# Réalisation d'un modèle d'architecture de réseau open source, simple et peu coûteux

(Projet 20 : La mise-en-place d’un réseau d’entreprise sécurisé avec des outils open source)



Etudiant : Matteo BENNIS & Olivier PORCO

Tuteur : Faissal BAKALI

Année 2022-2023

# Remerciement

M Faissal BAKALI, M Philippe EGEA et M Alexandre GUY

Nous remercions sincèrement M Faissal BAKALI pour sa contribution en prêtant le matériel nécessaire pour notre projet d'étude. Nous sommes profondément reconnaissants de son engagement envers notre projet grâce à son soutien, nous avons pu mener à bien notre travail de manière efficace et professionnelle. De même, nous tenons à exprimer notre gratitude à Mr Philippe EGA pour son expertise et les conseils précieux qu'il nous a prodigués. Ses suggestions nous ont été d'une aide précieuse pour orienter nos travaux dans la bonne direction. Nous remercions également M Alexandre GUY, pour son cours sur le routage dans environnement Linux, car les connaissances transmises on était très utiles pour la mise en place notre mini infrastructure.

Encore une fois, merci pour votre soutien et votre collaboration. Nous sommes convaincus que notre travail serait beaucoup plus difficile sans votre aide.

Sincèrement,

Matteo BENNIS & Olivier PORCO.

**Sommaire**

# Introduction

La mise en réseau est un élément clé pour les entreprises et les particuliers, car elle permet de connecter les personnes, les appareils et les objets connectés pour faciliter la communication, la collaboration et l'accès aux données; cependant, ce genre de travail, peuvent s'avérer coûteuses et complexes, surtout pour les petites entreprises et les particuliers mais de plus en plus indispensable, car les cyberattaques sont devenue beaucoup plus fréquent.

Ce projet tutoré vise à résoudre ces problèmes en proposant un modèle d'architecture de réseau open source, simple et peu coûteux, qui peut s'adapter aux besoins spécifiques des petites entreprises et des particuliers, facile à mettre en place et configurée pour les utilisateurs, tout en étant évolutive et adaptable aux différents besoins et intégrant des fonctionnalités pour garantir la sécurité des données.

Dans ce rapport, nous allons détailler les besoins et les objectifs de ce projet, les caractéristiques techniques et les livrables attendus, ainsi que les délais et le budget estimé.

# Problématiques

Pour les petites entreprises :

- Offrir un modèle d'architecture de réseau open source, simple et peu coûteux, qui peut s'adapter aux besoins spécifiques des petites entreprises qui n'ont pas encore leur propre réseau.

- Proposer des solutions pour améliorer la connectivité des employés en présentiel et en distance, faciliter la collaboration et la communication, et protéger les données sensibles.

- Intégration des fonctionnalités de cybersécurité pour protéger les données contre les attaques et les violations de données.

Pour les particuliers :

- Permette aux particuliers de mieux contrôler leur réseau domestique en leur offrant plus de possibilités que les paramètres définis par leur fournisseur d'accès internet (FAI), comme la possibilité de choisir leur propre DNS, d'appliquer des règles de filtrage, et de profiter d'outils normalement réservés aux entreprises comme un NAS ou un serveur VPN pour les sauvegardes sécurisées à distance .

- Intégration des fonctionnalités de domotique pour connecter et gérer les objets connectés (Internet des objets) à travers le réseau domestique.

- Intégration des fonctionnalités de cybersécurité pour protéger la confidentialité numérique.

En utilisant des solutions open source, ce projet vise à offrir une solution abordable, facile à mettre en place et configurer pour les utilisateurs, tout en étant évolutive et adaptable aux différents besoins et à la fois sécurisé pour protéger les données.

# Cahier des charges

Caractéristiques techniques :

- Utilisation exclusive de solutions open source

- Facilité de mise en place et de configuration pour les utilisateurs

- Évolutivité et adaptabilité aux différents besoins

- Sécurité de base pour protéger les données

- Conformité aux normes réseaux

- Facilité de gestion, d'administration et de surveillance

**Livrables :**

- Modèle d'architecture de réseau open source complet

- Documentation technique détaillée

- Manuel d'utilisation pour les utilisateurs

**Budget :**

- Coût total estimé de 500 euro

**Responsabilités :**

- Utilisateurs : responsables de la mise en place et de la configuration du modèle d'architecture de réseau, de la maintenance et de la mise à jour.

# Solution propose

## Réflexion

L'infrastructure proposée utilise un Raspberry Pi 4 comme routeur, pare-feu, DNS et VPN, ainsi que deux autres Raspberry Pi 4 comme serveurs d'application et serveur DHCP.

