Administration Avancée

Installation des logiciels

➢ Paquets Linux :

❖ *Gestionnaire de paquets*

- **Un gestionnaire de paquets est un outil qui automatise le processus d'installation, désinstallation, mise à jour de logiciels installés sur un système informatique. Un paquet est une archive comprenant les fichiers informatiques, les informations et procédures nécessaires à l'installation d'un logiciel sur un système d'exploitation, en s'assurant toujours de la cohérence fonctionnelle du système.**

❖ *Utilité*

- **Le gestionnaire de paquets permet d'effectuer différentes opérations sur les paquets disponibles**
  - ✓ **Installation, mise à jour, et désinstallation ;**
  - ✓ **Utilisation des paquets provenant de supports variés (CD d'installation, dépôts sur internet, partage réseau …) ;**
  - ✓ **Vérification des sommes de contrôle de chaque paquet récupéré pour en vérifier l'intégrité ;**
  - ✓ **Vérification des dépendances logicielles afin d'obtenir une version fonctionnelle d'un paquetage**

Installation des logiciels

➢ Paquets Linux :

❖ *Les types des systèmes de paquets*

- On trouve deux grands types de système de paquets selon les grandes familles de distributions Linux :

- **RPM : Redhat Enterprise Linux, Fedora, Centos, …**
- **DPKG: Debian, Ubuntu, Mint, Raspbian, …**

- D'autres systèmes existent aussi :

- **Portage/emerge : Gentoo**
- **Pacman : Archlinux**
- **opkg : OpenWRT**

Installation des logiciels

➢ Paquets Linux :

❖ *Utilitaire dpkg*

- **Dpkg est utilisé pour installer, supprimer et fournir des informations à propos des paquets *.deb qui sont supportés par les distributions basées sur Ubuntu. Outil de bas niveau, dpkg -i / dpkg -r permettent d'installer ou de désinstaller des fichiers .deb. Pour ces tâches, on préfère utiliser des outils plus avancés comme aptitude ou apt-get, apt-cache.**

Commandes utiles	Signification
dpkg –l ou « dpkg --get-selections »	Pour lister tous les paquets installés avec des droits priviliégés
dpkg -s wget	Pour vérifier qu'un paquet soit installé
dpkg -L wget	Pour lister les fichiers installés par un paquet
dpkg-reconfigure locales	Pour reconfigurer un paquet installé

Installation des logiciels

➢ Paquets Linux :

❖ *Dépôt de paquets*

- **Un gestionnaire de paquet avancé comme apt ou yum gère des sources de logiciels et leur authenticité. « apt pour les distribution Debian/Ubuntu, yum pour les RedHat, Fedora »**
- **Le lieu où sont placés ses sources est appelé dépôt de paquet. Cette source est la plupart du temps une source locale comme un CD ou un DVD, un serveur Internet HTTP/FTP ou encore un miroir de dépôt local.**
- **Son principe de fonctionnement est de :**
  - ✓ **Mettre à jour les logiciels disponibles qui sont contenus dans une liste afin d'assurer leur cohérence au niveau de système.**
  - ✓ **Au moment de la demande d'installation, cette liste est consultée pour prendre les fichiers nécessaires.**
  - ✓ **Éventuellement, le système de paquetage installe automatiquement un service et le démarre**

Installation des logiciels

➢ Paquets Linux :

❖ *Dépôt de paquets*

- **Ses Taches sont comme suit :**
  - ❑ Vérification de l'existence d'un paquet
  - ❑ Version du logiciel dans le paquet
  - ❑ Fichiers de configuration
  - ❑ Source
  - ❑ Fichiers de configuration /etc
  - ❑ Désinstallation
  - ❑ Purge des fichiers
  - ❑ Suppression des dépendances orphelines

APT

APT simplifie l'installation, la mise à jour et la désinstallation de logiciels en automatisant la récupération de paquets à partir de sources APT (sur Internet, le réseau local, des CD-ROM, etc.), la gestion des dépendances et parfois la compilation.

```
apt-get update
```

Lorsque des paquets sont installés, mis à jour ou enlevés, la commande apt peut afficher les dépendances des paquets, demander à l'administrateur si des paquets recommandés par des paquets nouvellement installés devraient aussi être installés, et résoudre les dépendances automatiquement.

- Source d'apt sous Ubuntu Xenial :

```
cat /etc/apt/sources.list
```

```
deb https://archive.ubuntu.com/ubuntu xenial main universe
deb https://archive.ubuntu.com/ubuntu xenial-updates main universe
deb https://archive.ubuntu.com/ubuntu xenial-security main universe
```

Installation des logiciels

➢ Paquets Linux :

❖ *Comparatif des gestionnaires de paquets par distribution*

- Au point de vue de l'administrateur système, les distributions Linux peuvent se distinguer par :

- ✓ le gestionnaire et le système de paquets
- ✓ les scripts d'initialisation et les niveaux d'exécution
- ✓ le chargeur de démarrage
- ✓ l'emplacement des fichiers de configuration du réseau et des dépôts

Action	Debian/Ubuntu	Fedora/RHEL/SL/Centos
1. Mise à jour de la liste des paquets	apt-get update	yum update, yum check-update
2. Affichage des mises-à-jour disponibles	apt-get upgrade --simulate	yum list updates
3. Installation de paquets spécifiques	apt-get install package1 package2	yum install package1 package2
4. Réinstallation d'un paquet	apt-get install --reinstall package	yum reinstall package
5. Mise à jour d'un paquet	apt-get upgrade package1 package2	yum update package

Installation des logiciels

➢ Paquets Linux :

❖ *Comparatif des gestionnaires de paquets par distribution*

Action	Debian/Ubuntu	Fedora/RHEL/SL/Centos
6. Mise à jour du système	apt-get upgrade, apt-get dist-upgrade, apt upgrade, apt full-upgrade	yum upgrade
7. Recherche de paquets	apt-cache search searchword, apt-cache search --full --names-only searchword	yum search searchword
8. Liste de paquets installés	dpkg -l, apt list --installed	rpm -qa
9. Information sur un paquet	apt-cache show package, apt show package, dpkg -s package	yum info package, yum list package, yum deplist package
10. Désinstaller des paquets	apt-get remove --purge package1 package2, apt-get autoremove	yum remove package1 package2
11. Téléchargement de paquets sans installation	apt-get install --download-only package1 package2	yum install --downloadonly --downloaddir=<directory> <package>
12. Effacement des paquets téléchargés	apt-get clean, apt-get clean (paquets dépassés)	yum clean all
13. Configuration des dépôts	etc/apt/sources.list	/etc/yum.repos.d/

Installation des logiciels

➢ Installation par les sources :

- **On peut trouver les sources sous forme de paquet qui les placera dans le x. Ces sources ne sont pas exactement celles de kernel.org**

```
# apt-get update
# apt-cache search ^linux-source
linux-source-3.16 - Linux kernel source for version 3.16 with Debian patches
linux-source - Linux kernel source (meta-package)
# apt-get -y install linux-source-3.16
# ls /usr/src/linux-source*
/usr/src/linux-source-3.16.tar.xz
```

- **On peut aussi prendre les sources officielles sur ftp.kernel.org. Par la commande :**

wget « lien_URL »

Installation des logiciels

➢ Mettre en place un dépôt de paquets:

❖ *Dépôt de paquets*

- **Ses Taches sont comme suit :**
  - ❑ **Vérification de l'existence d'un paquet**
  - ❑ **Version du logiciel dans le paquet**
  - ❑ **Fichiers de configuration**
  - ❑ **Source**
  - ❑ **Fichiers de configuration /etc**
  - ❑ **Désinstallation**
  - ❑ **Purge des fichiers**
  - ❑ **Suppression des dépendances orphelines**

Objectif d'un dépôt local :

- ❑ **Se passer d'un dépôt distant**
- ❑ **Diminuer le temps et la bande passante consommée par des mise-à-jour et des installations**
- ❑ **Offrir des dépôts de paquets supplémentaires**

- Source d'apt sous Ubuntu Xenial :

Installation des logiciels

➢ Mettre en place un dépôt de paquets:

❖ *Apt-mirror*

La création d'un miroir pour les paquets accessibles par votre gestionnaire de paquets permettre de créer et de maintenir la copie conforme de dépôts (officiels ou non) en local.

La raison principale est de ne plus avoir besoin de connexion vers le net pour pouvoir installer un paquet ou bien faire des mises à jour.

