

• Config de l'interface en mode statique ou dynamique:

• Lister les cartes :

```
# ifconfig -a  
eth0      ---      ↙ @  
          ---      addn: 192.168.31.10  
          ---  
          ---  
lo         ---      ↙ fixe  
          ---      addr: 127.0.0.1
```

• Pour savoir si on a l'@ IP en mode stat ou dynamique:

```
# ifdown eth0  
# grep /etc/sysconfig/network-scripts/ifcfg-eth0
```

```
Device = eth0  
Bootproto = "dhcp" ← mode dynamique (@ IP dynamique)  
Onboot = yes ← lors de démarrage, l'interface va être actf.
```

- Pour télécharger un package, il faut que Bootproto = "dhcp"
Puis il faut le modifier en mode statique Bootproto = "none"

```
Bootproto = "none"  
IPADDR = 192.168.2.10  
BROADCAST = ---  
Netmask = ---  
Network = ---
```

• faire le redémarrage:

```
# ifup eth0  
ou  
# service network restart  
ou  
# /etc/init.d/network reload
```

• Configuration statique pour l'interface eth0 d'une manière temporaire

```
# ifconfig eth0 10.0.0.2/24 (on change 192.168.2.10)
```

→ En cas de démarrage de machine ou de service, cette adresse va être supprimée, car il s'agit d'une config temporaire

• Creation d'une nouvelle interface alias :

```
# ifconfig eth0:1 192.168.5.5/24
```

↙ d'une façon temporaire

```
# ifconfig -a
```

↙ elle est virtuelle

```
eth0 _____
eth0:1 _____ ← elle prend m @ Mac que eth0
                    car elle est basé sur cette interface physique
lo _____
```

⇒ Si on veut quel soit permanent, On crée son fichier de config

```
# gedit /etc/sysconfig/network-scripts/ifcfg-eth0:1
```

• Fichier Hosts :

```
# gedit /etc/hosts
```

objectif : associer des noms d'hosts à des @ IP.

```
127.0.0.1    localhost.localdomain
::1          localhost6
192.168.2.10 www.esprit.com
```

⇒ on trouve dans ce fichier : les correspondances / Nom de domaine @ IP

• Lister les serveurs DNS : (name server)

```
# gedit /etc/resolv.conf
```

```
DNS1 = xxx xxx
DNS2 = xxx xxx
```

• Tester les noms de domaine :

```
# nslookup 8.8.8.8
```

```
_____
_____ name = dns.google
```

```
# nslookup www.google.com
```

```
Address: 172.217.20.100
```


Config de Base

Protocole:

c'est l'ensemble des procédures qui va nous permettre de communiquer deux machines de m niveau. (modèles: OSI / TCP-IP)

Réseau:

c'est l'ensemble de machine interconnectés, son objectif est de partager les informations. , faire une communication à partir d'un protocole

Modèle OSI

Accès au Service	App ⑦
Syntaxe des données	Présentation ⑥
Assurer le dialogue	Session ⑤
Qualité de Trans.	Transport ④
Sélection de chemin	Réseau ③
Adressage Physique	Liaison de données ②
Transmission sur support Physiq.	Physique ①

PDU's

Donnée

Segment

Paquet

Trame

Bit

Modèle TCP-IP

Application ④
Transport ③
Réseau ②
Accès Réseau ①

⇒ Conception d'un nouveau protocole, on utilise OSI. car il a détaillé les couches

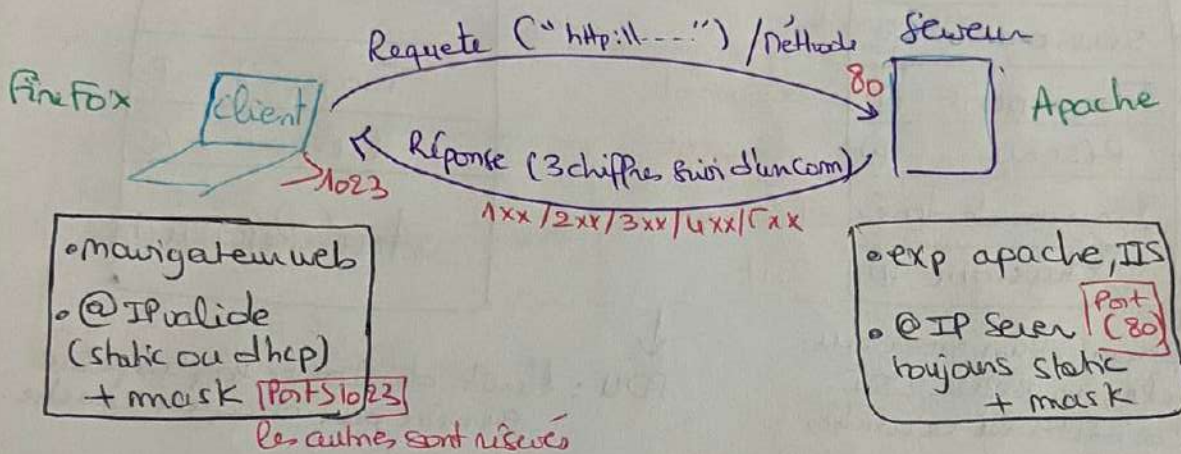
↓
PDU: l'unité de donnée la plus petite générée par chaque couche.