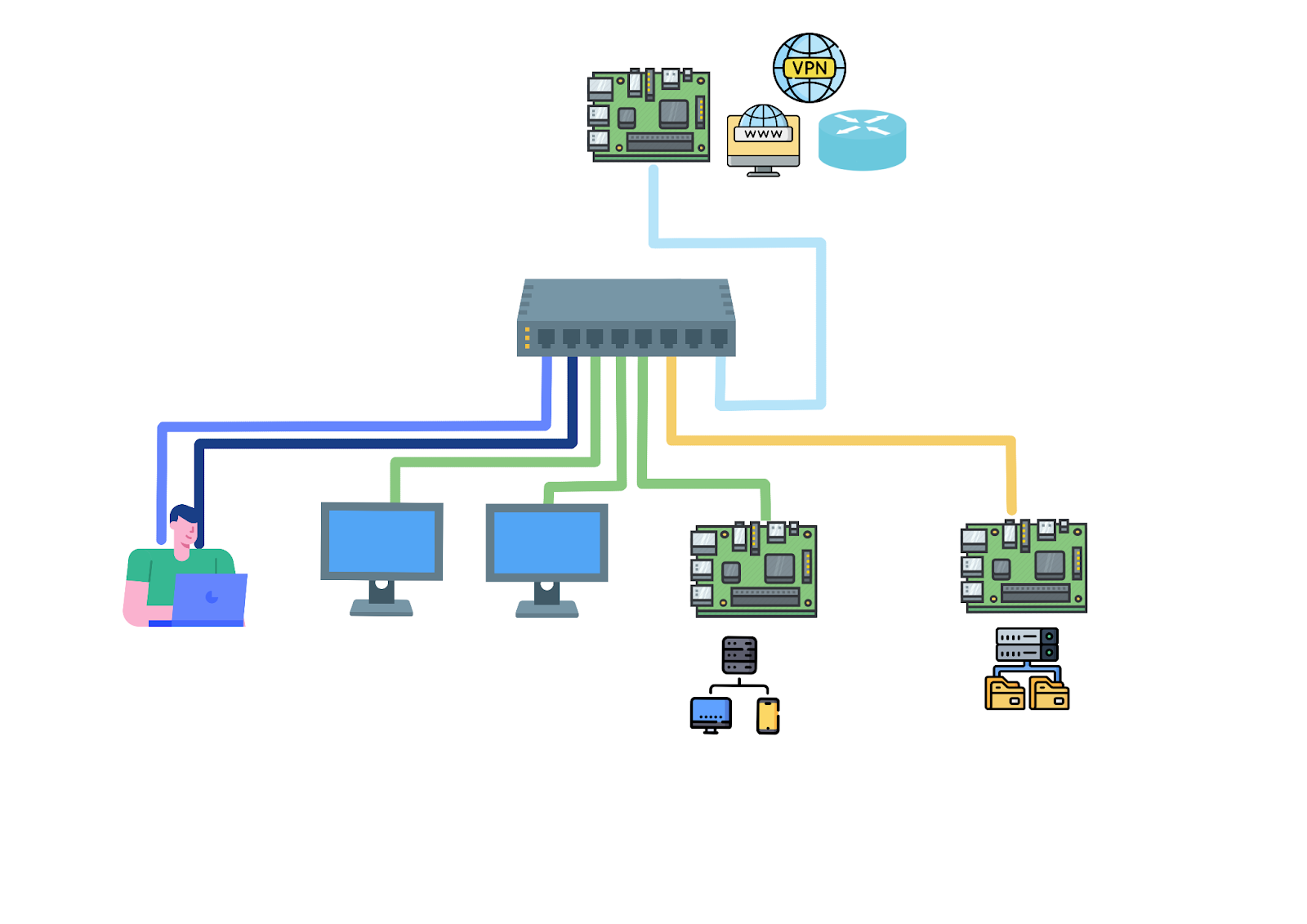
L'architecture réseau proposée utilise un commutateur 8 ports configurables pour diviser le réseau en plusieurs VLAN. Le VLAN 1 est utilisé pour configurer le commutateur, le VLAN 10 est le réseau principal qui contient les postes de travail et les serveurs, le VLAN 20 est utilisé pour le serveur de fichiers et d'application, et le VLAN 99 est réservé à l'administrateur système pour le contrôle des VLAN 10 et 20. Le Raspberry Pi 4 est utilisé comme routeur, pare-feu, DNS et VPN pour connecter les différents VLAN et offrir une solution sécurisée et flexible.

