

Routeurs Riders

123 rue des champs
25200 Montbéliard
(+33) 03 12 12 13 15

Rapport de la SAE 502

Vue d'ensemble

La clinique de campagne a récemment entrepris des démarches pour refaire son infrastructure réseau dans le but de renforcer sa connectivité et d'établir une liaison fluide avec l'hôpital B, situé dans la région. Cette initiative est motivée par la nécessité d'améliorer la qualité des soins de santé offerts aux résidents locaux, en permettant un partage rapide et sécurisé d'informations médicales cruciales entre les deux établissements.

Objectifs

L'objectif principal de ce projet est de moderniser et de renforcer l'infrastructure réseau de la clinique de campagne en mettant en place un système performant et sécurisé qui facilite la transmission efficace des données médicales et administratives vers l'hôpital B. Le projet vise également à garantir une connectivité fiable et stable pour améliorer la coordination des soins et des services entre les deux établissements de santé.

Délai

Le projet doit être mené à bien dans un délai court afin de minimiser les perturbations opérationnelles au sein de la clinique. Une approche stratégique et efficace sera adoptée pour garantir la réalisation des objectifs dans les délais impartis, tout en assurant la qualité et la fiabilité des résultats finaux.

Cloud du Projet

Nos fichiers des codes du site web, la bdd ainsi que toute notre documentation est disponible sur ce lien github : <https://github.com/Yasko0x000/Sae502-Piloter-un-projet-informatique>

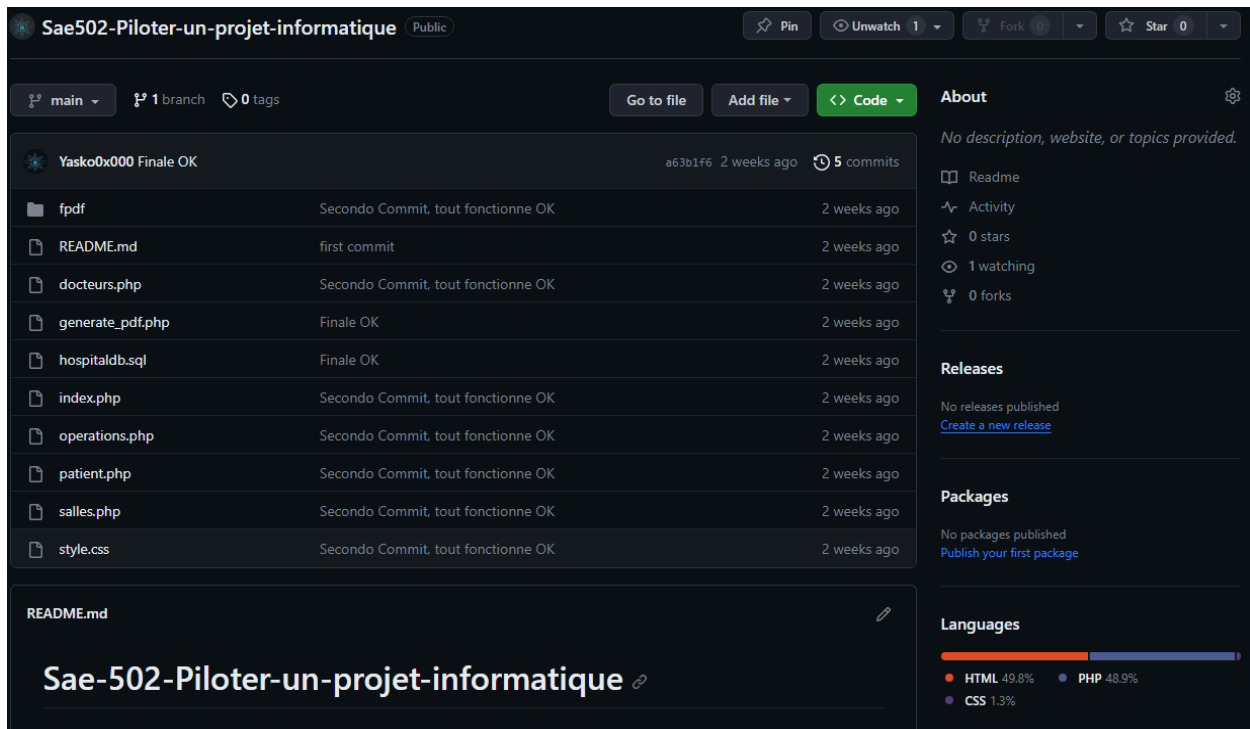


Figure 1 : Page Cloud Github

Grandes étapes

1. Pour la base de données, nous avons mis en place une base de données robuste et sécurisée en utilisant PHPMyAdmin et SQL. Cette base de données gère efficacement les informations médicales et administratives. Elle stocke les données des patients, des médecins, des opérations, des salles d'opération et des badges RFID. Les tables de la base de données sont liées pour permettre une récupération rapide et précise des données. Des requêtes SQL sont utilisées pour extraire des informations pertinentes, telles que la recherche de patients associés à une opération donnée, la liste des médecins et de leurs patients associés, et la salle d'opération associée à une opération en cours.
2. Développement d'un site Web : nous avons développé une interface utilisateur conviviale avec une mise en page claire et des formulaires de recherche intuitifs. Les utilisateurs peuvent rechercher des informations sur les patients, les médecins, les opérations et les salles d'opération. Les résultats sont affichés de manière structurée et lisible dans des tables. Nous avons également intégré des fonctionnalités interactives telles que la validation des formulaires et la génération de rapports PDF pour les listes de patients, médecins, opérations et historiques de salle d'opération.
3. Concernant le VPN, nous avons choisi d'utiliser pfSense pour établir un tunnel sécurisé entre la clinique de campagne et l'hôpital B. Cela permet une communication sécurisée des données sensibles entre les deux établissements, garantissant que les informations médicales et administratives restent confidentielles. Le VPN assure également une connectivité fiable et stable, renforçant ainsi la coordination des soins et des services.
4. Configuration de l'Active Directory (AD), du DHCP et du DNS pour une gestion centralisée et fluide des ressources réseau et des identités utilisateur.

5. Intégration d'un système de lecture de badge RFID avec un serveur MQTT pour un contrôle d'accès précis et une gestion efficace de l'identification du personnel et des patients.

Organisation

Organisation du Projet

Coordination à distance

L'équipe a été organisée de manière à faciliter la collaboration à distance et à optimiser l'efficacité de la gestion du projet. Des réunions virtuelles régulières ont été tenues via Discord pour discuter des progrès, des problèmes éventuels et des prochaines étapes à suivre. Ces réunions ont permis de maintenir une communication constante malgré la distance géographique entre les membres de l'équipe.

Utilisation de Trello

Trello a été l'outil central pour la gestion des tâches, offrant une vue d'ensemble claire des activités en cours et des étapes à venir. Les tableaux de Trello ont permis de répartir les tâches de manière équilibrée, de suivre les progrès et de résoudre les problèmes de manière collaborative. Cela a également facilité la mise en place de priorités et l'identification des points critiques nécessitant une attention immédiate.

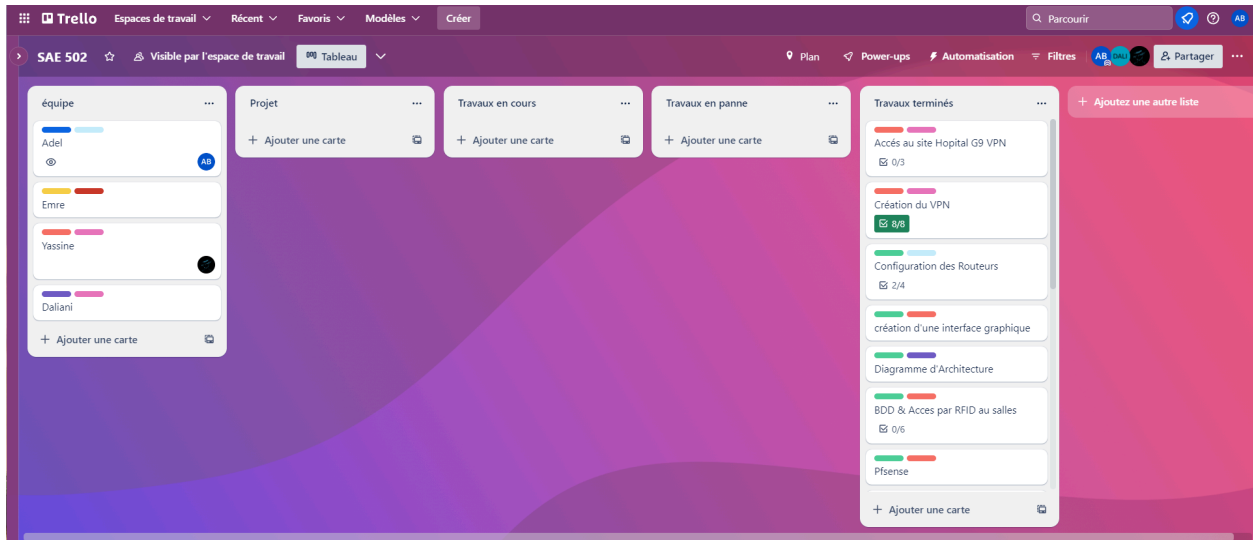


Figure 2 : TRELLO

Voici un exemple de notre Trello à la fin de la SAE.

