UNIVERSITÉ DE FRANCHE-COMTÉ

MOURAD AMGHAR
SAMI HARCHAOUI
LOCQMEN HAMDI

2A/2B

JALON 4

Au cours de notre projet jalonné, nous avons exploré divers aspects de la mise en place d'un environnement de streaming vidéo et de partage de fichiers. Ces expérimentations ont impliqué l'utilisation de matériel informatique divers, notamment des machines virtuelles et un Raspberry Pi. Nous avons également abordé la configuration d'un serveur vidéo, la création de pages HTML pour la diffusion de contenus multimédias, et l'utilisation de JavaScript pour optimiser la lecture vidéo, en particulier avec le lecteur Shaka Player. De plus, nous avons examiné la mise en place de VPN avec WireGuard pour sécuriser les communications réseau. Enfin, nous avons exploré la création d'un serveur NAS pour le stockage de fichiers, en utilisant à la fois des partages Samba et FTP.

Matériels utilisés pour la deuxième partie SAE:

Dans cette SAE nous avons premièrement utilisé une machine virtuelle que nous avons créé précédemment avec des caractéristiques attendues.

De plus, nous avons aussi utilisé un Raspberry Pi à partir duquel nous nous connections grâce à VNC.

Ensuite nous avons aussi créé une autre Machine Virtuelle mais cette fois-ci aucune interface graphique dessus. Tout serait géré depuis openmediavault Workbench.

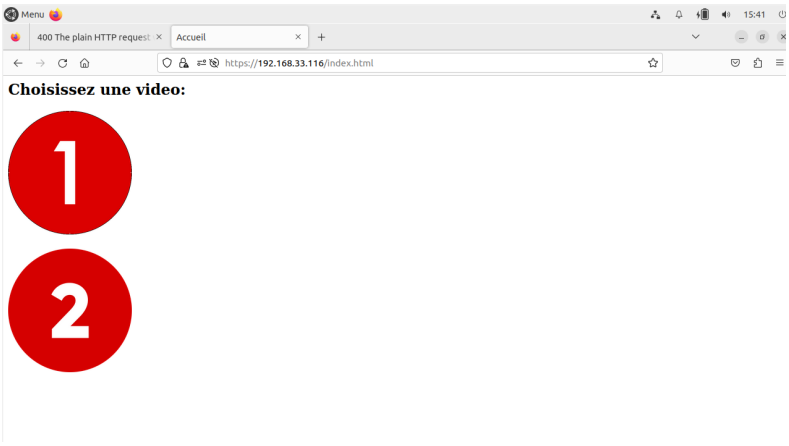
Mise en place du serveur vidéo:

Un serveur vidéo est un ordinateur ou un système informatique qui stocke, gère et distribue des fichiers vidéo sur un réseau, permettant ainsi aux utilisateurs d'accéder et de diffuser des contenus vidéo en ligne.

Html:

Tout d'abord, dans le cadre de notre projet, nous avons commencé par créer une page HTML. Cette page, que nous nommerons "index.html", constitue notre point de départ. Sur cette page, nous avons incorporé deux images, chacune associée à un lien hypertexte. Lorsqu'un utilisateur clique sur l'une de ces images, cela déclenche un changement de page HTML, le conduisant vers une nouvelle destination.

La page HTML de destination, vers laquelle chaque image redirige, est conçue pour être un lecteur vidéo. Cette page est conçue spécifiquement pour permettre la lecture de vidéos. Le lecteur vidéo peut prendre en charge divers formats, tels que le MP4 ou le MPD, et offrir une expérience de visualisation interactive. En résumé, en utilisant une combinaison d'images et de pages HTML, nous avons mis en place un système de navigation qui permet aux utilisateurs de basculer entre une page de choix (index.html) et une page de lecture vidéo, offrant ainsi une expérience de visualisation multimédia fluide.



Javascript:

Ce script JavaScript configure Shaka Player, une bibliothèque spécialisée dans la lecture de vidéos adaptatives, pour lire une vidéo au format MPD. Il commence par préparer le lecteur en installant les polyfills nécessaires pour assurer une compatibilité maximale avec les navigateurs. Ensuite, il identifie un élément HTML de la page, généralement une balise vidéo avec l'ID "MaVideo", qui servira de conteneur pour la vidéo. Le lecteur Shaka est associé à cet élément, permettant la lecture et l'affichage de la vidéo. Enfin, le script spécifie l'URL ou le chemin du manifeste MPD de la vidéo, un fichier XML qui contient des informations sur les différentes versions de la vidéo et les segments associés, permettant ainsi une lecture adaptative en fonction de la qualité de la connexion Internet de l'utilisateur.

```
shaka.polyfill.installAll();

var video = document.getElementById('MaVideo');
var player = new shaka.Player(video);
window.player = player;

var UrlMpd = 'nom_du_mpd';

player.load(UrlMpd);
```

