

**PRÉPARÉ PAR**

Jean-michel BOUILLET; Alexis CHARTON

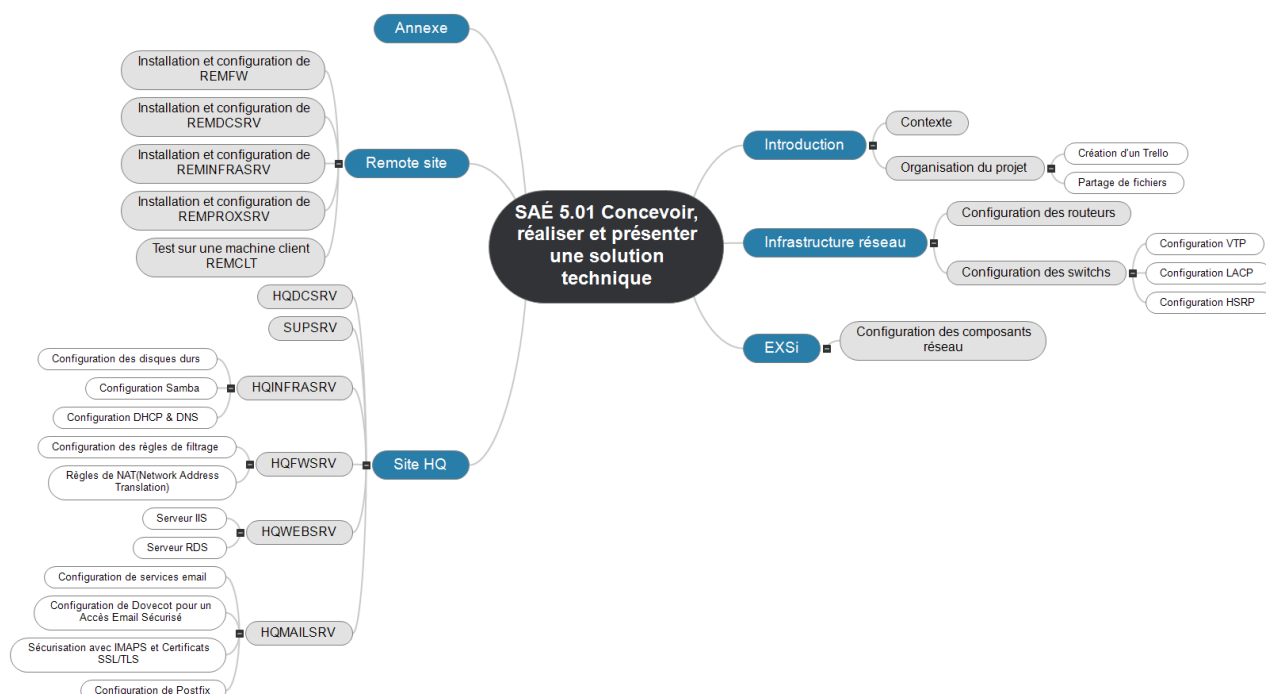
# **SAÉ 5.01**

## **CONCEVOIR, RÉALISER ET PRÉSENTER UNE SOLUTION TECHNIQUE**

CLÉMENT PETIT  
DALIANI ALI  
ILYAS TUNAY  
JULIEN WIEDERKEHR  
YOHANN MITEL  
YASSINE EL HAMIOUI

**22/12/2023**

# Concevoir, réaliser et présenter une solution technique



<b>1. Introduction</b>	<b>4</b>
1.1. Contexte	4
1.2. Organisation du projet	4
1.2.1. Création d'un Trello	4
1.2.2. Partage de fichiers	5
<b>2. Infrastructure réseau</b>	<b>7</b>
2.1. Configuration des routeurs	7
2.2. Configuration des switches	9
2.2.1. Configuration VTP	11
2.2.2. Configuration LACP	12
2.2.3. Configuration HSRP	13
<b>3. EXSi</b>	<b>14</b>
3.1. Configuration des composants réseau	14
<b>4. Site HQ</b>	<b>16</b>
4.1. HQDCSRV	16
4.2. SUPSRV	17
4.3. HQINFRASRV	20
4.3.1. Configuration des disques durs	20
4.3.2. Configuration Samba	21
4.3.3. Configuration DHCP & DNS	22
4.4. HQFWSRV	24
4.4.1. Configuration des règles de filtrage	24
4.4.2. Règles de NAT(Network Address Translation)	28
4.5. HQWEBSRV	31
4.5.1. Serveur IIS	31
4.5.2. Serveur RDS	37

4.6. HQMAILSRV .....	40
4.6.1. Configuration de services email.....	40
4.6.2. Configuration de Dovecot pour un Accès Email Sécurisé .....	40
4.6.3. Sécurisation avec IMAPS et Certificats SSL/TLS.....	41
4.6.4. Configuration de Postfix.....	42
<b>5. Internet .....</b>	<b>46</b>
5.1. Contexte du projet internet.....	46
5.2. Installation et configuration de REMFW.....	46
5.3. Configuration du serveur FTP avec VSFTPD .....	47
5.4. Configuration de VSFTPD.....	48
5.5. Configuration de l'environnement Web pour worldskills.org .....	48
5.6. Difficultés rencontrées.....	48
5.7. Solutions apportées .....	49
5.8. Conclusion de la configuration de l'environnement Web .....	49
5.9. Etapes de configuration .....	49
5.10. Difficultés rencontrées.....	50
5.11. Solutions apportées .....	50
5.12. Conclusion de la configuration de l'infrastructure HA .....	50
5.13. Récapitulation des configurations réussies.....	50
5.14. Perspectives d'amélioration .....	51
<b>6. Remote site .....</b>	<b>52</b>
6.1. Installation et configuration de REMFW.....	53
6.2. Installation et configuration de REMDCSRV.....	56
6.3. Installation et configuration de REMINFRASRV .....	59
6.4. Installation et configuration de REMPROXSRV.....	61
6.5. Test sur une machine client REMCLT .....	63
<b>7. Conclusion .....</b>	<b>65</b>
<b>8. Annexe.....</b>	<b>66</b>
8.1. Configuration routeur WANRTR.....	66
8.2. Configuration routeur EDGE1 .....	66
8.3. Configuration routeur EDGE2 .....	71
8.4. Configuration CORESW1.....	75
8.5. Configuration CORESW2.....	81
8.6. Configuration ACCSW1 .....	85
8.7. Configuration ACCSW2 .....	92

## 1. Introduction

Cette SAÉ 5.01, intitulée *Concevoir, réaliser et présenter une solution technique*, est encadrée par M.BOUILLET ainsi que M.CHARTON. Nous avons réalisé ce projet à l'IUT de Montbéliard du lundi 11 décembre au vendredi 22 décembre tout en se consacrant à la rédaction de ce rapport ainsi que sur la soutenance orale mais également sur les différents jalons à rendre afin d'avancer étape par étape dans ce projet.

Durant cette semaine de travail, nous avons pu mettre en œuvre tout ce que nous avons vu lors des ressources que nous avons pu avoir durant cette troisième année de BUT Réseaux et Télécommunications mais également des cours de deuxième année. L'objectif de ce projet est de mettre en place une infrastructure complète système et réseau basée sur des équipements réseaux réels et plusieurs serveurs de machines virtuelles (Proxmox VE et ESXi) dont le sujet s'inspire des Worldskills de cette année.

### 1.1. Contexte

Notre groupe, composé de six membres, a travaillé sur le cœur de réseau mais également sur le déploiement des différents services réseau sur plusieurs serveurs sous forme de machines virtuelles à l'aide des serveurs Proxmox et ESXi. Nous nous sommes concentrés sur les aspects techniques et de sécurité de ce projet afin de réaliser notre objectif principal qui était de démontrer la viabilité et l'efficacité de notre infrastructure.

### 1.2. Organisation du projet

Pour mener à bien ce projet, nous avons utilisé plusieurs outils afin de nous organiser au mieux dans ce projet.

#### 1.2.1. Création d'un Trello

Nous avons mis en œuvre un outil de gestion de projet à l'aide de Trello qui est une plateforme de gestion de projet en ligne qui aide à organiser et à suivre le travail en équipe. Son principal objectif est de permettre aux utilisateurs de créer des tableaux virtuels où ils peuvent ajouter des listes, des cartes et des éléments, les déplacer facilement et collaborer avec d'autres personnes.

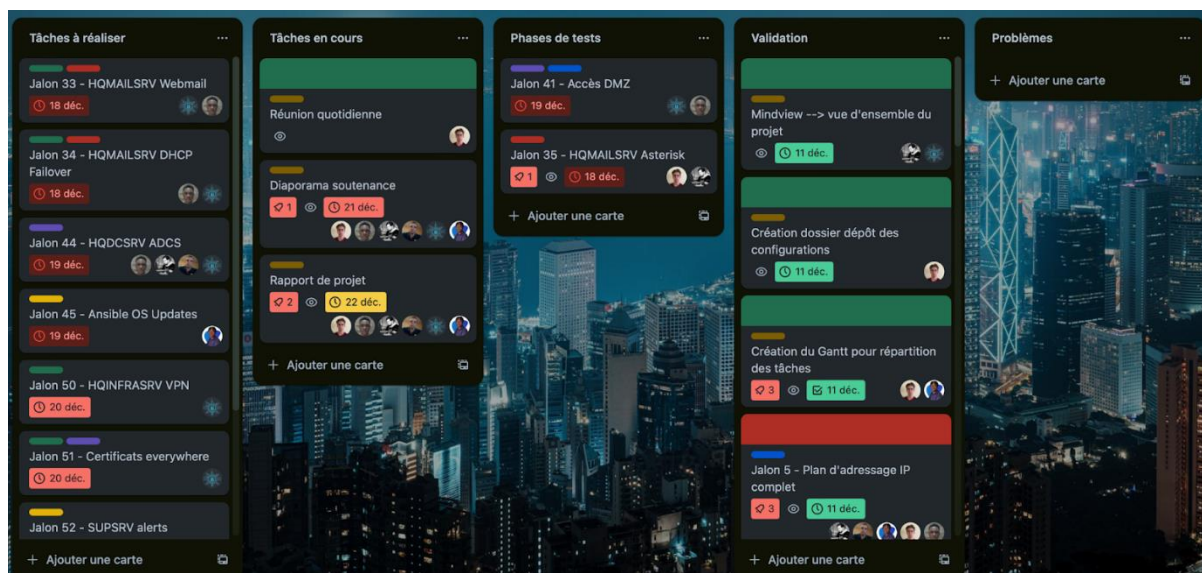


Figure 1 : Organisation du Trello

Au sein de notre Trello, nous avons créé 6 listes dans l'optique d'avoir une organisation optimale et cohérente dans le groupe. La première liste contient la légende des tâches que nous avons à réaliser afin de voir en un coup d'œil le domaine de travail auquel une tâche est rattachée. Ensuite, nous retrouvons les tâches à réaliser où une carte représente un jalon avec

la date limite pour le réaliser ainsi que les membres du groupe qui y sont affectés, ce qui permet à chacun de voir quelles tâches a-t-il à réaliser dans la journée. De plus, une colonne avec les tâches en cours de réalisation a été créée pour voir quelles sont les tâches en cours de réalisation et si nous ne sommes pas en retard pour les réaliser afin de prioriser des tâches plus importantes. Par ailleurs, nous avons créé une colonne appelée validation où nous plaçons les tâches pour lesquelles nous devons réaliser des tests afin de les valider. Nous avons mis en place une colonne validation où nous pouvons voir en un coup d'œil les tâches que nous avons terminées, ce qui permet de passer plus rapidement à la réalisation d'autres tâches. Pour finir, une colonne nous permettant de visualiser les tâches où nous avons pu rencontrer des problèmes, nous nous sommes occupés en priorité des problèmes avant de passer à d'autres tâches dans le but de présenter des éléments fonctionnels.

### 1.2.2. Partage de fichiers

Nous avons partagé un *Drive* entre tous les membres du groupe afin de centraliser et d'avoir accès à tous les fichiers nécessaires pour le bon déroulement du projet.




















Nom ↑	Propriétaire	Dernière modification ▼	Taille du fich	
 Configurations	 moi	11 déc. 2023 moi	—	    
 Plan adressage IP	 moi	20 déc. 2023 moi	—	
 Notes Quotidiennes 	 mitelychann971@gmail.c...	15:05 moi	143 Ko	
 Rapport 	 moi	10:26 moi	1 Ko	

Figure 2 : Contenu du drive

Nous retrouvons le dossier *Configurations* dans lequel nous avons créé deux dossiers, à savoir *Routeurs* et *Switchs* contenant tous les deux les configurations des équipements réseau. Cela nous a permis d'analyser l'évolution de nos configurations afin d'identifier les éléments dans la configuration qui nous posait problème. Le dépôt des différentes configurations de nos équipements réseau avait un autre intérêt en cas de la perte des la configuration sur un switch, nous avons tout de même une version récente de la configuration sur le drive et donc par extension de ne pas perdre de temps à se rappeler ce que nous avons configuré sur le switch.

Partagés avec moi > ... > Switchs > CORESW1 

Type  Date de modification ▼

























Nom ↑	Propriétaire	Dernière modification ▼	Taille du fich	
 CORESW1_V1.txt 	 moi	15 déc. 2023 moi	4 Ko	
 CORESW1_V2.txt 	 moi	15 déc. 2023 moi	5 Ko	
 CORESW1_V3.txt 	 moi	18 déc. 2023 moi	4 Ko	
 CORESW1_V4.txt 	 moi	19 déc. 2023 moi	8 Ko	
 CORESW1_V5.txt 	 moi	18:01 moi	8 Ko	    

Figure 3 : Fichiers configuration CORESW1

Nous avons mis à disposition notre plan d'adressage IP afin que le schéma de l'infrastructure réseau pour tous les membres du groupe. Cela permet de visualiser les plages d'adresses pour nos différents sous-réseaux, d'organiser de manière efficace et logique les adresses IP afin de permettre la communication entre les différents appareils connectés à un réseau où l'objectif est de ne pas réutiliser une adresse IP similaire sur deux machines pour ne pas créer de conflit d'adresse.

Fonction	Position	Nom machine	FQDN/ ALIAS	IP /FQDN
serveur	HQ	hqdcsvr		10.3.10.1
serveur	HQ	hqinfrsfrsv		10.3.20.1 & 10.3.10.3
serveur	HQ	hqmailsrv		10.3.10.2
serveur	HQ	hqfwsrv		217.3.160.1 / 10.3.10.4 / 10.3.30.2
serveur	HQ	supsrv		10.3.99.1
serveur	HQ	hqwebsrv		10.3.30.1
serveur	remote	remdcsvr		10.3.100.1
serveur	remote	reminfrsfrsv		10.3.100.2
serveur	remote	remproxsvr		10.3.100.3
serveur	internet	dnssrv		8.8.3.1
serveur	internet	inetsrv1		8.8.3.2
		inetsrv2		8.8.3.3
Switch		CORESW1 VLAN 10		10.3.10.60
		CORESW1 VLAN 20		10.3.20.252
		CORESW1 VLAN 30		217.3.160.252
		CORESW1 VLAN 99		10.3.99.252
		CORESW1 VLAN 100		10.3.254.2
Switch		CORESW2 VLAN 10		10.3.10.61
		CORESW2 VLAN 20		10.3.20.253
		CORESW2 VLAN 30		217.3.160.253
		CORESW2 VLAN 99		10.3.99.253
		CORESW2 VLAN 200		10.3.254.18
Switch		ACCSW1 VLAN 99		10.3.99.241

Figure 4 : Plan d'adressage IP

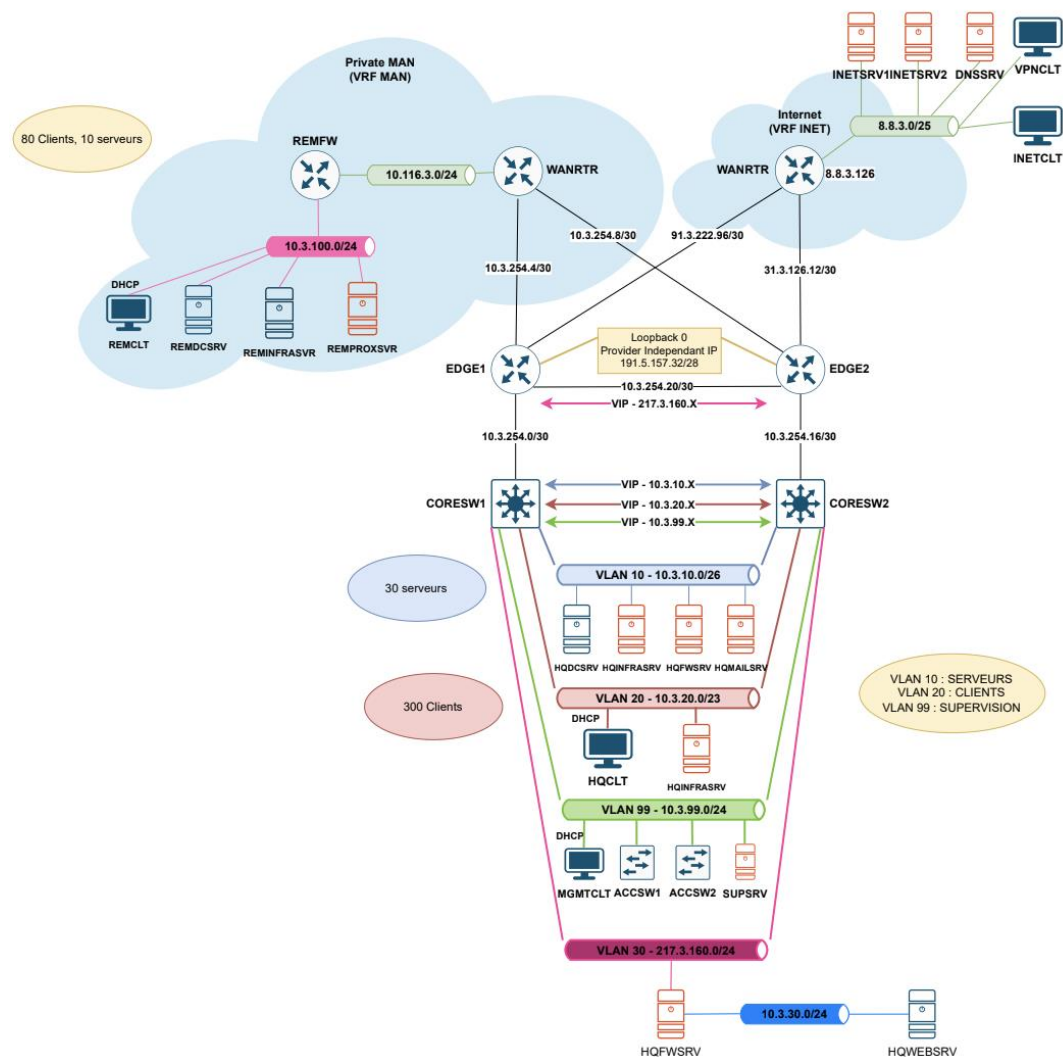


Figure 5 : Schéma infrastructure réseau



Nous retrouvons le fichier *Notes quotidiennes* qui permet à tous les membres du groupe de parler plus en détails de ce qu'ils ont fait dans la journée et des problèmes rencontrés. Nous nous sommes basés sur ce fichier pour rédiger le contenu des réunions quotidiennes car nous essayons d'apporter le maximum d'informations à l'oral tout en étant concis. Nous évoquons également l'avancement des tâches réalisées dans la journée.

## 2. Infrastructure réseau

### 2.1. Configuration des routeurs

Pour la partie cœur de réseau au niveau des routeurs, nous avons donc 3 routeurs à disposition :

- 1 Routeur 1941 avec 4 interfaces Ethernet (WANRTR)
- 2 Routeurs 2901 avec 2 interfaces Ethernet (EDGE1 et EDGE2)

L'objectif de ces différents routeurs est donc d'établir une bonne connectivité qui est efficace, redondante entre les différents sites de l'infrastructure. Ainsi, assurer l'accès à Internet, et déployer divers protocoles de routage, comme OSPF et BGP.

Premièrement, on configure les interfaces des routeurs en fonction de notre adressage IP et effectue les branchements, voici ce que cela donne :

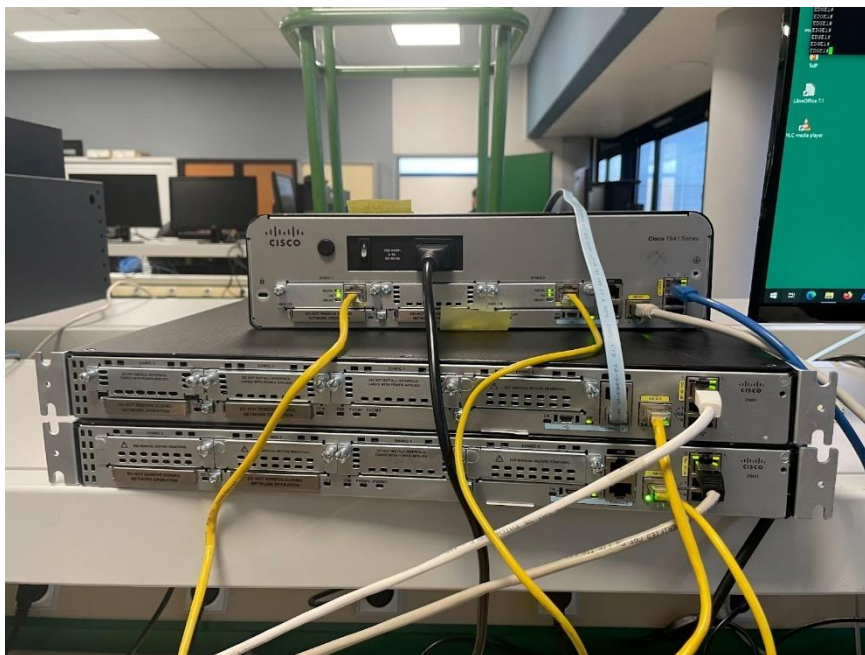


Figure 6 : Mise en place routeurs réels

Ensuite, on effectue étape par étape les différentes configuration des protocoles de routage. Dans notre infrastructure, pour optimiser le cœur de réseau, on met en place le protocole VRF (Virtual Routing Forwarding) sur le routeur WANRTR. On met donc en place deux VRF distincts.

- Un VRF MAN pour la gestion du trafic interne qui est privé. Cette configuration favorise la création de liaisons OSPF entre les routeurs REMFW (routeur virtuel dans Internet), EDGE et WANRTR.

- Un VRF INET qui favorise la connexion Internet des routeurs EDGE. Tout cela en effectuant une connexion eBGP entre les routeurs EDGE et le WANRTR. Puis, du iBGP entre les routeurs EDGE qui permettra d'assurer une connexion optimale, même en cas de défaillance.

eBGP sera conçu pour effectuer la communication entre des systèmes autonomes externes, tandis que iBGP assurera une connectivité à l'intérieur du même système. Tout cela en garantissant une connexion approfondie à travers le réseau interne.

Voici comment on crée un VRF dans le routeur WANRTR:

```
!
ip vrf INET
 rd 65330:2
!
ip vrf MAN
 rd 65330:1
!
!
```

*Figure 7 : Configuration VRF*

On crée donc deux VRF avec un rd unique, ce rd sert donc à différencier de manière correcte les routes.

De plus, on précise par la suite, selon les bonnes interfaces, s'il correspond au côté MAN ou au côté INET afin de bien diviser le trafic.

```
interface FastEthernet0/0/0.20
 encapsulation dot1q 20
 ip vrf forwarding INET
 ip address 31.3.126.13 255.255.255.252
!
interface FastEthernet0/1/0
 ip vrf forwarding MAN
 ip address 10.3.254.6 255.255.255.252
 duplex auto
 speed auto
```

*Figure 8 : Configuration des interfaces avec VRF*

Par la suite, pour garantir une redondance au niveau des routeurs EDGE, on met en place ensuite le protocole HSRP ( Hot Standby Router Protocol). Lors de la configuration de HSRP, on attribue une valeur de priorité au routeur. La valeur la plus élevée est prioritaire. Dans notre cas, EDGE1 à une priorité à 110 et EDGE2 une priorité à 100.

En configurant cela, on peut vérifier le fonctionnement de la redondance avec la commande "show standby".



```

EDGE1#show standby
GigabitEthernet0/0.30 - Group 1
  State is Active
    2 state changes, last state change 00:02:08
  Virtual IP address is 217.3.160.249
  Active virtual MAC address is 0000.0c07.ac01
  Local virtual MAC address is 0000.0c07.ac01 (v1 default)
  Hello time 3 sec, hold time 10 sec
  Next hello sent in 0.864 secs
  Preemption enabled
  Active router is local
  Standby router is 217.3.160.251, priority 100 (expires in 9.136 sec)
  Priority 110 (configured 110)
  Group name is "hsrp-Gi0/0.30-1" (default)

```

Figure 9 : Show standby EDGE1

```

EDGE2#show standby
GigabitEthernet0/0.30 - Group 1
  State is Standby
    1 state change, last state change 00:00:50
  Virtual IP address is 217.3.160.249
  Active virtual MAC address is 0000.0c07.ac01
  Local virtual MAC address is 0000.0c07.ac01 (v1 default)
  Hello time 3 sec, hold time 10 sec
  Next hello sent in 0.896 secs
  Preemption enabled
  Active router is 217.3.160.250, priority 110 (expires in 8.560 sec)
  Standby router is local
  Priority 100 (default 100)
  Group name is "hsrp-Gi0/0.30-1" (default)

```

Figure 10 : Show standby EDGE2

On voit donc la priorité sur les routeurs et que le routeur EDGE1 est en mode actif contrairement au EDGE2 qui est en standby.

Finalement, on va mettre en place de la NAT/PAT, pour permettre l'accès à Internet depuis les sous-réseaux privés.

## 2.2. Configuration des switches

Dans notre infrastructure réseau, nous avons 4 switches dont 2 switches de niveau 3 (*Cisco Catalyst 3750*) ainsi que 2 switches de niveau 2 (*Cisco Catalyst 2960*). Sur chaque switch, nous avons dû ajouter un mot de passe chiffré de sorte à ce qu'il n'apparaisse pas en clair dans la configuration. Pour cela, nous utilisons la commande **enable secret P@ssw0rd**. Par la suite, nous avons créé plusieurs VLANs dont :

- VLAN 10 utilisé pour les serveurs

- VLAN 20 utilisé pour les clients
- VLAN 30 utilisé pour la DMZ
- VLAN 99 utilisé pour la supervision
- VLAN 100 utilisé pour la connexion entre CORESW1 et EDGE1
- VLAN 200 utilisé pour la connexion entre CORESW2 et EDGE2
- VLAN 300 utilisé pour l'interconnexion iBGP entre EDGE1 et EDGE2
- VLAN 666 qui est le VLAN par défaut

Les clients seront sur les switchs *ACCSW*, 4 ports seront configurés en mode *Access* pour 4 VLANs différents dont :

- gigabitEthernet2/0/3 pour le VLAN 10
- gigabitEthernet2/0/4 pour le VLAN 20
- gigabitEthernet2/0/5 pour le VLAN 30
- gigabitEthernet2/0/6 pour le VLAN 99

Sur ces ports, seulement 3 adresses MAC seront apprises, si ce nombre est dépassé les ports seront désactivés. Pour réaliser cela, nous devons configurer les ports en mode *Access* avec la commande **switchport mode access**. Ensuite, nous devons autoriser uniquement des machines authentifiées sur ces ports avec la commande **switchport port-security**. Pour finir, nous devons limiter le nombre d'adresses MAC que le port peut apprendre à l'aide de la commande **switchport port-security maximum 3** et si le nombre d'adresses MAC est dépassé, on éteint le port avec la commande **switchport port-security violation shutdown**.

Voici le schéma de niveau 2 de notre infrastructure afin de visualiser tous les *Trunk*.

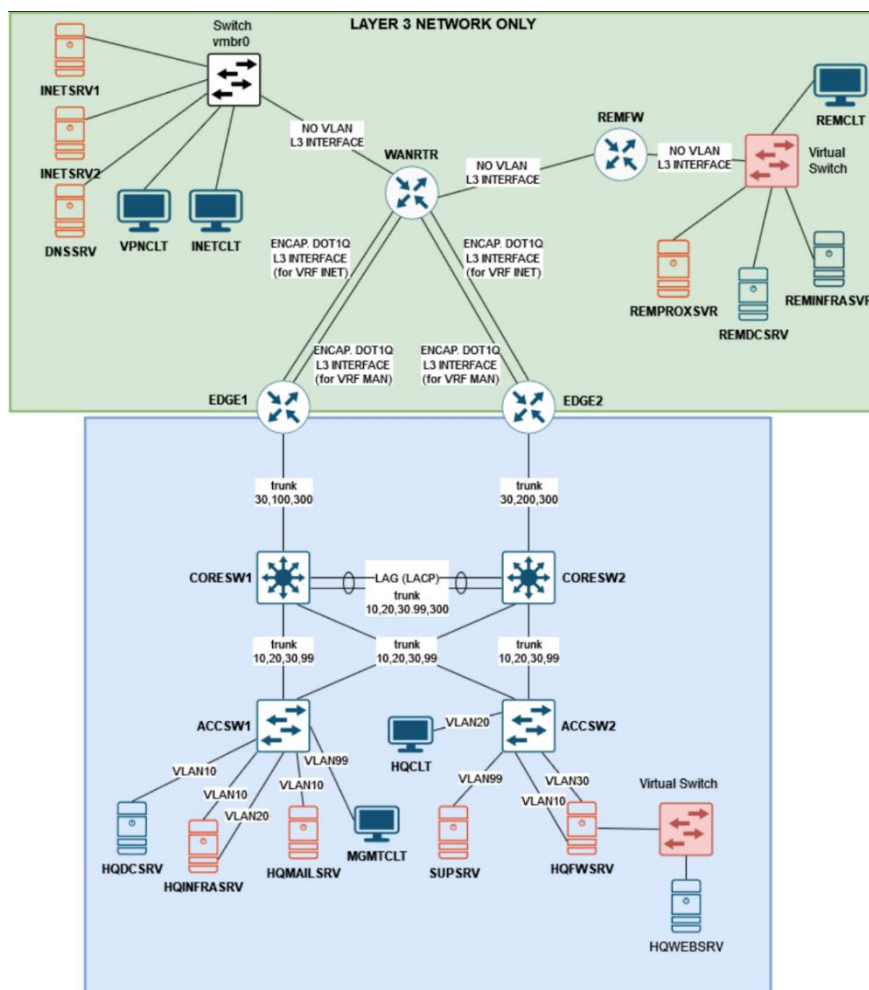


Figure 11 : Schéma infrastructure réseau niveau 2

### 2.2.1. Configuration VTP

VTP est un protocole réseau utilisé dans les environnements Cisco pour faciliter la gestion des VLANs. Quand nous configurons un nouveau VLAN sur un serveur VTP, le VLAN est distribué pour tous les commutateurs dans le domaine. Le serveur tient à jour une table de VLANs déclarés où celle-ci est diffusée à l'ensemble des clients étant sur le même domaine VTP. De ce fait, chaque modification de la table est répercutée à l'ensemble des clients. Ainsi tous les VLANs définis sur le serveur pourront transiter par l'ensemble des ports trunk des switches clients, ce qui réduit la nécessité de configurer le même VLAN partout. Nous retrouvons 3 types de modes, à savoir *server*, *client* ainsi que *transparent*. En mode *server*, le switch est associé à un domaine VTP et la déclaration des VLANs s'effectue sur le serveur. Il tient à jour la liste des VLANs déclarés et la diffuse à l'ensemble des clients présents dans le même domaine. En mode *client*, le switch est associé au même domaine VTP que le serveur, il reçoit la liste des VLANs. En mode *transparent*, le switch n'est associé à aucun domaine VTP. Sa liste de VLAN est locale et n'est pas mise à jour lorsqu'il reçoit une trame VTP, cependant, il propage les listes de VLAN qu'il reçoit. Dans notre cas, nous avons configuré les switches CORESW en mode *server* et les switches ACCSW en mode *client*. Nous avons rencontré un problème lors de la configuration de VTP car les versions de VTP étaient différentes d'un switch à un autre donc nous les avons configurées en version 2. De plus, les switches ACCSW ne récupéraient pas les VLANs des serveurs car nous n'avions pas configuré les ports en mode *Trunk*.