Service	Protocole	Port	Client	Serveur	Couche 5
Web	http / https	80 / 443	Firefox / Opera / Chrome	Apache / IIS / Lighttpd	TCP
Transfert de fichier	ftp	20 / 989 21 / 990	FileZilla Flash FXP Cute FTP	VSftpd Pro FTP Filezilla server	TCP
Messagerie	Smtp Pop Imap	25 110 / 995 143 / 993	Outlook Thunderbird Evolution	Postfix - dovecot exchange, Lotus / Domino	TCP
Résolution de nom	DNS	53	Resolver	BIND Rcld DNS	UDP TCP
Admin à distance	telnet ssh	23 22	cmd Shell Linux Putty	Telnetd openSSH	TCP
Auto Config IP	Dhcp	67	OS	Rcld dhcp Dhcpd	udp
Contrôleur de domaine	Ldap	389	Client / serveur messagerie	OpenLdap Active Directory	Tcp / UDP
Surveillance	Snmp Snmp trap	161 162	Nagios Cacti	Agent Snmp	Tcp / udp

L'ensemble du réseau local peuvent accéder à Internet par l'intermédiaire de la passerelle. Elle effectue donc le routage des paquets entre l'Internet et les machines du réseau local.

1/ Service Web : Protocole : http/https
80 443

- Les numéros de port sont affectés par la couche **transport**.
- Il utilise **TCP** qui donne la fiabilité lors de l'envoi du message.
- @IP par rapport au serveur doit être fixe **statique**.



www.esprit.lm

Sous Domain

SLD:

second Level domain

↳ TLD: top level domain

Les codes de statut:

Ces codes sont utilisés par le serveur pour informer le client de la manière dont sa requête a été traitée

1 x x

2 x x

3 xx

Lxxv

xxx

Rep st info

Succès

Redirection

Enemy Declist

Enemy Code Seven

Refus et en cours
de traitement

Reçu / compris /
accepté

- utiliser une autre ressource

1871

Methodes Http:

Get: Récupération d'une ressource localisée par un URI

Post: Envoyer des ressources vers le serveur, utiliser en cas de formulaire

Post: Envoyer des ressources
Put: " " " , utilisé pour faire téléchargement.

Put: " "

Virtual hosts:

Permet d'héberger plusieurs sites web sur un m^em serveur apache
Il peut être divisé sur: Norm / Port / IP.

80 www.espit1.fr
80 www.espit2.fr

80 www.espit1.fr
8080 www.espit4.fr

↳ création de interfaces dédiées et les affecter à www.espit1.fr
www.espit3.fr

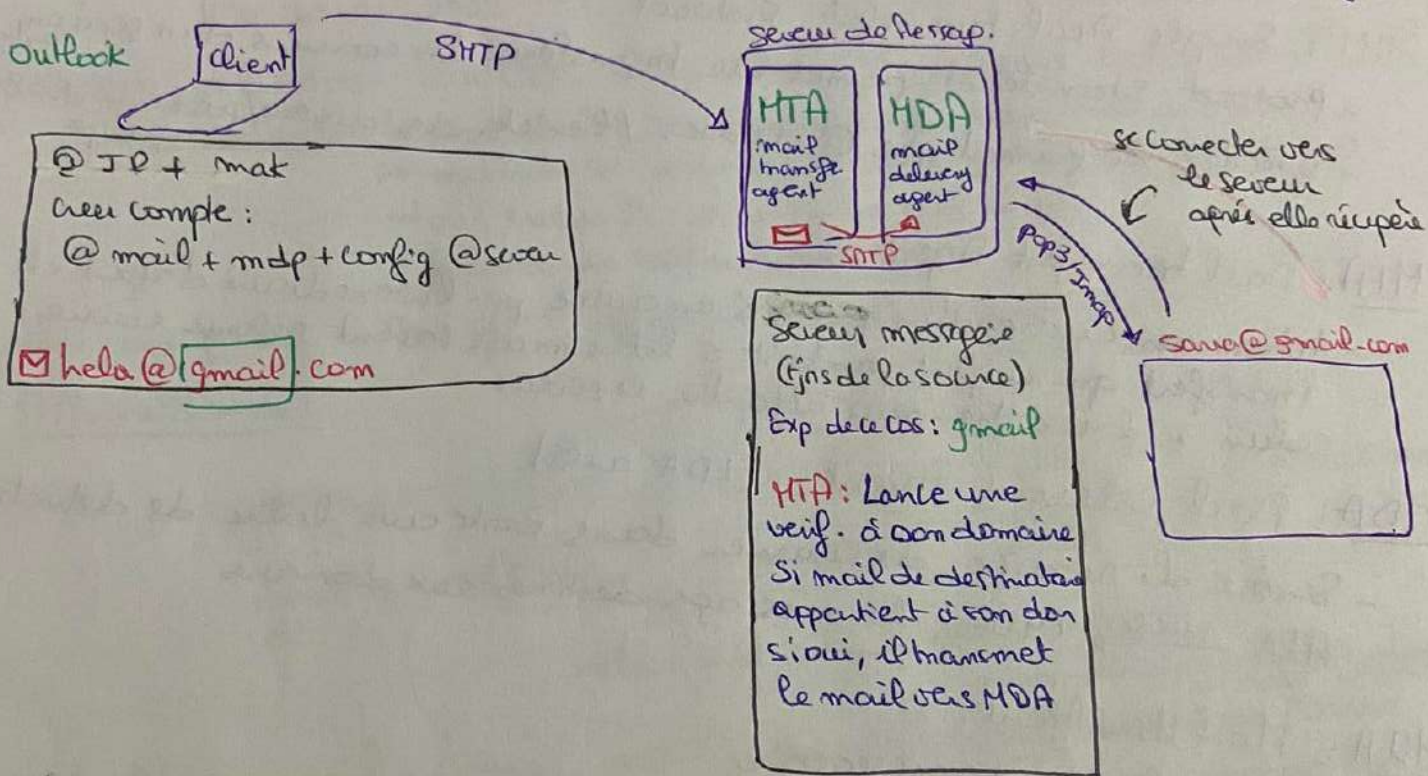
2/ Service de Messagerie: Protocoles: SMTP / POP / IMAP