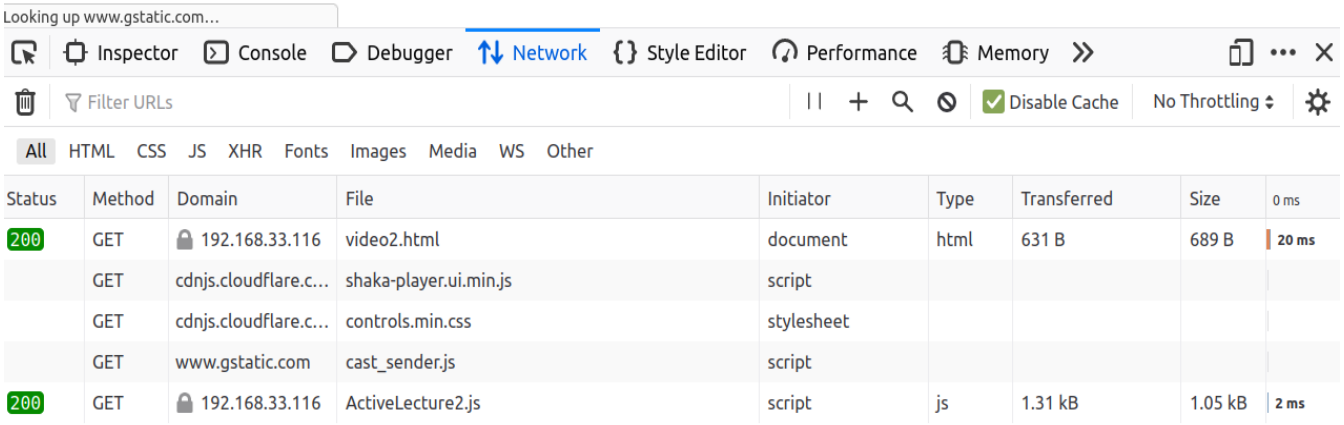
En complément, nous avons développé un script distinct qui offre la capacité de transcoder un fichier vidéo au format MP4, ayant une résolution de 1080p, en trois fichiers MPD segmentés, chacun à des résolutions différentes : 1080p, 720p et 480p. Cette opération de transcodage vise à adapter la vidéo à diverses conditions de bande passante et de dispositifs de lecture. La vidéo originale en 1080p est divisée en segments, et pour chaque segment, trois versions de qualité différente sont générées.

Diffusion de Contenus MP4 et MPD:

La lecture adaptative est un élément clé de cette solution. Le lecteur Shaka Player est capable de démarrer la lecture de la vidéo en se basant sur les informations du manifeste MPD. Tout au long de la lecture, il surveille constamment la vitesse de la connexion Internet de l'utilisateur. Si la

connexion devient plus lente, le lecteur réagit instantanément en ajustant automatiquement la qualité de la vidéo. Par exemple, il peut passer de la résolution 1080p à 720p ou même à 480p

pour éviter les interruptions de lecture et les problèmes de mise en mémoire tampon. De même, si la connexion s'améliore, le lecteur peut revenir à une meilleure qualité vidéo. Cette adaptation dynamique garantit que les spectateurs bénéficient d'une expérience de visionnage fluide et optimale, en profitant de la meilleure qualité possible en fonction des conditions réseau, offrant ainsi une expérience de streaming vidéo de qualité, même en cas de connexions Internet moins rapides.

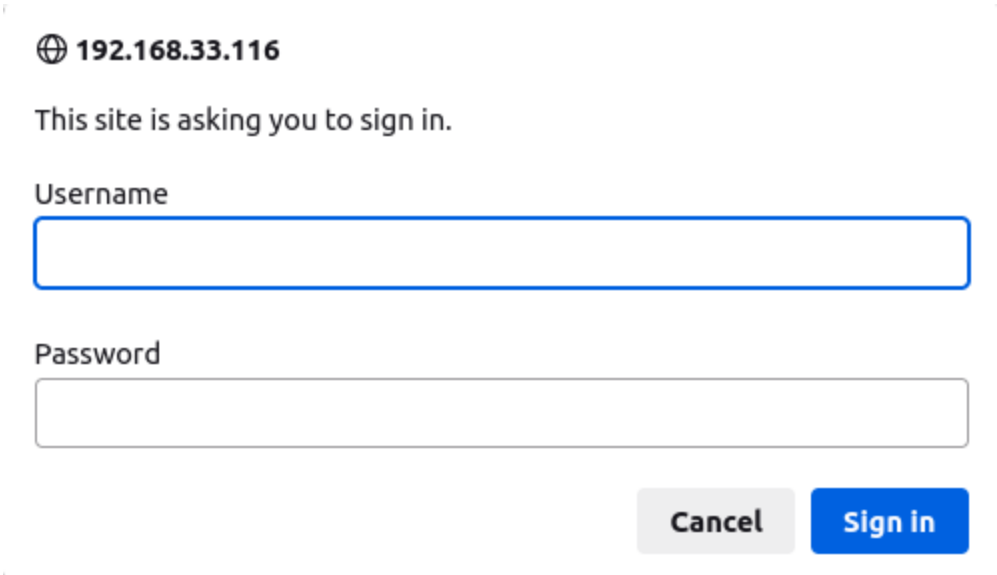


Https et Authentification:

Passer votre site web hébergeant le serveur VoD en HTTPS (port 443) est une étape cruciale pour sécuriser les échanges de données entre le serveur et les utilisateurs. Cela garantit que les informations transmises, telles que les identifiants de connexion et les données vidéo, sont cryptées, réduisant ainsi les risques de piratage ou d'interception de données sensibles. La création d'un certificat autosigné avec des outils tels qu'OpenSSL est une solution viable pour des environnements de développement ou d'essai, mais dans un contexte de production, il est recommandé d'obtenir un certificat SSL valide auprès d'une autorité de certification reconnue.