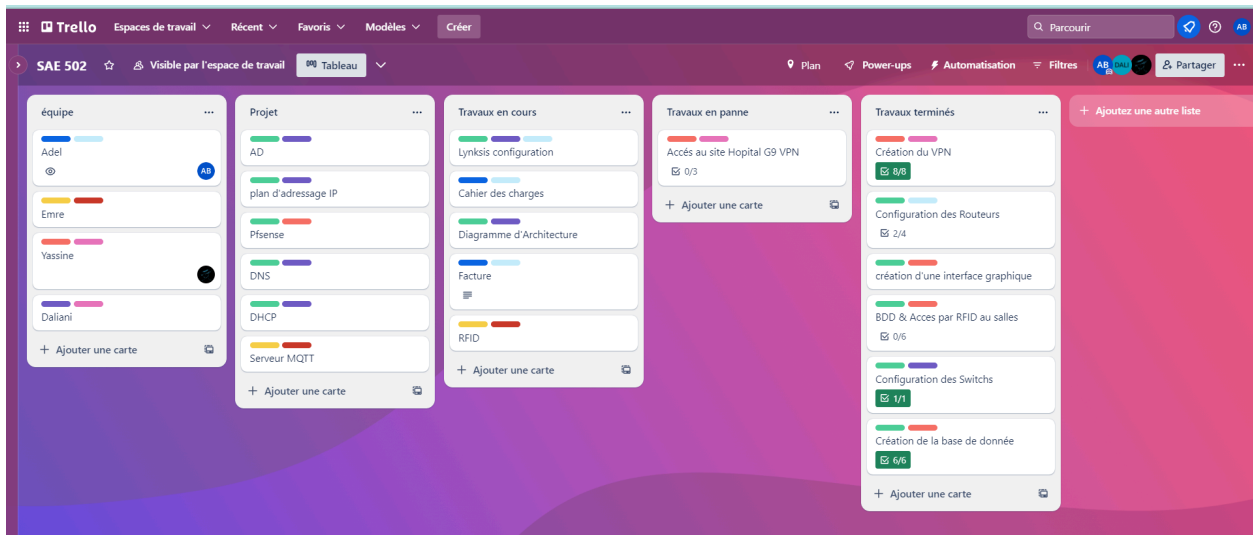


Figure 3 : TRELLO

Voici encore un autre exemple pour montrer ce que ça pourrait donner lors du projet.

Collaboration visuelle avec Lucidchart

Lucidchart a été utilisé pour créer des schémas d'architecture détaillés et des plans de réseau, offrant ainsi une représentation visuelle claire de l'ensemble du projet. Les diagrammes ont permis à l'équipe de mieux comprendre la configuration du réseau et de discuter des solutions potentielles en cas de problèmes ou de défis inattendus.

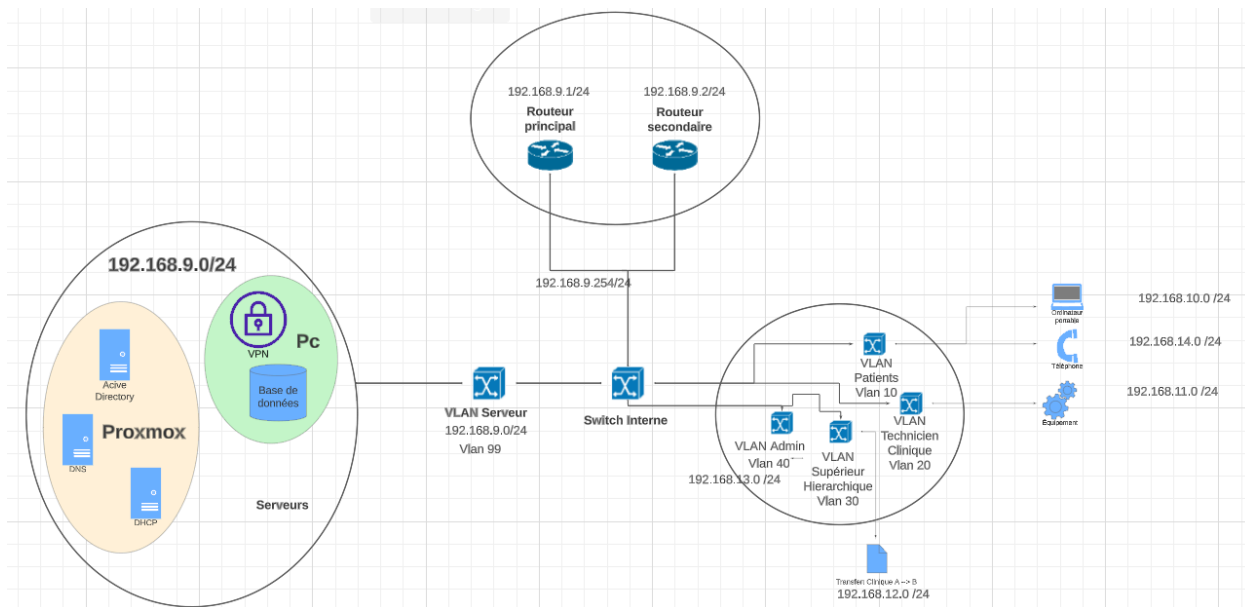


Figure 4 : Organigramme Infrastructure Réseau

Dans notre cas, nous avons décidé de notre topologie réseaux de la manière ci-dessus, ce qui apporte une certaine sécurité à l'architecture réseau au vue des différents Vlan.

Gestion du Temps Flexible

En raison de la nature virtuelle de la coordination, chaque membre de l'équipe a bénéficié d'une flexibilité accrue pour gérer son temps de manière optimale. Cela a permis de résoudre rapidement les problèmes et de s'adapter aux éventuelles modifications de plan sans compromettre la qualité ou les délais du projet.

Assistance Technique à Distance

Malgré la distance, l'équipe a assuré un support technique continu en utilisant des outils de contrôle à distance et de partage d'écran pour résoudre les problèmes rencontrés lors de la configuration de l'infrastructure. Cette approche proactive a permis de minimiser les interruptions et d'assurer un déploiement fluide de l'ensemble du système.

Solutions Proposées

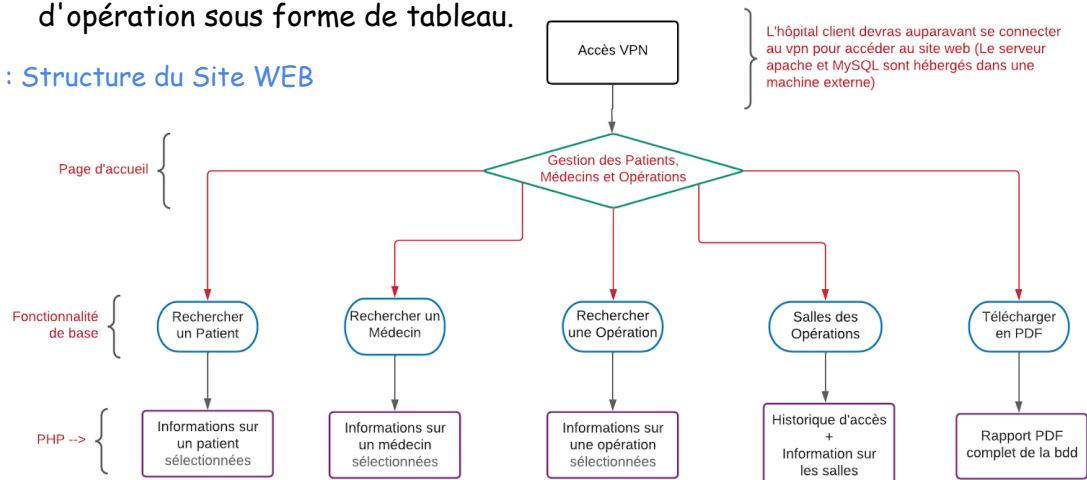
Site WEB

Notre site web est basé sur le langage de programmation PHP pour la logique côté serveur, HTML pour la structure de la page, CSS pour la mise en page, et JavaScript pour les fonctionnalités interactives. La base de données MySQL stocke les informations relatives aux patients, médecins, opérations et salles d'opération.

Les fonctionnalités principales sont les suivantes :

- [Recherche de Patient]
 - Les utilisateurs peuvent rechercher des patients en sélectionnant leur nom dans une liste déroulante. Les détails du patient, y compris les opérations associées, sont affichés.
- [Recherche de Médecin]
 - De même, les utilisateurs peuvent rechercher des médecins et voir les patients qui leur sont attribués.
- [Recherche d'Opération]
 - La recherche d'opérations permet aux utilisateurs de visualiser les détails d'une opération spécifique, y compris les patients, médecins et l'état de la salle d'opération.
- [Historique des Salles d'Opération]
 - Les utilisateurs peuvent consulter l'historique des salles d'opération, y compris les opérations en cours, les badges RFID associés et les derniers accès.
- [Génération de PDF]
 - Une fonctionnalité permet de générer des rapports PDF (bibliothèque FPDF) pour les listes de patients, médecins, opérations et historiques de salle d'opération sous forme de tableau.

