

PAQUELET Etienne VADAM Julien ZERRAR Yanis MARCHAND Théo LUCAS Antoine

# SAE303 - Concevoir un réseau multi-sites - Documentation technique

---



Messieurs BOUILLET et LECOQ - Du lundi 5 février au vendredi 10 février



---

<b>1. Introduction.....</b>	<b>3</b>
a. Présentation du client et du projet.....	3
b. Méthodologie mise en place pour le bon fonctionnement du projet.....	3
c. Objectif du projet.....	3
<b>2. Schémas du réseau et de son infrastructure.....</b>	<b>4</b>
a. Inventaire.....	4
b. Adressage IP.....	6
c. Topologie logique.....	6
<b>3. Routage.....</b>	<b>6</b>
a. Simulation Packet Tracer.....	7
b. Configuration Magasin, Showroom et Siège.....	7
i. Configuration du Switch :.....	8
ii. Configuration Edge routeurs :.....	9
iii. Configuration PCs.....	12
c. Configuration coeur de réseau.....	14
i. Protocoles utilisés :.....	14
ii. Mise en place des protocoles :.....	16
<b>4. Mise en place du serveur d'infrastructure.....</b>	<b>20</b>
a. Configuration DNS.....	20
i. Configuration serveur DNS.....	20
ii. Configuration client DNS.....	21
b. Configuration Active Directory.....	22
c. Configuration RADIUS.....	24
d. Configuration des Services de Bureau à Distances.....	26
<b>5. Mise en place du serveur d'applications.....</b>	<b>28</b>
a. Configuration de la Messagerie.....	28
b. Configuration du serveur NextCloud.....	33
c. Configuration du serveur Téléphonique.....	36
d. Configuration du Service Wifi avec portail captif Pfsense.....	38
<b>6. Sécurisation des services réseaux.....</b>	<b>41</b>
<b>7. Mise en place d'Objets Connectés.....</b>	<b>45</b>
<b>8. Problèmes rencontrés.....</b>	<b>49</b>
<b>9. Conclusion.....</b>	<b>51</b>
<b>10. Annexes.....</b>	<b>51</b>

---

---

# 1. Introduction

## a. Présentation du client et du projet

Le client est l'entreprise Beerok, leader et spécialiste français des chaussures sportives. Cette entreprise possède actuellement 48 magasins en France et compte une centaine d'employés. Le projet demandé par le client est le suivant : restructurer l'infrastructure réseau et informatique de l'ensemble de ces sites situés en France.

## b. Méthodologie mise en place pour le bon fonctionnement du projet

La méthodologie que nous avons adoptée tient d'une gestion de projet dite *AGILE*. Cette approche de la gestion de projet permet d'apporter de la souplesse et de la performance à la gestion de projet. Cette méthode est centrée sur l'humain et la communication et pour ce faire, elle utilise 3 piliers : Transparence, Inspection, Adaptation. Afin de mener à bien ce projet, nous avons mis en place une méthode KanBan. La méthode Kanban consiste à créer différents tableaux où nous y mettons toutes les tâches à faire, les tâches en cours, terminées, en test et les tâches bloquées. Cette méthode nous permet donc de savoir où nous sommes dans la progression du projet, qui s'occupe de réaliser une action afin d'avancer dans le projet de façon rapide, efficace et coordonnée. Voici le tableau Kanban que nous avons utilisé : nous l'avons mis en place sur [Trello](#).

## c. Objectif du projet

L'Objectif de ce projet est multiple : nous devons restructurer l'infrastructure réseau de l'entreprise Beerok et restructurer également son infrastructure de services informatique tout en implémentant différents services réseaux permettant ainsi son indépendance vis-a-vis d'opérateurs ou de cloud extérieurs. Nous devons également implémenter des solutions d'objets connectés ainsi que la sécurisation de ces services.

---

## **2. Schémas du réseau et de son infrastructure**

### **a. Inventaire**

Les appareils nécessaires au bon fonctionnement de ce projet sont les suivants :

- 2 serveurs DHCP
- 8 routeurs
- 3 switches
- 4 PCs
- 3 téléphones
- 1 serveur de messagerie
- 1 serveur de bureau à distance
- 1 serveur Active Directory
- 1 serveur RADIUS
- 2 bornes wifi
- 3 ESP8266
- 1 serveur de certificat
- 1 serveur de téléphonie
- 1 serveur de stockage NextCloud
- 1 serveur MQTT
- 1 Portail Captif
- 1 tablette

---

Ces appareils sont répartis selon les sites. La répartition des services selon les sites est la suivante :

Les services se situant au Siège de Meaux sont les suivants :

- Serveur Active Directory
- Serveur RADIUS
- Serveur de messagerie
- Serveur de certificats
- Serveur MQTT
- Serveur cloud NextCloud
- Serveur de Bureau à Distance
- 1 switch
- 1 routeur
- 1 téléphone
- 1 borne wifi

Les services se situant au Showroom de Paris sont les suivants :

- 1 serveur DHCP
- 2 PCs
- 1 borne wifi
- 1 routeur
- 1 switch
- 1 téléphone
- 1 portail captif
- 1 tablette

Les services se situant dans chaque magasin seront les suivants :

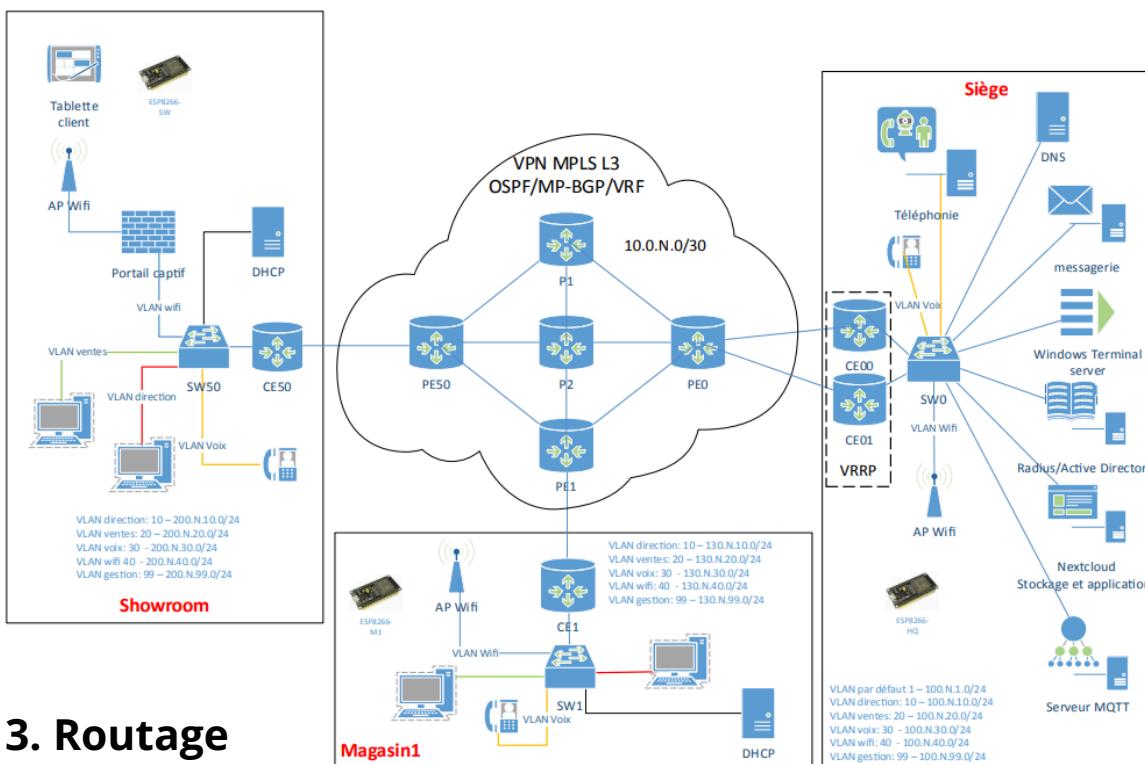
- 1 serveur DHCP
- 2 PCs
- 1 borne wifi
- 1 routeur
- 1 switch
- 1 téléphone

## b. Adressage IP

Un plan d'adressage IP est un document montrant comment les adresses IP sont réparties entre les périphériques en fonction de l'architecture ou de la topologie du réseau d'une manière qui prend en charge les services requis. Le réseau de l'entreprise possède ainsi l'adressage IP que vous trouverez en annexe de ce rapport.

## c. Topologie logique

Une topologie logique désigne la manière dont un réseau transfère les trames d'un nœud à l'autre. Voici la carte du réseau représentant les différents services que l'on va mettre en place au sein de l'entreprise Beerok :

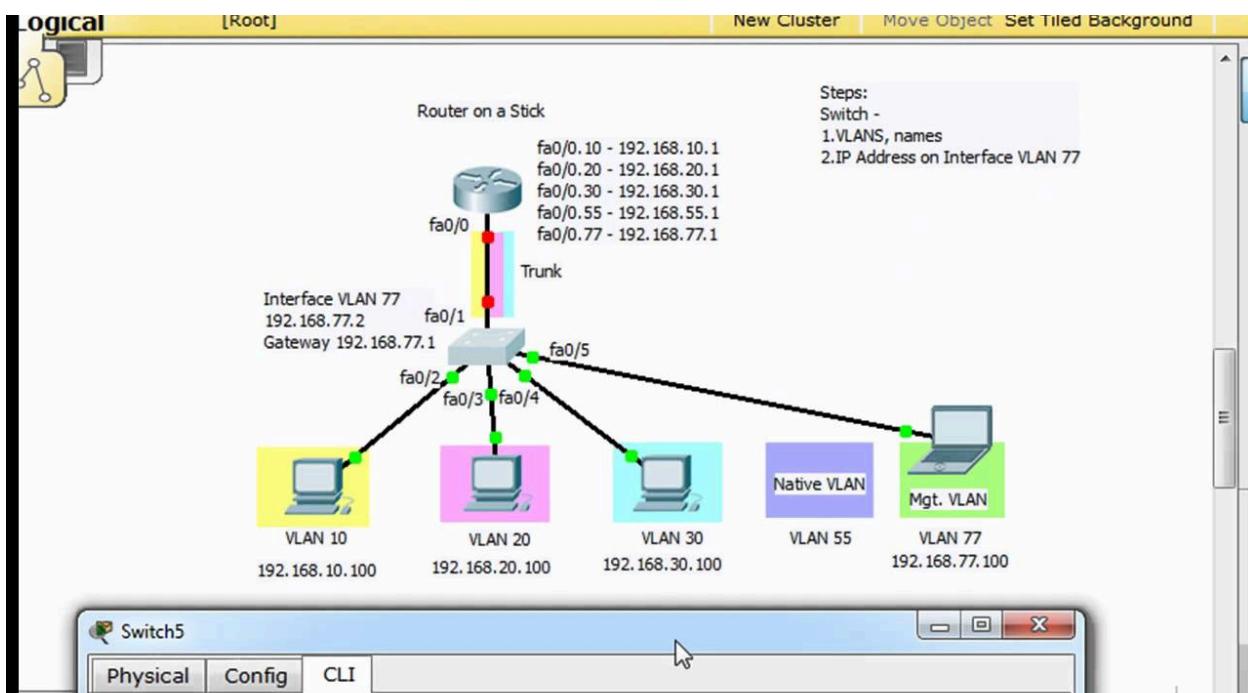


## 3. Routage

### a. Simulation Packet Tracer

## i. Simulation du réseau des sites distants

Suivant la topologie logique, nous avons mis en place une simulation du réseau présent dans un magasin avec tous les VLANs afin de tester les différentes solutions pour le réseau de l'entreprise. Après des recherches sur internet ainsi qu'une formation, nous avons implémenté la solution du Router-on-a-Stick. Cette configuration utilise une seule interface du switch et du routeur pour faire communiquer plusieurs VLAN. Ainsi, le réseau d'un magasin comprenant 5 VLANs, cette solution est donc idéale afin de faire communiquer ces VLANs entre eux à travers le magasin. Nous avons donc décidé de simuler cette configuration sur packet tracer afin de tester et de valider cette solution. Nous avons également implémenté le serveur DHCP sur le routeur de cette configuration pour les tests.



Configuration en mode Router-on-a-Stick

---

## b. Configuration Magasin, Showroom et Siège

La solution Router-on-a-Stick ayant été testée et validée sur packet tracer, nous avons copié cette configuration sur notre réseau local du magasin, du showroom et du siège, tout en prenant en compte le nombre de vlan des sites. Voici comment nous avons configuré notre switch :

### i. Configuration du Switch :

Nous avons tout d'abord configurer les différents VLAN sur le switch ainsi que leurs interfaces :

**SW1>en**

**SW1#conf t**

**SW1(config)#vlan 10**

**SW1(config-vlan)#name direction**

Les commandes ci-dessus permettent la création des VLANs ainsi que l'assignement de leurs noms respectifs.