Envoyer les e-mail: SMTP

(un protocole de transport spécifique)
TCP

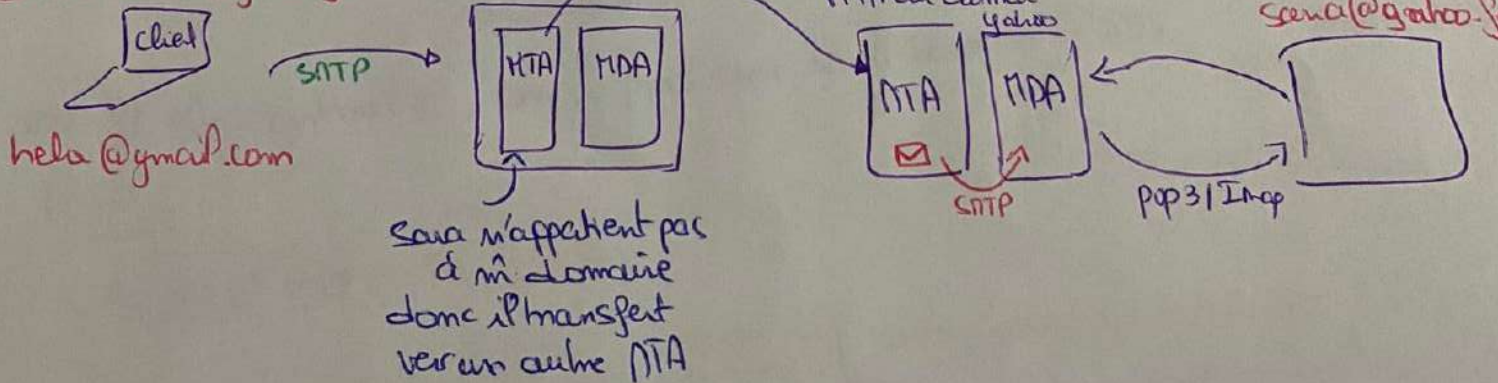
Reception: POP3 / IMAP

(protocoles de transport spécifiques)



2ème cas:

dest: sana@yahoo.fr



Chercher l'IP avec:

- 1) Résolution locale
- 2) Résolution DNS

POP3: - Boîte à lettre va être téléchargé en totalité sur Machine en locale
Post office (chercher / trier / lire) les emails en local ✓
Protocol - protocole hors ligne ✓ / utilisation min de ressource ✓
- nécessite une seule connexion juste pour télécharger la boîte à lettre
- limite d'utilisation de la cap de réseau X / Gestion de sauvegarde par l'uti X

IMAP: - Client a une connexion permanente vers le serveur ✓
Internet message - Pas de téléchargement → Boîte à lettre sur serveur
Access Protocol - Avoir un accès pour voir les mails.
- Gérer plusieurs comptes au même temps ✓
gère son espace X
travaille plus rapide X
pour les recherches
Certains mails lent à récupérer X

SMTP: Simple Mail Transfer Protocol
- Protocol standard permet de transférer les courriers d'un serveur à un autre
- spécifie le format des @ des usages / l'enlèvement des courriers / possibilité d'envoi groupé

MTA: Mail Transfer Agent
- Acheminer courrier d'un point à un autre par l'intermédiaire d'agent de transfert qui ne gère pas boîte à lettre mais savent relayer courrier d'un à un autre pour atteindre le serveur

MDA: Mail Delivery Agent (LDA aussi)
- Service de remise de courrier dans Boîte aux lettres de destinataire
MTA transfert MDA les messages destinés aux domaines.

MUA: Mail User Agent
- C'est le client de messagerie

3/ Service de Résolution de Nom : Protocole DNS (Domain Name System)

DNS : un système hiérarchique qui assure la correspondance entre Nom et @IP

Couche Transport : TCP (S/S) / UDP (C/S)

→ la mission de ce protocole est : offrir une correspondance / translation

[Nom domaine ⇒ @IP // Résolution directe
@IP ⇒ nom domaine // Résolution inverse]

Navigateur / Browser ⇒ App intermédiaire

Client DNS ⇒ **Resoluer** programme (Encapsuler la requête DNS dans Paquet UDP)

⇒ Serveur DNS ?? (plusieurs serveurs DNS) (IANA ⇒ 13 serveurs root)

