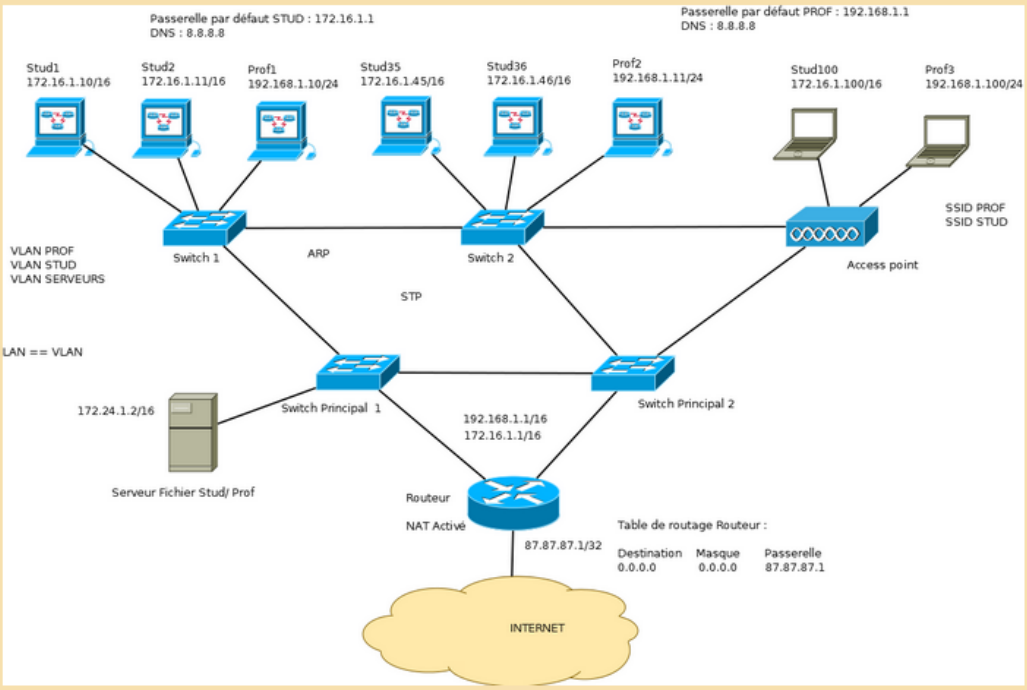
Infra – théorie – uniquement les points rouges

Réseaux et Infrastructure

- Connaître les différents composants matériels d'un réseau et leur rôle
- Comprendre quels éléments sont utilisés lors de l'envoi d'une requête réseau HTTP/HTTPS
- Savoir manipuler les différentes commandes de base liées au réseau
- Découper un réseau d'entreprise en sous-réseaux

Voici un exemple de réseau d'entreprise (très fortement inspiré de l'IPL). **Vous retrouverez dans ce schéma tous les composants nécessaires à un réseau d'entreprise aussi appelé LAN.** Certains sont sans doute déjà connu pour vous (IP,DNS, ...), d'autres non (VLAN, STP, ..). Les 2 premiers chapitres de ce syllabus vont les détailler et surtout expliquer leur rôle dans ce schéma.



Voici également une capture réseau réalisé avec le logiciel open source Wireshark. Celui-ci vous permet de mieux comprendre ce qui se passe dans un réseau par exemple lorsque vous faites une requête HTTP.
Attention les écoutes réseau sont encadrées par des textes de lois. Faire des écoutes réseau sans l'accord de l'entreprise, les personnes concernées est un délit.

Couche 1 : Matériel

Au niveau de la couche matérielle, différentes solutions existent pour implémenter son réseau physique. Cependant, une implémentation s'impose : **l'utilisation d'un réseau avec une topologie en étoile et des câbles Ethernet**. Ce réseau peut être renforcé avec de la fibre optique pour les liaisons les plus sollicitées. Un réseau sans fil accompagne la plupart du temps également cette architecture.

2.4 Couche 2 : Liaison de données

La couche 2 est généralement implémentée dans une infrastructure par une série de switchs(commutateur en français) qui concentrent les câbles Ethernet. Ces switchs, aussi appelés concentrateur, disposent d'un grand nombre de ports(24,48…) afin d'y connecter plusieurs machines. Au niveau de la couche 2, on parle de frames (trame en français). Celles-ci permettent à des PC présents sur le même sous-réseau de dialoguer. **L'identifiant d'une machine est la MAC address.** Une trame sera donc composée d'une MAC address source, d'une MAC address destination et du message. **Le composant principal de cette couche (switch) conserve une table des MAC address permettant d'associer chaque port du switch à une MAC address.** Cette table se met à jour continuellement en fonction de l'activité réseau. Si une machine ne communique plus pendant un laps de temps, sa MAC address est retirée de la table. Cette table permet au switch d'aiguiller les trames directement vers la machine de destination. Si le switch n'a pas connaissance dans sa table de la MAC address de destination, la trame est envoyée sur tous les ports.

2.4.1 VLAN

Les VLAN (Virtual Local Area Network) permettent de séparer certains ports d'un switch par rapport à d'autres. Chaque switch disposera alors d'une table des MAC address par VLAN. L'intérêt des VLAN est de pouvoir utiliser un switch pour connecter des machines appartenant à des réseaux différents toute en garantissant la sécurité. **Les ports se verront attribués un numéro de VLAN et les ports ayant un même numéro de VLAN pourront uniquement communiquer entre eux.**

Exemple : Imaginons une entreprise avec des employés et une direction. Nous décidons de créer un sous-réseau pour chacune des populations. Sans la technique des VLAN, je devrai utiliser 2 switchs(un switch pour les machines "direction" et un autre switch pour les machines "employés") tandis qu'avec les VLAN je pourrai utiliser un seul switch.

2.5 Couche 3 : Réseau

La couche réseau est implémentée dans une infrastructure par des routeurs. Ceux-ci feront transiter des paquets IP d'un réseau à un autre. L'identifiant d'une machine pour cette couche est une adresse IP. Une adresse IP est composée d'une adresse réseau et d'un masque. Le masque d'une adresse IP est noté soit sous le format A.B.C.D (Ex: 255.255.0.0) ou sous le format CIDR (Ex : /16). Actuellement l'adressage IP version 4 (IPv4) est encore largement utilisé même si l'adressage IPv6 commence à faire son apparition.

2.5.1 Adresses spéciales

Certaines adresses IP sont particulières. Vous connaissez peut-être déjà l'adresse 127.0.0.1 qui souvent utilisé pour la boucle locale(localhost). Il existe en fait des plages d'adresse réservées pour certains usages.

Plage	Usage
127.0.0.0/8	Boucle locale
10.0.0.0/8	Adresses privées
172.16.0.0/12	Adresses privées
192.168.0.0/16	Adresses privées
224.0.0.0/4	Multicast

Tableau à connaître

La boucle locale permet de créer un mini-réseau sur la machine permettant à l'utilisateur de tester/utiliser un service réseau sans être connecté (à un véritable réseau). Ceci permet notamment à un développeur d'utiliser un serveur Web sur sa machine de développement en l'absence de connexion réseau. **Les adresses privées sont réservées pour être utilisées dans les réseaux d'entreprises (LAN) et ne sont pas routables sur Internet** (Voir NAT). Vous devez utiliser ces adresses lorsque vous créez votre réseau d'entreprise. Les adresses multicast sont utilisées pour l'envoi de paquet en multicast (voir mode d'envoi ci-dessous).

2.5.2 Mode d'envoi

- Unicast : un à un
- Broadcast : un à tout son sous-réseau.
  - Attention les broadcasts ne passent pas les routeurs, on reste dans son sous-réseau.
- Multicast : un à un groupe
  - le protocole IGMP sera employé pour l'abonnement à un groupe
  - le matériel réseau s'occupera de dupliquer les paquets à destination

2.5.3 Commandes réseau

Linux	Windows	Explication
ifconfig / ip addr	ipconfig	Configuration / Consultation des informations IP d'une machine
route	route	Configuration / Consultation des informations de routage d'une machine
ping	ping	Tester la connectivité IP d'une machine
traceroute	tracert	Voir le chemin parcouru par un paquet IP

Tableau à connaître

2.5.4 ARP

Il permet à une machine de demander l'adresse MAC d'une autre machine sur base de son adresse IP.

2.5.5 Découpage en sous-réseaux

Vous devez être capable de découper un réseau en sous-réseau. Ceci peut être demandé dans la partie pratique de l'examen. La tâche d'un administrateur système sera de découper de manière intelligente son réseau. Ce découpage permet d'isoler les réseaux pour des raisons de sécurité et d'efficacité. Il devra calculer une adresse réseau et un masque de sous-réseau. Celui-ci permettra de donner le nombre de machines maximales que le sous-réseau pourra accueillir.

Exemple : un réseau employés composé de 25 machines

1. 25 machines → recherche de la puissance de 2 supérieure à 25 → 2^5
2. Choix d'une adresse réseau parmi les plages d'adresses privées : 192.168.5.0 par ex.
3. Calcul du masque : 32 bits – 5 bits (puissance de 2 calculée précédemment) → /27
4. Résultat : 192.168.5.0/27

Une plage d'adresses pour un réseau débutera par l'adresse du réseau et se terminera par l'adresse de diffusion(broadcast). Entre ces 2 adresses, les autres adresses peuvent être utilisées par des machines.

