UNIVERSITÉ HASSAN 1ER :
FACULTÉ DES SCIENCES ET TECHNIQUES
SETTAT

Projet de Fin de Module
Licence Science et Techniques Génie Informatique

Module : Administration système et Sécurité réseau

Réalisé par:
Harfi Aya
Hamassi Chorouk
Ibnolkabir Jamaa
Glissi Salma

Encadré par:
Pr. soukaina Mihi

Année Universitaire:**2024/202**

- **Sommaire :**

2.3 FTPS (File Transfer Protocol Secure)

- Définition
- Fonctionnement (FTP explicite / FTP implicite)
- Avantages et inconvénients
- Ports utilisés

3. Mise en Place d'un Serveur FTPS avec vsFTPd

3.1 Objectifs

3.2 Prérequis

3.3 Étapes de Configuration

- Mise à jour des machines
- Installation de vsFTPd
- Génération du certificat SSL/TLS
- Configuration du fichier vsftpd.conf
- Ouverture des ports dans le pare-feu
- Redémarrage du service
- Test de connexion depuis un client Ubuntu
- Test de connexion depuis Windows (FileZilla)
- Transfert de fichiers

4. Comparaison entre SSHFS, SCP et FTPS

- Critères de comparaison (sécurité, vitesse, facilité d'utilisation, cas d'usage)
- Tableau comparatif

5. Conclusion

- Résumé des points importants
- Avantages de chaque solution

1. ***Introduction:***

- *Présentation générale du projet:*

Dans le domaine des systèmes et réseaux informatiques, le transfert de fichiers entre différentes machines est une opération courante et indispensable. Que ce soit dans un environnement professionnel, académique ou personnel, la gestion des échanges de données doit être réalisée de manière fiable et sécurisée.

Ce projet s'inscrit dans cette logique et a pour objectif d'étudier et de mettre en œuvre des solutions de transfert de fichiers sécurisés. Nous allons nous intéresser principalement à trois technologies largement utilisées dans le monde de l'administration des systèmes : SSHFS, SCP et FTPS.

Ces protocoles permettent d'assurer des échanges de fichiers en garantissant la confidentialité, l'intégrité et la sécurité des données circulant sur le réseau.

- *Importance des transferts de fichiers sécurisés:*

La sécurité des échanges de fichiers est devenue un enjeu majeur dans un contexte où les cyberattaques, les vols de données et les intrusions malveillantes sont de plus en plus fréquents.

Lorsqu'un fichier transite sur un réseau non sécurisé, il est exposé à des risques importants:

- Interception de données sensibles (identifiants, mots de passe, informations confidentielles).
- Modification ou falsification des fichiers transférés.
- Vol ou destruction des données.

Les entreprises, les organisations et les administrations doivent donc mettre en place des solutions fiables pour protéger les transferts de fichiers, en conformité avec les exigences de sécurité et les normes internationales (ex : RGPD, ISO 27001).

- *Technologies étudiées : SSHFS, SCP, FTPS:*

Dans le cadre de ce projet, trois technologies seront étudiées et comparées :

- SSHFS (SSH File System) : Un système de fichiers permettant de monter un répertoire distant en local via une connexion sécurisée SSH.

- SCP (Secure Copy Protocol) : Un protocole simple et rapide pour transférer des fichiers en toute sécurité à l'aide de SSH.

- FTPS (File Transfer Protocol Secure) : Une version sécurisée du protocole FTP qui utilise le chiffrement SSL/TLS pour protéger les transferts de fichiers.

Chacune de ces solutions présente des caractéristiques techniques spécifiques, des avantages et des limites, que nous allons explorer en détail au cours de ce projet.

2. *Présentation des Protocoles:*

2.1 SSHFS:

1. Contexte Général :

➢ **SSHFS(SSH Filesystem):**

Est un outil permettant d'utiliser le ssh protocole ssh comme un système de fichiers et ainsi monter un répertoire distant à travers le protocole ssh ssh .

