.1. DNS

Les fichiers importants nécessaires pour la configuration du DNS sur des machines Linux sont les suivants :

- Le fichier local de configuration (***/etc/hosts***)

```
[~]$ cat >> /etc/hosts
192.168.1.115        test
```

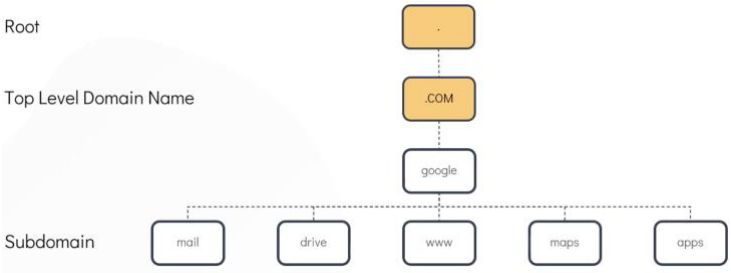
- Le fichier permettant de configurer le serveur DNS à utiliser (***/etc/resolv.conf***)

```
[~]$ cat >> /etc/resolv.conf
nameserver          192.168.1.100
nameserver          8.8.8.8
```

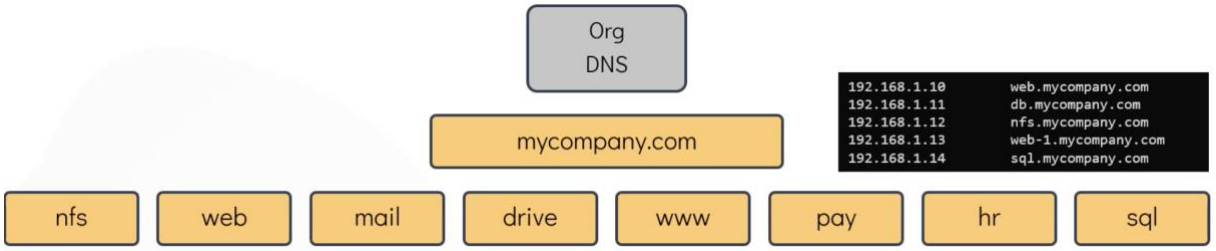
- Le fichier permettant de choisir l'ordre de priorité entre le fichier ***/etc/hosts*** et le serveur DNS (***/etc/nsswitch.conf***)

```
[~]$ cat /etc/nsswitch.conf
…
hosts:          files dns
```

5.2. Domain Names



Il est possible d'ajouter dans le fichier de résolution DNS /etc/resolv.conf une entrée correspondante au nom de domaine que l'on souhaite utiliser, afin que notre hôte complète systématique nos noms d'hôtes par ce nom de domaine. Pour cela on utilise l'entrée « search »



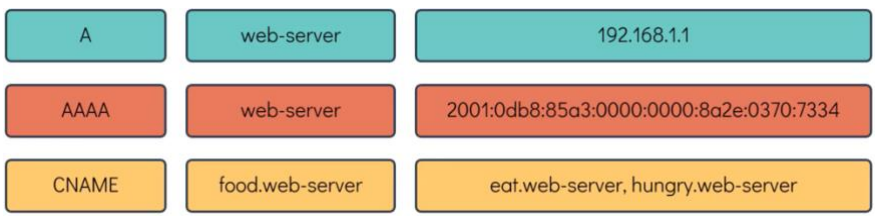

```
[~]$ cat >> /etc/resolv.conf
nameserver          192.168.1.100
search              mycompany.com
```

```
[~]$ ping web
PING web.mycompany.com (192.168.1.10) 56(84) bytes of data.
64 bytes from web.mycompany.com (192.168.1.10): … time=0.052 ms
64 bytes from web.mycompany.com (192.168.1.10): … time=0.079 ms
```

Il est également possible d'ajouter plusieurs noms de domaines à une seule entrée ***search*** et dans ce cas notre hôte devra tester l'ensemble de nos de domaines entrés afin de trouver quel est celui disponible et à utiliser

```
[~]$ cat >> /etc/resolv.conf
nameserver          192.168.1.100
search              mycompany.com    prod.mycompany.com
```

5.3. DNS Record Types



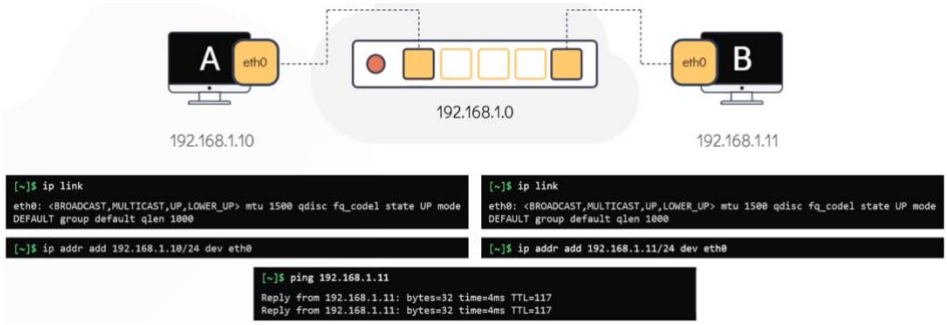
NB : On peut utiliser les cdes « ***nslookup*** ou ***dig*** » pour identifier l'adresse d'un nom DNS.

```
[~]$ nslookup www.google.com
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
Name:   www.google.com
Address: 172.217.0.132
```

```
[~]$ dig www.google.com
```

5.4. SWITCHING



Afin d'identifier les interfaces réseaux et leur statut sur une machine linux, on utilise la commande « ***ip link*** ». Une fois l'interface détectée, on peut utiliser la commande « ***ip addr <subnet> <network_name> dev <interface_name>*** ».