Figure 5 : Structure du Site WEB



Analysons la fonctionnalité "Recherche de Patient", qui permet aux utilisateurs de rechercher des informations spécifiques sur un patient en particulier, notamment ses données personnelles, ses opérations associées, et son état de santé. Voici comment cette fonctionnalité est mise en œuvre :

Le code PHP qui gère la recherche de patient est basé sur des requêtes SQL pour interagir avec la base de données MySQL et extraire les données pertinentes. Voici un exemple simplifié de la manière dont cela est implémenté :

On commence par établir une connexion à la base de données MySQL.

```
1  <?php
2  // Connexion à la base de données
3  $connexion = mysqli_connect("localhost", "root", "", "hospitaldb");
```

Puis on récupère le nom du patient que l'utilisateur recherche.

```
5  // Récupération de la valeur de recherche depuis le formulaire (ID du patient sélectionné)
6  $patientID = $_GET["patient"];
7
8  // Requête SQL pour rechercher le patient par son ID
9  $sql_patient = "SELECT * FROM Patients WHERE ID = $patientID";
10 $resultat_patient = mysqli_query($connexion, $sql_patient);
```

Une requête SQL est ensuite construite pour chercher le patient par son nom. Les résultats de la requête sont récupérés et affichés à l'utilisateur.

La fonction de recherche de patient est généralement déclenchée lorsque l'utilisateur sélectionne le nom du patient dans le formulaire de recherche sur le site Web, puis soumet ce formulaire. Le code PHP traitera la requête et affichera les informations du patient correspondant.

```
12 // Requête SQL pour récupérer les opérations effectuées par le patient, les médecins associés et, le cas échéant, les détails de la salle d'opération
13 $sql_operations = "SELECT o.NomOperation, o.DateOperation, o.Etat, d.Nom AS NomDocteur, d.Prenom AS PrenomDocteur, s.NomSalle, s.OperationEnCoursID
14 FROM Operations o
15 JOIN Patients p ON o.ID = p.OperationID
16 JOIN Docteurs d ON p.DocteurID = d.ID
17 LEFT JOIN SalleDesOperations s ON o.ID = s.OperationEnCoursID
18 WHERE p.ID = $patientID";
19
20 $resultat_operations = mysqli_query($connexion, $sql_operations);
21 ?>
```

Cette fonctionnalité permet aux utilisateurs de localiser rapidement et précisément les informations sur un patient spécifique, ce qui est essentiel pour la gestion efficace des dossiers médicaux et la fourniture de soins de santé de haute qualité au sein de la clinique de campagne. Nous analyserons une requête sql dans la prochaine catégorie.

Base de données

La base de données repose sur MySQL. Le schéma de la base de données est conçu pour stocker des informations sur les patients, les médecins, les opérations, les salles d'opération et les badges RFID. Voici un aperçu du schéma de la BDD ::

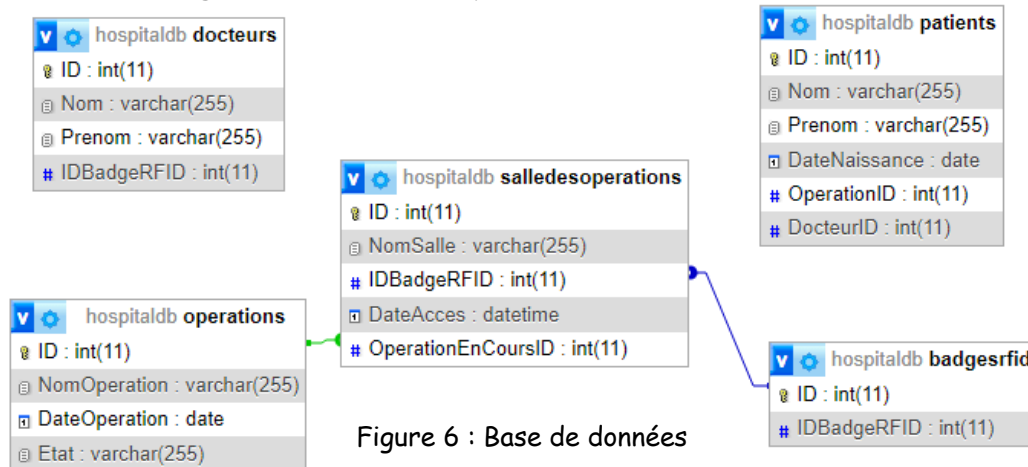


Figure 6 : Base de données

- [Table "Patients"]
 - Cette table contient des informations sur les patients, y compris leur nom, prénom, date de naissance et l'identifiant du médecin qui les prend en charge. Elle est liée à la table Médecins par le biais de l'id du médecin.
- [Table "Médecins"]
 - Cette table stocke des données sur les médecins, leur nom et prénom. Elle est associée à la table "Patients" par le biais de l'identifiant du médecin.
- [Table "Opérations"]
 - Cette table enregistre les détails des opérations, tels que le nom de l'opération, la date de l'opération et son état. Chaque opération est liée à un patient par l'intermédiaire de l'identifiant de l'opération.
- [Table "SallesDesOpérations"]
 - Les informations relatives aux salles d'opération, notamment le nom de la salle, la date d'accès et l'identifiant du badge RFID associé, sont stockées dans cette table. Elle est liée à la table "Opérations" par le biais de l'identifiant de l'opération.
- [Table "BadgesRFID"]
 - Cette table gère les badges RFID et stocke leur identifiant unique. Les badges sont associés aux salles d'opérations.

Exemple de Requêtes SQL Importantes :

```

8 // Requête SQL pour rechercher l'opération par son ID
9 $sql_operation = "SELECT o.NomOperation, o.DateOperation, o.Etat, s.NomSalle
10 FROM operations o
11 LEFT JOIN SalleDesOperations s ON o.ID = s.OperationEnCoursID
12 WHERE o.ID = $operationID";
13 $resultat_operation = mysqli_query($connexion, $sql_operation);
14
15 // Requête SQL pour récupérer les patients associés à l'opération
16 $sql_patients = "SELECT p.Nom, p.Prenom FROM patients p WHERE p.OperationID = $operationID";
17 $resultat_patients = mysqli_query($connexion, $sql_patients);
18
19 // Requête SQL pour récupérer les médecins associés aux patients de l'opération
20 $sql_medecins = "SELECT d.Nom, d.Prenom FROM docteurs d
21 JOIN patients p ON d.ID = p.DocteurID
22 WHERE p.OperationID = $operationID";

```

Figure 7 : Exemple de requête SQL

→ [\$sql_operation]

Cette requête SQL récupère des informations sur une opération spécifique. Elle sélectionne le nom de l'opération, la date de l'opération, l'état de l'opération et le nom de la salle d'opération associée. Elle utilise une jointure gauche (LEFT JOIN) entre la table "operations" (alias "o") et la table "SalleDesOperations" (alias "s") pour obtenir le nom de la salle d'opération associée à l'opération. La condition de filtrage est basée sur l'ID de l'opération (o.ID) correspondant à la valeur de la variable \$operationID.

→ [\$sql_patients]

Cette requête SQL est utilisée pour récupérer la liste des patients associés à l'opération spécifiée. Elle sélectionne les noms et prénoms des patients de la table "patients" où l'ID de l'opération (p.OperationID) correspond à la valeur de la variable \$operationID. Cela permet d'obtenir la liste des patients qui ont subi cette opération.

→ [\$sql_medecins]

Cette requête SQL a pour but de récupérer les médecins associés aux patients de l'opération spécifiée. Elle sélectionne les noms et prénoms des médecins de la table "docteurs" (alias "d") en utilisant une jointure (JOIN) avec la table "patients" (alias "p"). La jointure se fait en comparant l'ID du médecin (d.ID) avec l'ID du médecin associé au patient (p.DocteurID). La condition de filtrage est basée sur l'ID de l'opération à partir de la table "patients" (p.OperationID), qui correspond à la valeur de la variable \$operationID. Cela permet d'obtenir la liste des médecins qui ont traité les patients de cette opération.

Réseaux

Pour assurer une connectivité robuste et résiliente au sein de notre infrastructure réseau, nous avons mis en place une configuration spécifique avec des routeurs stratégiquement positionnés. Initialement, nous avons établi une liaison entre deux routeurs pour la clinique A, qui sont ensuite connectés à un cœur de réseau composé de sept autres routeurs. Cette conception en étoile vise à optimiser la fluidité du trafic et à garantir des performances optimales même en cas de défaillance d'un des nœuds du réseau.

En déployant cette architecture, nous nous assurons que la clinique A dispose d'une redondance adéquate pour maintenir la continuité des opérations, minimisant ainsi les interruptions potentielles des services critiques. De plus, en reliant ces routeurs à deux autres situés à l'hôpital B, nous établissons un accord de redondance croisée, ce qui renforce la résilience globale du réseau. Cette approche stratégique nous permet d'anticiper les pannes éventuelles et de garantir une disponibilité élevée des services réseau pour les besoins cliniques et opérationnels.

En associant cette topologie à une configuration en étoile et en établissant des accords de redondance avec l'hôpital B, nous visons à garantir une stabilité et une continuité de service optimales, réduisant ainsi au minimum les risques de perturbation des opérations critiques.