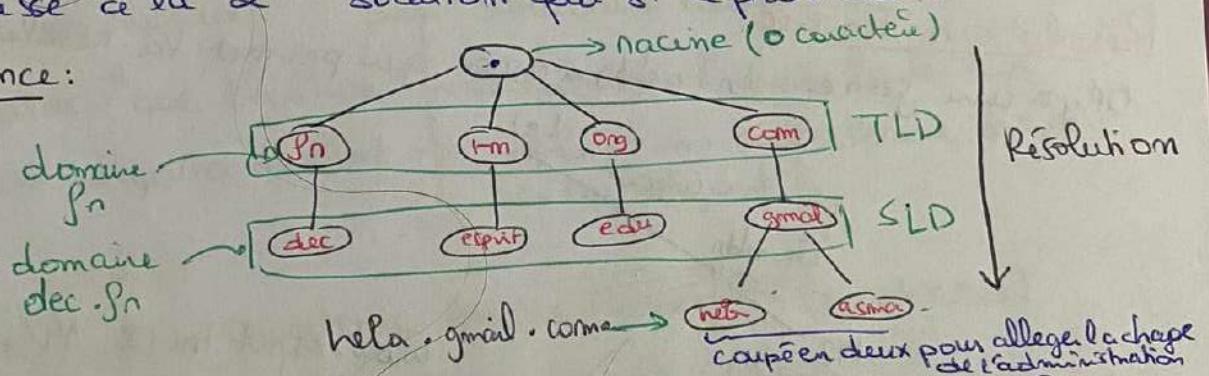
• Avant l'utilisation du DNS on a utilisé le fichier Hosts ⇒ **Resol Local**

Inconvénients :

- Nom sécurisé = accès par un lien non autorisé
- admin se charge de remplir fichier manuellement
- d'où erreur ? d'où tâche diff à gérer
- fichier de taille limitée

d'où on passe à la 2^e solution qui est le protocole **DNS**

Arborescence :



⇒ helo . gmail . com

va être ajouté
implicitement

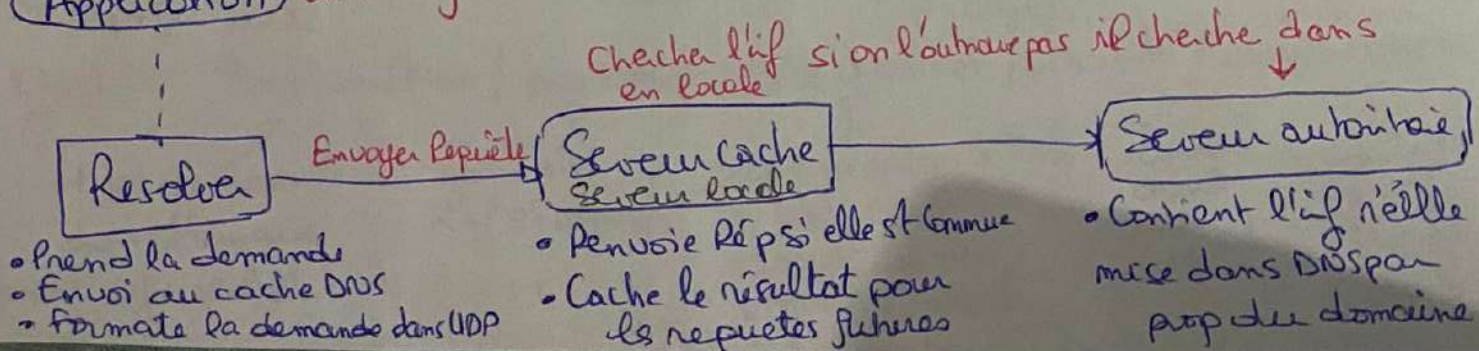
⇒ **FQDN** : Full Qualified Domain Name

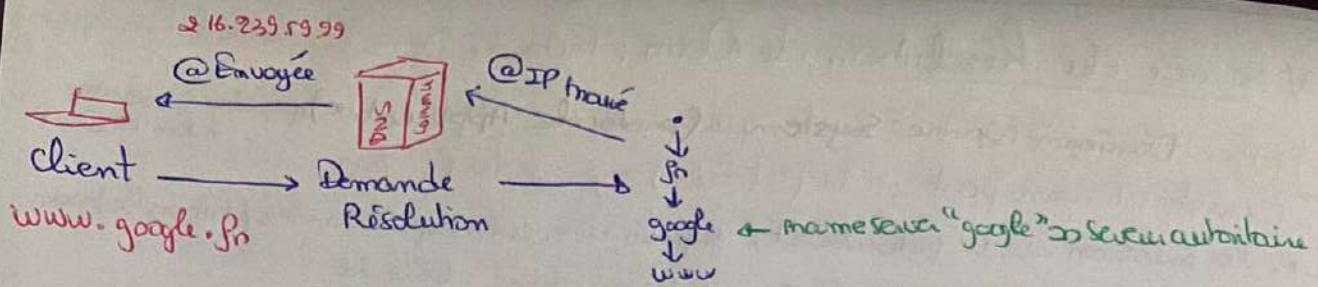
⇒ Nom de domaine qui donne la portion exacte de son nœud dans l'arborescence DNS en indiquant à la fois le domaine de niveau supérieur.

⇒ Il est centralisé, on a pas un seul serveur DNS.

Rôles :

Application ex NavigWeb





- Pour que le serveur DNS soit tjrs dispo:
On fait la disposition d'un serveur autoritaire (maître) et d'un serveur dupliqué (esclave) (copie de la base de données entre les deux)
→ Si un tombe en panne on utilise l'autre, pour le but d'une haute disponibilité de service.