Après avoir configuré la carte Ethernet et lui avoir donné une adresse IP, il est important de configurer sa passerelle réseau ou Gateway si jamais l'on souhaite qu'il puisse communiquer avec un autre réseau. Pour ce faire on utilise les commandes suivantes :

- ***route*** : permet d'afficher l'ensemble des routes définies sur l'hôte
- ***ip route add <réseau_cible> via <gateway>*** : permet d'ajouter un Gateway ou une route vers un autre réseau
- ***ip route add default via <gateway>*** : permet de spécifier la Gateway à utiliser pour tout autre réseau (0.0.0.0/0) non spécifié dans les routes existantes.
- ***sudo ip link set dev eth0 up*** : active une carte réseau qui est à l'état DOWN
- ***traceroute <@ip>*** : permet d'identifier la route utilisée afin d'atteindre la cible

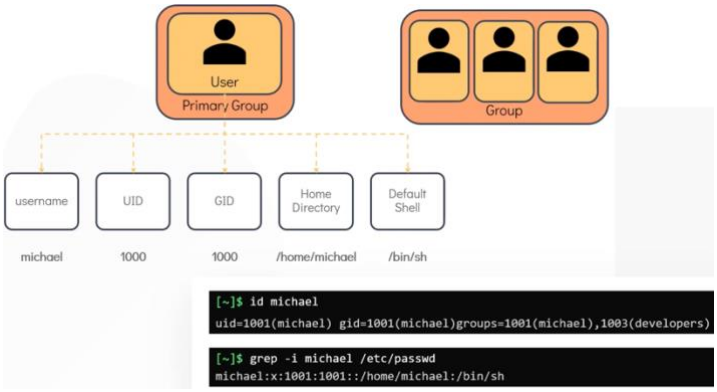
6. Security and file permissions

6.1. Linux Accounts

Les informations sur les utilisateurs sont stockées dans le fichier /***etc/passwd***. Chaque utilisateur a un username et un identifiant unique (UID) qui leur est associé. Les mots de passe des différents utilisateurs sont cryptés et stockés dans le fichier ***/etc/shadow***

Les informations concernant les groupes sont stockées dans le fichier ***/etc/group***, chaque groupe a un identifiant unique appelé GID. Ce GID désignant un groupe peut être associé à un utilisateur si jamais cet utilisateur appartient au groupe en question. Pour avoir les informations sur un user, on utilise la commande :

- ***id <username>*** : affiche le UID, GID, les groupes auxquels appartient un user
- ***grep -i <username> /etc/passwd :*** permet de vérifier les informations du user dans le fichier ***/etc/passwd*** permettant de connaitre son répertoire home et également le Shell utilisé par défaut.



6.1.1. Account types

Il existe différents types de comptes utilisateur Linux :

Pour avoir les informations sur un User, comme vu précédemment, on utilise la commande « ***id*** » qui nous affiche son ***UID*** ; ***GID*** et ses groupes. Pour avoir la liste des users actuellement connecté au système, on

```
[~]$ id
uid=1000(michael) gid=1000(michael) groups=1000(michael)
```

```
[~]$ who
bob      pts/2        Apr 28 06:48 (172.16.238.187)
```

```
[~]$ last
michael   :1     :1                    Tue May 12 20:00  still logged in
sarah     :1     :1                    Tue May 12 12:00  still running
reboot    system boot 5.3.0-758-gen    Mon May 11 13:00  - 19:00 (06:00)
```

utilise la commande « ***who*** », et si on veut savoir les users précédemment connectés, on utilise la commande « ***last*** ». Pour changer de user, on utilise la commande « ***su*** » ou alors le ***sudo***. Pour pouvoir utiliser le ***sudo*** avec un user, il faut au préalable que cet user soit défini dans le fichier ***/etc/sudoers*** qui peut être modifié avec la commande ***visudo***.

6.1.2. SUDO

```
[~]$ cat /etc/sudoers
User privilege specification
root    ALL=(ALL:ALL) ALL
# Members of the admin group may gain root
privileges
%admin ALL=(ALL) ALL
# Allow members of group sudo to execute any
command
%sudo   ALL=(ALL:ALL) ALL
# Allow Bob to run any command
bob ALL=(ALL:ALL) ALL
# Allow Sarah to reboot the system
sarah localhost=/usr/bin/shutdown -r now
# See sudoers(5) for more information on "#include"
directives:
#includedir /etc/sudoers.d
```

Field	Description	Example
1	User or Group	bob, %sudo (group)
2	Hosts	localhost, ALL(default)
3	User	ALL(default)
4	Command	/bin/ls, ALL(unrestricted)

6.2. Managing Users

6.2.1. les commandes

- ***useradd <username>*** : créer un nouveau user à qui sera assigné un UID et GID par défaut (en tant que sudo)
- ***passwd <username>*** : Créer ou modifier le mot de passe d'un utilisateur (en tant que sudo)
- ***whomami*** : permet d'afficher le user actuel
- ***passwd*** : sans argument, elle permet à un user de modifier son propre mot de passe
- ***userdel <username>*** : permet de supprimer un user (sudo)
- ***groupadd -g 1011 <group_name>*** : creer un groupe
- ***groupdel <grpup_name>*** : supprimer un groupe

Il est possible de créer un user en spécifiant les valeurs de ses différents paramètres :

```
[~]$ useradd -u 1009 -g 1009 -d /home/robert -s /bin/bash -c "Mercury Project member" bob
```

-G create user with multiple secondary groups

-s specify login shells

-u specific UID

-c Custom Comments

-d custom home directory

-e Expiry date

-g specific GID

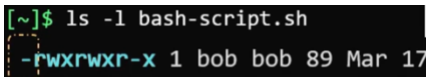
NB : si vous souhaitez attacher le user à un GID, vous devrez au préalable créer le groupe en question.