- ⇒ **Une solution pratique et efficace pour une installation party, pour un utilisateur qui n'a pas une connexion vers le net, ou dont la connexion est trop lente.**
- ⇒ **La mise à jour d'un parc de machines (dans ce cas le miroir peut être couplé avec un serveur, un proxy, etc.) ou, pour en finir, la mise à jour ou l'installation chez une personne**

⇒ **Apt-mirror est le logiciel qui vous permet de créer le miroir des dépôts, pour l'utiliser c'est très simple, il suffit d'installer le paquet via « apt-mirror »**

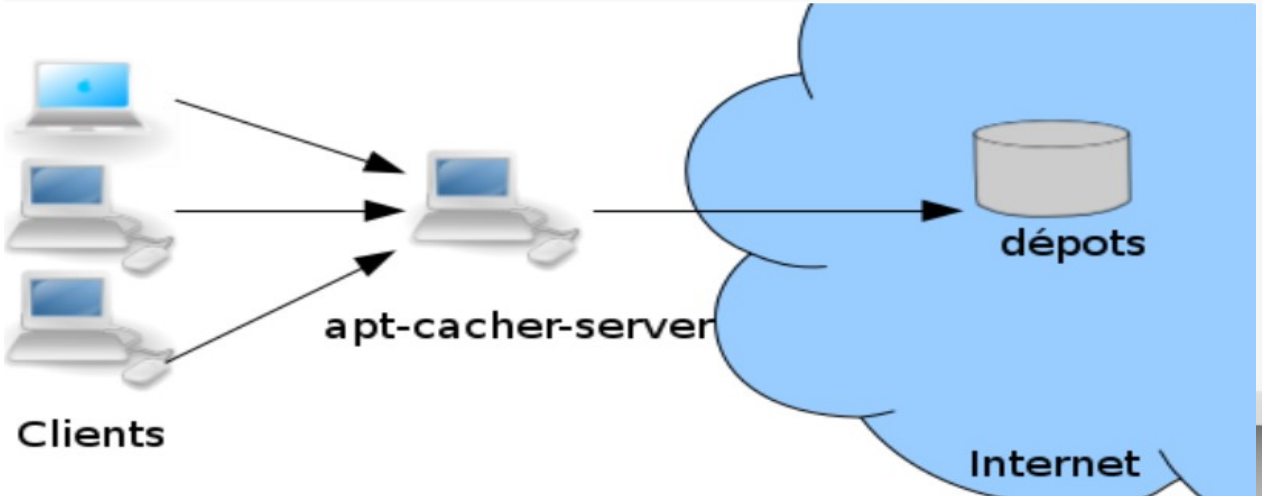
Installation des logiciels

➢ Mettre en place un dépôt de paquets:

❖ *Apt-cacher*

✓ **apt-cacher est une solution proxy de mise en cache des paquets Debian. À travers ce proxy, un ensemble d'ordinateurs clients accède indirectement aux dépôts.**

✓ **Quand un paquet est demandé pour la première fois, il est téléchargé par le proxy et transmis au client tout en conservant une copie en local. Pour toute future demande du même paquet, le proxy ne télécharge pas les paquets mais transmet la copie locale. Ainsi, on économise la bande passante externe et du temps pour les clients.**



Pré-requis

✓ **Vérifier que le dépôt universel soit bien activé et mis à jour.**

✓ **Avoir les droits d'administration sur toutes les machines.**

Installation des logiciels

➢ Mettre en place un dépôt de paquets:

❖ *Apt-cacher*

✓ **Installer les paquets apt-cacher**

```
sudo apt-get install apt-cacher
```

✓ **Configuration du serveur**

1. **Activer apt-cacher automatiquement**
2. **Il est recommandé pour des raisons de performances et d'utilisation de la mémoire de lancer apt-cacher en mode autonome (Stand-alone Daemon) :**
3. **Éditer le fichier /etc/default/apt-cacher et mettre l'option AUTOSTART à 1**

```
AUTOSTART=1
```

✓ **Lancement d'apt-cacher**

```
sudo service apt-cacher start
```

⇒ **À partir de Ubuntu 12.04, il faut modifier allowed_hosts dans /etc/apt-cacher/apt-cacher.conf.**

Par exemple :

```
allowed_hosts = *
```

puis on relance le apt-cacher

```
sudo service apt-cacher restart
```

Installation des logiciels

➢ Installation automatique :

❖ Caractéristiques principales

Un outil pour une installation automatisée sans surveillance. que Les administrateurs système aiment le faire; cette installation a ces caractéristiques :

- ❖ **Installation réseau à distance de différentes versions de Linux**
- ❖ **Système de gestion centralisé facile à utiliser pour votre déploiement Linux.**
- ❖ **C'est rapide. Cela ne prend que quelques minutes pour une installation complète.**
- ❖ **Installation automatique montre un menu basé sur les curseurs pour sélectionner un profil**
- ❖ **Création facile de supports d'installation Linux personnalisés tels que CD, DVD ou clé USB**
- ❖ **Découverte automatique du serveur d'installation automatique lors d'une installation réseau**
- ❖ **Évolutif. Les administrateurs système utilisent installation automatique pour gérer leurs infrastructures informatiques à partir de quelques ordinateurs jusqu'à plusieurs milliers de machines.**

……..etc

Installation des logiciels

➢ Installation automatique :

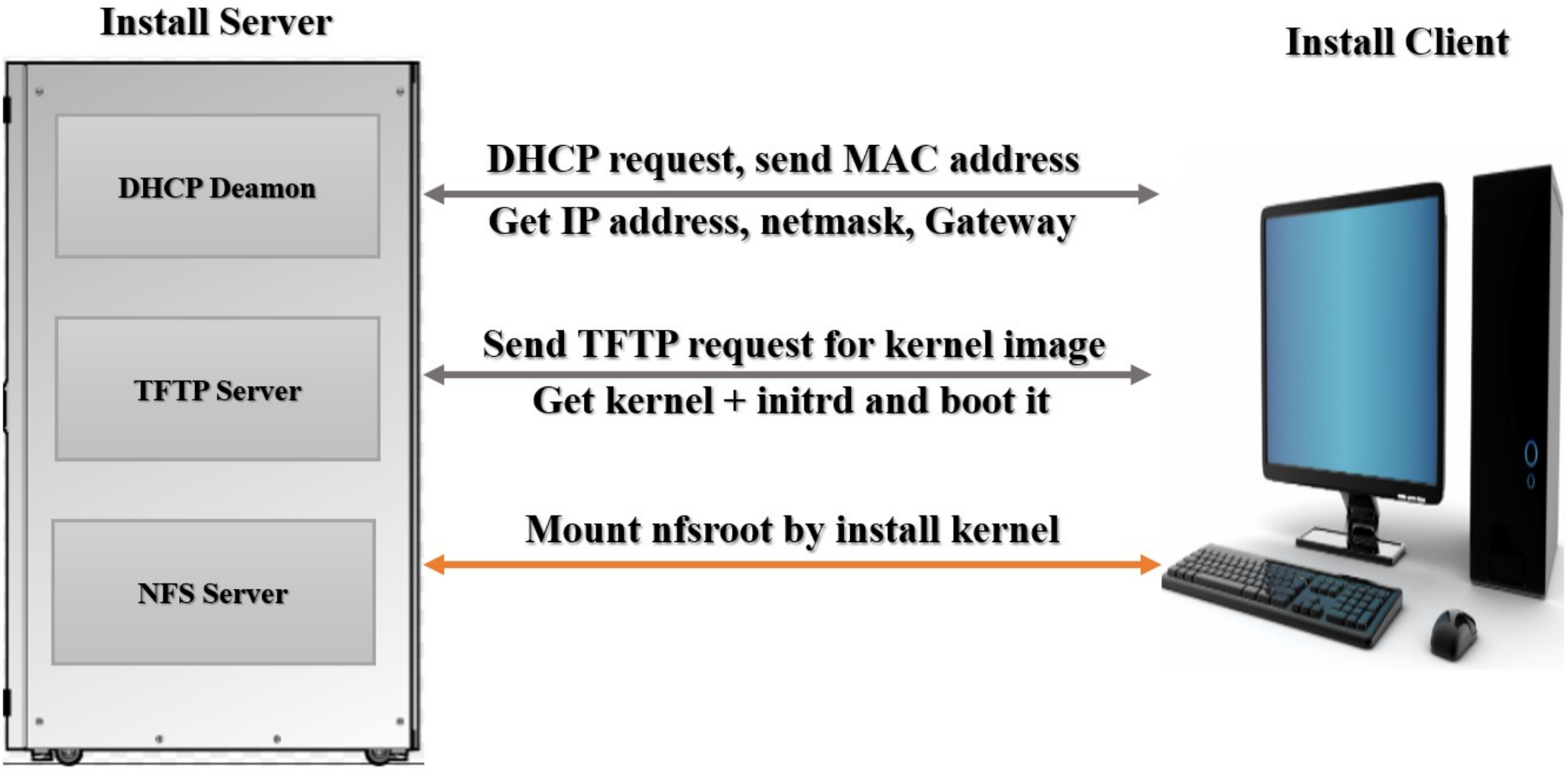
❖ Étapes d'installation automatique

- ❑ Démarrage réseau via PXE
- ❑ Recevez les données de configuration via HTTP, NFS, svn ou git
- ❑ Exécutez des scripts pour déterminer les classes et les variables FAI
- ❑ Partitionnez les disques durs locaux et créez la configuration RAID, LVM et les systèmes de fichiers
- ❑ Installer et configurer des packages logiciels
- ❑ Personnalisez le système d'exploitation et les logiciels en fonction de vos besoins locaux
- ❑ Redémarrez la machine fraîchement installée
- ❑ Tout cela peut également être fait via une installation de CD sans surveillance

Installation des logiciels

➢ Installation automatique :