Afin de leur assigner un port, nous avons effectué les commandes suivantes et nous mettons ces ports en mode access puisque nous relierons un vlan par port :

**SW1#conf t**

**SW1(config)#interface GigabitEthernet1/0/1**

**SW1(config-if)#switchport mode access**

**SW1(config-if)#switchport access vlan 10**

---

Nous avons réalisé cette configuration pour les 4 premiers ports Gigabit Ethernet, assignant ainsi les VLAN Direction, Ventes, Voix et Wifi. Le VLAN Gestion n'étant pas relayé après le switch, aucune interface ne lui est désignée. Cependant, une adresse IP est assignée au VLAN afin que le service informatique puisse se connecter en ssh à distance afin de gérer le switch. Pour ce faire, nous avons réalisé les commandes suivantes :

**SW1#conf t**

**SW1(config)#interface vlan 99**

**SW1(config-if)#ip addr 130.2.99.253 255.255.255.0**

Une fois les VLAN créés et assignés aux ports du switch, nous les ajoutons maintenant en mode trunk sur le port Gigabit Ethernet 1/0/24, qui sera relié au Edge Routeur. Le mode trunk est un mode de vlan permettant la communication de plusieurs VLAN différents à travers un seul lien physique, comme un tunnel, alors que le mode access permet la communication que d'un seul VLAN par lien physique.

**SW1#conf t**

**SW1#(config)#interface GigabitEthernet1/0/24**

**SW1(config-if)#switchport mode trunk**

**SW1(config-if)#switchport trunk allowed vlan 10,20,30,40,99**

Afin de vérifier la configuration, nous avons effectué des commandes de vérification telles que **sh interfaces trunk**. En pièce jointe et en annexe de cette documentation technique, vous trouverez les captures de configuration résultant de l'exécution de la commande "show run" ainsi que d'autres commandes afin de présenter la configuration que le switch devrait avoir après l'application des commandes ci-dessus.

---

## **ii. Configuration Edge routeurs :**

Afin de mettre en place la configuration Router on a stick sur le réseau réel, nous avons tout d'abord créé des sous interfaces à l'interface Gigabit Ethernet 0/1 puis nous l'avons activée :

**CE1#conf t**

**CE1(config)#interface GigabitEthernet0/1**

**CE1(config-if)#description CONNECTEE AU SWITCH**

**CE1(config-if)#no shutdown\***

Une fois cela fait, nous créons les sous interfaces et leur ajoutons leurs adresses IP :

**CE1#conf t**

**CE1(config)#interface GigabitEthernet0/1.10**

**CE1(config-if)# ip address 130.2.10.254 255.255.255.0**

Nous répétons ces commandes pour chaque sous interface, une sous interface par VLAN. Ainsi, nous arrivons à 5 sous interfaces. Ensuite, nous avons appliqué une encapsulation *dot1q* à chaque sous-interface, ce qui permettra d'attribuer des balises aux trames conformément à la norme [IEEE 802.1q](#). Cela facilitera la reconnaissance des trames pour les transmettre au VLAN approprié et ainsi segmenter le réseau.

**CE1#conf t**

**CE1(config)#interface GigabitEthernet0/1.10**

**CE1(config-if)#encapsulation dot1Q 10**

---

Afin que le routeur du magasin, du showroom ou du siège communiquent entre eux via le coeur de réseau, il a fallu configurer l'interface qui est relié au cœur de réseau suivant le plan d'adressage IP ainsi :

**CE1#conf t**

**CE1(config)#interface GigabitEthernet0/0**

**CE1(config-if)#ip address 10.0.2.26 255.255.255.252**

**CE1(config-if)#no shutdown**

Conformément au cahier des charges mis en vigueur au démarrage du projet, le site du Showroom et du Magasin devaient comporter un serveur dhcp afin de fournir une adresse IP et des informations réseau nécessaires au bon fonctionnement des clients.

Pour réaliser cette tâche, nous avions la possibilité d'utiliser une machine réelle fournissant ce service à l'intérieur du réseau. Mais finalement il était plus simple de configurer ce service directement sur les routeurs. Comme l'indique le plan d'adressage IP, chaque vlan du réseau a son propre réseau IP, nous avons donc eu besoin de réaliser un pool d'adresses IP pour chaque vlan du réseau avec ses informations : adresses disponibles, masque du réseau, passerelle par défaut, nom de domaine (beerok.com) et l'adresse IP du serveur DNS :

**CE1#conf t**

**CE1(config)#ip dhcp pool vlan\_10**

**CE1(dhcp-config)#network 130.2.10.0 255.255.255.0**

**CE1(dhcp-config)#domain-name beerok.com**

**CE1(dhcp-config)#default-router 130.2.10.254**

*\*Par souci de compréhension, toutes les commandes du switch et du routeur présentées dans cette partie ont été implémenté dans le réseau du site Magasin*

---

### iii. Configuration PCs

Afin de pouvoir faire communiquer différents PC sur le réseau, il nous a fallu configurer les PC pour qu'ils reçoivent une adresse IP par DHCP. Pour cela, nous ouvrons le fichier de configuration réseau présent dans **/etc/network/interfaces** et nous modifions le fichier comme suit :

**auto eth0**

**allow-hotplug eth0**

**iface eth0 inet dhcp**

Puis nous effectuons la commande **/etc/init.d/networking restart** qui redémarrera les services réseau. Afin d'obtenir une adresse IP, nous stoppons tout d'abord le précédent processus DHCP avec la commande **dhclient -r** puis nous recréons ce processus avec la commande **dhclient -v** en demandant une adresse IP via DHCP.

```
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 17
DHCPOFFER of 130.2.20.2 from 130.2.20.254
DHCPREQUEST for 130.2.20.2 on eth0 to 255.255.255.255 port 67
DHCPACK of 130.2.20.2 from 130.2.20.254
bound to 130.2.20.2 -- renewal in 37356 seconds.
root@rt:~#
```

*Demande d'adresse IP via DHCP*

Le DHCP est un protocole client/serveur qui fournit automatiquement une adresse IP et d'autres informations de configuration telles que les informations de DNS, de passerelle par défaut, indispensable pour que le DHCP fonctionne.

---

Une fois cela réalisé, nous pouvons ping entre deux appareils qui n'appartiennent pas au même vlan, toutes les communications de ce type passent donc obligatoirement par le routeur du site. Une méthode simple pour vérifier cela est d'utiliser la commande *traceroute* ou *ping* par exemple.

```
root@rt:~# ping 130.2.10.2
PING 130.2.10.2 (130.2.10.2) 56(84) bytes of data.
64 bytes from 130.2.10.2: icmp_seq=1 ttl=63 time=0.590 ms
64 bytes from 130.2.10.2: icmp_seq=2 ttl=63 time=0.570 ms
64 bytes from 130.2.10.2: icmp_seq=3 ttl=63 time=0.547 ms
^C
--- 130.2.10.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 46ms
rtt min/avg/max/mdev = 0.547/0.569/0.590/0.017 ms
```

*Ping entre deux vlan*

---

## c. Configuration cœur de réseau

### i. Protocoles utilisés :

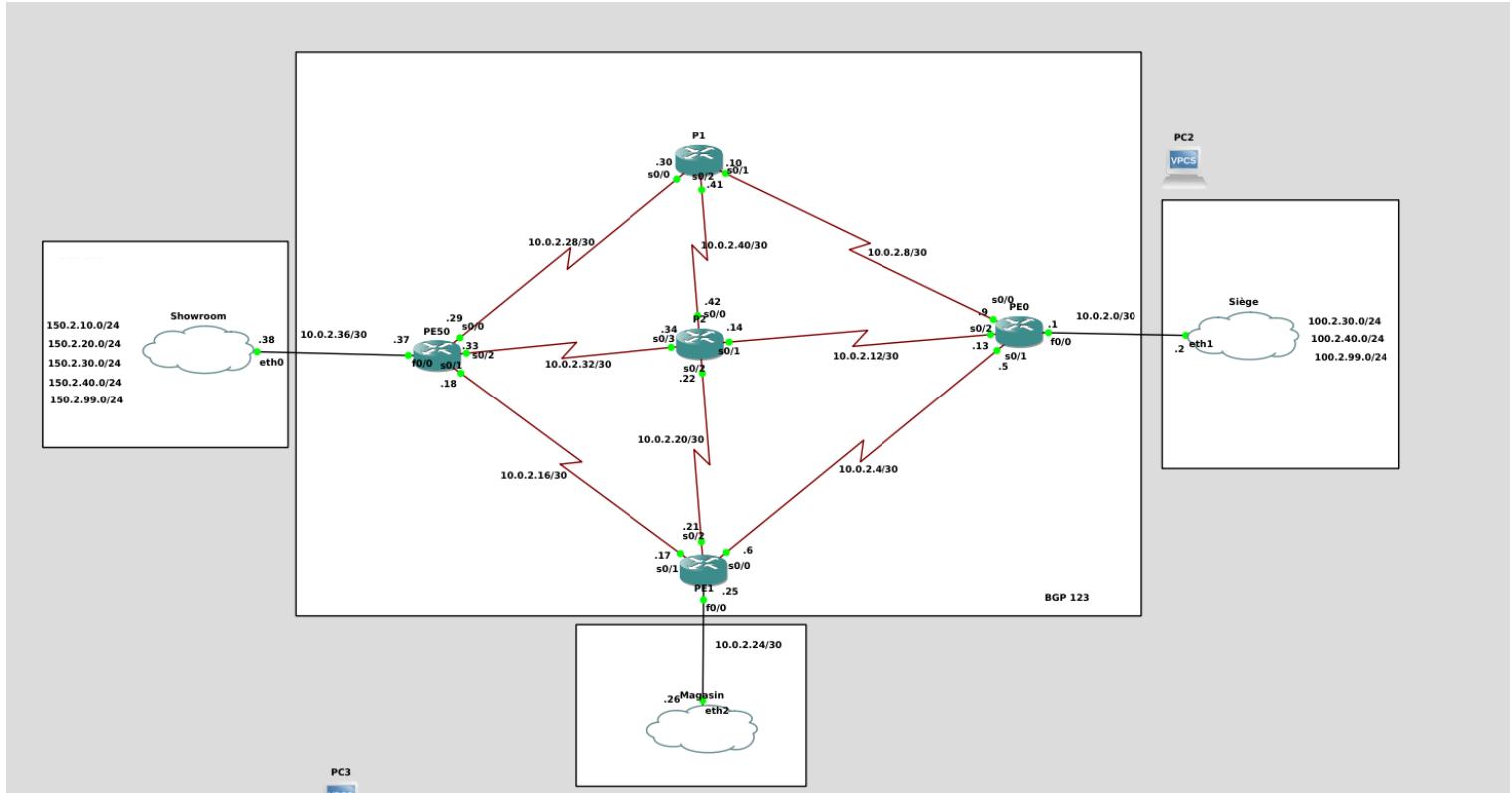
Afin de réaliser la configuration de notre cœur de réseau simulant un opérateur télécom, nous avons utilisé trois protocoles :

- OSPF : OSPF (Open Shortest Path First) est un protocole dynamique de routage interne IP (IGP) de type « à état de liens ». C'est-à-dire que ce protocole collectera les informations de tous les liens au sein d'une zone (area) et choisira le meilleur chemin qu'il connaît depuis sa table de routage vers la destination du paquet. Ce protocole enverra également des messages de mise à jour de ces tables de routage lorsqu'il détectera un changement de topologie et toutes les 30 secondes. Ce protocole nous a permis de mettre en commun les tables de routage de tous les routeurs du cœur de réseau exclusivement.
  
- BGP : BGP (Border Gateway Protocol) est un protocole de routage externe (EGP) notamment utilisé sur internet pour faire communiquer différents réseaux d'entreprise, chacun étant désigné par un Autonomous System (AS). Ce protocole nous a servi à réaliser le routage entre les différents sites et le cœur de réseau, pour cela nous avons utilisé quatre Autonomous Systems, un pour chaque site et un pour le cœur de réseau. Un AS est un numéro unique au réseau d'une organisation ou au réseau d'une entreprise, il sert à localiser un réseau sur internet, ces informations sont contenues dans la table de routage du protocole BGP.

- 
- MPLS : MPLS (Multi Label Switching Protocol) est un mécanisme de transport de données basé sur la commutation de labels conçue pour améliorer la vitesse et l'efficacité du transfert des données au sein de réseaux étendus. Il fonctionne généralement dans un réseau privé virtuel (VPN). Il peut ainsi transporter des paquets IPv4, IPv6, des trames Ethernet et ATM. Ce protocole nous a servi à réaliser des tunnels statiques entre différents routeurs du cœur de réseau, son utilité est que toutes les informations transitent toujours à travers les mêmes routeurs et ne changent pas en fonction de l'état du réseau (dans notre cas).