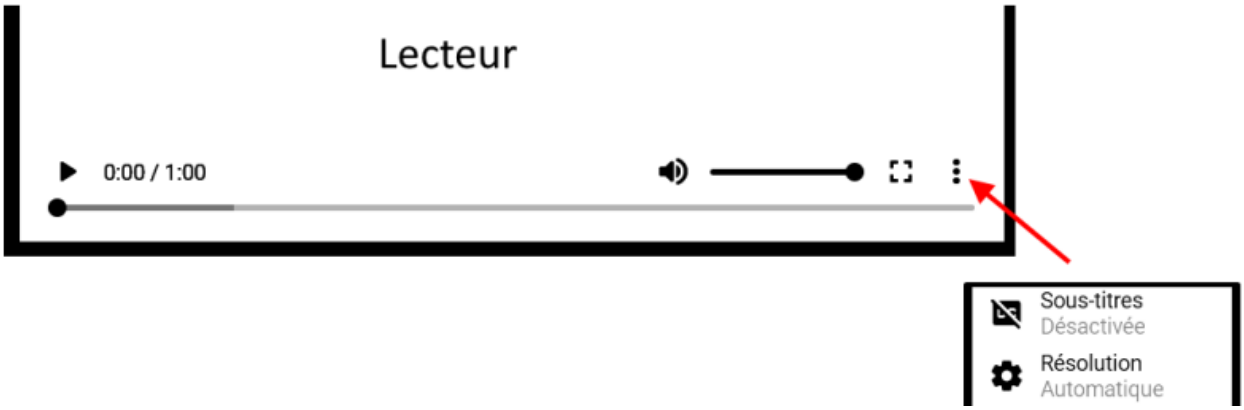
Une fois que le certificat est généré, vous devez configurer votre serveur pour utiliser ce certificat, ce qui signifie que les connexions vers votre site se feront via HTTPS.

L'activation de l'authentification, où le serveur demande un nom d'utilisateur et un mot de passe, renforce la sécurité de votre serveur VoD en restreignant l'accès aux utilisateurs autorisés. Vous pouvez créer un compte administrateur avec un mot de passe sécurisé de votre choix, qui sera nécessaire pour accéder aux fonctionnalités d'administration ou à des zones sensibles du site. Cette couche de sécurité garantit que seules les personnes disposant des identifiants appropriés peuvent effectuer des opérations importantes sur le serveur VoD, ce qui contribue à protéger vos ressources vidéo et à prévenir tout accès non autorisé. Il est essentiel de gérer les comptes d'utilisateurs et de s'assurer que les mots de passe sont suffisamment robustes pour éviter toute vulnérabilité potentielle.



Bonus:

Le bonus lié à la partie du serveur vidéo consistait à intégrer une roue crantée dans l'interface du lecteur vidéo, symbolisée par une icône d'engrenage. Cette roue crantée permet aux utilisateurs de modifier manuellement la qualité de la vidéo en cours de lecture, adaptant ainsi leur expérience de visionnage à leurs préférences ou aux fluctuations de la vitesse de leur connexion Internet. De plus, cette fonctionnalité peut également être configurée pour s'ajuster automatiquement en fonction des conditions du réseau, garantissant une lecture fluide même en cas de connexion lente, et offrant une meilleure qualité lorsque la connexion s'améliore.



Mise en place d'un Vpn Wireguard:

WireGuard est un protocole de réseau virtuel privé (VPN) open source reconnu pour sa simplicité et son efficacité WireGuard s'appuie sur une cryptographie moderne pour établir des connexions VPN sécurisées. Il se distingue par son code simple et transparent, sa légèreté, son installation aisée, une utilisation minimale des ressources système, et des avantages en termes de performances.

Interfaces:

Voici un ifconfig sur notre VM et RPi on l'on peut voir l'interface wg0 (wireguard)

```
root@saeserv:~# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.33.116  netmask 255.255.255.0  broadcast 192.168.33.255
        ether 08:00:27:f9:a9:16  txqueuelen 1000  (Ethernet)
        RX packets 8144337  bytes 1958072133 (1.9 GB)
        RX errors 0  dropped 2  overruns 0  frame 0
        TX packets 914500  bytes 737367558 (737.3 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 216045  bytes 1494378586 (1.4 GB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 216045  bytes 1494378586 (1.4 GB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wg0: flags=209<UP,POINTOPOINT,RUNNING,NOARP>  mtu 1420
        inet 10.0.0.1  netmask 255.255.255.0  destination 10.0.0.1
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 1000  (UNSPEC)
        RX packets 12  bytes 1556 (1.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 16  bytes 1976 (1.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@saeserv:~# S
```

```
pi@raspberrypi:~ $ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.33.16  netmask 255.255.255.0  broadcast 192.168.33.255
        inet6 fe80::87a:9525:412e:c62e  prefixlen 64  scopeid 0x20<link>
        ether dc:a6:32:ad:54:fb  txqueuelen 1000  (Ethernet)
        RX packets 331505  bytes 272535559 (259.9 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 143912  bytes 29872619 (28.4 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 65  bytes 6314 (6.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 65  bytes 6314 (6.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wg0: flags=209<UP,POINTOPOINT,RUNNING,NOARP>  mtu 1420
        inet 10.0.0.2  netmask 255.255.255.0  destination 10.0.0.2
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 1000  (UNSPEC)
        RX packets 84944  bytes 123124256 (117.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 43090  bytes 4127260 (3.9 MiB)
        TX errors 23  dropped 0 overruns 0  carrier 0  collisions 0
```