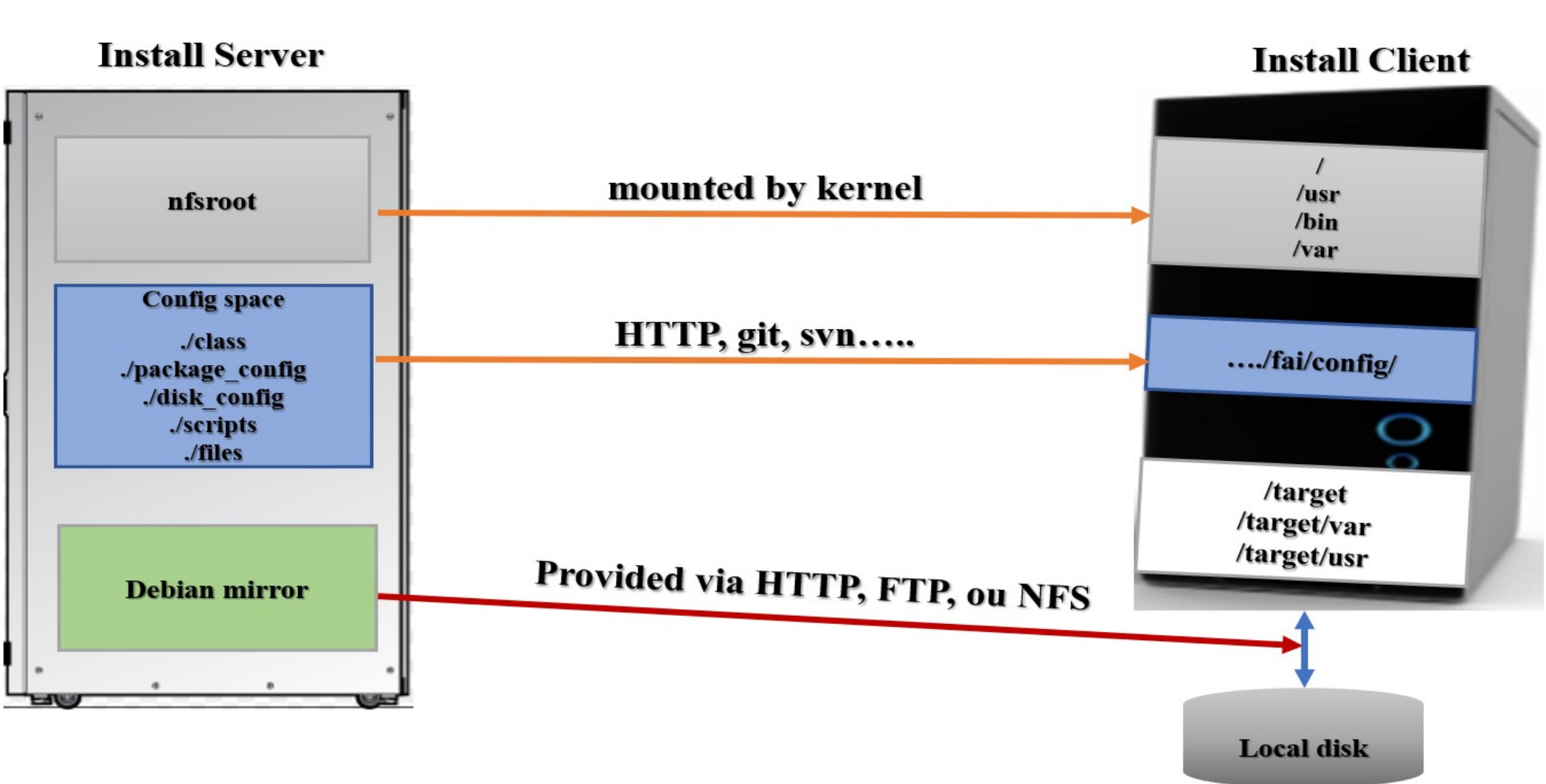
❖ Boot via Network Card (PXE)



Installation des logiciels

➢ Installation automatique :

❖ Configuration des données



Installation des logiciels

➢ Installation automatique :

❖ Exécution de l'installation

- ❑ **Partitionner les disques dur et créer les fichiers de système**
- ❑ **Installation des logiciels en utilisant la commande « apt-get »**
- ❑ **Configuration du système d'exploitation et ajout de d'autre application**
- ❑ **Sauvegarder les fichiers log puis redémarrage du nouveau système**

Installation des logiciels

➢ Stockage LVM :

- ✓ **LVM est un ensemble d'outils de l'espace utilisateur Linux pour fournir des commodités de gestion du stockage (volumes).**
- ✓ **LVM (Logical Volume Manager) répond principalement aux besoins de :**

  ⇒ **Evolutivité des capacités de stockage**

  ⇒ **Assurer la disponibilité du service.**

  Il s'agit de redimensionner un système de fichiers (FS) dynamiquement (en augmentant ou en réduisant le nombre de disques physiques disponibles) avec un minimum d'interruption.

Installation des logiciels

➢Stockage LVM :

❖La fonctionnalité LVM

✓La commande lsblk vous indique la manière dont vos disques sont montés. Aussi, la commande df –h vous donne des informations utiles.

✓On peut illustrer la fonctionnalité LVM dans le cas suivant.

✓Habituellement, un disque est constitué d'une ou plusieurs partitions :

a. soit monté en racine unique d'un système,

b. soit qui héberge le point de montage d'une application (/home, /var/www/html, /opt/nfs-share/, …)

c. ou une partition Swap

✓Par exemple, les partitions configurées occupent entièrement les 128Go que pour offrir un disque /dev/sda.

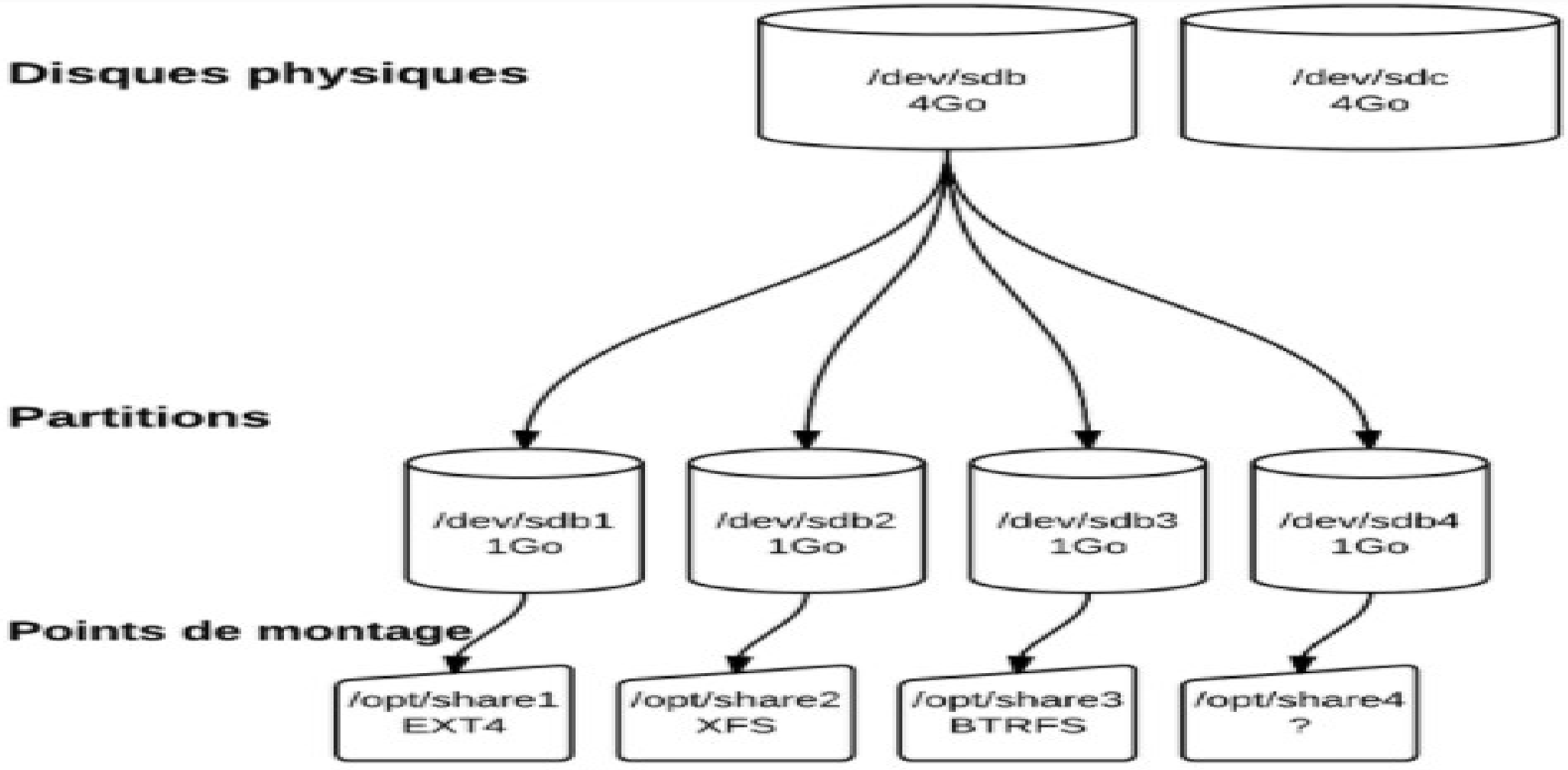
✓La solution sans LVM consisterait à copier les données du système de fichiers saturé sur le système de fichiers d'un nouveau disque de plus grande capacité ajouté. On peut aussi réaliser le redimensionnement avec des outils comme parted ou d'autres biens connus.