## **ii. Mise en place des protocoles :**

Tout d'abord, nous avons réalisé le protocole OSPF :



Comme illustré sur le schéma ci-dessus qui était notre simulation dans le logiciel gns3, nous avons configuré ce protocole sur les 5 routeurs suivants : PE50, PE0, PE1, P1 et P2, les nuages représentant les routeurs des sites distants. Voici les commandes que nous avons tapées sur P2 pour configurer et l'activer :

---

```
P1>conf t
```

```
P1(config)#router ospf 10
```

```
P1(config-router)#network 10.0.2.8 0.0.0.3 area 0
```

```
P1(config-router)#network 10.0.2.28 0.0.0.3 area 0
```

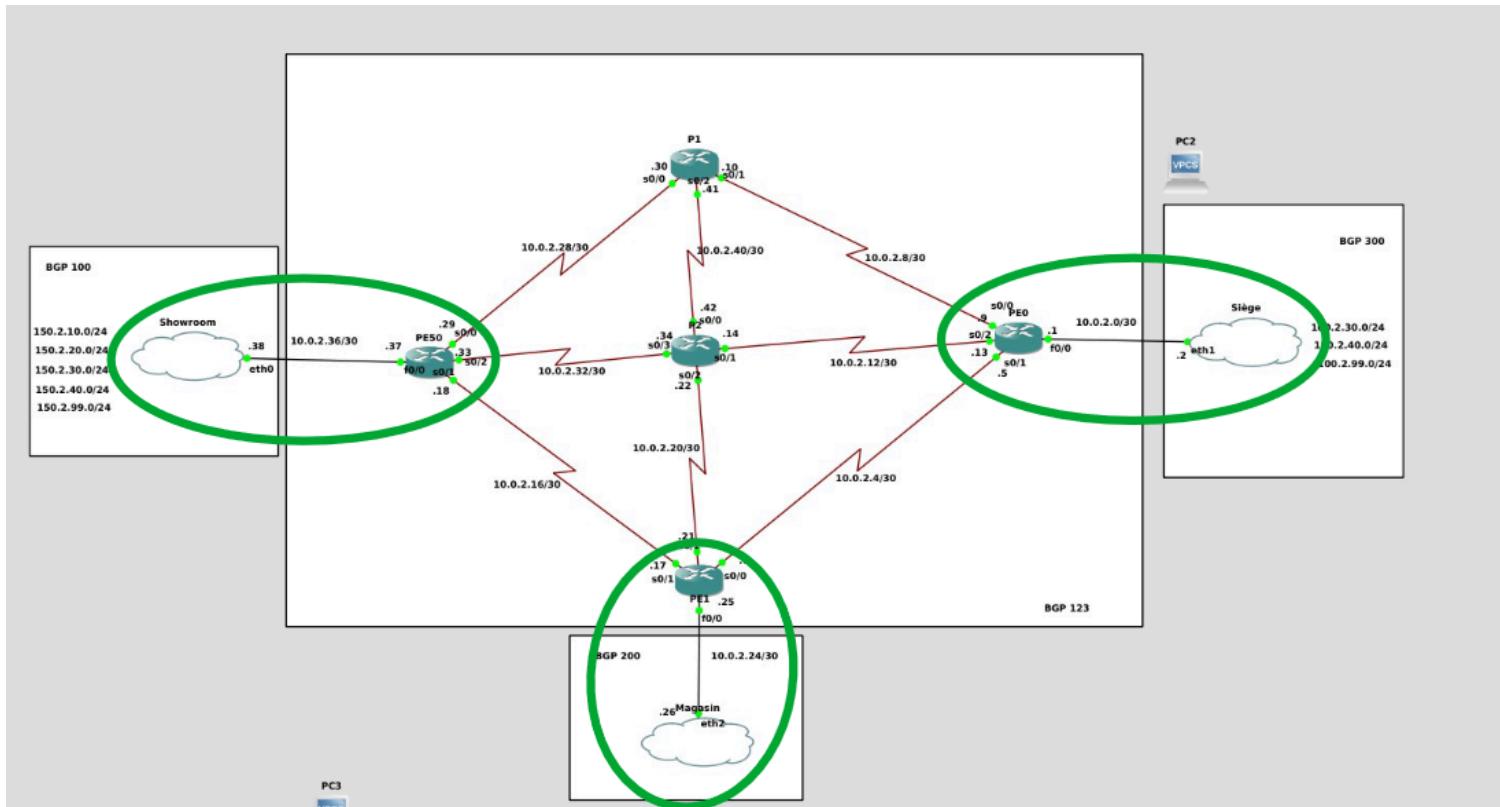
```
P1(config-router)#network 10.0.2.40 0.0.0.3 area 0
```

```
P1(config-router)#redistribute connected subnets
```

```
P1(config-router)#redistribute bgp 123 subnets
```

Une fois ces informations renseignées, le routeur, par le protocole OSPF, est capable de partager ses informations de sa table de routage à ses voisins, il reçoit également celle des autres.

- Protocole BGP



Cette fois-ci, comme l'indique le schéma, le protocole bgp sera configuré entre le cœur de réseau de notre opérateur et les trois sites, on utilise ainsi quatre AS différents. Pour activer ce protocole sur les routeurs PE0, PE50, P2, CE50, CE1 et CE0, nous avons tappé ces commandes sur le routeur PE0 :

---

```
PE0>conf t
```

```
PE0(config)#router bgp 123
```

```
PE0(config-router)#network 10.0.2.0 mask 255.255.255.0
```

```
PE0(config-router)#network 10.0.2.4 mask 255.255.255.0
```

```
PE0(config-router)#network 10.0.2.8 mask 255.255.255.0
```

```
PE0(config-router)#network 10.0.2.12 mask 255.255.255.0
```

```
PE0(config-router)#neighbor 10.0.2.1 remote-as 300
```

```
PE0(config-router)#redistribute connected
```

```
PE0(config-router)#redistribute ospf 10 internal external 1 external 2
```

La commande **network 10.2.2.4 mask 255.255.255.0** permet d'inscrire le réseau dans le protocole de routage BGP.

La commande habituelle redistribute ospf 10 ne fonctionnant pas, nous avons effectué des recherches et nous sommes parvenus à la conclusion que chaque version IOS des routeurs et switchs peut différer. La commande "**redistribute ospf 10 internal external 1 external 2**" permet alors de redistribuer les routes apprises par le protocole bgp au protocole ospf tout comme la commande dans la configuration ospf "**redistribute bgp 123**". Ainsi les tables de routage sont mises en commun et partagées entre les différents protocoles présents sur les mêmes routeurs, permettant le transit des informations et des communications entre les réseaux.

---

## 4. Mise en place du serveur d'infrastructure

### a. Configuration DNS

Un serveur DNS (Domain Name Server) permet de convertir des noms de domaine (beerok.com dans notre cas) en adresses IP et inversement.

#### i. Configuration serveur DNS

Suivant les spécifications du cahier des charges, nous avons mis en place un serveur DNS au siège de l'entreprise, sur un serveur Windows. Pour assurer un fonctionnement cohérent sur tous les clients, indépendamment de leur emplacement, nous avons ajouté une nouvelle **zone DNS principale** appelée "beerok.com". Dans cette configuration, nous avons décidé de ne pas autoriser les mises à jour dynamiques (DDNS).

Les principales étapes de la configuration incluent l'ajout d'enregistrements A pour chaque machine du réseau et d'enregistrements MX pour la gestion des e-mails. Ces enregistrements servent à associer des noms de domaine à des adresses IP et à définir les serveurs de messagerie.

Voici la table de relations DNS que nous avons utilisée sur notre serveur Windows après configuration :

debserveur	Hôte (A)	100.2.99.251
mail	Hôte (A)	100.2.99.251
mail	Serveur de messagerie (...)	[10] mail.beerok.com.
mosquito	Hôte (A)	100.2.99.250
nextcloud	Hôte (A)	100.2.99.251
voice	Hôte (A)	100.2.30.252
win-rtp0bnk2q2i	Hôte (A)	100.2.99.252
win-rtp0bnk2q2i	Hôte IPv6 (AAAA)	fec0:0000:0000:0000:fc8a:9...
winserveur	Hôte (A)	100.2.99.252

---

## ii. Configuration client DNS

Pour les clients fonctionnant avec DHCP (Showroom et Magasin), nous avons dû rajouter une ligne de configuration sur le routeur permettant de fournir l'adresse IP de celui-ci. Pour ce qui est des clients n'ayant pas de DHCP (Siège), ils nous à fallu renseigner l'adresse IP du serveur DNS, ainsi que le nom de domaine directement dans le fichier **/etc/resolv.conf** sur les hôtes Linux. Voici le contenu de chaque fichier **/etc/resolv.conf** que nous avons configuré :

**domain beerok.com**

**search beerok.com**

**nameserver 100.2.99.252**

Voici un exemple de résolution DNS après un ping sur le nom d'une machine de notre nom de domaine :

```
root@Debian11:~# ping winserv
PING winserv.beerok.com (100.2.99.252) 56(84) bytes of data.
64 bytes from pool-100-2-99-252.nycmny.fios.verizon.net (100.2.99.252): icmp_seq=1 ttl=128 time=0.777 ms
64 bytes from pool-100-2-99-252.nycmny.fios.verizon.net (100.2.99.252): icmp_seq=2 ttl=128 time=0.862 ms
64 bytes from pool-100-2-99-252.nycmny.fios.verizon.net (100.2.99.252): icmp_seq=3 ttl=128 time=0.924 ms
64 bytes from pool-100-2-99-252.nycmny.fios.verizon.net (100.2.99.252): icmp_seq=4 ttl=128 time=0.991 ms
64 bytes from pool-100-2-99-252.nycmny.fios.verizon.net (100.2.99.252): icmp_seq=5 ttl=128 time=0.953 ms
64 bytes from pool-100-2-99-252.nycmny.fios.verizon.net (100.2.99.252): icmp_seq=6 ttl=128 time=0.901 ms
64 bytes from pool-100-2-99-252.nycmny.fios.verizon.net (100.2.99.252): icmp_seq=7 ttl=128 time=0.836 ms
^C
--- winserv.beerok.com ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6010ms
rtt min/avg/max/mdev = 0.777/0.892/0.991/0.067 ms
root@Debian11:~# ping debserv
PING debserv.beerok.com (100.2.99.251) 56(84) bytes of data.
64 bytes from pool-100-2-99-251.nycmny.fios.verizon.net (100.2.99.251): icmp_seq=1 ttl=64 time=0.062 ms
64 bytes from pool-100-2-99-251.nycmny.fios.verizon.net (100.2.99.251): icmp_seq=2 ttl=64 time=0.057 ms
^C
--- debserv.beerok.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.057/0.059/0.062/0.002 ms
root@Debian11:~# ping nextcloud
PING nextcloud.beerok.com (100.2.99.251) 56(84) bytes of data.
64 bytes from pool-100-2-99-251.nycmny.fios.verizon.net (100.2.99.251): icmp_seq=1 ttl=64 time=0.039 ms
^C
```

## b. Configuration Active Directory

Nous avons ensuite installé et configuré un serveur Active Directory. Active Directory est un **annuaire LDAP** pour les systèmes d'exploitation Windows créé par Microsoft. Cet annuaire contient différents objets, de différents types (utilisateurs, ordinateurs, etc.). Afin que Active Directory fonctionne, il est important de créer un DNS (Domain Name Server). En effet, celui-ci utilise le serveur de noms afin d'appliquer ces stratégies. Nous avons donc installé le rôle Active Directory et nous l'avons configuré ainsi :

Nous avons donc créé des OU dans le but de classer les différents utilisateurs en fonction de leur sites. Une unité organisationnelle ou unité d'organisation est un conteneur dans un domaine Microsoft Active Directory qui peut contenir des utilisateurs, des groupes et des ordinateurs en fonction de paramètres comme le lieu géographique, la relation entre les utilisateurs...

Dans notre cas, nous avons donc créé 3 OU : ventes, informatique, et Direction. Nous avons également créé 3 groupes globaux : G\_Direction, G\_Ventes et G\_Informatique. Suivant la méthode AGDLP, nous créons également des groupes locaux auxquels nous ajoutons dans les groupes globaux : DL\_Siège, DL\_Magasin, DL\_Showroom. Le gestionnaire des utilisateurs et sites d'Active Directory ressemble donc ainsi :

 DL_magasin1	Groupe de sécurité - Domaine lo...
 DL_Showroom	Groupe de sécurité - Domaine lo...
 DL_siège	Groupe de sécurité - Domaine lo...
 G_Direction	Groupe de sécurité - Global
 G_informatique	Groupe de sécurité - Global
 G_Ventes	Groupe de sécurité - Global

La méthode **AGDLP (Account, Global, Domain Local, Permission)** résume les recommandations émises par Microsoft pour le bon fonctionnement des droits relatifs aux services de Active Directory. Cette méthode consiste également à manipuler l'imbrication des groupes de sécurité et des étendues associées à ces groupes de sécurité. Il existe 3 types d'étendues : globale, locale et universelle.