Choix des Raspberry Pi 4 : Le choix d'utiliser des Raspberry Pi 4 comme serveurs d'application et serveur DHCP a été motivé par plusieurs raisons. Tout d'abord, les Raspberry Pi 4 sont abordables, ce qui en fait une solution idéale pour les petites entreprises ayant un budget limité. En outre, ils sont faciles à configurer et à gérer grâce à une interface utilisateur conviviale.

En utilisant des Raspberry Pi 4 comme serveurs d'application et DHCP, on peut offrir une solution flexible qui peut être adaptée aux besoins de l'entreprise. Les serveurs d'application peuvent être configurés pour exécuter des logiciels spécifiques en fonction des besoins de l'entreprise, tandis que le serveur DHCP peut être configuré pour fournir des adresses IP à tous les périphériques connectés au réseau. En outre, l'utilisation de Raspberry Pi 4 permet de centraliser la gestion des serveurs sur une seule plateforme, ce qui simplifie la gestion et réduit les coûts.

Ainsi l'infrastructure réseau proposée pour une petite entreprise est une solution abordable, flexible et sécurisée. L'utilisation de Raspberry Pi 4 comme routeur, pare-feu, DNS, VPN, serveurs d'application et serveur DHCP permet de répondre aux besoins de l'entreprise tout en limitant les coûts et en simplifiant la gestion. En outre, la configuration en plusieurs VLAN du commutateur permet de séparer les différents services et limiter le trafic entre eux, ce qui assure la sécurité et la stabilité de l'infrastructure réseau.

# Schéma



Une fois que nous avons compris le travail à faire, nous l’avons divisé en trois parties :

1. Configuration du routage et du switch
   1. Routage entre VLANs
   2. Création des VLANs
   3. Création et configuration du service DNS
2. Création et configuration de serveurs intranet
   1. Serveurs DHCP
   2. Serveurs NAS (OpenMediaVault)
3. Création et configuration du VPN

# Vocabulaire

Avant de présenter le projet, il y a quelques termes importants à connaître :

1. Interface : Chaque port qui permet l’interconnexion entre des périphériques est appelé interface.
   1. Sous-interfaces : permettent de diviser une interface physique en plusieurs sous-interfaces logiques, chacune ayant ses propres paramètres de configuration réseau. (Généralement utilisées pour le système router-on-a-stick)
2. Paquet : Chaque fois qu’une périphérique A, envoie une donnée à une autre B, il encapsule les données et toutes les informations utiles pour le transport dans un paquet
3. Switch/Commutateur : un dispositif qui permet l’interconnexion sécurisée entre plusieurs périphériques via une interface généralement Ethernet ou fibre.
4. Virtual Lans (VLANs) : Chaque commutateur mangeable dispose d’un système de gestion de LAN virtuel par lequel, à l’aide de tag et untag des paquets (assignation/suppression de numéro d’identification du vlan sur le paquet), permet de le diviser en réseaux séparés et distincts ; brièvement, les LAN virtuels se trouvent physiquement dans le même switch, mais sont complètement séparés, ce qui implique qu’ils ont besoin d’un routeur pour communiquer entre eux.
5. Trunk : Une interface en mode Trunk permet au switch d’envoyer des paquets aux autres appareils avec les tags VLANs.
6. Routeur : Un routeur est un dispositif qui permet à deux ou plusieurs réseaux différents de communiquer entre eux, c’est ce qu’on appelle le routage.
7. Linux : Linux est un système d'exploitation open-source et gratuit, c'est-à-dire un logiciel qui gère les ressources d'un ordinateur et permet aux utilisateurs d'interagir avec celui-ci.
8. Kernel : Le noyau central du système d’exploitation Linux, gère les ressources matérielles de l’ordinateur et fournit une interface avec l’OS.
9. Pare-feu (Firewall): Un système de sécurité qui contrôle et limite le trafic réseau entre les périphériques et les réseaux ou entre deux réseaux différents.
10. Pilotes (Driver): Logiciel qui agit comme un intermédiaire entre le système d’exploitation et le périphérique matériel, leur permettant d’échanger des informations et de travailler ensemble efficacement.
11. Composants :
    1. CPU signifie "Central Processing Unit" en anglais, ce qui se traduit par Unité Centrale de Traitement. C'est le composant principal d'un ordinateur qui est responsable de l'exécution des instructions et des opérations de calcul.
    2. GPU signifie "Graphics Processing Unit" en anglais, ce qui se traduit par Unité de traitement graphique. C'est un processeur spécialisé conçu pour accélérer les calculs liés aux graphismes et à l'affichage.
    3. RAM, ou mémoire vive en français, est un type de mémoire informatique volatile qui stocke temporairement les données et les instructions que l'ordinateur utilise lors de son fonctionnement.
12. Overclocking : c’est une technique est utilisée pour améliorer les performances des composants comme CPU et GPU.