⇒Dans ce cas, il y'a un manque en disponibilité et en évolutivité de la solution de stockage.

Installation des logiciels

➢ Stockage LVM :

❖ Situation sans LVM



Installation des logiciels

➢ Stockage LVM :

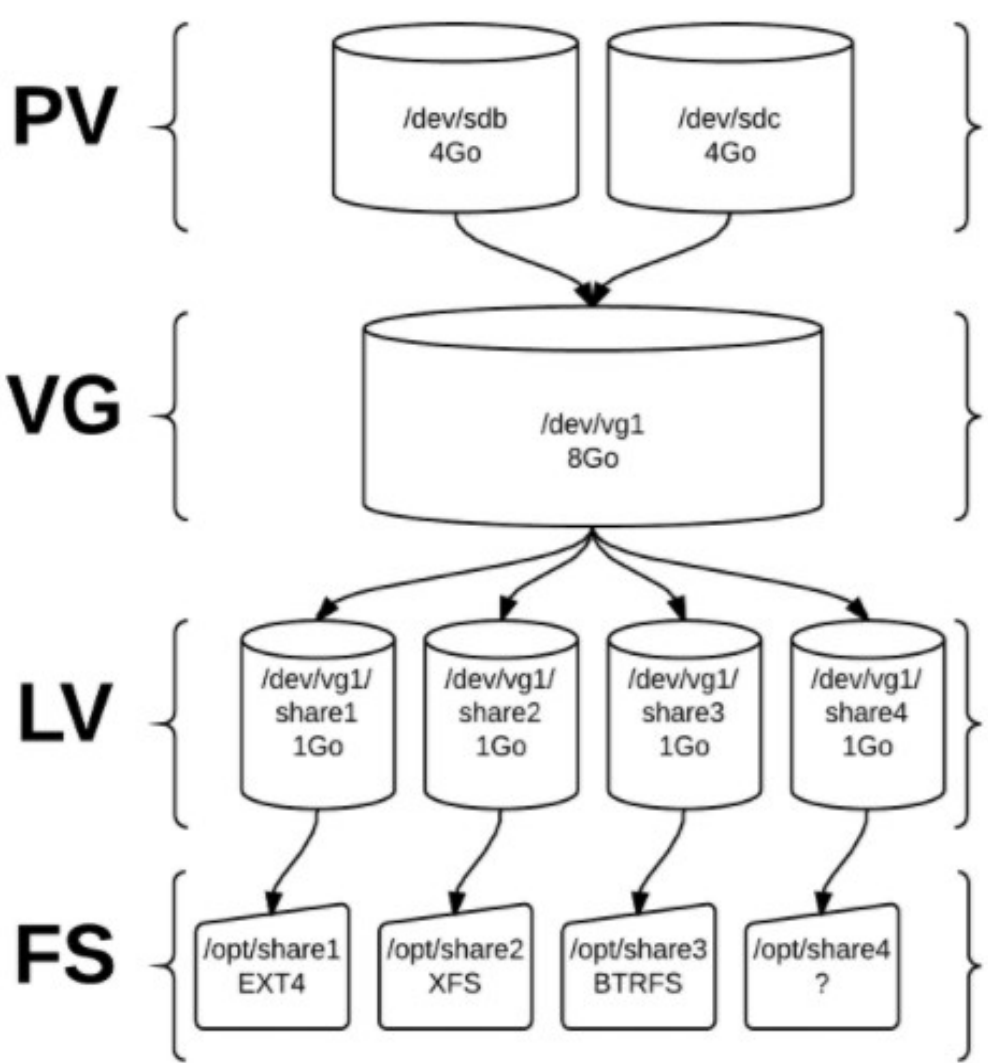
❖ Solution LVM

- ✓ **Sous certaines conditions, LVM autorisera un taux de disponibilité proche du maximum lors du redimensionnement du système de fichiers qui consiste souvent en une extension en capacité.**
- ✓ **En supplément, LVM supporte deux fonctionnalités qui améliorent ces critères : le mirroring et les snapshots**
- ✓ **Avec LVM, le système de fichiers (FS : EXT4, XFS, BTRFS, …) est supporté par un Logical Volume (LV) au lieu d'être supporté par une partition ou autre périphérique. Un LV est un container de FS.**
- ✓ **Les LV appartiennent à un Volume Group (VG). Un VG est une sorte d'entité logique qui représente une capacité de stockage**
- ✓ **Le noyau voit les VG comme des périphériques de type block (commande lsblk) et leurs LV comme leurs partitions. Ces périphériques sont dénommés par UUID, selon le schéma /dev/mapper/vg-lv ou encore selon le schéma /dev/vg/lv.**
- ✓ **Un VG est constitué d'un ensemble de Physical Volume (PV).**

Installation des logiciels

➢ Stockage LVM :

❖ Solution LVM



- ✓ **Avec LVM, le système de fichiers (FS : EXT4, XFS, BTRFS, …) est supporté par un Logical Volume (LV) au lieu d'être supporté par une partition ou autre périphérique. Un LV est un container de FS.**
- ✓ **Les LV appartiennent à un Volume Group (VG). Un VG est une sorte d'entité logique qui représente une capacité de stockage**
- ✓ **Le noyau voit les VG comme des périphériques de type block (commande lsblk) et leurs LV comme leurs partitions. Ces périphériques sont dénommés par UUID, selon le schéma /dev/mapper/vg-lv ou encore selon le schéma /dev/vg/lv.**
- ✓ **Un VG est constitué d'un ensemble de Physical Volume (PV).**

Installation des logiciels

➢Stockage LVM :

❖Solution LVM

✓Les PV sont les périphériques physiques de stockage. Ils peuvent être :

a.Un disque entier dont on a effacé le secteur d'amorçage (les premiers 512 octects du disque).

b.Une partition d'un disque marquée par fdisk

c.Un fichier de loopback.

d.Un array RAID.

❖Déploiement du LVM

✓Installation

```
apt-get install lvm2
```

✓Liste des commandes LVM

```
dpkg -L lvm2
```

✓Partition racine unique est /boot vérification par :

```
fdisk -l /dev/sda
```

```
lvmdiskscan
```

✓Initialisation de PV

```
pvcreate /dev/sdx
```

✓Visualisation

```
pvscan
```

✓création d'un VG et un LV initial

```
vgcreate vg1 /dev/sdx /dev/sdy
```

```
lvcreate -L 8G -n lv1 vg1
```

Configuration du réseau

➢ Des commandes réseau sous Linux :

Il y a trois paramètres nécessaires pour établir une connexion TCP/IP globale à partir d'un ordinateur :

- **Une adresse IP et son masque**
- **Une passerelle par défaut**
- **Un serveur de résolution de nom**

Fichiers de configuration des interfaces

- **Debian : /etc/network/interfaces/**
- **Centos : /etc/sysconfig/network-scripts/ifcfg-$NETDEV**

Paramètres	Commande	Signification
@ IP et son masque	ip addr show ifconfig	Vérification des interfaces
	ping www.test.tf (en IPV4) ping6 www.test.tf (en IPV6)	Test de connectivité IP
Passerelle par défaut	ip route netstat -r	Vérification de la table de routage (IPv4/IPv6)
	traceroute 176.31.61.170	Vérification des sauts
Serveur de nom	cat /etc/resolv.conf	Commandes utiles
	nslookup dig	Requêtes DNS

Configuration du réseau

➢Gestion du réseau Linux avec NetworkManager :

✓**NetworkManager est le démon (par défaut sous Centos/RHEL 7) qui gère les connexions réseau. Il n'empêche pas l'usage des fichiers de configuration des interfaces.**

✓**En Debian/Ubuntu, il sera peut-être nécessaire de l'installer. Aussi, il sera nécessaire de supprimer les entrées des interfaces à gérer par NetworkManager dans le fichier /etc/network/interfaces à l'aide de cette**

```
apt-get install network-manager
systemctl stop networking
systemctl disable networking
systemctl enable NetworkManager
systemctl start NetworkManager
```

```
systemctl status NetworkManager
```

du

Pour plus d'informations vous pouvez consulter ce lien
https://help.ubuntu.com/community/NetworkManager

✓**L'outil NetworkManager qui se gère directement avec systemctl, il est accompagné de plusieurs outils de diagnostic et de configuration :**

a.nm-connection-editor et gnome-control-center network sont les outils graphiques de configuration du réseau.

b.nmtui est l'outil graphique dans un terminal texte

c.nmcli est l'outil en ligne de commande.