La méthode AGDLP consiste à appliquer le principe suivant :

Un **compte utilisateur(A)** doit être membre d'un **groupe de sécurité global (G\_)**,

Ce **groupe de sécurité global** doit ensuite être ajouté en tant que **membre d'un groupe de sécurité domaine local (DL\_)** - Ayant une portée uniquement sur le domaine d'appartenance,

Ce **groupe de sécurité domaine local** est utilisé pour ajuster les **permissions (P)** NTFS sur le répertoire partagé

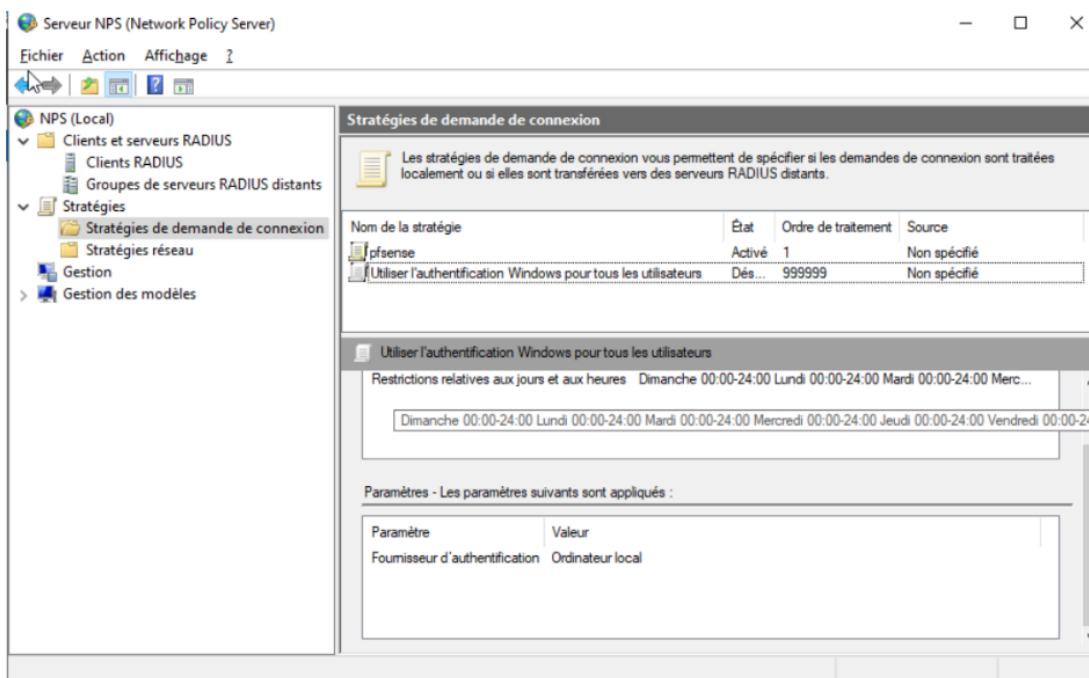
Nous avons ensuite créé les utilisateurs du réseau dans Active Directory selon le cahier des charges et les ajoutons à leur groupes globaux respectifs.

Utilisateurs et ordinateurs Active	Nom	Type
Requêtes enregistrées	Jack Dalton	Utilisateur
beerok.com	Averell Dalton	Utilisateur
Builtin		
Computers		
Direction		
Domain Controllers		
ForeignSecurityPrincipal:		
Groupes		
Informatique		
Managed Service Accour		
Users		
Ventes		

## c. Configuration RADIUS

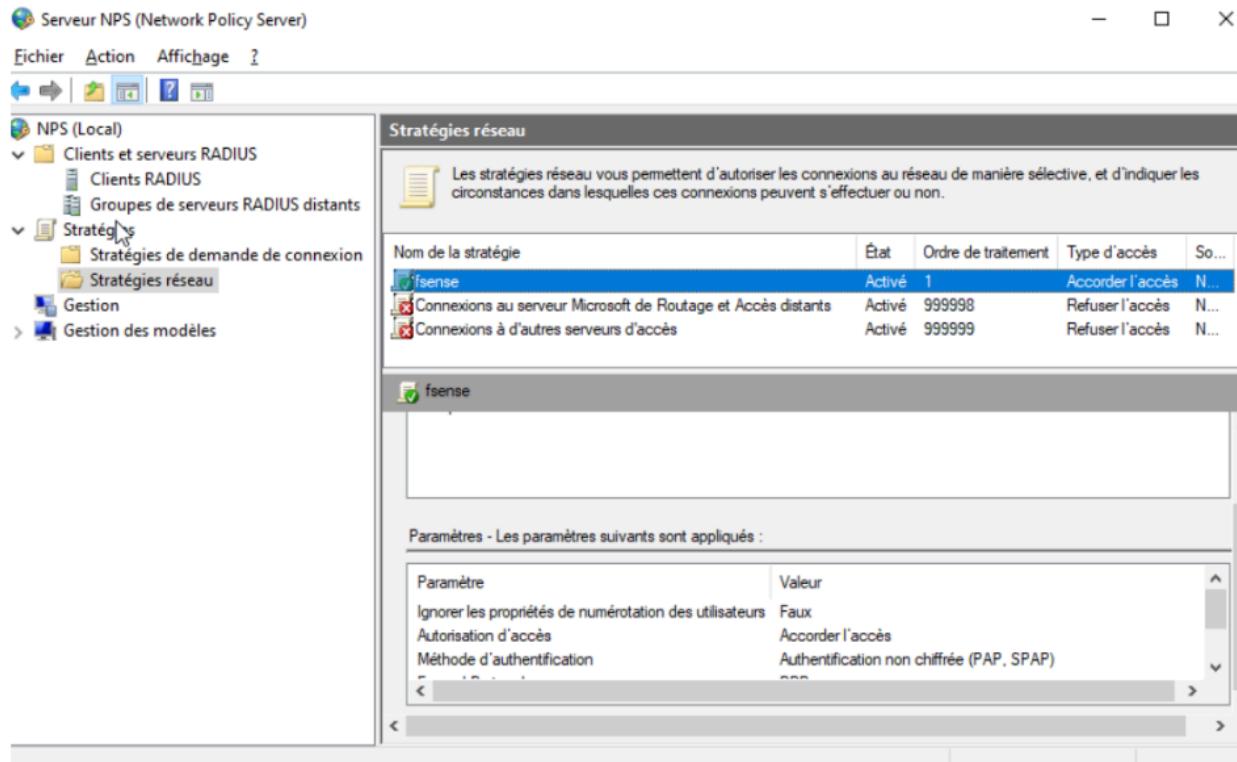
RADIUS (Remote Authentication Dial-In User Service) est un protocole d'authentification qui permet de vérifier l'identité des utilisateurs se connectant au réseau. Le serveur RADIUS, tel que le service NPS (Network Policy Server) de Microsoft, a donc un rôle important dans la sécurisation des accès au réseau.

NPS prend en charge l'authentification, l'autorisation et la comptabilité des utilisateurs. Il permet de définir des politiques d'accès en fonction des besoins spécifiques de l'organisation. Pour configurer NPS, nous avons définis des stratégies qui spécifient les conditions d'authentification et d'autorisation. Il existe différents types de stratégies : Stratégies de demande de connexion et Stratégie réseau. La stratégie que nous avons appliquée en premier est une stratégie de demande de connexion appelée "pfSense". Nous avons défini dans cette stratégie de transmettre les demandes de connexions au réseau au serveur RADIUS qui les traitera et en fonction si l'utilisateur existe dans Active Directory, acceptera ou pas cette demande.



Première stratégie

La deuxième stratégie que nous avons mise en place est une stratégie réseau. Cette stratégie autorise l'accès au réseau pour les utilisateurs authentifiés.



### Deuxième stratégie

La différence entre ces deux stratégies est que la première stratégie gère le processus d'authentification des demandes de connexion, tandis que la deuxième stratégie définit les autorisations d'accès au réseau une fois que l'authentification a été réussie.

Lors de la configuration de NPS, afin que ces stratégies fonctionnent, nous avons également défini des clients RADIUS, qui sont les périphériques réseau qui envoient les demandes d'authentification au serveur RADIUS.



### Client Radius

---

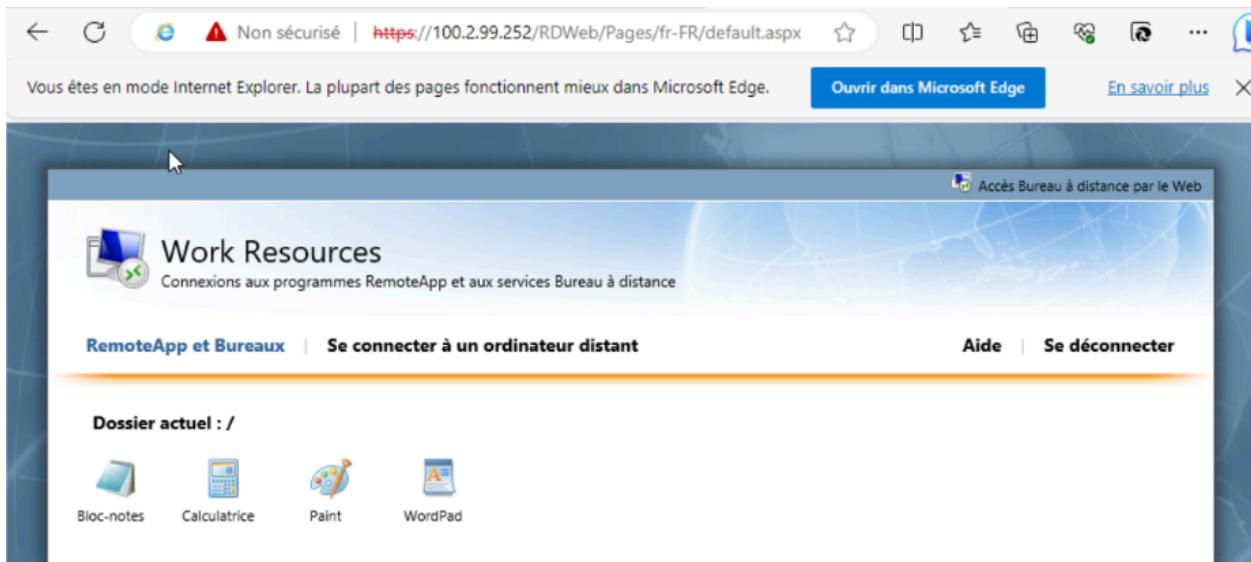
## d. Configuration des Services de Bureau à Distances

Une fois que le serveur Active Directory a été configuré, nous avons pu mettre en place des Services de Bureau à Distance. L'objectif des services de Bureau à Distance est l'optimisation de l'espace de stockage de chaque machine cliente du réseau et permet de centraliser sur des serveurs, dans un cloud sécurisé, les données de chaque utilisateurs ainsi que leur espace de travail. Cette solution peut être utile notamment lorsque les PC client se situent dans des espaces géographiques différents. Afin de mettre en place ce service, nous avons tout d'abord installé et configuré les Bureaux à Distance comme suit.

Nous avons tout d'abord configuré le Gestionnaire de Licence des Services de Bureaux à Distances. Ce gestionnaire permet d'activer la licence du serveur de bureau à distance pour permettre celui-ci de publier des applications. Pour ce faire, nous sommes allés dans le gestionnaire des licences et nous avons rempli les informations demandées lors de l'activation de la Licence du serveur. Ce système permet aux serveurs Terminal Servers d'obtenir et de gérer les licences d'accès client des services Terminal Server pour les périphériques et les utilisateurs connectés à un serveur Terminal Server.

Une fois activé, nous activons également la Passerelle des services de Bureau à distance. Cela permet aux utilisateurs de se connecter via Internet en utilisant le protocole de transport de communication HTTPS et le protocole UDP, respectivement.

Une fois ces services activés, nous avons créé une collection dans laquelle nous avons publié des applications. Une collection de session représente un ensemble d'applications rendues accessibles aux utilisateurs via une adresse spécifique. Nous avons donc publiés des applications via cette collection et nous pouvons y accéder via le web :



---

## 5. Mise en place du serveur d'applications

### a. Configuration de la Messagerie

La configuration du serveur de messagerie implique plusieurs étapes pour assurer un fonctionnement robuste et sécurisé. Un système de messagerie est composé de trois services : MTA, MSA, MDA.

- **Configuration du MSA/MTA**

Le **Mail Submission Agent (MSA)** constitue la composante chargée de recevoir les courriers électroniques créés par le Mail User Agent (MUA) lors de leur envoi. Son rôle essentiel est d'accepter ces courriers sortants, puis de les acheminer vers le MTA pour leur transfert ultérieur à travers le réseau de messagerie. Le port que le MSA utilise afin de transmettre les courriers électroniques au MTA via SMTP est le port 587. SMTP est le protocole de transport des courriels. Il est basé sur une architecture Client/Serveur et il est basé sur des files d'attente. Les échanges se déroulent en trois étapes que nous décrirons dans l'exemple qui suit notre configuration.

Le **Mail Transfer Agent (MTA)** est le composant chargé de transmettre les courriels venant du MSA, d'un MUA ou d'un autre MTA aux autres MTA ou au MDA du destinataire. Il utilise le protocole SMTP sur TCP sur le port 25.

Le **Mail User Agent (MUA)** est le client de messagerie, il peut être lourd (thunderbird...) ou léger (webmail).

Nous avons donc choisi comme MSA et MTA le serveur Postfix. Afin de le faire fonctionner, il faut tout d'abord ajouter dans le DNS un enregistrement MX pointant sur le serveur de messagerie avec une priorité de 10. Ensuite, nous avons supprimé le programme de messagerie de base de linux : exim. Puis, nous avons configuré postfix comme étant un serveur de messagerie de type site internet. Nous avons autorisé le relais du courrier en laissant vide les champs lorsqu'on exécute la commande **dpkg-reconfigure postfix** afin de le configurer.