Pour configurer VTP, nous devons ajouter 3 commandes :

- **vtp domain wsl2024.org**
- **vtp passwd P@ssw0rd**
- **vtp mode server** ou **vtp mode client**

```
CORESW1#sh vtp status
VTP Version capable      : 1 to 3
VTP version running      : 2
VTP Domain Name          : wsl2024.org
VTP Pruning Mode         : Disabled
VTP Traps Generation     : Enabled
Device ID                : 0817.3573.4a00
Configuration last modified by 10.3.10.29 at 3-1-93 23:51:59
Local updater ID is 10.3.10.60 on interface Vl10 (lowest numbered VLAN interface found)

Feature VLAN:
-----
VTP Operating Mode       : Server
Maximum VLANs supported locally : 1005
Number of existing VLANs : 13
Configuration Revision   : 1
MD5 digest               : 0xE2 0xFA 0xD2 0xC1 0x77 0xF9 0x4F 0x0F
                        : 0xFB 0x62 0x50 0x4D 0xA5 0x9B 0x7E 0xA0
```

Figure 12 : Configuration VTP CORESW1

```
ACCSW1#sh vtp status
VTP Version capable      : 1 to 3
VTP version running      : 2
VTP Domain Name          : wsl2024.org
VTP Pruning Mode         : Disabled
VTP Traps Generation     : Enabled
Device ID                : 20bb.c0a6.d280
Configuration last modified by 10.3.10.29 at 3-1-93 23:51:59

Feature VLAN:
-----
VTP Operating Mode       : Client
Maximum VLANs supported locally : 255
Number of existing VLANs : 13
Configuration Revision   : 1
MD5 digest               : 0xE2 0xFA 0xD2 0xC1 0x77 0xF9 0x4F 0x0F
                        : 0xFB 0x62 0x50 0x4D 0xA5 0x9B 0x7E 0xA0
```

Figure 13 : Configuration ACCSW1

## 2.2.2. Configuration LACP

Le protocole *LACP* (*Link Aggregation Control Protocol*) permet la mise en place d'agrégation de liens qui permet de regrouper plusieurs liens physiques en un seul lien logique et ainsi améliorer les performances en termes de bande-passante, de haute disponibilité et de répartition de charge.

Nous avons configuré LACP sur *CORESW1* ainsi que *CORESW2* sur les interfaces *fastEthernet1/0/2* et *1/0/3*. Tout d'abord, nous devons créer le groupe EtherChannel 1 avec le protocole LACP en mode *active*, c'est-à-dire que les interfaces en mode *active* enverront des requêtes LACP pour former le groupe agrégé. Les interfaces en mode *passive* attendront de recevoir des requêtes LACP d'une interface configurée en mode *active* pour former le groupe agrégé. Pour que EtherChannel fonctionne avec LACP, il faut être *active/active* ou *active/passive*. Pour cela, nous utilisons la commande **channel-group 1 mode active**. Ensuite, nous avons passé l'interface *port-channel 1* en mode *Trunk* pour faire passer plusieurs VLANs à l'aide des commandes **switchport mode trunk**, **switchport trunk encapsulation dot1q** qui indique au port que la méthode d'encapsulation des trames est *IEEE 802.1Q* qui permet de modifier la trame Ethernet au niveau de la couche 2 et de taguer les trames pour identifier les VLANs, **switchport trunk allowed vlan 10,20,30,99,300,666**, **switchport nonegotiate**.

```

!
interface FastEthernet1/0/2
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 10,20,30,99,300,666
 switchport mode trunk
 switchport nonegotiate
 channel-group 1 mode active
!
interface FastEthernet1/0/3
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 10,20,30,99,300,666
 switchport mode trunk
 switchport nonegotiate
 channel-group 1 mode active
!

```

Figure 14 : Configuration LACP

### 2.2.3. Configuration HSRP

HSRP (Hot Standby Router Protocol) est un protocole de routage utilisé pour offrir une redondance au niveau des passerelles par défaut. Sur des switches de niveau 3, HSRP permet à ces switches de travailler ensemble afin de fournir une passerelle par défaut virtuelle pour permettre aux appareils du réseau de communiquer avec la passerelle par défaut, ce qui garantit une certaine disponibilité du réseau en cas de défaillance. Lorsque plusieurs switches sont configurés pour utiliser HSRP, un seul est actif à la fois, tandis que les autres restent en veille. Le switch actif est celui qui répondra aux requêtes des appareils du réseau pour les données provenant d'autres réseaux. En cas de ce dernier, un autre switch configuré avec HSRP prendra automatiquement le relais pour assurer la continuité des communications sans qu'il y ait de coupure perceptible pour les utilisateurs.

Nous avons configuré HSRP sur les switches *CORESW1* et *CORESW2*. Tout d'abord, nous devons créer un groupe pour chaque VLAN contenant les deux switches (le groupe du VLAN 10 sera le groupe 1) mais également définir l'adresse IP virtuelle que les switches vont se partager qui sera la passerelle à l'aide de la commande **standby 1 ip 10.3.10.62**. Le switch actif est celui qui possède la priorité (comprise entre 0 et 255) la plus haute, par défaut elle est à 100. Le switch actif sera *CORESW1* et le switch en mode standby sera *CORESW2*. Nous allons fixer la priorité du *CORESW1* à l'aide de la commande **standby 1 priority 110**. Nous devons activer l'option preempt qui permet au switch actif de reprendre son rôle après une panne. Le switch en mode *standby* qui remplaçait le switch actif lors d'une panne de ce dernier retournera en mode *standby* si le switch actif revient en ligne. Pour cela, nous utilisons la commande **standby 1 preempt**.

```

CORESW1#sh standby
Vlan10 - Group 1
  State is Active
    17 state changes, last state change 00:48:53
  Virtual IP address is 10.3.10.62
  Active virtual MAC address is 0000.0c07.ac01
    Local virtual MAC address is 0000.0c07.ac01 (v1 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 0.432 secs
  Preemption enabled
  Active router is local
  Standby router is 10.3.10.61, priority 100 (expires in 10.800 sec)
  Priority 110 (configured 110)
  Group name is "hsrp-Vl10-1" (default)

```

Figure 15 : Configuration HSRP CORESW1

```

CORESW2#sh standby
Vlan10 - Group 1
State is Standby
  5 state changes, last state change 00:43:33
Virtual IP address is 10.3.10.62
Active virtual MAC address is 0000.0c07.ac01
Local virtual MAC address is 0000.0c07.ac01 (v1 default)
Hello time 3 sec, hold time 10 sec
  Next hello sent in 2.256 secs
Preemption enabled
Active router is 10.3.10.60, priority 110 (expires in 9.392 sec)
Standby router is local
Priority 100 (default 100)
Group name is "hsrp-Vl10-1" (default)

```

Figure 16 : Configuration HSRP CORESW2

## 3. EXSi

Au cours du projet, l'infrastructure a impliqué l'utilisation de deux hôtes ESXi sur lesquels sont stockées plusieurs machines virtuelles. Pour assurer une gestion efficace du réseau virtuel, des port groups et des switches ont été créés. Ces éléments ont joué un rôle clé dans l'organisation et la connectivité des machines virtuelles.

Cartes Physiques : Deux cartes physiques ont été utilisées, l'une pour la gestion (vmnic0) et l'autre pour la connectivité à l'infrastructure réel (vmnic1).

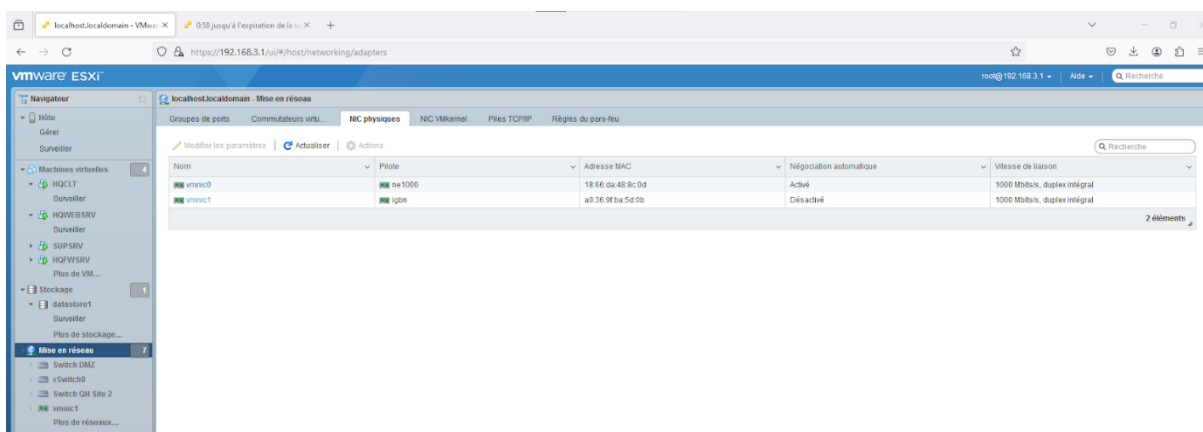


Figure 17 : Carte physique ESXi

### 3.1. Configuration des composants réseau

Port Groups : Des ports groups ont été configurés pour segmenter et organiser le trafic réseau, facilitant ainsi la gestion des communications entre les machines virtuelles et le réseau physique.

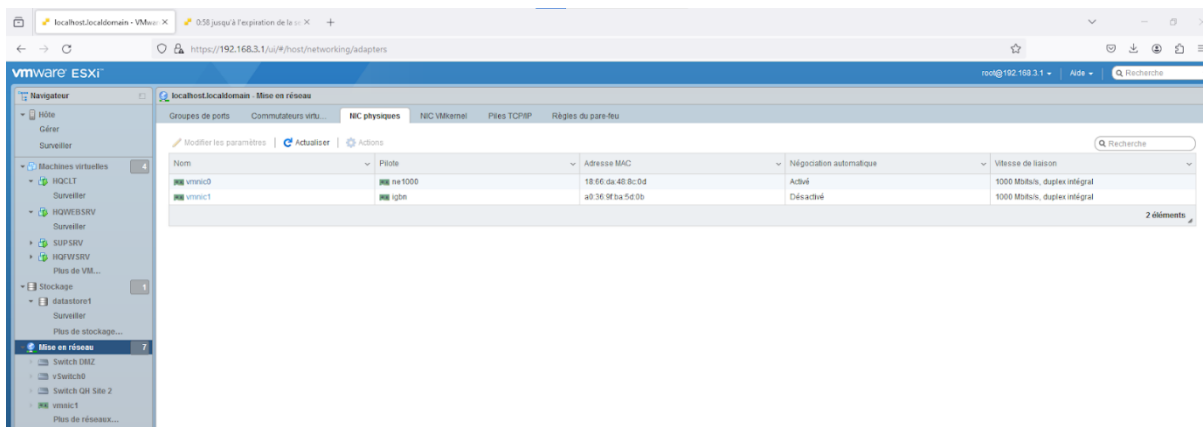




Figure 18 : Portgroup (VLAN) ESXi

Switchs Virtuels : La création de switchs virtuels a permis de relier et de diriger le trafic au sein des machines virtuelles et entre les différents hôtes ESXi, assurant ainsi une connectivité fluide et sécurisée.

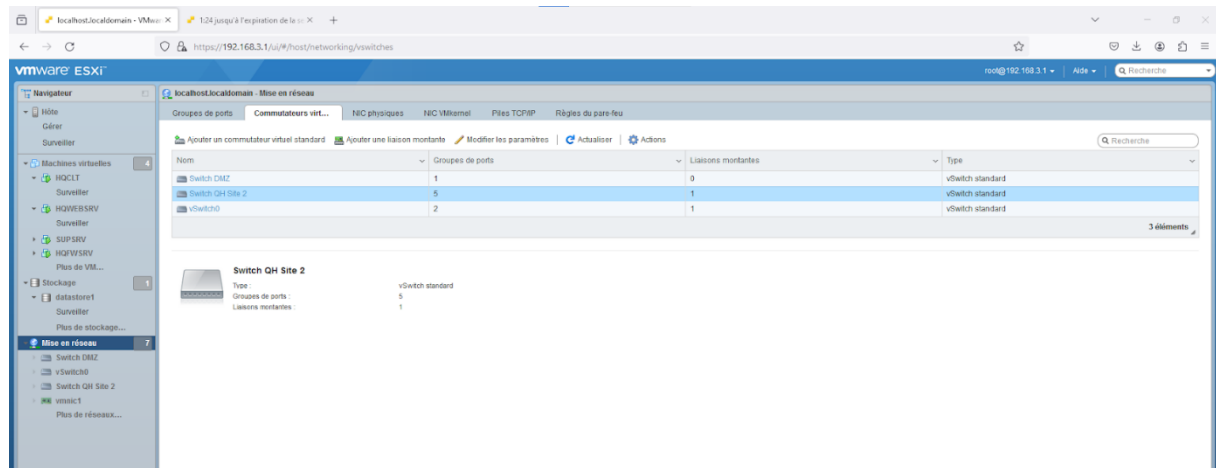


Figure 19 : Switchs Virtuels ESXi

Machines Virtuelles : Plusieurs machines virtuelles ont été déployées sur les hôtes ESXi, chacune étant connectée aux port groups appropriés pour assurer des communications efficaces et sécurisées.

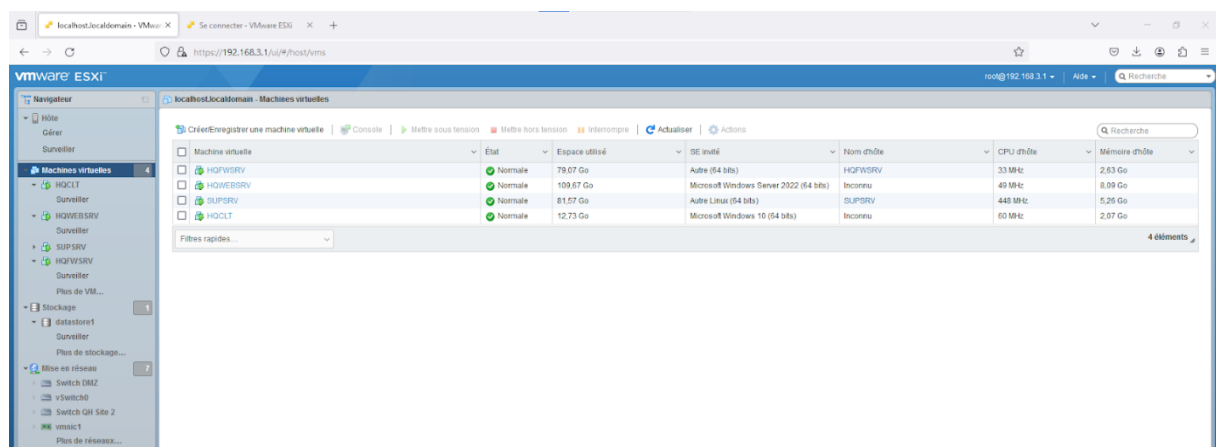


Figure 20 : Machines Virtuelles ESXi

La capture d'écran du switch, détaillant les machines connectées ainsi que l'utilisation des VLAN, offre une représentation visuelle complète de la topologie réseau.

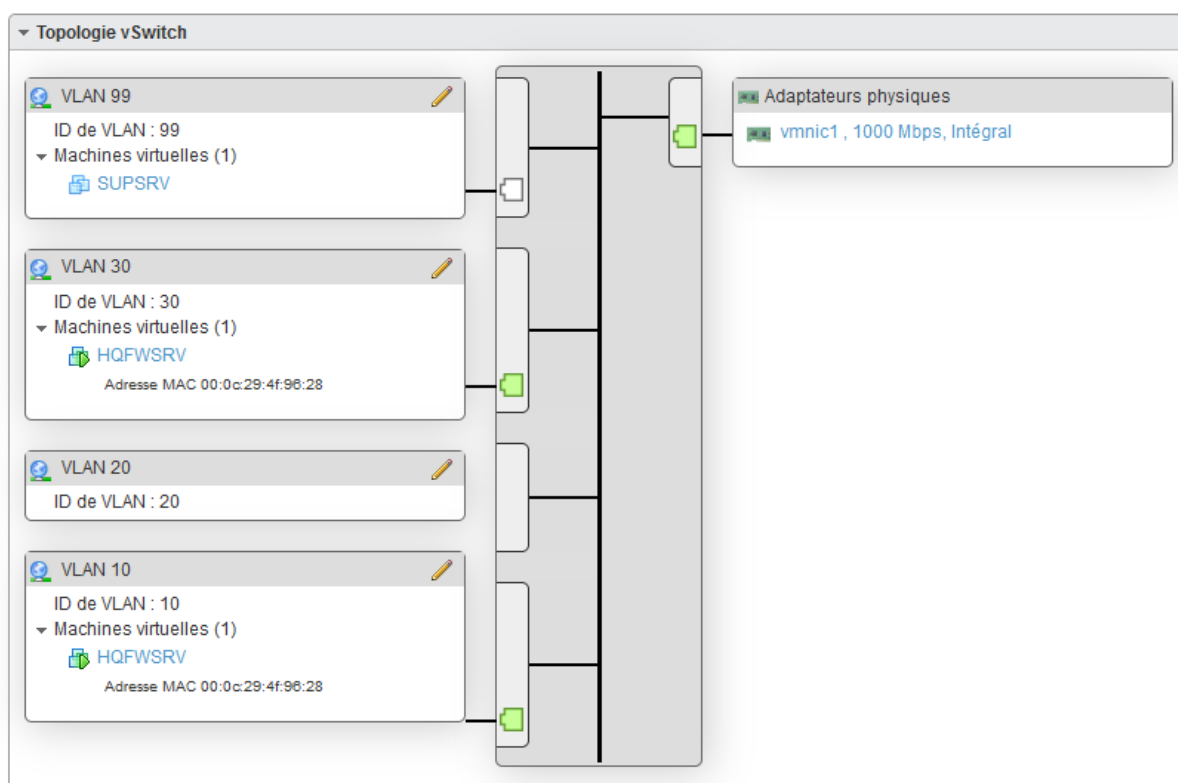


Figure 21 : Schéma d'un switchs virtuelles et des différentes connexions

Cette capture illustre clairement la disposition des machines virtuelles, réparties dans différents VLAN. Cette représentation visuelle offre un aperçu précis de la configuration réseau, mettant en valeur la structure des VLAN et la connectivité des machines virtuelles dans l'ensemble de l'infrastructure.

La connectivité globale sur le switch a été assurée pour garantir un fonctionnement harmonieux et sécurisé de l'infrastructure virtuelle. Ces configurations ont été cruciales pour l'optimisation des performances et la gestion efficace du réseau dans le cadre du projet.

## 4. Site HQ

### 4.1. HQDCSRV

Nous avons commencé par téléverser l'ISO de Windows Server sur l'ESXi, puis nous avons procédé à la création d'une machine virtuelle Windows Server. Une fois la VM Windows créée, nous avons mis en place un serveur DNS pour la zone `hq.wsl2024.org`. Toutes les autres demandes DNS sont redirigées vers le serveur DNS de présent sur HQINFRASRV et ayant pour domaine `wsl2024.org`.

Après la création du serveur DNS, nous avons continué avec la configuration d'Active Directory. Cette phase a impliqué la création de divers scripts, notamment des scripts de création de 1000 utilisateurs et de suppression de ces utilisateurs. Nous avons dû suivre des règles strictes, notamment la représentation du site HQ par une unité organisationnelle contenant les OU suivantes :

- "UsersHQ" contenant des utilisateurs, avec une sous-OU pour chaque département. Les utilisateurs appartenant à chaque département ont été placés dans cette sous-OU.
- "Computers" contenant les objets ordinateurs. Les ordinateurs et serveurs (à l'exception du contrôleur de domaine) appartenant à ce site ont été placés dans cette OU.
- L'OU "Groups" contenant tous les groupes appartenant à ce site.

- Création d'une OU à la racine nommée "Groupes d'ombre" et d'un groupe global nommé "OU\_Shadow" contenant tous les utilisateurs de l'OU "HQ". Lors de la création d'un nouvel utilisateur pour ce site, l'utilisateur a été automatiquement ajouté à ce groupe dans la minute suivante.

En ce qui concerne la provision des utilisateurs, nous avons créé 1000 utilisateurs avec les noms de compte SamAccountName/UserPrincipalName de la forme wslusrXXX, où wslusr001 est le premier utilisateur et wslusr1000 est le dernier. Ces utilisateurs ont été placés dans l'unité organisationnelle Active Directory suivante : OU=AUTO,OU=USERS,DC=wsl2024,DC=org. De plus, les 500 premiers utilisateurs ont été ajoutés au groupe global de sécurité "FirstGroup", tandis que les 500 derniers utilisateurs ont été ajoutés au groupe global de sécurité "LastGroup". Les groupes ont été situés dans l'unité organisationnelle suivante : OU=Groupes,DC=wsl2024,DC=org.

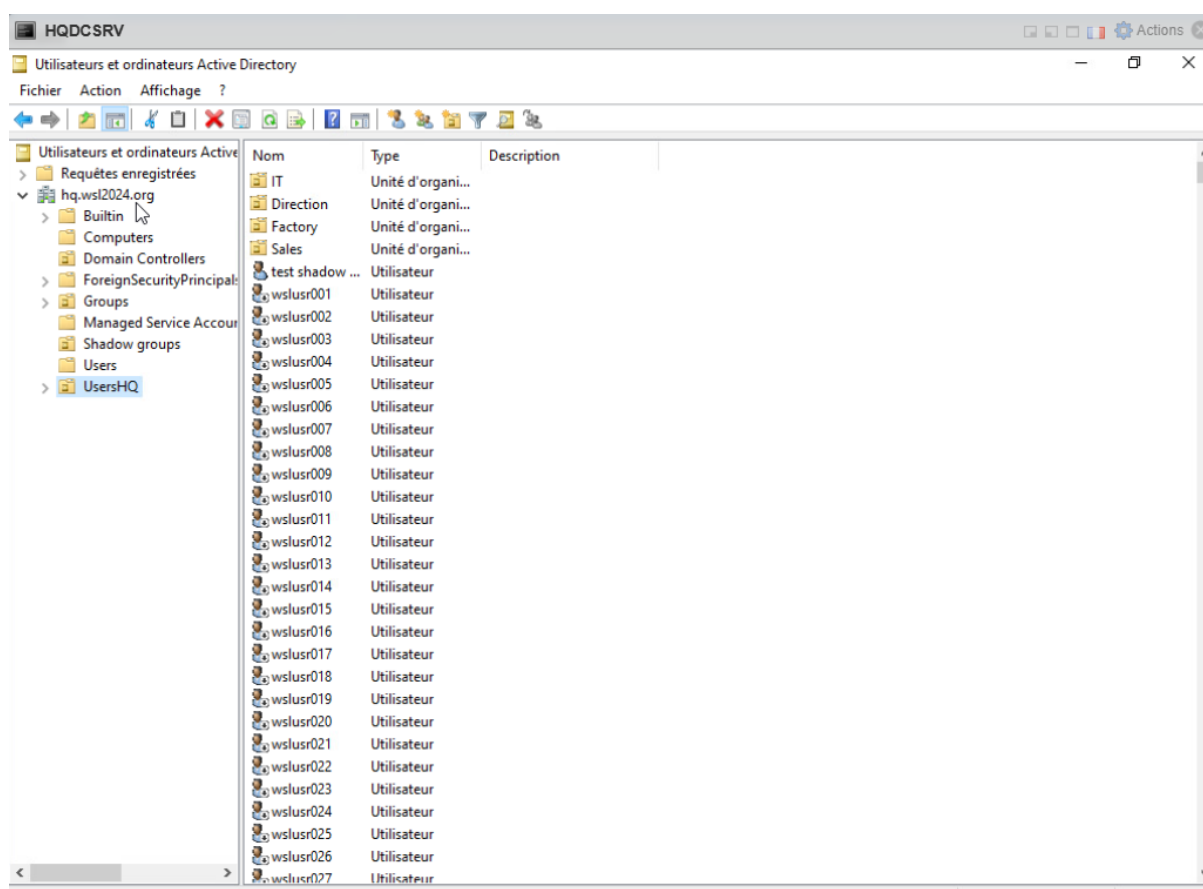


Figure 22 : 1000 Users

## 4.2. SUPSRV

SUPSRV joue le rôle de serveur de gestion du réseau. Lors de la mise en place de ce serveur, nous avons été confrontés à divers choix quant à l'outil réseau à utiliser. Notre choix s'est finalement porté sur Zabbix, principalement parce que nos collègues du BTS SIO avaient une expérience significative avec cet outil, offrant ainsi une source d'assistance en cas de problèmes éventuels.

Il a d'abord fallu installer Zabbix. Pour ce faire, le site officiel de Zabbix nous guide assez bien en précisant notre système d'exploitation et la version de Zabbix que l'on souhaite. Ensuite, cela génère des commandes que l'on a simplement à copier-coller à partir du lien du site : [Download and install Zabbix](#)

Choose your platform

ZABBIX VERSION	OS DISTRIBUTION	OS VERSION	ZABBIX COMPONENT	DATABASE	WEB SERVER
6.4	Alma Linux	12 (Bookworm)	Server, Frontend, Agent	MySQL	Apache
6.0 LTS	CentOS	11 (Bullseye)	Proxy	PostgreSQL	Nginx
5.0 LTS	Debian	10 (Buster)	Agent		
7.0 PRE-RELEASE	OpenSUSE Leap	9 (Stretch)	Agent 2		
	Oracle Linux		Java Gateway		
	Raspberry Pi OS		Web Service		
	Red Hat Enterprise Linux				
	Rocky Linux				
	SUSE Linux Enterprise Server				
	Ubuntu				
	Ubuntu (arm64)				

Figure 23 : Installation ZABBIX

Une fois les commandes exécutées et Zabbix installé et configuré, on peut accéder à l'interface web du serveur en utilisant localhost/zabbix comme URL. On arrive ensuite sur cette page où il faut faire une brève configuration.

Figure 24 : C

Il a ensuite fallu aller sur ce site pour connaître l'utilisateur Zabbix par défaut : 1 Login and configuring user (zabbix.com) Qui a pour user : Admin et password : zabbix



**ZABBIX**

Nom d'utilisateur

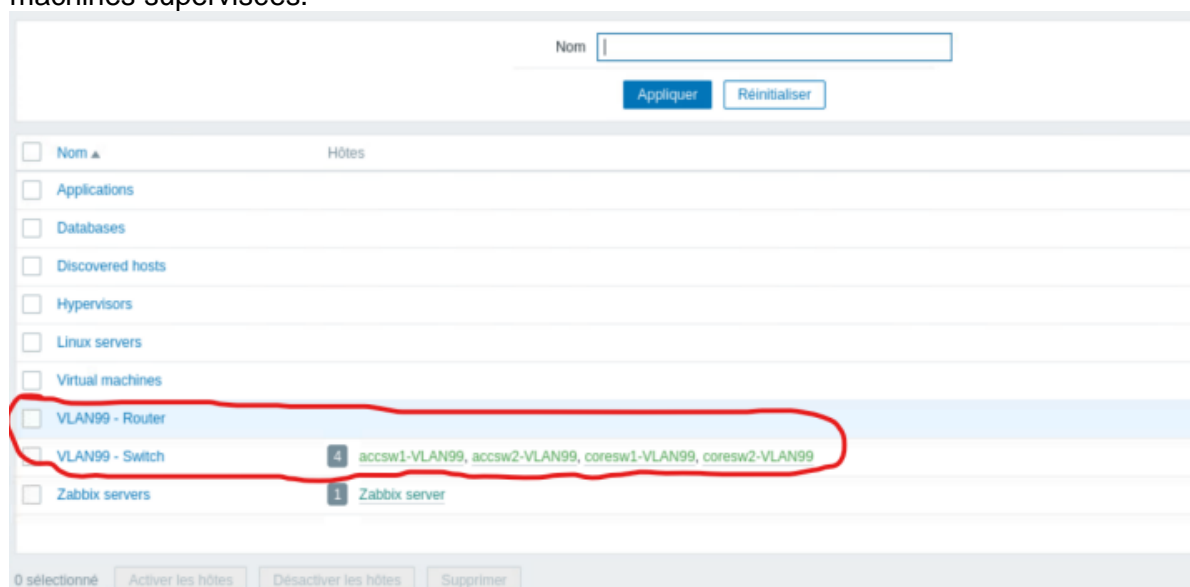
Mot de passe

☒ Me rappeler toutes les 30 jours

**S'enregistrer**

[Aide](#) • [Support](#)

Sur la page web de management il a ensuite fallu ajouter des groupes pour les différentes machines supervisées.



Nom

**Appliquer** **Réinitialiser**

<input type="checkbox"/> Nom ▲	Hôtes
<input type="checkbox"/> Applications	
<input type="checkbox"/> Databases	
<input type="checkbox"/> Discovered hosts	
<input type="checkbox"/> Hypervisors	
<input type="checkbox"/> Linux servers	
<input type="checkbox"/> Virtual machines	
<input checked="" type="checkbox"/> VLAN99 - Router	
<input checked="" type="checkbox"/> VLAN99 - Switch	4 accsw1-VLAN99, accsw2-VLAN99, coresw1-VLAN99, coresw2-VLAN99
<input type="checkbox"/> Zabbix servers	1 Zabbix server

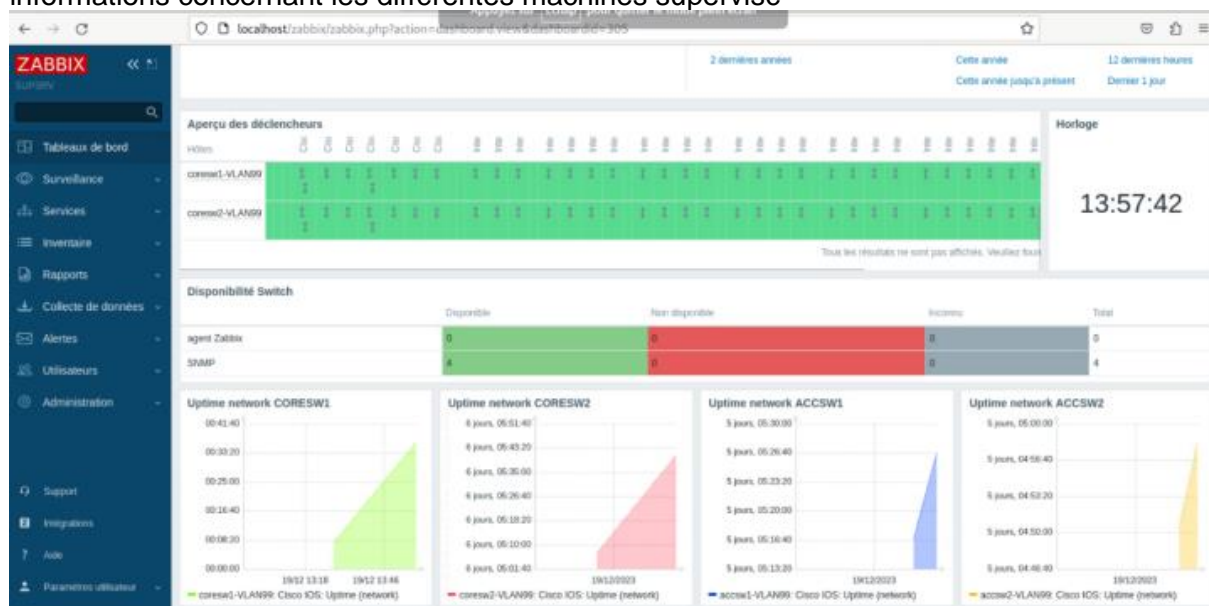
0 sélectionné **Activer les hôtes** **Désactiver les hôtes** **Supprimer**

Et nous avons ajouté des hôtes avec les adresses IP correspondantes des machines. Pour que le SNMP soit actif il a fallu entrer ces commandes sur les switches : CORESW1# configure

```
terminal CORESW1(config)# snmp-server community supervision ro CORESW1(config)#
snmp-server enable traps CORESW1(config)# snmp-server host 10.3.99.3 supervision
```

Nom	Éléments	Déclencheurs	Graphiques	Découverte	Web	Interface	Proxy	Modèles	État	Disponibilité
accsw1-VLAN99	176	83	19	1	10.3.99.241:161	Cisco IOS by SNMP	Active	SNMP		
accsw2-VLAN99	176	83	19	1	10.3.99.242:161	Cisco IOS by SNMP	Active	SNMP		
coresw1-VLAN99	178	85	19	1	10.3.99.252:161	Cisco IOS by SNMP	Active	SNMP		
coresw2-VLAN99	169	81	18	1	10.3.99.253:161	Cisco IOS by SNMP	Active	SNMP		
Zabbix server	128	69	24	5	127.0.0.1:10050	Linux by Zabbix agent, Zabbix server health	Active	ZBX		

Il est ensuite possible de configurer à sa guise le tableau de bord pour y mettre des informations concernant les différentes machines supervisé



### 4.3. HQINFRASRV

Le serveur DNS de HQINFRASRV fonctionne dans le cadre du domaine wsl2024.org, assurant lui-même le transfert des requêtes vers le domaine worldskills.org. Ce serveur intègre également un service DHCP dédié aux utilisateurs du VLAN 20, lesquels sont membres de l'Active Directory de HQDCSRV. De plus, HQINFRASRV abrite un serveur Samba qui utilise un espace de stockage formé par deux disques durs de 5 Go configurés en LVM.