Serveur autoritaire: possède une base d'inf d'une zone dont il a l'autorité ^{admin}

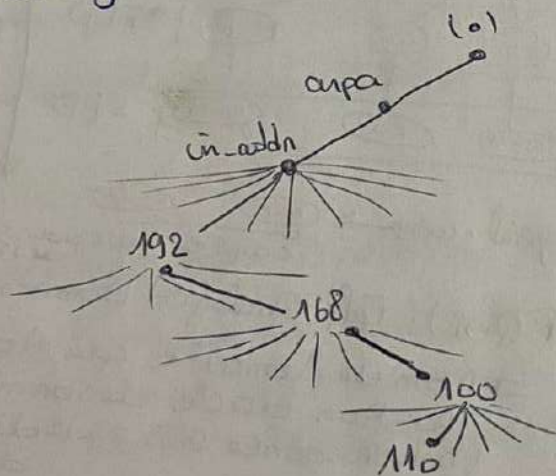
Serveur secondaire dupliqué: les données sont copiées ici (par protocole TCP)

Pas de diff entre 2 types de serveurs

Serveur peut être primaire ou secondaire

Résolution Inverse:

DNS a une zone in-addr-arpa qui permet la résolution inverse d'IP



www.esprit.fr = 192.168.100.110

Correspond au sous-domaine

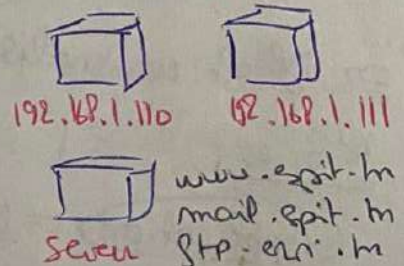
110.100.168.192 in-addr.arpa

mon nom de domaine sur @ diff

→ www.esprit.fr

DNS n'est pas bijective:

- Un service peut être déployé sur deux serveurs
- Sur une machine, je peux installer plusieurs services



4) Service Annuaire : Protocole : LDAP

- Problème d'accès
- Problème que les données sont distribuées sur l'ensemble du système et répliqués sur plusieurs serveurs.
- Problème de dépendance

⇒ Il faut avoir une seule session d'où on utilise le service annuaire

Annuaire :

Conteneur d'informations organisées
Un syst de stockage de données
organisé d'une manière hiérarchique

Dérivé des BDD relationnelles (annuaire → BDD)

Exp: téléphonie / carnet d'@ / catalogue de vente / DVS

Comparaison entre BDD et Annuaire :

Annuaire	BDD
Structure hiérarchique	Structure tabulaire
La lecture plus ↑ que l'écriture	L'écriture ↑ que la lecture
L'extensibilité d'ajout d'attribut	Obligé de reconstruire la base en cas d'ajout

Protocole LDAP:

LDAP: Lightweight Directory Access Protocol (avant X500)

Objectif: Fournir aux utilisateurs de nif facile
Permet aux users de n'aj eux-m leur nif
Rendre le nif accessible de façon contrôlée
Eviter la redondance d'nif (un seul annuaire)
Faciliter la gestion des postes de travail, equip. réseau

Caractéristique: Dynamique / souple / sécurisé / personnalisé

Terminologie: Objet / Attribut / Conteneur / Schéma

Modèle: Information / nommage / fonctionnel / sécurité / duplication.

• type de données dans l'annuaire

• comment définir ses objets

comment l'nif est organisé

(DIT)
de directory info tree

• interrogation de base
• n'aj d'nif
• Comparaison
• Authentification / contrôle

comment données sont protégées

comment la base est répartie entre serveurs

↑ constitue:
objet
attribut
conteneur

↓ décrit l'objet

↑ regroupe objets

5) Service Network Management: Protocole : SNMP ¹⁶¹₁₆₂

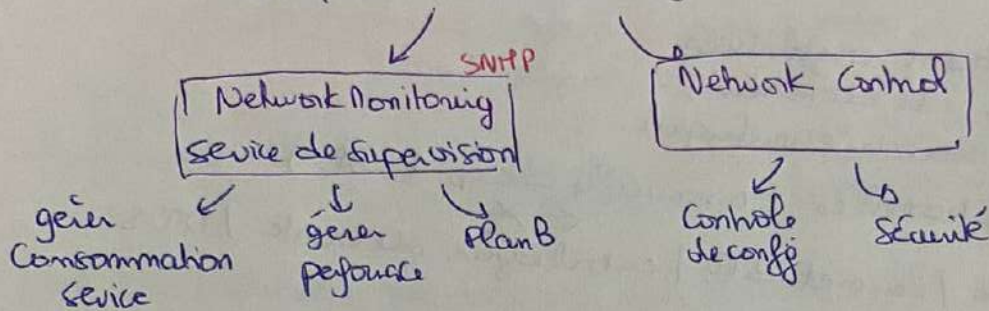
Problème: temps d'indisponibilité au niveau d'un serveur.
• Perte de données

→ ça touche la disponibilité du réseau.

d'où → ça touche Business (l'agent) / Perte productivité / Perte Crédibilité / touche la relation

Sol: Il faut déployer des mécanismes pour éviter ça

→ Appliquen Network management.



Rôle: - Atteindre la disponibilité, performance.
- Assurer la sécurité du Réseau IP
- Améliorer les paramètres réseau
- Faciliter l'extension à d'autres réseaux.

Il y a 5 filiales:

① Fault Management: localiser le problème en cas de panne
• découvrir / isoler / fixer le problème

② Configuration Management: tracer la config de serveurs
• contrôle sur la config des équipements.