Administration sécurisée

Sécurité locale

➢ Utilisateurs et groupes Linux :

❖ Commande su

- **su (substitute user ou switch user) est une commande Unix permettant d'exécuter un interpréteur de commandes en changeant d'identifiant de GID et de UID. « GID=UID=0 ⇔ sont les identifiants de root »**
- **Cette commande est surtout utilisée pour obtenir les privilèges d'administration à partir d'une session d'utilisateur normal, c'est-à-dire, non privilégiée.**

su « Nom_utilisateur » ou bien su - « Nom_utilisateur »

❖ Commande sudo

- **sudo (abréviation de substitute user do, en anglais : «exécuter en se substituant à l'utilisateur») est une commande qui permet à l'administrateur système d'accorder à certains utilisateurs (ou groupes d'utilisateurs) la possibilité de lancer une commande en tant que root**
  - **Pour ajouter un utilisateur au système en tant que non-root :**

```
sudo useradd zozo
```

Sécurité locale

➢ Utilisateurs et groupes Linux :

❖ Utilisateurs

- Toute entité (personne physique ou programme particulier) interagit avec un système UNIX doit s'authentifier sur cet ordinateur par un utilisateur ou "user". Ceci permet d'identifier un acteur sur un système UNIX. Un utilisateur est reconnu par un nom unique et un numéro unique.
- Sur tout système UNIX, il y a un super-utilisateur, généralement appelé root, qui a tous les pouvoirs sur le système. Il peut accéder librement à toutes les ressources de l'ordinateur, y compris à la place d'un autre utilisateur, c'est-à-dire sous son identité. => l'administrateur système possède le mot de passe root.

❖ Utilisateurs : fichier /etc/passwd

- On peut créer un utilisateur via le fichier /etc/passwd, par exemple on ajoute un utilisateur "user1":

```
echo "user1:x:2000:2000:user1:/home/user1:/bin/bash" >> /etc/passwd
```

- Mais faut-il encore :

  - créer le groupe correspondant, créer le répertoire utilisateurs, y donner les droits
  - vérifier la validité des UID et GID,

Sécurité locale

➢ Utilisateurs et groupes Linux :

❖ Mots de passe : fichier /etc/shadow

Le mot de passe est écrit dans le fichier /etc/shadow avec ses paramètres :

- ✓ **Nom de connexion de l'utilisateur (« login »)**
- ✓ **Mot de passe chiffré : 1 (MD5),2 (Blowfish), 5 (SHA-256), 6 (SHA-512)**
- ✓ **Date du dernier changement de mot de passe**
- ✓ **Age minimum du mot de passe**
- ✓ **Age maximum du mot de passe**
- ✓ **Période d'avertissement d'expiration du mot de passe**
- ✓ **Période d'inactivité du mot de passe**
- ✓ **Date de fin de validité du compte**
- ✓ **Champ réservé**

Sécurité locale

➢Utilisateurs et groupes Linux :

❖Groupes

- ❑Un utilisateur UNIX appartient à un ou plusieurs groupes.
- ❑Les groupes servent à rassembler des utilisateurs afin de leur attribuer des droits communs.
- ❑Le groupe principal est le groupe initial de l'utilisateur.
- ❑L'utilisateur peut appartenir à des groupes secondaires.

❖Fichiers /etc/group et /etc/gshadow

- ❑Les fichiers /etc/group et /etc/gshadow définissent les groupes.
- ❑Le fichier /etc/group comporte 4 champs séparés par ":" :

1.nom du groupe

2.mot de passe du groupe (ou x si le fichier gshadow existe)

3.le GID

4.liste des membres séparés par une virgule

Sécurité locale

➢ Utilisateurs et groupes Linux :

❖ Appartenance à un groupe

- On peut vérifier l'identification et l'appartenance d'user aux groupes via les commandes id et groups:

```
id
```

- Le résultat de cette commande est le suivant :

```
uid=1000(francois) gid=1000(francois) groupes=1000(francois),10(wheel) contexte=unconfined_u:unconfin
```

- La commande :

```
groups
```

- Cette commande retourne ce résultat :

```
francois wheel
```

Sécurité locale

➢ Opérations sur les utilisateurs et les groupes :

❖ Création d'un utilisateur

- On utilise la commande « useradd » pour créer les nouveaux comptes utilisateurs avec cette syntaxe :

  Useradd [option] identifiant

- En tant que root ou avec sudo, ajouter par exemple l'utilisateur tintin

```
useradd tintin
```

❖ Définir un mot de passe

- C'est la commande passwd qui met à jour le mot de passe de l'utilisateur :

```
passwd tintin
```

❖ Ajouter un groupe

- On peut ajouter des groupes facilement avec

```
groupadd marketing
```

- On peut ajouter un utilisateur milou à un groupe avec

```
gpasswd -a milou marketing
```

- On peut retirer un utilisateur milou d'un groupe avec

```
gpasswd -d milou marketing
```

Sécurité locale

➢ Opérations sur les utilisateurs et les groupes :

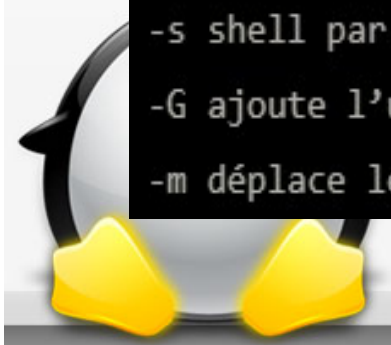
❖ Modifier les paramètres utilisateur

- On change les paramètres des groupes avec la commande **usermod.** Par exemple :

```
usermod -d /home/francois -a -G francois milou
```

- Les options de **usermod** sont :

```
-d répertoire utilisateur
-g définit le GID principal
-l identifiant utilisateur
-u UID utilisateur
-s shell par défaut
-G ajoute l'utilisateur à des groupes secondaires
-m déplace le contenu du répertoire personnel vers le nouvel emplacement
```



Sécurité locale

➢ Opérations sur les utilisateurs et les groupes :

❖ Modifier les paramètres d'un groupe

- C'est le programme **groupmod** qui permet de changer les paramètres d'un groupe. Ayant les options suivantes :

```
-g GID
-n nom du groupe
```

❖ Verrouiller un compte

- Pour verrouiller un compte d'utilisateur, voici les commandes suivantes :
  - ✓ pour verrouiller **passwd -l** ou **usermod -L**
  - ✓ pour déverrouiller **passwd -u** ou **usermod -U**
- De plus, il est également possible de supprimer le mot de passe avec **passwd -d**.
- On peut supprimer un compte utilisateur avec la commande **userdel.** Pour s'assurer de la suppression du répertoire utilisateur, utilisez l'option **-r**

```
userdel -r tintin
```

Sécurité locale

➢ Access control lists (ACLs) Linux :

- ❑ **Les Access Control Lists (ACLs) permettent de définir des permissions différentes pour un ou plusieurs utilisateurs / groupes sur un fichier / répertoire.**
- ❑ **A une époque, il fallait adapter le noyau et le FS au support des ACLs.**

❖ Visualiser les permissions ACLs

- ❑ **Créer un dossier /opt/partage et visualiser les permissions :**

```
mkdir /opt/partage
ls -ld /opt/partage
```

```
drwxr-xr-x. 2 root root 6 23 fév 20:16 /opt/partage
```

- ❑ **Visualiser les permissions ACLs du dossier par la commande getfacl :**

```
getfacl : suppression du premier « / » des noms de chemins absolus
# file: opt/partage
# owner: root
# group: root
user::rwx
group::r-x
other::r-x
```

Sécurité locale

➢ Access control lists (ACLs) Linux :

❖ Ajouter les ACLs à un répertoire :

- **Pour ajouter les ACLs à un répertoire c'est via la commande setfacl :**

```
setfacl -m g:omega:rx /opt/partage
setfacl -m u:alfa:rwx /opt/partage
getfacl /opt/partage
```

```
getfacl : suppression du premier « / » des noms de chemins absolus
# file: opt/partage
# owner: root
# group: root
user::rwx
user:alfa:rwx
group::r-x
group:omega:r-x
mask::rwx
other::r-x
```

Sécurité locale

➢Access control lists (ACLs) Linux :

❖ACLs par défaut

❑**Les ACLs par défaut permettent de donner des permissions ACL en héritage pour tout sous-répertoire ou fichier créé dans un répertoire. Toutefois, ces ACLs par défaut ne s'appliquent pas aux objets déjà présents dans le répertoire. Dans la configuration d'un partage avec des accès multiples, il sera donc nécessaire de procéder en deux étapes :**

1.Modifier l'ACL des fichiers existants

```
setfacl -R -m u:alfa:rx /opt/partage
```

2.Appliquer un ACL par défaut

```
setfacl -m d:u:alfa:rx /opt/partage
```

3.Visualiser l'ACL

```
getfacl /opt/partage
```

```
getfacl : suppression du premier « / » des noms de chemins absolus
# file: opt/partage
# owner: root
# group: root
user::rwx
user:alfa:r-x
group::r-x
group:omega:r-x
mask::r-x
other::r-x
default:user::rwx
default:user:alfa:r-x
default:group::r-x
default:mask::r-x
default:other::r-x
```

Sécurité locale

➢ Access control lists (ACLs) Linux :