#### 4.3.1. Configuration des disques durs

Pour configurer le stockage sur notre serveur Debian 10, nous avons suivi les étapes suivantes

Ajouter des deux dur 5Go sur la vm :



Informations générales	
Mise en réseau	
Nom d'hôte	HQINFRASRV
Adresses IP	1. 10.3.10.3
VMware Tools	VMware Tools n'est pas géré par vSphere
Stockage	3 disques
Notes	
Modifier les notes	

Configuration matérielle	
CPU	1 vCPUs
Mémoire	2 Go
Disque dur 1	80 Go
Disque dur 2	5 Go
Disque dur 3	5 Go
Contrôleur USB	USB 2.0
Adaptateur réseau 1	No (Connecté)

1. Tout d'abord, nous avons installé LVM en utilisant la commande suivante : `sudo apt-get install lvm2`

2. Ensuite, nous avons utilisé la commande `fdisk` pour créer des partitions sur les deux disques de 5 Go que nous avons ajoutés. Par exemple, pour créer une partition sur le premier disque, nous avons exécuté la commande suivante :

```
sudo fdisk /dev/sdb
```

3. Après avoir créé les partitions, nous avons utilisé la commande `pvcreate` pour créer des volumes physiques LVM sur les partitions. Par exemple, pour créer un volume physique sur la première partition du premier disque, nous avons exécuté la commande suivante :

```
sudo pvcreate /dev/sdb1
```

4. Ensuite, nous avons utilisé la commande `vgcreate` pour créer un groupe de volumes LVM à partir des volumes physiques. Par exemple, pour créer un groupe de volumes nommé "vgstorage" à partir des deux volumes physiques, nous avons exécuté la commande suivante :

```
sudo vgcreate vgstorage /dev/sdb1 /dev/sdc1
```

5. Après avoir créé le groupe de volumes, nous avons utilisé la commande `lvcreate` pour créer des volumes logiques LVM. Par exemple, pour créer un volume logique nommé "lvdatastorage" de 2 Go, nous avons exécuté la commande suivante :

```
sudo lvcreate -L 2G -n lvdatastorage vgstorage
```

6. Nous avons répété l'étape précédente pour créer un autre volume logique nommé "lvscsi" de 2 Go

7. Ensuite, nous avons utilisé la commande `mkfs.ext4` pour formater le volume logique "lvdatastorage" avec le système de fichiers ext4. Par exemple, pour formater le volume logique "lvdatastorage", nous avons exécuté la commande suivante :

```
sudo mkfs.ext4 /dev/vgstorage/lvdatastorage
```

8. Enfin, nous avons utilisé la commande `mount` pour monter le volume logique "lvdatastorage" sur le point de montage "/srv/datastorage". Par exemple, pour monter le volume logique "lvdatastorage", nous avons exécuté la commande suivante :

```
sudo mount /dev/vgstorage/lvdatastorage /srv/datastorage.
```

#### 4.3.2. Configuration Samba

Étape 1: Nous avons installé Samba. `sudo apt-get update sudo apt-get install samba`

Étape 2: Nous avons édité le fichier de configuration Samba. `sudo nano /etc/samba/smb.conf`

Étape 3: Nous avons ajouté les configurations suivantes à la fin du fichier.

[Public]

```
path = /srv/datastorage/shares/public
read only = yes
guest ok = yes
browseable = yes
writable = no
```

[Private]

```
path = /srv/datastorage/shares/private
valid users = tom Emma
browseable = no
writable = yes
create mask = 0644
directory mask = 0755
hide dot files = yes
read list = jean
veto files = /*.exe/*.zip/
```

Figure 24 : Configuration ZABBIX

Étape 4: Nous avons créé les répertoires et ajusté les autorisations.

```
sudo mkdir -p /srv/datastorage/shares/public
sudo mkdir -p /srv/datastorage/shares/private
sudo chown -R :sambashare /srv/datastorage
sudo chmod -R 775 /srv/datastorage
```

#### 4.3.3. Configuration DHCP & DNS

Dans le fichier de configuration `dhcpd.conf`, nous avons minutieusement défini plusieurs éléments cruciaux pour assurer le bon fonctionnement du service DHCP. Tout d'abord, nous avons spécifié le nom de domaine, permettant ainsi aux clients de recevoir cette information essentielle lors de l'attribution d'une adresse IP.

De plus, nous avons configuré le serveur DNS dans le fichier `dhcpd.conf`. Cette information est cruciale car elle permet aux clients DHCP d'obtenir les adresses IP des serveurs DNS nécessaires pour la résolution des noms de domaine.

La passerelle par défaut a également été précisément configurée dans le fichier. Ce paramètre indique aux clients DHCP la passerelle réseau par laquelle ils peuvent accéder à des réseaux externes et à Internet.

En outre, nous avons défini avec précision l'intervalle d'adresses dans le fichier dhcpd.conf. Cet intervalle détermine la plage d'adresses IP que le serveur DHCP peut attribuer aux clients. Une configuration appropriée de cette plage garantit une gestion efficace des adresses IP au sein du réseau, évitant ainsi les conflits d'adresses et assurant une distribution ordonnée des adresses aux périphériques du réseau.

```
GNU nano 5.4 /etc/dhcp/dhcpd.conf
# Configuration globale du serveur DHCP
option domain-name "hq.wsl2024.org";
option domain-name-servers hqdcsvr.hq.wsl2024.org;
default-lease-time 600;
max-lease-time 7200;

# Sous-réseau et plage d'adresses à attribuer
subnet 10.3.10.0 netmask 255.255.255.192 {
    range 10.3.10.10 10.3.10.50;
    option routers 10.3.10.62;

    default-lease-time 600;
    max-lease-time 7200;
    interface ens192;
}
```

La configuration du nom DNS dans le fichier dhcpd.conf s'assure que les clients obtiennent des informations cruciales pour la résolution des noms, y compris l'adresse IP du serveur DNS. Cette information est essentielle pour que les clients puissent effectuer des requêtes DNS avec succès et accéder aux services basés sur les noms de domaine, tels que la navigation sur Internet et l'accès aux ressources réseau.

```
$TTL      604800
@         IN      SOA      hqinfrasrv.wsl2024.org. admin.wsl2024.org. (
                                1              ; Serial
                                604800         ; Refresh
                                86400          ; Retry
                                2419200        ; Expire
                                604800 )       ; Negative Cache TTL

@         IN      NS       hqinfrasrv.wsl2024.org.

hqdcsvr.hq      IN      A       10.3.10.1
hqinfrasrv      IN      A       10.3.10.3
hqmailsrv       IN      A       10.3.10.2

hqwebsrv.hq     IN      CNAME   hqfwsrv.wsl2024.org.
www             IN      CNAME   hqfwsrv.wsl2024.org.
webmail         IN      CNAME   hqmailsrv.wsl2024.org.
pki.hq          IN      CNAME   hqdcsvr.hq.wsl2024.org.

coresw1         IN      A       10.3.99.252
coresw2         IN      A       10.3.99.253
edge1           IN      A       10.3.254.1
edge2           IN      A       10.3.254.17
wanrtr          IN      A       10.3.254.6
```

nslookup pour vérifier bon fonctionnement :

```

root@HQINFRA SRV:/home/hqinfrasrv# nslookup hqdcsvr.hq.wsl2024.org
Server:      10.3.10.3
Address:     10.3.10.3#53

Name:   hqdcsvr.hq.wsl2024.org
Address: 10.3.10.1

root@HQINFRA SRV:/home/hqinfrasrv# nslookup hqinfrasrv.wsl2024.org
Server:      10.3.10.3
Address:     10.3.10.3#53

Name:   hqinfrasrv.wsl2024.org
Address: 10.3.10.3

root@HQINFRA SRV:/home/hqinfrasrv# nslookup pki.hq.wsl2024.org
Server:      10.3.10.3
Address:     10.3.10.3#53

pki.hq.wsl2024.org      canonical name = hqdcsvr.hq.wsl2024.org.
Name:   hqdcsvr.hq.wsl2024.org
Address: 10.3.10.1

root@HQINFRA SRV:/home/hqinfrasrv# █

```

## 4.4. HQFWSRV

Cette étape vise à sécuriser les communications entrantes et à exposer de manière contrôlée les ressources internes sur Internet via la configuration du pare-feu à l'aide de **nftables**. L'objectif principal est de garantir un accès sécurisé aux services tout en limitant l'exposition des ports non utilisés.

**nftables** est un sous-système d'exploitation du noyau Linux, il permet de filter les paquets. Il introduit une structure de données unifiée appelée "tables". Ces tables contiennent des ensembles de règles pour gérer le trafic réseau. Chaque table peut comporter différentes "chains" (chaînes) telles que `input`, `output`, `forward`, `prerouting` et `postrouting`.

L'objectif central est de mettre en place des règles spécifiques au sein du réseau **VLAN 30** et **10** pour filtrer le trafic entrant de manière sélective. Cela implique la redirection des requêtes web (HTTP/HTTPS) vers le serveur web (HQWEBSVR) et la mise en place de règles pour diriger le trafic des services MS RDS (Remote Desktop Services) et AD (Active Directory) vers le serveur HQWEBSVR. De plus, la fermeture des ports inutilisés vise à renforcer la sécurité du réseau en réduisant les points d'entrée potentiels pour des attaques externes.

Les exigences précises pour ce projet incluent la restriction du trafic entrant sur le VLAN 30 pour permettre uniquement les requêtes web et les services MS RDS vers le serveur HQWEBSVR. De plus, la préservation du VLAN 10 pour l'authentification sur Active Directory nécessite une attention particulière, tout en tenant compte des besoins d'ouverture de ports supplémentaires pour certains services spécifiques tels que IPSec VPN, SSH, etc.

En combinant ces exigences, la configuration du pare-feu doit établir une balance entre la sécurité accrue et la disponibilité des services essentiels, tout en garantissant une vérification rigoureuse de chaque service après chaque modification du pare-feu.

### 4.4.1. Configuration des règles de filtrage

Dans le cadre de la sécurisation des communications entrantes pour le VLAN 30 et VLAN 10, la configuration du pare-feu à l'aide de **nftables** a été effectuée conformément aux exigences

spécifiques du projet. Cette configuration a été réalisée en mettant en place des règles de filtrage pour diriger le trafic entrant de manière sélective.

(HTTP/HTTPS) :

Les règles ont été déployées pour rediriger de manière sécurisée les requêtes HTTP et HTTPS vers le serveur web HQWEBSVR. Les exemples de règles nftables pour les requêtes HTTP et HTTPS :

```
table ip filter {
    chain input {
        # Règles spécifiques pour les ports utilisés
        # Ajoutez la règle pour bloquer les paquets ICMP Echo
        # Autoriser le trafic LDAP, IPsec VPN
        tcp dport { 80, 443 } ssh, 500, 4500 } accept
        ip daddr 10.3.10.1 tcp dport { 389, 636 } accept
        icmp type echo-reply drop
        drop
    }

    chain forward {
        # Redirection du trafic HTTP/HTTPS et MS RDS vers HQWEBSVR
        tcp dport { 80, 443, 3389 } accept
        tcp dport 53 accept
        # Fermeture des ports inutilisés
        drop
    }

    chain output {
        # Règles pour le trafic sortant
    }
}
```

Figure 34 : Règles de filtrage port HTTP/HTTPS

MS RDS (Remote Desktop Services) :

Le trafic des services MS RDS a été configuré pour être redirigé vers le serveur HQWEBSVR, permettant un accès spécifique à ces services. Un exemple de règle nftable pour les services MS RDS :

```
chain forward {
    # Redirection du trafic HTTP/HTTPS et MS RDS vers HQWEBSVR
    tcp dport { 80, 443, 3389 } accept
    tcp dport 53 accept
    # Fermeture des ports inutilisés
    drop
}
```

Figure 35 : Règles de filtrage port MS RDS

AD (Active Directory) :

La configuration du pare-feu avec nftables pour le VLAN 30 a été étendue pour inclure la redirection du trafic LDAP vers le serveur web HQWEBSVR, afin de permettre une authentification sécurisée sur le site web.

```
table ip filter {  
    chain input {  
        # Règles spécifiques pour les ports utilisés  
        # Ajoutez la règle pour bloquer les paquets ICMP Echo  
        # Autoriser le trafic LDAP, IPsec VPN  
        tcp dport { 80, 443, ssh, 500, 4500 } accept  
        ip daddr 10.3.10.1 tcp dport { 389, 636 } accept  
        icmp type echo-reply drop  
        drop  
    }  
}
```

Figure 36 : Règles de filtrage port LDAP

Fermeture des Ports Inutilisés :

Afin de renforcer la sécurité du réseau, tous les autres ports inutilisés ont été fermés. Les règles nftables pour la fermeture des ports inutilisés ont été appliquées pour réduire les points d'entrée potentiels pour des attaques externes. Un exemple de règle nftable pour la fermeture des ports inutilisés :



```

table ip filter {
    chain input {
        # Règles spécifiques pour les ports utilisés
        # Ajoutez la règle pour bloquer les paquets ICMP Echo
        # Autoriser le trafic LDAP, IPsec VPN
        tcp dport { 80, 443, ssh, 500, 4500 } accept
        ip daddr 10.3.10.1 tcp dport { 389, 636 } accept
        icmp type echo-reply drop ←
        drop ←
    }

    chain forward {
        # Redirection du trafic HTTP/HTTPS et MS RDS vers HQWEBSVR
        tcp dport { 80, 443, 3389 } accept
        tcp dport 53 accept
        # Fermeture des ports inutilisés
        drop ←
    }

    chain output {
        # Règles pour le trafic sortant
    }
}

```

Figure 37 : Règles de filtrage fermetures des ports

#### Analyse des Ports sur HQWEBSVR :

Pour évaluer la configuration de sécurité actuelle du serveur HQWEBSVR, un test de sécurité a été effectué à l'aide de la commande nmap -A -sS vers l'adresse IP du serveur.

```

root@HQINFRASRV:/home/hqinfrsrv# nmap -A -sS 10.3.10.4
Starting Nmap 7.80 ( https://nmap.org ) at 2023-12-21 12:47 CET
Nmap scan report for 10.3.10.4
Host is up (0.00048s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
80/tcp    filtered http
443/tcp    filtered https
MAC Address: 00:0C:29:4F:96:28 (VMware)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.80%E=4%D=12/21%OT=22%CT=1%CU=41575%PV=Y%DS=1%DC=D%G=Y%M=000C29%
OS:TM=658425F9%P=x86_64-pc-linux-gnu)SEQ(SP=FF%GCD=1%ISR=10E%TI=Z%CI=Z%II=I
OS:%TS=A)OPS(O1=M5B4ST11NW7%O2=M5B4ST11NW7%O3=M5B4NNT11NW7%O4=M5B4ST11NW7%O
OS:5=M5B4ST11NW7%O6=M5B4ST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6
OS:=FE88)ECN(R=Y%DF=Y%T=40%W=FAF0%M5B4NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=0
OS:%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=
OS:0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0
OS:S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(
OS:R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=
OS:N%T=40%CD=S)

Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT ADDRESS
1 0.48 ms 10.3.10.4

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.41 seconds
root@HQINFRASRV:/home/hqinfrsrv#

```

Figure 38 : Nmap sur HQFWSRV

Le test a révélé la présence des ports utilisés conformément à la configuration établie pour le VLAN 30. Cependant, une vulnérabilité potentielle a été identifiée concernant la révélation de la version de SSH, ou bien l'OS. Représentant une faille de sécurité, permettant à un attaquant de cibler des vulnérabilités connues pour cette version spécifique.

Pour contrer cette faille de sécurité, l'utilisation de règles spécifiques dans nftables peut être mise en place pour masquer la version de SSH, renforçant ainsi la sécurité du serveur. Une règle peut être ajoutée pour éviter la divulgation de la version de SSH dans les réponses aux requêtes de connexion.

Nous devons rajouter ceci dans « chain output » :

#### tcp dport 22 modulate stateful

Cela modifie les réponses sortantes de la connexion SSH pour obscurcir la version du service.

#### 4.4.2. Règles de NAT(Network Address Translation)

Les règles de NAT (Network Address Translation) ont été créées afin de répondre à des besoins spécifiques de redirection et de modification des adresses IP. Ces règles, réparties entre les chaînes "prerouting" et "postrouting", visent à diriger sélectivement le trafic entrant et à modifier les adresses source du trafic sortant en fonction de critères prédéfinis. Et tout ceci afin d'accéder à HQWEBSRV qui se situe dans la DMZ :

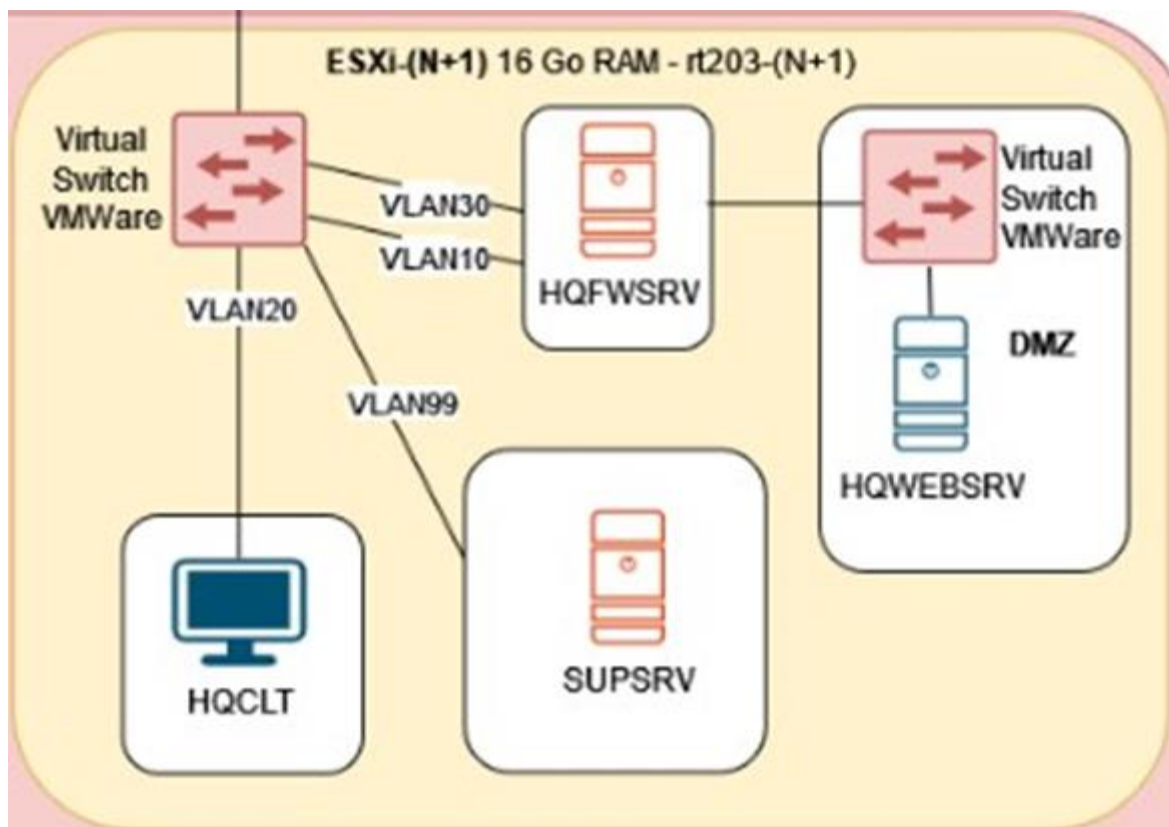


Figure 39 : Schéma infrastructure site HQ côté HQFWSRV

### Règles de "prerouting" :

Les règles appliquées à la chaîne "prerouting" dirigent sélectivement le trafic entrant des interfaces "ens160" et "ens192", respectivement les VLAN 10 et le VLAN 30, ciblant les ports 80 (HTTP) et 443 (HTTPS), vers l'adresse IP 10.3.30.1 via la redirection DNAT. Cette configuration centralise efficacement le trafic Web entrant provenant de ces interfaces vers une destination unique.

```

GNU nano 5.4                                vlan30 ruleset.nft
table ip nat {
    chain prerouting {
        type nat hook prerouting priority 0;
        iifname "ens160" tcp dport {80, 443} dnat to 10.3.30.1
        iifname "ens192" tcp dport {80, 443} dnat to 10.3.30.1
    }
    chain postrouting {
        type nat hook postrouting priority 0;
        ip saddr 10.3.30.1 ip daddr 10.3.10.1 snat to 10.3.10.4
        ip saddr 10.3.30.1 snat to 217.3.160.1
    }
}

```

Figure 40 : Règles NAT

### Règles de "postrouting" :

La chaîne "postrouting" modifie les adresses source des paquets sortants, impliquant une logique spécifique pour le traitement des adresses sources.

```

GNU nano 5.4                                vlan30 ruleset.nft
table ip nat {
    chain prerouting {
        type nat hook prerouting priority 0;
        iifname "ens160" tcp dport {80, 443} dnat to 10.3.30.1
        iifname "ens192" tcp dport {80, 443} dnat to 10.3.30.1
    }
    chain postrouting {
        type nat hook postrouting priority 0;
        ip saddr 10.3.30.1 ip daddr 10.3.10.1 snat to 10.3.10.4
        ip saddr 10.3.30.1 snat to 217.3.160.1
    }
}

```




Figure 41 : Règles NAT

La première partie de la règle spécifie une modification de l'adresse source de 10.3.30.1 à 10.3.10.4 pour les paquets sortants ayant pour destination l'adresse IP 10.3.10.1.

```

GNU nano 5.4                                vlan30 ruleset.nft
table ip nat {
    chain prerouting {
        type nat hook prerouting priority 0;
        iifname "ens160" tcp dport {80, 443} dnat to 10.3.30.1
        iifname "ens192" tcp dport {80, 443} dnat to 10.3.30.1
    }
    chain postrouting {
        type nat hook postrouting priority 0;
        ip saddr 10.3.30.1 ip daddr 10.3.10.1 snat to 10.3.10.4
        ip saddr 10.3.30.1 snat to 217.3.160.1
    }
}

```




Figure 42 : Règles NAT

La seconde partie de la règle change simplement l'adresse source 10.3.30.1 pour qu'elle devienne 217.3.160.1 pour tous les paquets sortants, quelle que soit leur destination.

```

GNU nano 5.4                                vlan30 ruleset.nft
table ip nat {
    chain prerouting {
        type nat hook prerouting priority 0;
        iifname "ens160" tcp dport {80, 443} dnat to 10.3.30.1
        iifname "ens192" tcp dport {80, 443} dnat to 10.3.30.1
    }
    chain postrouting {
        type nat hook postrouting priority 0;
        ip saddr 10.3.30.1 ip daddr 10.3.10.1 snat to 10.3.10.4
        ip saddr 10.3.30.1 snat to 217.3.160.1
    }
}

```




Figure 43 : Règles NAT

## Problème d'Accès à la DMZ

Malgré la configuration de règles de redirection et de NAT, l'accès à la DMZ depuis INETCLT demeure problématique. L'exécution d'un traceroute vers l'adresse IP du serveur HQWEBSRV révèle que les paquets suivent un cheminement via le routeur, puis le switch, mais ne parviennent pas à atteindre la destination finale, ces deux derniers se renvoient continuellement les paquets, ceci est dû à un éventuel blocage ou à un problème de routage spécifique. (bien évidemment lors du test le protocole ICMP était autorisé).

## **Conclusion**

Les règles de "prerouting" et "postrouting" identifiées dans la configuration de NAT démontrent une utilisation spécifique et ciblée pour rediriger le trafic entrant et modifier les adresses sources du trafic sortant. Cette approche offre une flexibilité dans la gestion et la sécurisation du flux de données, répondant potentiellement à des besoins variés au sein du réseau. Ces règles offrent une perspective sur les méthodes utilisées pour gérer et sécuriser les communications au niveau du pare-feu, tout en adaptant dynamiquement les adresses source en fonction de critères prédéfinis.

## **4.5. HQWEBSRV**

Ce serveur, configuré avec le logiciel IIS (Internet Information Services), offre un accès à diverses ressources, via des sites HTTP et HTTPS, ainsi qu'à des applications Microsoft Office via un service de bureau à distance (MS-RDS). Cette introduction à la configuration du serveur HQWEBSRV explore l'usage d'IIS pour la création de sites sécurisés, la gestion des protocoles HTTP et HTTPS, l'installation de certificats SSL et l'authentification via le protocole LDAP, dévoilant ainsi une infrastructure garantissant un accès flexible et sécurisé aux ressources de l'entreprise.

### **4.5.1. Serveur IIS**

La création du site web se fera à travers l'interface d'IIS, permettant de spécifier les détails du site tel que le nom, le répertoire racine, les permissions d'accès.



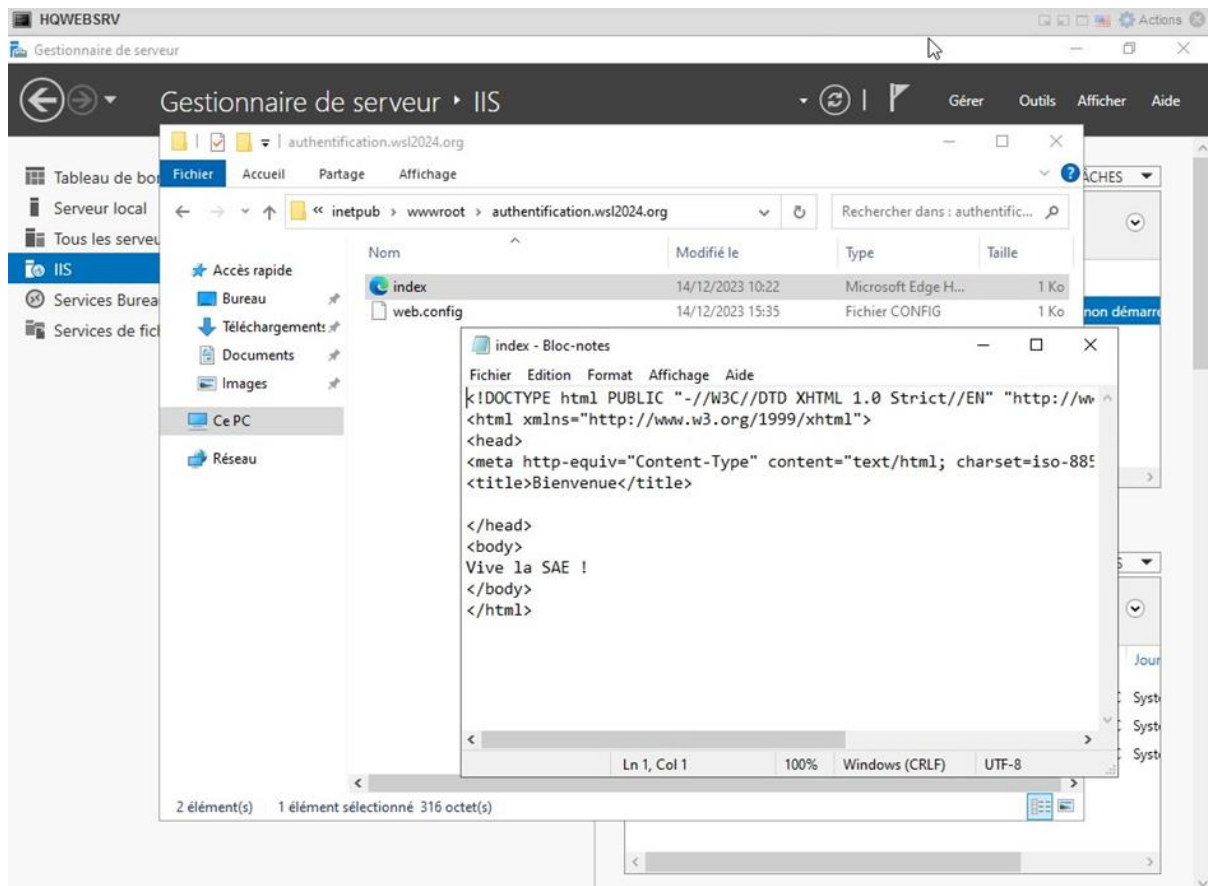


Figure 44 : Création du site

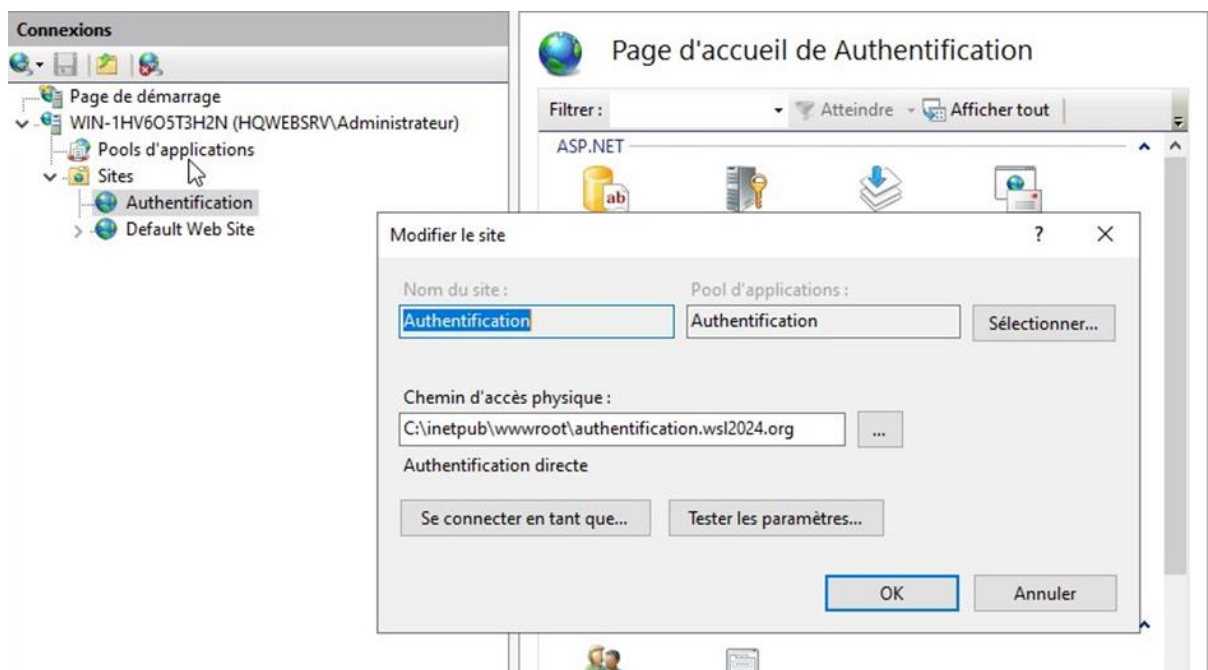


Figure 45 : Chemin d'accès du site web

Liaisons HTTP et HTTPS :



IIS sera configuré pour établir deux liaisons distinctes : une liaison en HTTP et une autre en HTTPS. Cela permettra d'assurer que le site est accessible à la fois en HTTP (automatiquement redirigé vers HTTPS) et en HTTPS, garantissant ainsi une connexion sécurisée.