Exemple : un réseau employés composé de 25 machines

1. Adresse réseau : 192.168.5.0/27
2. Adresse machine 1 : 192.168.5.1/27
3. Adresse machine 2 : 192.168.5.2/27
4. Adresse machine … : …
5. Adresse broadcast : 192.168.5.31/27

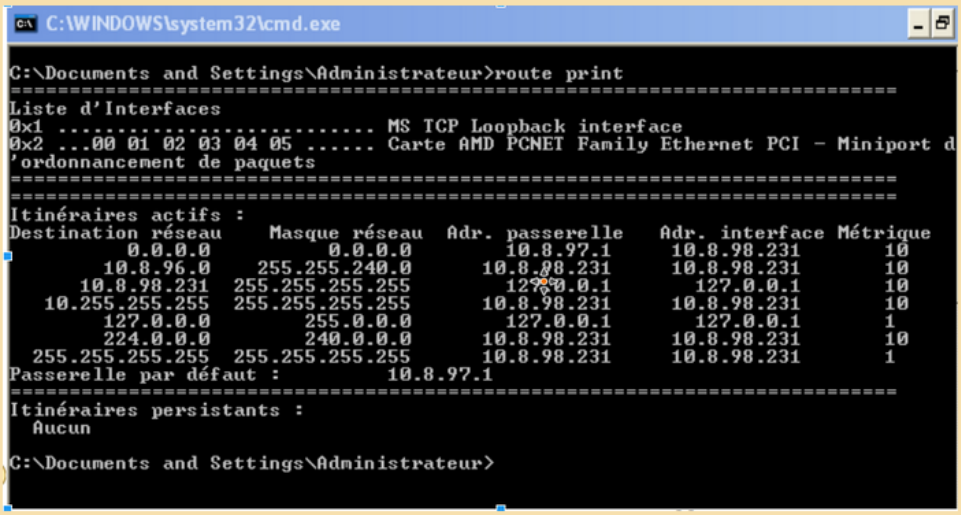
L'adresse de diffusion(broadcast) est la dernière adresse de la plage calculée. Tous les bits machines de cette adresse sont donc à 1.

Adresse réseau : 192.168.5.0/27 → 192.168.5.00011111 → 192.168.5.31/27 Astuce : une adresse réseau est toujours paire et une adresse de diffusion toujours impaire !

2.5.6 Table de routage

Le composant principal de cette couche 3, le routeur, maintient une table de routage. Vous pouvez imaginer par un routeur comme étant un carrefour routier et une table de routage comme étant les panneaux indicateurs présents au carrefour. **Une table de routage est un tableau qui précise pour une destination d'un sous-réseau (adresse réseau + masque) une passerelle (adresse IP d'une machine/routeur).**

Un exemple ci-dessous :



Pour chaque destination, la table de routage indique une passerelle et une métrique. À noter que les destinations sont indiquées par réseau (adresse IP + masque). Les entrées de cette table peuvent être ajoutées à la main ou de manière automatique. On parle dans le premier cas de routage statique et dans le second cas de routage dynamique. Quelques exemples de protocoles de routage dynamique : RIP, OSPF (utilisant l'algorithme de Dijkstra).

2.5.7 Passerelle par défaut

Vu qu'il est impossible pour chaque machine de constituer une table de routage avec tous les chemins possibles, **chaque machine possédera une passerelle par défaut (0.0.0.0/.0.0.0.0). Celle-ci sera utilisée lorsque'aucune règle plus précise ne pourra être utilisée dans la table de routage. Grâce à cette passerelle par défaut, il est également plus simple d'automatiser la connexion de machines clientes à son réseau. En effet, il suffira de fournir à chaque machine cliente une adresse IP, un masque et une passerelle par défaut et celle-ci pourra utiliser notre réseau. Cette automatisation pourra se faire via un serveur DHCP** (Voir DHCP).

2.6 Couche 4 : Transport

La couche 4 (Transport) va permettre une communication entre applications. Les applications respecteront le modèle d'architecture client-serveur. Cette couche 4 utilise la notion de port. **Une application sera donc identifiée via son port.** Typiquement, une application de type serveur se mettra en écoute sur un port tandis qu'une application cliente se connectera à ce serveur en précisant son adresse IP et port. Des ports par défaut ont été définis pour les applications les plus courantes :

Application	Port réservé
Web(HTTP)	80
SSH	22
HTTPS	443
DNS	53

Tableau à connaître

3 Services réseaux (DHCP-DNS-NAT)

3.1 Objectifs de ce chapitre

- **Savoir expliquer le rôle des 3 services réseaux de base**

3.2 Introduction

Pour qu'un réseau d'entreprise fonctionne correctement, il est nécessaire d'avoir au minimum 3 services à savoir :

1. **DNS** (Domain Name System)
2. **DHCP** (Dynamic Host Configuration Protocol)
3. **NAT** (Network Address Translation)

Un administrateur système doit donc veiller à ce que son réseau mette à disposition de ces clients ces 3 services.

3.3 DNS

Pour rappel, le système DNS permet de traduire un nom de domaine en une adresse IP et inversement. Ce système a été longuement décrit dans le cours d'IPP, nous ne détaillerons donc pas son fonctionnement ici. Par contre, nous allons nous attarder à l'usage d'un serveur DNS dans un réseau d'entreprise.

3.3.1 Configuration DNS

Un serveur DNS gère les enregistrements pour un nom de domaine aussi appelé zone DNS. Une zone DNS contient différents types d'enregistrements :

- **SOA** : Start of Authority (informations générales sur la zone DNS)
- **A** : Enregistrement d'un hôte (correspondance Nom → IP)
- **CNAME** : Alias
- **PTR** : Enregistrement d'une IP (correspondance (IP→ Nom)
- **MX** : Mail server (adresse IP du serveur mail de la zone)
- **NS** : Name Server (serveur DNS pour la zone)

Exemple d'une configuration d'un serveur DNS sous Linux :

```
$TTL    604800
@       IN      SOA     ns.monbeaurezo.be. emailadmin.monbeaurezo.be. (
                      2         ; Serial
                 604800         ; Refresh
                  86400         ; Retry
                2419200         ; Expire
                 604800 )       ; Negative Cache TTL

      @       IN      NS      ns.monbeaurezo.be.
      @       IN      A       127.0.0.1
      @       IN      AAAA    ::1
      ns      IN      A       192.168.1.1
      h1      IN      A       192.168.1.5
      aliasH1 IN      CNAME   h1
```

3.3.2 Résolution d'un nom

Il est très important de ne pas oublier que toute résolution de noms sur une machine commence par l'inspection du fichier hosts. Ce fichier est présent sous Linux à cet endroit /etc/hosts et sous Windows à cet endroit c:\Windows\System32\Drivers\hosts.

Exemple de fichier hosts sous Windows :

```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host

# localhost name resolution is handled within DNS itself.
#   127.0.0.1       localhost
#   ::1             localhost

127.0.0.1          localhost.vdsnb.com
127.0.0.1          siteHTML
127.0.0.1          sitePHP
127.0.0.1          siteJetty
127.0.0.1          contsitephpdb
```

Les administrateurs système utilisent abondamment ce fichier pour tester des services car cela évite l'installation d'un serveur DNS. En production, un serveur DNS sera bien évidemment employé.

3.4 DHCP

Pour qu'une machine cliente puisse se connecter à un réseau et surfer sur Internet, cette machine a besoin au minimum des informations suivantes :

- **se voir attribuer une adresse IP et un masque au sein de ce réseau**
- **obtenir une passerelle par défaut (pour aller sur Internet notamment)**
- **obtenir des serveurs DNS (pour traduire les noms en adresse IP)**

Un serveur DHCP permettra de répondre à ces besoins de manière automatique.

Les informations reçues par le serveur DHCP ne seront valables qu'un certain temps. On parle de bail. Le client pourra évidemment renouveler son bail.

La requête DHCP_DISCOVER se fait en broadcast. Les broadcast ne passent pas les routeurs. Donc un serveur DHCP devra être présent dans chaque sous-réseau ou celui-ci devra avoir une interface réseau dans chaque sous-réseau.

Un administrateur système installera plusieurs serveurs DHCP pour se prémunir de la panne d'un serveur. Deux serveurs DHCP ne pouvant pas distribuer les mêmes adresses IP, l'administrateur système devra répartir ces adresses de manière intelligente entre les 2 serveurs.

3.5 NAT

Le NAT est utilisé abondamment par les réseaux d'entreprise pour traduire les adresses IP privées de clients en adresses publiques routables sur Internet. *Pour rappel, les adresses IPv4 sont limitées à 4 milliards ce qui ne permet pas de couvrir la totalité du globe terrestre. Donc certaines plages d'adresses ont été réservées(adresses privées) pour pouvoir être attribuées dans les réseaux locaux des entreprises. Ceci est très bien, mais pour aller sur Internet, j'ai besoin d'une adresse publique autrement dit d'une adresse routable sur Internet. C'est ici qu'intervient le NAT. Vous allez comprendre comment la terre entière peut surfer sur Internet avec seulement 4 milliards d'adresses disponibles.*

3.5.1 Source NAT (SNAT)

Dans ce cas, l'adresse IP source du paquet IP est traduite, "natée".

Chaque machine cliente sera connectée au réseau de l'entreprise via une adresse privée et recevra une passerelle par défaut qui sera généralement le routeur qui vous permettra d'aller sur Internet. Ce routeur disposera d'une adresse IP publique reçue par votre fournisseur d'accès à Internet et il disposera également d'une adresse privée dans le réseau de l'entreprise. Nous activerons le NAT sur le routeur ce qui aura pour effet de traduire l'adresse IP source de tout paquet IP sortant sur Internet par l'adresse IP publique du routeur.

Le routeur va en fait identifier les machines clientes via un port choisi aléatoirement.

3.5.2 Port forwarding

Le NAT introduit une isolation du réseau de l'entreprise. En effet, il est impossible d'atteindre une machine du réseau de l'entreprise depuis l'extérieur. La seule machine que l'on sait atteindre est le routeur. Ceci est intéressant en termes de sécurité, mais nous devons quelquefois avoir accès à un serveur présent dans le réseau de l'entreprise. Pour ce faire, nous pouvons configurer le NAT pour qu'il fasse du port forwarding.