---

Nous avons seulement utilisé la couche IPv4 seulement et autorisé les autres options demandées par défaut. Afin de vérifier que Postfix fonctionne correctement, nous avons effectué la commande **postfix -v check**.

```
root@Debian11:/tmp# postfix -v check
postfix: name mask: ipv4
postfix: inet_addr_local: configured 2 IPv4 addresses
root@Debian11:/tmp#
```

Afin que les mails soient acheminés jusqu'à leurs destinataires, nous avons également configurés les interfaces auxquelles Postfix devait écouter : Pour cela, il faut modifier le champs **inet\_interface = ens19** dans le fichier de configuration de postfix, soit **/etc/postfix/main.cf**.

Afin de tester le bon fonctionnement du serveur de messagerie entre nos 3 utilisateurs Joe, Jim et Jack, nous avons utilisé les primitives de **SMTP**, à savoir HELO, RCPT, MAIL, DATA. Il faut tout d'abord ouvrir une session telnet sur le serveur de messagerie afin d'utiliser ces primitives

**HELO** est une primitive de SMTP qui indique au serveur de mail le début d'une session d'envoi de mail. Il faut ensuite spécifier le nom du serveur de mail.

**MAIL** est la primitive qui spécifie l'envoyeur du mail. Cette primitive s'accompagne des paramètres FROM:sendermail.

**RCPT** est la primitive qui spécifie le destinataire du mail. Cette primitive s'accompagne des paramètres TO:destinationmail.

**DATA** est la primitive qui signifie au serveur que le message à transmettre commence.

Afin de terminer un mail avec les primitives, il faut sauter une ligne et écrire un ":".

Pour quitter la session, il suffit d'écrire **QUIT**.

---

Voici le test réalisé sur notre serveur mail pour vérifier son fonctionnement :

```

root@Debian11:/tmp# telnet debserv 25
Trying 100.2.99.251...
Connected to debserv.beerok.com.
Escape character is '^>'.
220 Debian11 ESMTP Postfix (Debian/GNU)
HELO
501 Syntax: HELO hostname
HELO debserv
250 Debian11
MAIL FROM:jim@beerok.com
250 2.1.0 Ok
RCPT TO:joe@beerok.com
250 2.1.5 Ok
DATA
354 End data with <CR><LF>,<CR><LF>
Test de dimanche pour le rapport
.
250 2.0.0 Ok: queued as 0A79F100ED4
QUIT
221 2.0.0 Bye
Connection closed by foreign host.
root@Debian11:/tmp# 
```

Nous pouvons vérifier que le message à bien été envoyé en regardant les log dans **/var/log/syslog** avec la commande **tail -40 /var/log/syslog | grep 0A79 (id du message)**

```

root@Debian11:/tmp# tail -40 /var/log/syslog | grep 0A79
Feb  8 19:49:47 Debian11 postfix/smtpd[48423]: 0A79F100ED4: client=pool-100-2-99-251.nycmny.fios.verizon.net[100.2.99.251]
Feb  8 19:50:04 Debian11 postfix/cleanup[48432]: 0A79F100ED4: message-id=<20240208184947.0A79F100ED4@Debian11>
Feb  8 19:50:04 Debian11 postfix/qmgr[47956]: 0A79F100ED4: from=<jim@beerok.com>, size=348, nrcpt=1 (queue active)
Feb  8 19:50:04 Debian11 postfix/local[48435]: 0A79F100ED4: to=<joe@beerok.com>, relay=local, delay=26, delays=26/0.02/0/0.02, dsn=2.0.0, status=sent (delivered to mailbox)
Feb  8 19:50:04 Debian11 postfix/qmgr[47956]: 0A79F100ED4: removed
root@Debian11:/tmp# 
```

Nous pouvons également vérifier l'envoi du mail avec la commande effectué dans le compte du destinataire suivante : **cat /var/mail/joe**

```

From jim@beerok.com Thu Feb  8 19:50:04 2024
Return-Path: <jim@beerok.com>
X-Original-To: joe@beerok.com
Delivered-To: joe@beerok.com
Received: from debserv (pool-100-2-99-251.nycmny.fios.verizon.net [100.2.99.251])
        by Debian11 (Postfix) with SMTP id 0A79F100ED4
        for <joe@beerok.com>; Thu,  8 Feb 2024 19:49:38 +0100 (CET)
Message-Id: <20240208184947.0A79F100ED4@Debian11>
Date: Thu,  8 Feb 2024 19:49:38 +0100 (CET)
From: jim@beerok.com

Test de dimanche pour le rapport 
```

---

Une fois le test réalisé et réussi, nous passons à l'installation du frontal de messagerie. Nous devions faire un client lourd et un client léger, mais pour des soucis de temps, nous n'avons pu réaliser le client léger. Nous avons donc choisi comme client lourd Evolution.

#### - Configuration du frontal de messagerie (MDA)

Le **Mail Delivery Agent (MDA)** a deux fonctions : recevoir les mails depuis le MTA via SMTP et transmettre les mails reçus au MUA via **POP3** ou **IMAP**. POP3 est un protocole qui récupère les courriers reçus par le MDA. Il fonctionne sur le port 110 en non chiffré et 993 en chiffré avec SSL. IMAP est un protocole qui permet directement d'accéder aux mails sur le MDA. Il fonctionne sur le port 143 en TCP en non chiffré et sur le port 993 en chiffré.

Afin d'installer un client lourd, nous avons tout d'abord configuré et installé dovecot-pop3 et dovecot-imap. Nous les installons ainsi :

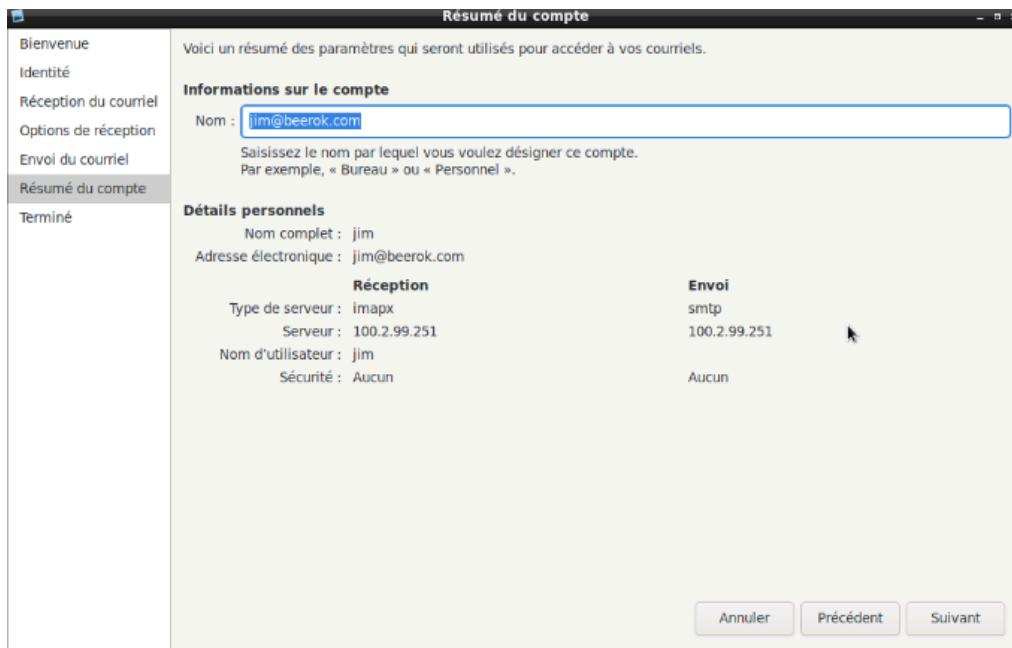
**apt-get install dovecot-pop3d dovecot-imapd**

Nous installons ensuite evolution sous linux et le configurons :

**apt-get install evolution**

Une fois installé, nous le configurons pour le compte jim dalton. Pour cela, nous sommes allés dans le menu édition, préférences, puis ajouter un compte. Une fois sur la fenêtre, nous remplissons le compte de jim avec comme nom d'utilisateur : jim, comme adresse de messagerie [jim@beerok.com](mailto:jim@beerok.com), comme protocole de réception du courriel : IMAP avec l'adresse IP du serveur mail d'après le plan d'adressage IP. Nous définissons que l'envoi de mail n'est pas chiffré. Ainsi, le port utilisé est 143. Nous mettons les mêmes paramètres pour configurer le serveur d'envoi de mail, à la différence que le protocole utilisé est SMTP et que le port utilisé est 25. Nous avons également désactivé l'authentification dans postfix et dans dovecot avec l'ajout de la variable **smtpd\_use\_tls=no** dans **/etc/postfix/main.cf** et **disable\_plaintext\_auth = no** dans **/etc/dovecot/conf.d/10-auth.conf**

Voici le résumé de la configuration obtenue sur le client lourd :



---

## b. Configuration du serveur NextCloud

NextCloud est un cloud local que l'on peut déployer sur un serveur ou sur une infrastructure. Il permet le stockage partagé de données en local, et permet ainsi une meilleure gestion et sécurité des données. Nous l'avons installé sous Linux. Durant l'installation et la configuration de NextCloud, nous avons rencontré de nombreux problèmes pour l'installer. Nous vous présentons ici la méthode que nous avons utilisée, et nous expliquerons les problèmes rencontrés en conclusion. Afin de l'installer, il faut tout d'abord installer des prérequis :

- Une base de donnée SQL (mariadb ou MySQL)
- Un serveur web (Apache ou nginx, dans notre cas apache2)
- un serveur PHP.

- Installation et configuration de la base de donnée MariaDB)

Nous avons installé tous les paquets nécessaire au bon fonctionnement de NextCloud :

```
apt-get install apache2 mariadb-server php php-common php-curl php-gd php-intl  
php-mbstring php-xmlrpc php-mysql php-xml php7.4-cli php-zip
```

Afin de créer une base de données pour NextCloud, nous avons tout d'abord configuré le serveur MariaDB avec la commande **mysql\_secure\_installation**. Nous y avons défini le mot de passe du compte root, si les utilisateurs anonymes pouvaient s'y connecter ainsi que d'autres options. Nous nous sommes ensuite connectés à cette base de données avec **mysql -u root -p**

Une fois connectés, nous créons la base de données nécessaire pour NextCloud :

```
CREATE DATABASE db32nextcloud;
```

Et nous accordons tous les droits à l'utilisateur de nextcloud admin.

```
GRANT ALL ON db23nextcloud.* TO 'admin'@'localhost' IDENTIFIED BY 'fuel';
```

---

Puis, nous mettons à jour les autorisations et notre base de données est fonctionnelle.

### **FLUSH PRIVILEGES;**

```
MariaDB [(none)]> show databases
    -> ;
+-----+
| Database      |
+-----+
| db23nextcloud |
| information_schema |
| mysql          |
| performance_schema |
+-----+
4 rows in set (0,003 sec)

MariaDB [(none)]> GRANT ALL ON db23nextcloud.* TO 'admin'@'localhost' IDENTIFIED BY 'fuel';
Query OK, 0 rows affected (0,016 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> █
```

La base de données est maintenant configurée. Nous installons donc NextCloud maintenant que le serveur php et apache2 ont été installé. Nous installons la version 20.0.5 de NextCloud par souci de compatibilité avec la version php7.4 sur le site de NextCloud. Une fois téléchargés, nous nous rendons dans le répertoire /tmp et nous décompressons l'archive avec unzip

### **unzip nextcloud-20.0.5.zip**

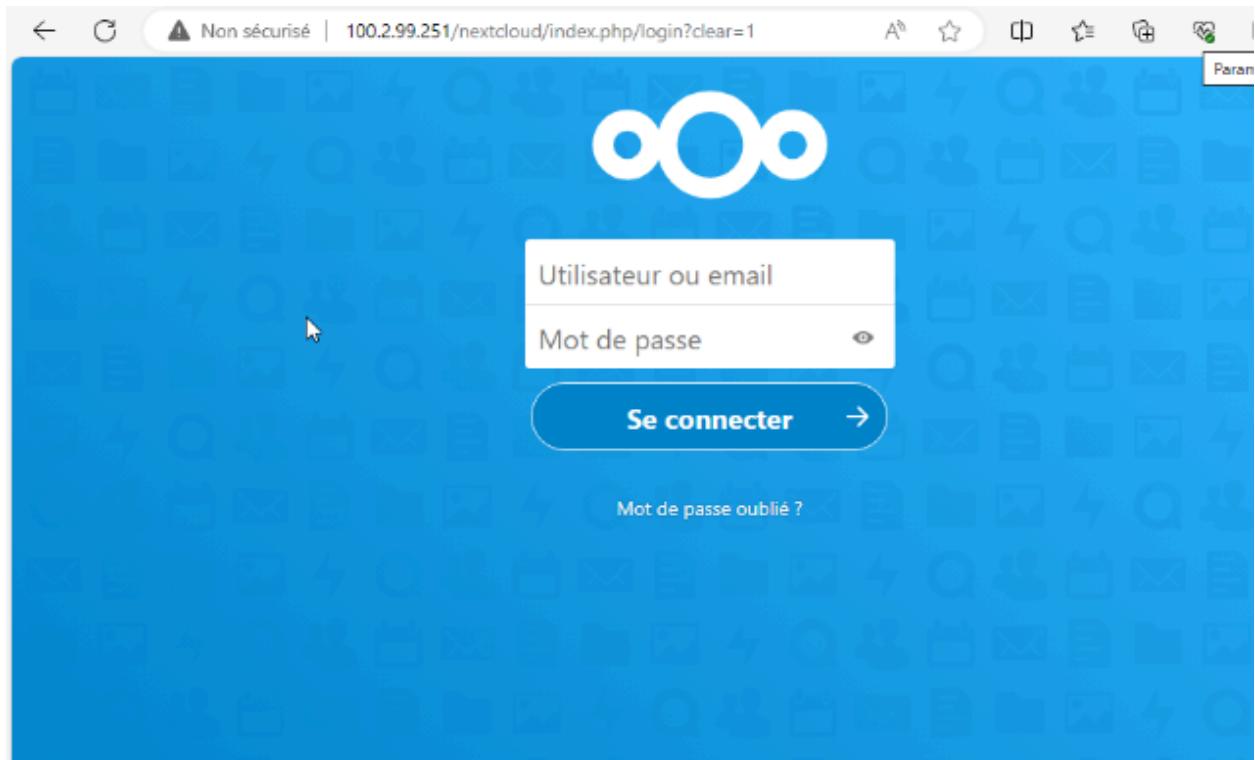
Puis, nous déplaçons le fichier décompressé dans le répertoire du serveur web :