❖ ACLs par défaut

- ❑ **Il est intéressant d'utiliser les ACLs par défaut pour définir les droits des autres (other) sur les fichiers nouvellement créés.**
- ❑ **Par exemple pour empêcher tous les autres en termes de permissions pour tout nouveau fichier ou sous-répertoire créé, on fait les commandes suivantes :**

```
setfacl -m d:o::- /opt/partage
getfacl /opt/partage
```

```
getfacl : suppression du premier « / » des noms de chemins absolus
# file: opt/partage
# owner: root
# group: root
user::rwx
user:alpha:r-x
group::r-x
group:omega:r-x
mask::r-x
other::r-x
default:user::rwx
default:user:alpha:r-x
default:group::r-x
default:mask::r-x
default:other::---
```

Pour restaurer les ACLs on effectue la commande suivante

```
setfacl --restore=acls
```

Secure Shell

➢ SSH :

- ❑ **Secure Shell (SSH) est un protocole qui permet de sécuriser les communications de données entre les ordinateurs connectés au réseau.**
- ❑ **Il permet d'assurer la confidentialité, l'intégrité, l'authentification et l'autorisation des données dans des tunnels chiffrés. Il utilise TCP habituellement sur le port 22, mais il peut en utiliser d'autres simultanément. On peut l'utiliser comme console distante à la manière de Telnet, RSH ou Rlogin.**
- ❑ **Il supporte les authentifications centralisées (PAM), locale avec mot de passe ou sans échange de mot de passe (par le biais d'échange de clés).**
- ❑ **On peut transférer des sessions X graphiques dans un tunnel SSH.**
- ❑ **On peut y transférer des ports et utiliser le service comme proxy ou comme solution VPN, de manière distante ou locale. Y compris aussi les sous-protocoles SCP et SFTP offrent des services de transfert de fichiers.**
- ❑ **En terme de cible d'attaque, le port est très sollicité par les robots qui scannent les réseaux publics en quête de configurations faibles, nulles, négligées ou exploitables. Il peut arriver qu'un port SSH exposé publiquement soit l'objet de tentatives de Déni de Service (DoS) ou de connexions Brute Force qui rendent le service inaccessible. => D'où l'utilisation du OpenSSH**

Secure Shell

➢ Installation, configuration, connexion OpenSSH :

- **OpenSSH est une version libre de la famille d'outils du protocole Secure Shell (SSH) pour le contrôle à distance ou le transfert des fichiers entre les ordinateurs. Les outils traditionnels utilisés pour accomplir ces fonctions tels que telnet ou rcp ne sont pas sécurisés et transmettent le mot de passe utilisateur en clair lors de leurs utilisations. OpenSSH fournit un démon de serveur et des outils pour les clients afin de sécuriser le contrôle à distance chiffré et les opérations de transfert de fichiers, remplaçant ainsi les anciens outils.**

- **Le serveur OpenSSH : sshd, attend en permanence des connexions depuis des clients. Quand une requête de connexion a lieu, sshd établit la connexion correcte en fonction du type de client. Par exemple, si un client se connecte avec le client ssh, le serveur OpenSSH va établir une connexion sécurisée après une authentification. Si un client se connecte avec scp, le serveur OpenSSH va commencer un transfert de fichier sécurisé entre le serveur et le client après une authentification. OpenSSH peut utiliser de nombreuses méthodes d'authentification, par exemple un mot de passe, une clé publique.**

Secure Shell

➢ Installation, configuration, connexion OpenSSH :

❖ Installation

- L'installation des applications client et serveur d'OpenSSH est simple. Pour installer les applications clientes d'OpenSSH sur votre système Ubuntu, tapez cette commande dans un terminal :

  sudo apt install openssh-client

- Pour installer le serveur OpenSSH et les fichiers nécessaires, utilisez cette commande dans un terminal :

  sudo apt install openssh-server

❖ Configuration

- Vous pouvez configurer le comportement par défaut du serveur OpenSSH, sshd, en modifiant le fichier /etc/ssh/sshd_config.
- Avant de modifier le fichier de configuration, vous devriez faire une copie du fichier original et le protéger en écriture de façon à conserver les paramètres d'origine en référence et à pouvoir les réutiliser en cas de besoin.

  sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.original

  sudo chmod a-w /etc/ssh/sshd_config.original

Secure Shell

➢Installation, configuration, connexion OpenSSH :

❖Configuration

❑**Voici des exemples de configuration que vous pouvez changer :**

1.Pour que OpenSSH écoute sur le port TCP 2222 au lieu du port par défaut 22, changez la directive Port comme ceci : **Port 2222**

2.Pour que sshd accepte les informations de connexion basées sur une clef publique, il suffit d'ajouter ou de modifier la ligne : **PubkeyAuthentication yes**

3.Si la ligne est déjà présente, alors assurez-vous qu'elle n'est pas commentée.

4.Pour que le serveur OpenSSH affiche le contenu du fichier /etc/issue.net comme une invite avant l'affichage de l'écran de connexion, il suffit d'ajouter ou de modifier la ligne : Banner /etc/issue.net dans le fichier /etc/ssh/sshd_config.

5.Après avoir modifié le fichier /etc/ssh/sshd_config, enregistrez-le et redémarrez le service sshd afin de prendre en compte les changements.

6.Pour cela, saisissez la commande suivante dans un terminal : sudo systemctl restart sshd.service

Secure Shell

➢Installation, configuration, connexion OpenSSH :

❖Authentification par clé avec OpenSSH

❑**L'authentification par clé fonctionne grâce à 3 composants :**

1.Une clé publique : elle sera exportée sur chaque hôte sur lequel on souhaite pouvoir se connecter.

2.Une clé privée : elle permet de prouver son identité aux serveurs.

3.Une passphrase : optionnelle, elle permet de sécuriser la clé privée (notons la subtilité, passphrase et pas password… donc « phrase de passe » et non pas « mot de passe »).

❑**La sécurité est vraiment accrue car la passphrase seule ne sert à rien sans la clé privée, et vice-versa.**

a. Création de la paire de clés

❑**Les clés SSH permettent l'authentification entre deux hôtes sans avoir besoin de mot de passe. L'authentification par clé SSH utilise deux clés, une clé privée et une clé publique. Ceci est fait par cette commande :**

ssh-keygen -t rsa

Secure Shell

➢ Installation, configuration, connexion OpenSSH :

❖ Authentification par clé avec OpenSSH

a. Création de la paire de clés

- **Par défaut:**
  - ✓ **La clé publique est sauvegardée dans le fichier ~/.ssh/id_rsa.pub,**
  - ✓ **alors que la clé privée est dans ~/.ssh/id_rsa.**
- **Quatre méthodes de transmission de la clé à partir d'une station distante sont proposées ici. Soit par :**

▪ **Méthode traditionnelle :**

- **Copiez maintenant le fichier id_rsa.pub sur l'hôte distant et ajoutez le à ~/.ssh/authorized_keys en entrant:**

ssh-copy-id identifiant@hôte

- **Pour finir, vérifiez les permissions du fichier authorized_keys. Seul l'utilisateur authentifié doit avoir les droits en lecture et écriture. Si les permissions sont incorrectes, changez-les en tapant :**

chmod 600 .ssh/authorized_keys

Secure Shell

➢ Installation, configuration, connexion OpenSSH :

❖ Authentification par clé avec OpenSSH

a. Création de la paire de clés

- **Via une console SSH :**

```
cat ~/.ssh/id_rsa.pub | ssh user@ip_machine "cat - >> ~/.ssh/authorized_keys"
```

- **Via le protocole de transfert SCP :**

```
scp ~/.ssh/id_rsa.pub user@ip_machine:/tmp
ssh user@ip_machine
cat /tmp/id_rsa.pub >> ~/.ssh/authorized_keys
rm /tmp/id_rsa.pub
```

- **Via le binaire ssh-copy-id :**

```
ssh-copy-id -i ~/.ssh/id_rsa.pub user@ip_machine
```

Secure Shell

➢ Shell distant :

❖ Le client OpenSSH

- **On obtient un shell distant en utilisant la commande ssh utilisateur@machine. A la première connexion, on reconnaîtra l'hôte de destination comme étant valide. L'emprunte de sa clé est enregistrée dans le fichier ~/.ssh/known_hosts.**

```
ssh user@127.0.0.1 -p 22
```

```
The authenticity of host '127.0.0.1 (127.0.0.1)' can't be established.
ECDSA key fingerprint is bf:ab:65:84:a3:2f:0b:f9:2c:68:88:c9:a8:24:3f:64.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '127.0.0.1' (ECDSA) to the list of known hosts.
user@127.0.0.1's password:
Last login: Tue Sep 27 17:52:26 2016 from 172.16.98.1
$ exit
déconnexion
Connection to 127.0.0.1 closed.
```

Pour exécuter une commande à distance :

```
ssh localhost id
```

Voici le résultat obtenu :

```
user@localhost's password:
uid=1000(user) gid=1000(user) groupes=1000(user),10(wheel) contexte=unconfined_u:unconfined_r:unconfi
```

Secure Shell

➢ Transfert de fichiers SCP et SFTP :

