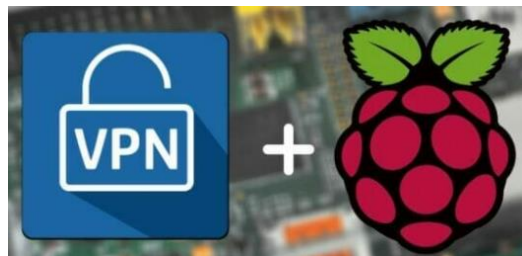




Projet #1

Étude et mise en place d'un VPN nomade sur Raspberry PI



Groupe numéro : 2

Rédigé par : Mohamed Fatima

Doboehloui Danian Yann Micael

Blanchard Julien

Kaddouri Ahmed

Fekrane Youssef

Coordonné par : KALLA, Christian Gille

SECS1026 - Projet d'intégration en cybersécurité

CYSE 2 – 2021 – 2022

Table des matières

Table des matières	2
1. Remerciements.	4
2. Liste des figures.	5
3. Avant-propos.	9
4. Introduction et présentation générale du projet.	10
4.1 Introduction.	10
4.1.1 Finalités du projet.	10
4.1.2 Espérance de retour sur investissement.....	10
4.2 Contexte du projet.	10
4.2.1 Situation du projet par rapport aux autres projets de l'entreprise.	10
4.2.2 Études déjà effectuées.....	11
4.2.3 Suites prévues.	11
4.2.4 Parties concernées par le projet et ses résultats (demandeurs, utilisateurs).	11
1.2.1 Caractère confidentiel s'il y a lieu.	11
4.3 Énoncé du besoin.	12
4.3.1 Finalités du produit pour le futur utilisateur tel que prévu par le demandeur.	12
4.4 Environnement du produit recherché.	12
4.4.1 Listes exhaustives des éléments (personnes, équipements, matériels...).....	12
4.4.2 Contraintes et environnement.	13
4.4.3 Caractéristiques pour chaque élément de l'environnement.....	13
5. Expression fonctionnelle du besoin.	14
5.1 Fonctions de service et de contrainte.....	14
5.1.1 Fonctions de service principales (raison d'être du produit).	14
5.1.2 Amélioration du service rendu.	14
5.1.3 Contraintes (limitations à la liberté du concepteur-réalisateur).	15
5.2 Critères d'appréciation (déterminants pour l'évaluation des réponses).....	15
5.3 Niveaux des critères d'appréciation et ce qui les caractérise.....	15
6. Cadre de réponse.	16
6.1 Pour chaque fonction.....	16
6.1.1 Architecture de la solution proposée.	16
6.1.2 Part du prix attribué à chaque fonction.....	17
6.2 Pour l'ensemble du produit.	17

6.2.1	Prix de la réalisation de la version de base.....	17
6.2.2	Options et variantes proposées non retenues au cahier des charges.....	18
6.2.3	Mesures prises pour respecter les contraintes et leurs conséquences.....	18
6.2.4	Outils d'installation, de maintenance à prévoir.....	18
6.2.5	Décomposition en modules, sous-ensembles.	18
6.2.6	Prévisions de fiabilité.	21
6.2.7	Perspectives d'évolution technologique.....	21
7.	Planification et conception du projet.	22
7.1	L'outil utilise pour planifier et exécuter le projet.	22
7.2	Exemples des Dashboard qui concerne la planification du projet.....	24
8.	Préconisation technique et organisationnelle.	26
8.1	Mise en œuvre de la solution.	26
8.1.1	Etape1 : Rassembler Raspberry Pi.	26
8.1.2	Etape2 : Introduction.	28
8.1.3	Etape3 : Ce dont on a besoin.	29
8.1.4	Etape4 : La grande image.....	29
8.1.5	Etape5 : Installation OpenWRT system.....	30
8.1.6	Etape6 : Connecter sur Pi.....	35
8.1.7	Etape7 : OpenWRT Basique configuration.....	38
8.1.8	Étape 8: Configuration reseau OpenWRT.	42
8.1.9	Etape 9 : Connexion OpneWRT sur le WIFI.....	47
8.1.10	Etape10 : Config du USB Wireless Adapter.....	53
8.1.11	Etape11 : Configuration du réseau sans fil.	58
8.1.12	Etape12 : RPV configuration avec ProtonVPN.....	61
8.1.13	Test que VPN marche.....	70
8.1.14	Test rapidité de la connexion avant et après l'utilisation de notre VPN.	72
8.2	Planning du déploiement de notre projet.	73
1.1.1	Les étapes du projet et leur durée.....	73
1.1.2	Tableau prévisionnel du projet.	74
8.3	Évaluation et le budget financier du nouveau système.....	74
8.4	Conclusion.....	76
8.5	Glossaire.....	78
9.	Références :	80

1. Remerciements.

Nous tenons à remercier toutes les personnes qui ont contribué à la réussite de notre projet et qui nous ont aidés lors de la rédaction de ce projet.

Tout d'abord, nous tenons à remercier notre professeur, **M. CHRISTIAN KALLA**, pour sa patience, sa disponibilité et surtout ses conseils avisés, qui ont contribué à alimenter notre réflexion sur ce projet.

Nous remercions également tous nos enseignants du Collège communautaire du Nouveau-Brunswick (CCNB), particulièrement à : **M. CHARLES LEVESQUE, M. GOGUEN RENÉ, M. MELANSON PASCAL, M. MORISSETTE LOUIS, M. DUBÉ TONY, Mme MALLARD KATY** Merci pour vos conseils, orientations et suggestions tout au long de notre parcours;

Merci à l'administration du Collège communautaire du Nouveau-Brunswick (CCNB), pour la qualité de l'enseignement qui nous a été donnée et qui a été un grand atout pour nous lors de nos différentes matières.

2. Liste des figures.

Figure 1.1.0	Architecture du projet	17
Figure 1.1.1	Prix attribué	18
Figure 1.1.2	Prix pour l'ensemble du projet	19
Figure 1.1.3	Canakit complet	20
Figure 1.1.4	Wireless WiFi USB Dongle Stick Adapter	21
Figure 1.1.5	Écran Tactile	21
Figure 1.1.6	Clavier	22
Figure 1.1.7	Logo de VPN	22
Figure 1.1.8	Dashboard Jira pour les tâches déjà faites	23
Figure 1.1.9	Dashboard Jira pour les tâches attribuées	24
Figure 1.2.0	Dashboard Jira pour les tâches assignées, en cours et terminées.	24
Figure 1.2.1	Dashboard Jira pour l'organigramme cumulatif.	25
Figure 1.2.2	Dashboard Jira pour un graphique à barres.	26
Figure 1.2.3	Dashboard Jira pour un graphique circulaire.	26
Figure 1.2.4	Dashboard Jira pour un graphique en courbe.	27
Figure 1.2.5	Ensemble de Raspberry Pi	28
Figure 1.2.6	Composant de Raspberry Pi	29
Figure 1.2.7	Les étapes pour rassembler le Raspberry Pi	30
Figure 1.2.8	Carte et lecture SSD	32
Figure 1.2.9	Raspberry Pi imager	33
Figure 1.3.0	Lors du choix de la carte	34
Figure 1.3.1	Lors du choix pour l'image du OpenWRT	34

Figure 1.3.2	Lors du choix du stockage pour la carte SSD.	35
Figure 1.3.3	Lors du choix pour l'écriture.	36
Figure 1.3.4	Lors du choix pour accepte l'installation.	37
Figure 1.3.5	Nous continuons l'installation.	38
Figure 1.3.6	Première connexion du Raspberry pi	39
Figure 1.3.7	Changement de mot de passe	39
Figure 1.3.8	Connexion de Raspberry pi avec PuTTY	40
Figure 1.3.9	Configuration basique d'openwrt	41
Figure 1.4.0	Connexion de Raspberry pi avec PuTTY	42
Figure 1.4.1	Étape de configuration de Raspberry pi	42
Figure 1.4.2	Affichage du dossier firewall, network et wireless.	43
Figure 1.4.3	Modifications du dossiers backup et NETWORK	43
Figure 1.4.4	Modifications du Lan network	44
Figure 1.4.5	Création d'une nouvelle interface	44
Figure 1.4.6	Changement de Wireless	45
Figure 1.4.7	Changement de l'interface LAN	46
Figure 1.4.8	Modification du fichier firewall	47
Figure 1.4.9	Réglage du fichier dans la zone de configuration	48
Figure 1.5.0	Reconfiguration du câble réseau LAN	49
Figure 1.5.1	Connexion entre notre Raspberry Pi et notre réseau domestique	50
Figure 1.5.2	Résultat après la commande : Vi /etc/config/wireless	50
Figure 1.5.3	Ajout d'option supplémentaire	51
Figure 1.5.4	Résultat du bon fonctionnement de la configuration	52

Figure 1.5.5	Connexion de Raspberry pi dans le navigateur web	52
Figure 1.5.6	Wireless dans le navigateur Web	53
Figure 1.5.7	Scan du réseau	53
Figure 1.5.8	Join du network	53
Figure 1.5.9	Configuration du wireless et ajout d'un mot de passe	54
Figure 1.6.0	Sauvegarde de la configuration	54
Figure 1.6.1	Enregistrer et appliquer la configuration.	55
Figure 1.6.2	Ping pour s'assurer que vous avez une connexion internet	55
Figure 1.6.3	Wireless WiFi USB Dongle Stick Adapter RT5370	56
Figure 1.6.4	Mise à jour du paquet	56
Figure 1.6.5	Résultat de la mise à jour.	57
Figure 1.6.6	La commande utilise après la mise à jour.	57
Figure 1.6.7	Résultat de la commande après la mise à jour.	58
Figure 1.6.8	lsusb pour voir combien de périphériques USB sont connectés	59
Figure 1.6.9	Activation de Wireless	59
Figure 1.7.0	Affichage de l'interfaces réseau	60
Figure 1.7.1	Configuration du réseau sans fil	61
Figure 1.7.2	Modification à apporter à la configuration du réseau sans fil	62
Figure 1.7.3	Résultat de la commande Uci	62
Figure 1.7.4	Accès Wifi depuis notre machine	63
Figure 1.7.5	Connexion avec le wifi de notre machine.	63
Figure 1.7.6	Téléchargement de ProtonVPN.	64
Figure 1.7.7	Étapes de configuration d'OpenVPN	65

Figure 1.7.8	Renommez le fichier ProtonVPN.	65
Figure 1.7.9	Création d'un fichier d'authentification	65
Figure 1.8.0	Modification du fichier ProtonVPN.ovpn	66
Figure 1.8.1	Remplir notre fichier ProtonVPN.auth	67
Figure 1.8.2	Copier des fichiers entre la machine locale et le système OpenWRT.	67
Figure 1.8.3	Vérification du fichier copié	68
Figure 1.8.4	Interface OpeWRT sur le navigateur	69
Figure 1.8.5	Téléchargement du fichier ProtonVPN.OVPN	70
Figure 1.8.6	Ajout de la configuration de l'instance OpenVPN	71
Figure 1.8.7	Enregistrement de configuration d'instance OpenVPN	72
Figure 1.8.8	Bon fonctionnement d'OpenVPN.	73
Figure 1.8.9	Connexion avec notre réseau internet	73
Figure 1.9.0	Connexion wifi sur notre Raspberry PI	74
Figure 1.9.0	Position actuelle après connexion wifi sur notre Raspberry PI	74
Figure 1.9.2	Connexion sans VPN	75
Figure 1.9.3	Connexion avec VPN	75
Figure 1.9.4	Étapes du projet et leur durée	76
Figure 1.9.5	Tableau prévisionnel du projet	77
Figure 1.9.6	Évaluation et budget financier	77
Figure 1.9.7	Glossaire	81- 82

3. Avant-propos.

Le parcours collégial en cybersécurité prend fin avec un projet d'intégration, ce projet vise à créer à l'aide des compétences acquises un outil informatique, permettant de faciliter la vie, assurer la sécurité face aux menaces informatiques.

L'objectif de ce programme était de nous aider à acquérir compétences aux niveaux techniques et administratifs en science informatique plus précisément en sécurité informatique. Notre but ultime à la sortie de ce programme est de protéger les appareils et les utilisateurs contre toutes formes de menaces. Grâce à ce programme nous avons la capacité désormais d'analyser un réseau et même extraire les vulnérabilités de celles-ci et apporter des solutions. Cependant, le risque zéro "0" n'existe pas, mais la sensibilisation des usagers permettra de mettre fin aux attaques.

4. Introduction et présentation générale du projet.

4.1 Introduction.

Les besoins d'utilisation d'internet dans les lieux publics (Restaurants, Hôtels, bus, supermarchés, etc...) augmente fortement, notamment avec l'utilisation des smartphones, tablettes et PC portables, mais l'utilisation de ces wifi publics pose un grand problème de sécurité car ils ne sont pas à 100% sécurisées.

Toutefois notre thème se base particulièrement sur l'étude et la mise en place d'un VPN nomade sur Raspberry PI qui permet de renforcer l'intégrité, la confidentialité et la preuve lors de l'échange de données sur un réseau public comme l'internet pour les utilisateurs nomade.

4.1.1 Finalités du projet.

Déployer et configurer le Raspberry Pi comme un retour RPV (réseau privé virtuel) pour renforcer la confidentialité et l'intégrité des données pour les utilisateurs nomades qui utilise les réseaux publics internet.

4.1.2 Espérance de retour sur investissement.

Nous espérons un retour sur investissement positive pour ce projet en appliquant cette formule

Retour sur investissement (%) = (gain de l'investissement – coût de l'investissement) / coût de l'investissement.

Exemple pour un seul Raspberry PI configurer et vendu :

Retour sur investissement (%) = $(400-340) / 340 = 17\%$

4.2 Contexte du projet.

4.2.1 Situation du projet par rapport aux autres projets de l'entreprise.

C'est le seul projet encours en ce moment donc on mettra tous les efforts pour livrer ce dernier dans le temps.

4.2.2 Études déjà effectuées.

- Ce projet réglera le problème de la sécurité des utilisateurs nomades.
- Ce projet utilisera un matériel moins cher et flexible à installer c'est le Raspberry PI 4.
- On utilisera des logiciels moins chère et open source comme linux et OpenVPN.
- On utilisera JIRA comme outil pour la gestion du projet.

4.2.3 Suites prévues.

- Diviser l'objectif final en tâches individuelles en respectant le temps pour chaque tâche.
- Délivrer des livrables pour chaque tâche.
- Faire en sorte que les tâches soient liées.
- Définir les heurs, jours et les ressources disponibles pour accomplir chaque tâche.
- Définir des responsables pour toutes les tâches.
- Faciliter la communication entre l'équipe en utilisant un outil efficace pour la gestion des projets.
- Délivrer le projet en production en respectant le temps.

4.2.4 Parties concernées par le projet et ses résultats (demandeurs, utilisateurs).

Ce projet sera utilisé par tous type d'utilisateurs qui veut sécuriser ces communications ou l'utilisation du réseau public Internet.

1.2.1 Caractère confidentiel s'il y a lieu.

Ce projet sera un projet open source pour donner aux autres étudiants et à n'importe quel chercheur l'occasion d'améliorer le projet.

4.3 Énoncé du besoin.

4.3.1 Finalités du produit pour le futur utilisateur tel que prévu par le demandeur.

- S'assurer que les besoins personnels sont satisfaits.
- Savoir recueillir et comprendre les objectifs du projet.
- Savoir trouver le bon niveau de détail pour le projet.
- Réaliser le projet selon les instructions appropriées.
- Obtenir des installations qui permettent d'effectuer le travail de manière productive et sécuritaire.
- Assurer le succès du projet.

4.4 Environnement du produit recherché.

4.4.1 Listes exhaustives des éléments (personnes, équipements, matériels...).

- Un Raspberry Pi.
- Une carte micro SD ou un lecteur de carte SD USB, Ethan au câble (livré avec le kit).
- Un adaptateur Wi-Fi USB.
- OpenWRT système d'exploitation
- PortonVPN pour créer notre VPN.

4.4.2 Contraintes et environnement.

Chaque projet comporte des limites et des risques, qui doivent être pris en compte et gérés afin d'assurer le succès ultime du projet.

Par exemple :

- La livraison du projet, incluant les dates de soumission pour chaque phase du projet, ainsi que la date de soumission du document final.
- Les objectifs du livrable, ses caractéristiques et fonctions, ainsi que les tâches requises.
- Toutes les ressources financières nécessaires pour mener à bien le projet dans les délais prédéfinis.
- Planification stratégique (structure, ressource, procédures, etc..).

4.4.3 Caractéristiques pour chaque élément de l'environnement.

- Un Raspberry Pi est un nano-ordinateur monocarte à processeur ARM de la taille d'une carte de crédit, son faible coût et son système d'exploitation open source Raspbian, il s'agit toujours d'un ordinateur adapté aux besoins de base en matière de programmation.
- Une carte micro SD ou un lecteur de carte SD USB : c'est pour pouvoir installer le système d'exploitation. Est un élément important. Elle stocke les contenus à long terme.
- Un adaptateur Wi-Fi USB : Remplace la fonctionnalité sans fil intégrée de l'ordinateur, offrant une connexion plus rapide et plus fiable aux signaux réseau disponibles via le port USB.
- OpenWRT un système d'exploitation open source pour le routeur.

- ProtonVPN un fournisseur du VPN pour sécuriser notre réseau.

5. Expression fonctionnelle du besoin.

5.1 Fonctions de service et de contrainte.

Internet est désormais un outil indispensable dans la société actuelle, cependant Internet présente des avantages mais également des inconvénients. Parmi les inconvénients, la sécurité car les réseaux WIFI public présentent un danger grave maintenant. Face à cela nous pensons qu'un Raspberry Pi configuré comme un routeur portatif pourra renforcer la sécurité pour les gens qui voyagent ou les gens nomades.

5.1.1 Fonctions de service principales (raison d'être du produit).

Le Raspberry Pi est un micro-ordinateur que l'on connecte à un écran. Le Raspberry Pi est un outil idéal afin de renforcer l'intégrité, la confidentialité et la preuve lors d'échange de données sur un réseau public. Tout au long de notre projet nous le configurerons comme un routeur portatif, à usage fixe en tout lieu connecté à un réseau Internet comme dans un bus ou dans un hôtel, à la maison et au travail.

5.1.2 Amélioration du service rendu.

Dans ce projet, exclusivement le Raspberry-Pi sera utilisé comme un routeur mais il présente aussi d'autres fonctionnalités. L'utilisateur du Raspberry Pi pourra l'utiliser comme une Box TV car elle pourra établir la connexion entre une télévision standard et l'Internet. De plus, le Raspberry peut être également utilisé comme un serveur de stockage de données. Il permet

de centraliser les tâches utilisées de tous les appareils de la maison. Finalement, Cet outil peut être qualifié de magique car il permet d'exécuter multiples tâches.

5.1.3 Contraintes (limitations à la liberté du concepteur-réalisateur).

Les principales contraintes portent sur les performances du processeur et la mémoire vive disponible (256 Mo). Ce dernier point a toutefois été corrigé avec l'arrivée de versions embarquant 512 Mo de mémoire vive, puis d'une version embarquant 1 Go de mémoire vive. La version 4 (juillet 2019) embarque jusqu'à 8 Go de mémoire vive.

5.2 Critères d'appréciation (déterminants pour l'évaluation des réponses).

Ici on parle sur la sécurité, la vitesse de connexion, le nombre d'appareils pouvant se connecter en même temps, le nombre de serveurs, la diversité géographique.

5.3 Niveaux des critères d'appréciation et ce qui les caractérise.

5.3.1 Niveaux dont l'obtention est imposée.

Les niveaux relatifs aux principales fonctionnalités telles que l'aspect sécurité doivent avoir obtenu un meilleur succès. Une fonction qui ne correspond pas à ce niveau devra être travaillée jusqu'à l'obtention du niveau opération souhaitée.