# Routage et VLANs

Afin de pouvoir travailler avec les équipements à notre disposition, nous avons opté pour une configuration routeur-on-a-stick, appelée comme ça car, représentée par schéma, (figure 1), il semble que le routeur soit posé sur un bâton.

Figure 1

Switch

#### Matériels

Router : Raspberry pi 4B 4Gb di RAM + Adaptateur USB Ethernet

Commutateurs : D-Link DGS-1210-10

Câbles : Ethernet cat 6

#### Commutateur et VLANs

Le commutateur D-Link DGS-1210-10, 8 ports avec interface de configuration web, adapté pour les petits réseaux, avec un prix abordable, environ 100€ et une consommation de courant relativement faible d’environ 13Wh.

#### Difficultés rencontrées

Le commutateur joue un rôle important dans ce projet, la division du réseau en réseaux virtuels (VLANs).

Dans le passé, j’ai travaillé exclusivement avec des équipements Cisco, qui ont une procédure de configuration totalement différente du DLink, donc la première difficulté, et aussi la plus importante, était de comprendre comment configurer les VLANs.

Malheureusement, nous n’avons pas trouvé un manuel spécifique de notre DLink ni dans sa boîte ni en ligne.

Après environ deux jours de recherche, grâce aux informations fragmentées que j’ai trouvées en ligne et à quelques conseils reçus d’un de nos collègues, j’ai compris deux choses qui m’ont permis de terminer la configuration du switch :

1. Il n’y a pas de mode Trunk dans les commutateurs DLink, on l’obtient en ajoutant manuellement le tag sur la porte en question pour toutes les VLANs
2. Il y a deux pages dans l’interface web importantes pour la création et la gestion des VLANs :
   1. Une s’occupe de la gestion des tags sur les interfaces.
   2. L’autre gère le comportement de l’interface avec les paquets en transit.

# Les VLAN

Les VLAN (Virtual Local Area Network) permettent de diviser un réseau physique en plusieurs réseaux logiques indépendants. Ce qui garantit une sécurité accrue et une meilleure organisation du trafic réseau en limitant les accès non autorisés à certaines parties du réseau.

Les différents VLAN ont des numéros d'identification uniques, appelés VID (VLAN ID), qui les distinguent les uns des autres. Les numéros d'identification peuvent varier selon les fabricants d'équipements de réseau, mais certains sont généralement réservés à des utilisations spécifiques :

ip réseau interne 172.16.0.0/16

* VLAN 1

Est utiliser pour se connecter au switch et le configure

Subnet Address: 172.16.0.0/24

Host Address Range: 172.16.0.1 - 172.16.0.254

Broadcast Address: 172.16.0.255

* LAN 10 (LAN)

Contient les postes utilisateur et le serveur DHCP

Subnet Address: 172.16.1.0/24

Host Address Range: 172.16.1.1 - 172.16.1.254

Broadcast Address: 172.16.1.255

* VLAN 20 (serveur de fichier et d’application)

Subnet Address: 172.16.2.0/24

Host Address Range: 172.16.2.1 - 172.16.2.254

Broadcast Address: 172.16.2.255

* VLAN 99 : (Management)

<https://youtu.be/5CENnfgniGI>

#### Routage

En tant que routeur, nous avons utilisé un Raspberry Pi 4B, un ordinateur à carte unique, avec les spécifications suivantes :

* CPU : ARM Cortex-A72 Quad-core 1,5 GHz
* RAM : 4 Go ddr4
* Connectivité Ethernet
* Connectivité Wi-Fi/Bluetooth.

Il peut être utilisé pour différentes applications, notamment :

* Serveur
* Projets IoT
* Robotique
* Routage.

Il se distingue par son bas cout, sa polyvalence et la facilité d’utilisation.

Pour la configuration router-on-a-stick, il a besoin de deux interfaces RJ45, mais notre carte n’en a qu’une, nous avons donc acheté une carte réseau externe, compatible avec Linux.

#### Difficultés rencontrées

Dans la gestion du routage via Raspberry pi la première étape est le choix du système d’exploitation, dans notre cas Openwrt, un système d’exploitation (OS) basé sur une version lite du kernel linux, qui permet, via une interface web, la gestion du routage et du pare-feu.

Après plusieurs tests, nous avons remarqué l’absence de compatibilité entre l’OS et l’adaptateur USB-Ethernet, le premier instinct a été de consulter la [documentation](https://openwrt.org/toh/raspberry_pi_foundation/raspberry_pi) d’Openwrt ; en fait elle indique que les seules cartes supporter sont Realteck ou Intel.