6.3. Access control files

Les 03 Access control files sur lesquels on s'attardera ici sont les suivants :

6.4. File Permissions and Ownership

Lorsque l'on réalise la commande « ***ls -l*** » sur un fichier dans linux, on obtient alors les différentes propriétés dudit fichier dans lesquels le premier caractère représente le type de fichiers et les autres caractères :

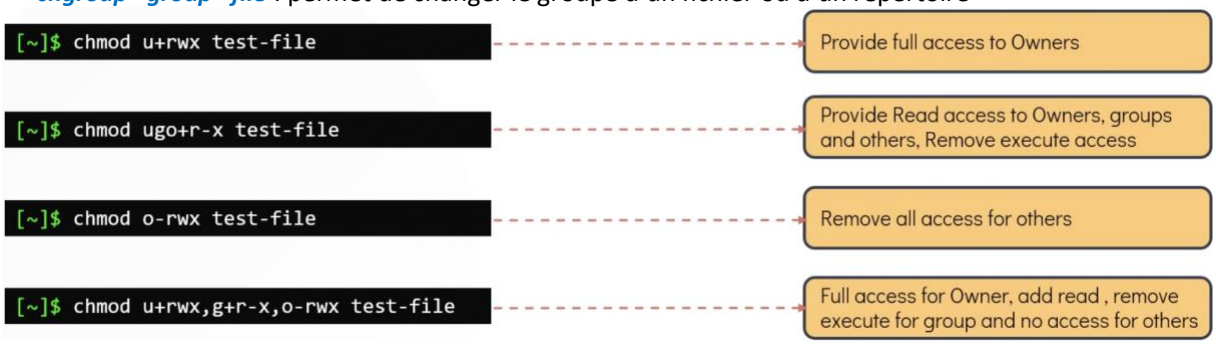


NB : Lors de l'application des autorisations ; Linux procède séquentiellement ; si la commande est passé par le user, alors Linux ne tiendra en compte que les droits liés au user et ignorera les autres ; si ce n'est pas le user et qu'il s'agit d'un user appartenant au groupe défini dans le paramètres du fichier, alors seuls les autorisations liés au groupe sont prises en compte…

Bit	Purpose	Octal Value
r	Read	4
w	Write	2
x	Execute	1
-	No permission	0

6.4.1. Les commandes

- ***chmod <permissions> file*** : permet de modifier les permissions d'un fichier
- ***chmod ugo+rwx file*** : donner les permissions r (read), w (write) et x (exécuter) au user (u) ; group (g) et autres (o)
- ***chmod 777 file*** : donne également les permissions r (read), w (write) et x (exécuter) au user ; au groupe et autres.
- ***chown <own:group> file*** : change owner permet de modifier le owner et le group auquel appartient un fichier
- ***chown <user> file*** : permet de changer juste le owner du fichier ou d'un repertoire
- ***chgroup <group> file*** : permet de changer le groupe d'un fichier ou d'un repertoire



6.5. SSH and SCP

6.5.1. SSH

Outil permettant de se connecter à un hôte distant. Pour les commandes SSH, il est obligatoire que le port 22 soit ouvert sur la machine distante et que vous ayez ses identifiants de connexion (username + password) ou alors que cette machine distante possède votre clé publique.

6.5.1.1. Les commandes

Les commandes pour le ssh sont les suivantes

- ***ssh <hostname or IPaddres>***
- ***ssh <user>@<hostname>***
- ***ssh -l <user> <hostname>***
- ***ssh-keygen -t rsa*** : générer la paire de clé ssh nécessaire
- ***ssh-copy-id <user@hostname>*** : permet de copier la clé publique sur la machine distante en passant par la méthode de connexion par mot de passe (il faut que le mode de connexion par mot de passe soit activé sur la machine distante et que vous ayez le mot de passe)

6.5.2. SCP

Outil permettant de copier un fichier ou répertoire sur un hôte distant. Cet outil utilise la méthode de connexion sécurisée du SSH et de ce fait nécessite les mêmes prérequis que pour le ssh (port 22 + password ou public key). La commande usuelle pour le ***scp*** est la suivante

scp <local_file-path> user@hostname:<remote_host_file_path> : il est nécessaire en effet que l'utilisateur utilisé pour la connexion est les autorisations nécessaires pour la copie du fichier dans le répertoire indiqué au niveau du serveur distant. On peut utiliser les options -r sur la commande ***scp*** si l'on souhaite copier un répertoire et si l'on veut conserver les permissions et owner du fichier, on peut utiliser l'option -p