Le principe est le suivant : pour accéder à un serveur de l'entreprise, on attribue un port sur le routeur dédié à ce serveur. Dès que le routeur reçoit une connexion de l'extérieur sur ce port, il transfère les paquets vers l'adresse privée du serveur.

Le port forwarding est utilisé abondamment dans les réseaux surtout depuis la virtualisation. Par exemple, VirtualBox définit par défaut un réseau NAT entre la machine hôte et la machine virtuelle invitée. Si vous installez un serveur dans la machine virtuelle invitée et que vous voulez y accéder depuis l'extérieur (votre machine hôte ici), vous devrez faire du port forwarding. Ceci n'est pas compliqué, il suffit de faire une association entre un port de la machine hôte et un port la machine virtuelle.

4 Installation Serveurs Linux et Windows

4.1 Objectifs de ce chapitre

- **Comprendre les différents éléments qui mérite attention lors de l'installation d'un serveur**

4.4.1 Logiciel Libre – GPL

Le logiciel libre se définit comme un logiciel qui peut être étudié, modifié et diffusé. Cela garantit donc l'accès au code source du programme et par cette occasion de vérifier la transparence et la sécurité du programme. Le logiciel libre a été initié par Richard Stallman qui créa en 1985 la FSF(Free Software Foundation). À partir de ce moment, de nombreux développeurs ont distribué leurs logiciels sous licence **GPL (GNU Public Licence).**

Cette licence se caractérise par 4 libertés à respecter :

1. **La liberté d'exécuter le logiciel, pour n'importe quel usage**
2. **La liberté d'étudier le fonctionnement d'un programme et de l'adapter à ses besoins, ce qui passe par l'accès aux codes sources**
3. **La liberté de redistribuer des copies**
4. **L'obligation de faire bénéficier la communauté des versions modifiées (copyleft)**

La dernière liberté est la plus contraignante, car elle nécessite qu'un logiciel utilisant une licence GPL doive être distribué sous licence GPL. C'est pourquoi d'autres licences ont vu le jour pour mieux s'adapter aux réalités des entreprises.

4.4.2 LGPL

La licence LGPL(Lesser GNU Public Licence) reprend les fondements de la licence GPL en supprimant la restriction sur l'hérédité de la licence GPL. Cette licence permet également la cohabitation de plusieurs licences(libres et propriétaires) au sein d'un logiciel. Il s'agit pour ces raisons de la licence préférée des développeurs de librairies.

4.4.5 Remarques

Gratuit ne veut pas dire libre ! La gratuité implique simplement que l'on ne paye pas pour un produit. Open source ne veut pas dire libre ! L'Open source implique simplement que nous avons la possibilité d'avoir accès au code source. La notion de logiciel libre fait référence aux 4 libertés citées ci-dessus.

4.5 RAID

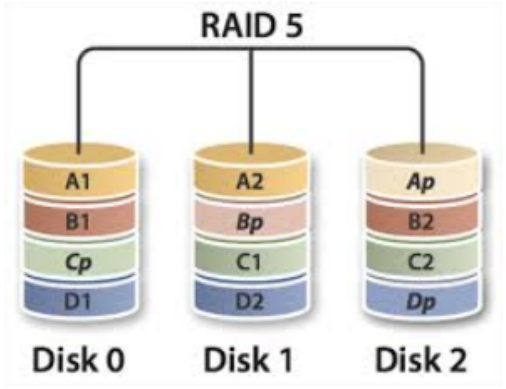
Il est important pour un serveur d'éviter toute panne de disque dur qui pourrait rendre le serveur indisponible.

Le RAID (Redundant Arrays of Inexpensive Disks) est un mécanisme de redondance de disques. Celui-ci a plusieurs objectifs combinables à savoir se prémunir contre la panne d'un disque et améliorer les performances en écriture sur les disques.

Niveau	Explication	Avantages	Inconvénients
RAID 0	Les données sont écrites en parallèle sur plusieurs disques	Gain en performance	Perte d'un disque == perte des données
RAID 1	Les données sont écrites sur des disque en miroir	Tolérance aux pannes disques	Coût -> disque en double
RAID 5	Les données sont écrites en parallèle sur au minimum 3 disques (2 disques de données + 1 disque de parité)	Tolérance aux pannes disques + performance	Minimum de 3 disques requis

Tableau à connaître

Le RAID 5 s'appuie sur l'opérateur logique XOR. La parité de chaque écriture est calculée via cet opérateur. En cas de perte d'un disque, l'information peut être reconstituée à partir des 2 autres disques toujours via cet opérateur XOR.



Disk 0	Disk 1	Disk 2
11	00	11
10	01	11

4.6.1 Partitionnement

Le partitionnement désigne l'opération de diviser un disque en partitions. Un partitionnement bien réfléchi facilitera la maintenance des serveurs. Une partition système et une partition "données utilisateur" faciliteront par exemple la mise en place de sauvegardes.

4.6.1.1 LVM

À l'installation, l'administrateur système devra réaliser un partitionnement. Cependant il parfois difficile d'imaginer comment vont évoluer les données (la taille et la quantité des données) au cours du temps. Afin de répondre à ce problème, les systèmes d'exploitation proposent une gestion dynamique des partitions. Ces partitions pourront grandir par l'ajout de nouveaux disques.

LVM(Logical Volume Manager) est une solution permettant une gestion dynamique du partitionnement disponible sous Linux.

4.6.1.2 Systèmes de fichiers

Lors du choix du partitionnement, il sera nécessaire de préciser le système de fichiers de chaque partition. Voici les principaux :

1. **Ext2,Ext3,Ext4 : système de fichiers Linux** incluant le concept de permissions (Ext4 est actuellement la version la plus utilisée sous Linux)
2. **NTFS : système de fichiers Windows** incluant le concept de permissions (NTFS est actuellement la version la plus utilisée sous Windows)
3. **FAT32 : ancien système de fichiers limité à des fichiers de maximum 4GB**, pas de systèmes de permissions
4. **Swap : partition d'échange (extension de la mémoire RAM sous Linux)**
5. **Btrfs : nouveau système de fichier permettant la prise d'instantanés (snapshot d'une partition) et le redimensionnement de partitions à chaud. Ce système utilise en interne les arbres B-tree.**

Chaque ligne du fstab indique donc une partition, son point de montage, le système de fichiers utilisé, les options, si une sauvegarde doit être faire avec l'utilitaire dump (peu utilisé), l'ordre de vérification des disques lors d'une demande de vérification(fsck).

4.6.1.4 Chiffrement des partitions

Une partition non chiffrée peut être lue facilement sans autorisations particulières via un live-cd si on a un accès physique à une machine.

Linux propose LUKS(Linux Unified Key Setup) qui permet un chiffrement des partitions (à l'installation ou plus tard). **Windows propose, quant à lui, BitLocker** pour crypter ses partitions.

4.6.2 Amorçage

Lors du démarrage d'une machine, un chargeur d'amorçage(bootloader) est lancé. Celui s'occupera de lancer le système d'exploitation ou de présenter les différents systèmes d'exploitation dans le cas d'un multi-boot. Windows propose winload comme chargeur d'amorçage tandis que Linux propose essentiellement GRUB (GRand Unified Bootloader).

4.7 Installation Windows (Windows Server 2016)

L'installation d'un serveur Windows est assez simple. C'est l'ajout de service, appelé rôle sous Windows, qui reste plus complexe. Les rôles permettent d'installer un Active Directory, un serveur DNS, DHCP, … .

Un aspect important sous Windows est la gestion des licences. Les licences serveur doivent être comptabilisées suivant le nombre de cœurs physiques du processeur. Il faut également comptabiliser les licences d'accès client (CAL).

5 Administration Linux (Debian)

5.1 Objectifs de ce chapitre

- **Savoir administrer un serveur Linux (Debian)**

5.2 APT

Toutes les distributions Linux disposent d'un système de gestion des packages permettant l'installation facile de logiciels et services. Ce système de gestion de packages résout en outre les problèmes de dépendances. Sous Debian ce programme est APT. Nous ne donnerons ici qu'un résumé du fonctionnement de l'outil APT. Différents dépôts contenant des paquets Debian (.deb) c'est-à-dire des logiciels prêts à être installés sont disponibles sur Internet. **L'outil APT dispose d'un fichier de configuration (/etc/apt/sources.list) permettant de renseigner les dépôts à utiliser.** Il suffit ensuite de mettre à jour depuis les dépôts (mise à jour du cache local) et de demander l'installation du logiciel à APT. L'outil installera automatiquement les dépendances nécessaires pour le logiciel demandé.

5.2.1 Commandes APT

Mise à jour du dépôt local :

```
apt-get update
```

Mise à jour des logiciels installés :

```
apt-get upgrade
```

Installer un logiciel :

```
apt-get install <paquet1> <paquet2> ...
```

Supprimer un logiciel :

```
apt-get remove <paquet1> <paquet2> ...
```

Rechercher un logiciel/paquet :

```
apt-cache search <word>
```

5.3 SSH

Les systèmes Linux actuels sont le plus souvent gérés en ligne de commande (pas d'interface graphique) et à distance. Pour ce faire, on utilisait telnet mais ce protocole a le gros inconvénient de ne rien crypter. Une simple écoute réseau permet alors de récupérer le mot de passe root. SSH est venu remplacer telnet.

5.3.1 Fonctionnement

Nous ne donnerons ici qu'un résumé du fonctionnement du protocole SSH. Le protocole SSH effectue un échange de clés de chiffrement avant d'utiliser ces dernières pour crypter toutes les communications entre le client et le serveur. Le port 22 est le port par défaut utilisé par SSH.

SSH est un service qui est initialisé/démarré par systemd.