5.3.2 Niveaux souhaités mais révisables.

Toutes les fonctionnalités sont souhaitées être fonctionnelles. Le nombre de serveurs disponible est important pour assurer que plusieurs connexions sont établies sans stresser le système. Bien qu'au moins un certain nombre soit requis par rapport au nombre d'utilisateurs. Il est aussi important de balancer le budget et l'habileté de les maintenir.

6. Cadre de réponse.

6.1 Pour chaque fonction.

6.1.1 Solution proposée.

Une façon sécurée à se connecter à l'internet, n'importe où, en utilisant un Raspberry Pi.

6.1.1 Architecture de la solution proposée.

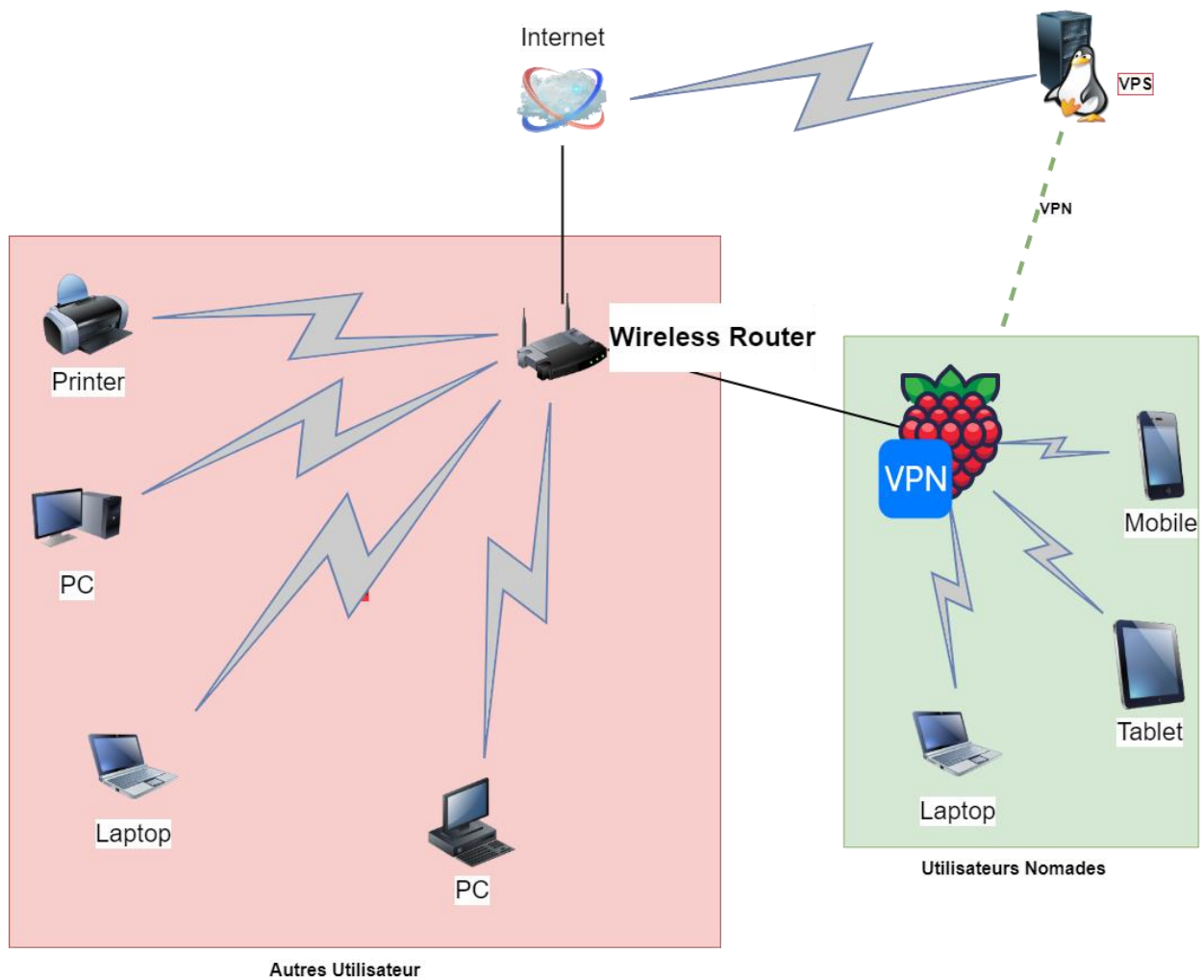


Figure 1.1.0 : Architecture du projet.

6.1.2 Part du prix attribué à chaque fonction.

On divise le projet en 4 parties pour le prix.

- Raspberry Pi Kit + adaptateur (56%)
- Périphériques (24%)
- Software (4%)
- Frais (17%)

Nom	Prix	Partie du prix (%)
Raspberry Pi 4 Kit	\$200	56%
WiFi Dongle	\$34.13	10%
Écran Tactile	\$35.92	10%
Clavier	\$12.99	4%
ProtonVPN (1 Mois)	\$13.74	4%
Frais (taxe + livraison)	\$59.73	17%

Figure 1.1.1 : Prix attribué.

6.2 Pour l'ensemble du produit.

6.2.1 Prix de la réalisation de la version de base.

Pour réaliser ce projet, nous devront avoir accès à un Raspberry Pi 4 Kit et un Wireless WiFi USB Dongle Stick Adapter. En raison de problème de stockage, nous avons dû aller avec un Raspberry Pi un peu plus dispendieux. On aura aussi besoin d'un VPN.

Si vous n'avez pas des écrans, claviers ou souris, il faudra s'en procurer aussi.

Raspberry Pi 4 Kit	\$200
Wireless WiFi USB Dongle Stick Adapter	\$34.13
Express Shipping	\$25.86
GST/HST	\$33.87
Total	\$293.86
Optionnelle, si vous l'avez, pas besoin d'achetez	
Écran Tactile	\$35.92
Clavier	\$12.99
ProtonVPN (1 Mois)	\$13.74
Total avec les extras	\$356.51

Figure 1.1.2: Prix pour l'ensemble du projet

Temps d'installation prévue : 4h

Les liens que nous avons utilisés pour acheter nos produits se trouvent ci-dessous dans les références.

6.2.2 Options et variantes proposées non retenues au cahier des charges.

Les options et variantes sont pas mal toutes dans le département des pièces et du choix du VPN.

6.2.3 Mesures prises pour respecter les contraintes et leurs conséquences.

La plus grosse mesure prise de notre bord est la grosseur de notre projet en raison des contraintes de temps, nous n'avons simplement pas assez de temps pour accomplir un projet de taille énorme.

6.2.4 Outils d'installation, de maintenance à prévoir.

- Installation de Openwrt
- Installation d'un VPN

6.2.5 Décomposition en modules, sous-ensembles.

Physique

- Raspberry Pi 4



Figure 1.1.3: Canakit complet

Périphériques

- WiFi Dongle



Figure 1.1.4: Wireless WiFi USB Dongle Stick Adapter.

- Écran Tactile

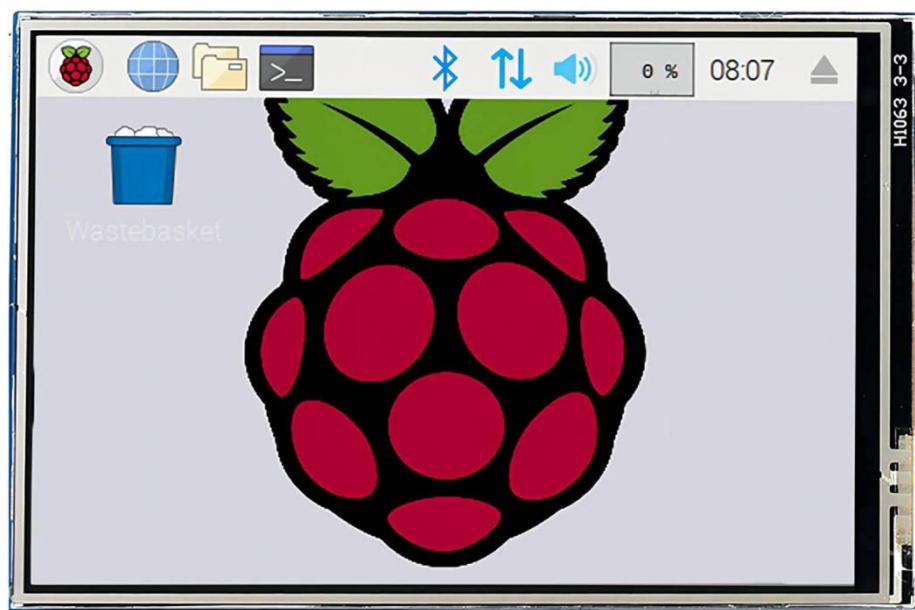


Figure 1.1.5 : Écran Tactile.

-Clavier



Figure 1.1.6 : Clavier.

Software

- ProtonVPN



Figure 1.1.7 : Logo de VPN.

6.2.6 Prévisions de fiabilité.

Notre objectif est d'atteindre 99.98 % de disponibilité et de fiabilité. Le 0.02 manquant est en raison qu'on ne peut jamais vraiment être 100% en informatique.

6.2.7 Perspectives d'évolution technologique.

Ce projet utilise de la technologie assez récente, donc elle sera bonne pour longtemps encore.

Il faudrait des découvertes immenses pour rendre ce projet inutile.

7. Planification et conception du projet.

7.1 L'outil utilise pour planifier et exécuter le projet.

On a choisi l'outil JIRA pour gérer notre projet en bas des captures d'écran qui montre l'utilisation de cet outil.

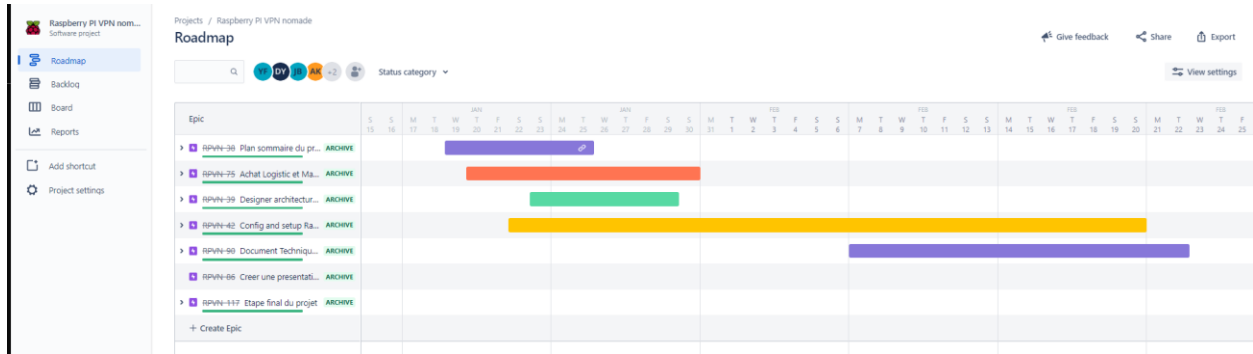


Figure 1.1.8 : Dashboard Jira pour les taches déjà faites.

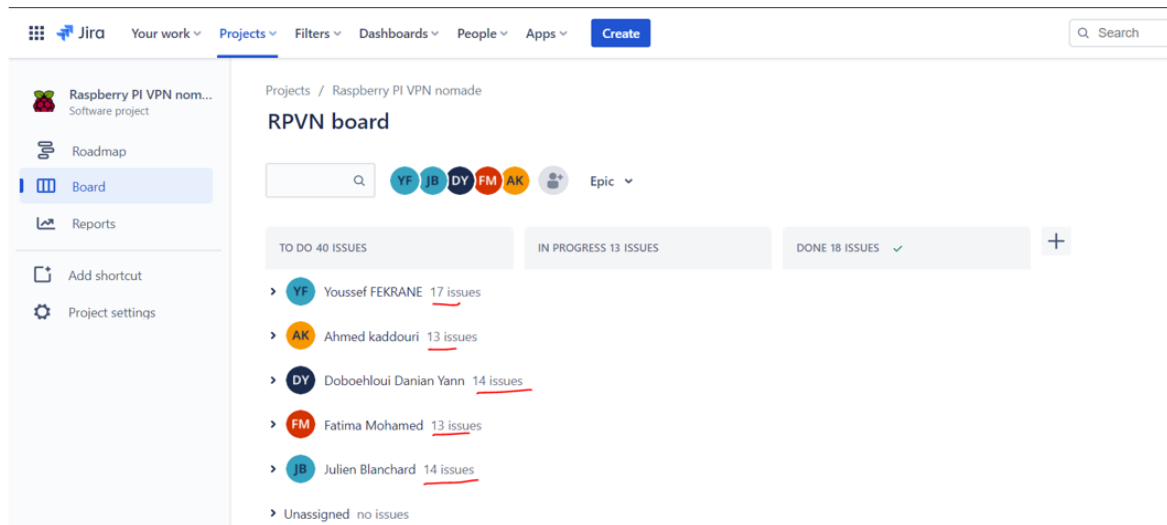


Figure 1.1.9 : Dashboard Jira pour les taches attribuées.

Projects / Raspberry PI VPN nomade

RPVN board

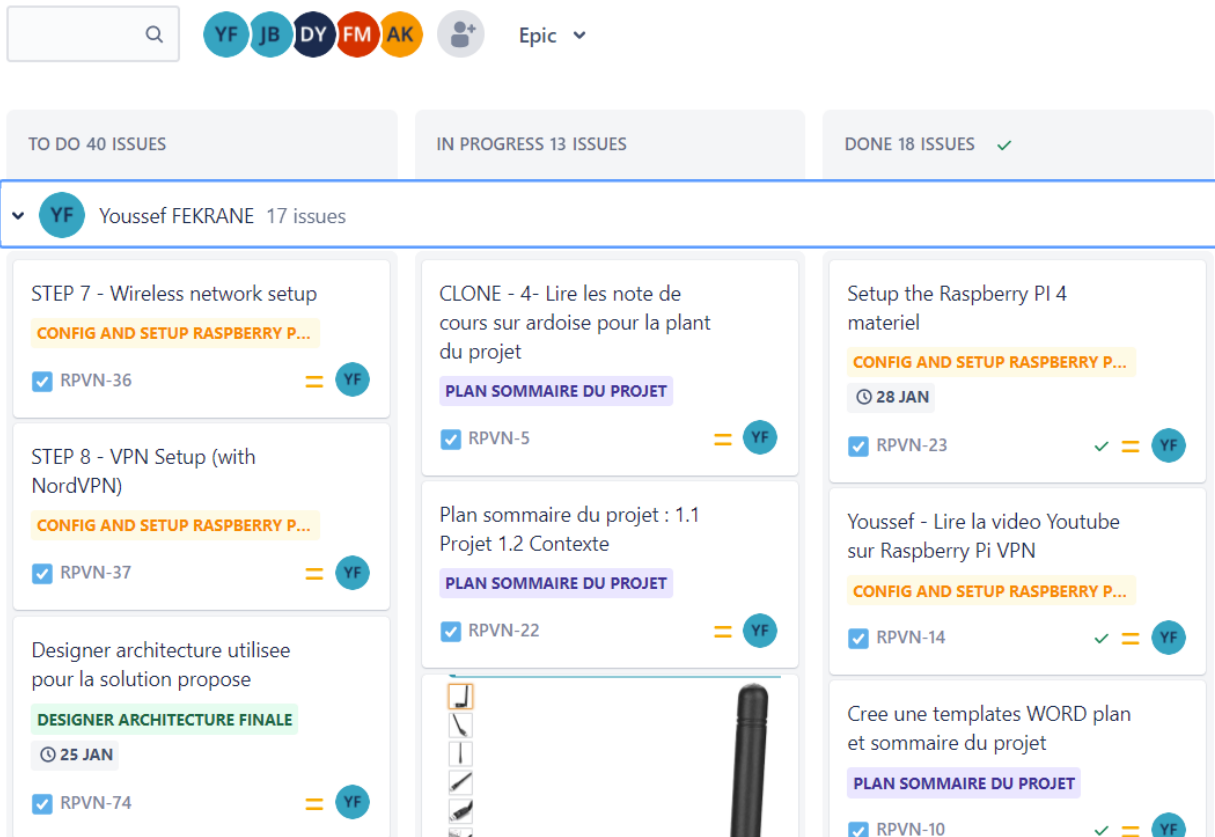


Figure 1.2.0 : Dashboard Jira pour les tâches assignées, en cours et terminées.

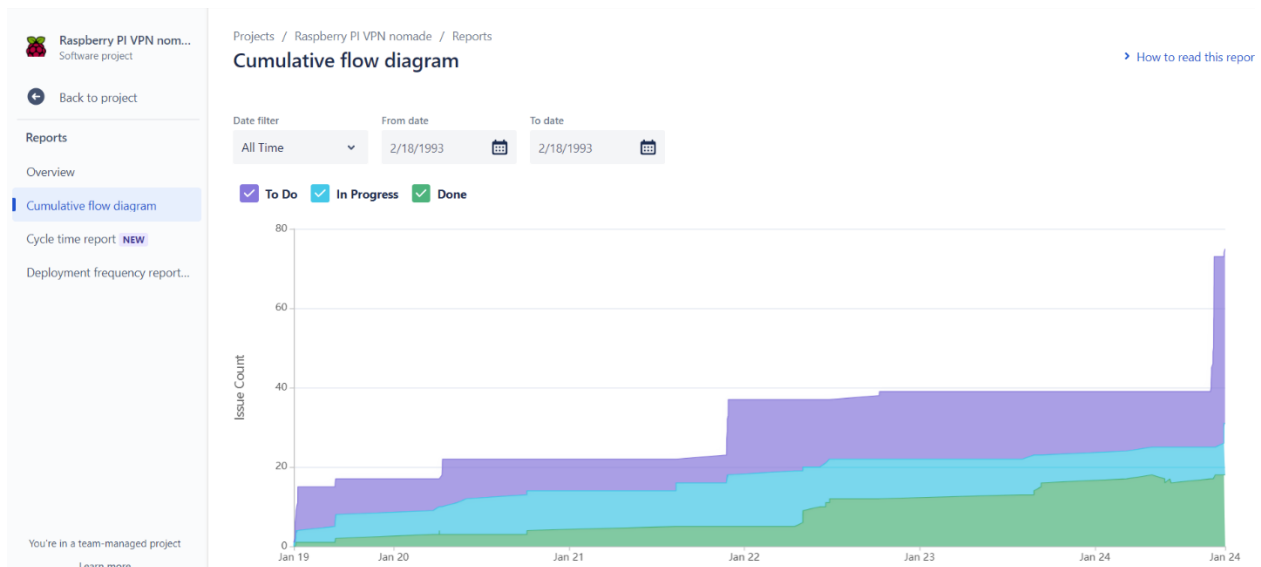
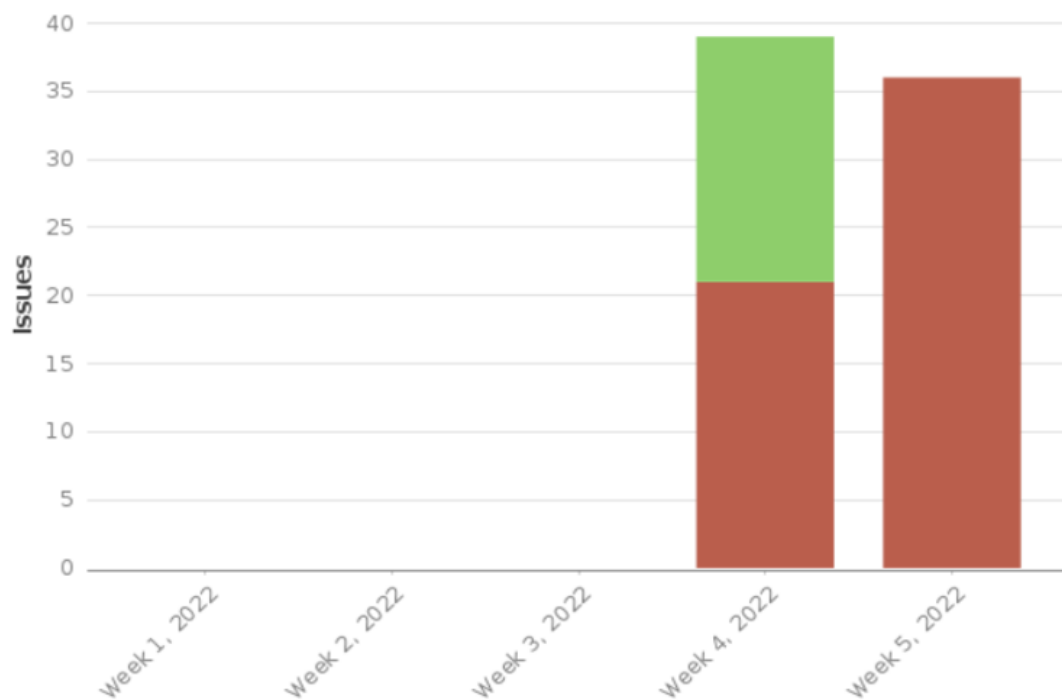


Figure 1.2.1 : Dashboard Jira pour l'organigramme cumulatif.