Ports:

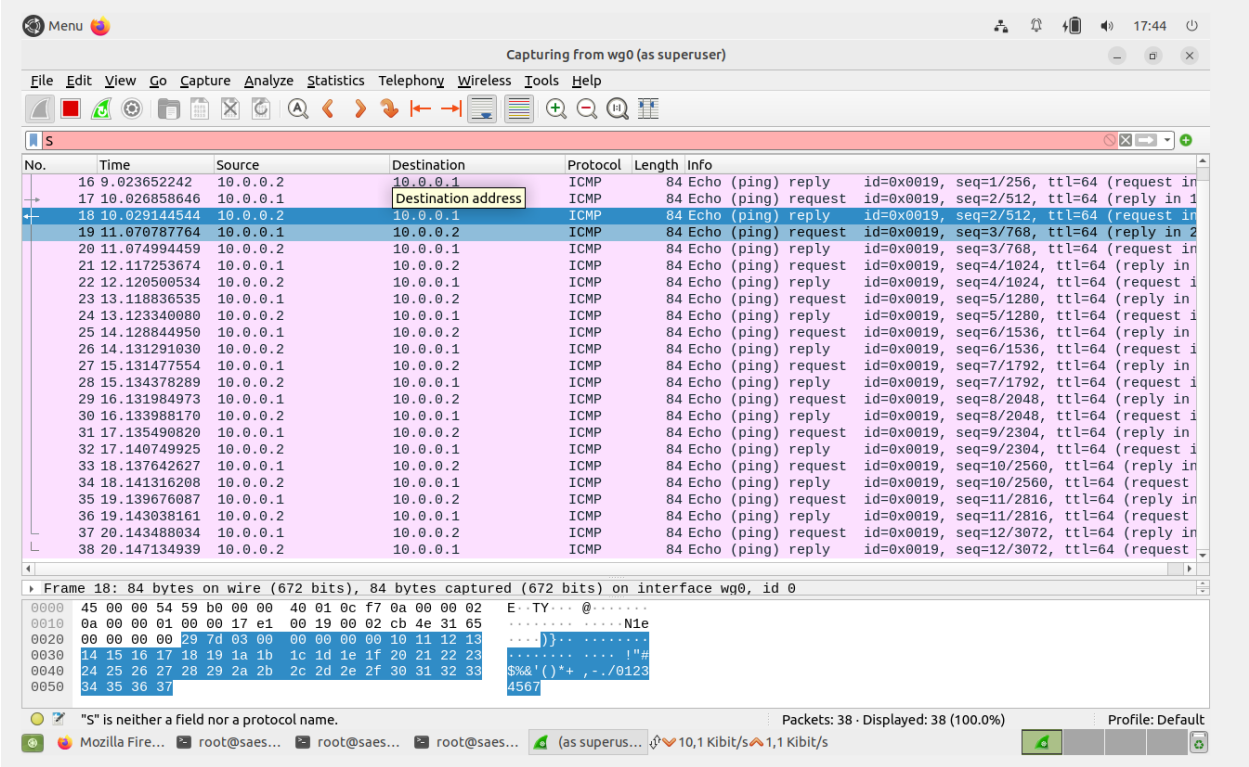
Voici un netstat -plantu sur la VM et le RPi permettant de voir le port utilisé par Wireguard.

```
root@saeserv:~# netstat -plantu
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      155736/nginx: maste
tcp        0      0 0.0.0.0:443             0.0.0.0:*               LISTEN      155736/nginx: maste
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      104928/cupsd
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      492/systemd-resolve
tcp        0      0 0.0.0.0:1935            0.0.0.0:*               LISTEN      155736/nginx: maste
tcp        0      0 127.0.0.1:80            127.0.0.1:44196         TIME_WAIT   -
tcp        0      0 192.168.33.116:55770    216.58.205.196:443      ESTABLISHED 25199/firefox
tcp        0      0 127.0.0.1:42312         127.0.0.1:4949          TIME_WAIT   -
tcp        0      0 192.168.33.116:34416    34.120.208.123:443      ESTABLISHED 25199/firefox
tcp        0      0 192.168.33.116:51754    142.250.200.193:443     TIME_WAIT   -
tcp        0      0 192.168.33.116:37560    34.107.221.82:80        ESTABLISHED 25199/firefox
tcp        0      0 192.168.33.116:39612    172.64.150.28:443       TIME_WAIT   -
tcp        0      0 127.0.0.1:44202         127.0.0.1:80            TIME_WAIT   -
tcp        0      0 192.168.33.116:54240    192.168.33.116:443      ESTABLISHED 25199/firefox
tcp        0      0 192.168.33.116:443      192.168.33.116:54240    ESTABLISHED 155737/nginx: worke
tcp        0      0 192.168.33.116:37546    34.107.221.82:80        ESTABLISHED 25199/firefox
tcp        0      0 127.0.0.1:80            127.0.0.1:46060         TIME_WAIT   -
tcp        0      0 192.168.33.116:58098    34.120.115.102:443      ESTABLISHED 25199/firefox
tcp        0      0 192.168.33.116:32842    34.117.65.55:443        ESTABLISHED 25199/firefox
tcp        0      0 192.168.33.116:43906    34.233.138.108:443      ESTABLISHED 25199/firefox
tcp        0      0 192.168.33.116:41924    34.117.237.239:443      TIME_WAIT   -
tcp6       0      0 :::80                   :::*                    LISTEN      155736/nginx: maste
tcp6       0      0 ::1:631                 :::*                    LISTEN      104928/cupsd
tcp6       0      0 :::433                  :::*                    LISTEN      155736/nginx: maste
tcp6       0      0 :::4949                 :::*                    LISTEN      52506/perl