Pour l'implémentation de ce réseau complexe, nous avons choisi d'utiliser plusieurs protocoles clés pour assurer la connectivité, la stabilité et la sécurité des opérations. Plus précisément, nous avons opté pour l'utilisation de BGP (Border Gateway Protocol), MPLS (Multiprotocol Label Switching) et OSPF (Open Shortest Path First). Voici comment ces protocoles sont intégrés dans notre infrastructure :

1. BGP (Border Gateway Protocol) : Nous avons mis en place BGP pour gérer les échanges de routage entre les différents systèmes autonomes présents dans notre réseau, notamment entre la clinique A et l'hôpital B. Grâce à BGP, nous pouvons assurer une connectivité fiable entre les réseaux distincts tout en garantissant une évolutivité optimale. De plus, BGP nous permet de mettre en place des politiques de routage avancées, ce qui est crucial pour optimiser les performances du réseau.

2. MPLS (Multiprotocol Label Switching) : L'adoption de MPLS s'est avérée essentielle pour garantir une commutation de paquets efficace et rapide dans notre infrastructure. Nous utilisons MPLS pour créer des chemins virtuels sécurisés et dédiés pour le transport des données, ce qui contribue à minimiser les goulots d'étranglement et à améliorer la qualité de service (QoS) pour les applications sensibles à la latence, telles que la télémedecine et la transmission d'imagerie médicale.

3. OSPF (Open Shortest Path First) : Nous avons déployé OSPF pour assurer une convergence rapide et efficace du routage à l'intérieur de notre réseau. OSPF joue un rôle crucial dans la détermination des chemins de communication les plus courts et les plus fiables entre les différents routeurs de notre infrastructure. Grâce à l'utilisation d'OSPF, nous pouvons optimiser la distribution des charges de travail, assurant ainsi une utilisation équilibrée des ressources réseau et une réduction des congestions potentielles.

En combinant ces protocoles de manière judicieuse, nous visons à établir un réseau robuste et sécurisé, capable de répondre aux besoins croissants de connectivité, de vitesse et de fiabilité au sein de notre écosystème de santé. Ces protocoles jouent un rôle clé dans la garantie de performances élevées, d'une redondance efficace et d'une résilience maximale face aux éventuelles pannes ou perturbations du réseau.

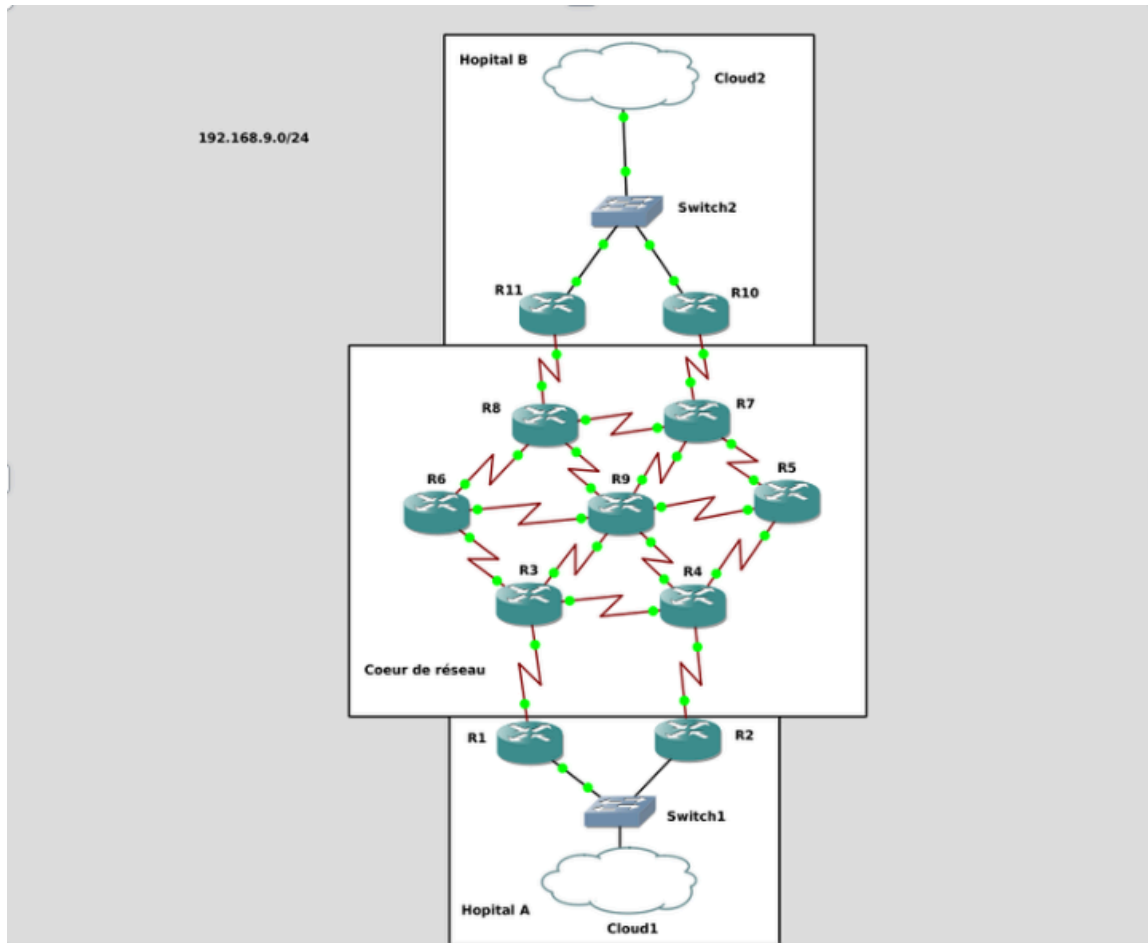


Figure 8 : Infrastructure réseau

Vlan

Nous avons mis en place des Vlan, ces derniers permettent de créer des réseaux locaux afin de séparer les différents acteurs pouvant interagir dans le réseau (Supérieur Hiérarchique, Patient, etc.). Cette séparation permet un cloisonnement des différents réseaux et rend donc étanche en cas de piratage ou de problèmes sur le réseau en n'impactant pas le reste de l'infrastructure.

Active Directory

Un serveur Active Directory, autrement-dit " AD" est un serveur d'annuaire sous Windows. Il est utile pour diverses choses, notamment à la gestion et à l'authentification des utilisateurs sur les machines d'un réseau et au référencement de ces machines sur son réseau informatique.

Il peut aussi apporter des stratégies de sécurité voir d'amélioration du confort du service à l'utilisateur.

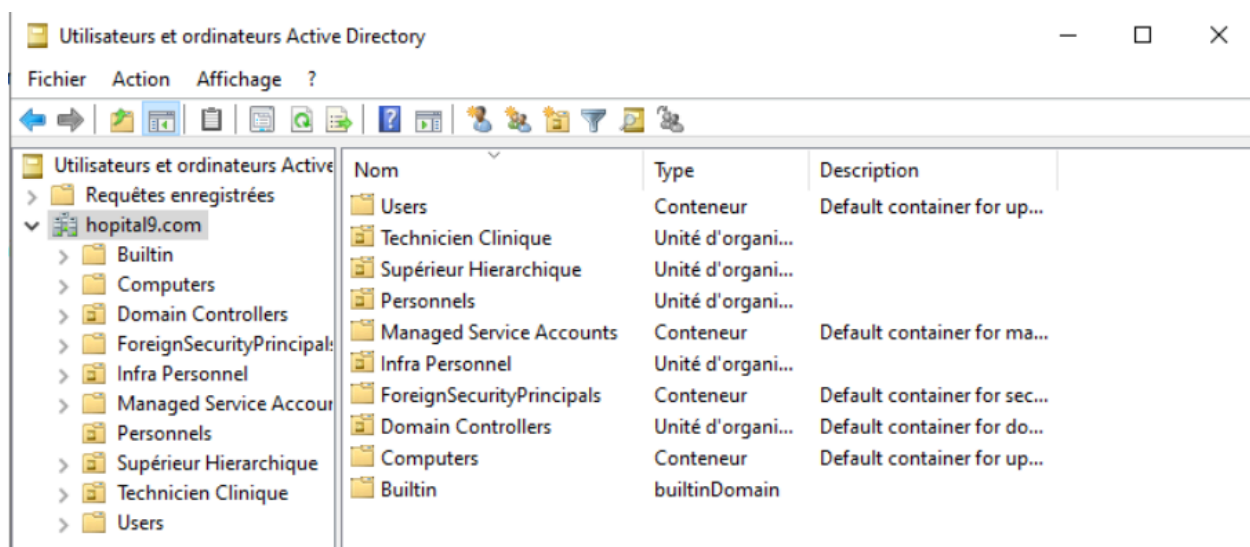


Figure 9 : Domaine Active Directory "hopital9"

Nous pouvons voir ci-dessus la création de différentes Unités d'Organisations afin de visualiser les différents acteurs de notre domaine, se nommant "hopital9".

La mise en place de plusieurs Unités d'Organisations est la solution tout à fait honorable, car nous pouvons attribuer une stratégie de sécurité seulement sur une Unité d'Organisation, ce qui favorise la sécurité, car les droits sont minutieusement appliqués.