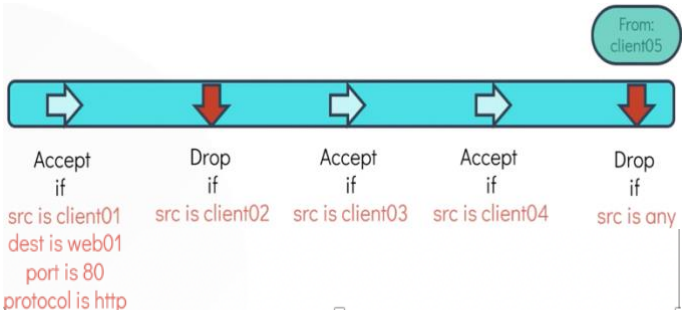
```
scp  /home/bob/caleston-code.tar.gz devapp01:/home/bob
bob@devapp01's password:
caleston-code.tar.gz
```

```
scp  /home/bob/caleston-code.tar.gz devapp01:/root
bob@devapp01's password:
Scp /root/caleston-code.tar.gz: Permission denied
```

```
scp -pr  /home/bob/media/ devapp01:/home/bob
```

6.6. IPTABLES

Il s'agit d'un outil Linux permettant de filtrer le Traffic réseau sur une machine. Il agit comme un pare feu et permet de contrôler, limiter et sécuriser l'accès réseau à une machine. Par défaut, l'outil iptable n'est pas installé sur linux, il faudrait donc au préalable l'installer avant de pouvoir s'en servir. Il existe 03 types de règles iptables : les règles entrantes (INPUT), permettant de filtrer le traffic entrant ; les règles OUTPUT, pour filtrer le traffic sortant et les règles FORWARD pour filtrer le traffic forwardé vers une autre machine, si jamais la machine en question est utilisée comme un routeur. Par défaut sur une machine tout le traffic entrant et sortant sont autorisés. IPTABLE classe les règles dans des chaines (CHAIN INPUT, FORWARD et OUTPUT. les chaines sont donc composées de règles qui sont ordonnées fonction de leur création (la première chaine est celle crée en premier) L'ordre de création des règles est très importante car iptable vérifie séquentiellement les règles d'une chaine dans l'ordre de création et une fois qu'une règle est respectée, il l'applique et ignore les suivantes ; si elle n'est pas respectée, il passe à la suivante et ainsi de suite.



Exemple : Pour cette chaine INPUT contenant plusieurs règles, toutes les conditions sont testées séquentiellement et dès que la condition est respectée la règle correspondante est appliquée et les autres sont ignorées.

6.6.1. Les Commandes

Option	Description
-A	Add Rule
-p	Protocol
-s	Source
-d	Destination
--dport	Destination port
-j	Action to take

- ***sudo apt install iptables*** : permet d'installer l'outil iptable sur une machine Linux
- ***sudo iptables -L*** : permet de lister les différentes règles iptables configurées sur le système
- ***iptables -A INPUT -p tcp -s 172.16.238.187 --dport 22 -j ACCEPT*** : permet d'ajouter une règle iptables de type INPUT afin de filtrer le traffic entrant qui acceptera toutes les requêtes tcp entrantes sur le port 22 et provenant de l'adresse IP définie.
- ***iptables -A INPUT -p tcp --dport 22 -j DROP :*** ajoute une règle qui rejette toutes les connexions tcp entrantes sur le port 22. lorsque la source n'est pas spécifiée, la règle concerne toutes les sources
- ***iptables -A OUTPUT -p tcp -d 172.16.238.11 --dport 5432 -J ACCEPT*** : autorise les connexions sortantes vers le port 5432 d'un hôte particulier
- ***iptables -I OUTPUT -p tcp -d 172.16.238.100 --dport 443 -J ACCEPT*** : ici on a utilisé l'option -I plutôt que l'option -A pour ajouter la règle ; en effet l'option -A ajoute la règle à la fin de la liste de règles de la chaine tandis que l'option _I permet d'insérer une règle en début de file de la chaine.
- ***iptables -D OUTPUT 5*** : permet de supprimer la règle numéro 5 de la chaine OUTPUT. l'option -D permet de supprimer une règle.

6.7. Cronjobs

Dans cette partie nous verrons comment planifier des jobs dans linux en créant des ressources appelées cron. les taches planifiées dans le cron sont exécutée de façon planifiée à l'aide du service crond qui tourne en arrière-plan et qui se chargera de lancer votre tâche.

Pour configurer une tache dans le cron, on tape la commande « ***crontab -e*** » (ne pas exécuter cette commande en tant que sudo sinon Linux ouvrira le cron du user root). Après avoir tapé cette commande, il suffit de se déplacer à la fin du fichier de configuration du cron afin d'insérer une nouvelle ligne pour notre tâche planifiée.


```
# m h  dom mon dow   command
  0 21 *   *   *    uptime >> /tmp/system-report.txt
```

Dans le fichier de configuration du cron, on doit configurer les variables de temps nécessaire pour la planification et par la suite spécifier la commande qui devra être exécutée. Au niveau de la planification, on dispose de 5 colonnes définies comme illustré dans l'image ci-après.

Requirement	Minute	Hour	Day	Month	Weekday
February 19th, 08:10 AM, Only if it's a Monday	10	8	19	2	1
February 19th, 08:10 AM, any weekday	10	8	19	2	*
19th of every month at 08:10 AM, any weekday	10	8	19	*	*
Every day of every month at 08:10 AM, any weekday	10	8	*	*	*
Every day of every month at 10 minutes past every hour or any weekday	10	*	*	*	*
Every day of every month at every minute past every hour or any weekday	*	*	*	*	*
Every day of every month at every other minute past every hour or any weekday. eg: 08:02, 08:04, 08:06... 09:02, 09:04, 09:06	*/2	*	*	*	*
Every day of every month at every other minute past every other hour or any weekday. eg: 08:02, 08:04, 08:06... 10:02, 10:04, 10:06... 12:02, 12:04, 12:06	*/2	*/2	*	*	*

Pour afficher la liste des taches cron « ***crontab -l*** » et pour vérifier qu'une tache s'est bien exécuté « ***tail /var/log/syslog*** » afin d'afficher les logs systèmes.