udp        0      0 127.0.0.53:53           0.0.0.0:*                           492/systemd-resolve
udp        0      0 0.0.0.0:51820           0.0.0.0:*                           -
udp        0      0 0.0.0.0:631             0.0.0.0:*                           104945/cups-browsed
udp        0      0 0.0.0.0:33957           0.0.0.0:*                           596/avahi-daemon: r
udp        0      0 0.0.0.0:5353            0.0.0.0:*                           596/avahi-daemon: r
udp6       0      0 :::51820                :::*                                -
udp6       0      0 :::39878                :::*                                596/avahi-daemon: r
udp6       0      0 :::5353                 :::*                                596/avahi-daemon: r
root@saeserv:~#
```

```
pi@raspberrypi:~ $ netstat -plantu
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:5900            0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:55104         127.0.0.1:37635         ESTABLISHED -
tcp        0      0 192.168.33.16:5900      192.168.33.89:6620      ESTABLISHED -
tcp        0      0 127.0.0.1:37635         127.0.0.1:55104         ESTABLISHED 1111/vncserverui
tcp6       0      0 :::4949                 :::*                    LISTEN      -
tcp6       0      0 :::5900                 :::*                    LISTEN      -
tcp6       0      0 ::1:631                 :::*                    LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
udp        0      0 0.0.0.0:631             0.0.0.0:*                           -
udp        0      0 0.0.0.0:68              0.0.0.0:*                           -
udp        0      0 224.0.0.251:5353        0.0.0.0:*                           3623/chromium-brows
udp        0      0 224.0.0.251:5353        0.0.0.0:*                           3623/chromium-brows
udp        0      0 0.0.0.0:5353            0.0.0.0:*                           -
udp        0      0 0.0.0.0:51514           0.0.0.0:*                           -
udp        0      0 0.0.0.0:36350           0.0.0.0:*                           -
udp6       0      0 :::5353                 :::*                                -
udp6       0      0 :::51514                :::*                                -
udp6       0      0 :::51694                :::*                                -
pi@raspberrypi:~ $
```