7.2 Exemples des Dashboard qui concerne la planification du projet.

Recently Created Chart: Raspberry PI VPN nomade



Total Issues: 75

Period: last 30 days (grouped Weekly)

Figure 1.2.2 : Dashboard Jira pour un graphique à barres.

Pie Chart: Raspberry PI VPN nomade

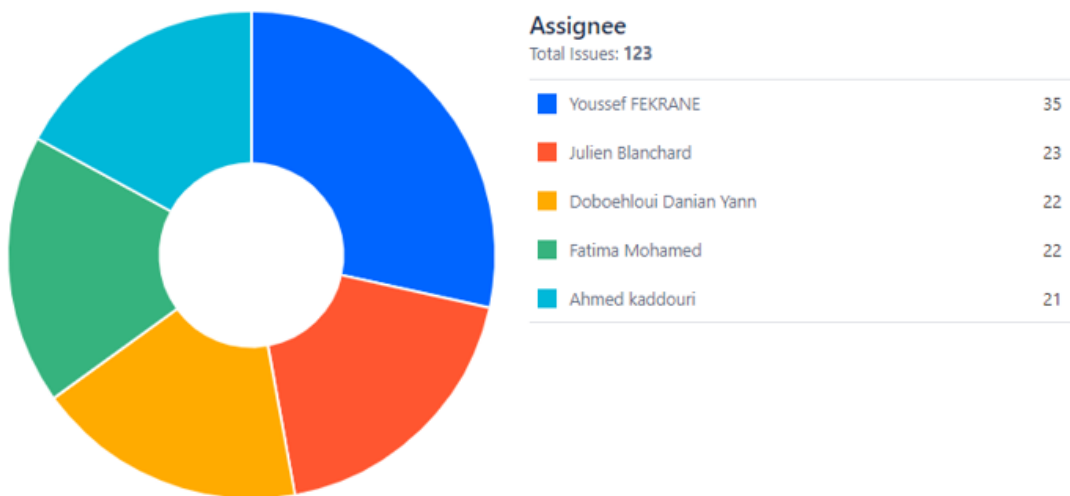


Figure 1.2.3 : Dashboard Jira pour un graphique circulaire.

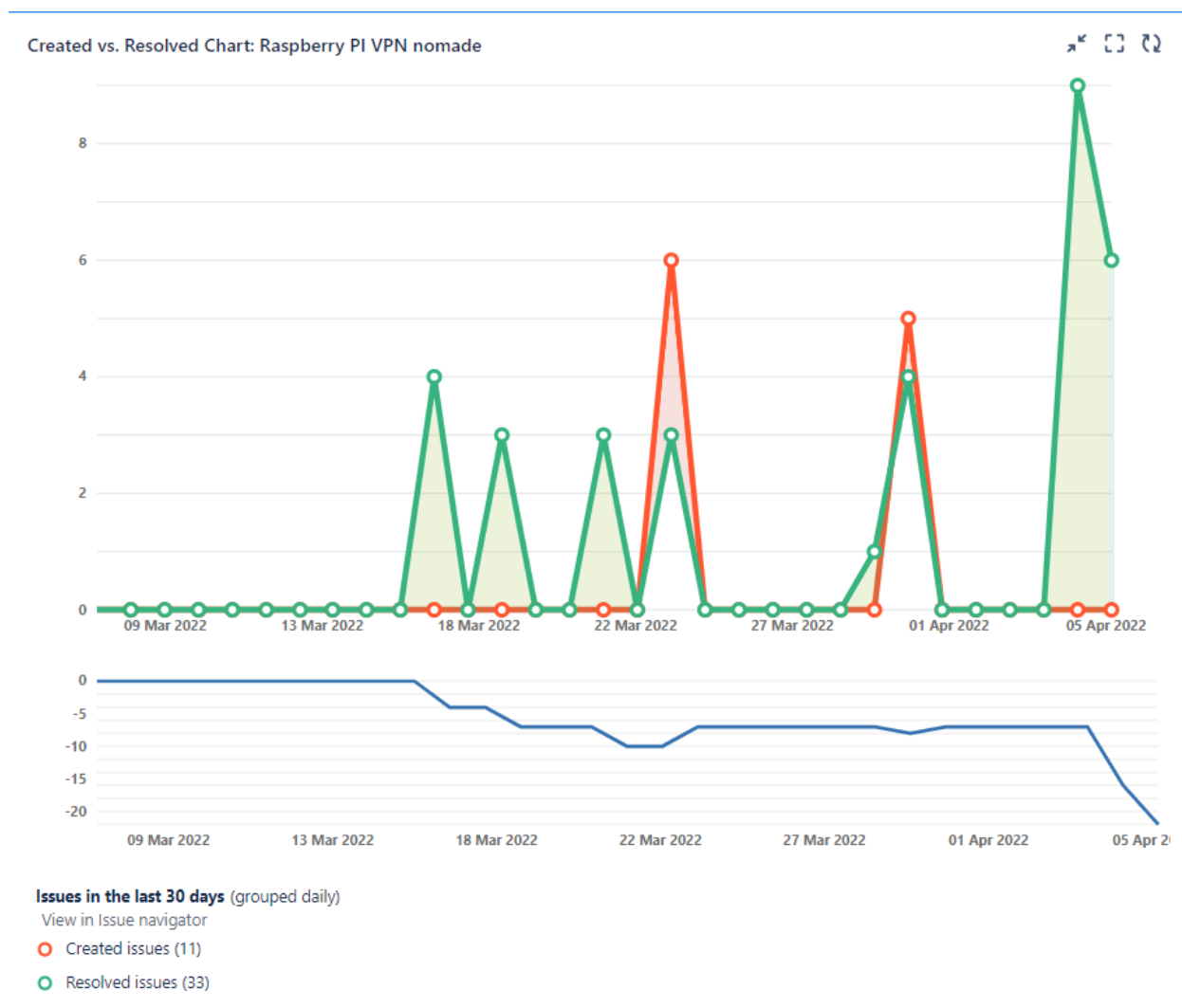


Figure 1.2.4 : Dashboard Jira pour un graphique en courbe.

8. Préconisation technique et organisationnelle.

8.1 Mise en œuvre de la solution.

8.1.1 Etape1 : Rassembler Raspberry Pi.

Le contenu est composé du Raspberry Pi, de son boîtier, d'une alimentation USB, d'un câble mini HDMI vers le port HDMI, de dissipateurs, d'un ventilateur et d'une carte micro SD accompagnée de son adaptateur USB pour y installer le système d'exploitation.



Figure 1.2.5 : Ensemble de Raspberry Pi

- ❖ Pour commencer, nous allons prendre le Raspberry 4 ainsi que les dissipateurs (Il s'agit des petits carrés métalliques argentés).
- ❖ Les dissipateurs vont alors venir se coller à l'aide de la surface collante sur les puces électroniques qui génèrent le plus de chaleur. Voir schémas ci-dessous.
- ❖ Puis le ventilateur nous avons deux fils rouge et noir donc nous connecterons les câbles rouges et noirs comme expliqué sur notre petit schéma.



Figure 1.2.6 : Composant de Raspberry Pi.

- ❖ A présent, prenons la partie basse du boîtier et venons y loger le nano ordinateur en prenant bien soin de l'insérer dans le bon sens. Les différentes encoches vous guideront.
- ❖ Tout est bien détaillé dans le schéma suivant du début à la fin de l'assemblage du kit.

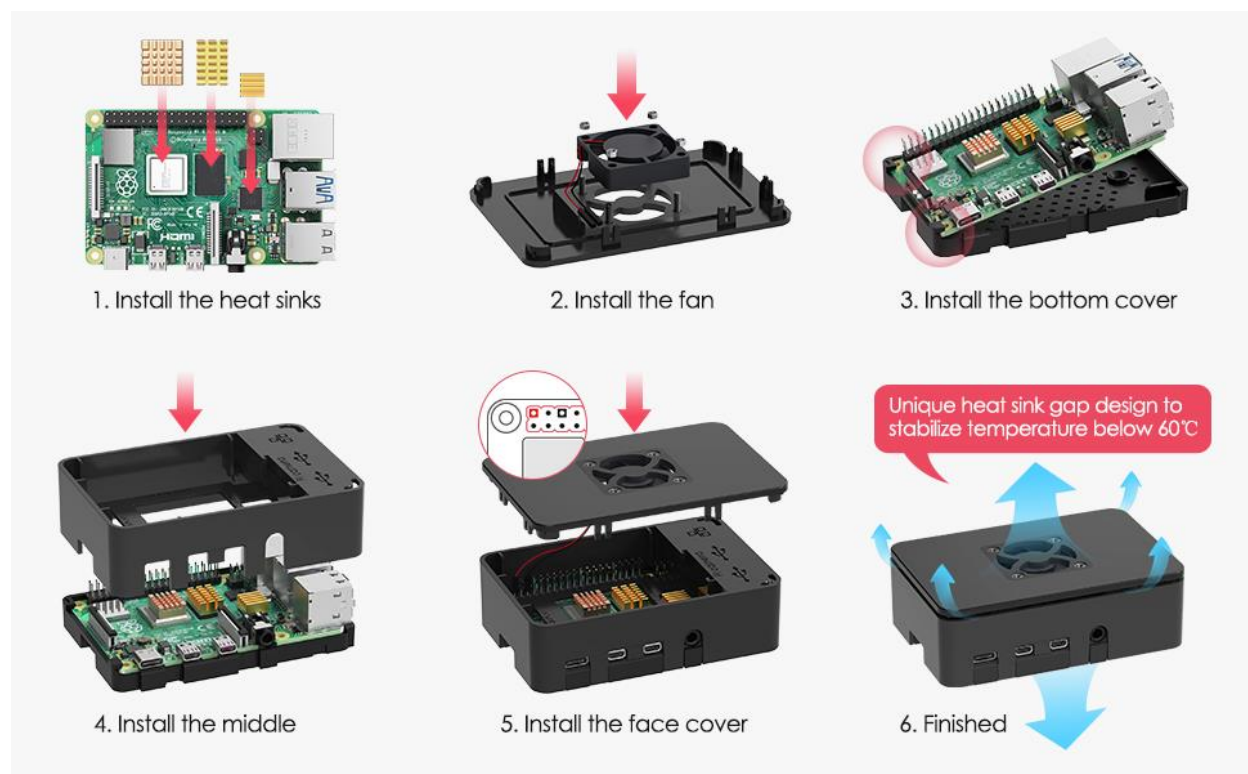


Figure 1.2.7 : Les étapes pour rassembler le Raspberry Pi.

8.1.2 Etape2 : Introduction.

Notre groupe 2 a pour projet de configurer un Raspberry-pi comme un routeur sécurisé wifi pour une connexion plus sécurée.

Le Raspberry pi est un puissant outil qui peut être utilise pour plusieurs taches mais spécialement ici comme un routeur. Avec la menace informatique grandissante, étant des spécialistes en cyber nous estimons qu'il est primordial d'utiliser un Raspberry-pi pour une toute connexion externe afin d'avoir une connexion plus sécurée et éviter ainsi le vol d'identité.

8.1.3 Etape3 : Ce dont on a besoin.

Outre le Raspberry Pi 4, vous aurez également besoin des éléments et applications suivants :

- Un appareil pour connecter la carte MicroSD incluse avec le Raspberry Pi (un adaptateur SD est inclus, mais si votre ordinateur n'a pas de fente pour carte SD, il existe d'autres adaptateurs MicroSD pour d'autres formats de port comme USB et USB-C)
- Un flasher Raspberry Pi OS (nous utiliserons Pi Imager, mais Etcher fonctionne également)
- Un adaptateur sans fil USB
- L'OS OpenWRT spécifique à votre Pi
- VPN pour le Pi (ProtonVPN est utilisé)

8.1.4 Etape4 : La grande image.

On utilise un Raspberry Pi 4, qui inclut un port à Ethernet, qui a une carte réseaux déjà installée, et on y ajoute un adaptateur USB wifi, ce qui nous donne deux interfaces Wifi.

La carte Wifi interne sera utilisée pour connecter le Raspberry pi 4 à n'importe quel wifi public, l'adaptateur wifi est utilisé pour qu'on puisse connecter nos appareils au Raspberry Pi 4.

Finalement, le port Ethernet est utilisé pour faire les configurations de bases.

8.1.5 Etape5 : Installation OpenWRT system.

La 1^{ère} étape est d'installer un system pour la gestion des retours en utilisant notre Raspberry PI donc on a choisi d'utiliser OpenWRT car c'est un système open source est-il a plusieurs versions pour les différentes version du PI.

Donc on doit aller sur ce [lien](#) et télécharger le Framework OpenWRT qui va avec notre PI.

Après téléchargement on doit installer cela sur une carte SSD en utilisant l'utilitaire **Raspberry Pi Imager**.



Figure 1.2.8 : Carte et lecture SSD.

Ensuite on branche notre lecteur Carte SSD dans le port USB et on lance l'utilitaire Raspberry Pi imager.

On clique sur Choose OS.

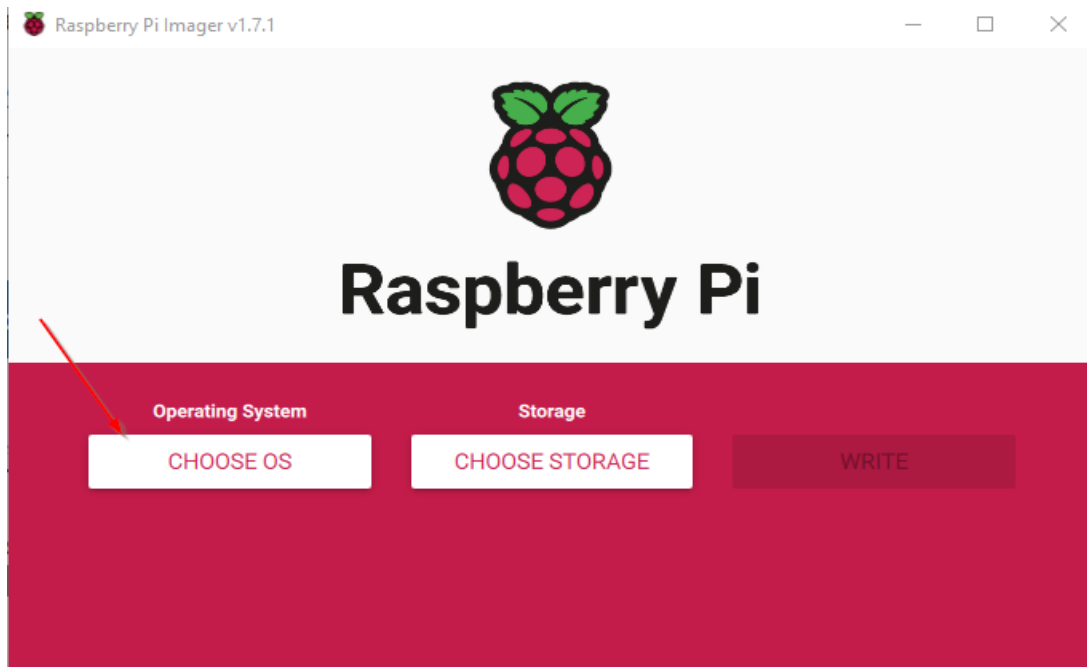


Figure 1.2.9: Raspberry Pi imager.

On choisit Use custom.

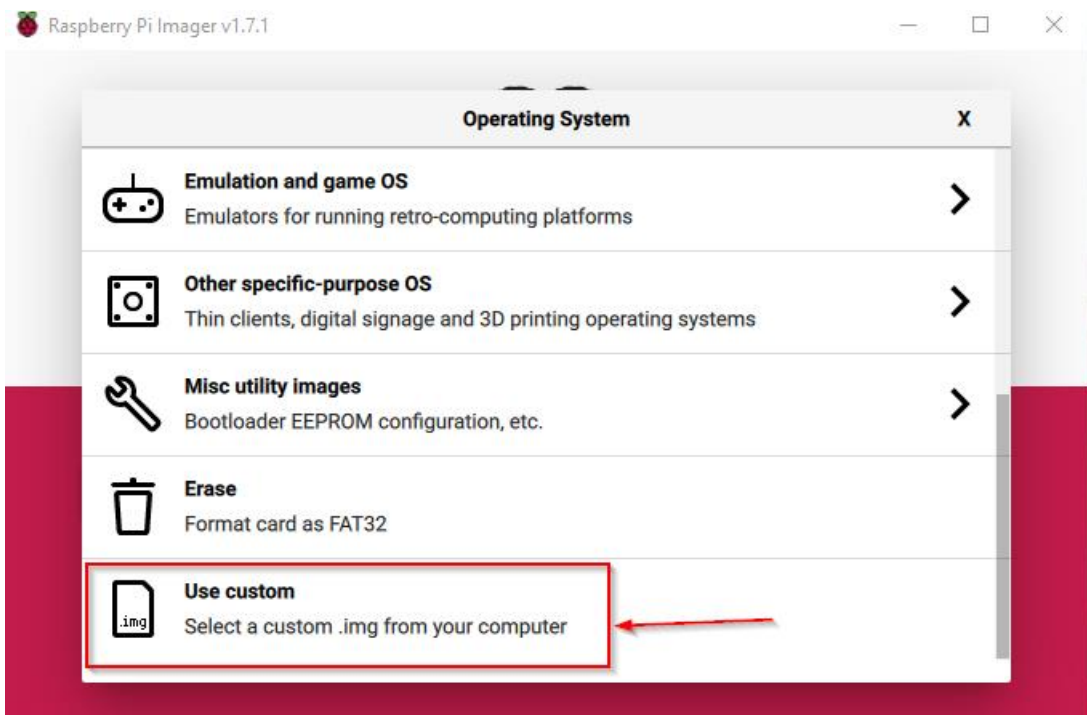


Figure 1.3.0 : Lors du choix de la carte.