7. Storage in Linux

7.1. Disk Partitions

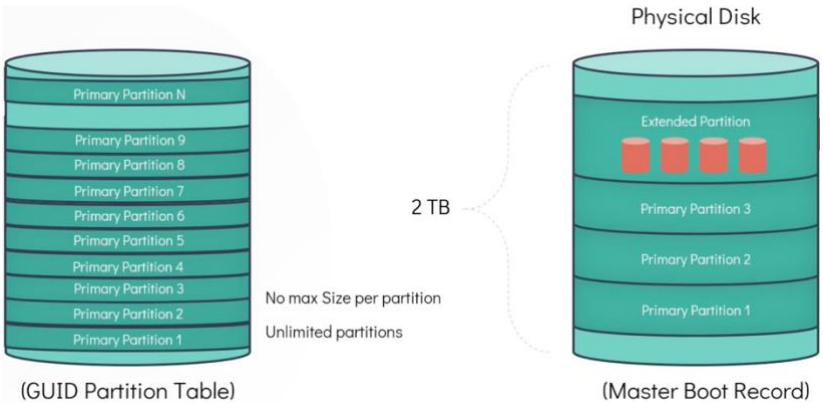
les devices ou périphériques de stockage sont considérés comme des fichiers de type bloc dans Linux qui sont stockées dans le répertoire ***/dev*** de la machine. Ils sont appelés blocs devices parce que les données sont stockées ou lues dans des blocs. Pour avoir la liste des blocs ou des devices d'une machine, on utilise les commandes suivantes :

- ***lsblk*** : liste l'ensemble des blocs
- ***ls -l /dev | grep ^b*** : affiche le contenu du répertoire ***/dev*** et juste les lignes commençant par b pour bloc
- ***sudo fdisk -l /dev/sda*** : permet de lister l'ensemble des partitions du bloc device ***/dev/sda***

Major Number	Device Type
1	RAM
3	HARD DISK or CD ROM
6	PARALLEL PRINTERS
8	SCSI DISK

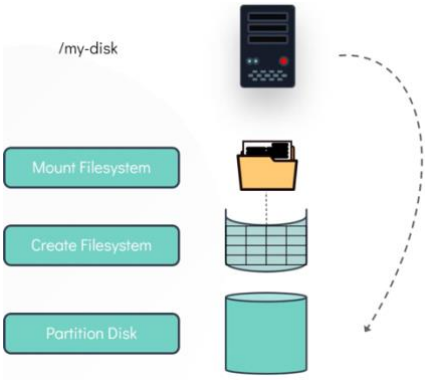
Il existe 03 types de partitions sur un disque :

- ***Primary partition ou partition primaire*** : partition principale généralement utilisée pour booter le système d'exploitation. Un disque physique de type MBR est limité à 04 partitions primaires maximum
- ***Extended partition ou partition étendue*** : partition ne pouvant pas être utilisée seule, mais pouvant héberger des partitions logiques. ce type de partition es beaucoup utilisé pour contourner la limitation de 04 partitions primaires par disque

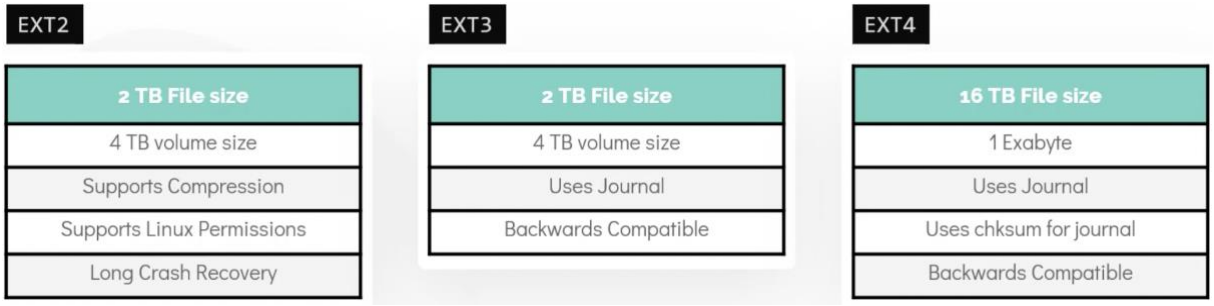


Pour créer une nouvelle partition sur le disque, on utilise l'outil gdisk qui est un outil plus amélioré que le fdisk :
« gdisk /dev/sdb » afin de créer une partition sur le disque sdb. la commande nous ouvre l'interface de l'outil gdisk ; on tape « ? » pour voir toutes les options ; ensuite n pour créer une nouvelle partition et w pour la sauvegarder.