Pour résoudre le problème, nous avons envisagé trois solutions possibles :

* Ajouter manuellement le pilote
* Acheter un nouvel adaptateur
* Changer d’OS

La première option est impraticable, car le kernel empêche l’installation de tout paquet supplémentaire, ce qui est compréhensible du point de vue de la sécurité.

La deuxième option, vus les prix élevés des adaptateurs USB-Ethernet Realteck/Intel compatibles avec Linux, il a été immédiatement écarter.

Donc, nous avons opté pour un changement d’OS en installant un Raspbian x64 Lite (Sans interface graphique).

En se basant sur le TP fait avec le professeur Alexandre GUY, j’ai écrit un script adapté pour la gestion du routage et du pare-feu que j’ai mis en exécution au démarrage.

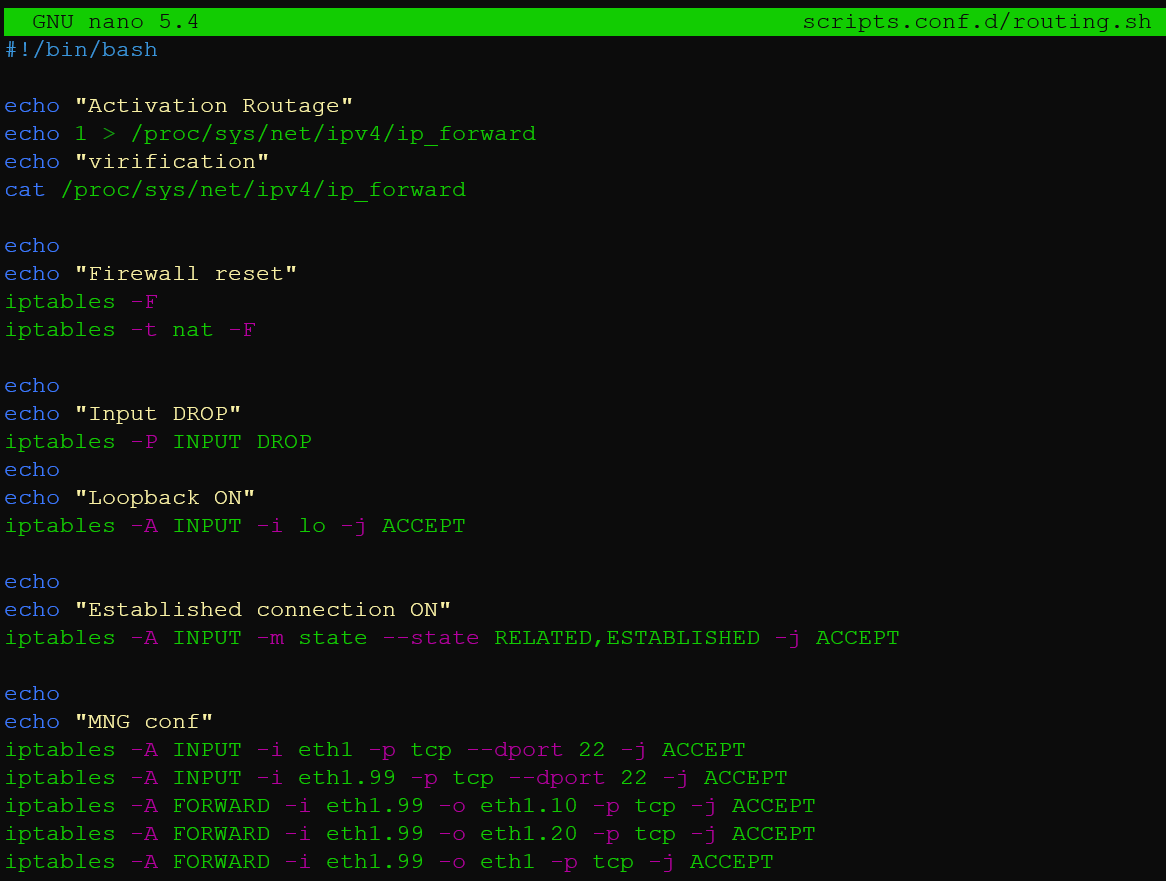


Fig i Script Routage Page 1

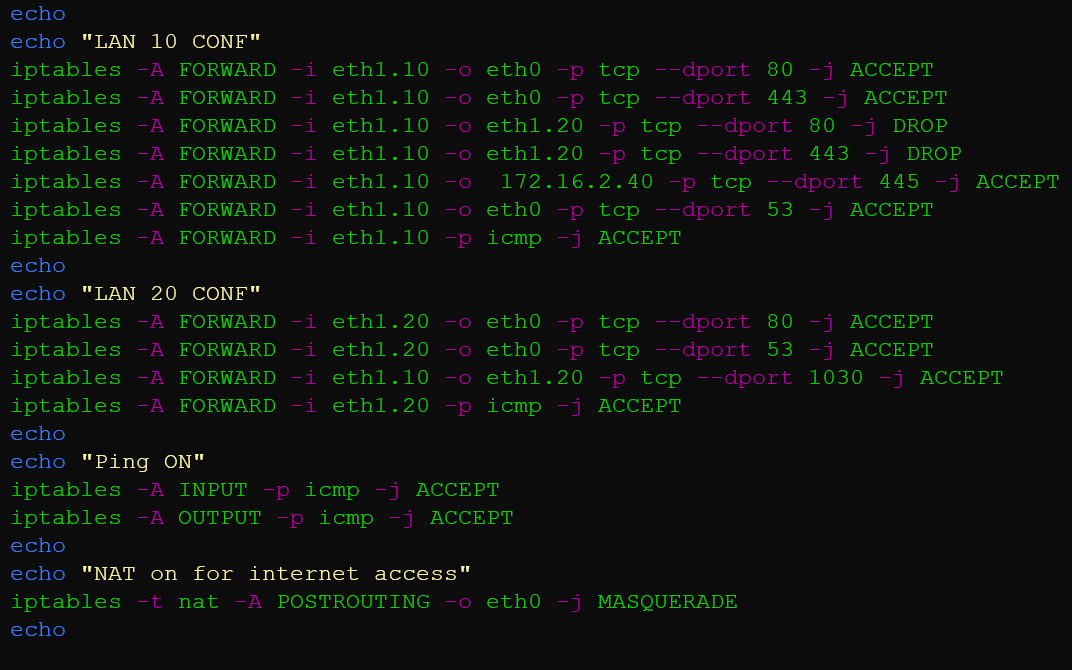


Fig ii Script Routage Page 1

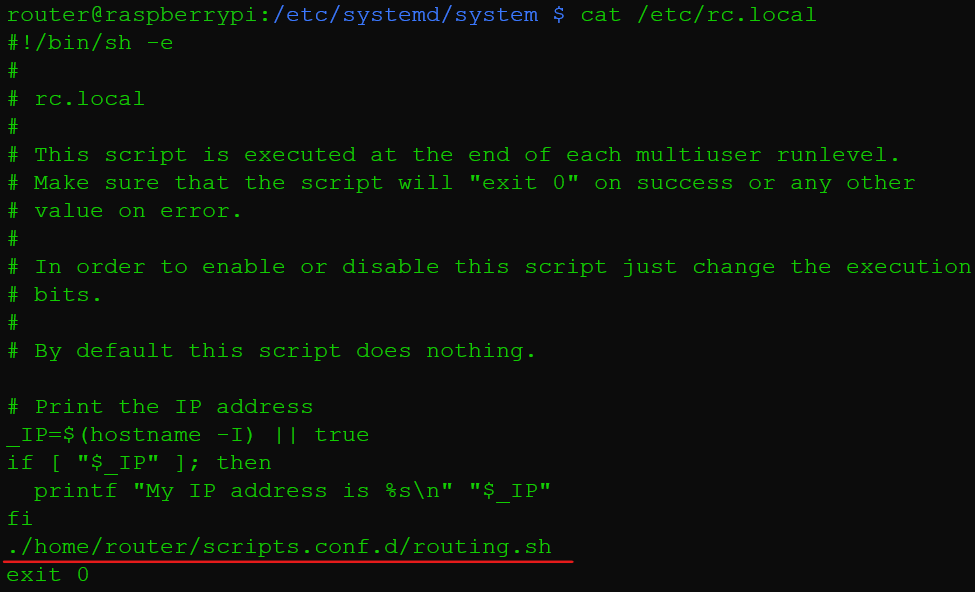


Fig iii fichier modifié pour l'exécution du script au démarrage

Nous avons ensuite rencontré un deuxième problème dans la gestion du routage entre vlans : le système de contrôle des interfaces de base dans Raspbian, avait du mal à gouverner les sous-interfaces.