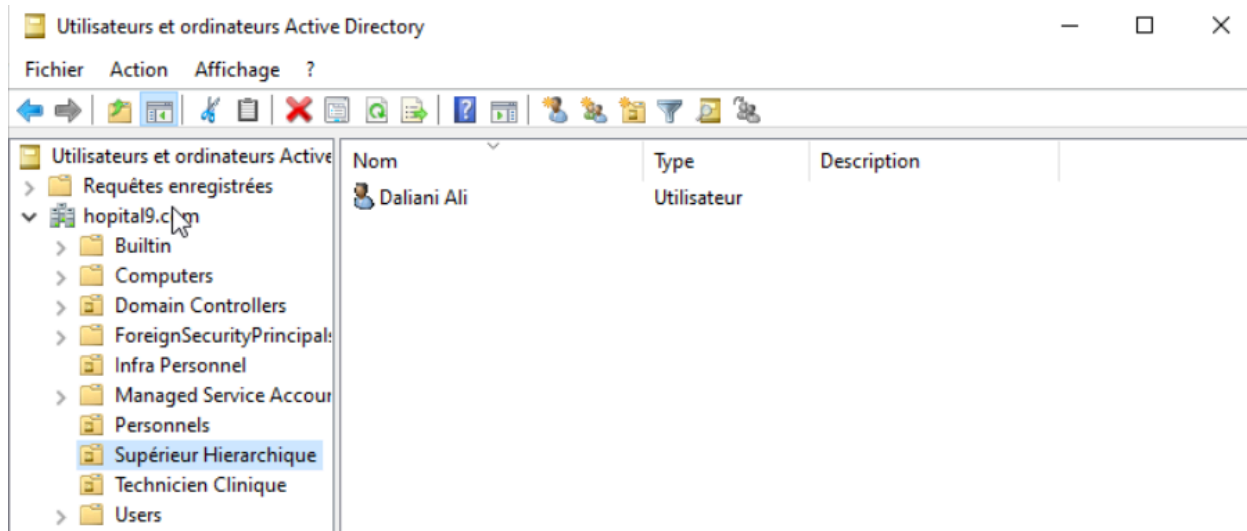


Figure 10 : Unité d'organisation "Supérieur Hiérarchique"

Comme le montre la capture ci-dessus nous avons mis en place l'Unité d'Organisation "Supérieur Hiérarchique" , afin d'appliquer les bons droits aux différentes personnes.

DNS

Un serveur DNS est très utile pour communiquer plus facilement avec une machine via un nom et non pas avec une adresse IP, pour ce faire nous devons lier l'adresse IP de la machine (ordinateur, imprimante, etc.) à un nom.

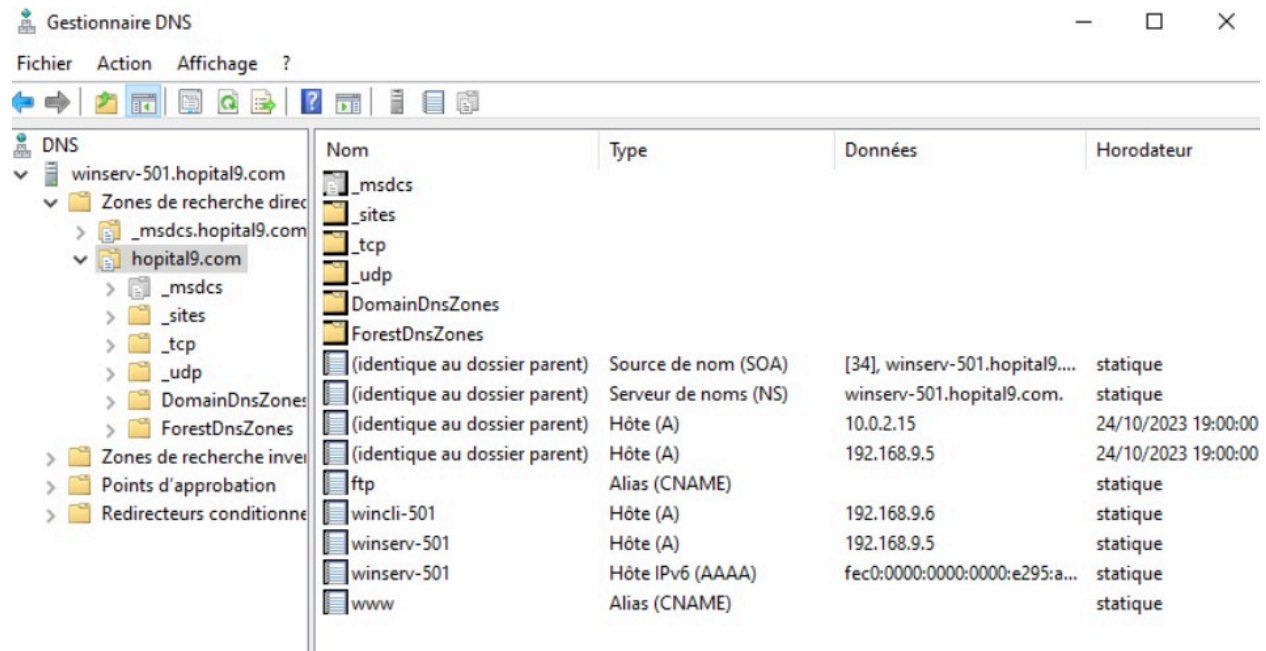


Figure 11 : Serveur DNS

Comme nous l'avons précisé auparavant, nous devons lier une adresse IP à un nom, comme par exemple, pour l'adresse IP 192.168.9.6 nous l'avons synchronisé avec le nom "wincli-501". Nous avons principalement fait des tests de ping sur les différentes machines non pas avec leurs adresses IP, mais avec leurs noms, afin de tester le bon fonctionnement du serveur.

DHCP

La mise en place d'un serveur DHCP au sein d'une entreprise est un élément crucial pour garantir un réseau informatique efficace et bien géré. Le DHCP est un protocole réseau qui permet d'attribuer automatiquement des adresses IP, ainsi que d'autres paramètres de configuration réseau, aux dispositifs connectés au réseau de l'entreprise. Cela réduit les erreurs de configuration manuelle et améliore la flexibilité de l'infrastructure réseau.

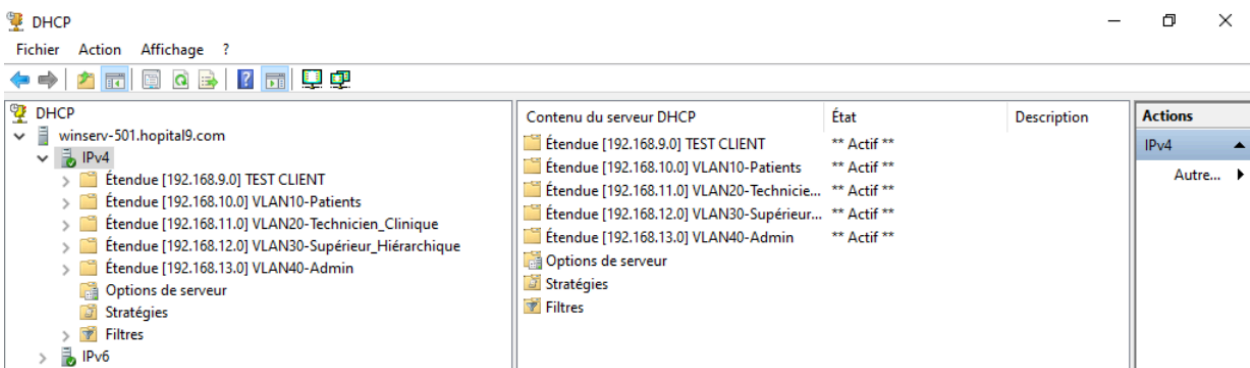


Figure 12 : Unité d'organisation "Supérieur Hiérarchique"

Nous avons mis en place un DHCP qui attribue des adresses IP sous le format IPv4. Suite à cela, nous avons mis en place plusieurs étendues, car nous avons différents Vlan, donc une étendue par Vlan. Nous avons aussi intégré une étendue "TEST CLIENT" afin de tester le bon fonctionnement du DHCP.

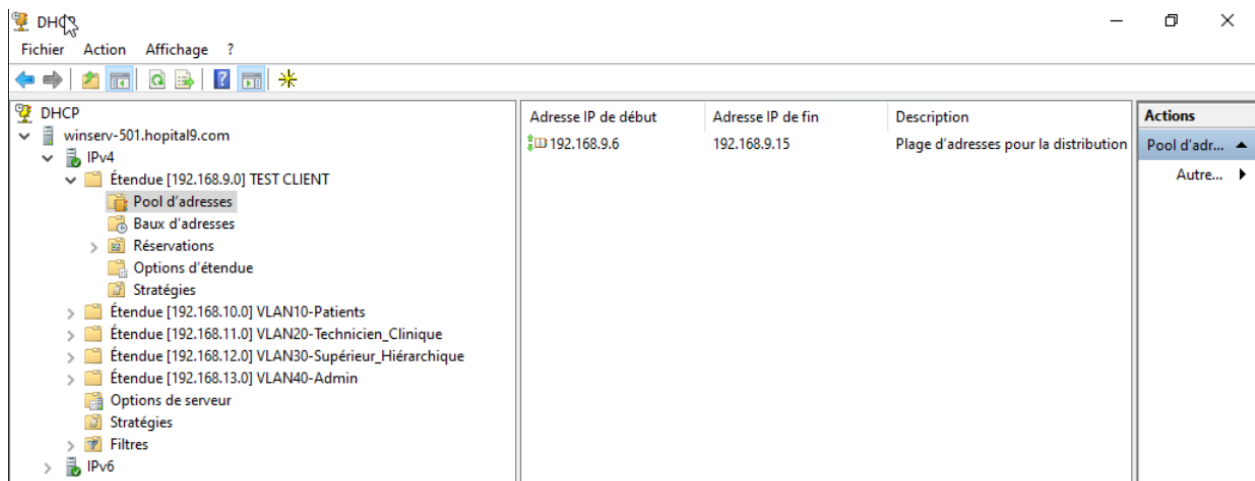


Figure 13 : Plage d'adressage IP

Prenons exemple sur ce dernier, cette étendue attribue des adresses IP allant de 192.168.9.6 à 192.168.9.15.