7.2. File systems in Linux



Une fois qu'on a vu comment créer des partitions, il est important de parler des systèmes de fichiers car avant de pouvoir écrire dans un disque ou une partition, il est indispensable qu'un système de fichier soit créé au niveau de la partition, ensuite ce système de fichier sera monté dans un répertoire afin de permettre l'écriture sur le disque via ce répertoire. Le système de fichiers permet de définir la facon dont les données seront stockées dans le disque. Les systèmes de fichiers supportés par Linux sont :



7.2.1. Les commandes

- ***sudo mkfs.ext4 /dev/sdb1*** : créer un système de fichier de type ext4 sur le disque sdb1
- ***mkdir /mnt/ext4*** : créer le répertoire ext4 dans le répertoire ***/mnt*** qui est le répertoire de montage par défaut
- ***mount /dev/sdb1 /mnt/ext4*** : permet de monter le disque sdb1 sur le répertoire ***/mnt/ext4*** précédemment crée. Ce type de montage est provisoire et est perdu au redémarrage du système
- ***mount | grep /dev/sdb1*** : permet d'afficher le point de montage utilisé pour le disque sdb1
- ***df -hP*** : permet d'afficher les différents disques et leurs points de montage respectifs
- ***sudo blkid /dev/sdb1*** : permet d'afficher l'ID et le type de système de fichiers d'un bloc ou disque

Pour faire un montage persistent, il faut modifier le fichier ***/etc/fstab*** et y ajouter une entrée correspondante au montage que l'on souhaite faire.

/etc/fstab

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>          <options>                       <dump>  <pass>
/dev/sda1       /               ext4    defaults,relatime,errors=panic      0       1 ~
```