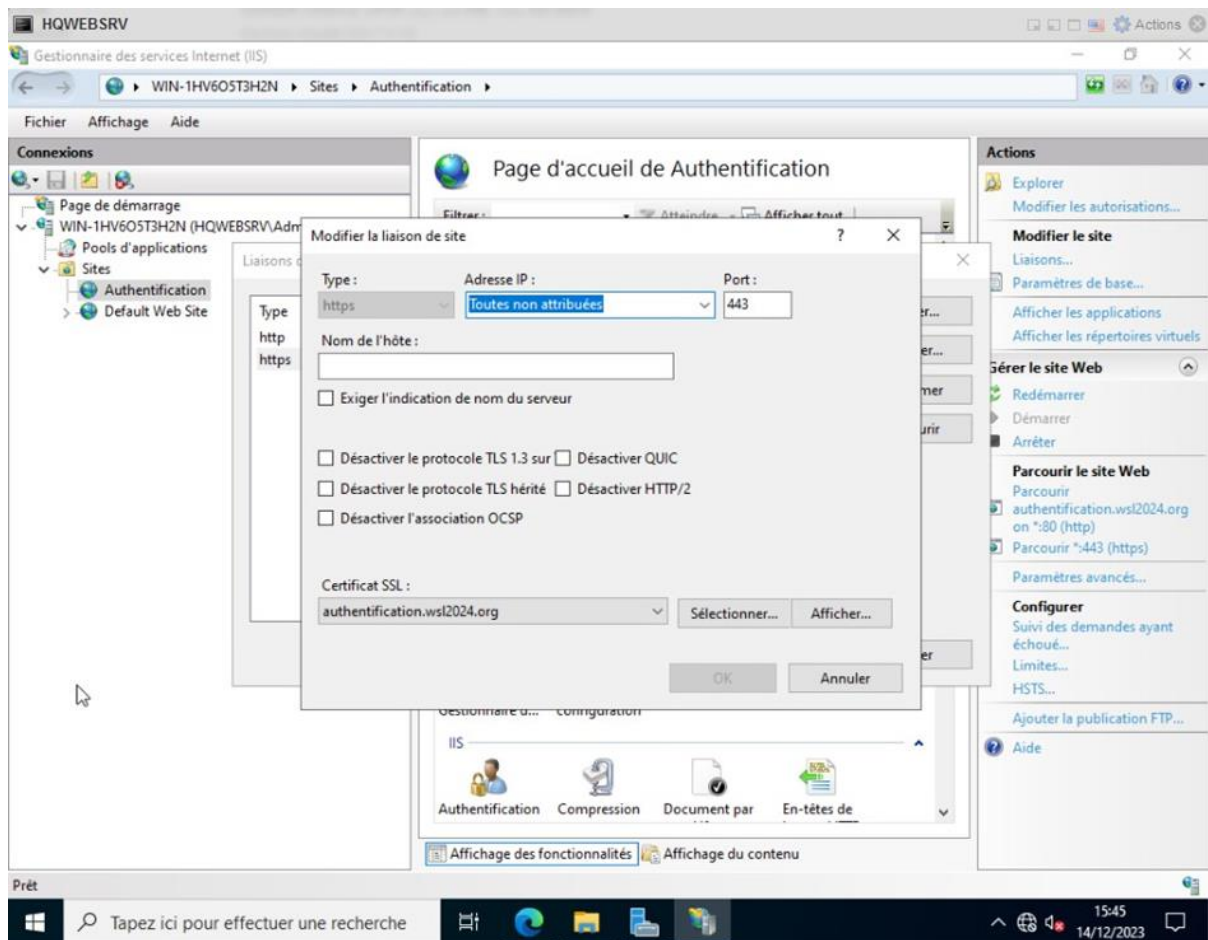


Figure 46 : Mise en place de la liaison HTTPS

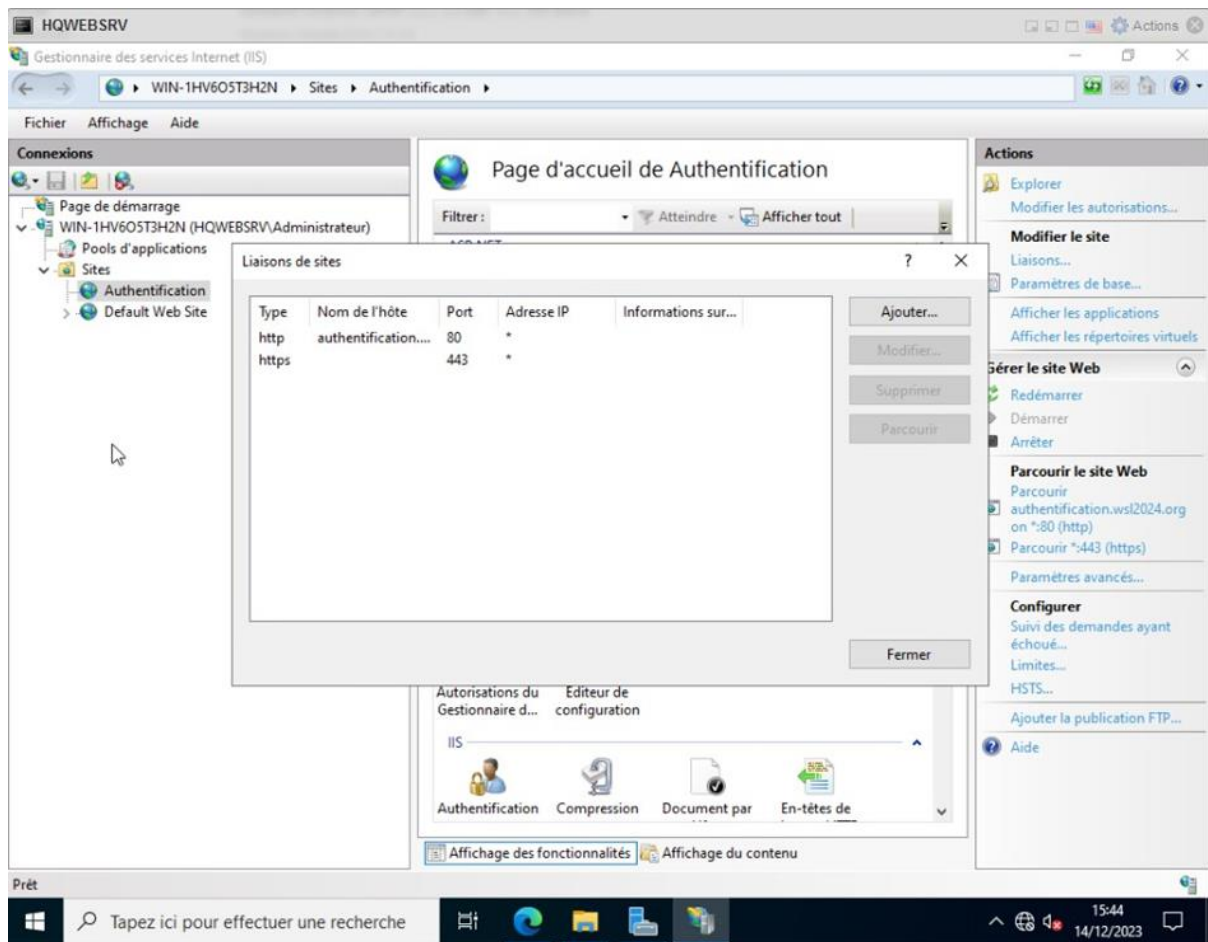


Figure 47 : Visualisation des liaison HTTP et HTTPS

#### Certificat SSL :

Un certificat SSL sera installé sur le site HTTPS via IIS. Ce certificat, émis par une autorité de certification, permettra le chiffrement des données transitant entre le serveur et les navigateurs des utilisateurs, garantissant ainsi une connexion sécurisée.

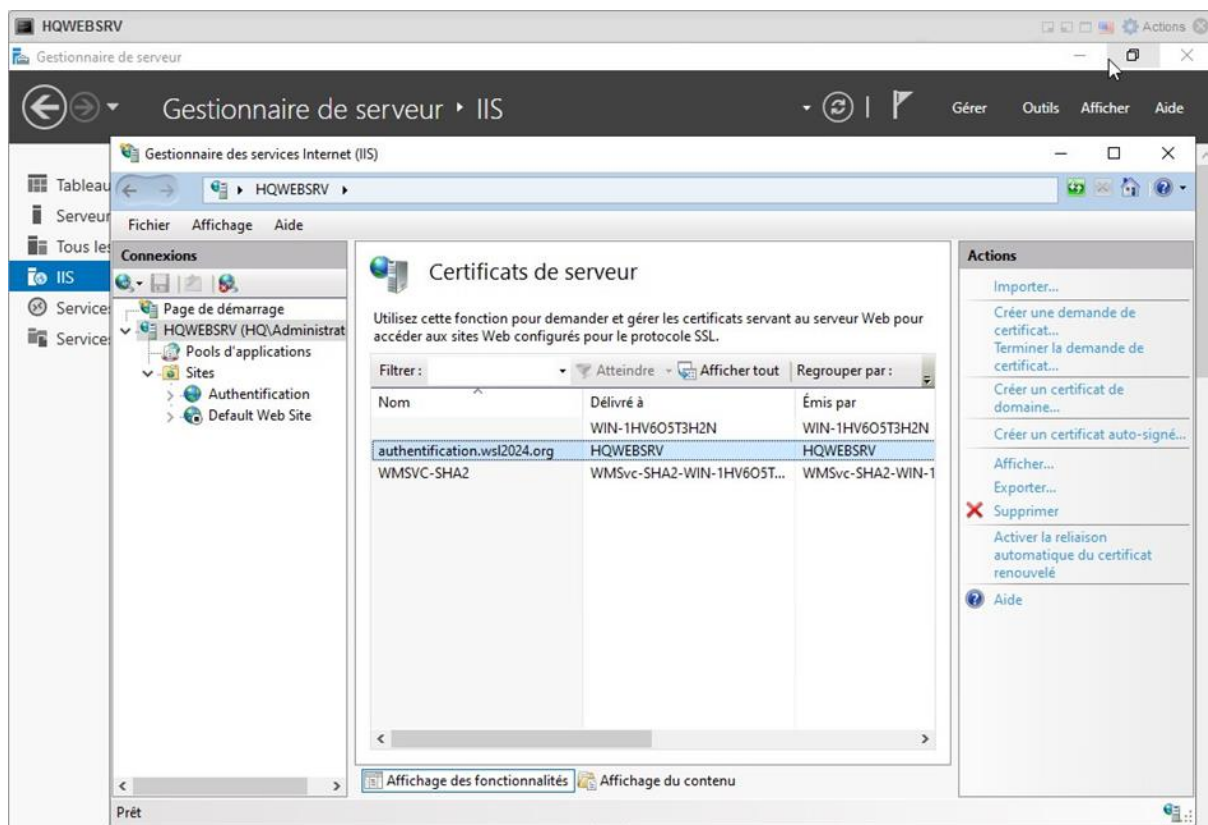


Figure 48 : Les différents certificats

#### Authentification avec LDAP :

L'IIS sera configuré pour autoriser l'authentification via LDAP. Cela permettra de valider les identités des utilisateurs à travers le protocole LDAP, habituellement en lien avec un serveur d'annuaire comme Active Directory, pour un accès sécurisé au site.

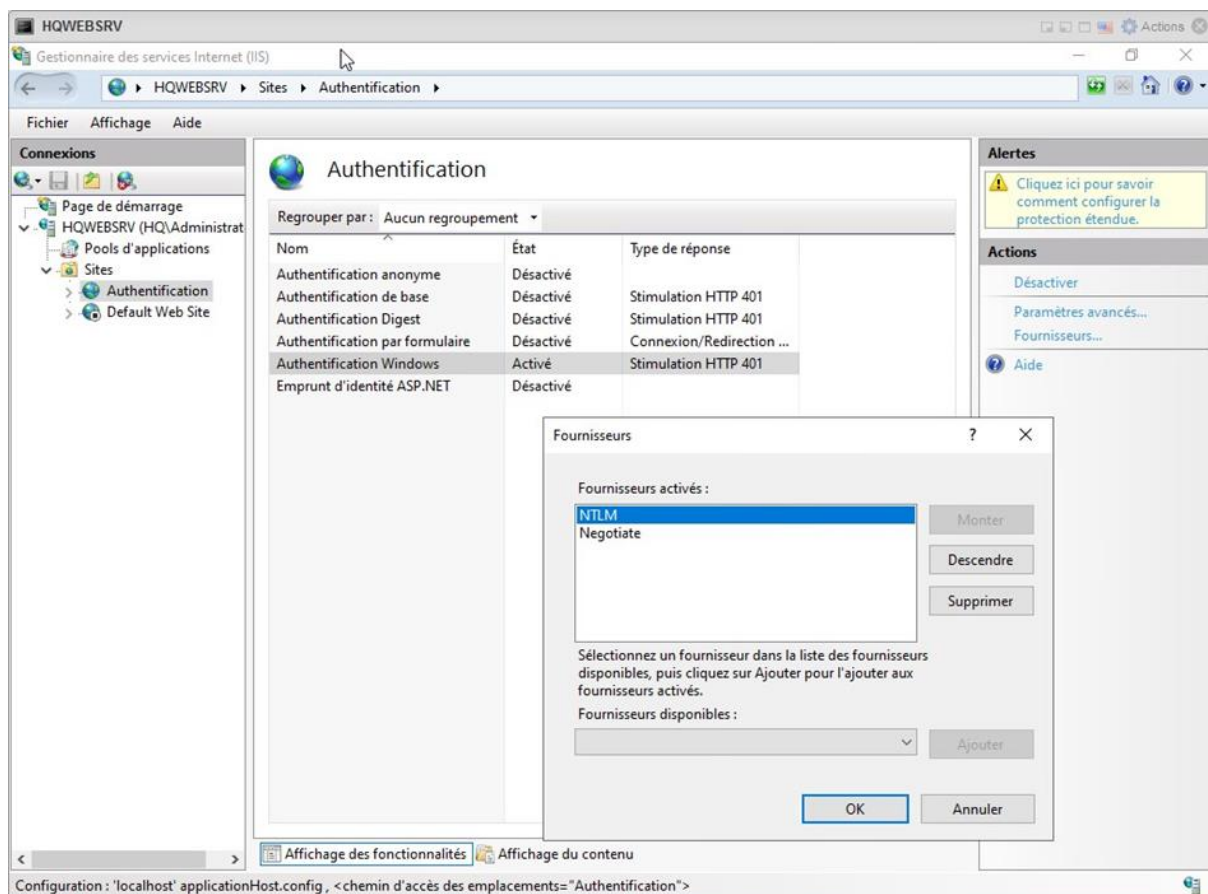


Figure 49 : Configuration de l'authentification avec LDAP

Test de Connexion HTTP :

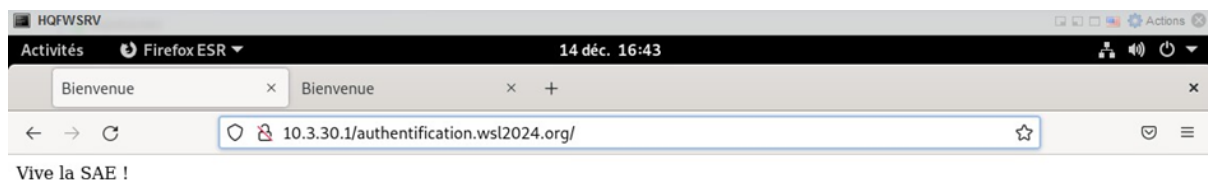


Figure 50 : Connexion sur le site web en HTTP

Test de Connexion HTTPS : La connexion via le protocole HTTPS vers le site a été établie avec succès, permettant un accès sécurisé aux ressources du serveur, sans qu'une authentification supplémentaire ne soit requise pour le moment.

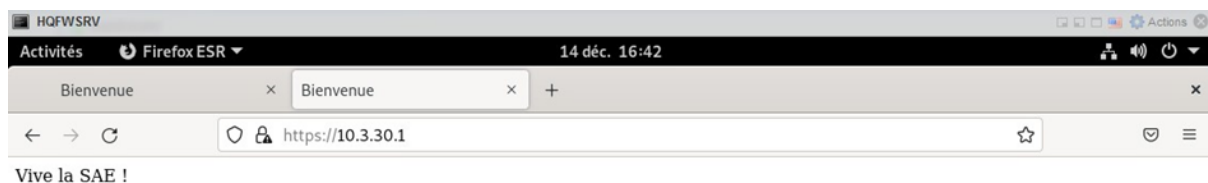


Figure 51 : Connexion sur le site web en HTTPS

Test de Connexion HTTPS avec Authentification LDAP : Suite à la mise en place de l'authentification via LDAP pour <https://10.3.30.1/authentification.wsl2024.org>, l'accès à ce site a nécessité une identification via Active Directory.

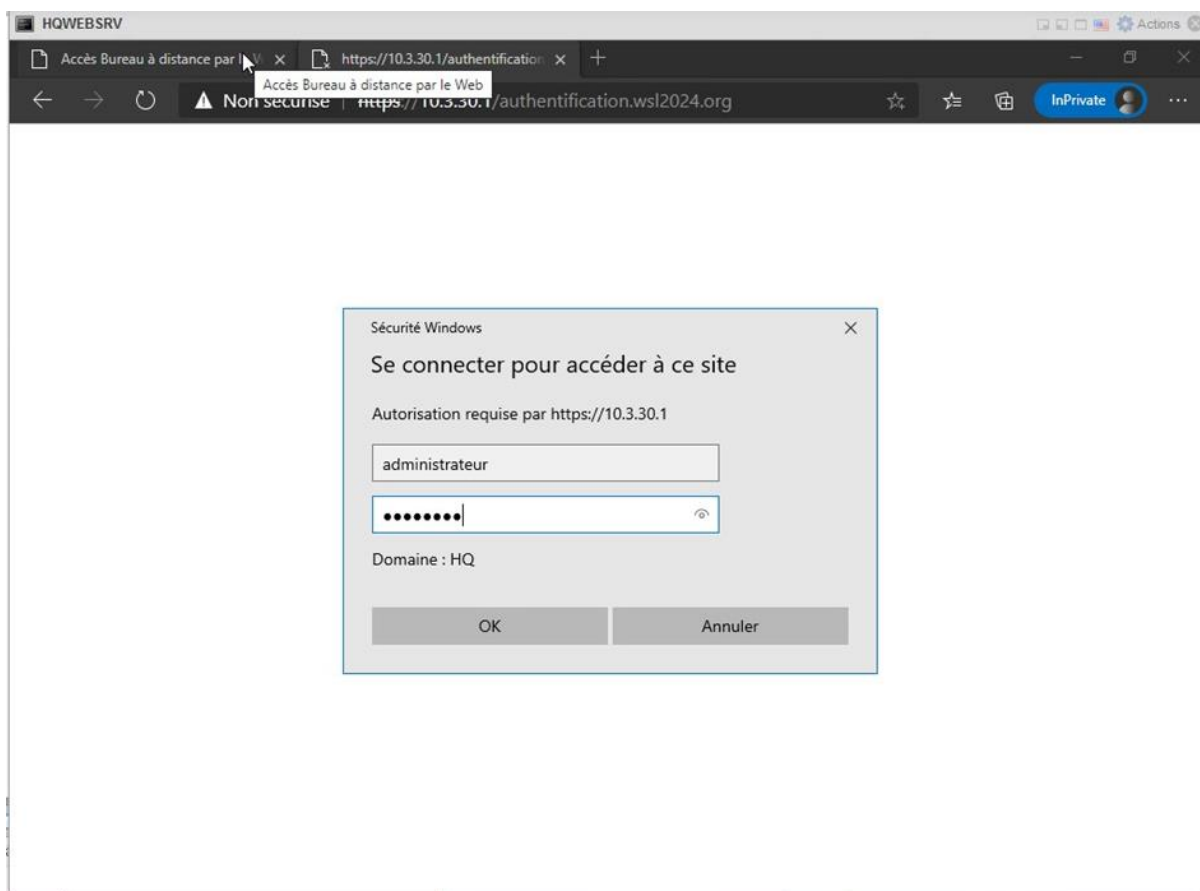


Figure 52 : Connexion sur le site web en HTTPS avec authentification

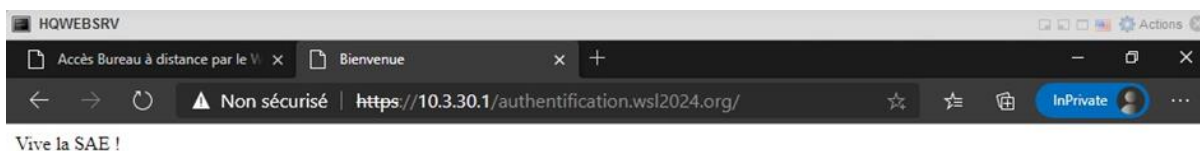


Figure 53 : Accès réussi sur le site web en HTTPS avec authentification

Cette architecture permet un accès sécurisé et contrôlé aux ressources essentielles de l'entreprise depuis Internet, tout en garantissant la confidentialité et la sécurité des données grâce à des protocoles d'authentification rigoureux.

#### 4.5.2. Serveur RDS

Le serveur HQWEBSRV est configuré pour fournir un service MS-RDS, permettant un accès convivial et sécurisé aux applications Microsoft Office telles que Excel et Word via un navigateur web, disponible après authentification.

Interface Affichant les Applications :

Après l'installation nous pouvons voir une gamme d'applications disponibles aux utilisateurs sur le serveur MS-RDS. Cette interface claire et organisée expose différents logiciels tels que Calculatrice et Paint, accessibles facilement via un navigateur web.

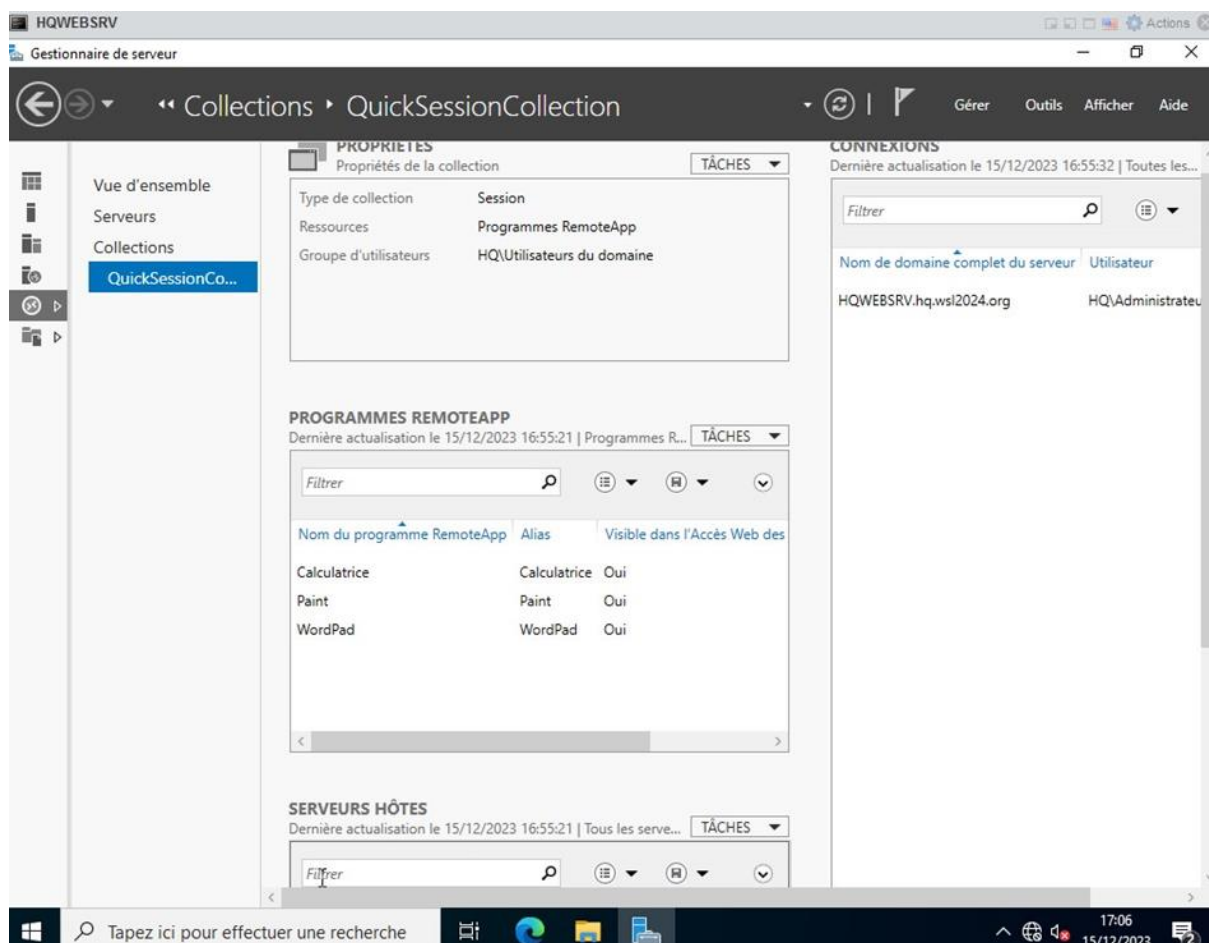


Figure 54 : Visualisation de la Collection contenant les logiciels

#### Authentification sur le Site Web :

Une authentification est requise pour accéder aux applications disponibles via le site web hébergé sur HQWEBSRV. Les utilisateurs doivent fournir leurs identifiants d'authentification Active Directory pour accéder à la page de sélection des applications.

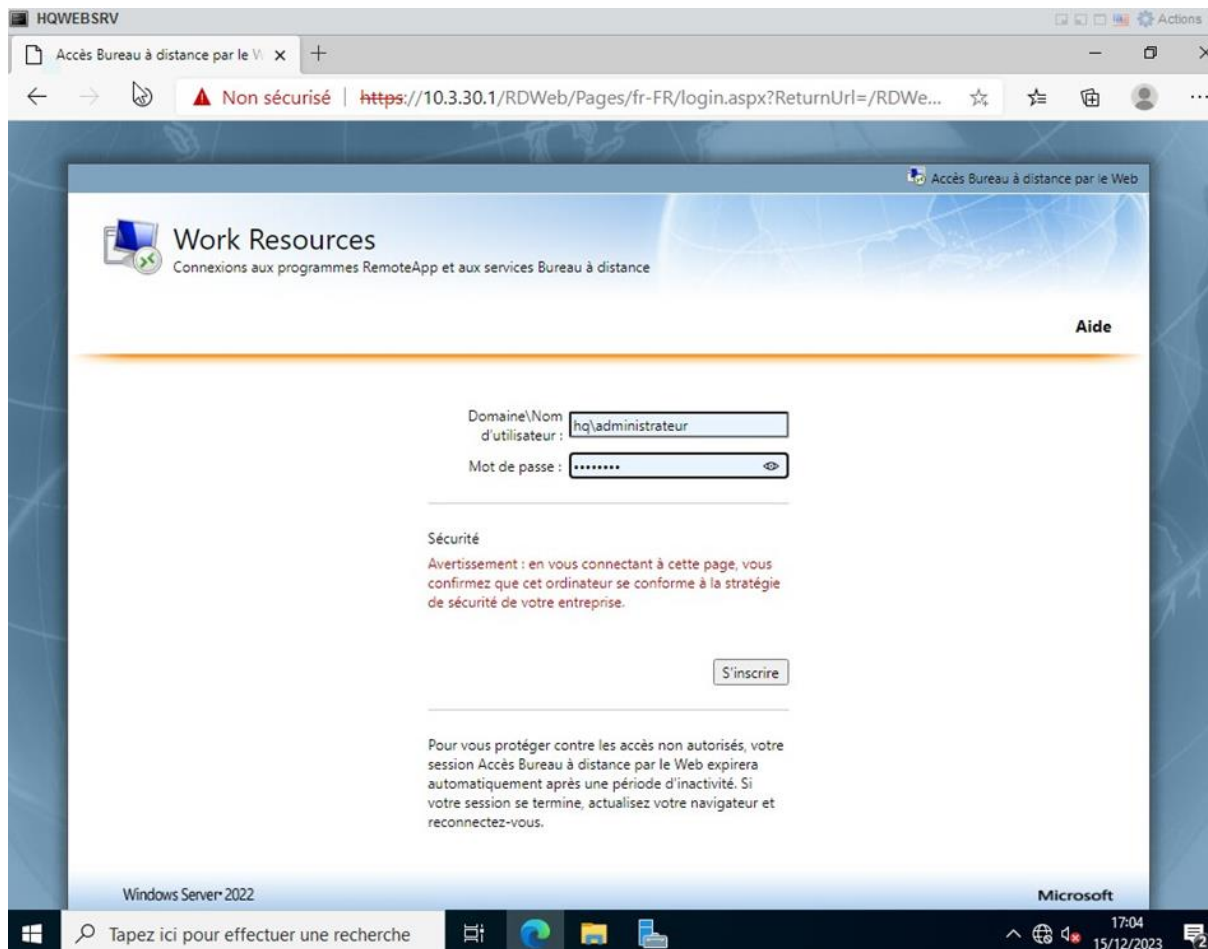


Figure 55 : Authentification sur RDWeb

#### Accès aux Applications depuis le Site Web :

L'interface après une authentification réussie. Les utilisateurs ont accès aux applications telles que Paint, Calculatrice et WordPad via le navigateur web. Ils peuvent ouvrir, utiliser et manipuler ces applications directement depuis le site web, offrant ainsi une expérience fluide et pratique sans nécessiter d'installation locale des logiciels.



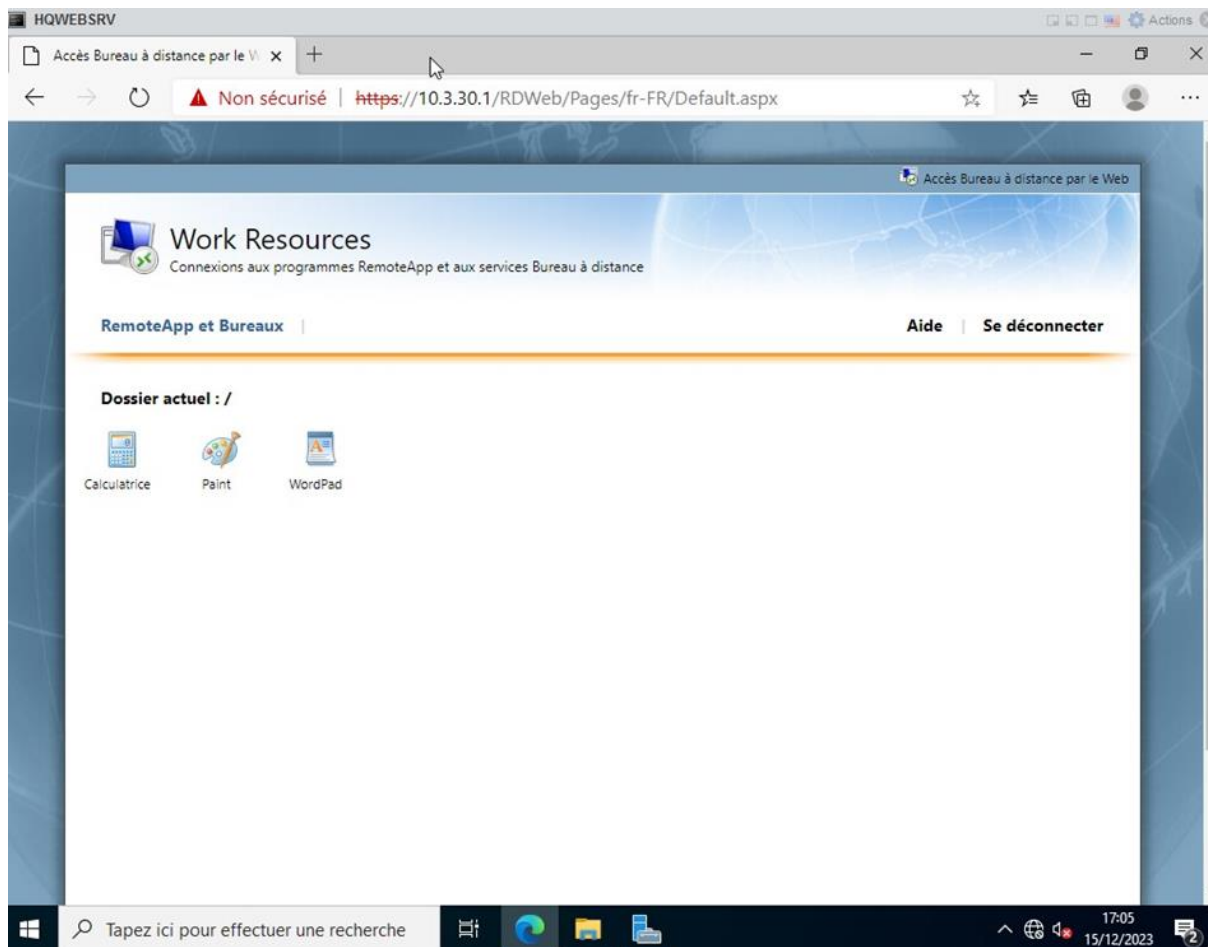


Figure 56 : Accès aux différents logiciels

Cette mise en place offre une expérience utilisateur transparente et sécurisée pour accéder aux ressources professionnelles essentielles depuis n'importe quel emplacement, tout en assurant une sécurité renforcée via l'authentification Active Directory. Optimisant ainsi la productivité et la flexibilité au sein de l'environnement de travail moderne.