Ping chiffré:

Voici une capture wireshark d’un ping chiffré grâce à la commande “ping -I 10.0.0.2” .



<u>Clé de chiffrement:</u>

Voici deux captures mettant en évidence la clé publique et privée de notre serveur VPN.

```
root@saeserv:~# cat server_public.key
NJB56Dno6EOVaZ06V2FA3YA6ZWDf1dykK2yisOLbNg4=
root@saeserv:~#
```

```
rtt min/avg/max/mdev = 1.998/3.396/18.438/2.522 ms
root@saeserv:~# cat server_private.key
8Nr4OQ3aJ89RuGckf6X4bmvQtkyTgi1+utNszyREyUU=
root@saeserv:~#
```

Protocole de chiffrement:

WireGuard utilise le chiffrement moderne pour sécuriser les données transitant sur le réseau. Il se base principalement sur le protocole de chiffrement Curve25519 pour l'échange de clés, le protocole Noise pour les négociations de sécurité et le chiffrement AES-256 pour chiffrer les données.

Test iperf classique et VPN:

```
root@saeserv:~# iperf3 -c 192.168.33.16
Connecting to host 192.168.33.16, port 5201
[  5] local 192.168.33.116 port 39016 connected to 192.168.33.16 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec  13.3 MBytes   111 Mbits/sec    0    591 KBytes
[  5]   1.00-2.00   sec  12.4 MBytes   104 Mbits/sec    0   1.13 MBytes
[  5]   2.00-3.00   sec  10.0 MBytes  83.8 Mbits/sec    0   1.69 MBytes
[  5]   3.00-4.00   sec  11.2 MBytes  94.4 Mbits/sec    0   2.25 MBytes
[  5]   4.00-5.00   sec  11.2 MBytes  94.3 Mbits/sec    0   2.81 MBytes
[  5]   5.00-6.00   sec  11.2 MBytes  94.5 Mbits/sec    0   3.13 MBytes
[  5]   6.00-7.00   sec  11.2 MBytes  94.3 Mbits/sec    0   3.13 MBytes
[  5]   7.00-8.00   sec  11.2 MBytes  94.4 Mbits/sec    0   3.13 MBytes
[  5]   8.00-9.00   sec  11.2 MBytes  94.4 Mbits/sec    0   3.13 MBytes
[  5]   9.00-10.00  sec  11.2 MBytes  94.4 Mbits/sec    0   3.13 MBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec   114 MBytes  96.0 Mbits/sec    0             sender
[  5]   0.00-10.30  sec   114 MBytes  93.2 Mbits/sec                  receiver

iperf Done.
root@saeserv:~#
```

```
root@saeserv:~# iperf3 -c 10.0.0.2
Connecting to host 10.0.0.2, port 5201
[  5] local 10.0.0.1 port 40612 connected to 10.0.0.2 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec  12.3 MBytes   103 Mbits/sec    0    574 KBytes
[  5]   1.00-2.00   sec  12.0 MBytes   101 Mbits/sec    0   1.09 MBytes
[  5]   2.00-3.00   sec  11.2 MBytes  94.4 Mbits/sec    0   1.63 MBytes
[  5]   3.00-4.00   sec  10.0 MBytes  83.8 Mbits/sec    0   2.17 MBytes
[  5]   4.00-5.00   sec  11.2 MBytes  94.4 Mbits/sec    0   2.31 MBytes
[  5]   5.00-6.00   sec  10.0 MBytes  83.9 Mbits/sec    0   2.31 MBytes
[  5]   6.00-7.00   sec  11.2 MBytes  94.4 Mbits/sec    0   2.47 MBytes
[  5]   7.00-8.00   sec  10.0 MBytes  83.9 Mbits/sec    0   2.59 MBytes
[  5]   8.00-9.00   sec  11.2 MBytes  94.4 Mbits/sec    0   2.59 MBytes
[  5]   9.00-10.00  sec  11.2 MBytes  94.4 Mbits/sec    0   2.59 MBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec   111 MBytes  92.8 Mbits/sec    0             sender
[  5]   0.00-10.28  sec   110 MBytes  89.3 Mbits/sec                  receiver

iperf Done.
root@saeserv:~#
```

Mise en place d’un serveur Nas:

Un serveur NAS, acronyme de "Network-Attached Storage", est un appareil de stockage connecté à un réseau informatique qui permet le stockage, la gestion et le partage de fichiers et de données. Contrairement aux serveurs traditionnels, un serveur NAS est généralement conçu pour une seule tâche principale : fournir un espace de stockage en réseau.

Tout d’abord nous avons créé la Machine Virtuelle avec une iso openmediavault pour pouvoir tout contrôler depuis une interface graphique en écrivant l'adresse IP de notre serveur Nas dans l’url d’un navigateur.

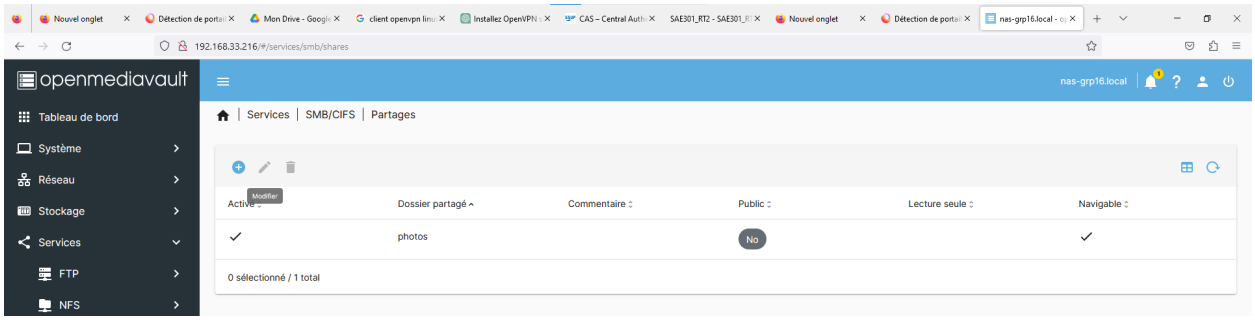
Partage Samba:

Un serveur Samba est un logiciel qui permet de partager des fichiers, des imprimantes et d'autres ressources entre des systèmes d'exploitation différents sur un réseau. Samba est un logiciel open source qui implémente le protocole SMB/CIFS (Server Message Block / Common Internet File System), utilisé principalement pour la communication entre les ordinateurs Windows, mais qui est également compatible avec d'autres systèmes d'exploitation tels que Linux, macOS et d'autres