5.3.2 Installation

apt-get install ssh

5.3.3 Configuration

Le fichier de configuration client est : /etc/ssh/ssh_config Le fichier de configuration serveur est : /etc/ssh/sshd_config Par défaut, SSH est installé pour permettre une authentification par login et mot de passe pour tous les utilisateurs présents sur le serveur (**hormis root**). **Après avoir effectué une modification dans un fichier de configuration, il faut redémarrer le service pour que les modifications soient prises en compte.**

systemctl restart ssh

5.3.4 Utilisation

Le client SSH a besoin des informations suivantes : un nom de machine ou une adresse IP, un login et un mot de passe. On peut remplacer l'authentification par login/mdp par une clé.

5.3.6 Copie de fichiers

Il est à noter que dès que vous avez un accès SSH, vous pouvez copier des fichiers entre votre machine hôte et invitée via SCP/SFTP. Ceci peut se faire en ligne de commande (Linux), avec le logiciel WinSCP (Windows) ou Cyberduck(Mac).

5.3.7 Authentification par clé sur un serveur Linux

Pour réaliser une authentification SSH par clé, les étapes suivantes sont nécessaires :

1. Générer une paire de clés (privée/publique) sur le client via par exemple Puttygen (Windows) ou ssh-keygen (Linux)

Attention si vous utiliser Puttygen → copier le contenu de la clé publique généré dans un fichier texte. N'utiliser pas le bouton « Save public key » car il enregistre la clé sous un format non reconnu sous Linux.

5.3.8 Tunnel SSH

La création d'un tunnel SSH permet de connecter 2 machines en encapsulant le trafic de la première et en le redirigeant vers la seconde. Cette technique est souvent appelée le VPN du pauvre car elle permet notamment de donner accès à une machine du réseau local de l'entreprise à des ordinateurs distants et à moindres frais (de configuration).

5.4 Gestion des utilisateurs

5.4.1 adduser-deluser-addgroup-delgroup

Ces commandes sont suffisamment explicites.Consulter la documentation à ce sujet pour connaître les options intéressantes.

Il est à noter que Adduser crée un profil pour l'utilisateur basé sur un répertoire squelette situé dans /etc/skel. Tout fichier placé par l'administrateur dans ce répertoire squelette sera copié par défaut dans le répertoire de l'utilisateur lors de l'appel à adduser.

Par défaut, la home directory créée par adduser est accessible en lecture à tout le monde (voir /etc/adduser.conf). Attention, ceci peut ne pas correspondre à votre politique de confidentialité. Ceci peut être changé dans /etc/adduser.conf.

5.4.3 SUDO

La commande sudo a pour objectif de permettre à des utilisateurs d'exécuter des commandes en tant que superutilisateur.

5.4.3.1 Fonctionnement

Pour qu'un utilisateur puisse exécuter une commande avec « sudo », il doit faire partie du groupe sudo.

5.4.3.4 Avantages de SUDO

Les avantages du SUDO sont les suivants:

1. **Permettre à des utilisateurs d'exécuter une commande en tant que superutilisateur sans devoir le mot de passe de root.**
2. **Travailler en mode non privilégié et n'utiliser le mode privilégié que quand cela est nécessaire. Ceci réduit le risque de commettre des dommages pour le système.**
3. **Contrôler et enregistrer qui fait quoi (SUDO enregistre toutes le commandes sudo effectuées dans /var/log/auth.log).**
4. **Renforcer la sécurité. En désactivant le compte root et en le remplaçant par un compte « sudo », un attaquant ne connaîtra pas le mot de passe, ni le nom du compte !**

5.5 Passwd

Passwd permet de changer le mot de passe de son compte et de tous les comptes (pour le root).

5.6 SystemD

Le système d'exploitation Linux est né sur base du système d'exploitation Unix. Il a donc récupéré énormément de caractéristiques de ce système notamment son système d'initialisation. Les systèmes Unix utilisent une architecture System V. Cette architecture possédait à l'origine de nombreux avantages tel que la mémoire partagée, les sémaphores (cfr. Cours Pgm Sys) qui sont d'ailleurs toujours utilisé aujourd'hui. Ce système d'initialisation était donc le premier processus (init) que lançaient les systèmes Unix et Linux. L'architecture System V divisait l'environnement d'exécution en une série de runlevel et disposait d'un fichier inittab qui précisait quelles applications étaient lancées suivant le **runlevel**.

Les niveaux d'exécutions :

- **0 : arrêt (la commande init 0 arrête le système)**
- **1 : mono-utilisateur (utiliser par exemple pour la maintenance)**
- **3 : multi-utilisateur sans environnement graphique**
- 2-4 : idem que 3, mais peut être défini par l'utilisateur (peu utilisé en pratique)
- **5 : multi-utilisateur avec environnement graphique**
- **6 : redémarrage (la commande init 6 redémarre le système)**

L'objectif principal de SystemD (tout comme SystemV) est de démarrer des services, appelé daemons dans le monde Linux. Il est donc normal que celui-ci propose différentes manières d'implémenter son service.

6 Serveurs Web (Apache)

6.1 Objectifs du chapitre

- **Déployer un site Web via Apache**

6.3 Etapes de déploiement d'un site Web

1. **Installer le paquet du serveur Web (Apache)**
2. **Transférer/Installer le code du site Web sur le serveur (/var/www)**
3. **Créer un VirtualHost**
4. **Activer le site**
5. **Faire correspondre la Directive ServerName et /etc/hosts**
6. **Tester le site Web en local**

6.5 Caractéristiques d'Apache

Apache étant hautement configurable, il se caractérise par une configuration morcelée. Le fichier de configuration de base est /etc/apache2/apache2.conf. Ce fichier inclut tout simplement d'autres fichiers et répertoires à savoir:

- **/etc/apache2/sites-available : définitions de site Web (VirtualHost)**
- **/etc/apache2/sites-enabled : définitions de site Web (VirtualHost) activés**
- **/etc/apache2/mods-available : liste des modules (SSL, proxy, ..) installés**
- **/etc/apache2/mods-enabled : liste des modules (SSL, proxy, ..) activés**
- **/etc/apache2/conf-available : liste des configurations (charset, ..) disponibles**
- **/etc/apache2/conf-enabled : liste des configurations (charset, …) activés**
- **/etc/init.d/apache2 : un service qui sera démarré/arrêté par SystemD**
- **/etc/apache2/ports.conf : la configuration des ports pour apache (80 et 443 par défaut)**

6.6 VirtualHost

Les virtualhosts permettent de déployer plusieurs sites Web sur un même serveur (même adresse IP). La distinction se fait en général sur le nom du site, apache doit en effet savoir suivant l'URL quel site il doit présenter.

Ensuite, il faut activer le site comme suit : a2ensite monsite.conf

6.7 Reverse proxy

Un proxy inverse est un serveur frontal c'est-à-dire un serveur exposé sur Internet et par lequel toutes les requêtes passeront. Ce serveur ne traitera pas les requêtes, mais se contentera de les rediriger vers d'autres serveurs internes à l'entreprise.

Les intérêts de ce mécanisme sont multiples. Vu qu'il n'y a qu'un seul point d'accès, la sécurité est plus facile à gérer. Cela permet également de mettre en œuvre du «load balancing» entre des serveurs internes. C'est également un moyen simple de rendre disponible un serveur interne sur le Web (pas besoin de configuration réseau).

6.9 Apache et HTTPS

Un serveur Web doit être sécurisé en particulier les échanges entre le client et le serveur doivent être cryptés. Ceci se fait aisément grâce au paquet Openssl. **Le port par défaut pour les communications https est le 443.**

6.9.3 Let's Encrypt

Let's encrypt est une autorité de certification libre, gratuite et automatisée. Ceci permet d'obtenir un certificat valide pour son site Web sans trop d'effort. Cependant, la machine servant le site Web doit être «publiquement» accessible ainsi que le nom du domaine. Cela veut dire qu'en test ce procédé n'est pas applicable.

7 Partage et accès réseau

7.1 Objectifs du chapitre

- **Connaître les différents moyens de partage et d'accès à un serveur et leur rôle**

7.2 NFS

NFS est un protocole réseau couramment employé pour partager des fichiers sur un réseau. NFS fonctionne en mode client-serveur. Les versions les plus utilisées sont la 2 et 3. Nous ne parlerons pas ici de la version 4. **NFS est un protocole performant, mais sans sécurité accrue. Ce protocole est le protocole de partage réseau par excellence sous Linux/MacOS.**

7.2.3 Fonctionnement NFS

NFS vérifie l'identité des utilisateurs via les UID, GID. Il faut donc que les UID, GID de la machine distante et locale corresponde. L'UID de root est toujours le même (0).

Il est donc important de comprendre que NFS (v2,V3) ne demande pas aux utilisateurs de s'authentifier.

7.3 SAMBA

SMB/CIFS est un protocole propriétaire utilisé sous Windows pour les partages réseau.

Vous connaissez certainement ces partages réseaux qui ont des paths UNC de la forme : \\machineserveur\partage.

Samba est un logiciel né sous Linux qui implémente ce protocole propriétaire SMB/CIFS. **Il permet donc une interopérabilité entre les 2 mondes(Linux/Windows)** notamment:

- Créer des partages sous Linux accessible sous Windows
- Transformer un serveur Linux en Domain Controler
- Authentifier des clients Linux sur un Active Directory

Samba fonctionne tout comme NFS en mode client-serveur. Samba possède une gestion plus élaborée au niveau de l'authentification et des droits des utilisateurs par rapport à NFS.

7.4 VPN

Un VPN(Virtual Private Network) est un système permettant de relier 2 réseaux via un réseau non sûr tout en garantissant un trafic sécurisé (crypté) et de manière transparente. On parle de tunnel.

Les VPN sont de plus en plus utilisés, car ils permettent notamment de se protéger des écoutes sur un réseau public (NSA, …) et d'anonymiser sa connexion.