Pour tester le bon fonctionnement de nos serveurs, nous avons mis en place une machine test sous Windows, nous lui avons aussi fait une réservation à l'aide de son adresse MAC.

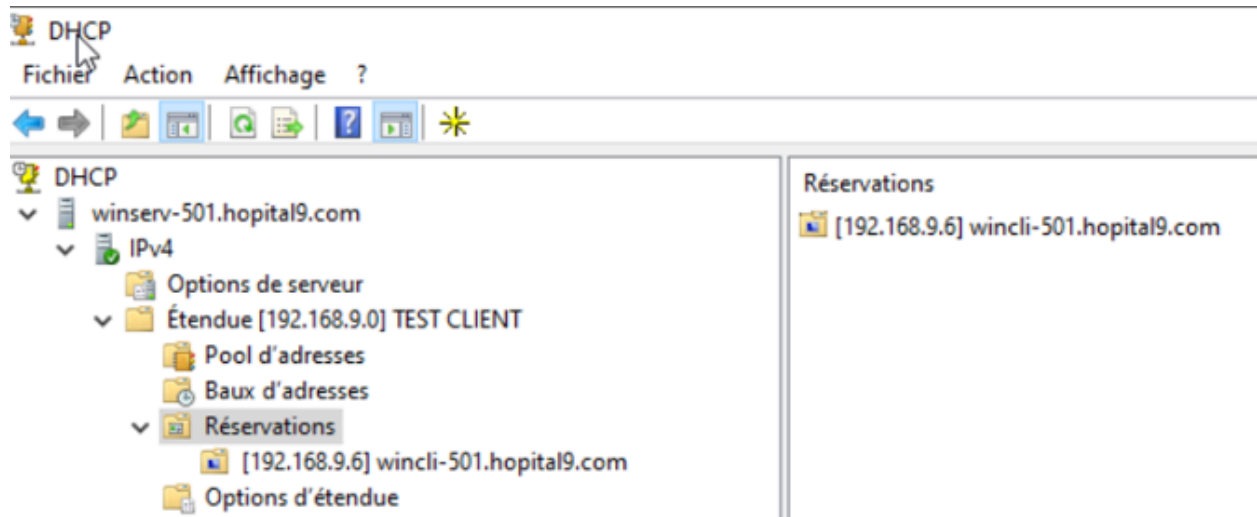


Figure 14 : Réserve de la machine test

Voici ci-dessous un test du bon fonctionnement de nos différents serveurs :

```

Carte Ethernet INTERNE :

Suffixe DNS propre à la connexion. . . : hopital9.com
Adresse IPv6 de liaison locale. . . . . : fe80::e432:b5d6:6e4:6225%5
Adresse IPv4. . . . . : 192.168.9.6
Masque de sous-réseau. . . . . : 255.255.255.0
Passerelle par défaut. . . . . : 192.168.9.5
  
```

Figure 15 : Machine test IP+DNS

Partie Cybersécurité

FireWall

Le pare-feu joue un rôle critique dans le projet d'infrastructure réseau intégrée entre la clinique de campagne et l'hôpital B (client). Il est chargé de garantir la sécurité et de contrôler l'accès au réseau, en particulier en ce qui concerne l'accès au site Web central de gestion des données médicales. Voici comment le pare-feu pfSense est configuré pour répondre à ces besoins :

Le pare-feu pfSense est essentiel pour sécuriser le réseau de la clinique et de l'hôpital, ainsi que pour garantir que l'accès au site Web de gestion des données médicales est limité aux utilisateurs autorisés. Son rôle consiste à :

- [Filtrer le trafic entrant et sortant]
 - Le pare-feu surveille et filtre le trafic réseau pour empêcher les accès non autorisés, les attaques et les intrusions.
- [Autoriser l'accès au site Web uniquement aux utilisateurs VPN]
 - Le pare-feu est configuré pour n'autoriser l'accès au site Web qu'aux utilisateurs connectés au réseau VPN, à savoir la plage IP 10.10.10.0/24.
- [Bloquer l'accès au site Web pour les clients non autorisés]
 - Tous les autres clients qui ne sont pas dans la plage IP du VPN se voient refuser l'accès au site Web.

Le pare-feu PfSense est placé en amont de l'ensemble du réseau du site WEB et la BDD de l'hôpital. Il fait office de passerelle sécurisée entre le réseau interne et le réseau extérieur (Internet). Cela garantit que tout le trafic réseau est acheminé à travers lui pour un filtrage et un contrôle appropriés.

Pour résumer, le processus est le suivant :

Internet→ pfSense→ (Réseau VPN)→ Site WEB (uniquement visionnage)→ Hôpital BDD

Les règles de pare-feu mises en place dans pfSense incluent :

Règles de filtrage

- ◆ Une règle PASS au VPN qui autorise la connexion au VPN par le port 1194 (le port défini de OpenVPN).

Règle Pass au VPN

Rules (Drag to Change Order)	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	✓	0/357 KiB	IPv4 UDP	*	*	WAN address	1194 (OpenVPN)	none		OpenVPN Connect Openvpn wizard	⬇ ✎ 🔒 🗑 ✖

Figure 16 : Règle pass VPN

- ◆ Une règle qui définit quels types de trafic sont autorisés. Pour permettre l'accès au site Web, une règle spécifique autorise le trafic provenant de la plage IP VPN (10.10.10.0/24).

Règle accès au site WEB par le tunnel VPN

Rules (Drag to Change Order)	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	✓	0/0 B	IPv4 *	*	*	*	*	none		OpenVPN Connect Openvpn wizard	⬇ ✎ 🔒 🗑 ✖
<input type="checkbox"/>	✓	0/0 B	IPv4 TCP/UDP	10.10.10.0/24	*	192.168.215.185	80 (HTTP)	none		Autoriser l'accès au site web	⬇ ✎ 🔒 🗑 ✖

Figure 17 : Règles Firewall qui autorise l'accès web par le VPN

→ Règles de blocage

- ◆ Toutes les autres sources de trafic sur le port 80 (port HTTP donc WEB) sont bloquées par défaut, garantissant que l'accès au site Web est réservé aux utilisateurs VPN uniquement.

Règle Block au site WEB

Rules (Drag to Change Order)	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	✗	0/0 B	IPv4 TCP/UDP	LAN net	*	192.168.215.185	80 (HTTP)	*	none	Bloquer l'accès au site web	⬇ ✎ 🔒 🗑
<input type="checkbox"/>	✓	3/3.30 MiB	IPv4 *	LAN net	*	*	*	none		Default allow LAN to any rule	⬇ ✎ 🔒 🗑 ✖
<input type="checkbox"/>	✓	0/0 B	IPv6 *	LAN net	*	*	*	none		Default allow LAN IPv6 to any rule	⬇ ✎ 🔒 🗑 ✖

Figure 18 : Règles Firewall qui bloque l'accès web au réseau LAN

VPN

Le VPN (Virtual Private Network) est une composante cruciale de l'infrastructure réseau, car il permet une communication sécurisée et privée entre l'hôpital de campagne et l'hôpital client. Voici comment le VPN est configuré dans le cadre de ce projet :

Le VPN est installé pour les raisons suivantes :

- [Sécurité des données]
 - Le VPN offre un canal de communication crypté, garantissant que toutes les données échangées entre la clinique et l'hôpital sont protégées contre les interceptions non autorisées.
- [Accès distant sécurisé]
 - Les membres du personnel de la clinique et de l'hôpital peuvent accéder au réseau à distance de manière sécurisée, ce qui est essentiel pour la télémédecine, la gestion à distance des systèmes ainsi que l'échange d'informations sur les patients.