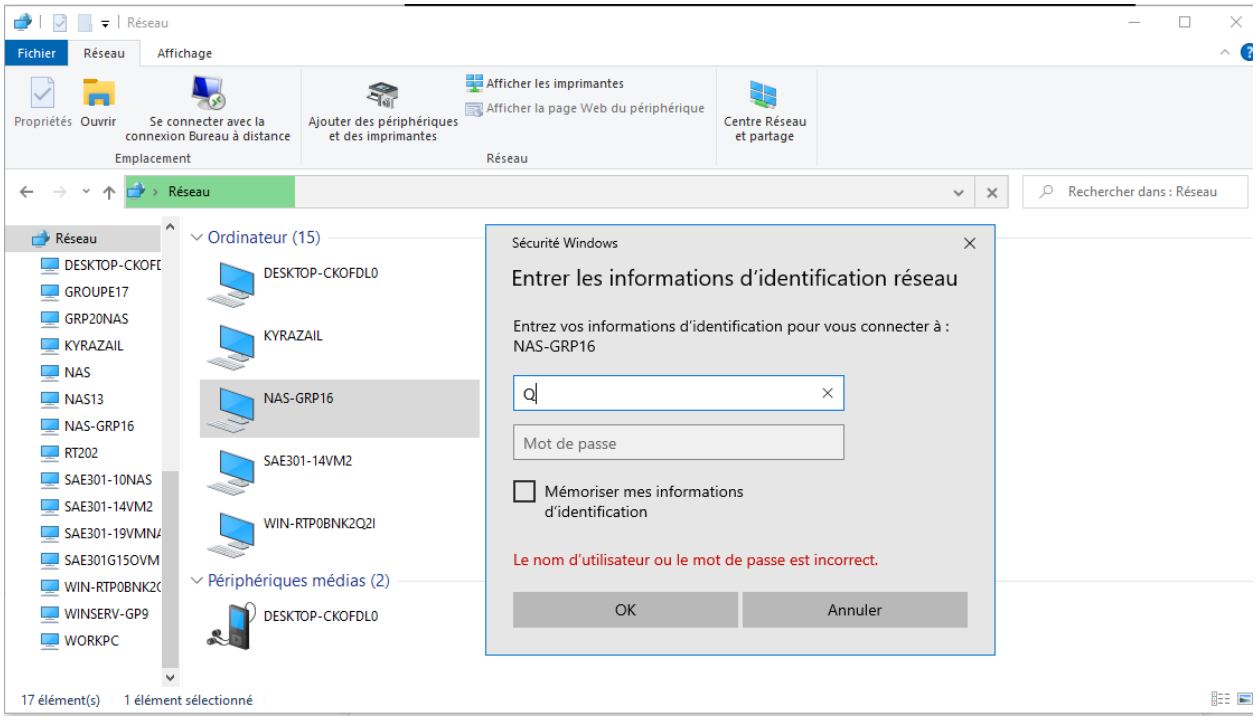
Pour configurer un serveur NAS, suivez ces étapes : Tout d'abord, montez la partition en accédant à "Stockage" et en sélectionnant "Système de fichiers". Vous devriez voir la partition montée, indiquée par "Oui" dans la colonne "Monté" et la taille de la partition dans la colonne "Capacité". Ensuite, créez un utilisateur en accédant à "Gestion des droits d'accès", puis "Utilisateur", et cliquez sur "Ajouter". Par défaut, cet utilisateur sera placé dans le groupe "users". Si vous souhaitez lui attribuer des droits d'administrateur, accédez à l'onglet "Groupes" et cochez "adm" et "root".

Créez ensuite deux répertoires de partage dans la partition du disque en allant à "Gestion des droits d'accès", puis "Dossiers partagés", et cliquez sur "Ajouter". Indiquez un nom pour le dossier, tel que "photos", sélectionnez la partition utilisée, le chemin d'accès (utilisez le nom du dossier pour plus de simplicité) et définissez les permissions.

Enfin, activez et configurez le service SaMBa pour permettre la visibilité du disque depuis un PC client connecté au réseau. Allez dans "Services", puis "SMB/CIFS", et cochez la case "activer". Vous pouvez également définir un nom de description pour la machine visible sur le réseau. Assurez-vous d'ajouter le dossier comme partage SaMBa en allant dans "Services", puis "SMB/CIFS", "Partages", et cliquez sur "Ajouter". Indiquez le nom visible sur le réseau, le dossier partagé, et configurez les options de sécurité, telles que la nécessité ou non d'un mot de passe.



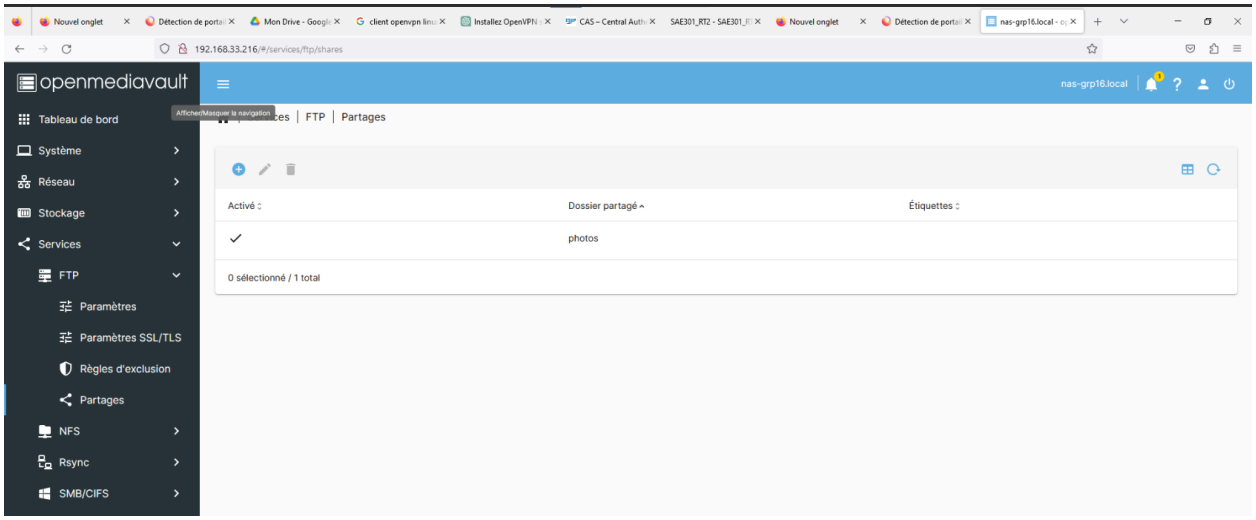
En suivant ces étapes, vous configurez avec succès votre serveur NAS pour le partage de fichiers sur le réseau, permettant à d'autres ordinateurs de voir et d'accéder aux données stockées. Assurez-vous d'appliquer les changements à chaque étape pour les rendre effectifs.