7.4.1 Classification VPN

Il existe différents types de VPN à savoir:

1. **LAN to LAN / Site to Site permettant de relier 2 réseaux (Ex: relier des succursales d'une entreprise éparpillées dans le monde)**
2. **RoadWarrior permettant à un PC externe de se connecter à l'entreprise (Nomadisme des employés)**

Il est important de comprendre que le VPN crée un réseau et le configure pour que le réseau distant (LAN to LAN) ou PC distant (Road Warrior) soit considéré comme s'il était dans l'entreprise. L'utilisateur pourra donc utiliser tout ce qui accessible dans le LAN de l'entreprise (imprimantes …).

OpenVPN	**Couche Application**	**OpenVPN est un logiciel créant un VPN en se basant sur SSL/TLS – RoadWarrior ou Site to Site**

7.5 FTP

Le FTP (File Transfer Protocol) est un protocole réseau standard. On le retrouve donc très facilement sous n'importe quel environnement (Windows/Linux). De nombreux clients (graphiques ou non) existent. On l'utilise encore fréquemment pour transférer du code source sur un serveur hébergé.

Comme son nom l'indique, ce protocole est fait pour transférer des fichiers entre ordinateurs. C' est le protocole le plus efficace pour cette tâche. IL est dès lors très utilisé DNS les tâches de backups.

Le protocole FTP utilise un canal pour le transfert des données et un autre pour le contrôle. C'est pourquoi il utilise par défaut 2 ports (20→ données, 21 → contrôle). Le canal de contrôle permet d'envoyer les commandes FTP (put, get, open, close, ls, …).

Il peut être sécurisé via SSL/TLS (FTPS) ou par SSH (SFTP). Regardez dans WinSCP, vous verrez que vous transférez par défaut vos fichiers par SFTP!

7.5.1 Modes

Le protocole FTP peut s'utiliser en mode actif ou passif.

Dans le mode actif, le client peut choisir son port de connexion pour la réception des données. Le serveur initialisera une connexion de son port 20 vers le port choisi par le client. Le client doit donc accepter les connexions entrantes sur le port choisi. Ceci pose souvent problème car les clients se trouvent généralement dans un LAN qui effectue du NAT vers l'extérieur. C'est pourquoi le mode actif est le moins utilisé.

Dans le mode passif, le serveur impose le port de connexion pour le transfert des données. Le port choisi par le serveur est envoyé au client qui initialise alors la connexion.

7.6 Terminal Server

L'administration d'un serveur Linux se fait assez facilement via SSH. Le pendant dans le monde Windows est le Terminal Server aussi appelé Bureau à distance. **Le Terminal Server repose sur le protocole RDP (Remote Desktop Protocol) développé par Microsoft.**

Terminal Server permet non seulement la prise à distance d'un serveur avec son interface graphique, mais aussi de monter des lecteurs locaux sur le serveur distant.

8 Annuaires et Authentification

8.1 Objectifs du chapitre

- **Connaître les différents composants d'un Active Directory et leurs rôles**

Dans un réseau d'entreprise, il devient rapidement nécessaire de centraliser les authentifications afin de faciliter la maintenance de la politique des accès, des droits, des stratégies de sécurité.

Le protocole LDAP (Lightweight Directory Access Protocol) a été défini pour permettre d'interroger et de modifier un annuaire. Il est depuis devenu une référence et un standard pour l'authentification.

LDAP est en fait devenu bien plus qu'un protocole, c'est une norme qui définit:

1. **un protocole: comment sont échangées les données**
2. **un modèle de nommage: comment sont nommées les entrées dans l'annuaire**
3. **un modèle fonctionnel: quelles sont les méthodes pour accéder aux données**
4. **un modèle d'information: nature et description des données**
5. **un modèle de sécurité: description de la sécurité des données (quel chiffrement …)**
6. **Réplication: comment répliquer des données entre serveurs LDAP pour se prémunir des pannes? Ce point n'est pas encore standardisé.**

Différentes implémentations de la norme LDAP existent. **OpenLDAP est la solution reconnue du monde libre (paquet slapd dans Debian). La solution Active Directory de Microsoft est sans doute la plus connue et répandue.** Nous en parlerons plus longuement (Voir Active Directory).

8.2 Modèle de nommage

Un annuaire sera élaboré à partir d'une structure de données de type arbre hiérarchique représentant l'organisation d'une entreprise. La racine de cet arbre sera un nom DNS (le nom DNS de l'entreprise généralement), les nœuds seront les divisions de l'entreprise (départements, sections, année d'étude …) et les feuilles seront les objets (machines ou utilisateurs principalement).

Voici le vocabulaire de base utilisé dans un annuaire LDAP:

- **DC : Domain Component. Racine de l'arbre**
- **DN : Distinguished Name. Chemin complet vers un élément (Les DN sont uniques)**
- **OU : Organizational Unit. Division de l'entreprise rassemblant des CN.**
- **CN : Common Name. Nom d'un élément**

8.3 Modèle fonctionnel

LDAP a défini différentes méthodes permettant de modifier et consulter l'annuaire. Voici la liste des principales méthodes :

- **Bind : s'authentifier auprès du serveur LDAP. Ceci est nécessaire avant de demander au serveur une opération au serveur**
- **Add/Modify/Delete : mise à jour de l'annuaire.**
- **Search : «search» permettra de rechercher un élément ou plusieurs éléments dans l'annuaire en précisant une base, une portée et éventuellement des filtres (voir ci-dessous).**
- **Compare : vérifie qu'un élément contient ou non un attribut**
- **Unbind : se déconnecter du serveur**

8.4 Modèle d'informations

LDAP définit le modèle d'information suivant :

- **Entrée: composé d'attributs, possède un type (classe d'objets)**
- **Schéma: définition des attributs possibles et classes d'objets**
- **DN (Distinguished Name)**

8.5 Modèle de sécurité

Le transport des messages LDAP sera chiffré via SSL/TLS. LDAP présente différentes méthodes d'authentification. **L'utilisateur, une fois authentifié (via un bind), aura accès aux données suivant les règles établies dans les ACL (Access Control List).**

8.6 Modèle de replication

Un modèle de réplication est prévu dans la norme LDAP, mais il n'est pas encore standardisé. La réplication est un élément important pour assurer une redondance qui reste le moyen privilégié par les administrateurs système pour se prémunir des pannes. Les différentes implémentations LDAP (Active Directory, OpenLDAP, ..) ont développé leurs systèmes de réplication.

À noter également que LDAP fournit un format d'échange standard nommé LDIF (LDAP Data Interchange Format) qui permet d'échanger/sauvegarder de l'information entre serveurs LDAP. Cependant celui-ci ne permet pas une réplication aisée.

8.8 Active Directory

L'Active Directory est une implémentation Microsoft d'un annuaire LDAP. Cette implémentation est évidemment adaptée aux environnements Microsoft. **L'Active Directory permettra de gérer l'authentification des utilisateurs d'un ou plusieurs domaines, de gérer les droits des utilisateurs via des groupes de sécurité.** L'Active Directory introduit son propre vocabulaire en plus du vocabulaire LDAP.

Un Active Directory définira une forêt qui sera composée d'arbres(domaine parent avec des domaines enfants) et/ou de domaines. Les domaines seront, quant à eux, constitués d'unités d'organisation et enfin d'ordinateurs, de groupes et utilisateurs.

Dans un Active Directory, nous allons retrouver différents objets. Les plus importants sont les utilisateurs, les ordinateurs du domaine, les groupes de sécurité, les GPO (Voir GPO), les unités d'organisation.

Un serveur hébergeant un Active Directory est appelé contrôleur de domaine. Il est conseillé d'avoir au minimum 2 contrôleurs de domaine par Active Directory pour se prémunir des pannes.

8.8.1 Unité d'organisation

Une unité d'organisation est un regroupement d'objets de l'Active Directory. C'est un nœud dans l'arbre LDAP. Les unités d'organisation trouvent essentiellement leur utilité par le fait que des GPO puissent être définies à ce niveau.

Ne pas confondre les unités d'organisation et les groupes de sécurité. Des GPO ne peuvent pas être définies sur des groupes et des permissions ne peuvent pas être définies sur des unités d'organisations. Un objet(utilisateur) peut appartenir à plusieurs groupes, il ne pourra pas contre être placé que dans une seule unité d'organisation.

8.8.2 Groupes de sécurité

Il est évidemment bien plus aisé de gérer les droits des utilisateurs via des groupes. Il existe différentes étendues de groupe :

- Domain Local : groupe utilisé uniquement dans son domaine de création
- Global: groupe utilisé dans tous les domaines approuvés
- Universel : groupe utilisé dans les domaines de la forêt

Microsoft propose une recommandation pour la gestion des droits et des groupes. Il s'agit de la recommandation AG(U)DLP.