On cherche notre image du OpenWRT

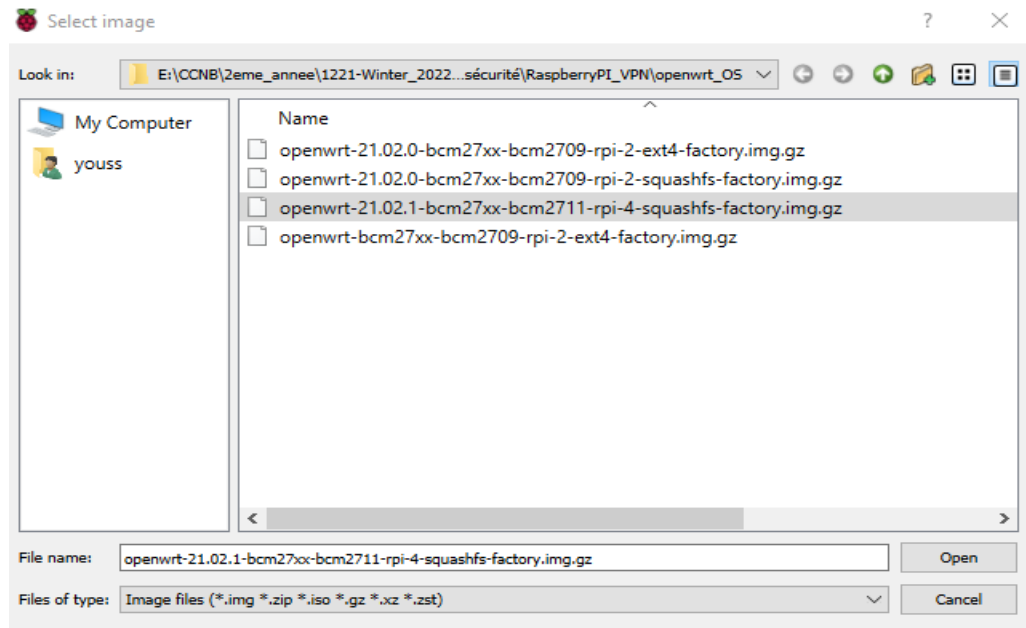


Figure 1.3.1 : Lors du choix pour l'image du OpenWRT.

Nous avons choisi le stockage pour notre carte SSD.

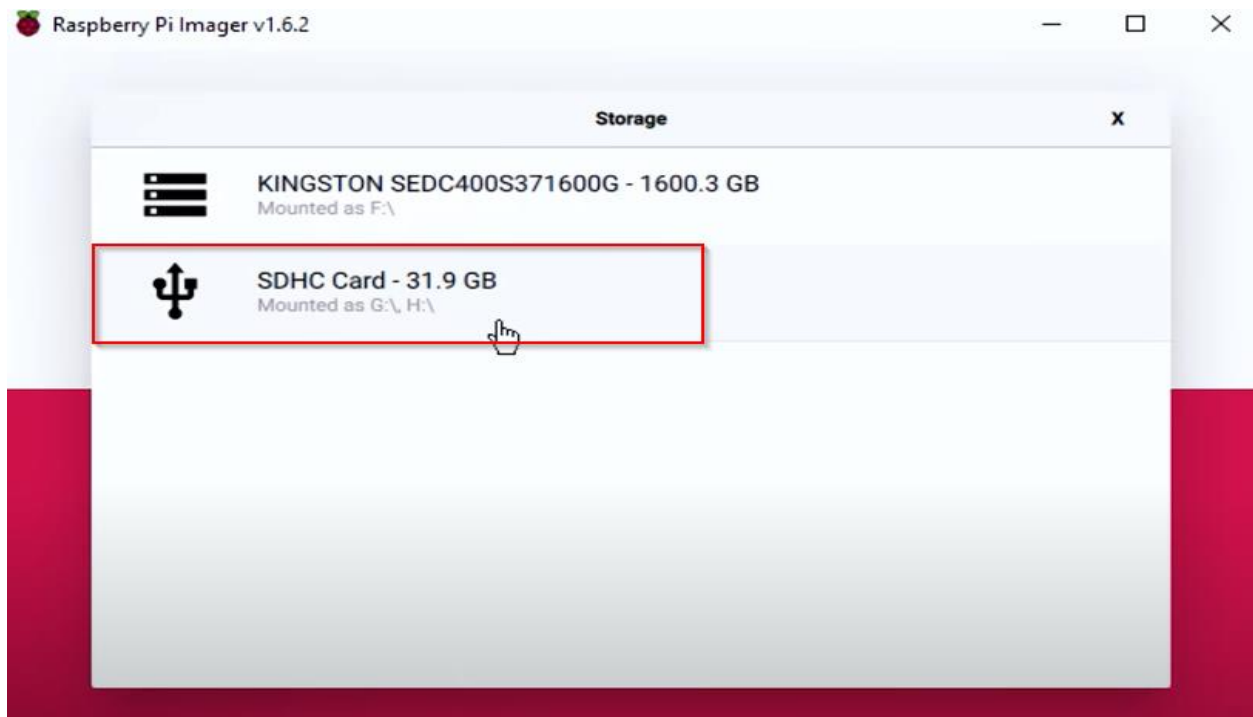


Figure 1.3.2: Lors du choix du stockage pour la carte SSD.

Après on clique sur Write (Écrire).



Figure 1.3.3 : Lors du choix pour l'écriture.

Et après on clique sur Yes.

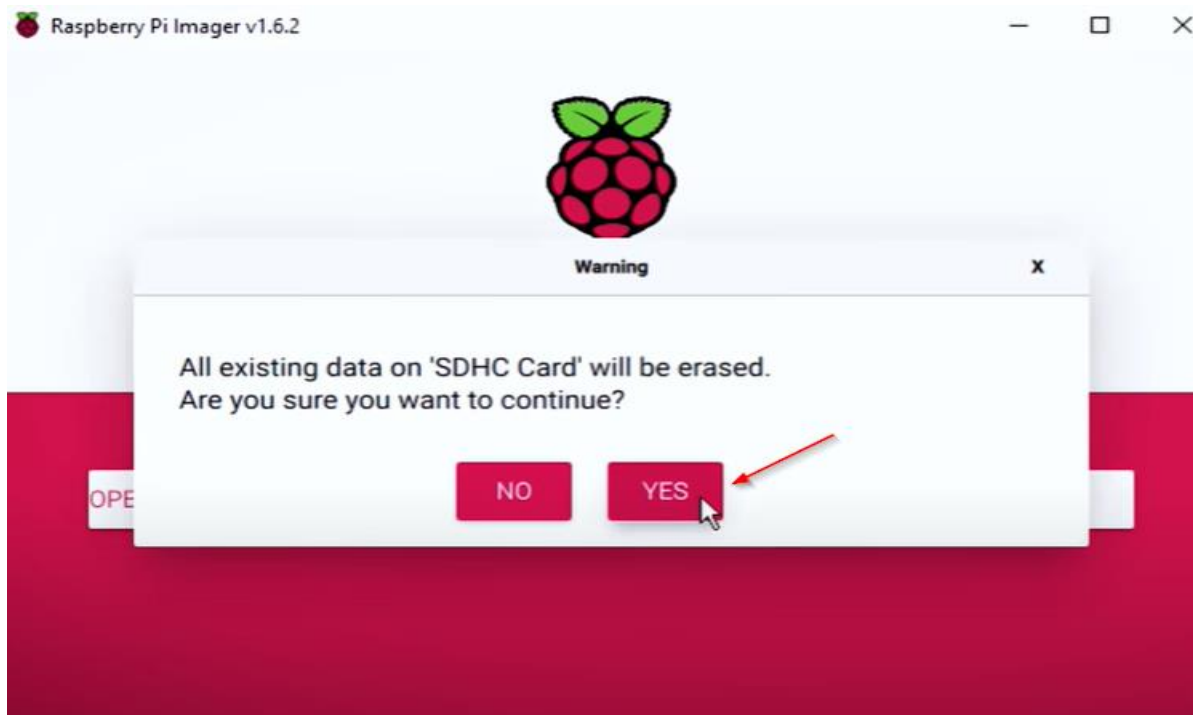


Figure 1.3.4: Lors du choix pour accepte l'installation.

Et là on peut prendre une pause du Café ☕! !!!

Après on clique sur continue.

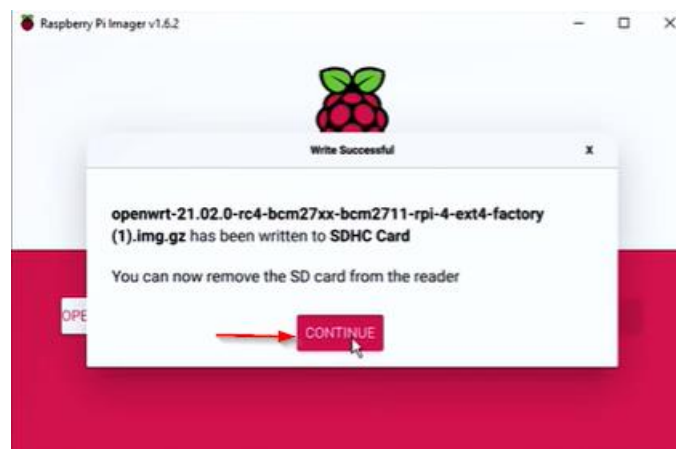


Figure 1.3.5 : Nous continuons l'installation.

Après on enlève notre carte SSD du USB reader et on mettra notre carte sur Raspberry Pi car la carte est prête avec le système OpenWRT.

8.1.6 Etape6 : Connecter sur Pi.

Pour vous connecter à votre Raspberry Pi, vous avez besoin d'un câble Ethernet qui connectera votre ordinateur au Raspberry Pi. Cela vous permettra d'accéder à l'adresse IP de votre Raspberry Pi.

❖ Pour connaître l'adresse IP du Raspberry Pi, il suffit de rentrer la commande suivante dans le terminal : **ifconfig**

- Après avoir trouvé l'adresse IP de votre Raspberry, vous pouvez maintenant vous connecter à cette adresse depuis votre invite de commande.
- Ouvrez simplement un terminal et entrez la commande suivante :

ssh root@votre_adress_ip

- Si vous avez un message vous demandant si vous êtes sûr et que vous souhaitez continuer, entrez « yes » et appuyez sur la touche entrée.

- Et voilà, vous devriez être connecté en SSH dans l'invite de commande.

```
C:\Users\celin>ssh root@192.168.1.1

BusyBox v1.33.1 (2021-08-31 22:20:08 UTC) built-in shell (ash)

 _ _ _ _ _
|_ _ _ _ _|_ _ _ _ _|_ _ _ _ _|_ _ _ _ _|_ _ _ _ _|
|_| W I R E L E S S   F R E E D O M

-----
OpenWrt 21.02.0, r16279-5cc0535800
-----
=== WARNING! =====
There is no root password defined on this device!
Use the "passwd" command to set up a new password
in order to prevent unauthorized SSH logins.
-----
```

Figure 1.3.6 : Première connexion du Raspberry pi.

- Ensuite vous pouvez modifier vous-même votre mot de passe directement en tapant : **passwd.**

```
root@OpenWrt:~# passwd
Changing password for root
New password:
Bad password: too weak
Retype password:
passwd: password for root changed by root
root@OpenWrt:~#
```

Figure1.3.7: Changement de mot de passe.

❖ Autre méthode pour se connecter en SSH au Raspberry Pi depuis un ordinateur Windows, nous aurons besoin du logiciel Putty.

- Il est téléchargeable en suivant le lien suivant :

[Télécharger PuTTY: dernière version \(0.76\) \(greenend.org.uk\)](https://greenend.org.uk/download/putty/)

- Une fois PuTTY téléchargé et installé, au lancement, la fenêtre suivante s'ouvre.

- Host name (ou adresse IP): Il s'agit de l'adresse IP du Raspberry Pi.
- Port : Laisser la valeur par défaut, 22.
- Connection type : Cocher la case SSH.
- Vous pouvez sauvegarder la session en entrant un nom dans Saved Sessions.
- Puis en cliquant sur Save. Ça peut s'avérer très pratique pour éviter de retaper l'IP à chaque ouverture de Putty.

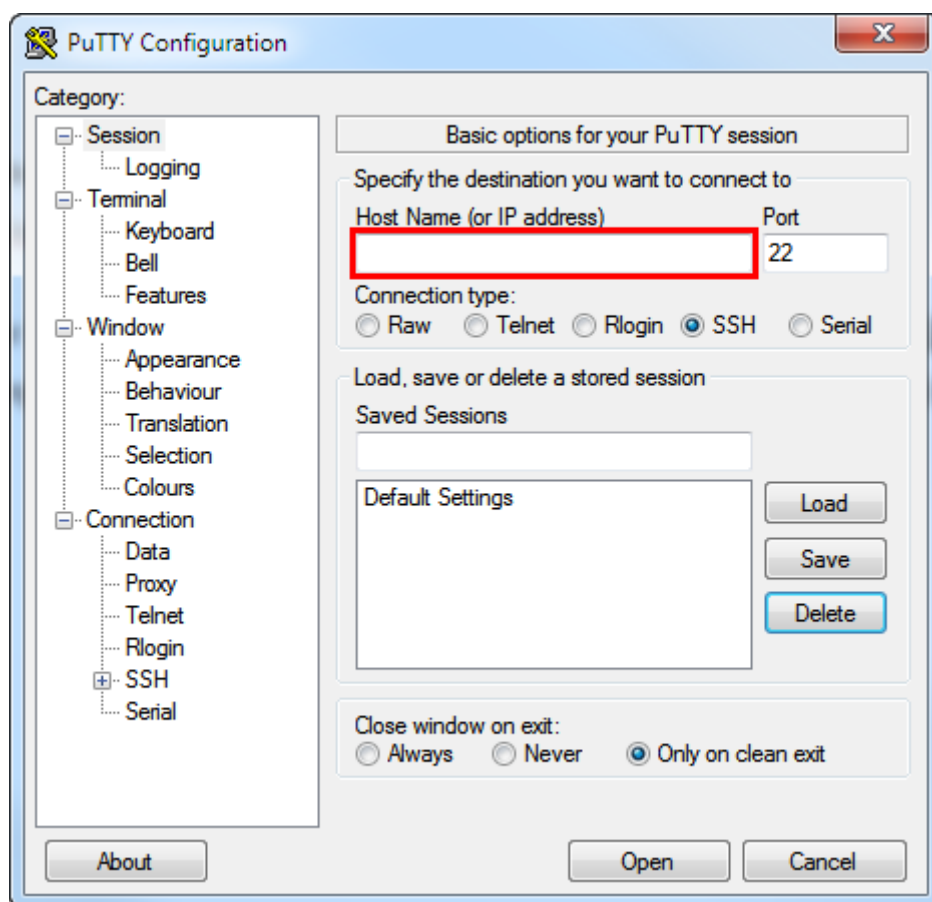


Figure 1.3.8 : Connexion de Raspberry pi avec PuTTY.

- Vous pouvez alors cliquer sur « Open » et une fenêtre de sécurité « Security Alert » apparaît, cliquez simplement sur « Yes ».
- Une fenêtre de ligne de commande devrait alors apparaître vous demandant « login as : ». Entrez « pi » puis appuyez sur la touche entrée.
- Entrez ensuite votre mot de passe et validez avec la touche entrée de votre clavier.

8.1.7 Etape7 : OpenWRT Basique configuration

Premièrement il faut connecter l'internet au Raspberry-pi et on plug in le Raspberry pi power.

Par défaut, l'adresse IP du Raspberry pi IP is 192.168.1.1

Control panel, ethernet port, properties, change automatic to manual (screen)

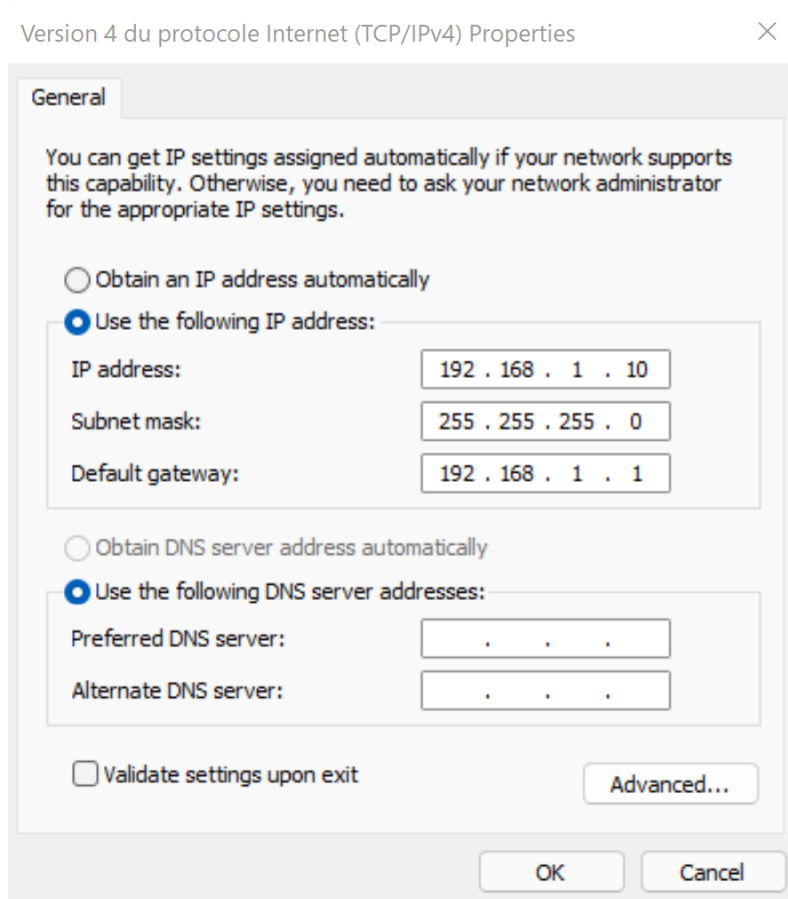


Figure 1.3.9 : configuration basique d'openwrt.

2-connexiion via SSH

On entre **ssh root @192.168.1.1**, ensuite yes et on Est connecter à l'Open Wrt mais à mon niveau le Raspberry est déjà connecté, j'ai juste à mettre mon mot de passe pour le connecter.

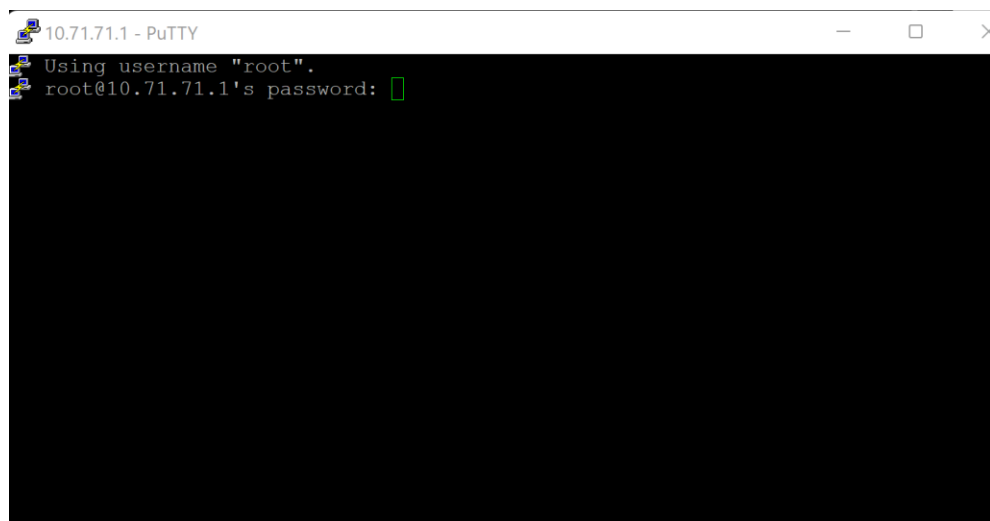


Figure 1.4.0 : Connexion de Raspberry pi avec pytty

Je vous conseille de toujours changer le mot de passe avec cette commande « **psswd** » après s'être connecter par SSH.

Ensuite après la connexion on entre dans le répertoire `cd /etc/config`.