❖ Transfert de fichiers SCP

- **SCP est la transposition de la commande cp à travers SSH, avec des arguments, une source et une destination. On désigne la ressource distante origine ou destination par user@machine:/path. Par exemple :**

```
scp /dossier/fichier user@machine:~
```

```
scp user@machine:~/dossier/fichier .
```

```
scp -R /dossier user@machine:~
```

- **Pour envoyer les fichiers du répertoire local psionic, vers le répertoire tmp qui est dans /home/user de la machine ENSAH :**

scp ~/psionic/* user@ENSAH:/home/user/tmp

- **Attention: c'est l'option -P numero_port qui permet de définir un port SSH avec les binaires clients scp et sftp.**

Secure Shell

➢ Transfert de fichiers SCP et SFTP :

❖ Transfert de fichiers SFTP

- ❑ **SFTP s'utilise comme un client FTP en mode sécurisé**

```
# sftp user@localhost
user@localhost's password:
Connected to localhost.
sftp> pwd
Remote working directory: /home/user
sftp> quit
```

Gestion sécurisée

➢ Planification des taches :

❖ Commande at

- **at est une commande Unix qui permet de programmer des commandes à n'exécuter qu'une fois à un moment donné. La commande enregistrée hérite de l'environnement courant utilisé au moment de sa définition. Par exemple, pour une exécution de la commande à 05:45 :**

```
$ echo "touch file.txt" | at 0545
```

- **Les options de la commande at sont les suivantes :**
  - ✓ **at -l ou atq : affiche la liste des jobs introduits par la commande « at ».**
  - ✓ **at -r JOB ou atrm JOB : efface le job identifié par son numéro de job.**
  - ✓ **at : sans paramètre, donne la ligne « Garbled time ».**

Gestion sécurisée

➢ Planification des taches :

❖ Création d'une tache planifiée par la commande at

- **at est une commande Unix qui permet de programmer des commandes à n'exécuter qu'une fois à un moment donné. La commande enregistrée hérite de l'environnement courant utilisé au moment de sa définition. Par exemple, pour une exécution de la commande à 05:45 :**

```
$ echo "touch file.txt" | at 0545
```

- **Sa syntaxe est la suivante : « at heure date »** → `root@ipower:~$ at 20:00 10/21/05` **une fois que vous tapez cette commande un prompt apparait dont lequel on pourra planifier notre tache « lancement du firefox » qui sera fait à 20:00**

```
warning: commands will be executed using /bin/sh
at> firefox
at>
```

Pour sortir du prompt appuyez sur CONTROLE+D.

Gestion sécurisée

➢ Planification des taches :

❖ Création d'une tache planifiée par la commande at

- La commande suivante permet d'exécuter une tâche dans 2 jours à minuit.

  « root@ENSAH:~$ at 00:00 +2 days »

- Celle-ci effectuera une tâche dans 2 heures à partir de cet instant.

  « root@ENSAH:~$ at now +2 hour »

- Les options de la commande at sont les suivantes :
  - ✓ at -l ou atq : affiche la liste des jobs qui sont en cours par la commande « at ».
  - ✓ at -r JOB ou atrm JOB : efface le job identifié par son numéro de job.
  - ✓ at : sans paramètre, donne la ligne « Garbled time ».
- La commande qui vous permet de déterminer quels sont les jobs ou travaux en cours est atq

  « root@ENSAH:~$ atq »

  Le résultat est affiché comme suit :

  4 2005-10-23 00:00 a ENSAH

  5 2005-10-21 21:52 a ENSAH

Pour annuler une tâche, utilisez la commande atrm suivi du numéro de tâche

« root@ENSAH:~$ atrm 5 »

« root@ENSAH:~$ atq »

4 2005-10-23 00:00 a ENSAH

Gestion sécurisée

➢ Planification des taches :

❖ Création d'une tache planifiée par la commande CRON et CRONTAB

- **Crontab est un outil qui permet de lancer des applications de façon régulière, pratique pour un serveur pour y lancer des scripts de sauvegardes….etc.**
- **Bien que par défaut, il soit souvent installé, mais voici les commandes pour l'installer sur les différentes distributions**

Commande d'installation	Distribution
apt-get install cron	Sous Debian
yum install cronie	Sous Fedora/Centos
emerge -av sys-process/cronie	Sous Gentoo

Gestion sécurisée

➢ Planification des taches :

❖ Création d'une tache planifiée par la commande CRON et CRONTAB

- **Configuration de crontab :** Pour être autorisé à utiliser la commande **crontab**, il faut que l'utilisateur soit présent dans le groupe **cron**. Dans ce cas, Les fichiers **/etc/cron.allow** et **/etc/cron.deny** permettent de définir les droits d'utilisation sur **crontab.**
- Si le fichier **/etc/cron.allow** existe, alors vous devez être présent dans ce fichier pour être autorisé à utiliser cette commande. Si le fichier **/etc/cron.allow** n'existe pas mais que **/etc/cron.deny** existe, alors vous ne devez pas être mentionné dans le fichier **/etc/cron.deny** afin de pouvoir utiliser cette commande.
- **Utilisation de la commande crontab :**

Commande de gestion CRON	Signification
crontab –l	lister les tâches planifiées de l'utilisateur courant
crontab -e	éditer les actions du fichier crontab
crontab -r	supprimer toutes les actions du fichier crontab
sudo crontab -e -u nom_utilisateur	pour modifier les taches d'un autre utilisateur

Gestion sécurisée

➢ Planification des taches :

❖ La syntaxe de la commande CRONTAB

❑ Une tâche planifiée dans un fichier de Cron est composée de 3 données différentes :

1. Sa période de répétition définie par 5 données différentes :
   - ✓Les minutes ;
   - ✓Les heures ;
   - ✓Les jours dans le mois ;
   - ✓Les mois ;
   - ✓Les jours de la semaine ;
2. L'utilisateur système sous lequel la tâche sera réalisée ;
3. La commande à réaliser ;

Gestion sécurisée

➢ Planification des taches :

❖ La syntaxe de la commande CRONTAB

- ❑ Voici la syntaxe à respecter d'un crontab
- ❑ Voici l'exemple suivant :

```
# 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-name  command-to-be-executed
```

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.**

Gestion sécurisée

➢ Planification des taches :

❖ Les exemples de la commande CRONTAB

- ❑ Voici les exemples suivants : pour Exécuter 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 toutes les heures :

00 */1 * * * /usr/sbin/ntpdate fr.pool.ntp.org

- ✓ 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 !"

Gestion sécurisée

➢ Planification des taches :

❖ Périodicité de la commande CRONTAB

- La périodicité est définie en séparant les 5 unités temporelles (minutes/heures/jours dans le mois/mois/jours de la semaine).

Raccourcis	Description	Équivalent
@reboot	Au démarrage du système	Aucun
@yearly	Tous les ans	0 0 1 1 *
@annually	Tous les ans	0 0 1 1 *
@monthly	Tous les mois	0 0 1 * *
@weekly	Toutes les semaines	0 0 * * 0
@daily	Tous les jours	0 0 * * *
@midnight	Tous les jours	0 0 * * *
@hourly	Toutes les heures	0 * * * *

Gestion sécurisée

➢ Localisation et synchronisation :

❖ Syntaxe de base de la commande rsync

- La commande **rsync** Linux permet de transférer et de synchroniser efficacement des fichiers ou des répertoires entre une machine locale, un autre hôte, un **shell distant**, ou toute autre correspondance de ceux-ci.
- La syntaxe de base de **rsync** fonctionne comme suit :

rsync [modificateurs optionnels] [SRC] [DEST]

⇒ Il existe plusieurs façons dont vous pouvez utiliser **rsync** Linux. Dans cet exemple, [modificateurs optionnels] indique les actions à effectuer, **[SRC]** est le répertoire source, et **[DEST]** est le répertoire ou la machine de destination.

❖ Syntaxe de base pour Shell distant

⇒ Lorsque vous utilisez un shell distant, tel que SSH ou RSH, la syntaxe de **rsync** sera légèrement différente.