### **mv nextcloud-20.0.5 /var/www/html**

Ainsi, NextCloud sera accessible sur le web depuis l'adresse <http://100.2.99.251/nextcloud>. Il ne reste plus qu'à changer le propriétaire des données de Nextcloud pour que ce soit l'utilisateur d'apache2

### **chown -R www-data:www-data /var/www/html/nextcloud**

Nous pouvons vérifier que le Cloud est accessible depuis le réseau en y accédant depuis le serveur Windows :



---

## c. Configuration du serveur Téléphonique

Afin de mettre en place des services de téléphonie au sein de l'entreprise nous nous sommes tout d'abord interrogés sur quel service de téléphonie nous allions configurer. Nous avons donc choisi d'utiliser le service freePBX. Ce serveur est configuré pour gérer les communications vocales de l'entreprise, il est donc inscrit sur le VLAN 30 de la voix.

Le serveur téléphonique basé sur FreePBX est configuré pour gérer les communications vocales de l'entreprise. FreePBX permet la configuration des extensions, des lignes téléphoniques. La sécurité est assurée par le chiffrement des appels, la gestion des droits d'accès et la surveillance des activités téléphoniques pour garantir la confidentialité et la sécurité des communications. Chaque réseau où le téléphone connecté au réseau via DHCP reçoit un identifiant SIP permettant l'établissement d'un appel avec un autre utilisateur.

Nous avons donc téléchargé l'OS de FreePBX sur une machine virtuelle pour obtenir l'adresse IP du serveur et ainsi accéder à l'interface graphique. Cette étape nous a permis de configurer plus facilement notre serveur téléphonique. Une fois l'accès à l'interface graphique obtenu, nous avons créé les numéros SIP attribués au téléphones SIP.

Un téléphone SIP est un poste téléphonique utilisant la Voix sur IP (VoIP) pour acheminer les communications. Il est à ce titre relié à une connexion IP comme les ordinateurs. Ainsi, Chaque personne au sein de l'entreprise a reçu un numéro, conformément aux exigences du cahier des charges.

---

Pour tester la mise en place de notre serveur, nous avons installé des téléphones IP sur le switch du magasin et du siège. Les adresses IP ont été attribuées en DHCP pour le magasin et en statique pour le siège. Une fois cette configuration réalisée, nous avons paramétré les Linphone, qui sont des softphone) et les téléphones physiques. Sur Linphone, nous devions fournir l'adresse IP du serveur PBX et le numéro SIP de la personne que nous souhaitions appeler. De plus, nous devions créer un compte en utilisant un numéro SIP préalablement créé.

Quant aux téléphones IP, la démarche était similaire. Nous utilisons l'adresse IP du serveur PBX et un numéro SIP pour joindre directement un poste en composant simplement le numéro SIP correspondant.

## d. Configuration du Service Wifi avec portail captif PfSense

La mise en place d'un portail captif avec pfSense sur une borne WiFi Linksys permet d'assurer un accès sécurisé et contrôlé au réseau sans fil de l'entreprise.

Tout d'abord, l'installation et la configuration de pfSense sur un serveur dédié implique le déploiement du logiciel pfSense sur une machine virtuelle et la configuration des interfaces réseau pour gérer le trafic LAN (réseau PfSense) et WAN (réseau entreprise ici le Showroom).

The screenshot shows the pfSense Community Edition web interface. At the top, there's a warning message: "WARNING: The 'admin' account password is set to the default value. Change the password in the User Manager." Below this, the main dashboard has two main sections: "System Information" on the left and "Interfaces" on the right. The "System Information" section contains details like Name (pfSense.home.arpa), User (admin@192.168.1.102), System (VirtualBox Virtual Machine, Netgate Device ID: a7b5e05a5755858b2622), BIOS (Vendor: innotek GmbH, Version: VirtualBox, Release Date: Fri Dec 1 2006), Version (2.7.2-RELEASE (amd64)), CPU Type (Intel(R) Xeon(R) CPU E5-1620 v4 @ 3.50GHz, 4 CPUs: 1 package(s) x 4 cache groups x 1 core(s), AES-NI CPU Crypto: Yes (inactive), QAT Crypto: No), Hardware crypto (Inactive), Kernel PTI (Enabled), MDS Mitigation (Inactive), and Uptime (00 Hour 50 Minutes 59 Seconds). The "Interfaces" section shows two interfaces: WAN (1000baseT <full-duplex>, IP 150.2.40.2) and LAN (1000baseT <full-duplex>, IP 192.168.1.1).

En parallèle, la borne WiFi Linksys est configurée pour fonctionner en tant que point d'accès (AP) dans le réseau sans fil. Cette configuration inclut la définition du SSID, la sécurisation de la connexion avec un mot de passe robuste et éventuellement la configuration des paramètres avancés tels que les canaux WiFi et les bandes de fréquence. Nous avons donc décidé d'utiliser une méthode d'authentification sécurisée **WPA2-PSK**.

Firmware: DD-WRT v3.0-44715 micro (11/03/20)  
Time: 09:20:35 up 8:20, load average: 0.00, 0.01, 0.00  
WAN IP: 0.0.0.0

Setup Wireless Services Security Access Restrictions NAT / QoS Administration Status

Basic Settings Radius Wireless Security MAC Filter Advanced Settings WDS

**Wireless Interface wl0 [2.4 GHz]**

Physical Interface wl0 - SSID [Groupe2\_Magasin] HWAddr [30:23:03:8B:CC:DF]

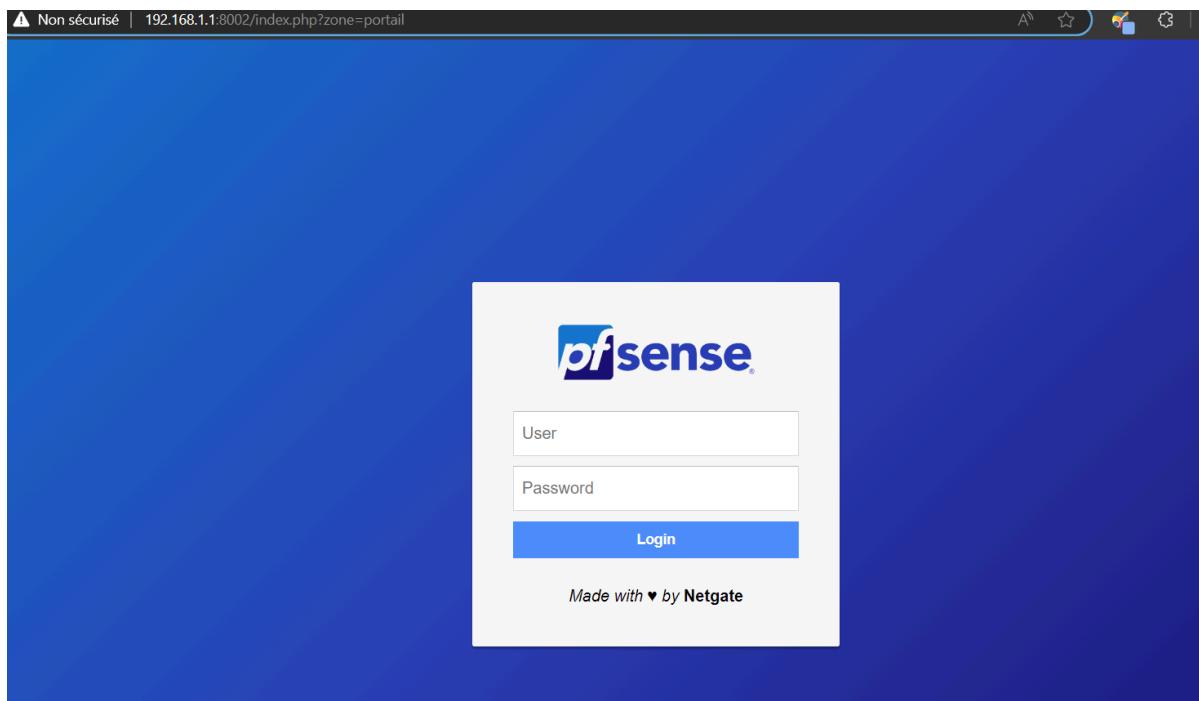
Wireless Mode: AP  
Wireless Network Mode: Mixed  
Wireless Network Name (SSID): Groupe2\_Magasin  
Wireless Channel: 6 - 2.437 GHz  
Wireless SSID Broadcast:  Enable  Disable  
Sensitivity Range (ACK Timing): 500 (Default: 500 meters)  
Network Configuration:  Unbridged  Bridged

Help more...  
Attention: It is recommended that you press **Apply Settings** after you change a value in order to update the fields with the corresponding parameters.

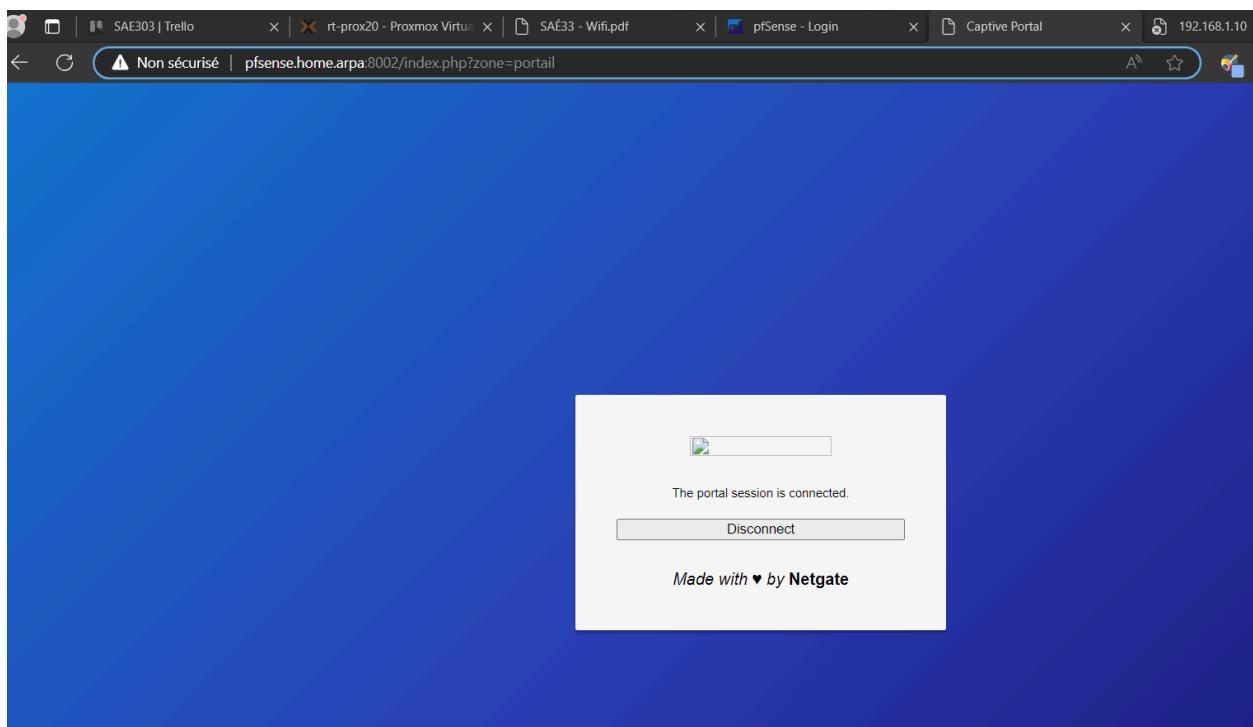
Copy Paste Virtual Interfaces Add Virtual AP