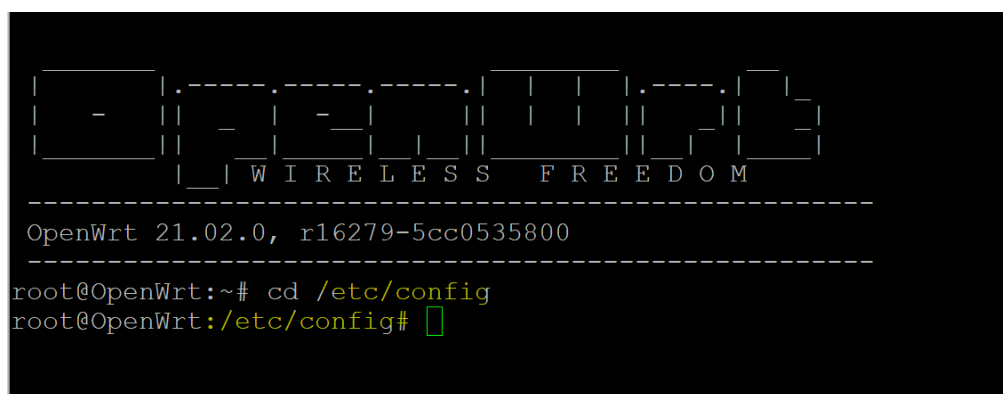


Figure 1.4.1 : Étape de configuration de Raspberry pi

Nous sommes bien dans le répertoire `cd /etc/config`.

On tape Ls pour voir les dossiers (firewall, network,wireless), on Créer également des backups pour Wireless network and firewall.

```
BusyBox v1.33.1 (2021-08-31 22:20:08 UTC) built-in shell (ash)

|_| .-----.-----.|_|_|_|_|_| .-----.|_|_|
|_| _ | _ | _||_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|
|_| W I R E L E S S   F R E E D O M
-----
OpenWrt 21.02.0, r16279-5cc0535800
-----

root@OpenWrt:~# cd /etc/config
root@OpenWrt:/etc/config# ls
dhcp                          luci                       openvpn_recipes             uhttpd
dropbear                      network                   rpcd                        wireless
firewall                     network.bk                system                      wireless.bk
firewall.bk                  openvpn                    ucitrack
root@OpenWrt:/etc/config# █
```

Figure 1.4.2: Affichage du dossier firewall, network et Wireless.

Premier changement a effectué après la création des dossiers backup est le dossier **NETWORK** :

```
config interface 'loopback'
    option device 'lo'
    option proto 'static'
    option ipaddr '127.0.0.1'
    option netmask '255.0.0.0'

config globals 'globals'
    option ula_prefix 'fd08:7c51:2ee7::/48'

config device
    option name 'br-lan'
    option type 'bridge'
    list ports 'eth0'

config interface 'lan'
    option device 'br-lan'
    option proto 'static'
    option ipaddr '10.71.71.1'
    option netmask '255.255.255.0'
    option ip6assign '60'
    option force_link '1'

- network 1/32 3%
```

Figure 1.4.3: Modifications du dossiers backup et NETWORK

Ici on a effectué des changements sur le **Lan network**, l'adresse IP car elle est connue de tous.

Ensuite on crée une nouvelle interface et on lui donne le nom de 'wwan' et on ajoute des options.

```
config device
    option name 'br-lan'
    option type 'bridge'
    list ports 'eth0'

config interface 'lan'
    option device 'br-lan'
    option proto 'static'
    option ipaddr '10.71.71.1'
    option netmask '255.255.255.0'
    option ip6assign '60'
    option force_link '1'

config interface 'wwan'
    option proto 'dhcp'
    option peerdns '0'
    option dns '1.1.1.1 8.8.8.8'
```

Figure 1.4.4: Modifications du Lan network

Ensuite on crée une nouvelle interface et on lui donne le nom de 'vpnclient' et on ajoute des options. Et on Enregistre SAVE.

```
config interface 'lan'
    option device 'br-lan'
    option proto 'static'
    option ipaddr '10.71.71.1'
    option netmask '255.255.255.0'
    option ip6assign '60'
    option force_link '1'

config interface 'wwan'
    option proto 'dhcp'
    option peerdns '0'
    option dns '1.1.1.1 8.8.8.8'

config interface 'vpnclient'
    option proto 'none'
    option device 'tun0'

root@OpenWrt:/etc/config#
```

1.4.5 : Création d'une nouvelle interface

On fait un changement sur Wireless, on va changer l'option input dans 'Wan' en ACCEPT et on sauve, REBOOT RASPERRY-PI.

```
config zone 'lan'
    option name 'lan'
    option input 'ACCEPT'
    option output 'ACCEPT'
    option forward 'ACCEPT'
    list network 'lan'

config zone 'wan'
    option name 'wan'
    option input 'ACCEPT'
    option output 'ACCEPT'
    option forward 'REJECT'
    option masq '1'
    option mtu_fix '1'
    list network 'wan'
    list network 'wan6'
    list network 'wwan'
    list device 'tun+'

```

Figure 1.4.6: Changement de Wireless

8.1.8 Étape 8: Configuration reseau OpenWRT.

La prochaine étape est de modifier le fichier Firewall la même façon qu'on a modifier le fichier Network.

1^{er} chose a faire on se déplace sur le chemin /etc/config avec la commande : `# cd /etc/config/`

Après vi network `# vi network`

On changera l'interface lan option ipaddr de 192.168.1.1 au 10.71.71.1 et on ajoute config

interface 'wwan' et config interface 'vpnclient'

```
option netmask '255.0.0.0'

config globals 'globals'
option ula_prefix 'fde9:bb89:c742::/48'

config device
option name 'br-lan'
option type 'bridge'
list ports 'eth0'

config interface 'lan'
option device 'br-lan'
option proto 'static'
option ipaddr '10.71.71.1'
option netmask '255.255.255.0'
option ip6assign '60'
option force_link '1'

config interface 'wwan'
option proto 'dhcp'
option peerdns '0'
option dns '1.1.1.1 8.8.8.8'

config interface 'vpnclient'
option ifname 'tun0'
option proto 'none'
```

1

change 192.168.1.1 to 10.71.71.1

Add this

2

Add this

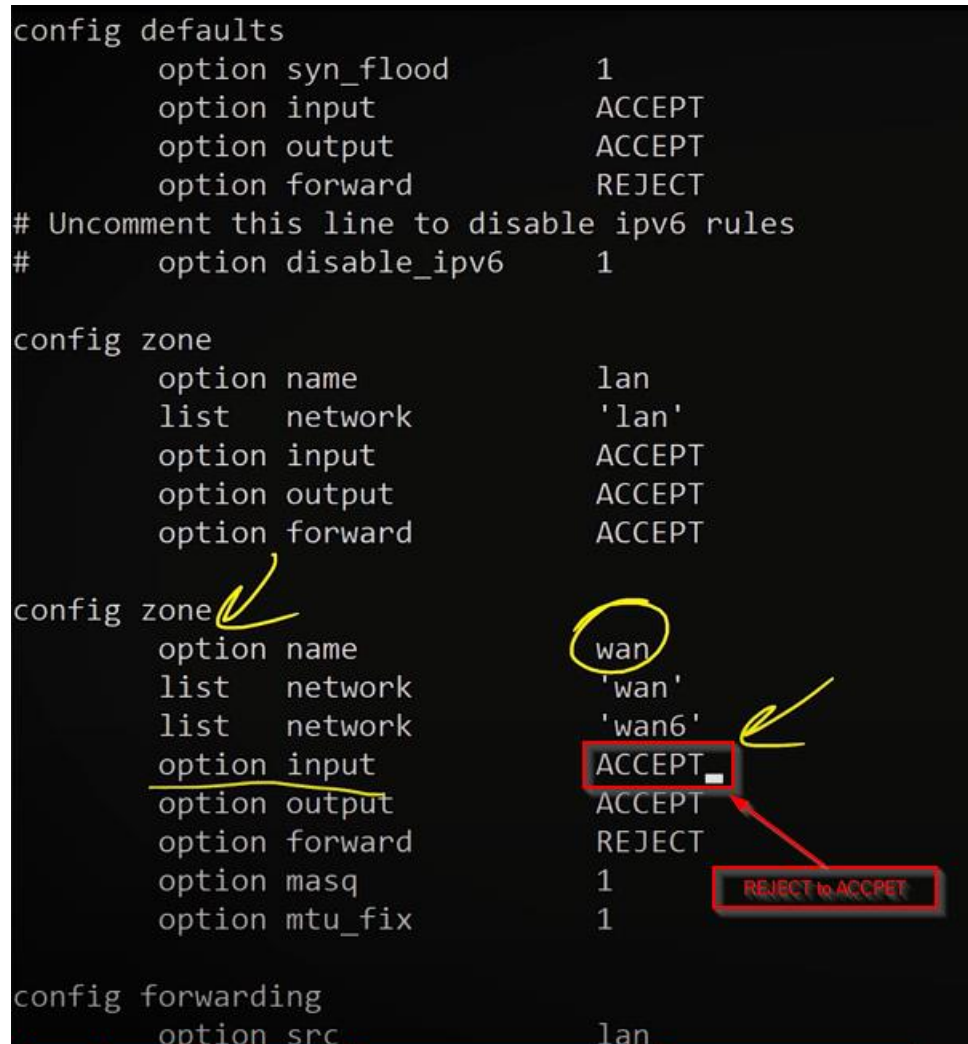
3

Add this

Figure 1.4.7: Changement de l'interface LAN

Après modification on enregistre le fichier Esc + :wq

Entrer la commande suivante : Vi Firewall



```
config defaults
    option syn_flood      1
    option input          ACCEPT
    option output          ACCEPT
    option forward         REJECT
# Uncomment this line to disable ipv6 rules
#    option disable_ipv6   1

config zone
    option name           lan
    list network          'lan'
    option input          ACCEPT
    option output          ACCEPT
    option forward         ACCEPT

config zone
    option name           wan
    list network          'wan'
    list network          'wan6'
    option input          ACCEPT
    option output          ACCEPT
    option forward         REJECT
    option masq            1
    option mtu_fix         1

config forwarding
    option src             lan
```

Figure 1.4.8: Modification du fichier firewall

Dans ce fichier, il faut faire un seul ajustement: allez dans “config zone” et changer “option input: REJECT” à “option input: ACCEPT”

```
config zone
    option name          lan
    list network         'lan'
    option input          ACCEPT
    option output         ACCEPT
    option forward        ACCEPT

config zone
    option name          wan
    list network         'wan'
    list network         'wan6'
    option input          ACCEPT
    option output         ACCEPT
    option forward        REJECT
    option masq           1
    option mtu_fix        1
```

Figure 1.4.9: Réglage du fichier dans la zone de configuration

Sauvegarder et quitter avec “: wq”

Il faut maintenant faire un reboot du Pi Pour compléter l’initialisation d’une connexion sans-fil.

Entrer la commande suivante: reboot

La connexion entre votre Pi et votre ordinateur va se couper. Avant d’essayer de se reconnecter, il faut reconfigurer notre câble réseau LAN pour qu’il se connecte automatiquement, grâce au DHCP.

Control Panel > Network and Internet > Network Status and Tasks > Ethernet > Properties > IPv4

Changer de manuel à automatique

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

☒ Obtain an IP address automatically

☐ Use the following IP address:

IP address:	<input type="text" value="."/> <input type="text" value="."/> <input type="text" value="."/>
Subnet mask:	<input type="text" value="."/> <input type="text" value="."/> <input type="text" value="."/>
Default gateway:	<input type="text" value="."/> <input type="text" value="."/> <input type="text" value="."/>

☒ Obtain DNS server address automatically

☐ Use the following DNS server addresses:

Preferred DNS server:	<input type="text" value="."/> <input type="text" value="."/> <input type="text" value="."/>
Alternate DNS server:	<input type="text" value="."/> <input type="text" value="."/> <input type="text" value="."/>

Figure 1.5.0 : Reconfiguration du câble réseau LAN

Maintenant, reconnecter-vous au Pi. Mais au lieu d'entrer avec le Gateway IP vous avez écrit dans Control Panel au début de l'installation, vous allez entrer avec le Gateway IP vous avez écrit dans le fichier Network

ssh [root@192.168.1.1](#) ✗

ssh [root@10.71.71.1](#) ✓

Figure 1.5.2 : Résultat après la commande : `Vi /etc/config/wireless`

On doit faire les changements suivants :

Option channel : 36 -> 7

Option hwmode : 11a -> 11g

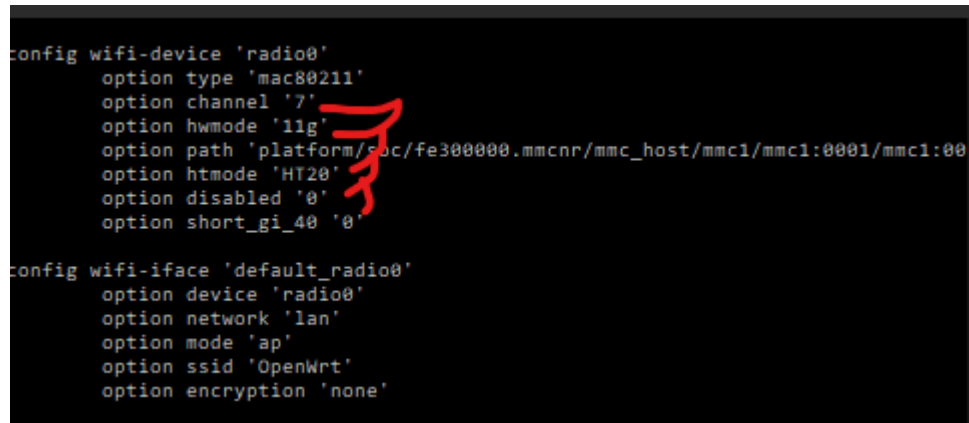
Option htmode : VHT80 -> HT20

Option disabled : 1 -> 0

Et finalement on ajoute une option d'extra juste en dessous d'option disabled : option

short_gi_40 '0'

Ce qui vous donnera le résultat suivant :



```
config wifi-device 'radio0'
    option type 'mac80211'
    option channel '7'
    option hwmode '11g'
    option path 'platform/soc/fe300000.mmcnr/mmc_host/mmc1/mmc1:0001/mmc1:00
    option htmode 'HT20'
    option disabled '0'
    option short_gi_40 '0'

config wifi-iface 'default_radio0'
    option device 'radio0'
    option network 'lan'
    option mode 'ap'
    option ssid 'OpenWrt'
    option encryption 'none'
```

Figure 1.5.3: Ajout d'option supplémentaire

On frappe la touche **ESC : wq**

Ce qui nous fait sauf garder nos configurations.

Ensuite on lance les deux commandes suivantes, si on n'a pas d'erreur, sa bien fonctionner.

Uci commit wireless

Wifi

```
root@OpenWrt:~# vi /etc/config/wireless
root@OpenWrt:~# root@OpenWrt:~# uci commit wireless
root@OpenWrt:~# wifi
root@OpenWrt:~#
```

Figure 1.5.4: Résultat du bon fonctionnement de la configuration

Ensuite, on ouvre chrome et on va se connecter à notre Raspberry pi en utilisant l'interface web en écrivant l'adresse IP de notre pi dans la barre de recherche. (10.71.71.1)

On utilise le même login qu'on a configurer au début.

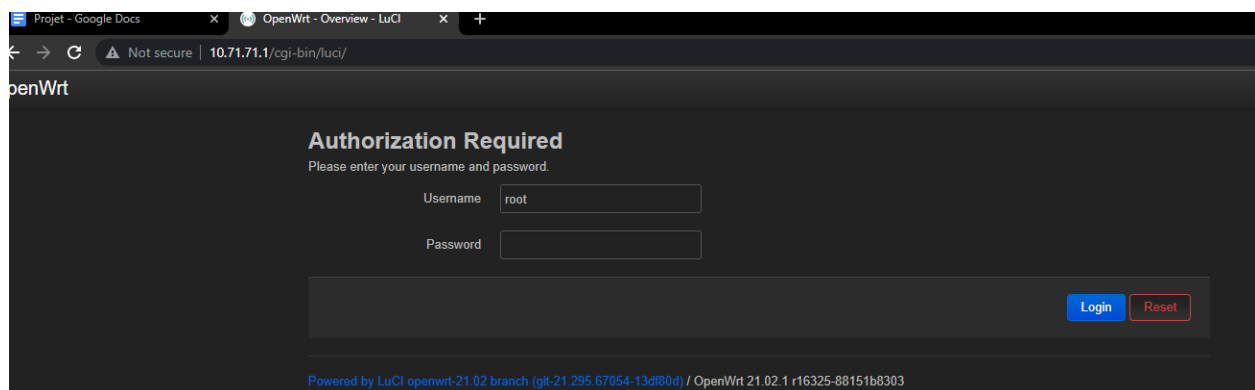


Figure 1.5.5: Connexion de Raspberry pi dans le navigateur web

On clique sur Network -> wireless

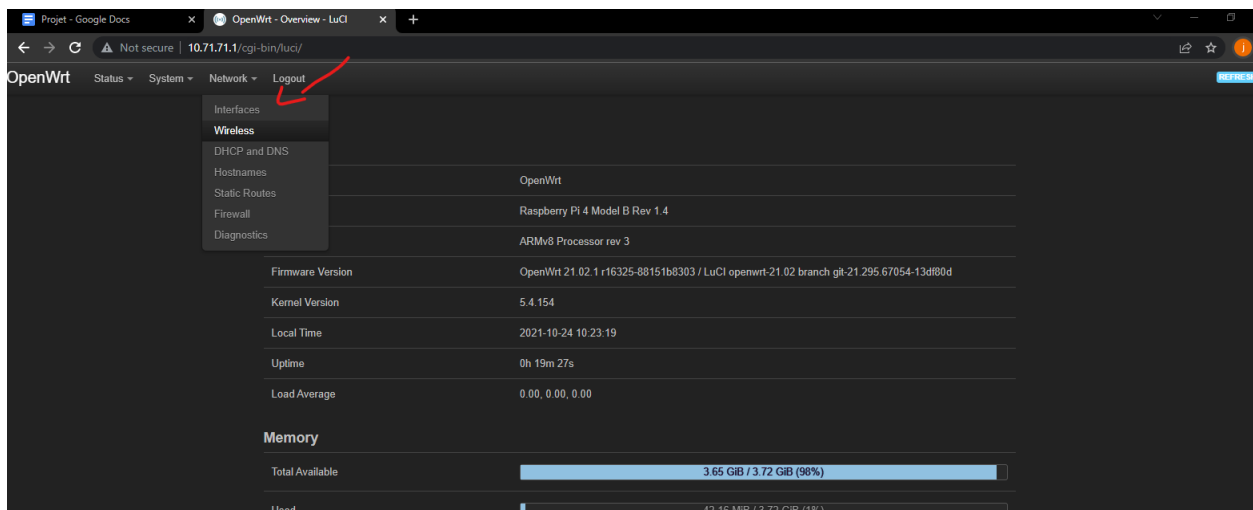


Figure 1.5.6: Wireless dans le navigateur Web

On clique sur scan.