Nous avons donc décidé de le configurer avec le système Netplan, qui permet une gestion concentrée des interfaces et sous-interfaces, à travers le même fichier. [[Source](https://askubuntu.com/questions/992428/netplan-with-multiple-vlans-on-single-interface-help-needed)]

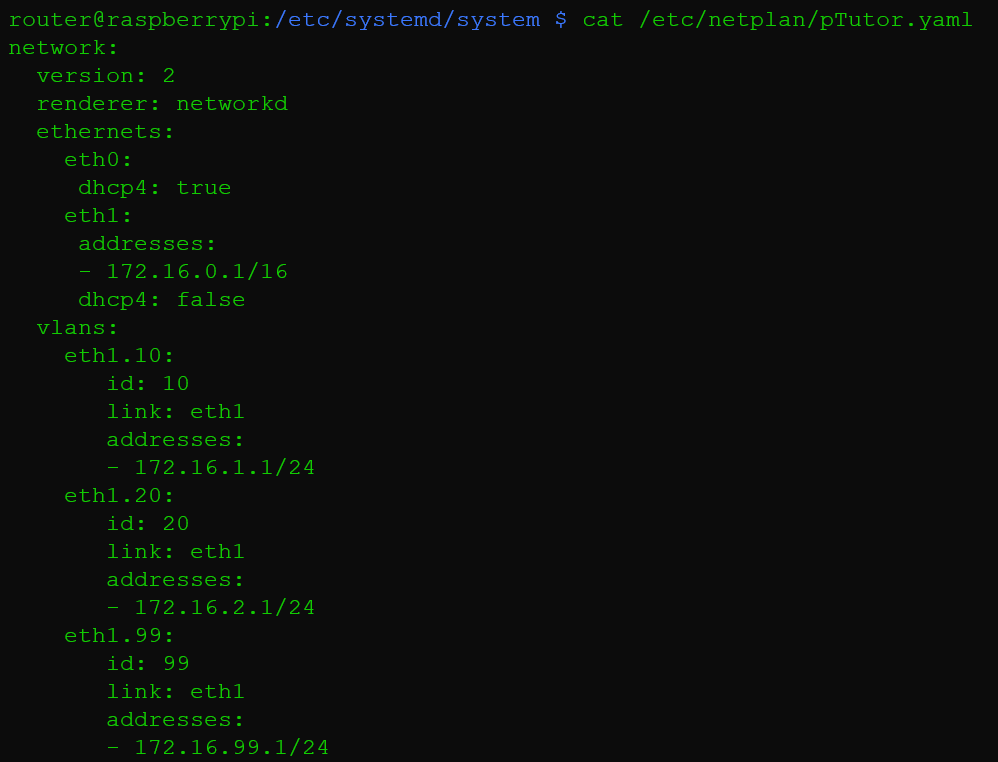


Fig iv fichier Netplan du router

Le troisième et dernier problème rencontré dans le routage, c’est la vitesse, même si nous avions un port RJ45 1Gb/s le processeur avait du mal à gérer cette quantité de données, donc nous avons décidé de faire un overclocking partiel du processeur, en suivant un [tutoriel](https://www.youtube.com/watch?v=TwDbQ26_Mp8&t=16s), trouvé en ligne.

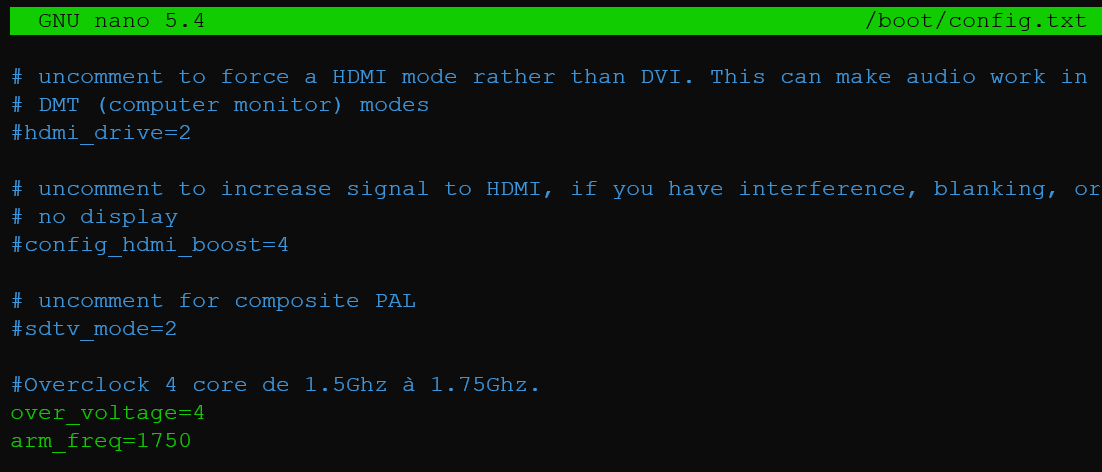


Fig Lignes à ajouter pour overclocker

# Serveur de fichier

**OpenMediaVault:**

OpenMediaVault est un logiciel qui peut être installé sur toute distribution basée sur Debian, comme Raspberry Pi OS (Lite est suffisant). Il peut être utilisé pour héberger et configurer un serveur de fichiers en quelques clics via une interface web intuitive.