③ Accounting Management: tracer le taux d'utilisation de diff services
• Si Nb d'utilisateurs ↗, serveur saturé donc on utilise filiale

④ Performance Management: Mesurer la performance
• Mesurer Débit / temps de rép / utilisation de mémoire

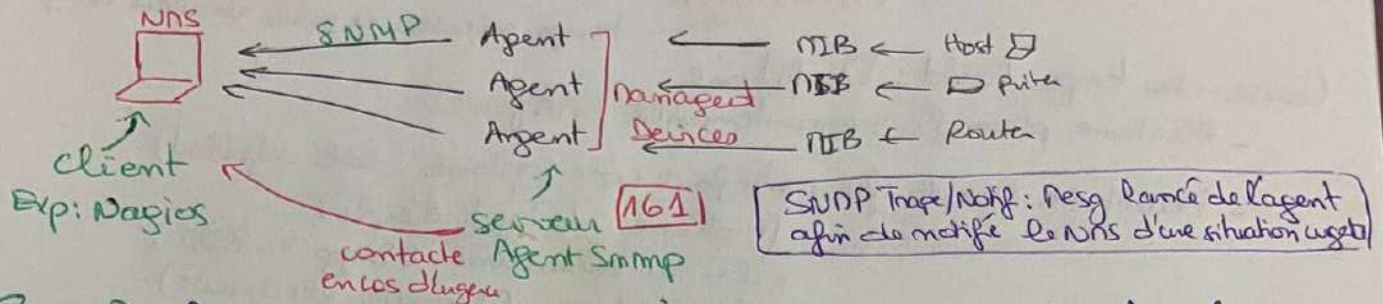
⑤ Security Management: Créer différemment username/pass
• gérer la sécurité

Composants Network Management:

Managed device: l'équipement qu'on veut contrôler, il collecte et management info.

Agent: Solution software qu'on va installer sur l'équipement

Network-management system (NMS): la solution qui va nous permettre de contrôler tout les équipements à travers admin.



SNMP: protocole standard qui est compatible avec tout les solutions

- Assurer la comm entre NMS et les Agents afin d'assurer le contrôle et la supervision de équipements
- CIP / CIS remplacé par SNMP (3 versions)

Rôle: permet aux administrateurs réseau de:

- gérer les performances du réseau
- trouver et résoudre les problèmes de réseau
- planifier la croissance du réseau

Composants: 1/ Network Manag station (manager) : Equipement

2/ Network Manag System (NMS) : App dont laquelle on va installer

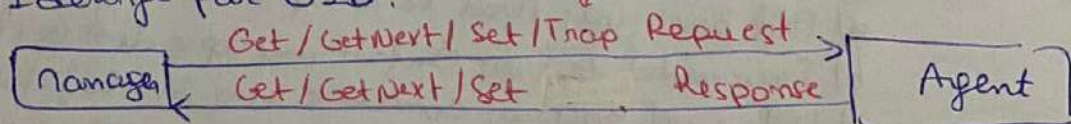
3/ Agent : solution sof.

4/ Management Inf Base (MIB) : Base de Données

que chaque agent va l'utiliser pour stocker les inf relative à son équipement

- Stockés en forme d'objet et ça va être une adresse unique et Identifié par OID.

Messages SNMP:



Avantages

- Conception et mise en œuvre simple
- Extensibilité (peut être rajouté pour répondre aux besoins futurs)

Inconvénients

- Faible de sécurité
- Ce n'est pas un protocole particulièrement efficace
- La bande passante gaspillée par des inf inutile

6/ Service HA et LB:

Problème: Indisponibilité de service

- Il faut avoir la haute disponibilité, le système doit être performant, fonctionnel même si il y a des temps d'arrêt.
elle est calculé en pourcentage:

$$\text{Disp (o/o)} = \frac{\text{MTTF}}{(\text{MTTF} + \text{MTTR})}$$

← système fonction normalement jusqu'à temps d'arrêt

← temps nécessaire pour récupérer le système

Cause de temps d'arrêt: Downtime:

- Downtime planifié: maintenance / NAj
- Downtime Non planifié:
 - Défaut côté système (cable débranché)
 - (Problème de Syst. d'explo)
 - Erreur humaine (Supp. la BDD)
 - Site outages (coupure électricité)

Solution pour Réduire Downtime:

- Eliminer Single points of failure
- Détecter les points de défaillance. (BDD / serveur / alimentation / cnx internet)

Comment: Redondance (2^e copie identique à la 1^{re})

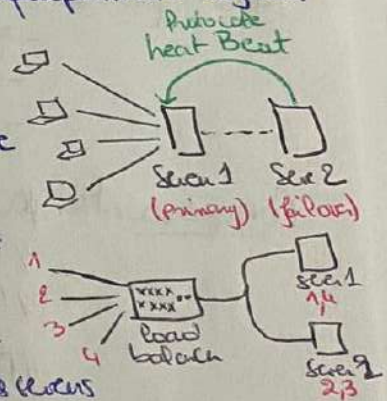
① Clustering: Créer un cluster composé d'un ensemble d'équipement ayant même caractéristique. (meilleur pour les serveurs)

a/ active-passive: constitué de 2 serveurs

- le 2^e serveur prend la relève lorsque le 1^{er} tombe en panne

b/ active-active: constitué de 2 serveurs active au même temps

- Installer un Load Balancer qui est un répartiteur de charge, il va recevoir les requêtes client et il distribut vers les serveurs



→ c'est avec le protocole Heart Beat, que Serveur 2 va détecter l'indispo du 1^{er} serveur il assure la synchronisation entre les deux serveurs.