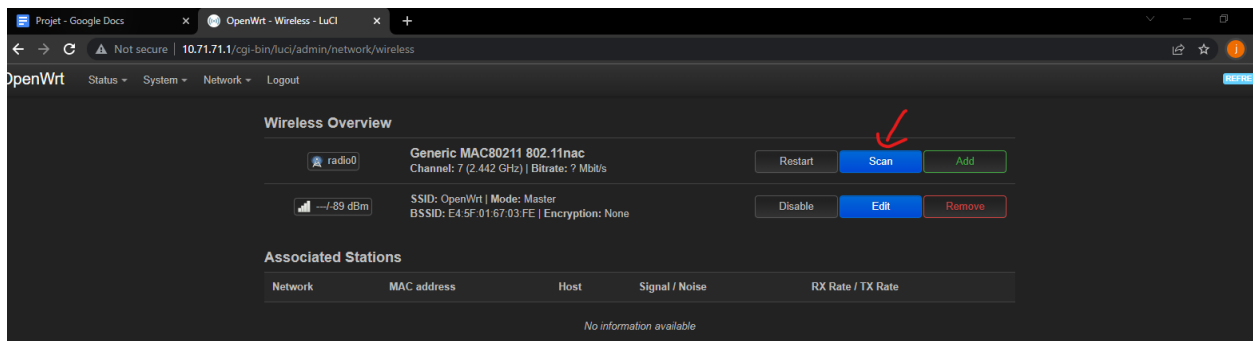


Figure 1.5.7: Scan(recherche) du réseau

Ensuite, on trouve notre connexion maison et on clique sur join Network.

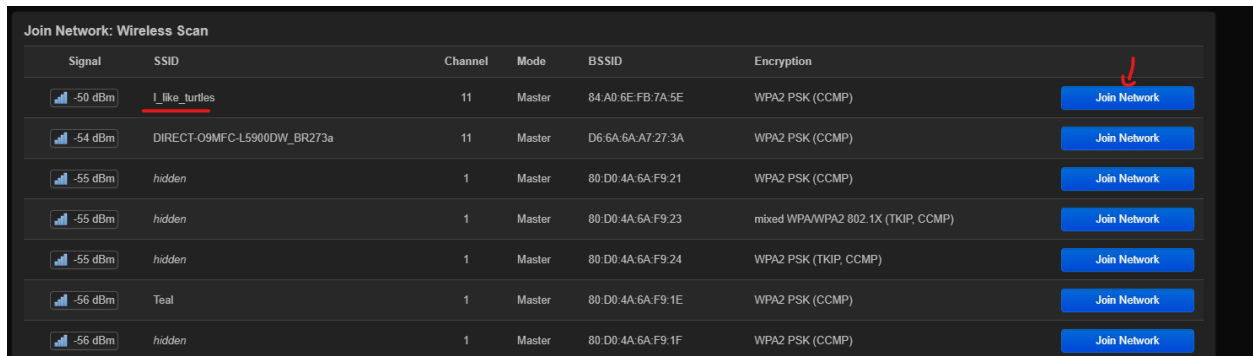


Figure 1.5.8: Join du network

On doit cocher la boîte Replace Wireless Configuration, ensuite on doit ajouter le mot de passe de notre Wifi, et après on a qu'à cliquer sur Submit.

Figure 1.5.9: Configuration du Wireless et ajout d'un mot de passe

Après on clique sur Save.

Figure 1.6.0: Sauvegarde de la configuration

Cliquer sur Save and Apply.

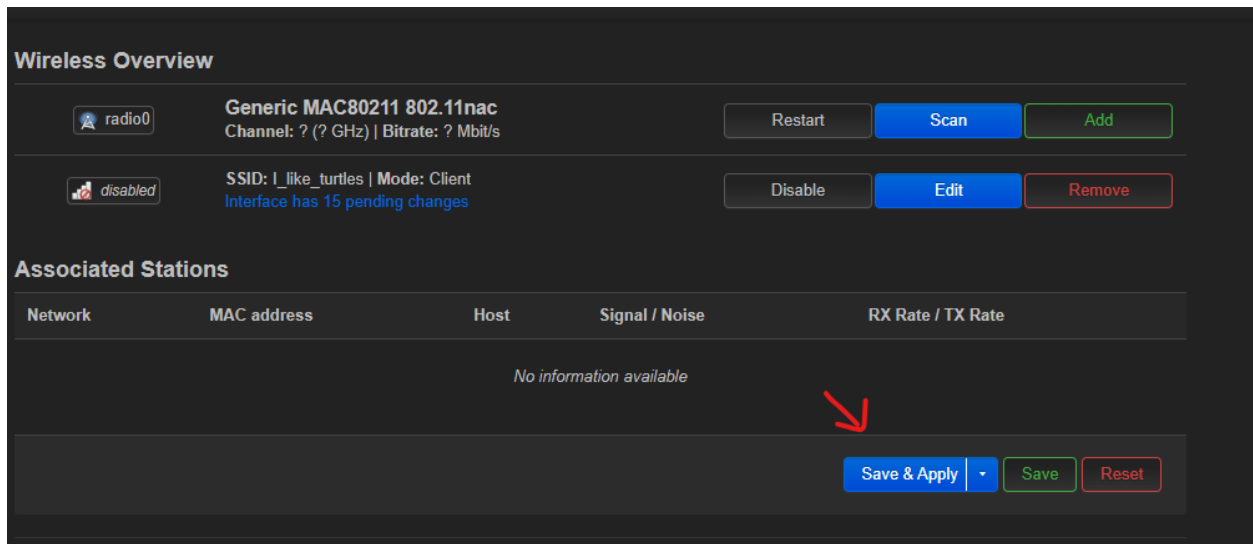


Figure 1.6.1: Enregistrer et appliquer la configuration.

Pour s'assurer qu'on a une connexion d'internet sur notre Raspberry Pi, on ouvre le terminal déjà connecter à OpenWRT et on fait un Ping test vers google : ping 8.8.8.8

```
BusyBox v1.33.1 (2021-10-24 09:01:35 UTC) built-in shell (ash)

root@OpenWrt:~# vi /etc/config/wireless
root@OpenWrt:~# root@OpenWrt:~# uci commit wireless
root@OpenWrt:~# wifi
root@OpenWrt:~# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: seq=0 ttl=116 time=35.549 ms
64 bytes from 8.8.8.8: seq=1 ttl=116 time=31.727 ms
64 bytes from 8.8.8.8: seq=2 ttl=116 time=28.542 ms
64 bytes from 8.8.8.8: seq=3 ttl=116 time=39.226 ms
64 bytes from 8.8.8.8: seq=4 ttl=116 time=26.853 ms
^C
--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 26.853/32.379/39.226 ms
root@OpenWrt:~#
```

Figure 1.6.2: Ping pour s'assurer que vous avez une connexion internet

Si on a une réponse, on a bien configuré le Wifi du Raspberry pi et on peut continuer à l'étape prochaine.

8.1.10 Etape10 : Config du USB Wireless Adapter.

Cette étape consiste à configurer Wireless Adapter (Wireless WiFi USB Dongle Stick Adapter RT5370 150Mbps for MAG 254 250 255 270 275 IPTV Set-Top Box, Jynxbox, Linkbox, Raspberry Pi, Pc Laptops Desktop, for Win7, Win8, Mac OS, Linux) acheté au Amazon : voilà le [lien](#) sur Amazon.



Figure 1.6.3: Wireless WiFi USB Dongle Stick Adapter RT5370

1^{er} chose à faire on doit télécharger et installer des drivers pour que OpenWRT reconnaisse ce Wireless adapter.

On tape cette commande dans le terminal du OpenWRT installer sur notre Raspberry PI.

`# opkg update` = cette commande sert à updater les packages sur notre système OpenWRT.

```
root@OpenWrt:~# opkg update
```

Figure 1.6.4: Mise à jour du paquet

Note : si cette commande ne marche pas donc OpenWRT n'a pas une connexion internet faut redémarrer OpenWRT après la configuration d'internet.

Résultat :

```
-----
OpenWrt 21.02.0, r16279-5cc0535800
-----
root@OpenWrt:~# opkg update
Downloading https://downloads.openwrt.org/releases/21.02.0/targets/bcm27xx/bcm2709/packages/Packages.gz
Updated list of available packages in /var/opkg-lists/openwrt_core
Downloading https://downloads.openwrt.org/releases/21.02.0/targets/bcm27xx/bcm2709/packages/Packages.sig
Signature check passed.
Downloading https://downloads.openwrt.org/releases/21.02.0/packages/arm_cortex-a7_neon-vfpv4/base/Packages.gz
Updated list of available packages in /var/opkg-lists/openwrt_base
Downloading https://downloads.openwrt.org/releases/21.02.0/packages/arm_cortex-a7_neon-vfpv4/base/Packages.sig
Signature check passed.
Downloading https://downloads.openwrt.org/releases/21.02.0/packages/arm_cortex-a7_neon-vfpv4/luci/Packages.gz
Updated list of available packages in /var/opkg-lists/openwrt_luci
Downloading https://downloads.openwrt.org/releases/21.02.0/packages/arm_cortex-a7_neon-vfpv4/luci/Packages.sig
Signature check passed.
Downloading https://downloads.openwrt.org/releases/21.02.0/packages/arm_cortex-a7_neon-vfpv4/packages/Packages.gz
Updated list of available packages in /var/opkg-lists/openwrt_packages
Downloading https://downloads.openwrt.org/releases/21.02.0/packages/arm_cortex-a7_neon-vfpv4/packages/Packages.sig
Signature check passed.
Downloading https://downloads.openwrt.org/releases/21.02.0/packages/arm_cortex-a7_neon-vfpv4/routing/Packages.gz
Updated list of available packages in /var/opkg-lists/openwrt_routing
Downloading https://downloads.openwrt.org/releases/21.02.0/packages/arm_cortex-a7_neon-vfpv4/routing/Packages.sig
Signature check passed.
Downloading https://downloads.openwrt.org/releases/21.02.0/packages/arm_cortex-a7_neon-vfpv4/telephony/Packages.gz
Updated list of available packages in /var/opkg-lists/openwrt_telephony
Downloading https://downloads.openwrt.org/releases/21.02.0/packages/arm_cortex-a7_neon-vfpv4/telephony/Packages.sig
Signature check passed.
root@OpenWrt:~#
```

Figure 1.6.5: Résultat de la mise à jour.

Après la mise à jours des packages on lance cette commande :

```
# opkg install kmod-rt2800-lib kmod-rt2800-usb kmod-rt2x00-lib kmod-rt2x00-usb kmod-rt2x00-
mmio kmod-usb-uhci kmod-usb-ohci kmod-usb2 kmod-mac80211 usbutils openvpn-openssl luci-
app-openvpn nano
```

```
root@OpenWrt:~# opkg install kmod-rt2800-lib kmod-rt2800-usb kmod-rt2x00-lib kmod-rt2x00-usb kmod-usb-core kmod-
usb-uhci kmod-usb-ohci kmod-usb2 usbutils openvpn-openssl luci-app-openvpn nano
```

Figure 1.6.6: La commande utilise après la mise à jour.

Résultat :

```
Installing libncurses6 (6.2-1) to root..  
Downloading https://downloads.openwrt.org/releases/19.07.0/target-arm/cortex-a72/cortex-a72_1_aarch64_cortex-a72.ipk  
Configuring terminfo.  
Configuring libevdev.  
Configuring libncurses6.  
Configuring nano.  
Configuring kmod-tun.  
Configuring kmod-mac80211.  
Configuring kmod-rt2x00-lib.  
Configuring kmod-rt2x00-usb.  
Configuring kmod-rt2800-lib.  
Configuring rt2800-usb-firmware.  
Configuring kmod-rt2800-usb.  
Configuring liblzo2.  
Configuring libusb-1.0-0.  
Configuring kmod-usb-ohci.  
Configuring libudev-zero.  
Configuring usbids.  
Configuring usbutils.  
Configuring kmod-usb-ehci.  
Configuring kmod-usb2.  
Configuring luci-compat.  
Configuring libopenssl1.1.  
Configuring openvpn-openssl.
```

Figure 1.6.7: Résultat de la commande après la mise à jour.

Avec le command `# lsusb` on peut voir combien du périphérique USB sont brancher sur notre Raspberry Pi.

C'est très intéressant de lancer cette commande (# lsusb) avant et après brancher notre Wireless Adapter pour voir.

```
root@OpenWrt:~# lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux 5.4.143 xhci-hcd xHCI Host Controller
Bus 001 Device 003: ID 148f:5370 Ralink 802.11 n WLAN
Bus 001 Device 004: ID 045e:07b2 Microsoft Microsoft® Nano Transceiver v1.0
Bus 001 Device 002: ID 2109:3431 USB2.0 Hub
Bus 001 Device 001: ID 1d6b:0002 Linux 5.4.143 xhci-hcd xHCI Host Controller
root@OpenWrt:~#
```

Figure 1.6.8: Lsusb pour voir combien de périphériques USB sont connectés

Notre Wireless Adapter a le nom **Ralink 802.11 n WLAN**.

Donc notre Wireless est branché mais n'est pas encore active pour l'activer on doit lancer cette commande : # ifconfig wlan1 up

```
root@OpenWrt:~# ifconfig wlan1 up
root@OpenWrt:~#
```

Figure 1.6.9: Activation de Wireless

Après on lance tous simplement ifconfig pour afficher tous les interfaces réseau.

```

root@OpenWrt:~# ifconfig
br-lan    Link encap:Ethernet HWaddr E4:5F:01:45:43:E9
          inet addr:10.71.71.1 Bcast:10.71.71.255 Mask:255.255.255.0
          inet6 addr: fd80:2a85:c246::1/60 Scope:Global
          inet6 addr: fe80::e65f:1ff:fe45:43e9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:16 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B) TX bytes:2504 (2.4 KiB)

eth0      Link encap:Ethernet HWaddr E4:5F:01:45:43:E9
          UP BROADCAST MULTICAST MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:65536 Metric:1
          RX packets:221 errors:0 dropped:0 overruns:0 frame:0
          TX packets:221 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:23159 (22.6 KiB) TX bytes:23159 (22.6 KiB)

tun0      Link encap:UNSPEC HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:10.35.0.29 P-t-P:10.35.0.29 Mask:255.255.0.0
          inet6 addr: fe80::c20d:5af0:bde:523/64 Scope:Link
          UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1500 Metric:1
          RX packets:949 errors:0 dropped:0 overruns:0 frame:0
          TX packets:749 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:969258 (946.5 KiB) TX bytes:48259 (47.1 KiB)

wlan0     1 Link encap:Ethernet HWaddr E4:5F:01:45:43:EA
          inet addr:192.168.2.106 Bcast:192.168.2.255 Mask:255.255.255.0
          inet6 addr: fe80::e65f:1ff:fe45:43ea/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:1230 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1019 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1047963 (1023.4 KiB) TX bytes:152196 (148.6 KiB)

wlan1     2 Link encap:Ethernet HWaddr 08:EA:35:E1:BA:55
          inet6 addr: fe80::aea:35ff:fee1:ba55/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:22 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B) TX bytes:3476 (3.3 KiB)

root@OpenWrt:~#

```

Figure 1.7.0: Affichage de l'interfaces réseau

L'interface numéro 1 : si pour le Wireless interne du Raspberry Pi.

L'interface numéro 2 : si pour notre USB Wireless adapter (externe).

8.1.11 Etape11 : Configuration du réseau sans fil.

Pour faire la configuration du réseau sans fil, on commence avec la commande suivante :

Nano /etc/config/wireless

```
config wifi-device 'radio1'
    option type 'mac80211'
    option channel '11'
    option hwmode '11g'
    option path 'scb/fd500000.pcie/pci0000:00/0000:00:00.0/0000:01:00.0/usb'
    option htmode 'HT20'
    option disabled '1'

config wifi-iface 'default_radio1'
    option device 'radio1'
    option network 'lan'
    option mode 'ap'
    option ssid 'OpenWrt'
    option encryption 'none'
```

Figure 1.7.1: Configuration du réseau sans fil

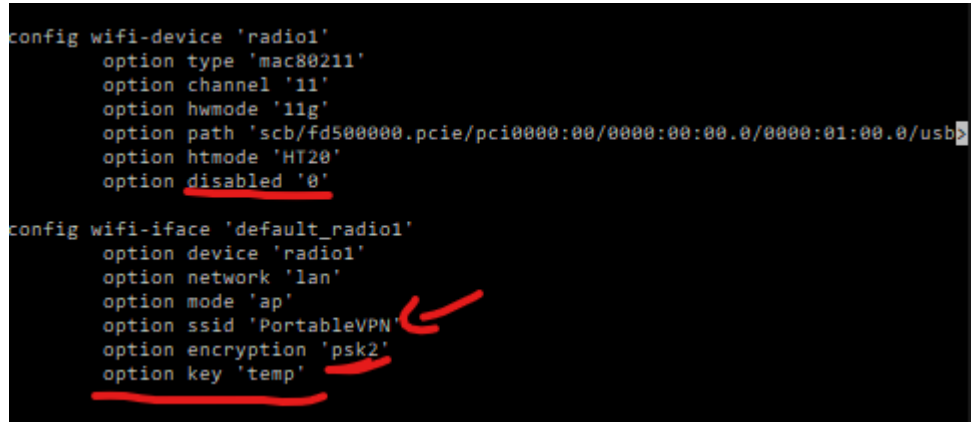
On doit faire les changements suivants :

Option disabled : 1 -> 0

Option ssid : OpenWRT -> PortableVPN (Tu peux y donner un nom différent, ceci est le nom qu'on voit quand on se connecte)

Option encryption : none -> psk2

Et finalement on ajoute une ligne d'extra à la fin : `option key '*MotDePasse*'` (**Note: option key est forcé à être plus de 8 caractères de long.)



```
config wifi-device 'radiol'
  option type 'mac80211'
  option channel '11'
  option hwmode '11g'
  option path 'scb/fd500000.pcie/pci0000:00/0000:00:00.0/0000:01:00.0/usb'
  option htmode 'HT20'
  option disabled '0'

config wifi-iface 'default_radiol'
  option device 'radiol'
  option network 'lan'
  option mode 'ap'
  option ssid 'PortableVPN'
  option encryption 'psk2'
  option key 'temp'
```

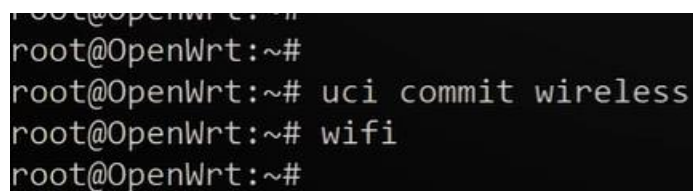
Figure 1.7.2: Modification à apporter à la configuration du réseau sans fil

Une fois que les modifications sont finies, on frappe les touches suivantes : `CTRL + X, Y, Enter`

Après on lance les commandes suivantes :

`Uci commit wireless`

`Wifi`



```
root@OpenWrt:~#
root@OpenWrt:~# uci commit wireless
root@OpenWrt:~# wifi
root@OpenWrt:~#
```

Figure 1.7.3: Résultat de la commande Uci

Après ces changements, on va être capable de se connecter en utilisant le Wifi de notre machine.

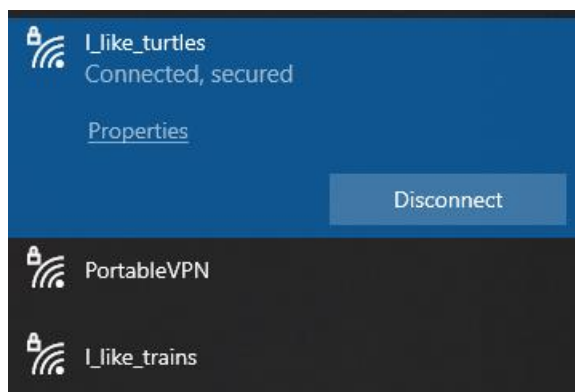


Figure 1.7.4: Accès Wifi depuis notre machine.