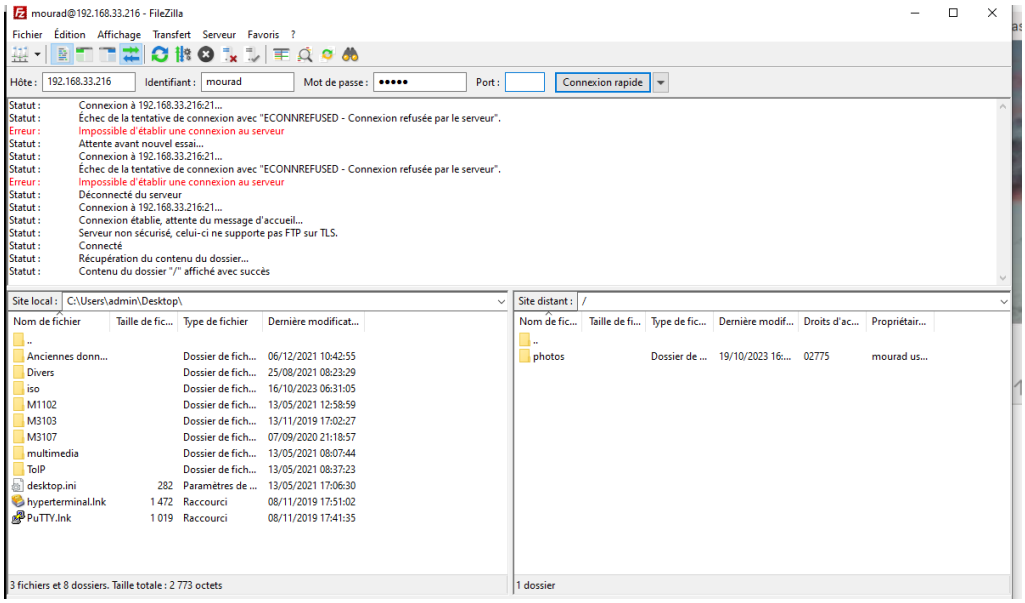
Partage Ftp:

Un partage FTP (File Transfer Protocol) est une méthode de partage de fichiers et de transfert de données sur un réseau, en utilisant le protocole FTP. FTP est un protocole de communication standard qui permet à un utilisateur d'accéder à des fichiers et des dossiers situés sur un serveur distant. Le partage FTP est couramment utilisé pour permettre à des utilisateurs d'accéder et de télécharger des fichiers à partir d'un serveur FTP centralisé.

Pour ajouter la fonctionnalité du partage ftp sur openmediavault il fallait d'abord ajouter un plugin ftp puis suivre les manipulations qui sont approximativement les mêmes que pour créer un partage samba.



En suivant ces étapes, vous configurez avec succès votre serveur FTP pour permettre le transfert de fichiers sur le réseau, autorisant ainsi d'autres ordinateurs à accéder aux données stockées. Assurez-vous d'appliquer les modifications à chaque étape pour qu'elles prennent effet. Une fois ces étapes terminées, votre serveur FTP sera prêt à gérer les transferts de fichiers entre les utilisateurs et les systèmes, fournissant un moyen efficace de partager et de gérer des données sur votre réseau.



Ce projet nous a permis de plonger dans un éventail de technologies essentielles pour la diffusion de contenu vidéo et le partage de fichiers au sein d'un environnement informatique. De la configuration d'un serveur vidéo adaptatif à l'exploration de protocoles de sécurité tels que WireGuard pour les VPN, en passant par la mise en place d'un serveur NAS polyvalent, nous avons acquis une expérience précieuse dans la gestion de données et de ressources au sein d'un réseau. Ces compétences sont applicables dans une variété de contextes, de la diffusion de contenu en streaming à la gestion de données au sein d'entreprises. En résumé, ce projet nous a offert une perspective approfondie sur les technologies et les compétences nécessaires pour tirer parti des avantages de la diffusion de contenu multimédia et de la gestion de fichiers au sein d'un environnement informatique moderne.