Save Apply Settings Cancel Changes

La fonctionnalité de portail captif de pfSense est activée et configurée pour gérer l'authentification des utilisateurs qui tentent d'accéder au réseau WiFi. Les options d'authentification, telles que l'authentification par nom d'utilisateur et mot de passe ont été configurées sur le réseau du portail captif :



Le serveur DHCP du point d'accès WiFi fut désactivé et fut remplacé par celui de pfSense. Des règles de redirection sont mises en place dans pfSense pour diriger le trafic des utilisateurs vers le portail captif, où ils doivent s'authentifier avant d'accéder à Internet. Ces règles permettent de canaliser le trafic vers le portail captif, où les utilisateurs sont invités à s'authentifier avant de pouvoir accéder à Internet. Des tests approfondis sont effectués pour vérifier le bon fonctionnement du portail captif, notamment l'authentification des utilisateurs, la redirection du trafic vers le portail captif et la navigation sur le serveur Nextcloud après l'authentification réussie :

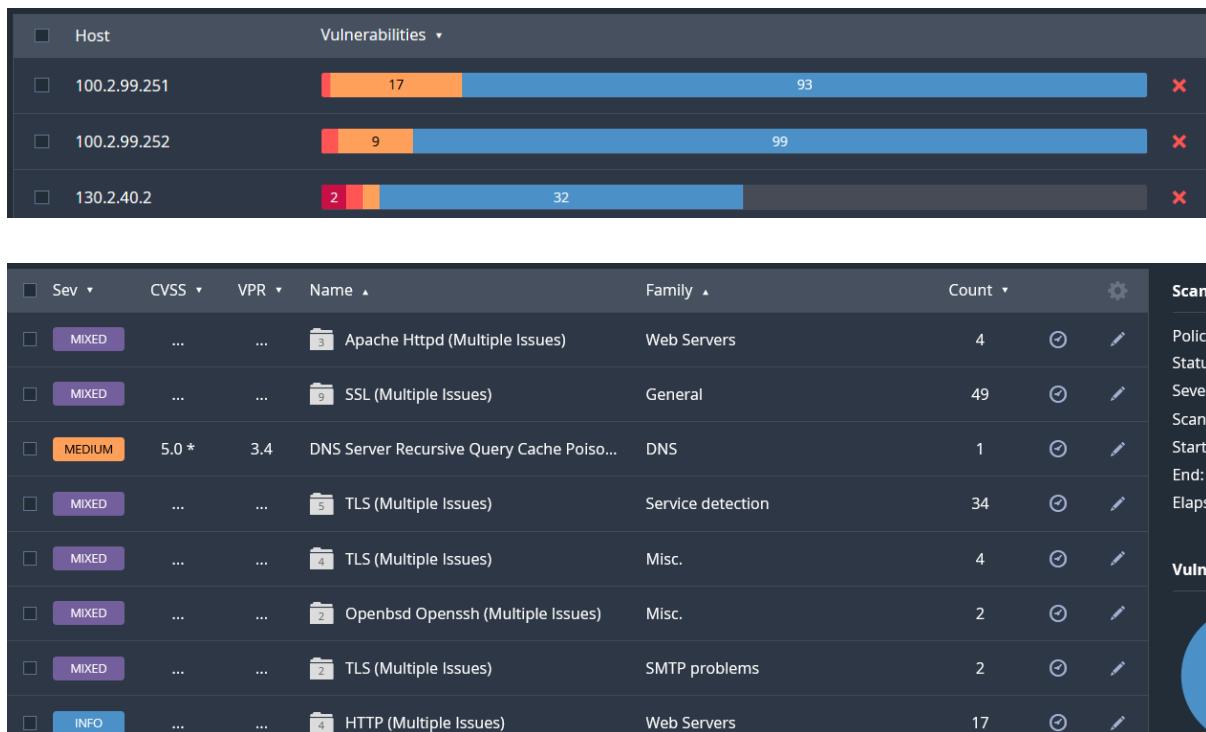


Enfin, des mesures de surveillance et de maintenance sont mises en place pour garantir la disponibilité et la sécurité du réseau WiFi. Cela comprend la surveillance des journaux système, la mise à jour régulière du système d'exploitation et des applications, ainsi que la gestion proactive des incidents de sécurité.

## 6. Sécurisation des services réseaux

Pour identifier les vulnérabilités de notre réseau, nous avons déjà fait une analyse nmap en utilisant un script spécialement créé pour cela avec l'option -script vuln. Nous avons obtenu des informations sur les vulnérabilités bien connues des appareils de notre réseau. Nous avons également obtenu des informations concernant le DNS, telles que le nom du serveur, le nom de l'étendue...

Puis nous avons fait la même chose avec l'outil de scan réseau Nessus, plusieurs sévérités de vulnérabilités ont été trouvées sur notre réseau sur différents appareils :



Nous pouvons voir qu'il y a une vulnérabilité sur le serveur DNS. Ainsi, il est important de sécuriser le réseau de sorte que les vulnérabilités soient comblées pour éviter les pertes d'informations au vu du scan nmap où nous pouvons récupérer des informations concernant le serveur DNS.

Pour garantir la sécurité des informations sensibles et éviter les risques de pertes, il est crucial de mettre en place une politique de sécurité complète, conforme aux

---

recommandations de l'ANSSI, notamment en ce qui concerne la complexité des mots de passe. Cette politique devrait inclure des exigences telles que l'utilisation de mots de passe d'au moins 8 caractères, comportant des caractères spéciaux, des lettres minuscules et des majuscules.

En outre, il est essentiel de contrôler l'accès aux salles des serveurs pour prévenir toute intrusion de logiciels malveillants via des ports USB, par exemple. Des mesures telles que l'accès aux salles de serveurs par badge et le blocage des ports inutilisés sur les switchs et routeurs peuvent être mises en place à cet effet. Restreindre l'accès SSH aux routeurs et switchs, ainsi que la mise en place d'une gestion des droits rigoureuse, sont également recommandés.

Les switchs pourraient par exemple limiter le nombre d'adresses MAC par port ou encore désactiver le port si l'adresse n'est pas enregistrée comme fiable.

Afin d'éviter la famine dhcp, il est également possible d'activer l'espionnage dhcp sur les ports.

L'espionnage DHCP ou DHCP Snooping, est une fonction de sécurité qui peut être utilisée pour empêcher les appareils malveillants d'usurper les messages DHCP et de perturber la connectivité réseau. Il fonctionne en examinant les messages DHCP et en n'autorisant que ceux qui proviennent de sources fiables. La surveillance DHCP peut être utilisée sur les commutateurs et les routeurs pour se protéger contre l'usurpation de serveur DHCP, l'usurpation de client et les attaques par déni de service. De plus, lorsque la surveillance DHCP est activée sur un commutateur ou un routeur, l'appareil garde une trace des ports autorisés à envoyer et recevoir des messages DHCP.

La famine DHCP est le fait que de nombreux appareils réseaux demandent envoient des requêtes DHCPDiscover. Ces requêtes sont souvent effectuées en utilisant des adresses MAC différentes pour chaque requête, qui sont usurpées pour donner l'impression qu'elles proviennent de clients différents. En conséquence, le serveur DHCP attribue une adresse IP à chacune de ces fausses requêtes, épuisant rapidement le pool d'adresses IP disponibles. L'attaquant peut ainsi créer son propre serveur DHCP sur le réseau attaqué et distribuer des adresses IP valides permettant ainsi de créer des attaques Man In the Middle.

Pour renforcer la sécurité, il est conseillé de bloquer les ports USB sur les ordinateurs du réseau pour éviter le déploiement de programmes malveillants via des pièces jointes. Une alternative pourrait être l'utilisation d'un cloud interne pour le stockage et le partage de fichiers, permettant un contrôle accru sur les types de fichiers partagés, par exemple NextCloud. en filtrant les extensions de fichiers.

Des mesures simples mais cruciales, telles que l'interdiction d'écrire les mots de passe sur des post-its collés aux ordinateurs, doivent être prises en compte. Pour remédier à ces failles, il est recommandé d'informer et de sensibiliser les utilisateurs tout en faisant respecter la politique de sécurité globale. De plus, la configuration d'un serveur de certificats Active Directory pourrait renforcer la sécurité en fournissant une authentification sécurisée aux services réseau.

Pour garantir une disponibilité élevée du réseau (99% du temps), l'introduction de redondances au sein du réseau, notamment à travers des protocoles de routage comme VRRP, peut contribuer à améliorer la tolérance aux pannes et la disponibilité globale du réseau :

```
R2#sh vrrp
FastEthernet0/0 - Group 1
  State is Backup
  Virtual IP address is 192.168.1.254
  Virtual MAC address is 0000.5e00.0101
  Advertisement interval is 1.000 sec
  Preemption enabled
  Priority is 100
  Master Router is 192.168.1.1, priority is 200
  Master Advertisement interval is 1.000 sec
  Master Down interval is 3.609 sec (expires in 2.969 sec)

R2#
*Mar 1 01:28:47.571: %VRRP-6-STATECHANGE: Fa0/0 Grp 1 state Backup -> Master
R2#sh vrrp
FastEthernet0/0 - Group 1
  State is Master
  Virtual IP address is 192.168.1.254
  Virtual MAC address is 0000.5e00.0101
  Advertisement interval is 1.000 sec
  Preemption enabled
  Priority is 100
  Master Router is 192.168.1.2 (local), priority is 100
  Master Advertisement interval is 1.000 sec
  Master Down interval is 3.609 sec

R2#
```

---

Il y a également la possibilité d'utiliser des certificats électroniques sur notre réseau :

En effet, les certificats informatiques sont des documents électroniques utilisés pour sécuriser les communications en ligne. Émis par des autorités de certification, ils servent à garantir l'authenticité, la confidentialité et l'intégrité des données échangées sur un réseau. En agissant comme des pièces d'identité électronique, ils permettent de vérifier l'identité des parties dans une communication et de chiffrer les données pour éviter toute interception ou altération par des personnes non autorisées. Ainsi, les certificats informatiques jouent un rôle crucial dans la protection des informations sensibles sur Internet.

---

## 7. Mise en place d'Objets Connectés

Notre objectif était de concevoir et mettre en place un système permettant de détecter la pression sur un bouton situé sur un ESP (système embarqué programmable) afin de refléter son état sur une page Web en temps réel. Une exigence supplémentaire était de différencier la pression d'un ou plusieurs boutons pour afficher les informations appropriées sur cette page Web. Cette tâche impliquait la mise en œuvre de plusieurs composants et technologies, notamment la programmation d'un serveur WebSocket en JavaScript avec Node.js, la création d'une page Web interactive en HTML et la configuration d'un microcontrôleur ESP8266 pour la détection de bouton et la communication MQTT avec le serveur. Tout au long de cette partie, nous détaillerons les différentes étapes de mise en place de ce système, en mettant en lumière les fonctionnalités spécifiques de chaque composant et leur interaction pour garantir un affichage précis et en temps réel de l'état des boutons sur la page Web.

Dans un premier temps, le code commence par la configuration de la connexion Wi-Fi et des paramètres MQTT pour se connecter au broker. Cela permet à l'ESP8266 de communiquer avec le serveur MQTT pour publier les informations sur l'état du bouton.

```
const char* ssid = "Groupe2_Showroom";
const char* password = "carapuce";
const char* mqtt_server = "150.2.99.111";
const char* mqtt_username = "ESP_Magasin";
const char* mqtt_password = "carapuce";
```

Nous avons configuré le broker MQTT pour avoir une publication mais aussi recevoir les messages des clients

```

aedes.on('client', function (client) {
  console.log('New client connection:', client.id);
});

aedes.on('publish', function(packet, client) {
});

const broker = net.createServer(aedes.handle);
broker.listen(1883, '150.2.99.111', function () {
  console.log('MQTT broker started on port 1883');
});

```

Une fois que nous recevions bien les message envoyé par les ESP nous devions faire en sorte qui publie que lorsque le bouton flash était appuyé. Nous avons donc fait un code pour sur Arduino.

```

bool etatBoutonMagasin = digitalRead(boutonFlashMagasin);
if (etatBoutonMagasin == LOW && etatBoutonMagasinPrecedent == HIGH && (millis() - tempsDernierAppuiMagasin) > delaiDebounce) {
  tempsDernierAppuiMagasin = millis();

  DynamicJsonDocument jsonDoc(128);
  jsonDoc["payload"] = "ESP_Magasin activé";
  client.publish("esp/etat", jsonDoc.c_str());
}

```

Une fois que nous avions bien les bonnes informations, nous devions les afficher sur une page Web pour cela nous avons utilisé des Websocket. Mais avant de voir la configuration de ce dernier nous allons voir la configuration du serveur Web.

```

app.use(express.static(path.join(__dirname, 'public')));
server.listen(8080, '150.2.99.111', () => {
  console.log('HTTP server started on port 8080');
});

```

---

Nous allons donc voir la dernière partie du serveur qui sont les Websockets. Les Websocket permettent d'envoyer les information reçus à la page Web sans que l'on est à recharger la page. Le serveur crée une instance WebSocket pour écouter les connexions entrantes des clients. Il enregistre ensuite un gestionnaire d'événement pour les messages et la fermeture de connexion. Les messages reçus sont traités et diffusés à tous les clients connectés.

```
const wss = new WebSocket.Server({ server });

wss.on('connection', ws => {
  console.log('New WebSocket connection');
  ws.on('message', message => {
    console.log('WebSocket message received:', message);
  });
  ws.on('close', () => {
    console.log('WebSocket connection closed');
  });
});
```

Ce bout de code permet une communication bidirectionnelle en temps réel entre les ESP et les clients Web. Une fois cela configuré nous devions donc faire la page web qui recevrait les informations publiées. La page établit une connexion WebSocket pour recevoir les messages des ESP en temps réel.

```
const socket = new WebSocket("ws://150.2.99.111:8080");
socket.onopen = function(event) {
  console.log("Connexion WebSocket ouverte avec succès.");
};
socket.onmessage = function(event) {
```

Une fois la connexion effectuée entre le serveur et la page Web nous devions analyser et afficher dans les sections appropriées en fonction de leur provenance (magasin, showroom ou simultané).

```
socket.onmessage = function(event) {  
let message = JSON.parse(event.data);  
};
```

Puis nous devions faire en sorte que la page met à jour dynamiquement le contenu pour refléter les nouveaux messages et mettre à jour les compteurs en conséquence.

```
function displayMessage(message, espid) {  
}  
  
function updateCounter(message, counterId) {  
}
```

La page web HTML fonctionne pour afficher en temps réel les messages MQTT provenant des ESP, en utilisant des extraits de code pour illustrer chaque fonctionnalité. La page web utilise des WebSockets pour une communication bidirectionnelle avec le serveur, permettant ainsi une mise à jour dynamique du contenu affiché. Voici le rendu final du site.