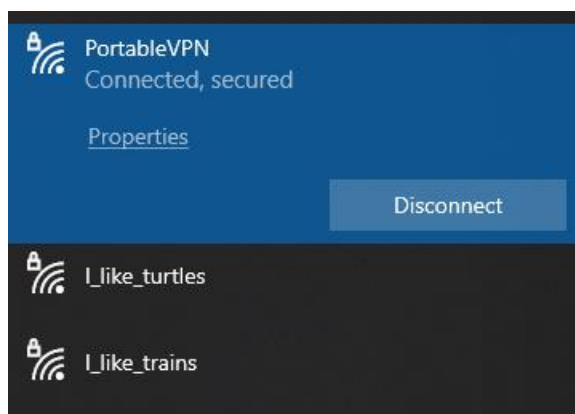


Figure 1.7.5: Connexion avec le wifi de notre machine.

8.1.12 Etape12 : RPV configuration avec ProtonVPN

1^{er} chose on doit créer un compte sur ProtonVPN, note ProtonVPN donne la possibilité d'utiliser VPN gratuit mais limite

Voilà le site : <https://protonvpn.com/>

Après création d'un compte gratuit et connecter sur le site protonvpn on clique sur le menu Downloads

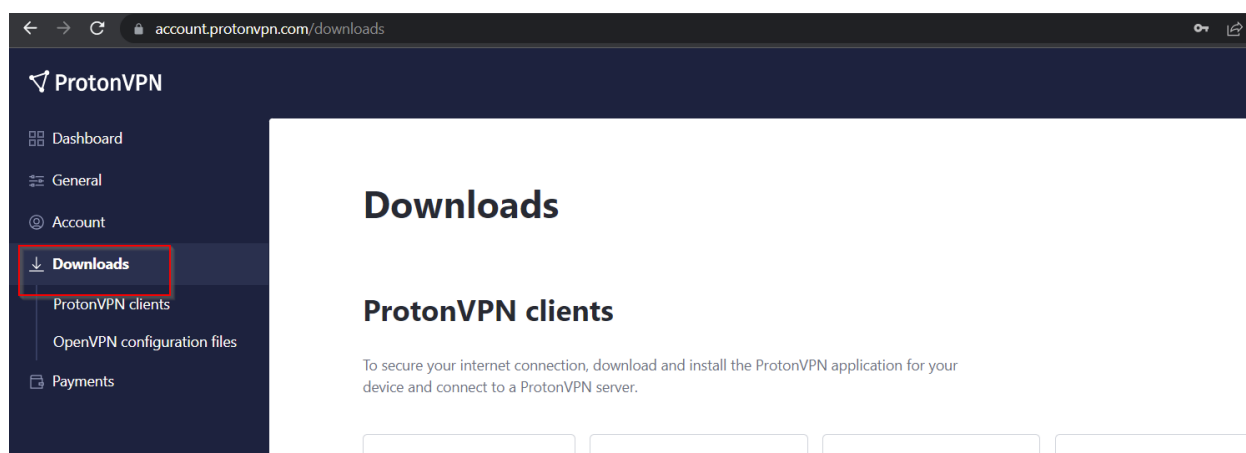


Figure 1.7.6: Téléchargement de ProtonVPN.

Sur la section OpenVPN configuration files on suite ces étapes :

1. On coche Router.
2. On coche le protocole UDP.
3. On coche free server config.
4. On choisit le serveur qu'on veut utiliser et on télécharge le fichier de la configuration.

OpenVPN configuration files

These configuration files let you choose which ProtonVPN server you connect to when using a third-party VPN app or setting up a VPN connection on a router.

1. Select platform

☐ Android [View guide](#)
☐ iOS [View guide](#)
☐ Windows [View guide](#)
☐ macOS [View guide](#)
☐ GNU/Linux [View guide](#)
☒ Router [View guide](#) 1

2. Select protocol

☒ UDP [What is the difference between UDP and TCP protocols?](#)
☐ TCP

3. Select config file and download

☐ Secure Core configs
 ☐ Country configs
 ☐ Standard server configs
 ☒ Free server configs 3

Install a Free server configuration file to connect to a specific server in one of the three free locations.

^ Japan

Name	Status	Action
JP-FREE#1	100%	Download 4
JP-FREE#2	92%	Download
JP-FREE#3	92%	Download
JP-FREE#4	93%	Download
JP-FREE#5	90%	Download
JP-FREE#6	91%	Download
JP-FREE#7	91%	Download

Figure 1.7.7: Étapes de configuration d'OpenVPN

Après avoir téléchargé le fichier, nous le renommons, capture d'écran ci-dessous nous l'avons renommé ProtonVPN.ovpn.

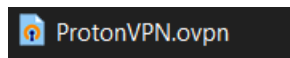


Figure 1.7.8: Renommez le fichier ProtonVPN.

On doit créer un fichier .auth pour mettre nos identification du site ProtonVPN.

Donc on doit avoir deux fichiers comme suite.

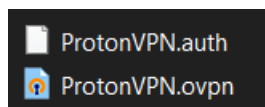


Figure 1.7.9: Création d'un fichier d'authentification

On ouvre le fichier ProtonVPN.ovpn avec un éditeur du texte comme Notepad++ et on ajoute ce chemin `/etc/openvpn/ProtonVPN.auth` a la ligne 69 : `auth-user-pass` pour avoir cette capture :

```
68 remote-cert-tls server
69 auth-user-pass /etc/openvpn/ProtonVPN.auth
70 pull
71 fast-io
```

On modifier le fichier ProtonVPN.ovpn normalement est vide on doit ajouter notre user et le mot de passe qu'on le trouve sur notre espace ProtonVPN site.

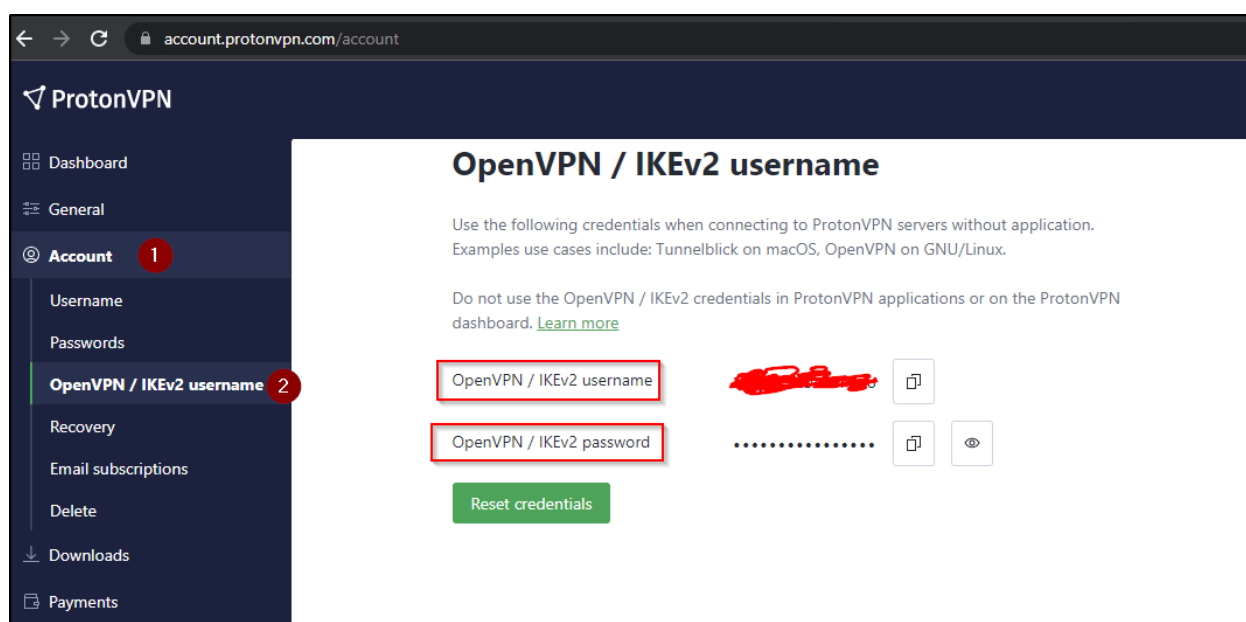


Figure 1.8.0: Modification du fichier ProtonVPN.ovpn

Sur notre copte ProtonVPN on clique sur le menu **Account** et après **OpenVPN / IKEv2 username**

Nous copions le nom d'utilisateur et le mot de passe puis nous remplissons notre fichier ProtonVPN.auth

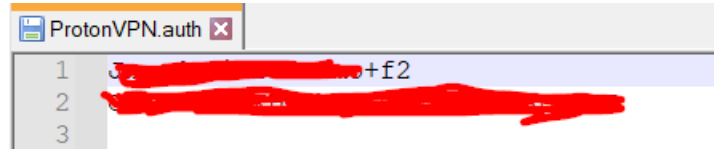


Figure 1.8.1: Remplir notre fichier ProtonVPN.auth

Ajoutez +f2 devant le nom d'utilisateur pour bloquer les ADS et les malwares

Donc ligne 1 on met le nom d'utilisateur +f2 et ligne 2 on met le mot de passe.

Nous sauvegardons le fichier et nous devons le copier sur notre système OpenWRT dans ce chemin /etc/openvpn

Nous avons utilisé cet utilitaire WinSCP pour copier des fichiers entre la machine locale et le système OpenWRT installé sur Raspberry pi

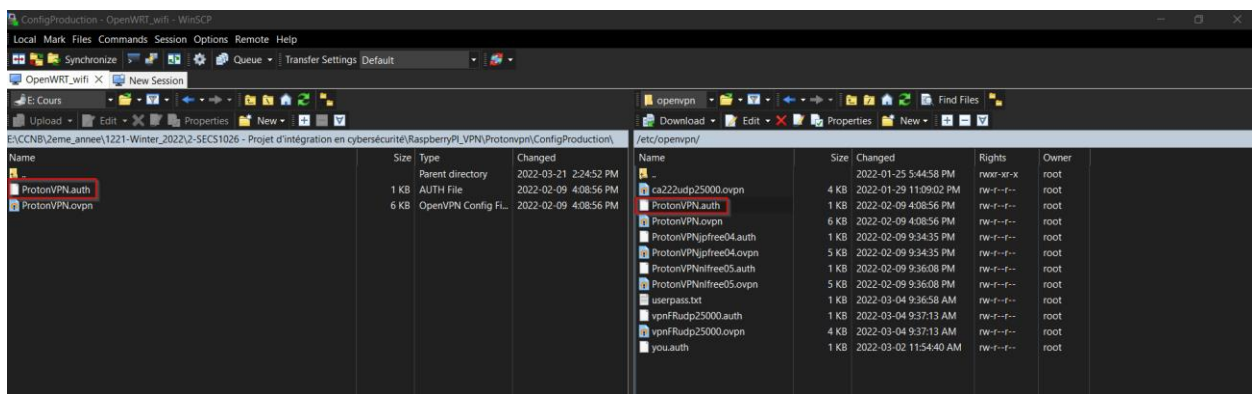


Figure 1.8.2: Copier des fichiers entre la machine locale et le système OpenWRT.

Sur le terminal nous pouvons vérifier si le fichier est bien copié dans le bon chemin ou non nous lançons ces commandes.

```
# cd /etc/openvpn/
```

```
# pwd
```

```
# ls -l
```

```
root@OpenWrt:~# cd /etc/openvpn/
root@OpenWrt:/etc/openvpn# pwd
/etc/openvpn
root@OpenWrt:/etc/openvpn# ls -l
-rw-r--r--  1 root  root    53 Feb  9 20:08 ProtonVPN.auth
-rw-r--r--  1 root  root  5286 Feb  9 20:08 ProtonVPN.ovpn
-rw-r--r--  1 root  root    53 Feb 10 01:34 ProtonVPNjpfree04.auth
-rw-r--r--  1 root  root  4872 Feb 10 01:34 ProtonVPNjpfree04.ovpn
-rw-r--r--  1 root  root    53 Feb 10 01:36 ProtonVPNnlfree05.auth
-rw-r--r--  1 root  root  4872 Feb 10 01:36 ProtonVPNnlfree05.ovpn
-rw-r--r--  1 root  root  3972 Jan 30 03:09 ca222udp25000.ovpn
-rw-r--r--  1 root  root    16 Mar  4 13:36 userpass.txt
-rw-r--r--  1 root  root    16 Mar  4 13:37 vpnFRudp25000.auth
-rw-r--r--  1 root  root  3971 Mar  4 13:37 vpnFRudp25000.ovpn
-rw-r--r--  1 root  root    53 Mar  2 15:54 you.auth
root@OpenWrt:/etc/openvpn#
```

Figure 1.8.3: Vérification du fichier copié.

Maintenant, nous ouvrons l'interface OpeWRT sur un navigateur, notre adresse IP OpenWRT est 192.168.2.106

Et on clique sur VPN à OpenVPN

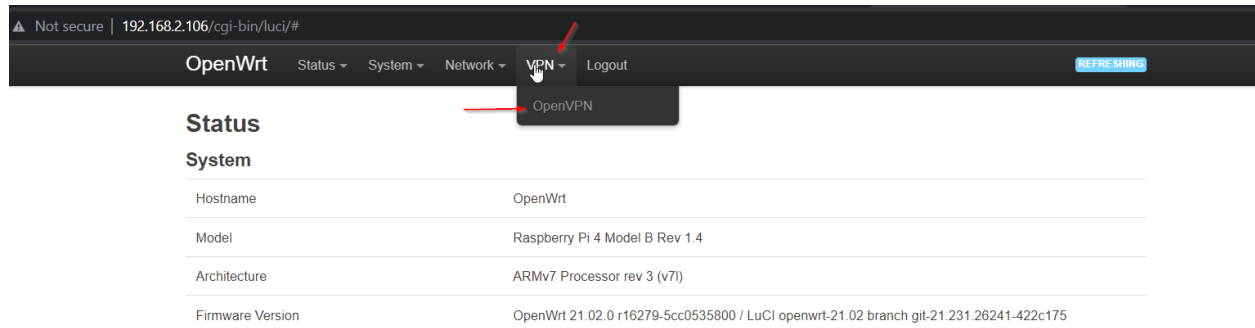


Figure 1.8.4: Interface OpeWRT sur le navigateur

OpenWrt Status System Network VPN Logout UNSAVED CHANGES: 1

OpenVPN

OpenVPN instances

Below is a list of configured OpenVPN instances and their current state

Name	Enabled	Started	Start/Stop	Port	Protocol		
custom_config	<input type="checkbox"/>	no	start	-	-	Edit	Delete
sample_server	<input type="checkbox"/>	no	start	1194	udp	Edit	Delete
sample_client	<input type="checkbox"/>	no	start	-	udp	Edit	Delete
ca222udp25000	<input type="checkbox"/>	no	start	25000	udp	Edit	Delete
vpnFRudp25000	<input type="checkbox"/>	no	start	25000	udp	Edit	Delete
ProtonVPN	<input type="checkbox"/>	no	start	4569	udp	Edit	Delete
ProtonVPNjpfree04	<input type="checkbox"/>	no	start	443	udp	Edit	Delete
ProtonVPNnlfree05	<input type="checkbox"/>	no	start	4569	udp	Edit	Delete

Template based configuration

Instance name Select template ... [Add](#)

OVPN configuration file upload

Instance name [Choose File](#) No file chosen [Upload](#)

[Save & Apply](#) [Save](#) [Reset](#)

Powered by LuCI openwrt-21.02 branch (git-21.231.26241-422c175) / OpenWrt 21.02.0 r16279-5cc0535800

Figure 1.8.5: Téléchargement du fichier ProtonVPN.OVPN

1. On met un nom pour notre config VPN, on met ProtonVPN
2. Nous recherchons notre fichier de configuration VPN, notre cas est le fichier de téléchargement ProtonVPN.ovpn
3. Nous cliquons sur télécharger.

Après cela, nous remarquerons qu'une configuration sera ajoutée à l'instance OpenVPN

192.168.2.106/cgi-bin/luci/admin/vpn/openvpn

OpenWrt

StatusSystemNetworkVPNLogout

UNSAVED CHANGES

OpenVPN

OpenVPN instances

Below is a list of configured OpenVPN instances and their current state

Name	Enabled	Started	Start/Stop	Port	Protocol		
custom_config	<input type="checkbox"/>	no	<button>start</button>	-	-	<button>Edit</button>	<button>Delete</button>
sample_server	<input type="checkbox"/>	no	<button>start</button>	1194	udp	<button>Edit</button>	<button>Delete</button>
sample_client	<input type="checkbox"/>	no	<button>start</button>	-	udp	<button>Edit</button>	<button>Delete</button>
ca222udp25000	<input type="checkbox"/>	no	<button>start</button>	25000	udp	<button>Edit</button>	<button>Delete</button>
vpnFRudp25000	<input type="checkbox"/>	no	<button>start</button>	25000	udp	<button>Edit</button>	<button>Delete</button>
ProtonVPN	<input type="checkbox"/>	no	<button>start</button>	4569	udp	<button>Edit</button>	<button>Delete</button>
ProtonVPNjpfree04	<input type="checkbox"/>	no	<button>start</button>	443	udp	<button>Edit</button>	<button>Delete</button>
ProtonVPNnlfree05	<input type="checkbox"/>	no	<button>start</button>	4569	udp	<button>Edit</button>	<button>Delete</button>

Template based configuration

Figure 1.8.6: Ajout de la configuration de l'instance OpenVPN

On est ok maintenant pour lancer notre VPN on coche notre **ProtonVPN** et on clique sur **Save & Apply**

192.168.2.106/cgi-bin/luci/admin/vpn/openvpn

OpenWrt

Status

System

Network

VPN

Logout

UNSAVED CHANGES: 1

OpenVPN

OpenVPN instances

Below is a list of configured OpenVPN instances and their current state

Name	Enabled	Started	Start/Stop	Port	Protocol		
custom_config	<input type="checkbox"/>	no	start	-	-	Edit	Delete
sample_server	<input type="checkbox"/>	no	start	1194	udp	Edit	Delete
sample_client	<input type="checkbox"/>	no	start	-	udp	Edit	Delete
ca222udp25000	<input type="checkbox"/>	no	start	25000	udp	Edit	Delete
vpnFRudp25000	<input type="checkbox"/>	no	start	25000	udp	Edit	Delete
ProtonVPN	<input checked="" type="checkbox"/>	no	start	4569	udp	Edit	Delete
ProtonVPNjpfree04	<input type="checkbox"/>	no	start	443	udp	Edit	Delete
ProtonVPNnlfree05	<input type="checkbox"/>	no	start	4569	udp	Edit	Delete

Template based configuration

Instance name

Select template ...