## 4.6. HQMAILSRV

### 4.6.1. Configuration de services email

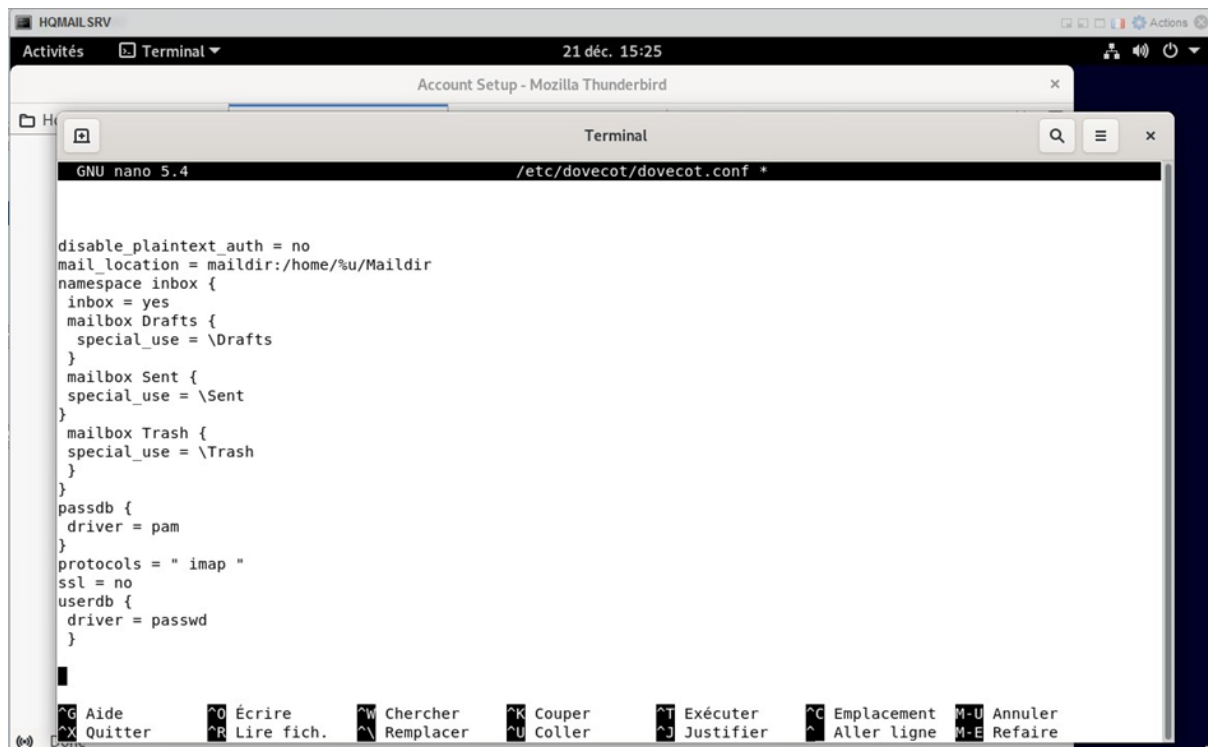
Cette étape représente la configuration d'un serveur d'email sécurisé hébergeant des services SMTP et IMAP. L'objectif principal est d'offrir des services d'email sécurisés utilisant un certificat émis par HQDCSRV. Les clients auront la capacité d'envoyer des emails via SMTPS et de recevoir des emails via IMAPS.

### 4.6.2. Configuration de Dovecot pour un Accès Email Sécurisé

La configuration actuelle de Dovecot a été testée avec le protocole IMAP sans chiffrement SSL/TLS. Cependant, il est prévu de sécuriser prochainement les échanges en mettant en place IMAPS (IMAP sécurisé) avec le support SSL/TLS et en utilisant des certificats pour garantir la confidentialité des données échangées entre le serveur et les clients.

Aperçu de la Configuration Actuelle de Dovecot :

Fichier de configuration :



```
GNU nano 5.4 /etc/dovecot/dovecot.conf *

disable_plaintext_auth = no
mail_location = maildir:/home/%u/Maildir
namespace inbox {
  inbox = yes
  mailbox Drafts {
    special_use = \Drafts
  }
  mailbox Sent {
    special_use = \Sent
  }
  mailbox Trash {
    special_use = \Trash
  }
}
passdb {
  driver = pam
}
protocols = " imap "
ssl = no
userdb {
  driver = passwd
}
```

Figure 57 : Fichier de configuration Dovecot

La configuration actuellement en place utilise le protocole IMAP pour les échanges email. Cependant, cette méthode ne chiffre pas les données transitant entre le serveur et les clients, exposant potentiellement les communications à des risques de sécurité :

#### 4.6.3. Sécurisation avec IMAPS et Certificats SSL/TLS

La prochaine étape consistera à sécuriser les communications email en configurant Dovecot pour utiliser le protocole IMAPS, offrant ainsi un échange de données crypté entre le serveur et les clients. De plus, des certificats SSL/TLS seront utilisés pour renforcer la sécurité des échanges et prévenir toute interception non autorisée des données.

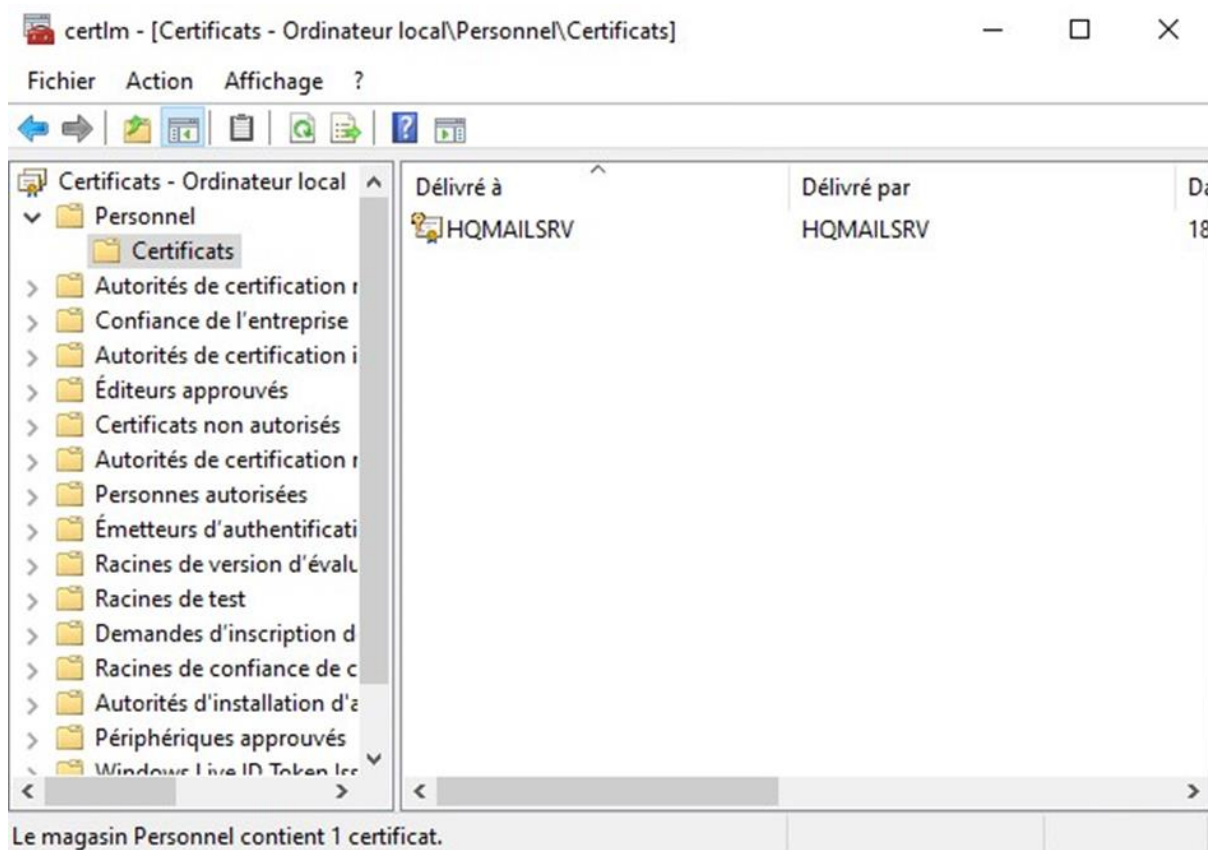


Figure 58 : Certificat HQMAILSRV

Après avoir exporter le certificat, il a été converti en .pem :

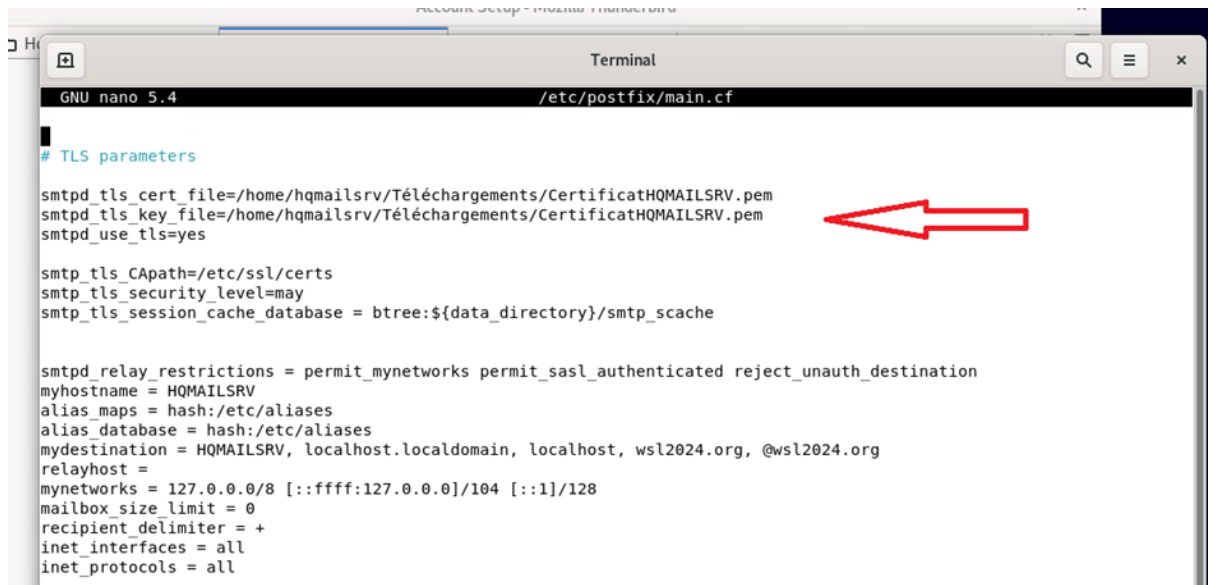
```
ssl_cert = </home/hqmailsrv/Téléchargements/CertificatHQMAILSRV.pem
ssl_key = </home/hqmailsrv/Téléchargements/CertificatHQMAILSRV.pem
ssl = required
```

Figure 59 : Chemin d'accès vers le certificat dans le fichier Dovecot

#### 4.6.4. Configuration de Postfix

La configuration du fichier principal de Postfix, main.cf, a été minutieusement ajustée pour renforcer la sécurité des services email. Les modifications apportées visent à garantir un échange sécurisé de données entre le serveur et les clients, tout en restreignant l'utilisation de protocoles non sécurisés.

Activation de SSL/TLS : Le support SSL/TLS a été activé et configuré pour les connexions SMTPS, assurant ainsi un échange de données crypté et sécurisé lors de l'envoi d'emails par les clients :



```
GNU nano 5.4 /etc/postfix/main.cf

# TLS parameters

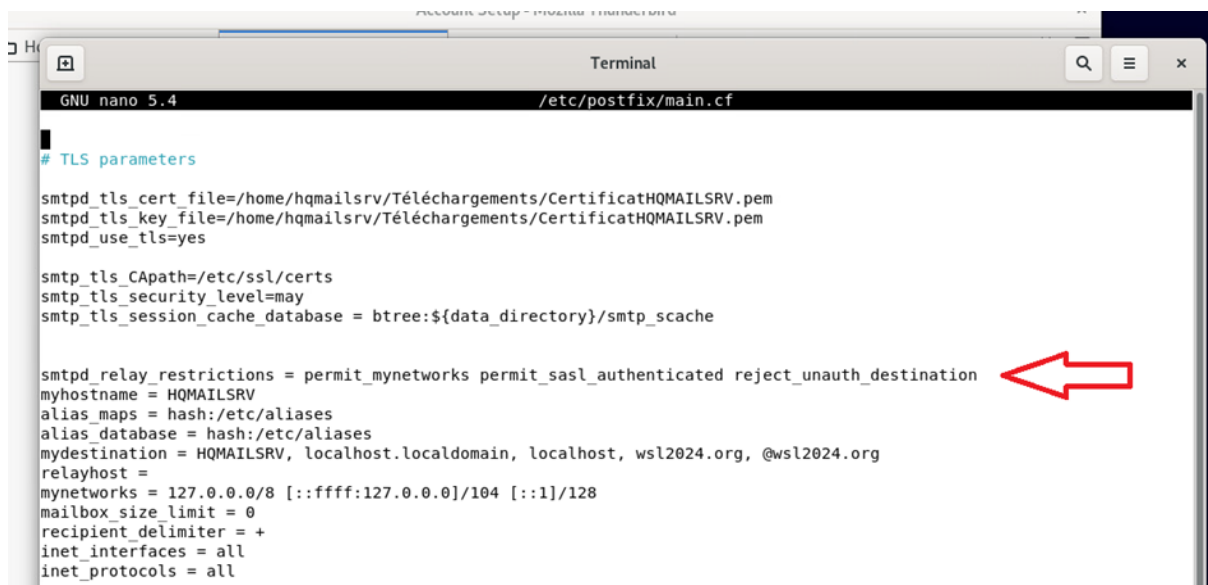
smtpd_tls_cert_file=/home/hqmailsrv/Téléchargements/CertificatHQMailsrv.pem
smtpd_tls_key_file=/home/hqmailsrv/Téléchargements/CertificatHQMailsrv.pem
smtpd_use_tls=yes

smtp_tls_CApath=/etc/ssl/certs
smtp_tls_security_level=may
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache

smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated reject_unauth_destination
myhostname = HQMAILSRV
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
mydestination = HQMAILSRV, localhost.localdomain, localhost, wsl2024.org, @wsl2024.org
relayhost =
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all
```

Figure 60 : Fichier de configuration Postfix

Définition des paramètres de sécurité : Des paramètres spécifiques ont été ajustés pour répondre aux exigences de sécurité, notamment en ce qui concerne la validation des identités lors des communications.



```
GNU nano 5.4 /etc/postfix/main.cf

# TLS parameters

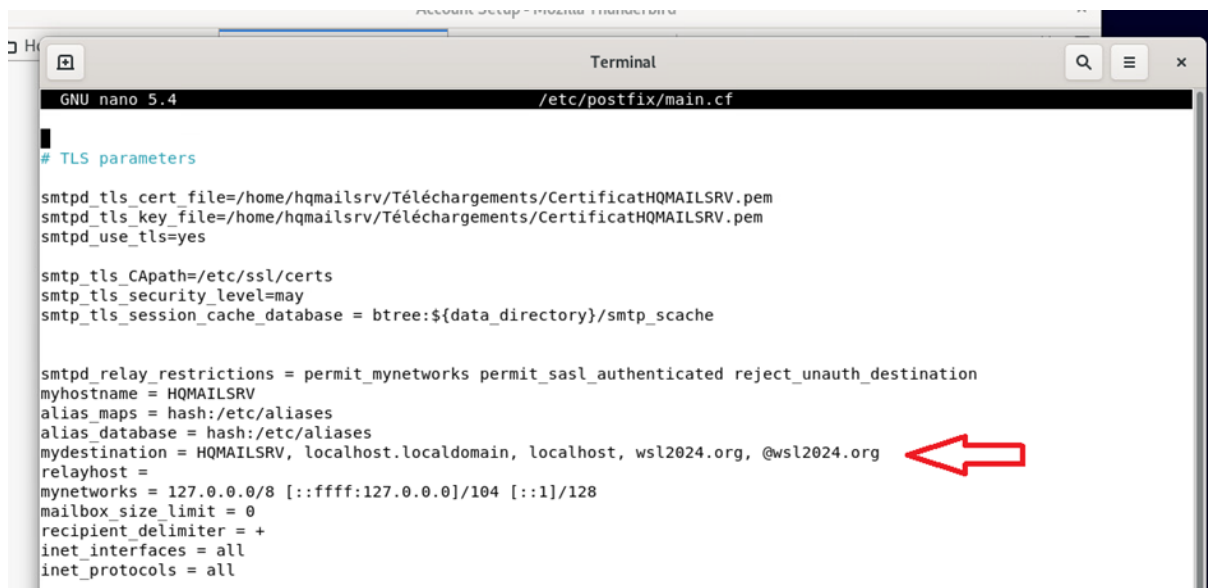
smtpd_tls_cert_file=/home/hqmailsrv/Téléchargements/CertificatHQMailsrv.pem
smtpd_tls_key_file=/home/hqmailsrv/Téléchargements/CertificatHQMailsrv.pem
smtpd_use_tls=yes

smtp_tls_CApath=/etc/ssl/certs
smtp_tls_security_level=may
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache

smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated reject_unauth_destination
myhostname = HQMAILSRV
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
mydestination = HQMAILSRV, localhost.localdomain, localhost, wsl2024.org, @wsl2024.org
relayhost =
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all
```

Figure 61 : Fichier de configuration Postfix

Intégration du Domaine wsl2024.org : Le domaine wsl2024.org a été ajouté à la configuration, permettant ainsi la gestion et la réception des emails pour ce domaine spécifique.



```
GNU nano 5.4 /etc/postfix/main.cf

# TLS parameters

smtpd_tls_cert_file=/home/hqmailsrv/Téléchargements/CertificatHQMailsrv.pem
smtpd_tls_key_file=/home/hqmailsrv/Téléchargements/CertificatHQMailsrv.pem
smtpd_use_tls=yes

smtp_tls_CApath=/etc/ssl/certs
smtp_tls_security_level=may
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache

smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated reject_unauth_destination
myhostname = HQMAILSRV
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
mydestination = HQMAILSRV, localhost.localdomain, localhost, wsl2024.org, @wsl2024.org
relayhost =
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all
```

Figure 62 : Fichier de configuration Postfix

La création d'un nouvel utilisateur sur le serveur de messagerie revêt une importance capitale pour deux raisons principales : l'authentification sécurisée sur des clients email tels que Thunderbird et la mise en place d'une boîte mail dédiée à cet utilisateur.

```
root@HQMailsrv:/home/hqmailsrv# sudo adduser vtim
adduser : L'utilisateur « vtim » existe déjà.
root@HQMailsrv:/home/hqmailsrv#
```

Figure 63 : Création des Users

Test d'Envoi via Telnet : Exécution d'un test d'envoi d'email via Telnet pour démontrer la fonctionnalité et la sécurité des services SMTPS :

```
hqmailsrv@HQMAILSRV: ~
221 2.0.0 Bye
Connection closed by foreign host.
root@HQMAILSRV:/home# sudo gedit /etc/dovecot/dovecot.conf
root@HQMAILSRV:/home# sudo gedit /etc/postfix/main.cf
root@HQMAILSRV:/home# sudo systemctl restart postfix
root@HQMAILSRV:/home# sudo systemctl restart dovecot
root@HQMAILSRV:/home# telnet localhost 25
Trying ::1...
Connected to localhost.
Escape character is '^]'.
220 HQMAILSRV ESMTP Postfix (Debian/GNU)
mail from:vtim@wsl2024.org
250 2.1.0 Ok
rcpt to:npresso@wsl2024.org
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>
test sae.
.
250 2.0.0 Ok: queued as 38DBD21F623
quit
221 2.0.0 Bye
Connection closed by foreign host.
root@HQMAILSRV:/home#
```

Figure 64 : Envoi de mail via TELNET

Réception via Thunderbird : Preuve visuelle de la réception réussie d'un email via IMAPS sur Thunderbird :

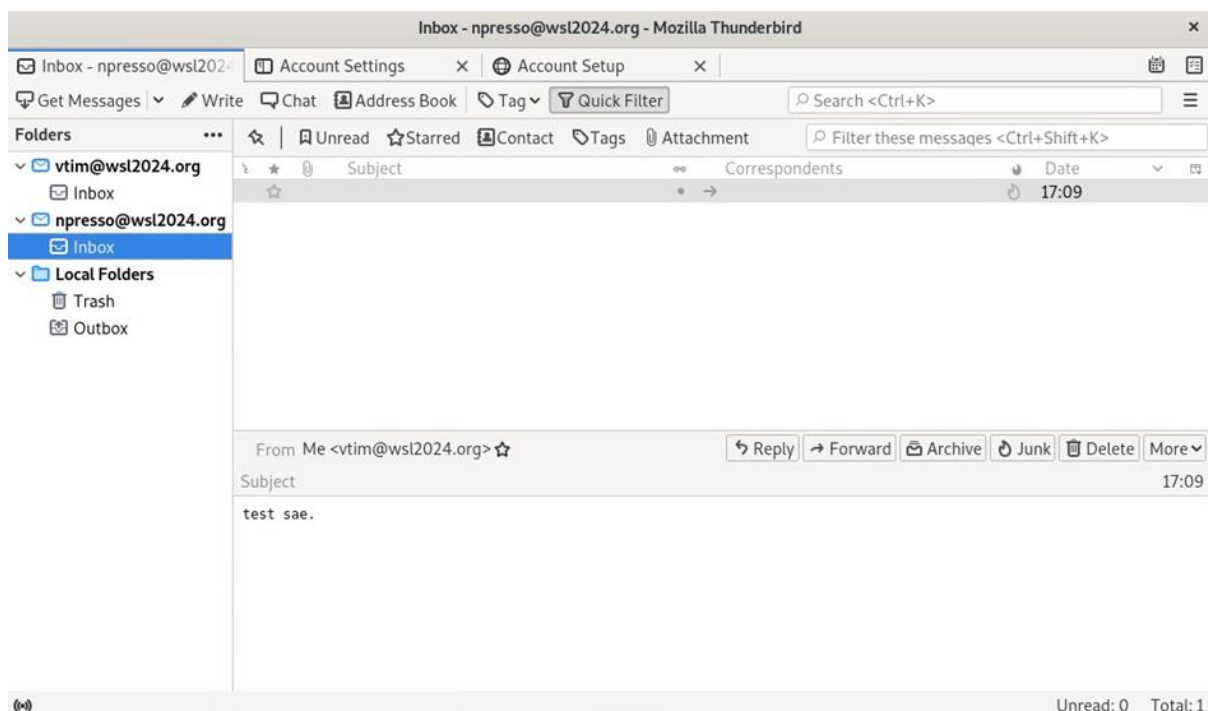


Figure 65 : Boîte mail sur Thunderbird

## 5. Internet

### 5.1. Contexte du projet internet

Le projet s'articule autour des composants clés suivants : INETSRV, INETCLT, et DNSSRV. L'objectif global est de configurer le serveur web à l'aide de Docker Compose, mettre en place une architecture haute disponibilité (HA) avec un load balancer, et configurer un serveur FTP sécurisé. Ce rapport détaille chaque étape du processus, mettant en lumière les difficultés rencontrées et les solutions appliquées.

Configuration du Serveur Web avec Docker Compose, et la configuration de HA avec load\_balancer

### 5.2. Installation et configuration de REMFW

La première étape cruciale de notre projet a consisté à mettre en place une infrastructure web robuste à l'aide de Docker Compose. Cette technologie nous a permis de simplifier considérablement la gestion de nos services, offrant une solution plus élégante que la gestion individuelle de chaque conteneur Docker.

Nous avons opté pour la création de conteneurs individuels NGINX, qui servent de fondation solide pour nos services web. Ces conteneurs sont configurés pour prendre en charge PHP, ajoutant une dimension dynamique à nos pages web. L'objectif était d'assurer une performance optimale tout en garantissant la flexibilité nécessaire pour répondre à nos besoins spécifiques.

```
services:
  nginx1:
    image: nginx:alpine
    ports:
      - "8081:80"
      - "4441:443"
    volumes:
      - ./public_html:/usr/share/nginx/html
      - ./nginx1/conf.d:/etc/nginx/conf.d
      - ./certs:/etc/nginx/certs
    networks:
      - nginx_network
  nginx2:
    image: nginx:alpine
    ports:
      - "8082:80"
      - "4442:443"
    volumes:
      - ./public_html:/usr/share/nginx/html
      - ./nginx2/conf.d:/etc/nginx/conf.d
      - ./certs:/etc/nginx/certs
    networks:
      - nginx_network
  php:
    image: php:7.4-fpm
    volumes:
      - ./public_html:/var/www/html
    networks:
      - nginx_network
  load_balancer:
    image: nginx:alpine
    ports:
      - "80:80"
      - "443:443"
    volumes:
      - ./lb.conf:/etc/nginx/conf.d/default.conf
```

Figure 66 : Capture d'écran de la config vsftpd



### 5.3. Configuration du serveur FTP avec VSFTPD

Dans un monde numérique où le partage de fichiers est essentiel, la configuration d'un serveur FTP sécurisé était une étape incontournable. Nous avons choisi VSFTPD comme notre solution, réputée pour sa robustesse et sa sécurité.

```
listen=NO
listen_ipv6=YES
anonymous_enable=NO
local_enable=YES
write_enable=YES
local_umask=022
dirmessage_enable=YES
use_localtime=YES
xferlog_enable=YES
connect_from_port_20=YES
chroot_local_user=YES
secure_chroot_dir=/var/run/vsftpd/empty
pam_service_name=vsftpd
force_dot_files=YES
pasv_min_port=40000
pasv_max_port=50000

user_sub_token=$USER
local_root=/home/$USER/ftp
```

Figure 67 : Capture d'écran de la config vsftpd

Cette phase de configuration a jeté les bases d'un environnement web solide et d'un serveur FTP robuste, prêt à répondre aux besoins de notre infrastructure. Le choix avisé de Docker Compose et de VSFTPD a permis de créer une solution cohérente et fiable, prête à faire face aux défis à venir.

#### 2.3 Création de l'Utilisateur FTP et Configuration du Répertoire

La mise en place de l'utilisateur "devops" a été une étape cruciale dans notre configuration du serveur FTP avec VSFTPD. Cette démarche a été accomplie avec la plus grande attention pour garantir un équilibre entre la sécurité et la praticité.

```
root@Debian11:~# adduser devops
Ajout de l'utilisateur « devops » ...
Ajout du nouveau groupe « devops » (1002) ...
Ajout du nouvel utilisateur « devops » (1002) avec le groupe « devops » ...
Le répertoire personnel « /home/devops » existe déjà. Rien n'est copié depuis « /etc/skel ».
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd: password updated successfully
Changing the user information for devops
Enter the new value, or press ENTER for the default
    Full Name []: Yasko
    Room Number []: Groupe3
    Work Phone []:
    Home Phone []:
    Other []:
Cette information est-elle correcte ? [0/n]o
```

Figure 68 : Capture d'écran de la création user devops

L'utilisateur "devops" a été soigneusement configuré avec des autorisations spécifiques, assurant ainsi un accès contrôlé aux fonctionnalités FTP. Cette approche minutieuse vise à maintenir une intégrité de sécurité tout en permettant une expérience utilisateur fluide et efficiente.

```

root@Debian11:~# mkdir /home/devops/ftp
root@Debian11:~#
root@Debian11:~# chown nobody:nogroup /home/devops/ftp
root@Debian11:~# chmod a-w /home/devops/ftp
root@Debian11:~#
root@Debian11:~# mkdir /home/devops/ftp/files
root@Debian11:~# chown devops:devops /home/devops/ftp/files
root@Debian11:~#

```

**Figure 69** : Capture d'écran de la préparation des répertoires

La configuration du répertoire FTP a été abordée avec une perspective de sécurité légère, garantissant que seuls les utilisateurs autorisés peuvent accéder aux fichiers pertinents. Cette étape revêt une importance cruciale dans la création d'un environnement FTP robuste et sécurisé.

## 5.4. Configuration de VSFTPD

La configuration initiale de VSFTPD sans TLS a été explorée dans le cadre de notre démarche visant à établir un environnement FTP sécurisé. Une tentative d'utilisation du certificat DNS a été entreprise, soulignant notre engagement envers la sécurité des communications. Cependant, des complications ont émergé, nécessitant une résolution temporaire avec OpenSSL.

```

nano /etc/vsftpd.conf
openssl req -x509 -nodes -days 3650 -newkey rsa:2048 -keyout /etc/ssl/private/vsftpd.pem -out /etc/ssl/private/vsftpd.pem

```

**Figure 70** : Capture d'écran de la génération de certificat avec openssl

Le processus a été l'occasion d'approfondir nos connaissances en matière de sécurité des transferts de fichiers et de renforcer notre engagement envers la protection des données. La démarche exploratoire avec OpenSSL a permis de maintenir la continuité du projet tout en recherchant des solutions plus pérennes.

## 5.5. Configuration de l'environnement Web pour worldskills.org

La configuration de l'environnement web dédié à [www.worldskills.org](http://www.worldskills.org) a été le point d'orgue de notre projet. Cela impliquait non seulement la mise en place des serveurs web NGINX, mais également la résolution de problèmes complexes liés à la haute disponibilité et à la redirection HTTP vers HTTPS. Cette phase délicate a été traitée avec soin pour garantir une expérience utilisateur optimale.

```

1 server {
2     listen 0.0.0.0:80;
3     server_name www.worldskills.org;
4
5     return 301 https://$host$request_uri;
6 }
7
8 server {
9     listen 0.0.0.0:443 ssl;
10    server_name www.worldskills.org;
11

```

**Figure 71** : Capture d'écran de la config web [www.worldskills.org](http://www.worldskills.org)

## 5.6. Difficultés rencontrées

Notre parcours vers la configuration de l'environnement web pour WorldSkills.org n'a pas été exempt de défis significatifs. Parmi ceux-ci, la redirection a posé un défi majeur en raison d'un problème d'itération infinie. De plus, les problèmes d'accès par adresse IP ont généré des complications inattendues, tandis que la configuration DNS a exigé une approche délicate pour

garantir une résolution correcte. Les erreurs de fichiers manquants ont également été un point de friction, tout comme les difficultés d'accès externe au site.

## 5.7. Solutions apportées

Chaque défi a été abordé avec détermination et expertise. La redirection infinie a été corrigée avec une stratégie de redirection plus précise, éliminant ainsi le problème. Pour résoudre les problèmes d'accès par adresse IP, des ajustements au niveau de la configuration ont été réalisés, garantissant une accessibilité sans heurts. La configuration DNS a été révisée avec soin pour garantir une résolution adéquate, tandis que les erreurs de fichiers manquants ont été rectifiées par des ajouts judicieux. Enfin, les problèmes d'accès externe ont été surmontés grâce à des configurations réseau spécifiques.

## 5.8. Conclusion de la configuration de l'environnement Web

Les enseignements tirés de ces défis ont enrichi notre compréhension des systèmes web complexes. Chaque obstacle surmonté a renforcé notre résilience et notre expertise dans la configuration d'un environnement web robuste et hautement disponible. Ces expériences ont contribué à affiner notre approche et à consolider notre savoir-faire dans la gestion des problèmes imprévus, renforçant ainsi la robustesse de notre infrastructure.

## 5.9. Etapes de configuration

Notre parcours vers la mise en place d'une infrastructure hautement disponible avec Nginx et le load balancer a suivi plusieurs étapes clés. Tout d'abord, les serveurs Nginx ont été modifiés pour intégrer les principes de haute disponibilité, suivis de la configuration du load balancer, qui a été soigneusement orchestrée pour assurer une distribution équilibrée du trafic. En parallèle, des ajustements ont été apportés pour permettre aux serveurs de recevoir des connexions sécurisées HTTPS, renforçant ainsi la sécurité de l'ensemble du système.

```
1 upstream backend {
2     server nginx1:443;
3     server nginx2:443 backup;
4 }
5
6 server {
7     listen 0.0.0.0:80;
8     server_name www.worldskills.org;
9
10    if ($scheme = http) {
11        return 301 https://$host$request_uri;
12    }
13 }
14
15 server {
16     listen 0.0.0.0:443 ssl;
17     server_name www.worldskills.org;
18
19     ssl_certificate /etc/nginx/certs/WSFR-ROOT-CA.crt;
20     ssl_certificate_key /etc/nginx/certs/WSFR-ROOT-CA-decrypted.key;
21
22     location / {
23         proxy_pass https://backend;
```

**Figure 74:** Capture d'écran de la config load\_balancer (Haute disponibilité)

## 5.10. Difficultés rencontrées

Cependant, notre parcours n'a pas été sans embûches. L'émergence d'une erreur 400 Bad Request lors de l'accès via HTTPS a constitué un défi significatif. Cette erreur, bien qu'apparemment simple, a nécessité une analyse approfondie pour identifier la source du problème et formuler une solution adéquate.