```
echo "/dev/sdb1  /mnt/ext4  ext4  rw 0 0" >> /etc/fstab
```

FIELD	Purpose
Filesystem	Such as /dev/vdb1 to be mounted
Mountpoint	Directory to be mounted on
Type	Example ext2, ext3, ext4
Options	Such as RW = Read-write, RO = Read Only
Dump	0 = Ignore, 1 = take backup
Pass	0 = ignore, 1 or 2 = FSCK filesystem check enforced.

Lorsqu'on renseigne une entrée dans le fichier ou table de montage ***/etc/fstab***, on commence par renseigner le disque ou la partition de disque dans laquelle on a déjà créé un système de fichier ; puis le répertoire dans lequel monté le disque ; ensuite le type de système de fichier à utiliser ; après on spécifie les options de montage qui spécifierons si on pourra Lire seulement ou lire et écrire sur le disque ; ensuite l'option de sauvegarde à l'aide d'un fichier backup (0 pour pas de sauvegarde et 1 pour sauvegarder) et pour terminer l'option de vérification du système de fichier au démarrage (à pour ignorer la vérif. et 1 ou 2 pour vérifier).

7.3. DAS, NAS and SAN

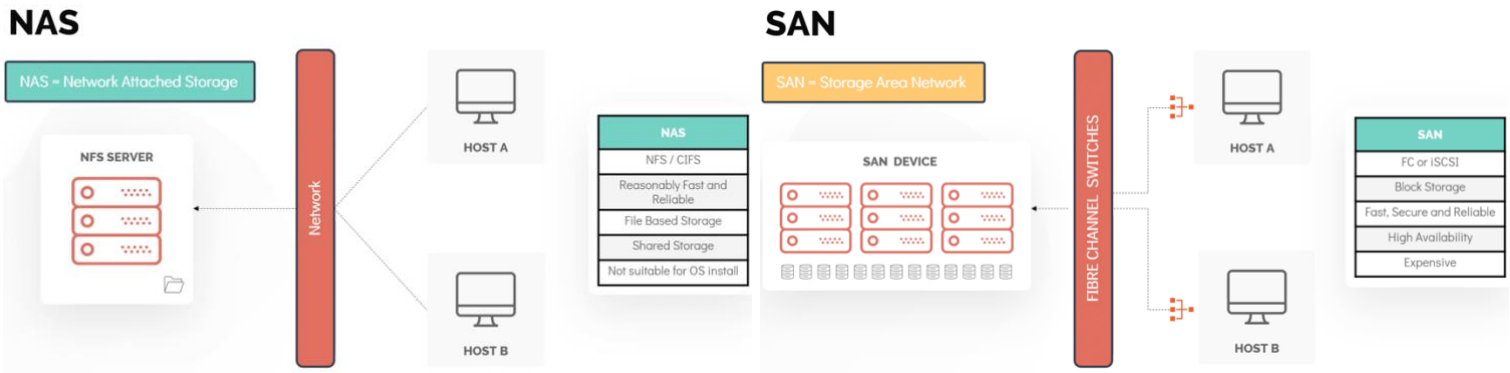
les différents types de stockages externes que nous verrons ici sont :

DAS = Direct Attached Storage

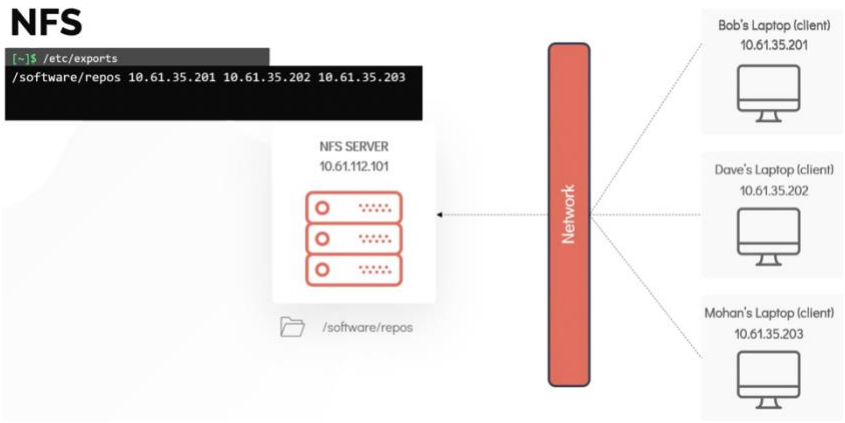
NAS = Network Attached Storage

SAN = Storage Area Network

- ***DAS*** : ici le périphérique de stockage externe est directement attaché à notre machine. Il n'existe aucun réseau ou pare-feu entre le disque et la machine, ce qui permet d'avoir de très bonnes performances par rapport aux NAS.
- ***NAS*** : ici le périphérique de stockage est relié à la machine à travers le réseau. idéal pour le stockage centralisé ou partagé entre plusieurs hôtes., il n'est pas recommandé pour installer un système d'exploitation
- ***SAN*** : fournit un stockage de type bloc utilisé par les entreprises pour les besoins ou applications critiques qui ont besoin d'une grande bande passante avec faible latence. le stockage est alloué aux hôtes à travers le réseau sous forme de ***LUN (logical unit number)***



7.4. NFS



Il faut noter que la technologie de stockage NFS utilisée sur les disques de type NAS ne sauvegarde pas les données dans des blocs, il les sauvegarde sous forme de fichiers. le NFS fonctionne généralement sous forme d'architecture client/serveur. Le NFS permet de partager des répertoires ou fichiers à travers le réseau et le terme utilisé pour le répertoire partagé en NFS est export. Le fichier de configuration des exports du serveur NFS est ***/etc/exports*** ; et pour partager un répertoire à travers le NFS, il faut ajouter une entrée dans ce fichier ***cf. image***.

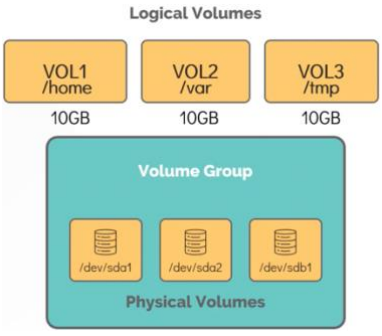
Dans ce fichier ***/etc/exports***, à la place du nom de ou des hôtes, on peut avoir l'adresse IP des machines devant y accéder, une plage réseau ou alors un Astérix (*) donnant l'accès à n'importe qui à ce répertoire partagé via NFS. après avoir paramétré le fichier ***/etc/export***, il faut lancer l'export des différents repos configuré à l'aide de la commande :

- ***exportfs -a*** : exporte tous les repos configurés dans le fichier ***/etc/exports***
- ***exportfs -o 10.61.35.201:/share-repo*** : permet d'exporter manuellement un répertoire (sans passer par le fichier de configuration)

Une fois le répertoire exporté et partagé avec les hôtes respectifs, il est par la suite possible de monter le répertoire exporté sur un hôte distant en utilisant la commande mount :

- ***mount 10.61.112.101:/software/repos /mnt/software/repos*** : monte ce répertoire distant partagé dans un répertoire de notre hôte

7.5. LVM (Logical volume manager)



LVM permet de grouper plusieurs disques physiques en un groupe de volume dans lesquels ces disques physiques peuvent être considérés comme des partitions du volume group. ces partitions du volume group peuvent être appelés volumes logiques. LVM permet de regrouper plusieurs disques physiques ou partitions entre elles afin de former un seul volume qui peut être partitionné par la suite en plusieurs volumes logiques. L'outil LVM a plusieurs avantages et permet au volumes logiques d'être redimensionné dynamiquement à souhait. Pour utliser l'outil LVM, on doit installer le paquet « ***lvm2 (apt-get install lvm2)*** ». la première chose à faire pour configurer le LVM est d'identifier les disques non utilisés du système afin de créer des objets LVM de types volumes physiques. les objets LVM de types volumes physiques sont utilisés pour identifeir les disques ou partitions du système. ils sont appelés physical volumes ou PV.

7.5.1. les commandes

- ***pvcrate /dev/sdb*** : permet de créer un volume physique LVM à partir du disque ou de la partition ***/dev/sdb*** qui est non utilisée.
- ***vgcreate frazer_vg /dev/sdb*** : permet de créer un volume group devant contenir le disque ***/dev/sdb*** (un volume group peut avoir plusieurs disques ou partitions)
- ***pvdisplay*** : affiche les informations détaillées sur les volumes physiques LVM
- ***vgdisplay*** : affiche les informations détaillées sur les volumes groupe
- ***lvcreate -L 1G -n vol1 frazer_vg*** : permet de créer un volume logique vol1 de 1Giga dans le volume groupe frazer_vg
- ***lvdisplay*** : affiche les informations détaillées sur les logical volume existants
- ***lvs*** : lister les logical volume existants
- ***pvremove, vgremove, lvremove*** : permettent de supprimer les différentes ressources
- ***mkfs.ext4 /dev/frazer_vg/vol1*** : formate le logical volume et crée le système de fichiers de type ext4
- ***mount -t /dev/frazer_vg/vol1 /mnt/vol1*** : permet de monter le logical volume récemment créé.
- ***vgs*** : permet d'avoir la liste de volume group existants
- ***lvresize -L +1G /dev/frazer_vg/vol1*** : permet d'étendre la taille du volume logique de plus 1 giga (bien vouloir vérifier au préalable s'il y'a suffisamment d'espace disponible au niveau du volume group de ce logical volume). NB : cette commande permet d'étendre le volume logique et non le système de fichiers créé sur le volume
- ***resize2fs /dev/frazer_sg/vol1*** : permet après avoir étendue le volume de façon logique ; d'étendre également le système de fichier précédemment créé sur ce disque. ce n'est qu'à ce moment que vous pourrez confirmer que la taille de votre volume a été étendue avec une commande « ***df -hP*** »

Logical Volume	Filesystem Path
vol1	/dev/caleston_vg/vol1
vol1	/dev/mapper/caleston_vg-vol1

NB : les volumes logiques créés à l'aide du LVM sont accessible à travers les 2 chemins de fichiers présentés dans l'image ci-après.

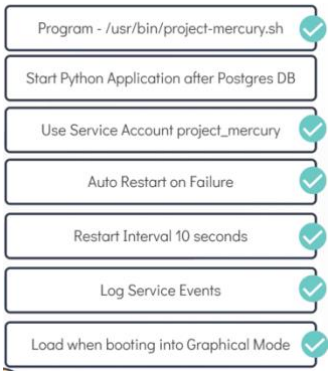
8. Service Management with SYSTEMD

8.1. Creating a SYSTEMD Service