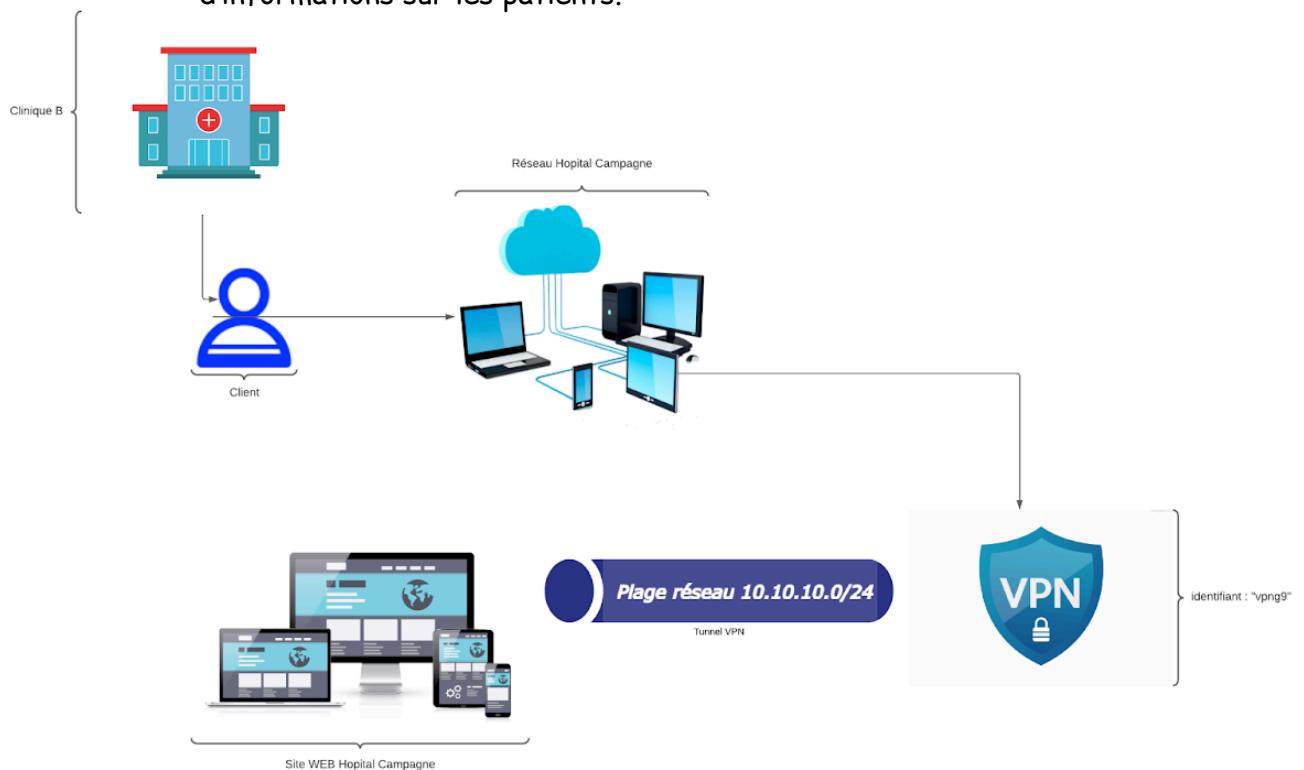


Figure 19 : Schéma VPN

Le VPN OpenVPN est configuré sur le pare-feu pfSense. Voici les détails de configuration qui garantissent la sécurité et la confidentialité des communications :

★ [Certificat d'autorité]

- Un certificat d'autorité est utilisé pour garantir l'authenticité du serveur OpenVPN et la sécurité de la connexion.





Certificate Authorities						
Name	Internal	Issuer	Certificates	Distinguished Name	In Use	Actions
CA-G9	✓	self-signed	1	ST=FR, OU=g9, O=g9, L=Paris, CN=FR, C=FR Valid From: Thu, 26 Oct 2023 21:03:35 +0000 Valid Until: Sun, 23 Oct 2033 21:03:35 +0000		   

Figure 20 : ScreenShot Certificat d'autorité sur le Firewall

★ [Serveurs]

- Dans la section des serveurs, les paramètres comprennent le port du VPN (UDP/1194), le mode de connexion (Remote Access), et les paramètres de chiffrement.




OpenVPN Servers						
Interface	Protocol / Port	Tunnel Network	Mode / Crypto	Description	Actions	
WAN Port VPN	UDP / 1194 (TUN)	10.10.10.0/24 @IP attribué aux clients	Mode: Remote Access (SSL/TLS + User Auth) Data Ciphers: AES-256-GCM, AES-128-GCM, CHACHA20-POLY1305, AES-256-CBC Digest: SHA256 D-H Params: 2048 bits	Connect Openvpn	  	

Figure 21 : ScreenShot Serveurs VPN sur le Firewall

★ [Utilisateurs VPN]

- Les utilisateurs autorisés, tels que l'utilisateur "vpng9", sont définis pour accéder au VPN en utilisant des informations d'identification spécifiques.




Users					
Username	Full name	Status	Groups	Actions	
<input type="checkbox"/> admin	System Administrator	✓	admins		
<input checked="" type="checkbox"/> vpng9	User VPN vpng9	✓	admins	 	

Figure 22 : ScreenShot User VPN sur le Firewall

Accès distant au réseau via OpenVPN :

Les membres du personnel autorisés doivent utiliser des clients OpenVPN pour se connecter au VPN à partir de leurs appareils, tels que des téléphones et des ordinateurs portables. Cela leur permet d'accéder en toute sécurité aux ressources du réseau, y compris le site Web de gestion des données médicales, depuis n'importe où avec une connexion Internet.

Pour ce faire, il suffit d'installer le logiciel OpenVPN client ou l'application OpenVPN sur mobile, puis importer le fichier de connexion contenant toutes les informations nécessaires à la configuration de la connexion VPN (sauf les identifiants et le mot de passe de connexion). Le client devra par la suite s'identifier avec un username et son mot de passe.

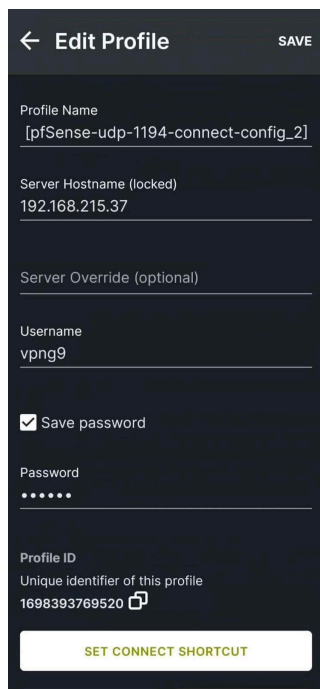


Figure 23 : Config VPN sur mobile

Configuration du VPN

Connexion établie ✓

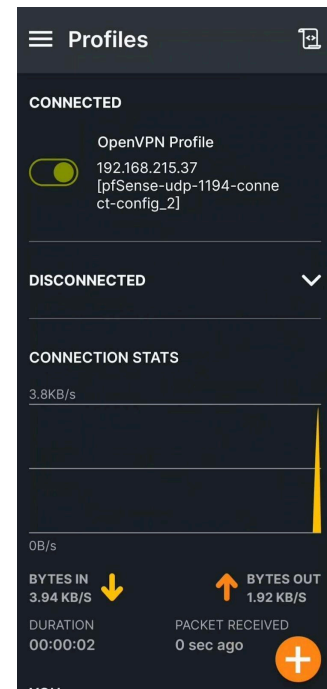


Figure 24 : Connexion établie

En combinant le pare-feu pfSense pour le contrôle de l'accès au site Web et le VPN pour une communication sécurisée, l'ensemble du projet bénéficie d'une sécurité et d'une confidentialité renforcées, ce qui est essentiel pour la gestion des données médicales sensibles et la coordination des soins de santé entre l'hôpital de campagne et l'hôpital client.

Partie IOM

RFID

L'utilisation d'un lecteur NFC en tant que contrôleur d'accès dans une clinique assure une sécurité accrue et une gestion plus efficace du flux de personnes. Avec ce système, l'accès aux différentes pièces peut être contrôlé de manière précise, permettant uniquement aux personnels autorisés d'entrer dans les zones restreintes, comme les salles d'opération, les pharmacies, ou les laboratoires. Cela aide à prévenir les intrusions non autorisées et les risques de contamination croisée. De plus, l'usage d'un lecteur NFC est intuitif et rapide, ce qui réduit les files d'attente et les délais d'attente pour le personnel. En outre, cette technologie peut enregistrer les horaires d'entrée et de sortie, fournissant des données précieuses pour la gestion du temps et la traçabilité en cas d'incident. Cela contribue également à une meilleure conformité aux réglementations concernant la confidentialité des données des patients et la sécurité des installations.

Serveur MQTT

Un serveur MQTT, ou broker MQTT, est une plateforme centrale dans le protocole de communication MQTT. Il sert de médiateur pour la transmission de messages entre les dispositifs, tels que les capteurs et les applications, en gérant les abonnements à des sujets spécifiques et en assurant la distribution des messages en fonction de ces abonnements. Le serveur assure également la fiabilité et la qualité de la livraison des messages à travers différents niveaux de service, maintient les sessions des clients, et fournit des options de sécurité comme l'authentification et le chiffrement. Sa capacité à gérer un grand nombre de connexions simultanément en fait un outil essentiel pour le déploiement de solutions IoT à grande échelle.