## 5.11. Solutions apportées

L'erreur 400 Bad Request a été résolue grâce à la détection précise de la source du problème. En examinant de près la configuration du load balancer, nous avons identifié une directive essentielle, `proxy_pass`, qui nécessitait une modification. La correction a été apportée en ajustant la directive pour pointer explicitement vers des connexions sécurisées HTTPS. Cette modification a éliminé l'erreur et a permis un accès ininterrompu au site via HTTPS.

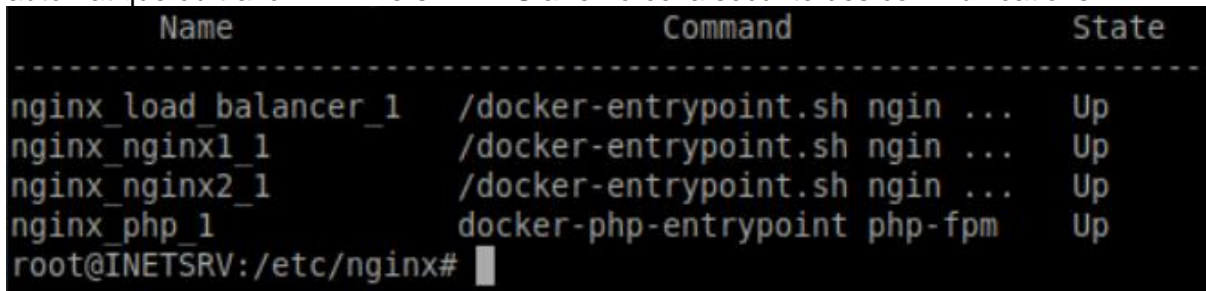
## 5.12. Conclusion de la configuration de l'infrastructure HA

Cette phase de configuration de l'infrastructure hautement disponible a été un succès, démontrant notre capacité à relever des défis complexes et à les surmonter avec succès. Les apprentissages issus de la résolution de l'erreur 400 ont été particulièrement instructifs, renforçant notre expertise dans la gestion des configurations avancées et des problèmes de connectivité. Ces succès s'inscrivent dans notre engagement continu à offrir une infrastructure fiable et résiliente à nos utilisateurs et clients.

### Récapitulation des Configurations Réussies

## 5.13. Récapitulation des configurations réussies

La réussite de notre configuration globale repose sur plusieurs points forts clés : Haute Disponibilité (HA) des Serveurs Web : La mise en place de deux serveurs web Nginx, orchestrés avec Docker Compose, a assuré une haute disponibilité. La redirection automatique du trafic HTTP vers HTTPS a renforcé la sécurité des communications.

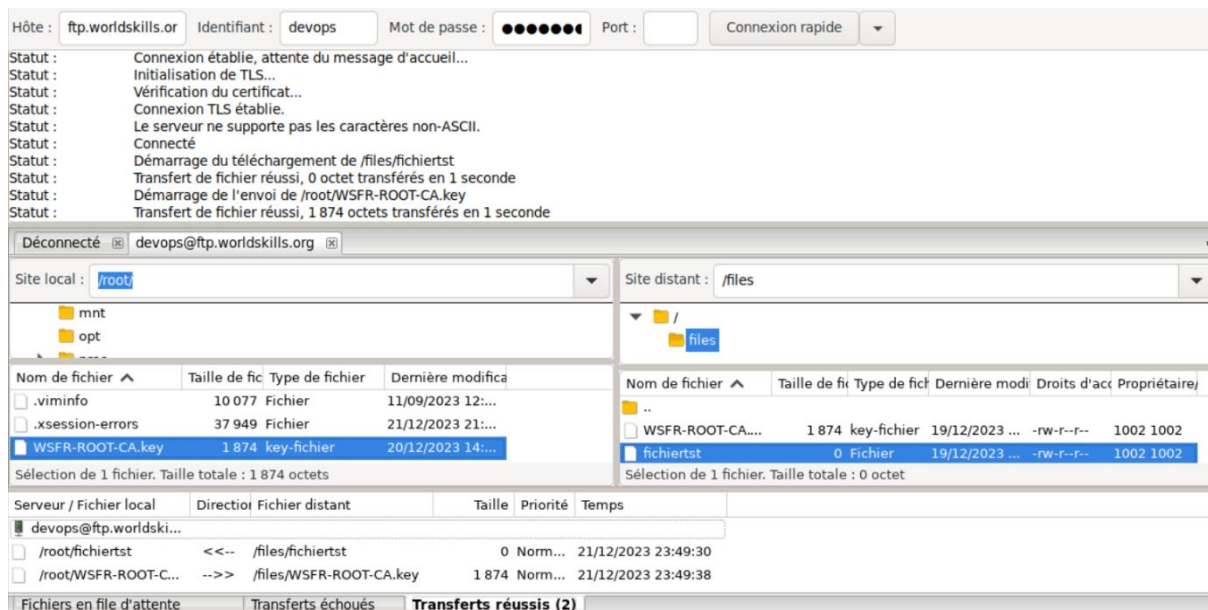


Name	Command	State
nginx_load_balancer_1	/docker-entrypoint.sh nginx ...	Up
nginx_nginx1_1	/docker-entrypoint.sh nginx ...	Up
nginx_nginx2_1	/docker-entrypoint.sh nginx ...	Up
nginx_php_1	docker-php-entrypoint php-fpm	Up

root@INETSRV:/etc/nginx#

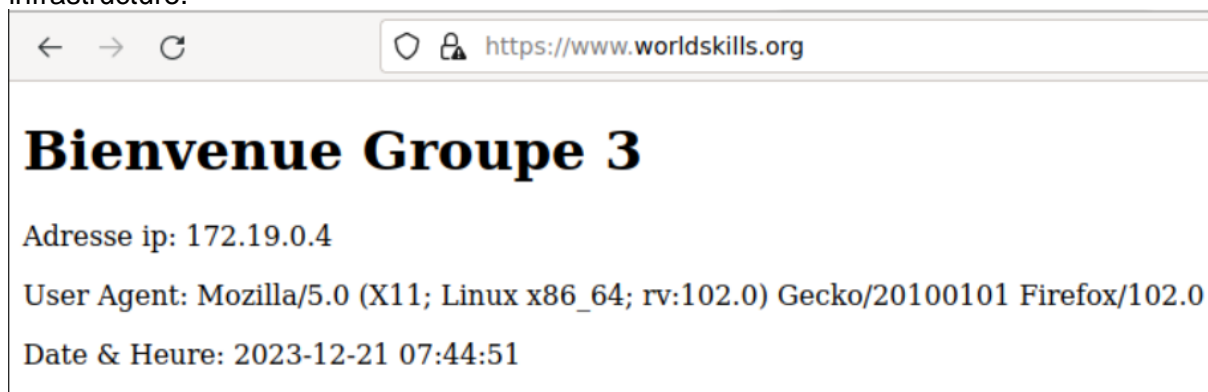
Figure 75 : Capture d'écran du statut des dockers

Serveur FTPS Sécurisé avec VSFTPD : La configuration sécurisée du serveur FTPS a été accomplie avec succès, permettant à l'utilisateur "devops" de télécharger et téléverser des fichiers en toute sécurité.



**Figure 76 :** Capture d'écran de la connexion ftps Fiezilla

Accès à [www.worldskills.org](https://www.worldskills.org) depuis INETCLT : L'accès réussi au site d'entreprise [www.worldskills.org](https://www.worldskills.org) depuis le client simulé INETCLT confirme la connectivité étendue de notre infrastructure.



**Figure 77 :** Capture d'écran de l'accès au site web

Résolution de l'Erreur 400 Bad Request : La résolution rapide et efficace de l'erreur 400 lors de l'accès HTTPS démontre notre réactivité face aux défis rencontrés.

## 5.14. Perspectives d'amélioration

Dans notre quête d'amélioration continue, plusieurs perspectives ont été identifiées :

**Renforcement de la robustesse :** Nous envisageons d'explorer des solutions d'orchestration plus avancées pour une gestion optimale des conteneurs. Cela pourrait inclure l'adoption de Kubernetes pour une orchestration plus puissante.

**Intégration de Mécanismes de Surveillance :** Pour renforcer la sécurité, nous envisageons l'intégration de mécanismes de surveillance avancés, tels que la mise en place de solutions comme Prometheus pour le suivi des performances.

**Évaluation Continue des Protocoles de Sécurité :** Nous restons engagés dans une évaluation continue des protocoles de sécurité pour nous assurer que notre infrastructure reste à la pointe des meilleures pratiques.

## 6. Remote site

La partie Remote Site de l'architecture est un site distant dont les ordinateurs et utilisateurs font partie du domaine rem.wsl2024.org mais ils appartiennent et communiquent avec la forêt principale de hq.wsl2024.org présent sur le site HQ.

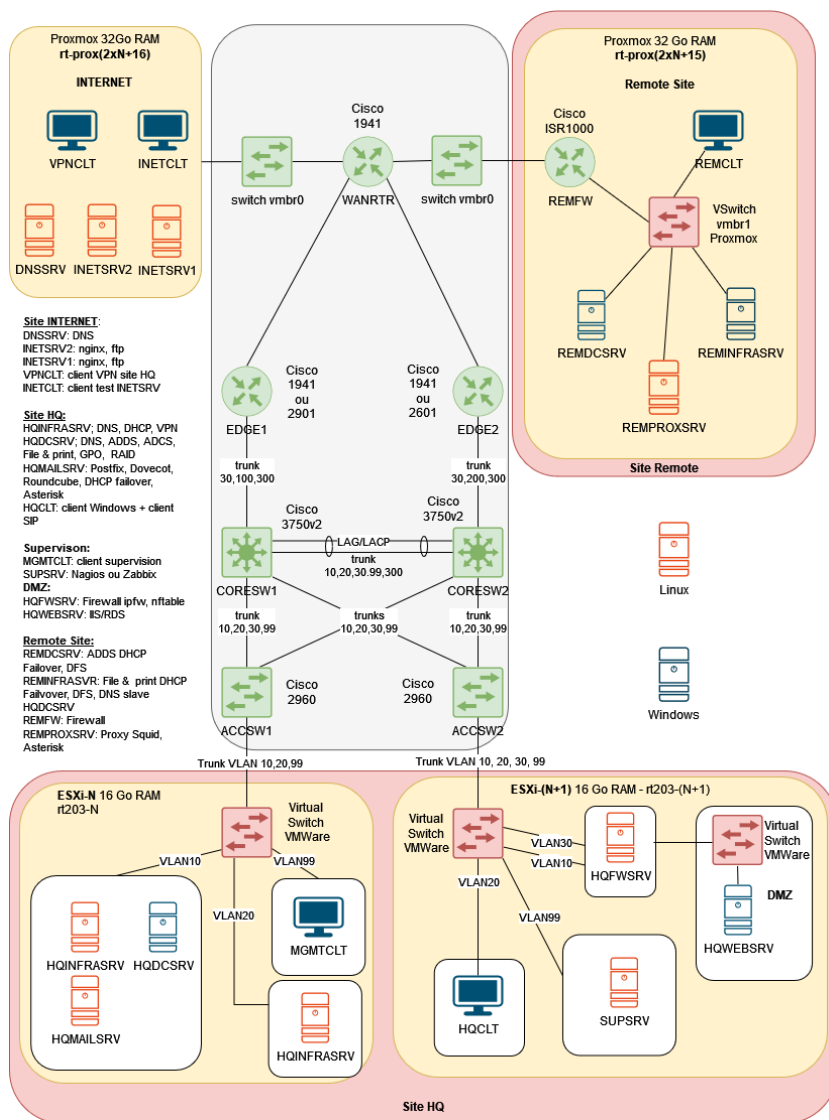


Figure 79 : Infrastructure réseau

Comme indiqué sur l'architecture, le Remote site communique avec le site HQ en passant par le cœur du réseau qui est très complexe.

Le Remote Site possède 3 serveurs et une machine cliente qui nous sert de test mais à l'avenir, il est prévu que 80 machines clientes soient installées et configurées dans cette partie.



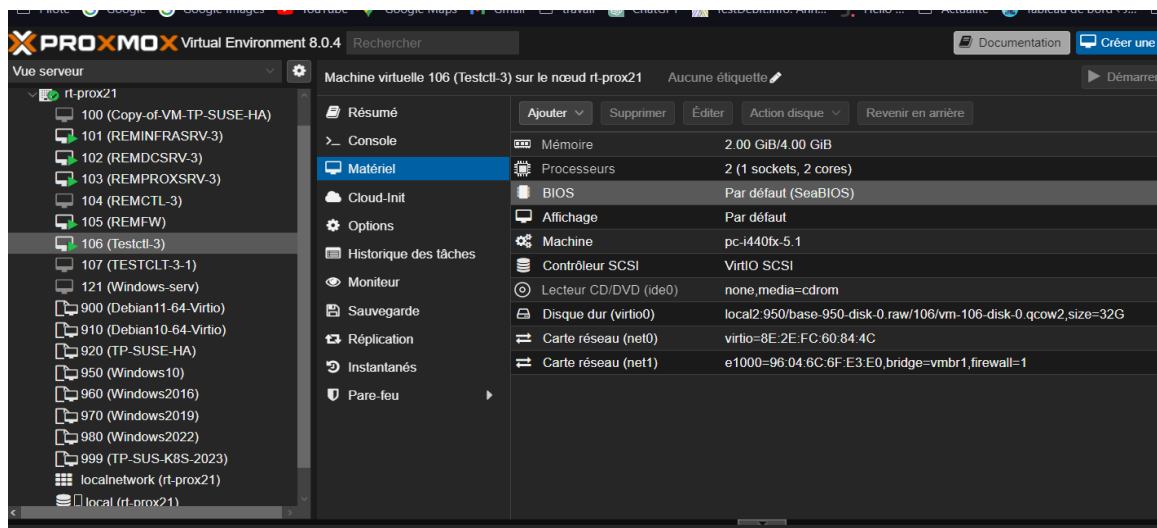


Figure 80 : Interfaces web de Proxmox

Les machines du Remote Site sont installées, configurées et stockées dans une machine de virtualisation Proxmox basée sur une distribution Linux qui nous permet de créer plusieurs machines virtuelles sur une machine physique.

## 6.1. Installation et configuration de REMFW

Pour ce faire, on a créé une machine virtuelle dans laquelle on installe une image ISO d'un routeur cisco pour permettre la connexion entre les machines de remote et le cœur de réseau qui connecte le reste de l'architecture et de configurer un firewall pour contrôler les flux entrant dans le réseau remote.


 csr1000v-universalk9.16.03.06.iso	12/12/2023 08:57	iso Archive	357 414 Ko
---	------------------	-------------	------------

Figure 81 : Fichier ISO

L'image ISO utilisée pour créer notre routeur virtuel cisco.



 Carte réseau (net0)	virtio=4E:BE:38:93:74:D4,bridge=vmbr0,firewall=1
 Carte réseau (net1)	virtio=9E:CB:E7:EE:4E:29,bridge=vmbr1,firewall=1

Figure 82 : Paramètre matériel de la VM REMFW

Ensuite on a attribué deux cartes réseaux au routeur virtuel, le premier qui est vmbr0 permet de nous connecter au cœur de réseau et le deuxième vmbr1 permet de nous connecter au réseau remote avec les serveurs et clients.

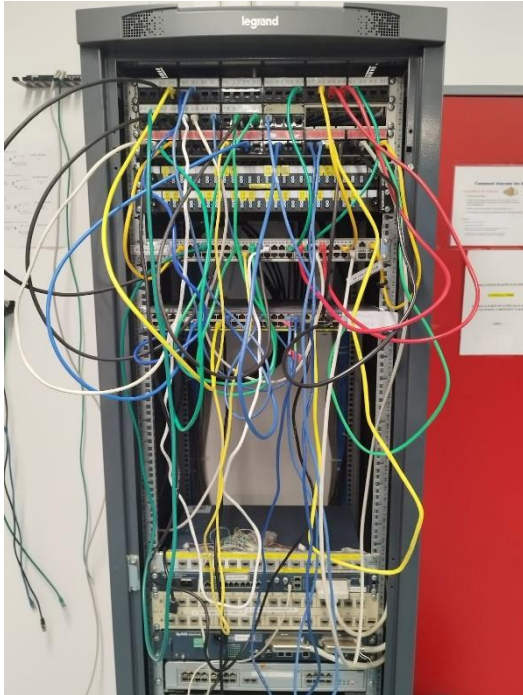


Figure 83 : Switch vmbr0

```
interface GigabitEthernet1
 ip address 10.116.3.1 255.255.255.252
 negotiation auto
 no mop enabled
 no mop sysid
!
interface GigabitEthernet2
 ip address 10.3.100.254 255.255.255.0
 ip access-group ACL-FIREWALL out
 negotiation auto
 no mop enabled
 no mop sysid
.
```

Figure 84: Interfaces réseauX

Après on démarre le routeur virtuel est on configure les interfaces réseaux du routeur en attribuant des adresses IP pour pour interconnecter les réseaux et permettre la communication.

L'Interface GigabitEthernet1 est le vmbr0 qui est connecté au WANRTR du cœur de réseau.

L'interface GigabitEthernet2 est le vmbr1 qui connecte le réseau remote.

```
ip tftp source-interface GigabitEthernet1
ip route 0.0.0.0 0.0.0.0 10.116.3.2
?
```

Figure 85: IP route par défaut

On insère une IP route pour que flux réseau se connecte à tous les réseaux en passant par WANRTR.

```
router ospf 10
 redistribute connected subnets
 network 10.3.100.0 0.0.0.255 area 3
 network 10.116.3.0 0.0.0.3 area 3
?
```

Figure 86: Configuration OSPF

On configure un OSPF sur le routeur qui permet de déterminer le meilleur chemin à emprunter pour les flux de données qui transitent dans le routeur.

```
ip access-list extended ACL-FIREWALL
 permit tcp any any eq 443
 permit tcp any any eq 22
 permit tcp any any eq domain
 permit tcp any any eq msrpc
 permit tcp any any eq 137
 permit tcp any any eq 138
 permit tcp any any eq 139
 permit tcp any any eq 445
 permit tcp any any eq 3389
 permit tcp any any eq 1433
 permit tcp any any eq 1434
 permit tcp any any range 1500 5000
 permit icmp any any
 deny ip any any
?
```

Figure 87 : Règles de Firewall

Et pour terminer, on crée une access list dans laquelle on va écrire des règles de filtrage qui va autoriser les ports à passer dans le réseau remote et une règle qui bloque tout le reste dont on n'a pas besoin.

Dans les règles qu'on autorise uniquement en TCP est :

Le port 443 pour HTTPS, 22 pour SSH, 135 pour RPC, 137 à 139 pour NetBIOS, 445 pour SMB, 3389 pour RDP, 1433 à 1434 pour SQL Server et 1500 à 5000 pour Exchange Server.

Et la dernière règle qui est présente dans la capture ci-dessus bloque tous autres trafics.

Ensuite on assigne notre accès liste dans l'interface réseau GigabitEthernet2 en sortie pour faire appliquer les règles à l'entrée du réseau.

## 6.2. Installation et configuration de REMDCSRV

REMDCSRV est une machine sur Windows serveur 2022 et qui fait ADDS, DNS et DHCP pour les ordinateurs et utilisateurs du réseau situé sur le site remote. Il gère tout ce qui est lecteurs partager, les droits, etc...

- ADDS

C'est un contrôleur de domaine qui gère les ordinateurs et les utilisateurs, tout est pilotable sur un serveur quand-t-il à un adds.

Il contrôle le domaine de rem.wsl2024.org pour le réseau remote mais dépend de la forêt principale de hq.wsl2024.org qui sur le serveur HQDCSRV situé dans la partie HQ de l'architecture.

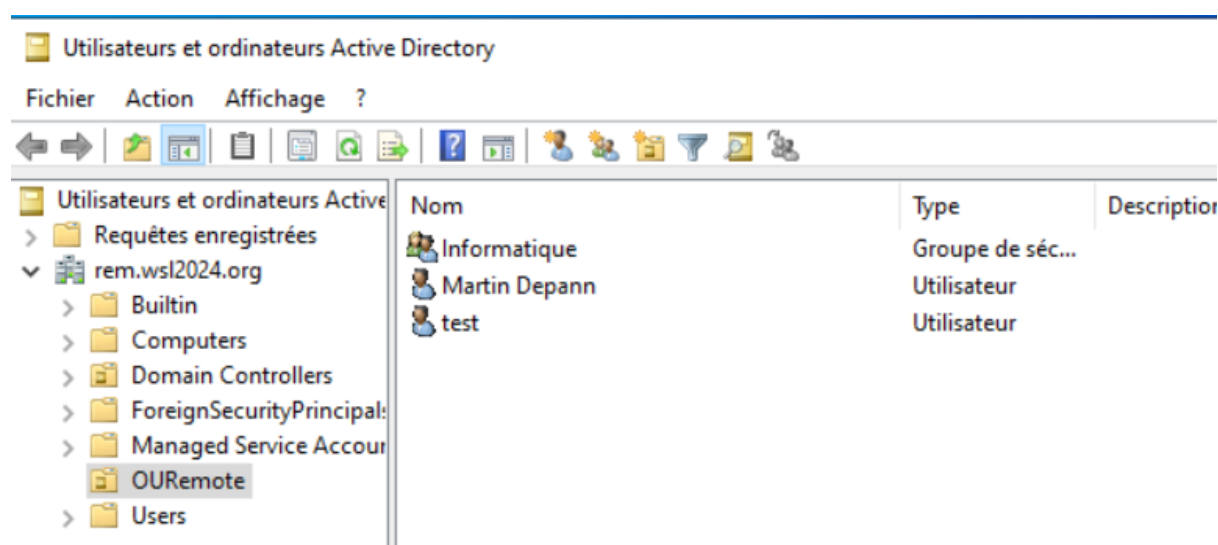


Figure 88 : Configuration OU utilisateurs et groupes

Comme décrit dans la capture ci-dessus, l'outil de créer des Unités d'organisation (OU) dans laquelle on peut créer des utilisateurs et des groupes de sécurité pour les droits.

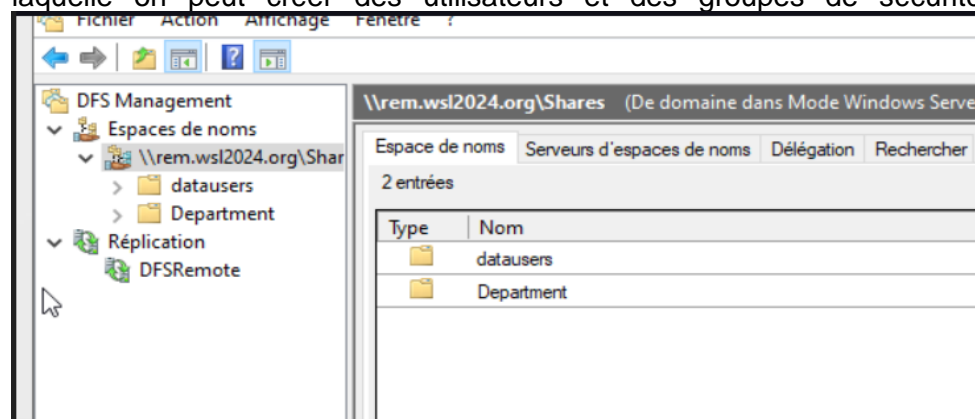


Figure 89 : Service DFS

Ensuite grâce à l'outil DFS qui permet de partager les ressources de stockage et des fichiers de données sur plusieurs machines, on a créé des dossiers qui vont nous permettre de les partager avec les utilisateurs en mappant des lecteurs.

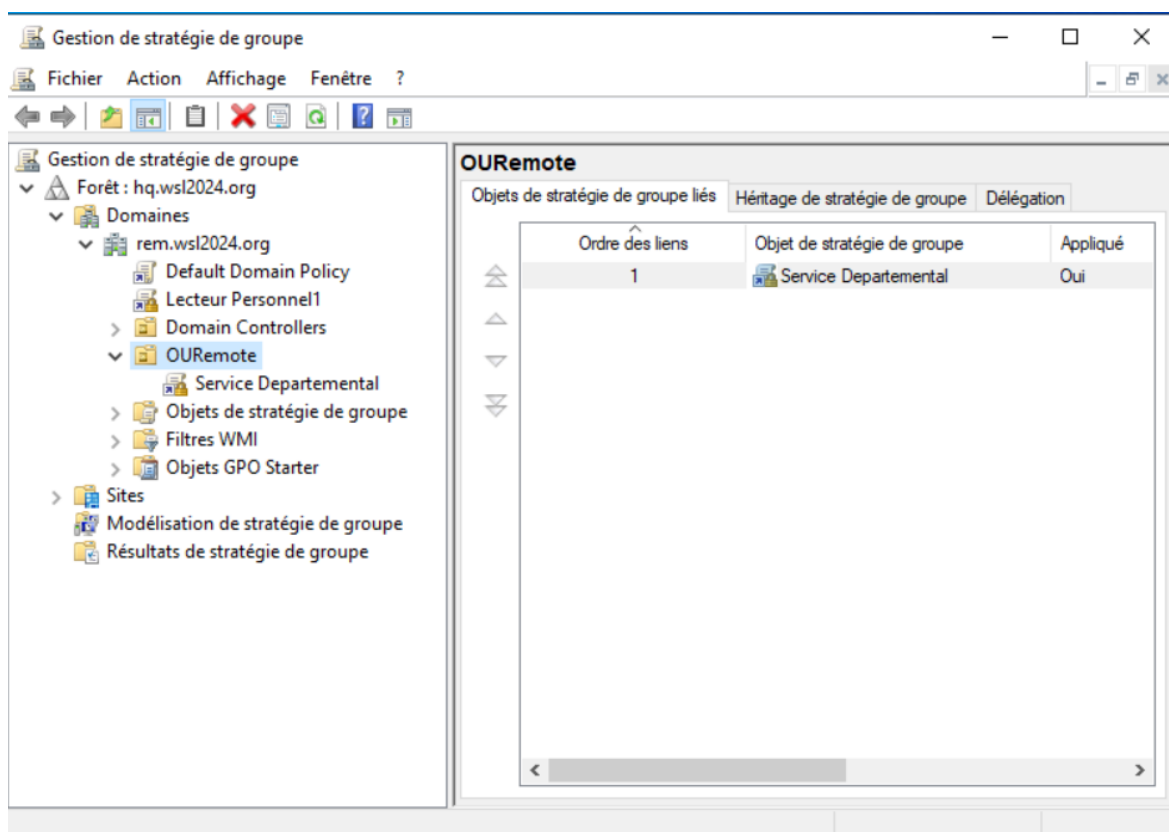


Figure 90: GPO dans le gestionnaire de stratégie de groupe

Après grâce au gestionnaire de stratégie de groupe, on a créé des GPO qui permette de mapper les lecteurs et de modifier les droits d'accès à un service pour un ou plusieurs utilisateurs.

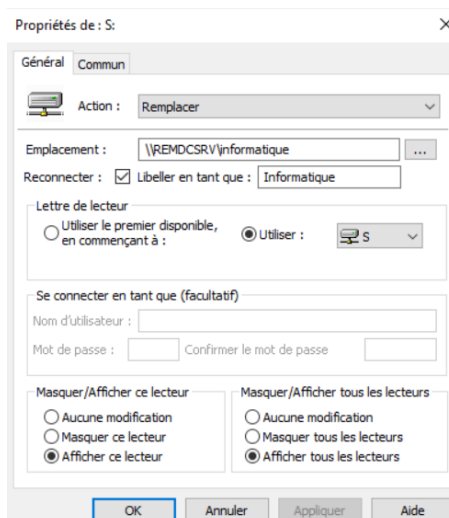


Figure 85 : Mappage lecteur département

- DHCP

Le DHCP permet d'attribuer une configuration IP à une machine client qui est présent dans le réseau qui soit ou pas dans le domaine. Cela permet de connecter les machines de manière automatique pour qu'il puissent communiquer.

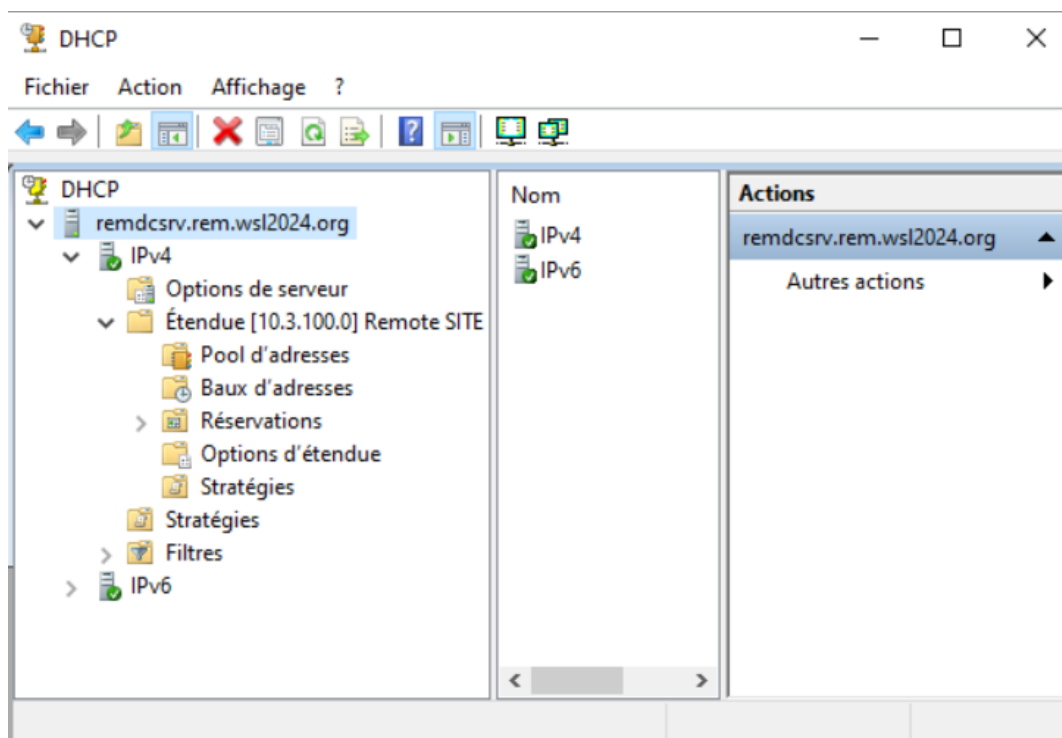


Figure 91 : Serveur DHCP de REMDCSRV

Dans l'exemple, on peut configurer le dhcp en créant une étendue DHCP correspondant au réseau avec une partie de réservation auxquelles les adresses ne bougent pas et les options avec dns, passerelle, etc...

#### Configuration DHCP

- Etendue : 10.3.100.10 à 10.3.100.90
- dns : 10.3.100.1 et 10.3.100.2
- passerelle : 10.3.100.254 celui de REMFW
- masque : 255.255.255.0
- bail : 2 heures

La configuration dhcp est renouvelée toutes les deux heures.

- DNS

Le DNS permet d'attribuer un nom de domaine à des machines et convertir des adresses IP en nom.

Il porte le nom de domaine de rem.wsl2024.org et il a une réplique du nom de domaine hq.wsl2024.org situé sur le serveur DNS de HQDCSRV.

Il fait un forwarding avec le serveur DNS de DNSSERV situé dans la partie internet de l'architecture.