② Backup: Côté data est meilleur (BDD)

a/ Full Back-up: Avoir la même copie à chaque Backup

b/ Incremental Backup: On fait une copie de tous les éléments modifiés depuis la précédente Sauvegarde

c/ Differential Backup: conserve une copie unique des fichiers modifiés depuis la dernière sauvegarde

③ Replication: Copie et maintenance des objets de la BDD

④ Disaster Recovery: Catastrophe naturelle (Avoir deux sites)

Tout le infra être répliqué sur un autre site

⑤ Load Balancing: Minimiser le temps de réponse (le temps entre l'envoi et la réception de requête)

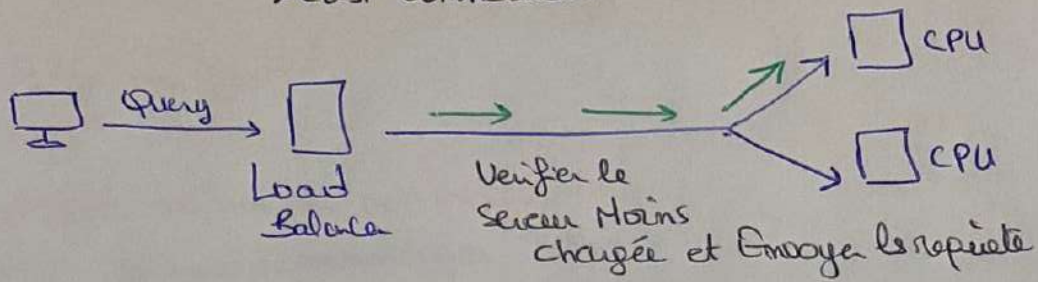
Facteurs du temps de rép: Bande passante / serveur surchargée.

deux sol: scale in: améliorer les performances des équipements

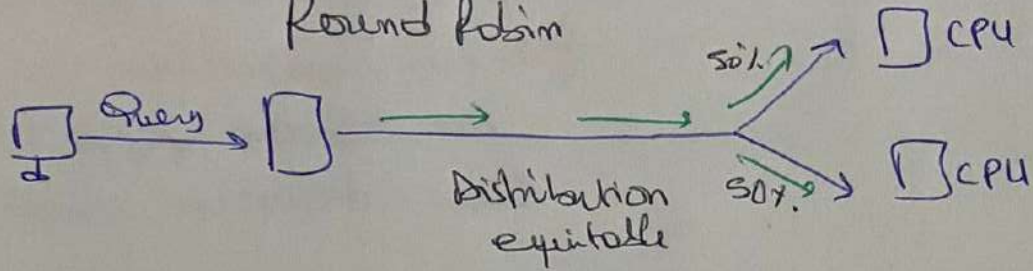
scale out: avoir plusieurs copies de même ressource

Algorithme Load Balancing:

Least Connections



Round Robin



TP 5 : Annuaire

1/ Installer Openldap:

```
# yum install openldap openldap-servers openldap-clients
```

```
# slapdpasswd
```

New pass:

Re-enter:

3SSH#A? - - - - -

```
# gedit /etc/openldap/slap.d/cn=config.ldif
```

```
[ olcrootpw: 3SSH#A? - - - - -
```

```
# cd /etc/openldap/slap.d/cn=config
```

```
# gedit olcDatabase
```


TP 6: Nagios:

1/ Install configure Nagios

```
# yum install httpd
# yum -y install php php-mbstring php-pecl
# gedit /var/www/html/index.php // creation du page php
# rpm -Uvh --- // installation Nagios
# gedit /etc/httpd/conf.d/nagios.conf
```

```
Order deny allow
Deny from all
Allow from 127.0.0.1 10.0.0.0/24
2 step add nagios admin user
```

```
# htpasswd /etc/nagios/passwd nagiosadmin
```

New pass : - - -

```
# /etc/rc.d/init.d/nagios start
# /etc/rc.d/init.d/nagios restart
# chkconfig nagios on
```

→ Faire l'accès à Nagios:

2/ Add Monitoring Target

```
# gedit /etc/nagios/nagios.cfg
```

```
cfg_dir = /etc/nagios/severs
```

```
# mkdir /etc/nagios/severs
```

```
# chgrp nagios /etc/nagios/severs
```

```
# chmod 770 /etc/nagios/severs
```

```
# gedit /etc/nagios/severs/mode01.cfg
```

```
# /etc/rc.d/init.d/nagios reload
```


TP7: HA et LB

1/ Install haproxy

```
# yum -y install haproxy
```

2/ Configure Haproxy

```
# gedit /etc/haproxy/haproxy.cfg
```

```
mettue@connecte.
```

3/ Start haproxy Service

```
# service haproxy start
```

```
# chkconfig haproxy on.
```


TP1 : Config de Base

Ex1:

1/ Connecter en root

\$ su

2/ Désactiver network manager

Service NetworkManager stop

chkconfig NetworkManager off

3/ Activer netManager par def

chkconfig network on

Service network start

Ex2:

1/ Donner @ Mac et @ IPv4 :

ifconfig

@ Mac

Interface eth0 --- #Waddr BC: 16:4r:94: b6:64
lo inet addr: 10.253.0.1 @ IPv4

dhclient

2/ Config @ IP statique :

if down eth0

// désactivation de l'interface

gedit /etc/sysconfig/network-scripts/ifcfg-eth0

Device=eth0

BootProto=static

OnBoot=yes

IPADDR=... Netmask=...