⇒ Pour accéder au shell distant **(PULL),** utilisez la commande **rsync** :

rsync [modificateurs optionnels] [USER@]HOST:SRC [DEST]

Gestion sécurisée

➢ Localisation et synchronisation :

⇒ Pour accéder au shell distant **(PUSH),** utilisez la commande rsync :

rsync [modificateurs optionnels] SRC [USER@]HOST:[DEST]

❖ Comment vérifier la version Rsync

❑ On trouve **rsync** préinstallé avec de nombreuses distributions Linux. Vous pouvez vérifier si **rsync** est installé sur votre machine en exécutant la commande suivante :

rsync -version

❖ Installation de Rsync

❑ Si votre machine n'a pas **rsync** préinstallé, vous pouvez le faire manuellement en une minute ! Sur les distributions basées sur Debian comme **Ubuntu,** vous pouvez le faire en utilisant la commande suivante :

apt-get install rsync

❖ Localisation

dpkg-reconfigure tzdata

Gestion sécurisée

➢ Journalisation Systemd

❖ Commande journalctl

- Configuration de journalctl

  …

- Droits d'accès. Les utilisateurs peuvent seulement voir leurs journaux. Pour voir tous journaux du système, l'utilisateur doit faire partie du groupe adm.

usermod -a -G adm francois

- **Pour Consulter le journal, on exécute la commande suivante :**

journalctl

- **Afficher les 5 dernières lignes du journal.**

journalctl -n 5

- **Affichage en temps réel.**

journalctl -f

- **Affichage des messages au démarrage.**

journalctl -b

Firewall

➢ Les objectifs d'un pare-feu :

- ❑ **Dans un système d'information, les politiques de filtrage et de contrôle du trafic sont placées sur un matériel ou un logiciel intermédiaire communément appelé pare-feu (firewall).**
- ❑ **Cet élément du réseau a pour fonction d'examiner, filtrer le trafic qui le traverse,**
- ❑ **Le pare-feu limite le taux de paquets et de connexions actives. Il reconnaît les flux applicatifs et contrôle les flux du réseau TCP/IP.**
- ❑ **Pare-feu a pour objectifs de répondre aux menaces et attaques suivantes, de manière non-exhaustive :**

  - ✓ **La manipulation d'informations**
  - ✓ **Les attaques de déni de service (DoS/DDoS)**
  - ✓ **Les attaques par code malicieux**
  - ✓ **La fuite d'information**
  - ✓ **Les accès non-autorisé (en vue d'élévation de privilège)**
  - ✓ **Les attaques de reconnaissance, d'homme du milieu, l'exploitation de TCP/IP**

Firewall

➢ Fonctionnement d'un pare-feu :

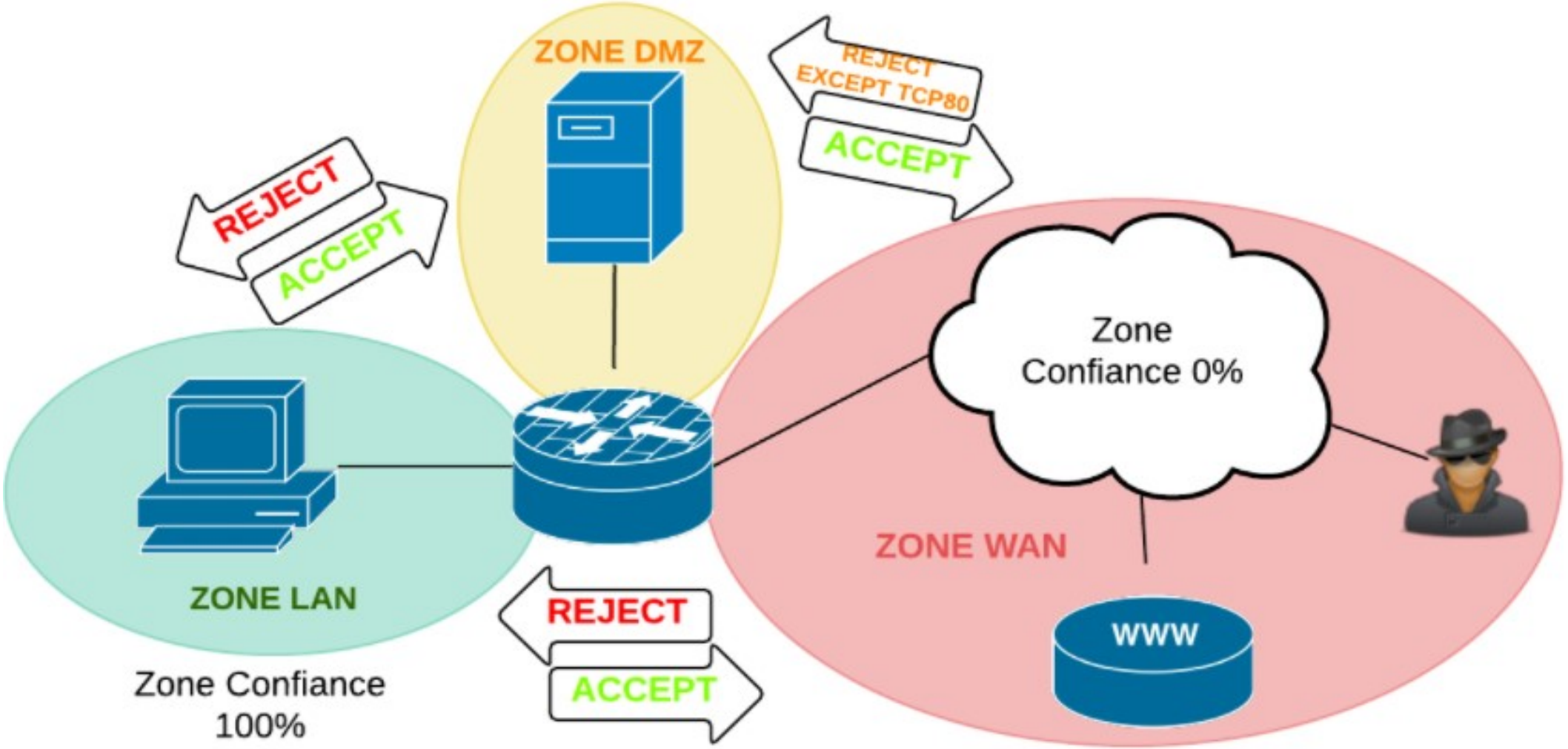
- **Il a pour principale tâche de contrôler le trafic entre différentes zones de confiance, en filtrant les flux de données qui y transitent.**
- **Généralement, les zones de confiance incluent l'Internet (une zone dont la confiance est nulle) et au moins un réseau interne (une zone dont la confiance est plus importante).**

- **Le but est de fournir une connectivité contrôlée et maîtrisée entre des zones de différents niveaux de confiance, grâce à l'application de la politique de sécurité et d'un modèle de connexion basé sur le principe du moindre privilège.**
- **Un pare-feu fait souvent office de routeur et permet ainsi d'isoler le réseau en plusieurs zones de sécurité appelées zones démilitarisées ou DMZ. Ces zones sont séparées suivant le niveau de confiance qu'on leur porte.**

Firewall

➢ Zone de confiance sur un pare-feu :

❑ Organisation du réseau en zones



Firewall

➢ Niveau de confiance :

- ❑ **Le niveau de confiance est la certitude que les utilisateurs vont respecter les politiques de sécurité de l'organisation.**

- ❑ **Ces politiques de sécurité sont édictées dans un document écrit de manière générale. Ces recommandations touchent tous les éléments de sécurité de l'organisation et sont traduites particulièrement sur les pare-feu en différentes règles de filtrage.**

- ❑ **On notera que le pare-feu n'examine que le trafic qui le traverse et ne protège en rien des attaques internes, notamment sur le LAN.**

Firewall

➢ Filtrage :

- La configuration d'un pare-feu consiste la plupart du temps en un ensemble de règles qui déterminent une action de rejet ou d'autorisation du trafic qui passe les interfaces du pare-feu en fonction de certains critères tels que :

- ✓ **L'origine et la destination du trafic,**
- ✓ **Des informations d'un protocole de couche 3 (IPv4, IPv6, ARP, etc.),**
- ✓ **Des informations d'un protocole de couche 4 (ICMP, TCP, UDP, ESP, AH, etc.)**
- ✓ **Des informations d'un protocole applicatif (HTTP, SMTP, DNS, etc.).**

Firewall

➢ Règles :

- Chaque règle est examinée selon son ordonnancement :

- **Si le trafic ne correspond pas à la première règle, la seconde règle est évaluée et ainsi de suite.**
- **Lorsqu'il y a correspondance entre les critères de la règle et le trafic, l'action définie est exécutée et les règles suivantes ne sont pas examinées.**
- **La terminologie des actions usuelles peuvent être accept, permit, deny, block, reject, drop, ou similaires.**
- **En général, un ensemble de règles se termine par le refus de tout trafic, soit en dernier recours le refus du trafic qui traverse le pare-feu. Ce comportement habituellement défini par défaut ou de manière implicite refuse tout trafic pour lequel il n'y avait pas de correspondance dans les règles précédentes.**

Firewall

➢ Pare-feu personnel Debian/Ubuntu :

- **Uncomplicated Firewall (ufw) pour l'installer on utilise cette commande :**

```
apt-get install ufw
```

- **Pour visualiser son statut on utilise cette commande :**

```
ufw status
Status: inactive
```

- **Pour le stopper :**

```
ufw disable
Firewall stopped and disabled on system startup
```

- **Pour l'activer :**

```
ufw enable
Firewall is active and enabled on system startup
```

- **Pour voir l'ensemble des règles :**

```
ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing)
New profiles: skip
```

```
ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing)
New profiles: skip

To                         Action      From
--                         ------      ----
22                         ALLOW IN    Anywhere
22                         ALLOW IN    Anywhere (v6)
```

Firewall

➢ Pare-feu personnel Debian/Ubuntu :

- **Autoriser une connexion entrante :**

```
ufw allow [règle]
```

- **Refuser une connexion entrante :**

```
ufw deny [règle]
```

- **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 "ou deny" [règle]
```

- **Supprimer simplement une règle d'après son numéro :**

```
sudo ufw delete [numéro]
```

Fin de Cours