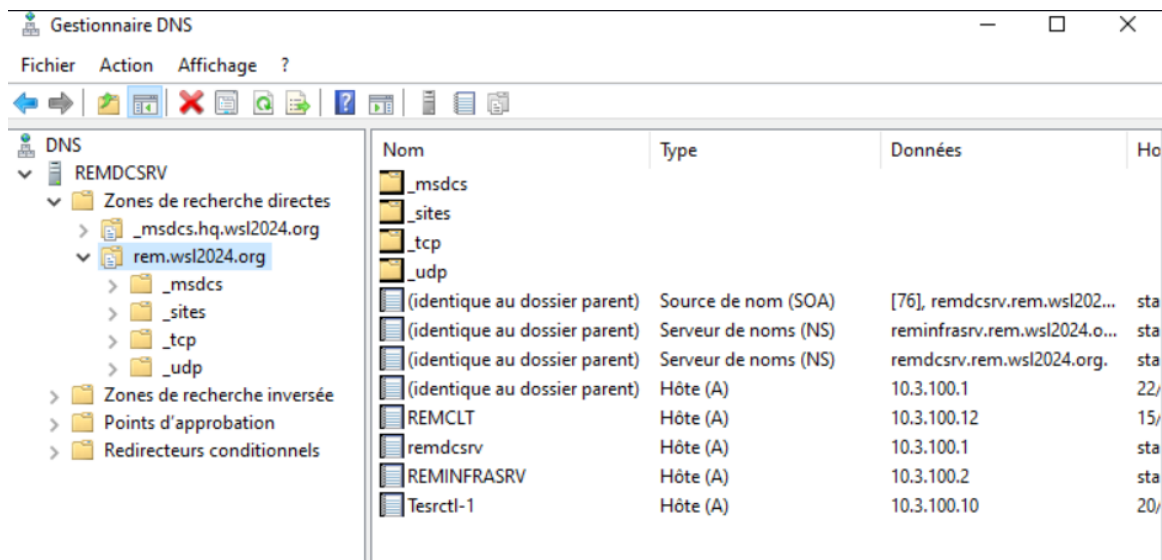


Figure 92: Serveur DNS

On peut voir dans la capture les zones DNS avec les machines qui ont intégré la zone DNS de rem.wsl2024.org.

### 6.3. Installation et configuration de REMINFRASRV

REMINFRASRV est une machine sous Windows serveur 2022 qui sert à faire la réplication du services DFS, du serveur DNS et du serveur DHCP de la machine REMDCSRV.

- DFS

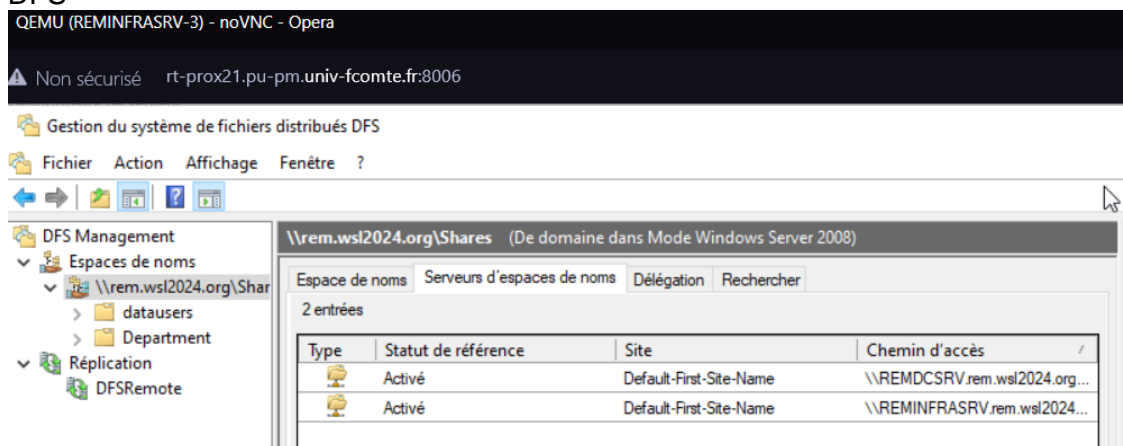


Figure 93 : Service DFS de REMINFRASRV

Comme pour REMSCSRV, on retrouve notre configuration dupliquée sur la machine REMINFRASRV avec les dossiers et les données qui sont dedans. Cela permet d'avoir une disponibilité des données en cas de panne d'une des deux machines.

- DHCP

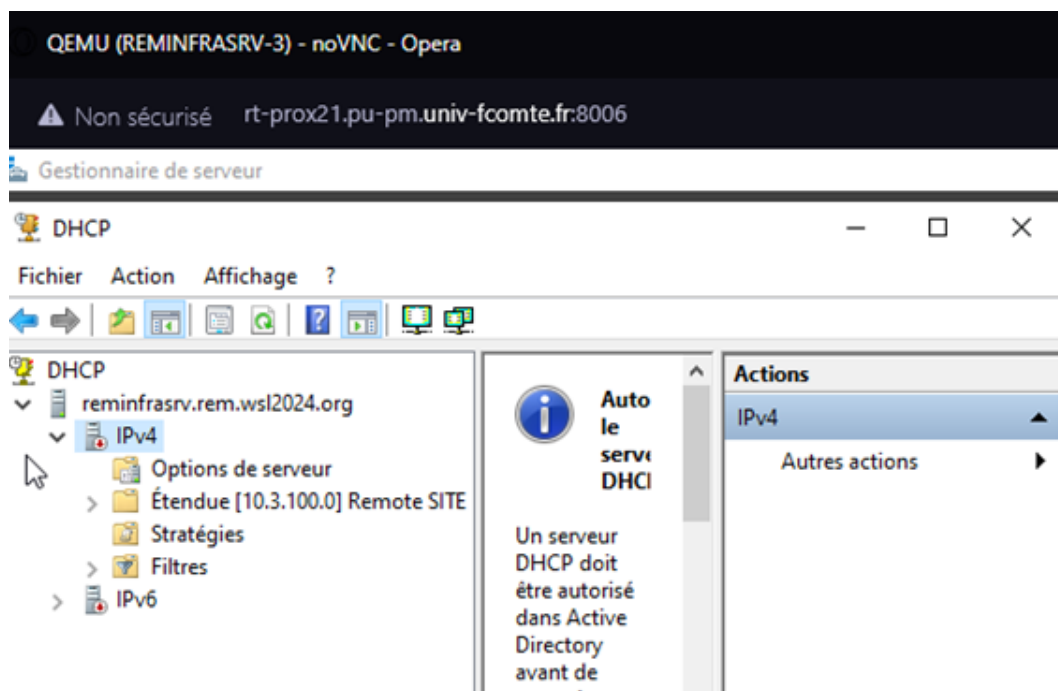


Figure 94 : Serveur DHCP de REMINFRASRV

Sur cette machine, on a configuré un dhcp failover qui réplique la configuration IP grâce à un basculement dhcp qu'on a configuré avec une répartition de charge du dhcp à 50/50 % entre les deux machines.

La répartition de charge permet au deux serveurs dhcp de fonctionner en même temps et d'attribuer les deux des configuration IP aux machines clientes.

!!! Mais on n'a pas pu le mettre en fonction car on n'a pas réussi à authentifier le dhcp dans le domaine.

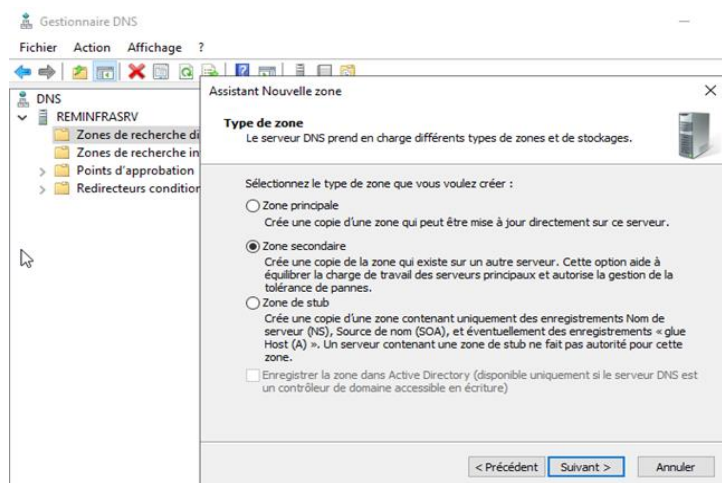


Figure 95 : Création de zone DNS

Lors de configuration du DNS sur REMINFRASRV, on va créer une nouvelle zone qui sera notre zone secondaire en copiant la zone déjà existante sur REMDCSRV.

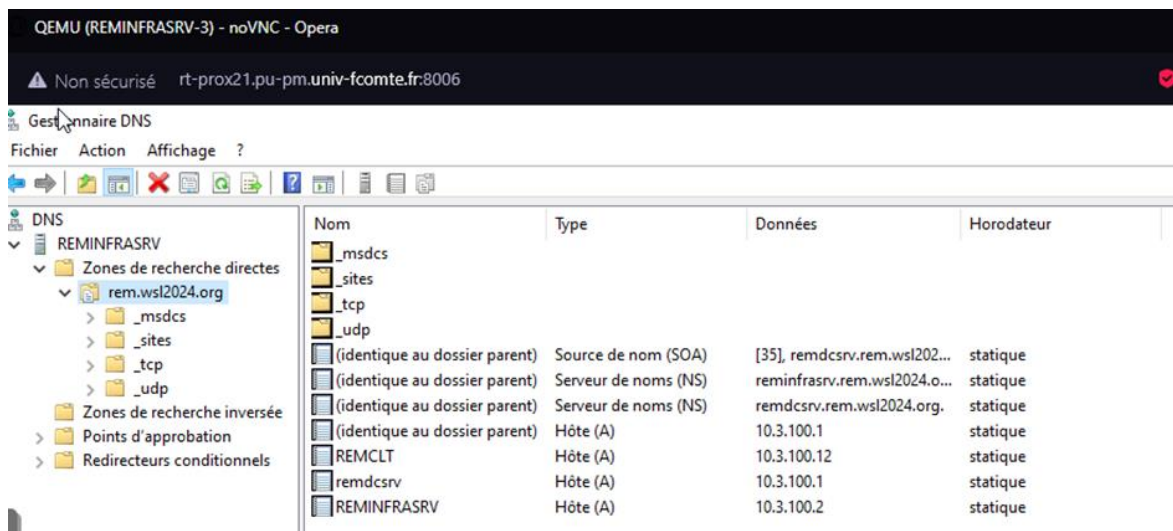


Figure 96 : Serveur DNS de REMINFRASRV

Et normalement, on est censé retrouver la même configuration DNS que sur REMDCSRV avec les machines qui ont reçu un DNS.

!!! Mais on n'a pas réussi à copier la zone `hq.wsl2024.org`

#### 6.4. Installation et configuration de REMPROXSRV

Le REMPROXSRV est une machine utilisant une distribution Linux dans laquelle on a installé le paquet SQUID qui permet de configurer et d'appliquer un proxy dans le réseau de site remote.

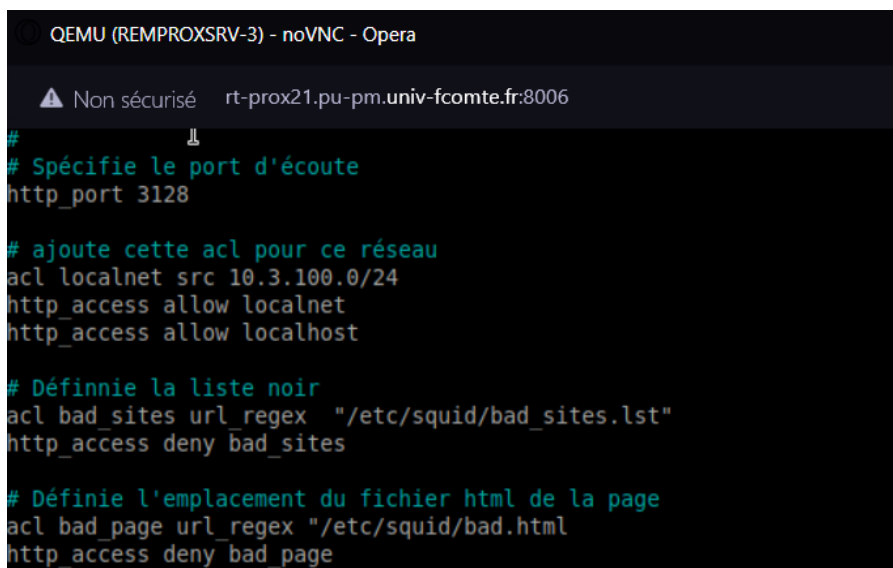
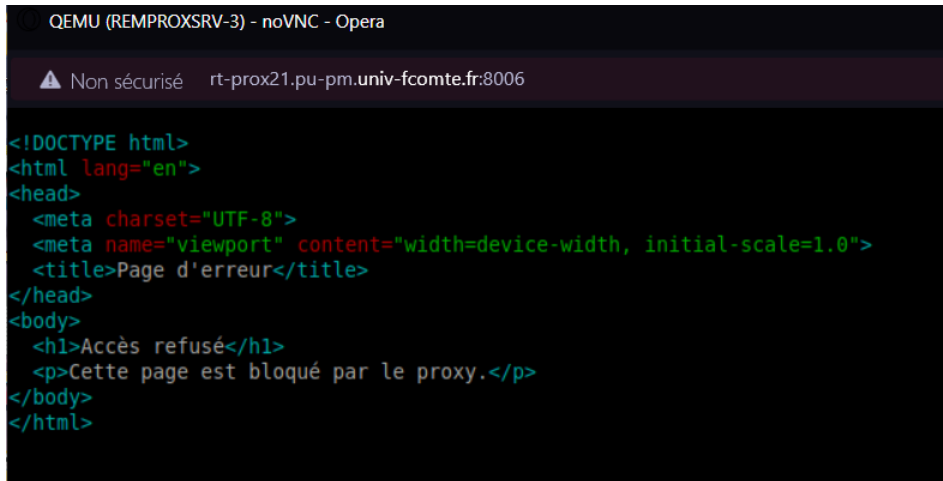


Figure 97 : Fichier squid.conf

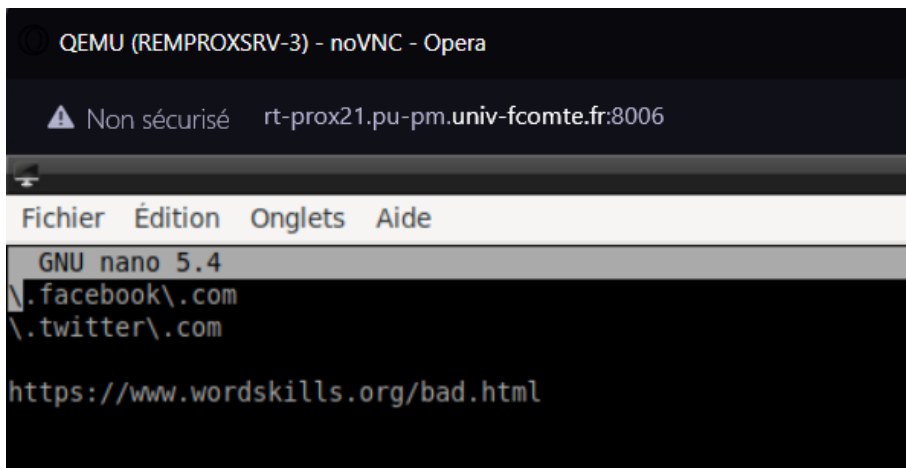
Pour commencer, on configure le fichier squid.conf qui permet de définir le fonctionnement du proxy en configurant le port d'écoute, la liste des sites bloqués, le fichier html de la page web, etc...



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Page d'erreur</title>
</head>
<body>
  <h1>Accès refusé</h1>
  <p>Cette page est bloqué par le proxy.</p>
</body>
</html>
```

Figure 98 : Fichier Bad.html

Ensuite, on configure le fichier bad.html qui permet d'afficher une page web lorsque le proxy bloque un site il affiche directement cette page.



```
GNU nano 5.4
\.facebook\.com
\.twitter\.com
https://www.wordskills.org/bad.html
```

Figure 99: Fichier Bad\_site.lst

Après, on édite le fichier de la liste des sites qu'on veut bloqué comme pour ce qui est demandé dans le sujet de bloqué le site : <https://www.wordskills.org/bad.html> pour que les utilisateurs de remote n'aient pas accès au site.

On a aussi ajouté d'autres sites très connus pour exemples qu'on peut tous bloqués.



Figure 100: Page web du proxy

Normalement, lorsqu'un utilisateur fait requêtes vers l'un des sites qu'on à bloqué, il doit avoir cette page web du proxy qui s'ouvre.

## 6.5. Test sur une machine client REMCLT

Une fois que nous avons configuré nos serveurs, on peut faire le test avec une machine client toute simple en utilisant Windows 10.

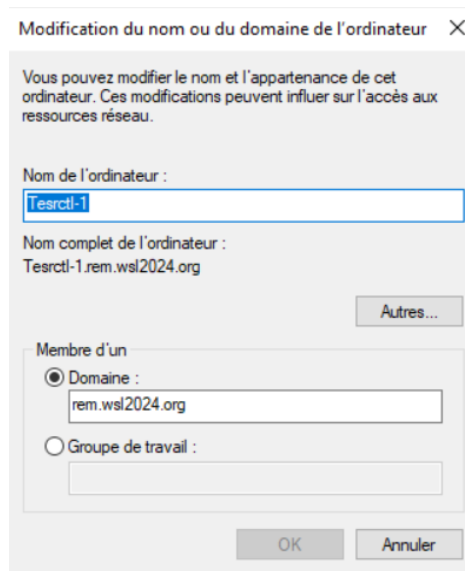


Figure 101: Modif du nom et mise en domaine

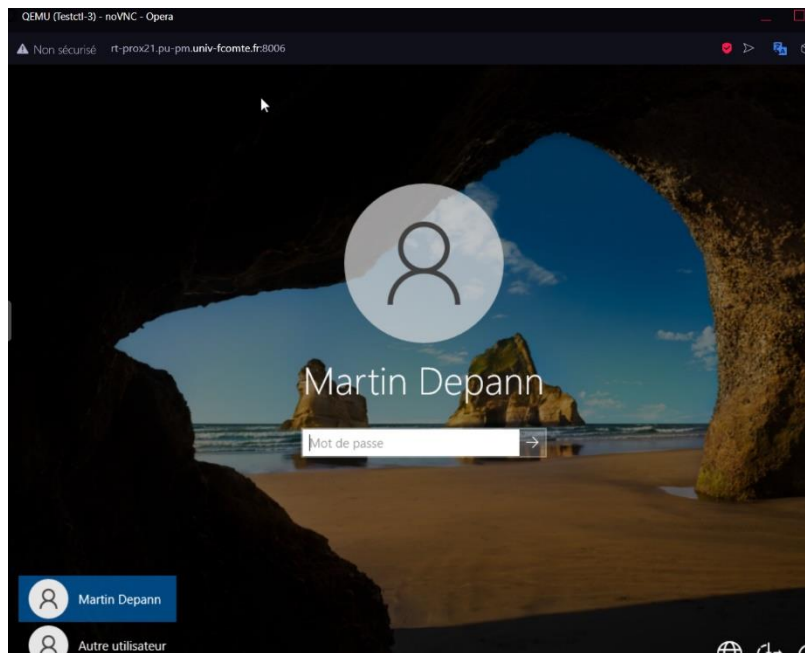


Figure 102 : Démarrage Windows

Une fois que la machine est dans le domaine, on va utiliser un utilisateur qu'on a créé sur le serveur pour pouvoir nous authentifier à la session. Le premier chargement est long car il importe la configuration utilisateur du serveur en local sur la machine.

```

Carte Ethernet Ethernet 6 :

Suffixe DNS propre à la connexion. . . : rem.wsl2024.org
Description. . . . . : Intel(R) PRO/1000 MT Network Connection #3
Adresse physique . . . . . : 96-04-6C-6F-E3-E0
DHCP activé. . . . . : Oui
Configuration automatique activée. . . : Oui
Adresse IPv6 de liaison locale. . . . : fe80::ff27:ff2b:2f59:ae14%15(préfééré)
Adresse IPv4. . . . . : 10.3.100.10(préfééré)
Masque de sous-réseau. . . . . : 255.255.255.0
Bail obtenu. . . . . : jeudi 21 décembre 2023 08:10:47
Bail expirant. . . . . : jeudi 21 décembre 2023 16:19:16
Passerelle par défaut. . . . . : 10.3.100.254
Serveur DHCP . . . . . : 10.3.100.1
IAID DHCPv6 . . . . . : 378930284
DUID de client DHCPv6. . . . . : 00-01-00-01-2D-14-F8-C2-8E-2E-FC-60-84-4C
Serveurs DNS. . . . . : 10.3.100.2
                        10.3.100.1
NetBIOS sur Tcpip. . . . . : Activé

```

Figure 103 : Configuration IP de la machine

Une fois que le profil de l'utilisateur a fini de charger sur la machine et que si on fait un ipconfig /all dans l'invite de commande, on voit que la machine client a une configuration IP à ce qui était configuré sur le serveur DHCP.



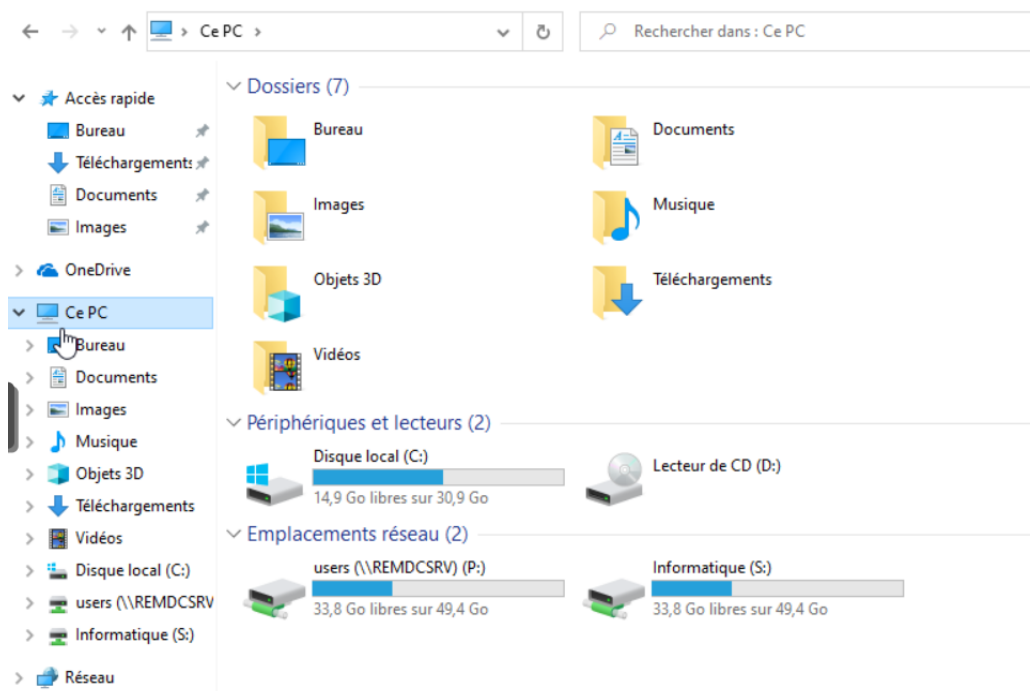


Figure 104 : explorateur de fichier

Et pour terminer, si on va dans “Ce PC” de l’explorateur de fichier on a deux lecteurs réseaux qui sont apparus : un lecteur personnel propre à l’utilisateur et un lecteur commun pour le département.

## 7. Conclusion

Pour conclure, cette SAE nous a permis de développer notre capacité à travailler plus efficacement en ayant une méthode de gestion de projet qui nous permet d’être mieux organiser et de mieux visualiser ce qu’on a à faire. En ce qui concerne la partie réseau, elle était beaucoup plus importante avec la mise en place de différents protocoles où nous n’avons pas eu l’occasion de les déployer lors des SAE précédentes donc cela a représenté un vrai défi pour nous. Nous sommes un peu déçus de ne pas avoir pu finir tous les jalons mais également le fait que le déploiement de certains services a échoué. Malgré cela, nous pensons que nous avons amélioré nos compétences lors de ce projet.

## 8. Annexe

## 8.1. Configuration routeur WANRTR

## 8.2. Configuration routeur EDGE1

```
EDGE1#sh run
Building configuration...

Current configuration : 8965 bytes
!
! Last configuration change at 06:39:09 UTC Thu Dec 21 2023
!
version 15.5
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
no service dhcp
!
hostname EDGE1
!
boot-start-marker
boot-end-marker
!
enable secret 5 $1$5VvR$eK7nS59k39B.KPndJ1p7m1
!
no aaa new-model
ethernet lmi ce
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!
!
```



```

interface GigabitEthernet0/1
ip address 10.3.254.5 255.255.255.252
ip nat inside
ip virtual-reassembly in
duplex auto
speed auto
!
interface GigabitEthernet0/1.10
encapsulation dot1Q 10
ip address 91.3.222.97 255.255.255.252
ip nat outside
ip virtual-reassembly in
!
!
router ospf 10
redistribute connected subnets
redistribute static subnets
redistribute bgp 65316 subnets
network 10.3.254.0 0.0.0.3 area 3
network 10.3.254.4 0.0.0.3 area 3
network 10.3.254.20 0.0.0.3 area 3
!
router bgp 65316
bgp log-neighbor-changes
neighbor 91.3.222.98 remote-as 65330
!
address-family ipv4
network 91.3.222.96 mask 255.255.255.252
network 217.3.160.0 mask 255.255.255.252
redistribute connected
redistribute static
redistribute ospf 10
neighbor 91.3.222.98 activate
exit-address-family
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
ip nat inside source list 1 interface GigabitEthernet0/1.10 overload
ip nat inside source static tcp 10.3.10.2 80 191.5.157.33 80 extendable
ip nat inside source static tcp 10.3.10.2 443 191.5.157.33 443 extendable
ip nat inside source static tcp 10.3.10.62 4443 191.5.157.33 4443 extendable
ip route 10.3.0.0 255.255.0.0 10.3.254.2
!
!
!
snmp-server community supervision RO
snmp-server enable traps snmp authentication linkdown linkup coldstart warmstart
snmp-server enable traps vrrp
snmp-server enable traps pfr
snmp-server enable traps flowmon
snmp-server enable traps transceiver all
snmp-server enable traps ds1
snmp-server enable traps call-home message-send-fail server-fail
snmp-server enable traps tty
snmp-server enable traps eigrp
snmp-server enable traps ospf state-change
snmp-server enable traps ospf errors

```

```

snmp-server enable traps ospf retransmit
snmp-server enable traps ospf lsa
snmp-server enable traps ospf cisco-specific state-change nssa-trans-change
snmp-server enable traps ospf cisco-specific state-change shamlink interface
snmp-server enable traps ospf cisco-specific state-change shamlink neighbor
snmp-server enable traps ospf cisco-specific errors
snmp-server enable traps ospf cisco-specific retransmit
snmp-server enable traps ospf cisco-specific lsa
snmp-server enable traps gatekeeper
snmp-server enable traps xgcp
snmp-server enable traps license
snmp-server enable traps envmon
snmp-server enable traps flash insertion removal low-space
snmp-server enable traps auth-framework sec-violation auth-fail
snmp-server enable traps c3g
snmp-server enable traps LTE
snmp-server enable traps ds3
snmp-server enable traps adslline
snmp-server enable traps vdsl2line
snmp-server enable traps icsudsu
snmp-server enable traps isdn call-information
snmp-server enable traps isdn layer2
snmp-server enable traps isdn chan-not-avail
snmp-server enable traps isdn ietf
snmp-server enable traps ds0-busyout
snmp-server enable traps ds1-loopback
snmp-server enable traps energywise
snmp-server enable traps vstack
snmp-server enable traps mac-notification
snmp-server enable traps trustsec-sxp conn-srcaddr-err msg-parse-err conn-config-err binding-err
conn-up conn-down binding-expn-fail oper-nodeid-change binding-conflict
snmp-server enable traps bgp
snmp-server enable traps bgp cbgp2
snmp-server enable traps isis
snmp-server enable traps ospfv3 state-change
snmp-server enable traps ospfv3 errors
snmp-server enable traps aaa_server
snmp-server enable traps atm subif
snmp-server enable traps cef resource-failure peer-state-change peer-fib-state-change
inconsistency
snmp-server enable traps memory bufferpeak
snmp-server enable traps cnpd
snmp-server enable traps config-copy
snmp-server enable traps config
snmp-server enable traps config-ctid
snmp-server enable traps entity-ext
snmp-server enable traps entity
snmp-server enable traps fru-ctrl
snmp-server enable traps resource-policy
snmp-server enable traps event-manager
snmp-server enable traps frame-relay multilink bundle-mismatch
snmp-server enable traps frame-relay
snmp-server enable traps frame-relay subif
snmp-server enable traps hsrp
snmp-server enable traps ipmulticast
snmp-server enable traps mempool
snmp-server enable traps msdp
snmp-server enable traps mvpn
snmp-server enable traps nhrp nhs
snmp-server enable traps nhrp nhc

```

```

snmp-server enable traps nhrp nhp
snmp-server enable traps nhrp quota-exceeded
snmp-server enable traps pim neighbor-change rp-mapping-change invalid-pim-message
snmp-server enable traps pppoe
snmp-server enable traps cpu threshold
snmp-server enable traps rsvp
snmp-server enable traps syslog
snmp-server enable traps l2tun session
snmp-server enable traps l2tun pseudowire status
snmp-server enable traps vtp
snmp-server enable traps waas
snmp-server enable traps pki
snmp-server enable traps ethernet cfm cc mep-up mep-down cross-connect loop config
snmp-server enable traps ethernet cfm crosscheck mep-missing mep-unknown service-up
snmp-server enable traps ethernet evc status create delete
snmp-server enable traps bstun
snmp-server enable traps dlsr
snmp-server enable traps ipsla
snmp-server enable traps stun
snmp-server enable traps bfd
snmp-server enable traps mpls traffic-eng
snmp-server enable traps mpls fast-reroute protected
snmp-server enable traps mpls rfc ldp
snmp-server enable traps mpls ldp
snmp-server enable traps pw vc
snmp-server enable traps lisp
snmp-server enable traps ipmobile
snmp-server enable traps snasw alert isr topology cp-cp port link dlus
snmp-server enable traps dial
snmp-server enable traps dsp card-status
snmp-server enable traps dsp oper-state
snmp-server enable traps dsp video-usage
snmp-server enable traps dsp video-out-of-resource
snmp-server enable traps gdoi gm-start-registration
snmp-server enable traps gdoi gm-registration-complete
snmp-server enable traps gdoi gm-re-register
snmp-server enable traps gdoi gm-rekey-rcvd
snmp-server enable traps gdoi gm-rekey-fail
snmp-server enable traps gdoi ks-rekey-pushed
snmp-server enable traps gdoi gm-incomplete-cfg
snmp-server enable traps gdoi ks-no-rsa-keys
snmp-server enable traps gdoi ks-new-registration
snmp-server enable traps gdoi ks-reg-complete
snmp-server enable traps gdoi ks-role-change
snmp-server enable traps gdoi ks-gm-deleted
snmp-server enable traps gdoi ks-peer-reachable
snmp-server enable traps gdoi ks-peer-unreachable
snmp-server enable traps firewall serverstatus
snmp-server enable traps ike policy add
snmp-server enable traps ike policy delete
snmp-server enable traps ike tunnel start
snmp-server enable traps ike tunnel stop
snmp-server enable traps ipsec cryptomap add
snmp-server enable traps ipsec cryptomap delete
snmp-server enable traps ipsec cryptomap attach
snmp-server enable traps ipsec cryptomap detach
snmp-server enable traps ipsec tunnel start
snmp-server enable traps ipsec tunnel stop
snmp-server enable traps ipsec too-many-sas
snmp-server enable traps rf

```



```

snmp-server enable traps bulkstat collection transfer
snmp-server enable traps vrfmib vrf-up vrf-down vnet-trunk-up vnet-trunk-down
snmp-server enable traps ethernet cfm alarm
snmp-server enable traps mpls vpn
snmp-server enable traps ccme
snmp-server enable traps srst
snmp-server enable traps voice
snmp-server enable traps dnis
snmp-server host 10.3.99.1 supervision
access-list 1 permit 10.3.254.4 0.0.0.3
!
control-plane
!
!
!
!
!
!
mgcp behavior rsip-range tgcp-only
mgcp behavior comedia-role none
mgcp behavior comedia-check-media-src disable
mgcp behavior comedia-sdp-force disable
!
mgcp profile default
!
!
!
!
!
!
!
gatekeeper
shutdown
!
!
vstack
!
line con 0
line aux 0
line 2
no activation-character
no exec
transport preferred none
transport output lat pad telnet rlogin lapb-ta mop udptn v120 ssh
stopbits 1
line vty 0 4
login
transport input none
!
scheduler allocate 20000 1000
!
end

```

### 8.3. Configuration routeur EDGE2

```

EDGE2#sh run
Building configuration...