Account → Global → (Universel) → Domain Local → Permission

Exemple :

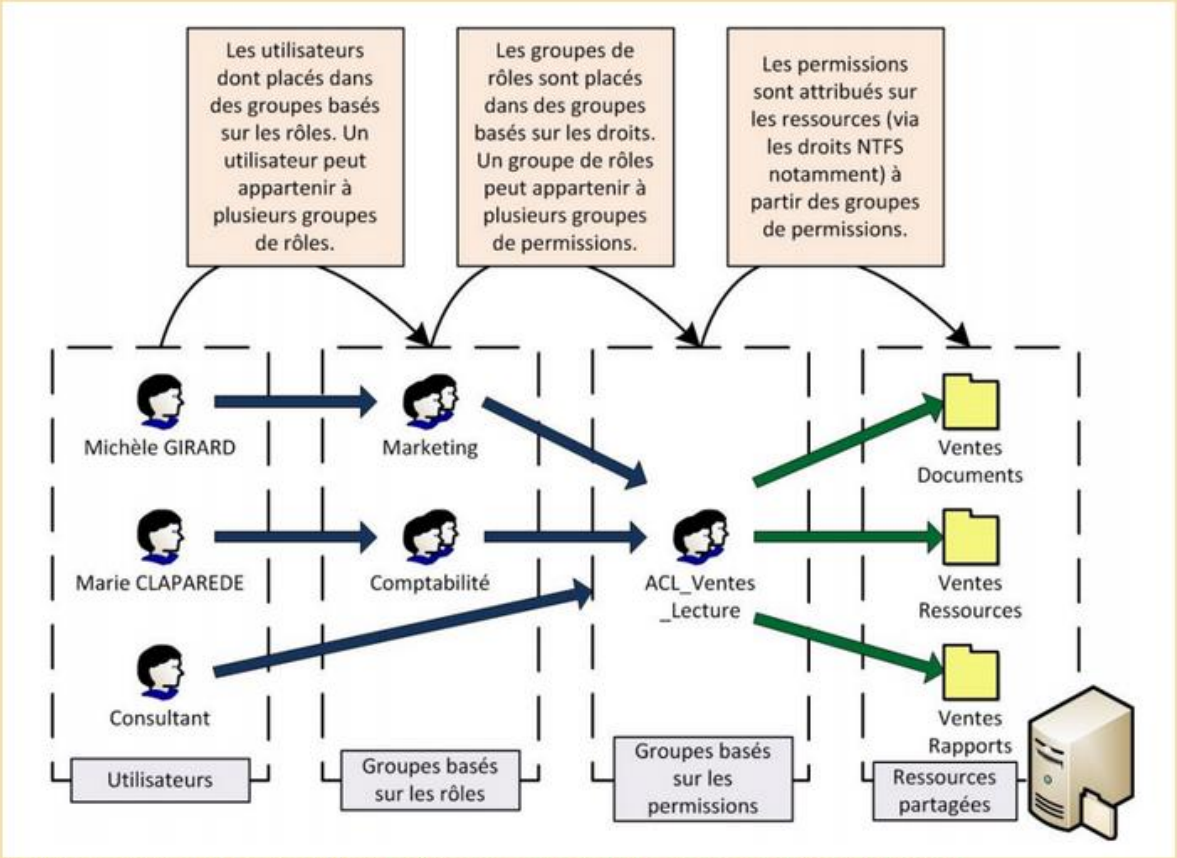


Image labo-microsoft.supinfo.com / **schéma utile à connaître pour expliquer AG(U)DLP**

8.8.3 Permissions

8.8.3.1 Permissions NTFS

Microsoft utilise le système de fichiers NTFS et dès lors les permissions seront des permissions NTFS. Les permissions NTFS sont très riches (beaucoup de possibilités).

8.8.3.2 Permissions Partage réseau

Les partages réseau sont des dossiers (présents généralement sur des serveurs) qui sont partagés et donc accessibles sur le réseau depuis n'importe quelle machine. Ces partages réseau utilisent le protocole SMB (Voir SAMBA).

Attention à ne pas confondre les permissions NTFS qui s'appliquent directement sur des fichiers et dossiers locaux (présents sur le système de fichiers) et les permissions de partage qui sont définies sur les partages réseau.

Les permissions de partage réseau sont plus limitées (R, M, FC). **Lors de l'accès à un partage réseau, le système évalue tout d'abord les permissions du partage réseau et ensuite les permissions NTFS (puisque tout partage réseau se retrouve toujours physiquement sur un système de fichiers).**

8.9 GPO

Les GPO (Group Policy Object) permettent de définir des stratégies de sécurité et/ou de configurer des paramètres de manière centralisée pour un domaine ou une forêt. Par exemple, on peut imposer un proxy pour les clients, définir un fond d'écran, définir une politique de mot de passe (longueur, complexité …).

Les GPO, aussi appelées stratégie de groupe en français, permettent de déployer des stratégies/paramètres sur 2 types objets de l'Active Directory : les utilisateurs et les ordinateurs. Les GPO se créent le plus souvent sur des Organizational Unit.

Les GPO machines (configuration ordinateur) s'appliquent au démarrage de la machine. Les GPO utilisateurs (configuration utilisateur) s'appliquent à l'ouverture de session d'un utilisateur.

9 Gestion des données

9.1 Objectifs du chapitre

- **Etablir un plan de sauvegarde des données**

9.2.2 Exemple PostgreSQL

Un moteur de base de données est constitué de plusieurs programmes. PostgreSQL est composé d'un programme superviseur (postmaster), du serveur exécutant les requêtes (postgres) et d'un client interactif (psql).

9.2.2.3 Sauvegarde PostgreSQL

Il est important de sauvegarder les bases de données. Voici les erreurs à ne pas commettre :

- **Pas de sauvegarde**
- **Sauvegarde jamais testée**
- **Ne pas documenter sa sauvegarde (qui l'a fait/ quand, comment …)**
- **Sauvegarder sur le même disque**
- **Sauvegarder au même endroit (incendie ?)**
- **N'avoir qu'une sauvegarde très récente car elle contiendra également l'erreur**

Les sauvegardes de type fichiers ou instantanés ne conviennent pas aux bases de données. Il est préférable d'utiliser l'outil fourni par le moteur de base de données. Dans le cas de PostgreSQL, nous avons pg_dump.

9.2.2.4 Performance PostgreSQL

Il est important de savoir que les limitations des bases de données se situent essentiellement du côté des composants sous-jacents (CPU, RAM, vitesse disque, capacité disque …).

Par exemple, la limite au niveau des tables pour PostgreSQL est actuellement de 64 To. Ce n'est donc pas ceci qui va poser problème. PostgreSQL reste stable et performant même avec des DB ≥ 30 GB.

Sachant ceci, il faut commencer par surveiller ces éléments via les commandes/outils du système d'exploitation. Par exemple : uptime pour la charge CPU, free pour la charge mémoire.

La performance passe également par l'ajout d'index. Un index peut accélérer par 1000 000 une requête ! PostgreSQL dispose de la commande EXPLAIN qui décrit le plan d'exécution d'une requête. Ceci s'avère être un outil puissant pour accélérer une requête problématique.

L'utilisation de LIMIT est également à envisager. Doit-on vraiment charger l'entièreté d'une table, les premiers résultats ne sont-ils pas suffisants ?

L'utilisation régulière de VACUUM pour supprimer définitivement les données expirées est à envisager, particulièrement sur les bases de données des développeurs qui créent et détruisent souvent leurs bases de données.

L'utilisation d'ANALYSE permet de collecter des statistiques en vue d'optimiser la base de données.

Un petit mot sur les ORM pour finir. Ceux-ci sont très utiles, mais l'optimisation s'avère souvent plus complexe à gérer vu que vous laissez l'ORM créer les requêtes SQL pour vous. En tant que développeur, posez-vous (ou à votre chef de projet) donc la question de la performance avant de vous décider.

9.2.2.5 Réplication PostgreSQL

Comme d'habitude, les administrateurs systèmes ou DBA vont vouloir se prémunir des pannes en installant de la redondance. Avec les bases de données, ceci peut se faire en activant la réplication. Ce mécanisme permet à 2 moteurs de base de données de partager des bases de données. Les données de ces bases de données se retrouvent alors répliquées/dupliquées au minimum sur 2 serveurs de bases de données.

La réplication la plus utilisée est celle dite du maître – esclave. Toutes les opérations sont faites sur le serveur maître et celui-ci envoie régulièrement au serveur esclave ses journaux de transactions. Ce dernier rejoue alors le journal de transactions sur sa base de données.

9.3 Sauvegardes

Un administrateur système sera en charge de la pérennité des données. Pour cette tâche, des sauvegardes ou backups seront nécessaires. Les sauvegardes permettent de réduire les risques liés à des pannes, virus (ransomware), erreurs humaines. **Bien sûr il est souvent impossible de tout sauvegarder, c'est pourquoi il faut définir une politique de sauvegarde qui répond aux questions suivantes.**

Les entreprises auront donc souvent recours à des NAS (Network Area Storage) ou SAN distant ou encore à un stockage dans le Cloud. Dans ce dernier cas, la confidentialité des données doit être évoquée.

9.3.3 Quel type de sauvegarde, quelle fréquence ?

Il existe différents types de sauvegarde à savoir :

1. **complète : une copie complète des données est faite**
2. **incrémentielle : une copie des modifications depuis la dernière sauvegarde (complète ou incrémentielle) est faite**
3. **différentielle : une copie des modifications depuis la dernière sauvegarde complète est faite**
4. **miroir : une seule sauvegarde complète.**
5. **instantané/snapshot : pas vraiment une sauvegarde car difficile de restaurer un seul fichier**

Souvent la fréquence de sauvegarde va influencer le type de sauvegarde.

9.3.4 Conservation des données ?

Les sauvegardes peuvent occuper beaucoup de place. Il est donc nécessaire à réfléchir au stockage et à comment gagner de la place. La compression des données est un moyen souvent utilisé. **La déduplication est un autre moyen mis en place par les entreprises. Ce mécanisme est souvent intégré au SAN.** Il s'agit de découper les données en bloc et toute nouvelle occurrence d'un bloc est remplacée par un pointeur. Ce mécanisme fonctionne très bien avec les backups car beaucoup de données se répètent. Imaginez une fréquence de backups complets tous les mois. Dans ces backups, beaucoup de données seront identiques.

Il est nécessaire de crypter ses backups pour éviter à une personne mal intentionnée de récupérer facilement les données de l'entreprise. L'utilisation grandissante du Cloud renforce également ce besoin.

9.3.6 La règle du 3-2-1

Il existe une règle fortement conseillée pour les sauvegardes de données. Il s'agit de la fameuse règle 3-2-1.