Open MediaVault est une solution appropriée à notre projet pour les raison suivante

Gestion de stockage : OpenMediaVault offre une gestion complète des disques durs, de la sauvegarde et de la restauration des données, et permet de créer facilement des espaces de stockage partagés pour les utilisateurs.

Interface utilisateur : l'interface Web d'OpenMediaVault est simple et intuitive, ce qui facilite la configuration et la gestion de la plateforme pour les utilisateurs.

Plugins : OpenMediaVault prend en charge de nombreux plugins pour ajouter des fonctionnalités supplémentaires, telles que la surveillance du système, le partage de fichiers via FTP, la configuration de RAID, etc.

Sécurité : OpenMediaVault propose des options de sécurité avancées, telles que la gestion des utilisateurs et des permissions, le chiffrement des disques.

**Configuration:**

**Configuration du Raspberry**

root = Azerty66 (desactive)

username: admisys

password: Azerty66

static ip\_adresse=172.16.2.40

**Configuration du réseau avec netplan**

Netplan est un outil de configuration de réseau pour les systèmes d'exploitation basés sur Linux, utilisé pour définir et configurer les interfaces réseau sur des ordinateurs avec des systèmes d'exploitation tels que Ubuntu. Il permet de définir les configurations réseau dans un fichier YAML, qui est ensuite converti en configurations système standard telles que les fichiers de configuration de NetworkManager ou de systemd-networkd.

**Configuration d’OpenMediaVault**

La mise en place du serveur de fichier est présenté dans la vidéo ci dessous :

<https://www.youtube.com/watch?v=7kUVd3-VuAY>

Il est possible d'ajouter des utilisateur manuellement ou avec un csv , cela offre la possibilité de créer des script pour automatiser la création d’utilisateur Lors de l’ajout manuel d’un utilisateur, OpenMediaVault utilise la commande useradd.

**Serveur DHCP**

**Présentation de DNSmasq**

Dnsmasq est un serveur DNS et DHCP léger pour systèmes Linux. Se servir de DNSmasq comme serveur DHCP sur le Raspberry Pi est une solution adaptée. En effet, ce service a été conçu pour fournir une solution simple et rapide pour les petits réseaux locaux. Il est léger et facile à configurer et à utiliser, même pour les débutants. Il est capable de fournir des adresses IP et des informations de configuration aux clients sur le réseau, tout en offrant une gestion simple de la résolution de nom. De plus DNSmasq est compatible avec une grande variété de systèmes d'exploitation, de matériels et de protocoles. Il peut être utilisé pour fournir des adresses IP aux clients Windows, Linux, macOS et autres.

En somme, DNSmasq est une solution simple, légère et polyvalente pour un serveur DHCP, qui répond parfaitement aux besoins d'une petite entreprise.

**Configuration**

**Configuration du Raspberry**

root = Azerty66 (desactive)

username: admisys

password: Azerty66

static ip\_adresse=172.16.1.250/24

Comme pour le serveur d’application l’ip statique est configuré avec **netplan.** La configuration du service **DNSmasq** est présenté dans la vidéo suivante :

<https://youtu.be/_ZNcAG8to_g>

# VPN

**Bibliographie**

**Service d’application OpenMediaVault**

[💻 CREEZ votre SERVEUR NAS avec un RASPBERRY PI sur OPENMEDIAVAULT ! Serveur de fichiers](https://www.youtube.com/watch?v=dKjfwzs5Myw&t)

<https://github.com/OpenMediaVault-Plugin-Developers/installScript>

<https://youtu.be/7kUVd3-VuAY>

**VPN**

[Raspberry | Faire son propre serveur VPN openVPN | Tutoriel](https://www.youtube.com/watch?v=HcPcdwacnOk&t=357s)

<https://github.com/pivpn/pivpn>

**Config du serveur DHCP**

<https://youtu.be/_ZNcAG8to_g>

**VLAN 99**

<https://youtu.be/5CENnfgniGI>

Links:  
<https://www.tala-informatique.fr/wiki/index.php/Iproute2>

<https://linux.goffinet.org/administration/configuration-du-reseau/gestion-du-reseau-linux-avec-iproute2/>

<https://www.digitalocean.com/community/tutorials/how-to-list-and-delete-iptables-firewall-rules>

<https://www.daddaily.life/building-a-router-with-raspberry-pi-router-on-a-stick-en/>