Configuration du lecteur NFC

```
void loop() {
  uint8_t success;
  uint8_t uidLength;
  uint8_t uid[] = {0, 0, 0, 0, 0, 0, 0, 0};

  success = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A, uid, &uidLength);

  if (success) {
    Serial.println("Carte détectée!");

    Serial.print("Longueur UID: ");
    Serial.print(uidLength, DEC);
    Serial.println(" octets");
    Serial.print("Valeur UID: ");

    // Convertir le tableau d'octets en chaîne de caractères
    String uidStr = "";
    for (uint8_t i = 0; i < uidLength; i++) {
      uidStr += String(uid[i], HEX);
    }

    // Envoyer l'UID à MQTT
    if (client.connect("ESP8266Client")) {
      client.publish(mqtt_topic, uidStr.c_str());
      client.disconnect();
      Serial.println("UID envoyé à MQTT");
    } else {
      Serial.println("Connection to MQTT broker failed");
    }

    Serial.println("Waiting 1 second");
    delay(1000);
  }
}
```

Figure 25 : Programme lecteur NFC

Ce code est utilisé pour détecter une carte RFID à l'aide d'un lecteur NFC, lire son numéro d'identification unique (UID), et puis envoyer ce numéro à un serveur via MQTT. Si une carte est détectée, le code envoie l'UID à un serveur.

Vérification de l'UID dans la base de données

Il est très important d'installer la bibliothèque 'paho-mqtt'. `Paho-mqtt` est une bibliothèque essentielle pour les développeurs Python qui travaillent avec le protocole MQTT. Elle offre une interface asynchrone robuste pour la connexion à des brokers MQTT, permettant l'envoi et la réception de messages avec différents niveaux de qualité de service (QoS). La bibliothèque facilite la gestion des sessions de communication, y compris les déconnexions et reconnexions automatiques.

Le script ci-dessous est configuré pour se connecter à un broker spécifié par `192.168.9.157` et écouter les messages sur le topic `salle4`. Lorsqu'un message contenant un UID est reçu, le script vérifie sa présence dans un fichier local `salle4.txt`. Si l'UID est trouvé, le script publie un message "validé" sur le topic `autoriser`, indiquant un accès autorisé.

```
MQTT_BROKER = '192.168.9.157'
MQTT_PORT = 1883
MQTT_TOPIC_LISTEN = 'salle4'
MQTT_TOPIC_PUBLISH = 'autoriser'

# Cette fonction est appelée lorsqu'une connexion est établie avec le broker
def on_connect(client, userdata, flags, rc):
    print("Connecté avec le code résultant " + str(rc))
    client.subscribe(MQTT_TOPIC_LISTEN)

# Cette fonction est appelée à chaque message reçu du topic souscrit
def on_message(client, userdata, msg):
    uid = msg.payload.decode()
    print(f"UID reçu: {uid}")
    try:
        with open('salle4.txt', 'r') as file:
            # Lecture de tous les UIDs valides à partir du fichier
            valid_uids = file.readlines()
            # Retire les caractères de nouvelle ligne et espaces
            valid_uids = [uid.strip() for uid in valid_uids]

            if uid in valid_uids:
                print("UID valide. Accès autorisé.")
                client.publish(MQTT_TOPIC_PUBLISH, 'validé')
            else:
                print("UID invalide. Accès refusé.")
                client.publish(MQTT_TOPIC_PUBLISH, 'refusé')
    except FileNotFoundError:
        print("Le fichier salle4.txt n'existe pas.")
        client.publish(MQTT_TOPIC_PUBLISH, 'refusé')
```

En cas d'UID non trouvé ou si le fichier n'existe pas, il publie "refusé", signalant un accès refusé. Le script tourne en continu, attendant et traitant les messages entrants, ce qui en fait un système de contrôle d'accès automatisé et réactif.

Figure 26 : Programme de vérification

Dans mon projet, j'ai opté pour l'utilisation d'un fichier `.txt` pour simuler une base de données afin de gérer les accès à une salle spécifique via un système de contrôle utilisant la technologie NFC. Le fichier `salle4.txt` contient une liste d'UIDs qui représente les identifiants des badges ou des tags NFC autorisés à ouvrir la porte de la salle 4. Lorsqu'un utilisateur tente d'accéder à la salle, le lecteur NFC lit l'UID de son badge et le transmet au système via MQTT. Mon script Python, qui tourne en continu sur un Raspberry Pi, reçoit cet UID et le compare avec les entrées du fichier `salle4.txt`.

J'ai choisi cette méthode, car elle est simple, directe, et suffisante pour les besoins actuels du projet. Elle permet une mise en œuvre rapide sans nécessiter la complexité et la maintenance d'un système de gestion de base de données complet, bien que je sois conscient que pour une application à plus grande échelle, une base de données réelle offrirait plus de sécurité et de fonctionnalités.

Partie Pilotage de Projet

Cahier des charges

Description du projet

Le présent cahier des charges décrit les spécifications techniques et fonctionnelles pour la mise en place d'une infrastructure réseau intégrée visant à relier la clinique de campagne à l'hôpital B. L'infrastructure doit permettre une transmission sécurisée et efficace des données médicales et administratives entre les deux établissements. Le projet implique la mise en place d'une base de données robuste, d'un système de contrôle d'accès, de VLAN distincts, ainsi que la configuration avancée du cœur de réseau.

Objectifs

1. Établir une connectivité fiable entre la clinique de campagne et l'hôpital B pour faciliter le partage sécurisé des informations.
2. Mettre en place des mesures de sécurité avancées pour protéger la confidentialité des données médicales et administratives.
3. Assurer une gestion efficace du réseau pour répondre aux besoins opérationnels de la clinique et de l'hôpital, y compris la gestion des identités, la gestion du trafic et la surveillance du réseau.

Spécifications techniques

1. Mise en place d'une base de données centralisée basée sur PHPMyAdmin et SQL pour stocker et gérer les données médicales et administratives.
2. Développement d'un site Web sécurisé et convivial pour la consultation des informations patients et la prise de rendez-vous en ligne.
3. Configuration d'un VPN utilisant pfSense pour assurer un tunnel sécurisé entre la clinique de campagne et l'hôpital B.
4. Mise en place d'un système de lecture de badge RFID avec un serveur MQTT pour le contrôle d'accès et la gestion des identités du personnel et des patients.
5. Configuration de quatre VLAN distincts (administratif, technique, clinique et patients) pour assurer la séparation des activités et garantir la sécurité des données.
6. Configuration avancée du cœur de réseau en utilisant les protocoles BGP, OSPF et MPLS pour garantir une gestion optimale du trafic et une connectivité robuste entre les différents sites.

Délai

Le projet doit être achevé dans un délai de 20 heures de travail, en tenant compte de la planification, de la configuration, des tests et de la formation du personnel.

Devis

Coûts de main-d'œuvre

1. 4 membres de l'équipe pendant 20 heures : 4000 €
2. Configuration du VPN, de l'Active Directory, du DHCP et du DNS : 1500 €
3. Intégration du système de lecture de badge RFID : 1200 €
4. Configuration des VLAN et des segments réseau, y compris le cœur de réseau avec les protocoles BGP, MPLS et OSPF : 2500 €

Coûts du matériel

1. 11 routeurs : 5500 €
2. 2 switches : 2000 €

Offre Standard

1. Fiabilité et fonctionnalité Essentielles : Une solution qui garantit la fiabilité de base et les fonctionnalités essentielles nécessaires pour relier efficacement la clinique de campagne à l'hôpital B.
2. Support Technique Inclus : Assistance complète incluse pendant la période de garantie pour assurer un fonctionnement fluide de l'infrastructure réseau.
3. Coût abordable : Un prix compétitif qui correspond à votre budget sans compromettre la qualité du service et des équipements.

1. Coût total de la main-d'œuvre pour l'offre standard : 9000 €
2. Coût total du matériel pour l'offre standard : 7000 €
3. Prix total de l'offre standard : 16000 €

Offre Premium

1. Performance de Pointe : Une solution haut de gamme offrant des performances exceptionnelles pour répondre aux besoins avancés de connectivité et de sécurité de la clinique et de l'hôpital.
2. Garantie Étendue : Une garantie prolongée assurant le bon fonctionnement de l'infrastructure pendant une période étendue, réduisant ainsi les coûts potentiels de maintenance à long terme.
3. Support Technique Prioritaire : Un support technique prioritaire avec un temps de réponse rapide pour résoudre les problèmes de manière efficace et minimiser les interruptions de service.

1. Coût total de la main-d'œuvre pour l'offre premium : 11000 €
2. Coût total du matériel pour l'offre premium : 8000 €
3. Prix total de l'offre premium : 19000 €

Conclusion

Le projet de mise en place d'une infrastructure réseau intégrée pour la clinique de campagne a été une expérience formatrice pour toute l'équipe. En combinant nos compétences variées en gestion de projet, développement web, sécurité réseau et configuration avancée, nous avons réussi à relever les défis techniques et à offrir une solution robuste répondant aux besoins croissants de l'établissement.

L'intégration de divers outils de gestion tels que Trello, Lucidchart et Discord a grandement amélioré notre efficacité de communication et de collaboration à distance, jouant un rôle clé dans la coordination réussie de l'équipe malgré la distance géographique (travail à la maison).

Cependant, avec un temps limité, nous reconnaissons qu'il reste des améliorations possibles. Nous aurions pu affiner davantage notre solution. Malgré ces contraintes, nous sommes fiers des résultats obtenus, et cette expérience nous a inspirés pour rechercher des opportunités futures où nous pourrions mettre en œuvre des solutions encore plus efficaces et robustes.

Cette expérience a renforcé notre conviction que l'investissement dans le développement continu de nos compétences et de nos connaissances est essentiel pour relever avec succès les SAE comme celle-ci.