```
/etc/systemd/system/project-mercury.service
[Unit]
Description=Python Django for Project Mercury
Documentation=http://wiki.caleston-dev.ca/mercury
After=postgresql.service

[Service]
ExecStart= /usr/bin/project-mercury.sh
User=project_mercury
Restart=on-failure
RestartSec=10

[Install]
WantedBy graphical.target
```

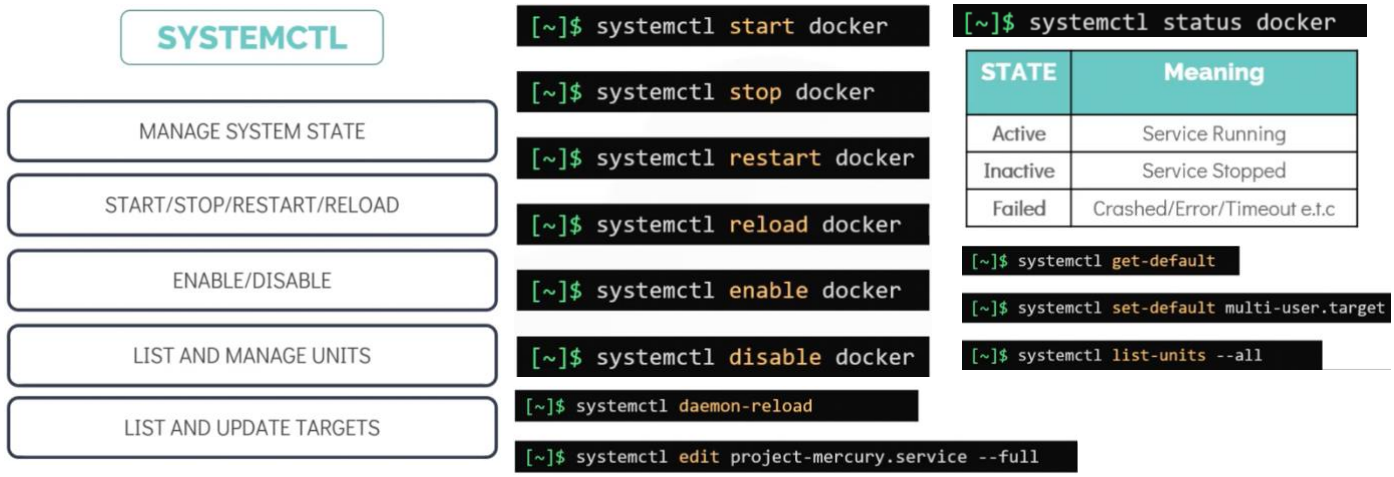


Pour exécuter un script ou une application en arrière-plan, on doit la créer comme étant un service. Il faut dans ce cas créer un fichier à l'extension ****.service*** qui devra être situé dans le répertoire ***/etc/systemd/system/<my-service>.service***.
Pour créer le service, dans le fichier de définition, on doit avoir la section ***[service]*** dans laquelle à l'aide de l'argument ***ExecStart*** on définit la commande à exécuter au lancement du service. une fois le service crée, on peut utiliser les commandes systemctl pour le démarrer, l'arrêter, voir son statut. Pour permettre au service d'être disponible durant le boot, il faut ajouter une autre section ***[install]*** et à l'intérieur de cette section ; renseigner la valeur de l'argument ***WantedBy*** avec les systemd target pour lequel le service devrait être disponible. Afin de spécifier l'utilisateur à utiliser pour démarrer le service durant la phase de boot, on ajoute dans la section ***[service]*** l'argument ***User*** et on définit sa valeur. puis on peut ajouter dans la même section l'argument ***Restart*** afin de définir à quel moment redémarrer le service et l'argument ***RestartSec*** pour définir le temps après lequel il doit redémarrer en cas de problèmes. NB avec Systemd ; les logs des différents services sont automatiquement enregistrés et ne nécessitent donc aucune configuration particulière lors de la définition du service. Si jamais notre service à des dépendances que l'on aimerait qu'elles soient au préalable démarré avant de démarrer notre service, on peut configurer cela en rajoutant la section ***[Unit]*** dans laquelle on utilisera l'argument ***After*** afin de définir le service qui devra être démarré avant le démarrage de notre service. dans cette même section, on peut rajouter les arguments ***Description*** et ***Documentation*** pour donner plus amples informations sur ledit service

```
[~]$ systemctl daemon-reload
```

NB : permet de recharger un service après l'avoir modifié.

8.2. SYSTEMD TOOLS



- ***journalctl*** : affiche les logs de tous les services du système du plus ancien au plus récent
- ***journalctl -b*** : affiche les logs du boot en cours
- ***journalctl -u <Unit>*** : affiche les logs d'un service en particulier