Current configuration : 5971 bytes

```

```
!  
! Last configuration change at 10:01:03 UTC Wed Dec 20 2023  
!  
version 15.4  
service timestamps debug datetime msec  
service timestamps log datetime msec  
no service password-encryption  
!  
hostname EDGE2  
!  
boot-start-marker  
boot-end-marker  
!  
!  
enable secret 5 $1$Kzct$zpfdB8Z2dvvc2aOc/SCAJ0  
!  
no aaa new-model  
!  
!  
!  
!  
!  
!  
!  
!  
!  
!  
!  
!  
!  
ip cef  
no ipv6 cef  
multilink bundle-name authenticated  
!  
!  
cts logging verbose  
!  
!  
license udi pid CISCO2901/K9 sn FCZ194660EF  
!  
!  
!  
redundancy  
!  
!  
!  
!  
!  
!  
interface Embedded-Service-Engine0/0  
no ip address  
shutdown  
!  
interface GigabitEthernet0/0  
no ip address  
duplex auto  
speed auto  
!  
interface GigabitEthernet0/0.30
```

```

encapsulation dot1Q 30
ip address 217.3.160.251 255.255.255.0
standby 1 ip 217.3.160.249
standby 1 preempt
!
interface GigabitEthernet0/0.200
encapsulation dot1Q 200
ip address 10.3.254.17 255.255.255.252
!
interface GigabitEthernet0/0.300
encapsulation dot1Q 300
ip address 10.3.254.22 255.255.255.252
!
interface GigabitEthernet0/1
ip address 10.3.254.10 255.255.255.252
ip nat inside
ip virtual-reassembly in
duplex auto
speed auto
!
interface GigabitEthernet0/1.20
encapsulation dot1Q 20
ip address 31.3.126.14 255.255.255.252
ip nat outside
ip virtual-reassembly in
!
router ospf 10
redistribute connected subnets
redistribute static subnets
redistribute bgp 65316 subnets
network 10.3.254.8 0.0.0.3 area 3
network 10.3.254.16 0.0.0.3 area 3
network 10.3.254.20 0.0.0.3 area 3
!
router bgp 65316
bgp log-neighbor-changes
neighbor 31.3.126.13 remote-as 65330
!
address-family ipv4
network 31.3.126.12 mask 255.255.255.252
network 217.3.160.0 mask 255.255.255.252
redistribute connected
redistribute static
redistribute ospf 10
neighbor 31.3.126.13 activate
exit-address-family
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
ip nat inside source list 1 interface GigabitEthernet0/1.20 overload
ip nat inside source static tcp 10.3.10.2 80 191.5.157.33 80 extendable
ip nat inside source static tcp 10.3.10.2 443 191.5.157.33 443 extendable
ip nat inside source static tcp 10.3.10.62 4443 191.5.157.33 4443 extendable
ip route 10.3.0.0 255.255.0.0 10.3.254.18
!
!
!

```

```

snmp-server community supervision RO
snmp-server enable traps snmp authentication linkdown linkup coldstart warmstart
snmp-server enable traps vrrp
snmp-server enable traps transceiver all
snmp-server enable traps ds1
snmp-server enable traps call-home message-send-fail server-fail
snmp-server enable traps tty
snmp-server enable traps eigrp
snmp-server enable traps ospf state-change
snmp-server enable traps ospf errors
snmp-server enable traps ospf retransmit
snmp-server enable traps ospf lsa
snmp-server enable traps ospf cisco-specific state-change nssa-trans-change
snmp-server enable traps ospf cisco-specific state-change shamlink interface
snmp-server enable traps ospf cisco-specific state-change shamlink neighbor
snmp-server enable traps ospf cisco-specific errors
snmp-server enable traps ospf cisco-specific retransmit
snmp-server enable traps ospf cisco-specific lsa
snmp-server enable traps license
snmp-server enable traps envmon
snmp-server enable traps flash insertion removal low-space
snmp-server enable traps auth-framework sec-violation auth-fail
snmp-server enable traps c3g
snmp-server enable traps ds3
snmp-server enable traps adslline
snmp-server enable traps vdsl2line
snmp-server enable traps icsudsu
snmp-server enable traps isdn call-information
snmp-server enable traps isdn layer2
snmp-server enable traps isdn chan-not-avail
snmp-server enable traps isdn ietf
snmp-server enable traps ds0-busyout
snmp-server enable traps ds1-loopback
snmp-server enable traps energywise
snmp-server enable traps vstack
snmp-server enable traps mac-notification
snmp-server enable traps trustsec-sxp conn-srcaddr-err msg-parse-err conn-config-err binding-err
conn-up conn-down binding-expn-fail oper-nodeid-change binding-conflict
snmp-server enable traps bgp
snmp-server enable traps bgp cbgp2
snmp-server enable traps isis
snmp-server enable traps ospfv3 state-change
snmp-server enable traps ospfv3 errors
snmp-server enable traps aaa_server
snmp-server enable traps atm subif
snmp-server enable traps cef resource-failure peer-state-change peer-fib-state-change
inconsistency
snmp-server enable traps memory bufferpeak
snmp-server enable traps cnpd
snmp-server enable traps config-copy
snmp-server enable traps config
snmp-server enable traps config-ctid
snmp-server enable traps entity-ext
snmp-server enable traps entity
snmp-server enable traps fru-ctrl
snmp-server enable traps resource-policy
snmp-server enable traps event-manager
snmp-server enable traps frame-relay multilink bundle-mismatch
snmp-server enable traps frame-relay
snmp-server enable traps frame-relay subif

```

```

snmp-server enable traps hsrp
snmp-server enable traps ipmulticast
snmp-server enable traps mempool
snmp-server enable traps msdp
snmp-server enable traps mvpn
snmp-server enable traps nhrp nhs
snmp-server enable traps nhrp nhc
snmp-server enable traps nhrp nhp
snmp-server enable traps nhrp quota-exceeded
snmp-server enable traps pim neighbor-change rp-mapping-change invalid-pim-message
snmp-server enable traps pppoe
snmp-server enable traps cpu threshold
snmp-server enable traps rsvp
snmp-server enable traps syslog
snmp-server enable traps l2tun session
snmp-server enable traps l2tun pseudowire status
snmp-server enable traps vtp
snmp-server enable traps waas
snmp-server enable traps rf
snmp-server enable traps bulkstat collection transfer
snmp-server enable traps vrfmib vrf-up vrf-down vnet-trunk-up vnet-trunk-down
snmp-server host 10.3.99.1 supervision
access-list 1 permit 10.3.254.8 0.0.0.3
!
control-plane
!
!
!
line con 0
line aux 0
line 2
  no activation-character
  no exec
  transport preferred none
  transport output pad telnet rlogin lapb-ta mop udptn v120 ssh
  stopbits 1
line vty 0 4
  login
  transport input none
!
scheduler allocate 20000 1000
!
end

```

#### 8.4. Configuration CORESW1

```

CORESW1#sh run
Building configuration...

Current configuration : 8293 bytes
!
version 12.2
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname CORESW1
!
boot-start-marker
boot-end-marker

```

```

!
enable secret 5 $1$2w9G$eLd7k6.moo4PZATxGKtzo0
!
!
!
no aaa new-model
switch 1 provision ws-c3750v2-24ts
system mtu routing 1536
ip routing
!
!
!
!
crypto pki trustpoint TP-self-signed-896748032
enrollment selfsigned
subject-name cn=IOS-Self-Signed-Certificate-896748032
revocation-check none
rsakeypair TP-self-signed-896748032
!
!
crypto pki certificate chain TP-self-signed-896748032
certificate self-signed 01
  3082023E 308201A7 A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  30312E30 2C060355 04031325 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 38393637 34383033 32301E17 0D393330 33303130 30303134
  385A170D 32303031 30313030 30303030 5A303031 2E302C06 03550403 1325494F
  532D5365 6C662D53 69676E65 642D4365 72746966 69636174 652D3839 36373438
  30333230 819F300D 06092A86 4886F70D 01010105 0003818D 00308189 02818100
  BB5C0E4C E1F019FC EA45A5A6 3E5E04F9 B0F150A1 DA6824FE E53AF3DD F75FDBD3
  1030EBF5 A33CFA05 BDD85FDF B9E65AA3 9ECF3BB1 9B818023 E4DF2DE9 4BE33E20
  186BA80C 4D95481E A5F635E7 5E5C8DC9 7EEBB1E0 6DE9958E 726DBACB F082ED39
  530B0F31 B8CAC2AF AC153511 CEDED149 4792AD64 9E82AD10 F2DAB0F9 95DCBAA7
  02030100 01A36830 66300F06 03551D13 0101FF04 05300301 01FF3013 0603551D
  11040C30 0A820843 4F524553 57312E30 1F060355 1D230418 30168014 347E2388
  75DA2FAE B83496E9 79295E55 9E7C13A6 301D0603 551D0E04 16041434 7E238875
  DA2FAEB8 3496E979 295E559E 7C13A630 0D06092A 864886F7 0D010104 05000381
  8100A784 47CE61FB C551EE1F C270C31C D179E4E2 E5D6B50B D5395890 219BE013
  BBA93352 6BC82974 320E1FAB 8C1C99A8 6B0EE06B 643392A2 245222F3 9557E39E
  20950257 D2ADCF65 83403858 0F3B65E5 EB521FD5 22EB1B90 0A43D52D 24217DE2
  989ECC47 2B2989BD CF524D9A C3893F0E E879EB69 2C7C4514 8937322A B8749A52 E23B
quit
!
!
!
!
spanning-tree mode rapid-pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
!
!
!
interface Port-channel1
switchport trunk encapsulation dot1q
switchport trunk allowed vlan 10,20,30,99,300,666
switchport mode trunk
switchport nonegotiate
!
interface FastEthernet1/0/1

```

```

switchport trunk encapsulation dot1q
switchport trunk allowed vlan 10,20,30,99,666
switchport mode trunk
switchport nonegotiate
!
interface FastEthernet1/0/2
switchport trunk encapsulation dot1q
switchport trunk allowed vlan 10,20,30,99,300,666
switchport mode trunk
switchport nonegotiate
channel-group 1 mode active
!
interface FastEthernet1/0/3
switchport trunk encapsulation dot1q
switchport trunk allowed vlan 10,20,30,99,300,666
switchport mode trunk
switchport nonegotiate
channel-group 1 mode active
!
interface FastEthernet1/0/4
switchport access vlan 666
switchport mode access
shutdown
!
interface FastEthernet1/0/5
switchport access vlan 666
switchport mode access
shutdown
!
interface FastEthernet1/0/6
switchport access vlan 666
switchport mode access
shutdown
!
interface FastEthernet1/0/7
switchport access vlan 666
switchport mode access
shutdown
!
interface FastEthernet1/0/8
switchport access vlan 666
switchport mode access
shutdown
!
interface FastEthernet1/0/9
switchport access vlan 666
switchport mode access
shutdown
!
interface FastEthernet1/0/10
switchport access vlan 666
switchport mode access
shutdown
!
interface FastEthernet1/0/11
switchport access vlan 666
switchport mode access
shutdown
!
interface FastEthernet1/0/12

```



```
switchport access vlan 666
switchport mode access
shutdown
!
interface FastEthernet1/0/13
switchport access vlan 666
switchport mode access
shutdown
!
interface FastEthernet1/0/14
switchport access vlan 666
switchport mode access
shutdown
!
interface FastEthernet1/0/15
switchport access vlan 666
switchport mode access
shutdown
!
interface FastEthernet1/0/16
switchport access vlan 666
switchport mode access
shutdown
!
interface FastEthernet1/0/17
switchport access vlan 666
switchport mode access
shutdown
!
interface FastEthernet1/0/18
switchport access vlan 666
switchport mode access
shutdown
!
interface FastEthernet1/0/19
switchport access vlan 666
switchport mode access
shutdown
!
interface FastEthernet1/0/20
switchport access vlan 666
switchport mode access
shutdown
!
interface FastEthernet1/0/21
switchport access vlan 666
switchport mode access
shutdown
!
interface FastEthernet1/0/22
switchport access vlan 666
switchport mode access
shutdown
!
interface FastEthernet1/0/23
switchport trunk encapsulation dot1q
switchport trunk allowed vlan 30,100,300
switchport mode trunk
switchport nonegotiate
!
```

```

interface FastEthernet1/0/24
switchport trunk encapsulation dot1q
switchport trunk allowed vlan 10,20,30,99
switchport mode trunk
switchport nonegotiate
!
interface GigabitEthernet1/0/1
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/2
switchport access vlan 666
switchport mode access
shutdown
!
interface Vlan1
no ip address
!
interface Vlan10
ip address 10.3.10.60 255.255.255.192
standby 1 ip 10.3.10.62
standby 1 priority 110
standby 1 preempt
!
interface Vlan20
ip address 10.3.20.252 255.255.252.0
standby 2 ip 10.3.20.254
standby 2 priority 110
standby 2 preempt
!
interface Vlan30
ip address 217.3.160.252 255.255.255.0
standby 3 ip 217.3.160.254
standby 3 priority 110
standby 3 preempt
!
interface Vlan99
ip address 10.3.99.252 255.255.255.0
standby 9 ip 10.3.99.254
standby 9 priority 110
standby 9 preempt
!
interface Vlan100
ip address 10.3.254.2 255.255.255.252
!
ip default-gateway 10.3.254.1
ip classless
ip route 0.0.0.0 0.0.0.0 10.3.254.1
ip http server
ip http secure-server
!
!
ip sla enable reaction-alerts
!
snmp-server community supervision RO
snmp-server enable traps snmp authentication linkdown linkup coldstart warmstart
snmp-server enable traps transceiver all
snmp-server enable traps tty
snmp-server enable traps eigrp

```

```

snmp-server enable traps ospf state-change
snmp-server enable traps ospf errors
snmp-server enable traps ospf retransmit
snmp-server enable traps ospf lsa
snmp-server enable traps ospf cisco-specific state-change nssa-trans-change
snmp-server enable traps ospf cisco-specific state-change shamlink interface-old
snmp-server enable traps ospf cisco-specific state-change shamlink neighbor
snmp-server enable traps ospf cisco-specific errors
snmp-server enable traps ospf cisco-specific retransmit
snmp-server enable traps ospf cisco-specific lsa
snmp-server enable traps cluster
snmp-server enable traps fru-ctrl
snmp-server enable traps entity
snmp-server enable traps cpu threshold
snmp-server enable traps power-ethernet group 1-9
snmp-server enable traps power-ethernet police
snmp-server enable traps vtp
snmp-server enable traps vlancreate
snmp-server enable traps vlandelete
snmp-server enable traps flash insertion removal
snmp-server enable traps port-security
snmp-server enable traps auth-framework sec-violation
snmp-server enable traps dot1x auth-fail-vlan guest-vlan no-auth-fail-vlan no-guest-vlan
snmp-server enable traps envmon fan shutdown supply temperature status
snmp-server enable traps stackwise
snmp-server enable traps license
snmp-server enable traps bgp
snmp-server enable traps cef resource-failure peer-state-change peer-fib-state-change
inconsistency
snmp-server enable traps config-copy
snmp-server enable traps config
snmp-server enable traps config-ctid
snmp-server enable traps event-manager
snmp-server enable traps hsrp
snmp-server enable traps ipmulticast
snmp-server enable traps isis
snmp-server enable traps msdp
snmp-server enable traps pim neighbor-change rp-mapping-change invalid-pim-message
snmp-server enable traps energywise
snmp-server enable traps vstack
snmp-server enable traps bridge newroot topologychange
snmp-server enable traps stpx inconsistency root-inconsistency loop-inconsistency
snmp-server enable traps syslog
snmp-server enable traps rtr
snmp-server enable traps mac-notification change move threshold
snmp-server enable traps vlan-membership
snmp-server enable traps errdisable
snmp-server host 10.3.99.1 supervision
!
!
line con 0
line vty 0 4
  login
line vty 5 15
  login
!
end

```

## 8.5. Configuration CORESW2

```
CORESW2#sh run
Building configuration...

Current configuration : 6527 bytes
!
version 12.2
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname CORESW2
!
boot-start-marker
boot-end-marker
!
enable secret 5 $1$.p0T$X91MCGCpQX33eUYXFKk/C0
!
!
!
no aaa new-model
switch 1 provision ws-c3750v2-24ts
system mtu routing 1536
ip routing
!
!
!
!
!
!
!
!
!
spanning-tree mode rapid-pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
!
!
!
interface Port-channel1
switchport trunk encapsulation dot1q
switchport trunk allowed vlan 10,20,30,99,300,666
switchport mode trunk
switchport nonegotiate
!
interface FastEthernet1/0/1
switchport trunk encapsulation dot1q
switchport trunk allowed vlan 10,20,30,99,666
switchport mode trunk
switchport nonegotiate
!
interface FastEthernet1/0/2
switchport trunk encapsulation dot1q
switchport trunk allowed vlan 10,20,30,99,300,666
switchport mode trunk
switchport nonegotiate
channel-group 1 mode active
```

```
!  
interface FastEthernet1/0/3  
  switchport trunk encapsulation dot1q  
  switchport trunk allowed vlan 10,20,30,99,300,666  
  switchport mode trunk  
  switchport nonegotiate  
  channel-group 1 mode active  
!  
interface FastEthernet1/0/4  
  switchport access vlan 666  
  switchport mode access  
  shutdown  
!  
interface FastEthernet1/0/5  
  switchport access vlan 666  
  switchport mode access  
  shutdown  
!  
interface FastEthernet1/0/6  
  switchport access vlan 666  
  switchport mode access  
  shutdown  
!  
interface FastEthernet1/0/7  
  switchport access vlan 666  
  switchport mode access  
  shutdown  
!  
interface FastEthernet1/0/8  
  switchport access vlan 666  
  switchport mode access  
  shutdown  
!  
interface FastEthernet1/0/9  
  switchport access vlan 666  
  switchport mode access  
  shutdown  
!  
interface FastEthernet1/0/10  
  switchport access vlan 666  
  switchport mode access  
  shutdown  
!  
interface FastEthernet1/0/11  
  switchport access vlan 666  
  switchport mode access  
  shutdown  
!  
interface FastEthernet1/0/12  
  switchport access vlan 666  
  switchport mode access  
  shutdown  
!  
interface FastEthernet1/0/13  
  switchport access vlan 666  
  switchport mode access  
  shutdown  
!  
interface FastEthernet1/0/14  
  switchport access vlan 666
```

```
switchport mode access
shutdown
!
interface FastEthernet1/0/15
switchport access vlan 666
switchport mode access
shutdown
!
interface FastEthernet1/0/16
switchport access vlan 666
switchport mode access
shutdown
!
interface FastEthernet1/0/17
switchport access vlan 666
switchport mode access
shutdown
!
interface FastEthernet1/0/18
switchport access vlan 666
switchport mode access
shutdown
!
interface FastEthernet1/0/19
switchport access vlan 666
switchport mode access
shutdown
!
interface FastEthernet1/0/20
switchport access vlan 666
switchport mode access
shutdown
!
interface FastEthernet1/0/21
switchport access vlan 666
switchport mode access
shutdown
!
interface FastEthernet1/0/22
switchport access vlan 666
switchport mode access
shutdown
!
interface FastEthernet1/0/23
switchport trunk encapsulation dot1q
switchport trunk allowed vlan 30,200,300
switchport mode trunk
switchport nonegotiate
!
interface FastEthernet1/0/24
switchport trunk encapsulation dot1q
switchport trunk allowed vlan 10,20,30,99,666
switchport mode trunk
switchport nonegotiate
!
interface GigabitEthernet1/0/1
switchport access vlan 666
switchport mode access
shutdown
!
```

```

interface GigabitEthernet1/0/2
switchport access vlan 666
switchport mode access
shutdown
!
interface Vlan1
no ip address
shutdown
!
interface Vlan10
ip address 10.3.10.61 255.255.255.192
standby 1 ip 10.3.10.62
standby 1 preempt
!
interface Vlan20
ip address 10.3.20.253 255.255.252.0
standby 2 ip 10.3.20.254
standby 2 preempt
!
interface Vlan30
ip address 217.3.160.253 255.255.255.0
standby 3 ip 217.3.160.254
standby 3 preempt
!
interface Vlan99
ip address 10.3.99.253 255.255.255.0
standby 9 ip 10.3.99.254
standby 9 preempt
!
interface Vlan200
ip address 10.3.254.18 255.255.255.252
!
ip default-gateway 10.3.254.17
ip classless
ip route 0.0.0.0 0.0.0.0 10.3.254.17
ip http server
ip http secure-server
!
!
!
snmp-server community supervision RO
snmp-server enable traps snmp authentication linkdown linkup coldstart warmstart
snmp-server enable traps transceiver all
snmp-server enable traps tty
snmp-server enable traps eigrp
snmp-server enable traps ospf state-change
snmp-server enable traps ospf errors
snmp-server enable traps ospf retransmit
snmp-server enable traps ospf lsa
snmp-server enable traps ospf cisco-specific state-change nssa-trans-change
snmp-server enable traps ospf cisco-specific state-change shamlink interface-old
snmp-server enable traps ospf cisco-specific state-change shamlink neighbor
snmp-server enable traps ospf cisco-specific errors
snmp-server enable traps ospf cisco-specific retransmit
snmp-server enable traps ospf cisco-specific lsa
snmp-server enable traps cluster
snmp-server enable traps fru-ctrl
snmp-server enable traps entity
snmp-server enable traps cpu threshold
snmp-server enable traps power-ethernet group 1-9

```



```

snmp-server enable traps power-ethernet police
snmp-server enable traps vtp
snmp-server enable traps vlancreate
snmp-server enable traps vlandelete
snmp-server enable traps flash insertion removal
snmp-server enable traps port-security
snmp-server enable traps auth-framework sec-violation
snmp-server enable traps dot1x auth-fail-vlan guest-vlan no-auth-fail-vlan no-guest-vlan
snmp-server enable traps envmon fan shutdown supply temperature status
snmp-server enable traps stackwise
snmp-server enable traps license
snmp-server enable traps bgp
snmp-server enable traps cef resource-failure peer-state-change peer-fib-state-change
inconsistency
snmp-server enable traps config-copy
snmp-server enable traps config
snmp-server enable traps config-ctid
snmp-server enable traps event-manager
snmp-server enable traps hsrp
snmp-server enable traps ipmulticast
snmp-server enable traps isis
snmp-server enable traps msdp
snmp-server enable traps pim neighbor-change rp-mapping-change invalid-pim-message
snmp-server enable traps energywise
snmp-server enable traps vstack
snmp-server enable traps bridge newroot topologychange
snmp-server enable traps stpx inconsistency root-inconsistency loop-inconsistency
snmp-server enable traps syslog
snmp-server enable traps rtr
snmp-server enable traps mac-notification change move threshold
snmp-server enable traps vlan-membership
snmp-server enable traps errdisable
snmp-server host 10.3.99.1 supervision
!
!
line con 0
line vty 5 15
!
end

```

## 8.6. Configuration ACCSW1

```

ACCSW1#sh run
Building configuration...

Current configuration : 7999 bytes
!
! Last configuration change at 08:26:58 UTC Mon Apr 4 2011
! NVRAM config last updated at 10:06:08 UTC Mon Apr 4 2011
!
version 15.0
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname ACCSW1
!
boot-start-marker
boot-end-marker

```

```
enable secret 4 g1rTD89b38NIXbGJse.zLc7Cega1TBTIKQNvYDh9Qo6
!
no aaa new-model
switch 1 provision ws-c2960s-48fpd-l
!
!
!
!
!
spanning-tree mode pvst
spanning-tree extend system-id
!
!
!
!
!
!
!
!
vlan internal allocation policy ascending
!
!
!
!
!
!
!
!
!
interface FastEthernet0
 no ip address
!
interface GigabitEthernet1/0/1
 switchport trunk allowed vlan 10,20,30,99
 switchport mode trunk
 switchport nonegotiate
!
interface GigabitEthernet1/0/2
 switchport access vlan 666
 switchport mode access
 shutdown
!
interface GigabitEthernet1/0/3
 switchport access vlan 10
 switchport mode access
 switchport port-security maximum 3
 switchport port-security
!
interface GigabitEthernet1/0/4
 switchport access vlan 20
 switchport mode access
 switchport port-security maximum 3
 switchport port-security
!
interface GigabitEthernet1/0/5
 switchport access vlan 30
```

```
switchport mode access
switchport port-security maximum 3
switchport port-security
!
interface GigabitEthernet1/0/6
switchport access vlan 99
switchport mode access
switchport port-security maximum 3
switchport port-security
!
interface GigabitEthernet1/0/7
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/8
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/9
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/10
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/11
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/12
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/13
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/14
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/15
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/16
switchport access vlan 666
switchport mode access
shutdown
!
```

```
interface GigabitEthernet1/0/17
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/18
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/19
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/20
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/21
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/22
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/23
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/24
switchport trunk allowed vlan 10,20,30,99
switchport mode trunk
switchport nonegotiate
!
interface GigabitEthernet1/0/25
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/26
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/27
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/28
switchport access vlan 666
switchport mode access
shutdown
!
```

```
interface GigabitEthernet1/0/29
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/30
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/31
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/32
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/33
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/34
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/35
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/36
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/37
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/38
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/39
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/40
switchport access vlan 666
switchport mode access
shutdown
!
```

```
interface GigabitEthernet1/0/41
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/42
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/43
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/44
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/45
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/46
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/47
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/48
switchport trunk allowed vlan 10,20,99
switchport mode trunk
switchport nonegotiate
!
interface GigabitEthernet1/0/49
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet1/0/50
switchport access vlan 666
switchport mode access
shutdown
!
interface TenGigabitEthernet1/0/1
!
interface TenGigabitEthernet1/0/2
!
interface Vlan1
no ip address
!
interface Vlan99
ip address 10.3.99.241 255.255.255.0
!
```

```

ip http server
ip http secure-server
!
!
snmp-server community supervision RO
snmp-server enable traps snmp authentication linkdown linkup coldstart warmstart
snmp-server enable traps transceiver all
snmp-server enable traps call-home message-send-fail server-fail
snmp-server enable traps tty
snmp-server enable traps license
snmp-server enable traps auth-framework sec-violation
snmp-server enable traps cluster
snmp-server enable traps config-copy
snmp-server enable traps config
snmp-server enable traps config-ctid
snmp-server enable traps dot1x auth-fail-vlan guest-vlan no-auth-fail-vlan no-guest-vlan
snmp-server enable traps energywise
snmp-server enable traps fru-ctrl
snmp-server enable traps entity
snmp-server enable traps event-manager
snmp-server enable traps ike policy add
snmp-server enable traps ike policy delete
snmp-server enable traps ike tunnel start
snmp-server enable traps ike tunnel stop
snmp-server enable traps ipsec cryptomap add
snmp-server enable traps ipsec cryptomap delete
snmp-server enable traps ipsec cryptomap attach
snmp-server enable traps ipsec cryptomap detach
snmp-server enable traps ipsec tunnel start
snmp-server enable traps ipsec tunnel stop
snmp-server enable traps ipsec too-many-sas
snmp-server enable traps power-ethernet group 1
snmp-server enable traps power-ethernet police
snmp-server enable traps cpu threshold
snmp-server enable traps rep
snmp-server enable traps ipsla
snmp-server enable traps vstack
snmp-server enable traps bridge newroot topologychange
snmp-server enable traps stpx inconsistency root-inconsistency loop-inconsistency
snmp-server enable traps syslog
snmp-server enable traps vtp
snmp-server enable traps vlancreate
snmp-server enable traps vlandelete
snmp-server enable traps flash insertion removal
snmp-server enable traps port-security
snmp-server enable traps envmon fan shutdown supply temperature status
snmp-server enable traps stackwise
snmp-server enable traps errdisable
snmp-server enable traps mac-notification change move threshold
snmp-server enable traps vlan-membership
snmp-server host 10.3.99.1 supervision
!
!
line con 0
line vty 5 15
!
end

```



## 8.7. Configuration ACCSW2

```
ACCSW2#sh run
Building configuration...

Current configuration : 8035 bytes
!
! Last configuration change at 06:32:16 UTC Sat Jan 7 2006
!
version 15.2
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname ACCSW2
!
boot-start-marker
boot-end-marker
!
enable secret 5 $1$hbja$LR7eOVIMx4.5CZYKy79C50
!
no aaa new-model
switch 2 provision ws-c2960s-48fpd-l
!
!
!
!
!
!
!
!
!
!
spanning-tree mode pvst
spanning-tree extend system-id
!
!
!
!
vlan internal allocation policy ascending
!
!
!
!
!
!
!
!
!
interface FastEthernet0
 no ip address
!
interface GigabitEthernet2/0/1
 switchport trunk allowed vlan 10,20,30,99
 switchport mode trunk
 switchport nonegotiate
!
```

```
interface GigabitEthernet2/0/2
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/3
switchport access vlan 10
switchport mode access
switchport port-security maximum 3
switchport port-security
!
interface GigabitEthernet2/0/4
switchport access vlan 20
switchport mode access
switchport port-security maximum 3
switchport port-security
!
interface GigabitEthernet2/0/5
switchport access vlan 30
switchport mode access
switchport port-security maximum 3
switchport port-security
!
interface GigabitEthernet2/0/6
switchport access vlan 99
switchport mode access
switchport port-security maximum 3
switchport port-security
!
interface GigabitEthernet2/0/7
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/8
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/9
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/10
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/11
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/12
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/13
```

```
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/14
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/15
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/16
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/17
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/18
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/19
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/20
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/21
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/22
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/23
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/24
switchport trunk allowed vlan 10,20,30,99
switchport mode trunk
switchport nonegotiate
!
interface GigabitEthernet2/0/25
```

```
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/26
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/27
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/28
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/29
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/30
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/31
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/32
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/33
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/34
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/35
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/36
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/37
```

```
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/38
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/39
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/40
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/41
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/42
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/43
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/44
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/45
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/46
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/47
switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/48
switchport trunk allowed vlan 10,20,30,99
switchport mode trunk
switchport nonegotiate
!
interface GigabitEthernet2/0/49
```

```

switchport access vlan 666
switchport mode access
shutdown
!
interface GigabitEthernet2/0/50
switchport access vlan 666
switchport mode access
shutdown
!
interface TenGigabitEthernet2/0/1
!
interface TenGigabitEthernet2/0/2
!
interface Vlan1
no ip address
!
interface Vlan99
ip address 10.3.99.242 255.255.255.0
!
ip http server
ip http secure-server
!
!
!
snmp-server community supervision RO
snmp-server enable traps snmp authentication linkdown linkup coldstart warmstart
snmp-server enable traps transceiver all
snmp-server enable traps call-home message-send-fail server-fail
snmp-server enable traps tty
snmp-server enable traps license
snmp-server enable traps auth-framework sec-violation
snmp-server enable traps cluster
snmp-server enable traps config-copy
snmp-server enable traps config
snmp-server enable traps config-ctid
snmp-server enable traps trustsec-sxp conn-srcaddr-err msg-parse-err conn-config-err binding-err
conn-up conn-down binding-expn-fail oper-nodeid-change binding-conflict
snmp-server enable traps energywise
snmp-server enable traps fru-ctrl
snmp-server enable traps entity
snmp-server enable traps event-manager
snmp-server enable traps ike policy add
snmp-server enable traps ike policy delete
snmp-server enable traps ike tunnel start
snmp-server enable traps ike tunnel stop
snmp-server enable traps ipsec cryptomap add
snmp-server enable traps ipsec cryptomap delete
snmp-server enable traps ipsec cryptomap attach
snmp-server enable traps ipsec cryptomap detach
snmp-server enable traps ipsec tunnel start
snmp-server enable traps ipsec tunnel stop
snmp-server enable traps ipsec too-many-sas
snmp-server enable traps power-ethernet group 2
snmp-server enable traps power-ethernet police
snmp-server enable traps cpu threshold
snmp-server enable traps rep
snmp-server enable traps vstack
snmp-server enable traps bridge newroot topologychange
snmp-server enable traps stpx inconsistency root-inconsistency loop-inconsistency
snmp-server enable traps syslog

```

```
snmp-server enable traps vtp
snmp-server enable traps vlancreate
snmp-server enable traps vlandelete
snmp-server enable traps flash insertion removal
snmp-server enable traps port-security
snmp-server enable traps envmon fan shutdown supply temperature status
snmp-server enable traps stackwise
snmp-server enable traps bulkstat collection transfer
snmp-server enable traps errdisable
snmp-server enable traps mac-notification change move threshold
snmp-server enable traps vlan-membership
snmp-server host 10.3.99.1 supervision
!
!
line con 0
line vty 5 15
!
end
```