### Messages MQTT des ESP actifs en temps réel

**ESP\_Magasin**

Nombre de messages reçus : 2

**ESP\_Showroom**

Nombre de messages reçus : 3

**ESP\_Simultané**

ESP\_Simultané activé

Nombre d'activations simultanées : 1

### Messages MQTT des ESP actifs en temps réel

**ESP\_Magasin**

ESP\_Magasin activé

Nombre de messages reçus : 1

**ESP\_Showroom**

ESP\_Showroom activé

Nombre de messages reçus : 1

**ESP\_Simultané**

Nombre d'activations simultanées : 0

---

## 8. Problèmes rencontrés.

Nous avons rencontré de nombreuses difficultés pendant ce projet de groupe que ce soit au niveau de l'infrastructure réseau, des serveurs d'applications ou encore pour la réalisation de la partie IoT.

### - **Problèmes rencontré lors de l'installation de NextCloud**

Un de nos problèmes majeurs a été d'installer Next Cloud. D'après la notice d'utilisation sur Moodle, nous avions le choix entre trois méthodes d'installation et nous avons tout d'abord choisi d'installer NextCloud avec l'utilitaire snap. Cependant lorsque nous la commande snap install core, l'installation stoppait systématiquement au milieu. Nous avions donc décidé de passer cette commande et d'installer directement NextCloud avec l'utilitaire snap. Ainsi, la commande devenait snap install nextcloud, mais l'installation se stoppait toujours au milieu. Nous ne voulions pas créer d'autres VM par souci d'optimisation, nous avons donc cherché sur internet des solutions pour installer NextCloud et nous avons décidé d'utiliser premièrement l'utilitaire flatpack et flathub. Afin d'installer NextCloud via fatpack et flathub, nous avons tout d'abord installé flatpack, et avec la commande

flatpak install flathub com.nextcloud.desktopclient.nextcloud, nous avons réussi à installer NextCloud. Cependant, lorsque nous sommes allés sur <http://localhost/nextcloud>, une erreur nous indiquait que la version de PHP de notre machine virtuelle était obsolète par rapport à la version de Next cloud qui avait été installée. Nous avons donc décidé de mettre à jour la version de PHP de notre VM, Cependant lors du téléchargement des paquets nous nous sommes rendus compte que la version de PHP restait la même, nous en avons donc conclu que notre système ne supportait pas la version 8.1 requise par next Cloud. Nous avons donc téléchargé une version ancienne de NextCloud compatible avec notre version de PHP. Lorsque nous avons lancé notre navigateur sur l'adresse IP de nextcloud, il a fonctionné. Nous avons rencontré un autre problème avec NextCloud pour authentifier les utilisateurs avec LDAP. En effet, suite à la version de php qui était inférieur à celle demandée par NextCloud, nous avons dû installer une version ancienne de NextCloud qui fait que le plugin de php nécessaire à la connexion à Active Directory n'était plus disponible dans notre version de php.

---

### **- Problème avec le protocole BGP**

L'autre problème conséquent de la Saé fût la configuration et le déploiement de bgp entre les sites et les routeurs du cœur de réseau.

En effet, nous n'arrivons pas à ping de bout en bout entre deux sites. Ce problème était causé par l'absence de redistribution des routes de ospf à l'intérieur des sites sur le protocole bgp. Pour résoudre ce problème, nous sommes allés sur internet et sur le site de cisco afin de comprendre notre erreur. Nous avons alors remarqué qu'il était nécessaire d'utiliser la commande redistribute router ospf 10 internal 1 external 2 sur les routeurs du cœur de réseau. Ainsi les routes apprises par bgp étaient correctement retransmises au protocole ospf de cœur de réseau. Il nous était alors à présent possible de communiquer avec des appareils de deux sites différents.

---

## 9. Conclusion

Lors de cette Saé, nous avons appris plus que jamais à s'organiser et à se coordonner en vue de réaliser le travail demandé dans les temps. En effet étant en groupes de cinq personnes avec des spécialités différentes, nous devions réaliser des tâches en parallèle et en autonomie tout en gardant une cohésion de groupe tout cela grâce à notre Kanban.

Nous avons rencontré de nombreuses difficultés pendant ce projet de groupe que ce soit au niveau de l'infrastructure réseau, des serveurs d'applications ou encore pour la réalisation de la partie IoM.

Néanmoins, nous avons réussi à surmonter tous nos problèmes grâce à notre forte cohésion de groupe et au modèle de travail Agile. Si un des membres était en difficulté nous pouvions l'aider, si une tâche était bloquée nous pouvions continuer les autres sans difficulté en suivant ce modèle de travail moderne et flexible.

Le suivi méticuleux de cette méthode et de cet état d'esprit nous a permis d'arriver à nos objectifs en trouvant une solution à nos défis.

Il reste naturellement encore beaucoup de choses à améliorer dans notre projet, au niveau de l'infrastructure, une sécurité au niveau des ports des switchs peut largement être implémentée. Les services du serveur Active Directory sont encore à améliorer. La téléphonie IP reste encore à sécuriser et avoir une meilleure qualité de service .

Néanmoins, tous ces nouveaux défis seraient largement réalisables en suivant une méthode de travail rigoureuse ayant fait ses preuves, la méthode Agile.

## 10. Annexes

Voici le tableau d'adresses IP avec les vlan correspondants que nous avons utilisés pour notre projet :

Pour le siège :

Nom du réseau	Adresse IP du réseau	IP	Appareil
Réseau Complet	100.2.0.0/16	100.2.99.254/24	CE 00
VLAN par défaut 1	100.2.1.0/24	100.2.99.253/24	CE 01
VLAN Direction 10	100.2.10.0/24	10.0.2.2/30	CE00/CE01
VLAN Ventes 20	100.2.20.0/24	100.2.99.248/24	SW0
VLAN Voix 30	100.2.30.0/24	100.2.99.252/24	DHCP/DNS (Windows)
VLAN Wifi 40	100.2.40.0/24	100.2.99.251/24	Messagerie (Linux)
VLAN Gestion 99	100.2.99.0/24	100.2.99.252/24	Windows Terminal Server (Windows)
<b>Siège</b>		100.2.99.252/24	Radius/Active Directory (Windows)
		100.2.99.251/24	NextCloud Stockage et application (Linux)
		100.2.99.250/24	Serveur MQTT
		100.2.99.249/24	Access Point
		100.2.30.254/24	CE 00
		100.2.30.8/24	Téléphone
		100.2.30.253/24	Asterix/PBX

Pour le magasin :

Nom du réseau	Adresse IP du réseau	IP	Appareil
Réseau Complet	130.2.0.0/16	130.2.99.254/24	CE1
VLAN Direction 10	130.2.10.0/24	10.0.2.26/30	CE1
VLAN Ventes 20	130.2.20.0/24	130.2.99.253/24	SW1
VLAN Voix 30	130.2.30.0/24	130.2.99.252/24	DHCP (Windows)
VLAN Wifi 40	130.2.40.0/24	130.2.20.1/24	PC1
VLAN Gestion 99	130.2.99.0/24	130.2.10.1/24	PC2
<b>Magasin</b>		130.2.30.1/24	Téléphone
		130.2.40.251/24	AccessPoint

Pour le showroom :

Nom du réseau	Adresse IP du réseau	IP	Appareil
Réseau Complet	150.2.0.0/16	150.2.99.254/24	CE50
VLAN Direction 10	150.2.10.0/24	10.0.2.38/30	CE50
VLAN Ventes 20	150.2.20.0/24	150.2.99.253	SW50
VLAN Voix 30	150.2.30.0/24	150.2.99.252	DHCP (Windows)
VLAN Wifi 40	150.2.40.0/24	150.2.20.1/24	PC1
VLAN Gestion 99	150.2.99.0/24	150.2.10.1/24	PC2
<b>Showroom</b>		150.2.30.1/24	Téléphone
		150.2.40.254/24	AccessPoint
		150.2.40.1/24	Portail Captif

Pour le coeur de réseau simulant un opérateur télécoms :

Nom de l'appareil	Liaison	Interface	IP	Réseau
PE0	PE0 --> CE01/CE00	Fe0/0	10.0.2.1	10.0.2.0/30
PE0	PE0 --> PE1	S0/1	10.0.2.5	10.0.2.4/30
PE0	PE0 --> P1	S0/0	10.0.2.9	10.0.2.8/30
PE0	PE0 --> P2	S0/2	10.0.2.13	10.0.2.12/30
PE1	PE1 --> PE0	S0/0	10.0.2.6	10.0.2.4/30
PE1	PE1 --> PE50	S0/1	10.0.2.17	10.0.2.16/30
PE1	PE1 --> P2	S0/2	10.0.2.21	10.0.2.20/30
PE1	PE1 --> CE1	FE0/0	10.0.2.25	10.0.2.24/30
PE50	PE50 --> PE1	S0/1	10.0.2.18	10.0.2.16/30
PE50	PE50 --> P1	S0/0	10.0.2.29	10.0.2.28/30
PE50	PE50 --> P2	S0/2	10.0.2.33	10.0.2.32/30
PE50	PE50 --> CE50	Fe0/0	10.0.2.37	10.0.2.36/30
P1	P1 --> P2	S0/2	10.0.2.41	10.0.2.40/30
P1	P1 --> PE50	S0/0	10.0.2.30	10.0.2.28/30
P1	P1 --> PE0	S0/1	10.0.2.10	10.0.2.8/30
P2	P2 --> P1	S0/0	10.0.2.42	10.0.2.40/30
P2	P2 --> PE0	S0/1	10.0.2.14	10.0.2.12/30
P2	P2 --> PE1	S0/2	10.0.2.22	10.0.2.20/30
P2	P2 --> PE50	S0/3	10.0.2.34	10.0.2.32/30

Voici les captures d'écran de la configuration des vlan d'un switch du réseau :

Afin de vérifier les interfaces qui sont en mode trunk, nous avons utilisé la commande **sh interfaces trunk**

```
SWA#sh interfaces trunk

Port      Mode          Encapsulation  Status        Native vlan
Gi1/0/24  on           802.1q         trunking     1

Port      Vlans allowed on trunk
Gi1/0/24  10,20,30,40,99

Port      Vlans allowed and active in management domain
Gi1/0/24  10,20,30,40,99

Port      Vlans in spanning tree forwarding state and not pruned
Gi1/0/24  10,20,30,40,99
```

Afin de vérifier la création des vlans, nous avons utilisé la commande **sh vlan**

```
SWA#sh vlan

VLAN Name                           Status      Ports
---  -----
1    default                         active     Gi1/0/5, Gi1/0/6, Gi1/0/7
                                         Gi1/0/8, Gi1/0/9, Gi1/0/10
                                         Gi1/0/11, Gi1/0/12, Gi1/0/13
                                         Gi1/0/14, Gi1/0/15, Gi1/0/16
                                         Gi1/0/17, Gi1/0/18, Gi1/0/19
                                         Gi1/0/20, Gi1/0/22, Gi1/0/23
                                         Gi1/0/25, Gi1/0/26, Gi1/0/27
                                         Gi1/0/28
10   direction                       active     Gi1/0/1
20   vente                           active     Gi1/0/2
30   voix                            active     Gi1/0/3
40   wifi                            active     Gi1/0/4
99   gestion                          active     Gi1/0/21
1002 fddi-default                   act/unsup
1003 token-ring-default             act/unsup
1004 fddinet-default                act/unsup
1005 trnet-default                  act/unsup
```

Vous trouverez en pièce jointe la configuration du switch et d'un routeur.