} Config de fichier

if up eth0 // activation

/etc/init.d/network restart // redémarrage du service

Sous Debian :

#if down eth0

#gedit /etc/network/interfaces

```
auto eth0
iface eth0 inet static
address 192.168.10.12
netmask .255.255.255.0
```

#if up eth0

/etc/init.d/networking restart

3/ Creation interface alias :

#ifconfig eth0:1 192.168.1.1 netmask 255.255.255.0

4/ Passerelle par def: (Gateway)

#route -m //afficher la table de routage

#route add default gw 192.168.0.254 //ajouter gateway

#gedit /etc/sysconfig/network

```
[ Gateway = 192.168.109.254
  Gateway Dev = eth0
```

} sauvegarder l'@ Gateway.

5/ Config du nom de machine

#hostname

#gedit /etc/hostname

TP2 : Web :

1/ Installation apache on CentOS:

```
# yum install httpd
# service httpd start
# chkconfig httpd on.
```

2/ Setup name

```
# mkdir /var/www/espit1.com/
# mkdir /var/www/espit2.com/
# vi /var/www/espit1.com/index.html // creation de la page de test
```

```
<html>
  <head>
    <title>www.egrit.com</title>
  ..
</html>
```

```
# vi /etc/httpd/conf/httpd.conf
NameVirtualHost 192.168.0.100:80 // Décommenter ce ligne
# j'ajoute des lignes de commandes
<VirtualHost 192.168.0.100:80>
    ServerAdmin webmaster@espit1.com
    DocumentRoot /var/www/espit1.com
    ServerName www.egrit1.com
    ErrorLog _____
    CustomLog _____
</VirtualHost>
```

```
# httpd -t
# service httpd start
# chkconfig httpd on
```

```
# gedit /etc/hosts
```

// ouvrir hosts et ajouter ces lignes.

```
192.168.0.100 www.egrit1.com
192.168.0.100 www.egrit2.com
```


3/ Set up IP address

```
# ifconfig eth0:1 192.168.0.101 netmask 255.255.255.0
```

```
# vi /etc/httpd/conf/httpd.conf
```

j'ajoute la m^e ligne pour `www.epit3.com`
<VirtualHost 192.168.0.101:80>

4/ Set up Port

```
# vi /etc/httpd/conf/httpd.conf
```

Listen 8080

j'ajoute la m^e ligne pour `www.epit4.com`
<VirtualHost 192.168.0.100:8080>

TP3: Hessagenie:

1/ Creation Users:

```
# useradd -m user1
```

passwd user1

2/ Install postfix server:

#yum remove sendmail

yum install postfix.

3/ Configuring Iptables:

- * `gedit` `/etc/sysconfig/iptables`

```
-A Input -m state --state New -m tcp -p http 25-j Accept
```

11 Par défaut les ports 25/110/143 sont fermé, donc il faut les ouvrir

4/ Configuring postfix :

gedit /etc/postfix/main.cf

11 décommenter des lignes

myhostname = esprit.com // nom du host

my host name = \$ my domain // nom du domaine

inet-interfaces = all // set network interfaces par que postfix peut utiliser

mydestination = \$

```
mydestination = $  
mynetworks = 192.168.1.0/24, 127.0.0.0/8
```

home-mailbox = nail dir/

5/Start service:

service postfix start

chkconfig postfix on.

6/ Test postfix using the command telnet:

telnet local host smtp.

ehlo local host

mail from: <user1@spit.com>

ncpt to: <user2@spit.com>

data quit

Check mail

```
# cd /home/user1/maildir/new/  
# ls  
# cat 1360236966.VSd00I35a8n1812r6..egrit.conf
```

7/ Install dovecot et config: # yum install dovecot

```
8/ # gedit /etc/dovecot/dovecot.conf  
    protocols = imap pop3 smtp // de commenter ce ligne  
# gedit /etc/dovecot/conf.d/10-mail.conf  
    mail_location = maildir:~/Maildir // Specifie le format et  
# gedit /etc/dovecot/conf.d/10-auth.conf // la localisation de  
    disable_plaintext_auth = no // Chapeau tout mail.  
    auth_mechanisms = plain login  
# gedit /etc/dovecot/conf.d/10-master.conf  
    user = postfix  
    group = postfix  
# Service dovecot start  
# chkconfig dovecot on
```

8/ Tester :

```
# telnet localhost pop3
```

```
...  
user user1
```

```
...  
pass user1
```

```
...  
list
```

```
...  
retrn 1
```

```
...  
quit
```


TP 4 : DNS :

1) Installation du Service:

```
# yum install bind  
# yum install caching-nameServer
```

2) Configuration du service maître:

```
# gedit /etc/named.conf
```

```
listen-on port 53 3 127.0.0.1; <@ IPv4 >;};
```

```
Zone "esprit.in" IN {
```

```
type master;
```

```
file "/var/named/db.esprit.in" <the database file>;
```

```
allow-update {none};
```

```
Zone "255.168.192.in-addr.arpa" {
```

```
type master;
```

```
file "/var/named/db.esprit.in"
```

```
allow-update {none};
```

```
};
```

Zone directe

Zone inverse

3) Config Zone directe:

```
# gedit /etc/named/db.esprit.in
```

```
ms IN A <name for IP address>  
host1 IN A <IP address>  
host IN A <IP address>
```

```
# named-checkzone <zone name> <path to the zone config file>
```

```
# named-checkzone esprit.in /var/named/db.esprit.in
```


4/ Config zone inverse

192.168.10.2 IN NS ms.epit.in
2 IN PTR ms.epit.in
< > PTR < host name >
< > PTR < host name >

named check-zone epit.in /etc/named/db.epit.in

ms lookup web.epit.in