- **3 : conserver 3 copies des données (originale + 2 sauvegardes)**
- **2 : conserver les données sur 2 supports différents**
- **1 : conserver 1 copie de sauvegarde dans un lieu différent (des 2 autres copies)**

Les quotas permettent donc de régler ce problème en imposant une limite aux utilisateurs. Le mécanisme de quotas propose une limite soft et une limite hard. Quand un utilisateur dépasse sa limite soft, il reçoit un avertissement lui indiquant qu'il peut continuer à travailler pour une période de grâce définie par l'administrateur. Une fois cette période grâce écoulée, il est bloqué s'il n'est pas redescendu en dessous de la limite soft.

Comprendre la virtualisation, les différents types de virtualisation

10.2 Avantages de la virtualisation

La virtualisation apporte des avantages certains dans un environnement informatique d'où l'engouement généré par ce concept actuellement. La définition de la virtualisation est la suivante :

La virtualisation consiste en l'abstraction d'un élément du monde réel en le rendant virtuel. Ceci dans l'objectif de rendre l'élément plus facilement :

- **Configurable**
- **Transportable**
- **Optimisé (meilleure allocation des ressources)**
- **Disponible (facilité de déploiement)**
- **Sécurisé (isolation)**

10.4 Inconvénients de la virtualisation

L'inconvénient majeur de la virtualisation est l'ajout d'une couche de virtualisation entre le système physique et le composant virtualisé. Ceci permet d'obtenir les avantages cités ci-dessus, mais dégrade les performances.

10.5 Hyperviseurs

La virtualisation des machines se fait via des hyperviseurs. **Un hyperviseur est un logiciel de virtualisation permettant à plusieurs machines virtuelles de fonctionner simultanément sur un même système physique.**

On distingue 2 types d'hyperviseurs : 1. **Hyperviseur type 1 / natif / bare-metal** 2. **Hyperviseur type 2 / hosted**

Un hyperviseur de type 1 sera utilisé dans le cadre de la virtualisation de serveurs.

A noter que depuis l'apparition du Cloud, de nombreuses offres proposent des **serveurs privés virtuel (VPS)**. Il s'agit d'une illustration de l'utilisation d'hyperviseur bare-metal par des hébergeurs Cloud.

Ce que vous connaissez le plus est certainement **les hyperviseurs de type 2. Ces hyperviseurs s'installent dans un système d'exploitation comme un logiciel classique.** On peut citer **VirtualBox**,

La virtualisation du stockage consiste donc en la création d'une baie de stockage (ensemble de disques physiques) qui sera présentée en un ou plusieurs ensembles logiques et dynamiques. On parle de LUN (Logical Unit Number). Une LUN est donc un espace de stockage logique et dynamique.

Ces baies de stockage sont accessibles via le réseau essentiellement via 2 protocoles (SMB, Fiber Channel). On parle de SAN (Storage Area Network). **Un SAN est donc une baie de stockage avec des LUN accessibles via le réseau.**

La possibilité dans le solutions de virtualisation de réaliser un instantané/snapshot est un énorme avantage. Ceci permet de revenir facilement à un état antérieur en cas de souci. **Il est donc vivement conseillé de réaliser un instantané avant toute modification importante d'un système.**

- **Comprendre les architectures à base de conteneurs**
- **Savoir créer un Dockerfile**
- **Déployer des applications via Docker, docker-compose**

Docker est une solution d'architecture à base de conteneurs. Les architectures à base de conteneurs sont une évolution avantageuse de la virtualisation.

Docker n'est donc pas un logiciel de virtualisation, mais un isolateur.

Les cgroups (Control Groups) permettent de fixer/limiter les ressources (CPU, Réseau, disque, nombre de processus) allouées à un conteneur ou un ensemble de conteneurs.

Les namespaces permettent d'isoler des ressources. Ainsi un conteneur ou ensemble de conteneurs ne voient que les ressources de son namespace.

Les conteneurs sont proches des machines virtuelles, mais présentent un avantage important. Alors que la virtualisation consiste à exécuter de nombreux systèmes d'exploitation sur un seul et même système, les containers se partagent le même noyau de système d'exploitation et isolent les processus de l'application du reste du système.

Les architectures à base de conteneurs sont très utilisées car elles permettent facilement des mises à l'échelle (scaling).

On appelle cela la mise à échelle horizontale (**horizontal scaling**). On peut également augmenter les ressources des conteneurs via les cgroups. On appelle cela la mise à échelle verticale (**vertical scaling**).

La grosse différence entre Docker et les systèmes de virtualisation classiques se situe au niveau de l'interaction avec le système d'exploitation hôte.

- **Docker partagera/utilisera le même noyau(Linux) pour tous les conteneurs.**
- **Chaque VM géré par un hyperviseur aura son propre OS**

Pour déployer une application via Docker, il y a donc 3 étapes essentielles :

- **Création d'un Dockerfile (recette de cuisine de l'application à déployer)**
- **Créer l'image Docker (docker build -t <imagename> <pathToDockerfile>)**
- **Créer un conteneur à partir de l'image créée (docker run -d -p 9000:80 –name <containername> <imagename>)**

La conteneurisation d'une application commence par la création d'un dossier contenant tout ce qui est nécessaire à l'application. Il est nécessaire d'avoir dans ce dossier au minimum :

- **le code de l'application**
- **un fichier Dockerfile**

Le Docker Hub est l'endroit où vous devez rechercher votre image de base !

11.2.5 Couches

A noter que Docker utilise un système de couche (layer). Chaque ligne du Dockerfile correspond à une couche et possède son empreinte (hash).

Ainsi nul besoin pour Docker de télécharger(pull) à chaque build la même image sur le DockerHub. Il va réutiliser les couches qui n'ont pas changé.

Docker est prévu pour faire fonctionner des architectures microservices. Dans ce type d'architecture, on crée un conteneur par service.

Ceci a pour but de pouvoir multiplier le nombre de conteneurs suivant la demande. Imaginez un site Web PHP MVC, nous créerons un Dockerfile avec le code et le serveur Web et un autre Dockerfile avec la base de données. Le Dockerfile avec le code et le serveur Web peut facilement être multiplié en plusieurs conteneurs pour augmenter les performances. C'est ce que l'on apelle de la mise à échelle horizontale (horizontal scaling).

Ceci est bien beau mais :

- **Comment gère-t-on la communication entre les différents conteneurs ? Le conteneur application devra vraisemblablement communiquer avec le conteneur db.**
- **Comment rend-t-on des données persistantes ? Un conteneur est stateless mais les données d'un conteneur db devront être persistées.**

C'est ici qu'intervient docker compose qui est un script python permettant de simplifier la création de ces architectures microservices. Avec un seul fichier YAML, on pourra créer notre architecture. docker compose ajoutera tous les services (conteneurs) dans un même réseau ce qui permettra une communication entre les conteneurs.

docker compose permet également facilement la création de volumes. Ceci permet d'effectuer la persistance de données en dehors d'un conteneur.

11.3.4 Directives intéressantes

La directive «build» permet de construire une image Docker à partir d'un Dockerfile.

La directive «ports» permet d'effectuer la redirection de ports (port forwarding) entre la machine hôte et le conteneur. Cette directive attend un tableau, c'est pourquoi chaque élément commence par un «-».

La directive volumes permet de lier un répertoire de la machine hôte au conteneur. Les volumes ne sont pas détruits par défaut lors de la destruction du conteneur. C'est donc via ce biais que les données persistantes des conteneurs sont gérées.

1.3.5 Réseau

Les noms des services(conteneurs) deviennent des noms réseaux.

La conception de Docker s'est inspirée de la méthodologie des 12 facteurs, une méthodologie pour créer des applications SaaS [Modèles de service]. Voici un lien vers cette méthodologie [https://12factor.net/fr/]. Voyons comment Docker applique celle-ci.

Retenez cinq facteurs avec leurs exemples Docker

1.5 Kubernetes (K8s)

11.5.1 Introduction

Vu le succès des conteneurs, la majorité des applications sont maintenant déployées de la sorte. Il devient donc nécessaire d'avoir un outil pour gérer ces conteneurs. Kubernetes répond à cette demande. **Kubernetes est un orchestrateur de conteneurs.**

11.5.2 Utilité

Kubernetes permet d'appréhender les défis liés à la gestion des conteneurs notamment :

- **L'équilibrage de charge entre plusieurs conteneurs (load balancing)**
- **La gestion des différents stockage pour les conteneurs (volumes). Ceux-ci peuvent être locaux, dans le Cloud, …**
- **La gestion et l'allocation des ressources au conteneurs de manière dynamique**
- **La gestion de l'état de santé des conteneurs**
- **La gestion des informations de configurations et des secrets (clé SSH, password, ..)**

11.5.3 Vue générale et concepts k8s

Kubernetes est composé d'un nœud maître permettant la gestion du cluster k8s. Un cluster est un ensemble de machines physiques ou virtuelles. Chaque machine du cluster est appelée un "worker node" et contiennent une solution de déploiement de conteneurs (Docker par ex.).

Chaque "worker node" héberge un pod. Un pod est l'unité de déploiement d'une application dans k8s. Il s'agit d'un ou plusieurs conteneurs. Dans la pratique, le plus souvent un pod == un conteneur/un processus

Un label est un couple "clé:valeur" que l'on peut attacher à un objet notamment un pod. Les selectors permettent de sélectionner un objet k8s suivant son label

Un fichier deployment.yml permet de définir comment un Pod sera déployé. Ce fichier contient une instruction "replica" qui indiquera à K8s combien d'instances de ce Pod il devra lancer/maintenir.

Un fichier services.yml permet de définir et gérer le réseau à l'intérieur du cluster. Vu que les pods peuvent être créés, détruits et recréés il est nécessaire qu'un service puisse cibler les pods encore actifs à un moment t.