Add

OVPN configuration file upload

Instance name

Choose File

No file chosen

Upload

2

Save & Apply

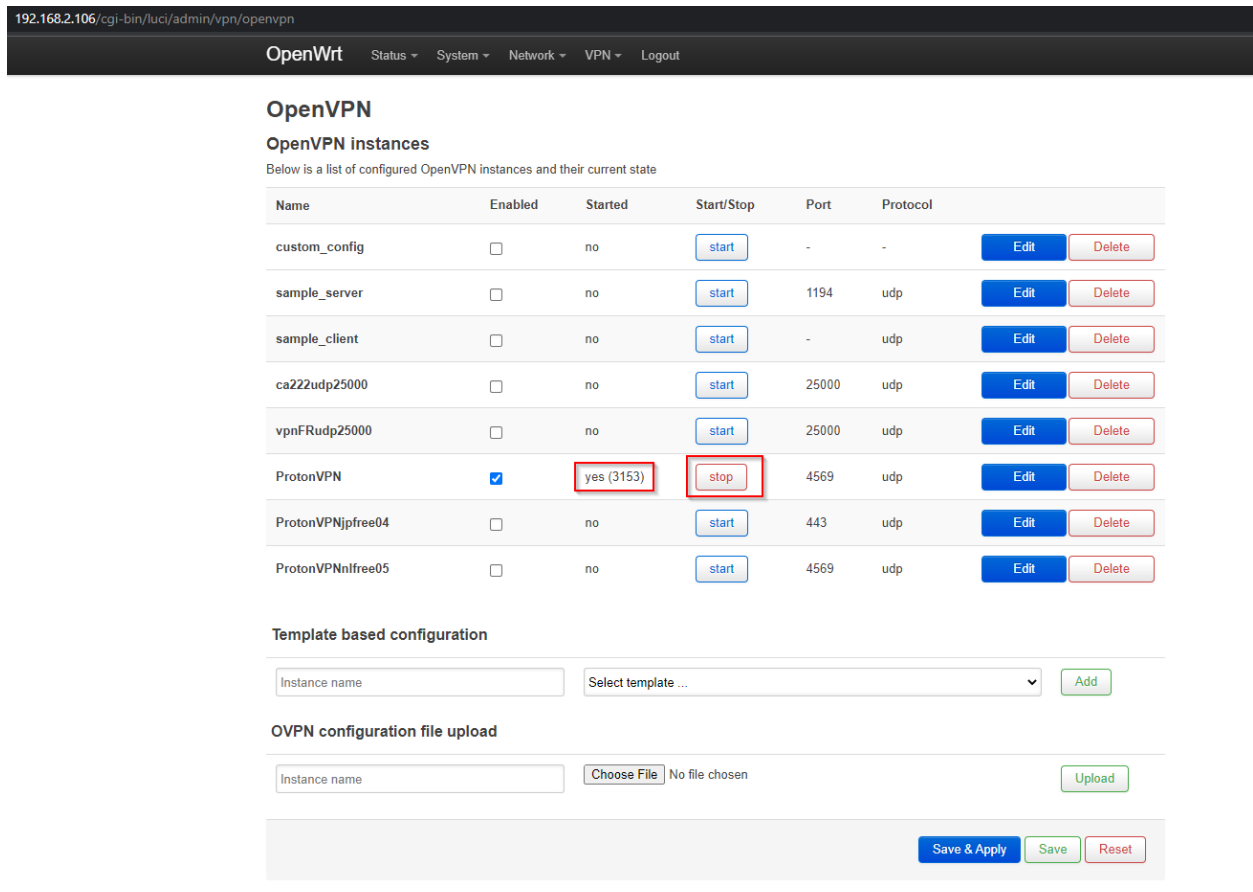
Save

Reset

Powered by LuCI openwrt-21.02 branch (git-21.231.26241-422c175) / OpenWrt 21.02.0 r16279-5cc0535800

Figure 1.8.7: Enregistrement de configuration d'instance OpenVPN

Si tous marchent bien on doit avoir comme cette capture d'écran.



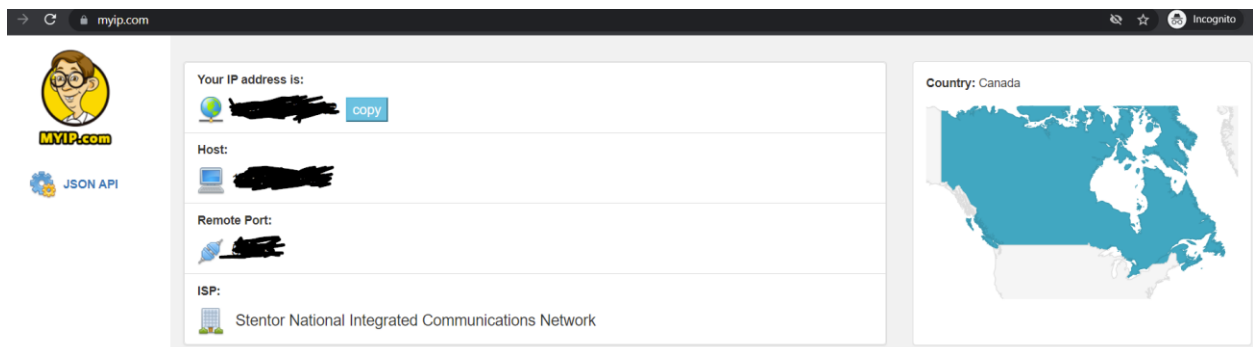
The screenshot shows the OpenVPN web interface. At the top, there's a navigation bar with 'OpenWrt', 'Status', 'System', 'Network', 'VPN', and 'Logout'. Below this, the 'OpenVPN' section is active, showing 'OpenVPN instances'. A table lists several instances, including 'ProtonVPN' which is enabled and has a 'stop' button highlighted with a red box. Below the table, there are sections for 'Template based configuration' and 'OVPN configuration file upload'.

Name	Enabled	Started	Start/Stop	Port	Protocol		
custom_config	<input type="checkbox"/>	no	start	-	-	Edit	Delete
sample_server	<input type="checkbox"/>	no	start	1194	udp	Edit	Delete
sample_client	<input type="checkbox"/>	no	start	-	udp	Edit	Delete
ca222udp25000	<input type="checkbox"/>	no	start	25000	udp	Edit	Delete
vpnFRudp25000	<input type="checkbox"/>	no	start	25000	udp	Edit	Delete
ProtonVPN	<input checked="" type="checkbox"/>	yes (3153)	stop	4569	udp	Edit	Delete
ProtonVPNjpfree04	<input type="checkbox"/>	no	start	443	udp	Edit	Delete
ProtonVPNnlfree05	<input type="checkbox"/>	no	start	4569	udp	Edit	Delete

Figure 1.8.8: Bon fonctionnement d'OpenVPN.

8.1.13 Test que VPN marche.

Pour test on se connecte avec notre réseau internet directement à notre routeur sans passer par le nouveau réseau wifi configurer sur Raspberry Pi et on check avec ce site <https://www.myip.com/>.



The screenshot shows the myip.com website. It displays the user's IP address, host, remote port, and ISP (Stentor National Integrated Communications Network). There is also a map of Canada and a 'Country: Canada' label.

Figure 1.8.9: Connexion avec notre réseau internet

Ça montre que je suis bien au Canada : avec mon IP adresse du fournisseur internet.

Maintenant on se connecte sur notre nouveau réseau wifi configurer sur notre Raspberry PI

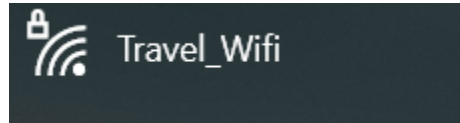


Figure 1.9.0: Connexion wifi sur notre Raspberry PI

Et on se connecte encore sur notre site pour checker notre position on voit bien qu'on est au

United States

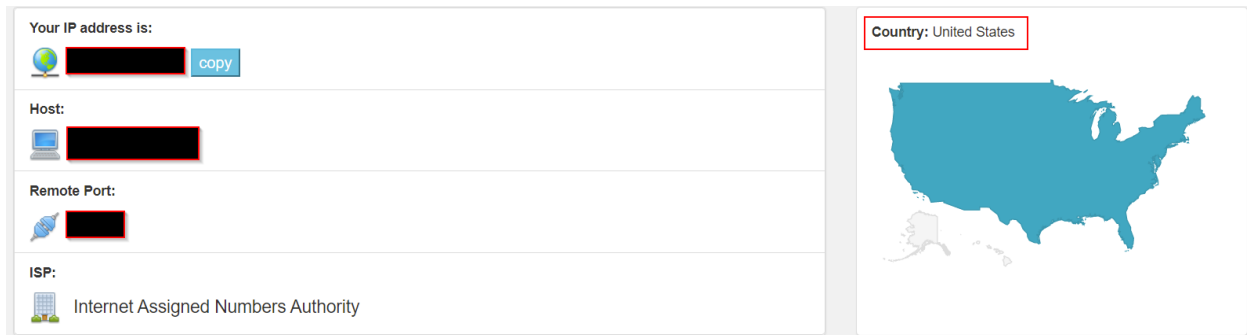


Figure 1.9.1 : Position actuelle après connexion wifi sur notre Raspberry PI.

8.1.14 Test rapidité de la connexion avant et après l'utilisation de notre VPN.

Connexion sans VPN :

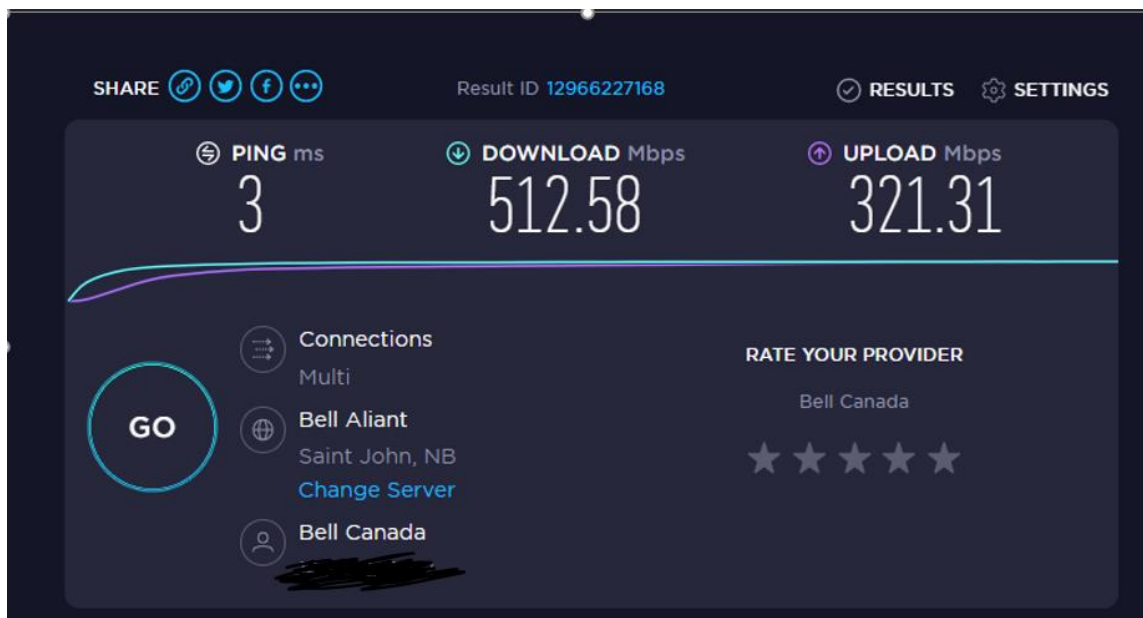


Figure 1.9.2: Connexion sans VPN.

Connexion avec VPN :

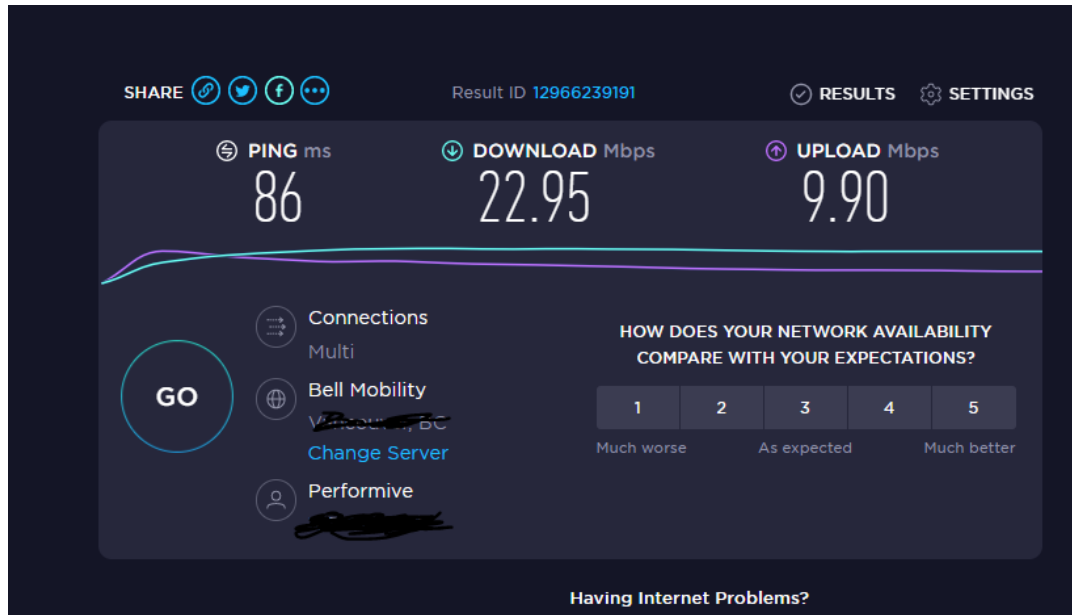


Figure 1.9.3 : Connexion avec VPN.

Remarque : Nous avons remarqué que la connexion est lente avec VPN par rapport au test sans VPN Parce que notre adaptateur USB sans fil est limité.

8.2 Planning du déploiement de notre projet.

1.1.1 Les étapes du projet et leur durée.

N	Les tâches	La durée par jours	Start Date	End date	Responsable
1	Phase étude de cadrage	8	2022-01-20	2022-01-28	Fatima, Julien, Ahmed, Yann et Youssef
2	Études d'impacts	9	2022-01-29	2022-02-07	Fatima, Julien, Ahmed, Yann et Youssef
3	Spécifications fonctionnelles et détaillées	15	2022-02-08	2022-02-23	Fatima, Julien, Ahmed, Yann et Youssef
4	Conception et réalisation du projet	20	2022-02-24	2022-03-16	Fatima, Julien, Ahmed, Yann et Youssef
5	Délivrable rapport de conception, guide d'utilisation, document technique...	25	2022-02-16	2022-04-01	Fatima, Julien, Ahmed, Yann et Youssef
6	Phase test d'assurance qualité	20	2022-03-15	2022-04-01	Fatima, Julien, Ahmed, Yann et Youssef
7	Recette d'utilisation et implémentation production	10	2022-03-25	2022-04-04	Fatima, Julien, Ahmed, Yann et Youssef
8	Support après production	20	2022-04-04	2022-04-30	Fatima, Julien, Ahmed, Yann et Youssef
Total		127			

Figure 1.9.4: Étapes du projet et leur durée.

Durée prévisionnelle = 3 mois et 15 jours

1.1.2 Tableau prévisionnel du projet.

N	Mois Taches	Jan	Fev	Mars	Avril
1	Phase étude de cadrage	***			
2	Etudes d'impacts	*	**		
3	Spécifications fonctionnelles et détaillées		***		
4	Conception et réalisation du projet		**	**	
5	Delivable rapport de conception, guide d'utilisation, document technique...		***	***	***
6	Phase test d'assurance qualité			***	*
7	Recette d'utilisation et implémentation production			*	***
8	Support après production				****

Figure 1.9.5: Tableau prévisionnel du projet.

8.3 Évaluation et le budget financier du nouveau système.

Raspberry Pi 4 Kit	\$200
Wireless WiFi USB Dongle Stick Adapter	\$34.13
Express Shipping	\$25.86
GST/HST	\$33.87
Total	\$293.86
Optionnelle, si vous l'avez, pas besoin d'acheter	
Écran Tactile	\$35.92
Clavier	\$12.99
ProtonVPN (1 Mois)	\$13.74
Total avec les extras	\$356.51

Figure 1.9.6 : Évaluation et budget financier

Pour la commercialisation de ce system, nous pouvons y vendre sans d'écran, sans de clavier et avec un VPN gratuit, ce qui rend le prix de fabrication à \$293.86. Si on se base sur les prix dans le tableau juste en haut, on peut facilement charger dans les alentours de \$400 par system assembler et configurer, car il y aura des gens qui allons avoir besoin d'acheter un clavier et d'un écran, mais une fois le produit est assembler, on n'a plus besoin de ces deux choses, ce qui nous permet de charger un peu plus cher pour le produit finale, on vous sauve le mal de tête qui est configurer un system plus on vous sauve le trouble d'acheter un clavier avec un écran si vous n'en avez pas d'extra chez-vous.

8.4 Conclusion.

Arrive au terme de la rédaction de notre projet, nous pouvons dire que l'élaboration de ce travail nous a permis d'acquérir des connaissances dans les domaines suivants :

- L'utilisation du Raspberry PI,
- La mise en place du système OpenWRT.
- Planification et gestion du projet via JIRA.
- La mise en place d'un VPN dans un routeur qui utilise OpenWRT comme système d'exploitation.
- Les différents protocoles intervenants dans la transmission des données dans le réseau VPN.

En outre, le fait d'avoir réalisé le projet en groupe nous a permis de comprendre comment collaborer en équipe pour délivrer les tâches assignées à nous en respectant les délais en utilisant l'outil de la gestion du projet JIRA.

Cette expérience enrichissante nous a également permis de comprendre l'importance réelle de l'utilisation des VPN pour les gens qui voyagent souvent pour sécuriser leurs communications et transactions dans les réseaux publics comme les hôtels, supermarchés et les aéroports.

Au final ce projet est un outil permettant aux gens nomades qui voyage souvent plus particulièrement aux personnes qui cherche d'être en sécurité en utilisant des réseaux publics non sécurisés d'être mieux protégés et d'utiliser l'internet en toute sécurité.

Au cours de ce travail de recherche au sein du CCNB, nous avons approfondi nos connaissances sur l'utilisation du Raspberry Pi, du VPN, de la sécurité informatique, des télécommunications et en réseau.

8.5 Glossaire.

A
ARM : Advanced RISC Machines
B
C
CCNB : Collège communautaire du Nouveau-Brunswick
D
DNS : Domain Name System (système de noms de domaine) DHCP : Dynamic Host Configuration Protocol
E
F
G
GO : Gigaoctet
H
HTTP : Hypertext Transfer Protocol, « protocol de transfer hypertext » HDMI : High-Definition Multimedia Interface
I
IP : Internet Protocol
J
K
L
LAN : Local Area Network
M
MO : Mega Octet MAC: Macintosh MAC : Media Access Control
N
NAT : Network Adresse Translation

O
OS : Operating System en anglais, un système d'exploitation en français ;
P
PKI : Public Key Infrastructure PC : Personal Computer
Q
R
S
SSL : (Secure Sockets Layer) SSD : Solid-state drive SSH : Secure Shell
T
Telnet : (Terminal Network ou Télécommunication Network, ou encore Télértype Network) TCP : Transmission Control Protocol (littéralement, « protocole de contrôle de transmissions ») TLS : Transport Layer Security
U
USB : Universal Serial Bus
V
VPN : Virtual Privat Network
W
WAN : Wide Area Network WIN : Windows WLAN : Wireless LAN
X
Y
Z

Figure 1.9.7 : Glossaire.

9. Références :

<https://www.youtube.com/watch?v=jlHWnKVpygw>

<https://www.le-vpn.com/fr/10-criteres-meilleur-vpn/>

<https://www.raspberrypi.com/software/>

https://openwrt.org/toh/raspberry_pi_foundation/raspberry_pi

[https://fr.wikipedia.org/wiki/Analyse_fonctionnelle_\(conception\)#::~:~:text=%C3%A0%20la%20r%C3%A9g%20l%C3%A9mentation.-,Fonction%20compl%C3%A9mentaire,son%20produit%20et%20la%20qualit%C3%A9](https://fr.wikipedia.org/wiki/Analyse_fonctionnelle_(conception)#::~:~:text=%C3%A0%20la%20r%C3%A9g%20l%C3%A9mentation.-,Fonction%20compl%C3%A9mentaire,son%20produit%20et%20la%20qualit%C3%A9)

<https://www.lemondeinformatique.fr/actualites/lire-10-projets-raspberry-pi-a-la-portee-de-tous-77155.html>

[Amazon - Raspberry Pi Extreme Kit](#)

[Amazon - Wireless WiFi USB Dongle Stick Adapter](#)

[Nord VPN](#)

[Amazon - 3.5inch IPS Screen 480x320 SPI Touch Port Resistive Touch Display LCD](#)

[Amazon - Clavier](#)

[Jira project : https://ccnbgrou2.atlassian.net/jira/software/projects/RPVN/boards/2/roadmap](https://ccnbgrou2.atlassian.net/jira/software/projects/RPVN/boards/2/roadmap)