Ingress est un load balancer qui permet de faire communiquer le monde extérieur avec le cluster k8s. Nginx peut fournir ce service.

Les Persistent Volume Claim(PVC) permettent de persister des données. Cela va beaucoup plus loin que l'utilisation de simples volumes (PV). Il s'agit d'une demande d'espace adressée à k8s. Celui-ci attribuera alors au pod un stockage suivant sa demande.

A noter aussi que l'intérêt de k8s par rapport au stockage est de pouvoir utiliser des stockages variés (Amazon, Azure, Local, …).

12 DevOps (Ansible notamment)

12.1 Objectifs du chapitre

- **Consolider les notions DevOps**
- **Déployer des configurations via Ansible**

Le déploiement d'une application faisant maintenant partie du produit ainsi que les mises à jour en continu, il est donc nécessaire d'avoir une infrastructure avec le concours des opérationnels pour mener à bien ce nouveau mode de développement logiciel.

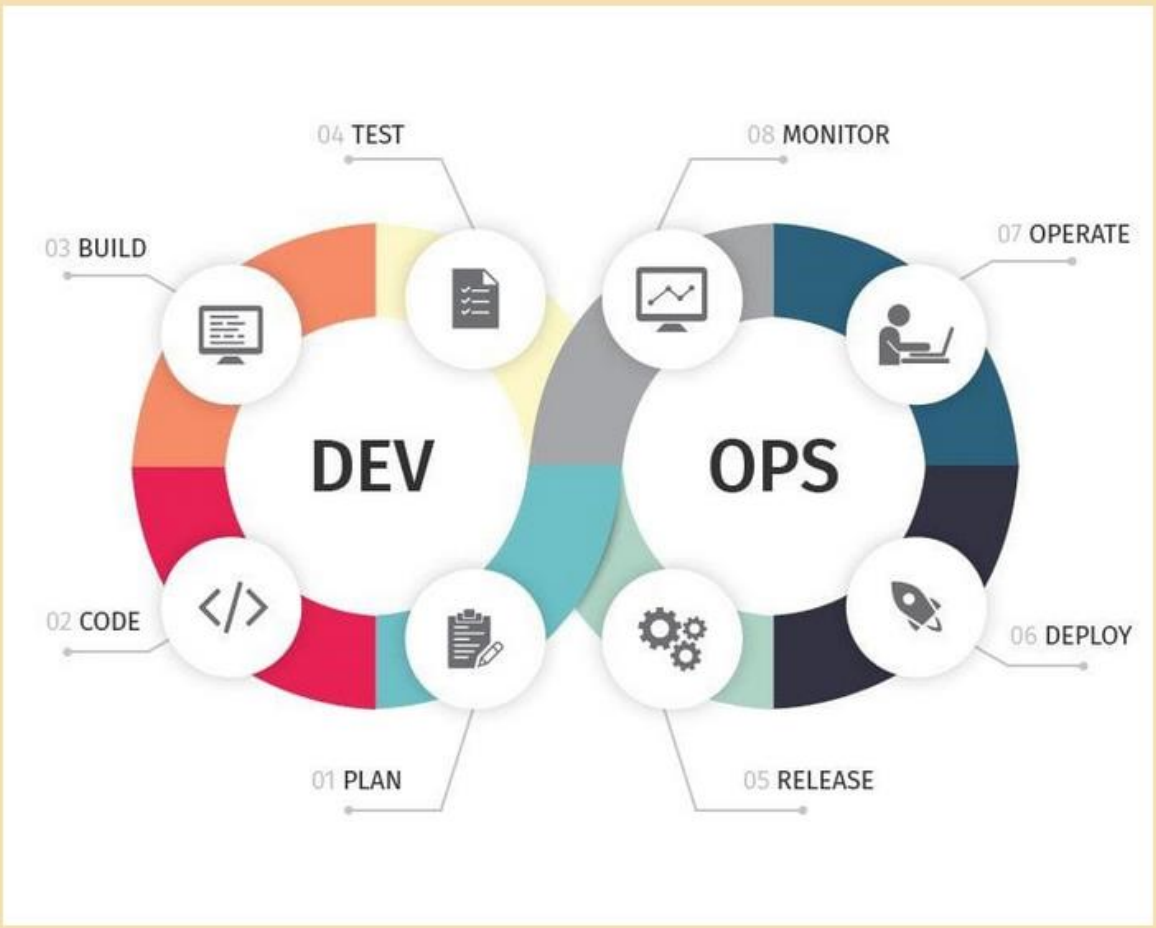


Image issue de (https://res.cloudinary.com/practicaldev/image/fetch/s---dbl8WY9--/c_limit%2Cf_auto%2Cfl_progressive%2Cq_auto%2Cw_880/http://aisaac.io/content/images/2018/11/DevOps.jpg)

Schéma à connaître

12.3 Pipeline de développement

L'approche DevOps mise également sur l'automatisation via des outils. On parlera souvent de pipeline de développement. Il s'agit d'une chaîne de production logicielle la plus automatisée possible jusqu'au client.

Sans surprise, les pipelines utiliseront les éléments vus dans les chapitres précédents surtout les conteneurs, les orchestrateurs, le Cloud et les outils de configurations.

12.4.2 Ansible

Ansible a défini son propre vocabulaire. On parle de *contrôleur* qui est est la machine qui exécute Ansible en tant que tel et de *cibles* qui sont les machines qu'Ansible configure. Les cibles sont définies dans l'inventaire d'Ansible, simple fichier reprenant des noms de machines ou IP de machines. *A noter que le contrôleur et la cible peuvent être la même machine.* Dans ce cas, cela signifie que nous automatisons alors simplement un ensemble de tâches pour une seule machine. Nous procéderons de cette manière durant les labos. *Vous verrez que l'on précisera toujours dans notre fichier Ansible localhost comme cible !*

Ansible se base sur SSH pour pouvoir se connecter et déployer des configurations sur les cibles. Ansible utilise des fichiers YAML. **Je vous conseille d'utiliser VS Code pour faire votre fichier YAML, vous aurez ainsi déjà une vérification syntaxique. Vous pouvez utiliser l'extension Remote SSH de VSCode pour éditer un fichier sur votre VM à travers SSH !**

Ansible parle de playbooks. Un playbook est simplement un fichier YAML décrivant un ensemble de tâches à effectuer sur un ensemble de cibles.

Ansible dispose de nombreux modules permettant d'effectuer des tâches. Voici les plus importants :

- apt : pour installer un package (Debian/Ubuntu)
- git : cloner un repo
- command : exécuter une command shell
- template : copier un fichier de configuration modèle
- lineinfile : modifier des lignes dans un fichier

Ces modules sont disponibles dans la liste des modules builtin : (https://docs.ansible.com/ansible/latest/collections/ansible/builtin/index.html#plugins-in-ansible-builtin)

Voici la liste complète des modules et plugins : (https://docs.ansible.com/ansible/latest/collections/index_module.html) **Utilisez la recherche par catégorie de modules ou par nom de module**

Vous devez être root pour lancer cette commande. D'autres options et subtilités existent dans Ansible mais nous nous limiterons pour ce cours à des playbook simples et lancés en tant que root.

Pour créer vos tâches, il est utile de consultez la liste des Modules Ansible disponibles

Ansible permet de séparer un playbook dans plusieurs fichiers. On appelle cela un **rôle** Ansible.

13 Cloud (Terraform notamment)

13.1 Objectifs du chapitre

- **Connaître les modèles de service Cloud**
- **Architecture Multi-tenant et Cloud**
- **Déployer des services PaaS avec Terraform**

Le Cloud est une abstraction qui s'est déroulée en plusieurs étapes du réseau, du Web, de l'infrastructure.

13.4 Modèles de service

13.4.1 Infrastructure as a Service (IaaS)

IaaS : Déploiement d'une infrastructure via les outils du fournisseur Cloud. L'utilisateur peut paramétrer le réseau, les serveurs, … . Ex: Amazon EC2, Azure VM, …

Terraform est certainement l'outil le plus utilisé pour déployer de l'IaaS. Il permet de créer des VM et d'autres éléments d'infrastructure sur AWS, Azure, … . *Cet outil est décrit dans la section suivante.*

13.4.2 Platform as a Service (PaaS)

PaaS : Déploiement d'une plateforme via les outils du fournisseur Cloud. L'utilisateur peut paramétrer la plateforme mais celui ci n'a aucun accès, vue sur l'infrastructure. Ceci est sans le doute le modèle de service Cloud le plus employé par les développeurs. Ex: Heroku, AWS Elastic Beanstalk, Cloud Foundry, …

13.4.3 Software as a Service (SaaS)

SaaS: Accès à une application via Internet. Aucun accès, vue sur l'infrastructure, paramétrage par l'utilisateur limité. Ex: Gmail, Office365, …

13.5 Architecture Multi-tenant et Cloud

13.5.1 Introduction

Qu'est qu'une architecture multi-tenant ?

Dans une architecture multi-tenant, une même instance d'une application logicielle est utilisée par plusieurs clients, ces derniers étant des « tenants ».

Le modèle multi-tenant peut s'avérer économique, étant donné que les coûts liés au développement et à la maintenance des logiciels sont partagés.

Les modèles de service SaaS présentes dans le Cloud propose souvent les approches multi-tenant et single-tenant. Ainsi suivant ses priorités et son budget, on peut effectuer un choix.

13.6 Terraform

13.6.1 Introduction

Terraform est un outil permettant de déployer des ressources dans le Cloud. En clair, il permet d'automatiser la création de ressources(VM, image docker, …) en local ou dans le Cloud. Il dispose de nombreux connecteurs (providers) permettant de créer des ressources aussi bien dans AWS (Amazon), Azure, en local, … . Ce dernier point le rend particulièrement intéressant au sein des entreprises.