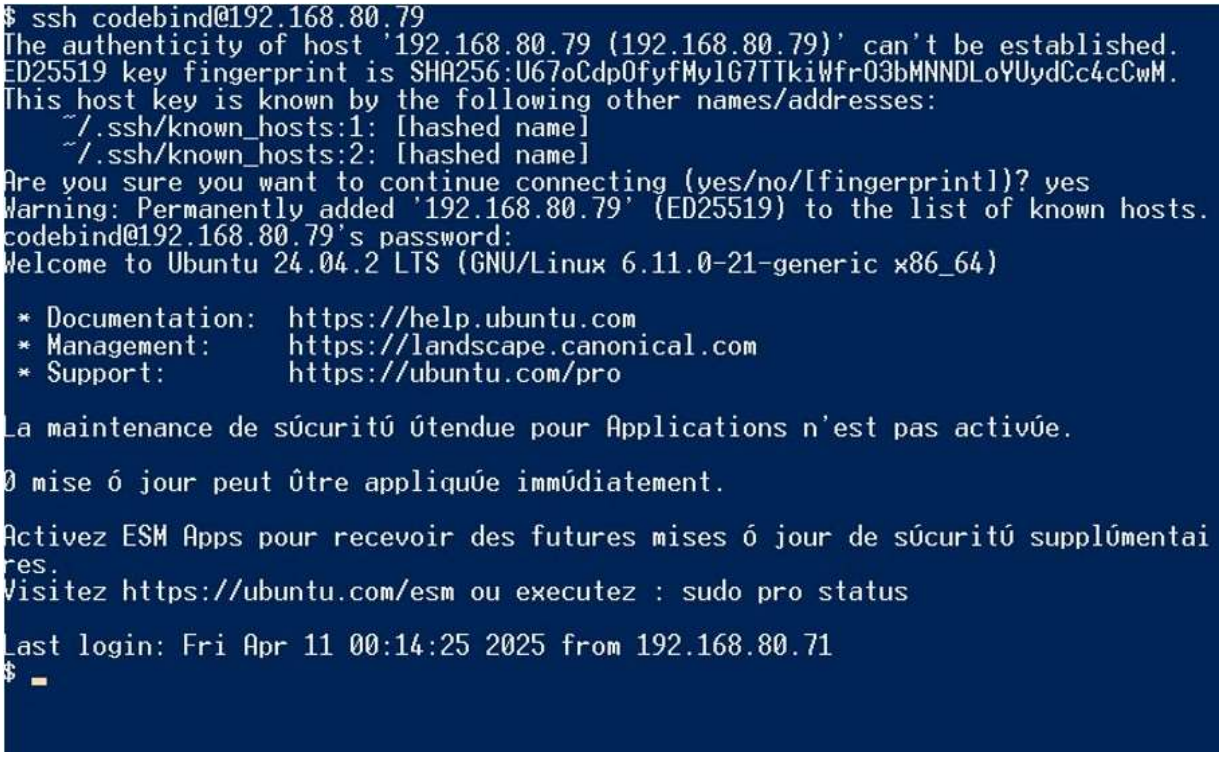
Il s'appuie sur FUSE (Filesystem in Userspace) pour fonctionner, ce qui permet à un utilisateur normal (sans root) d'accéder à un répertoire distant comme s'il était local

2.Installation:

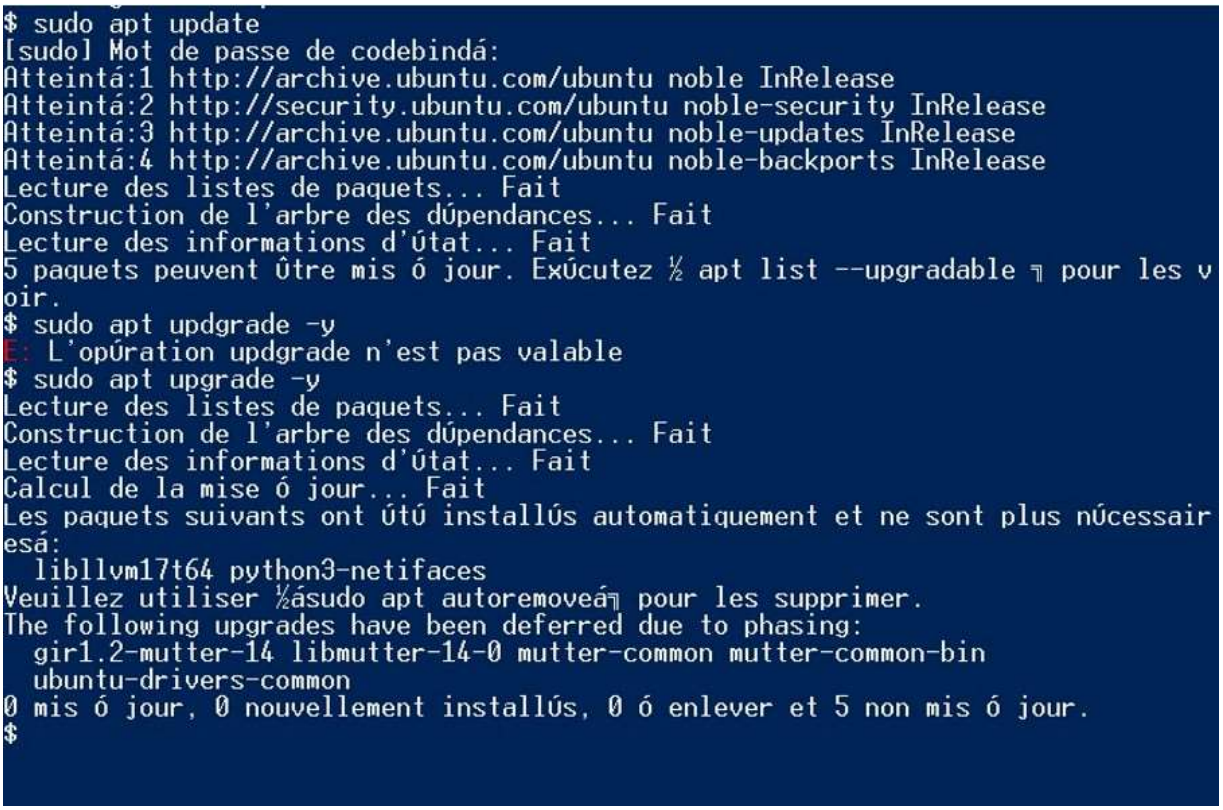
➔ **Installation côté serveur:**

Il suffit d'avoir un serveur ssh fonctionnel:

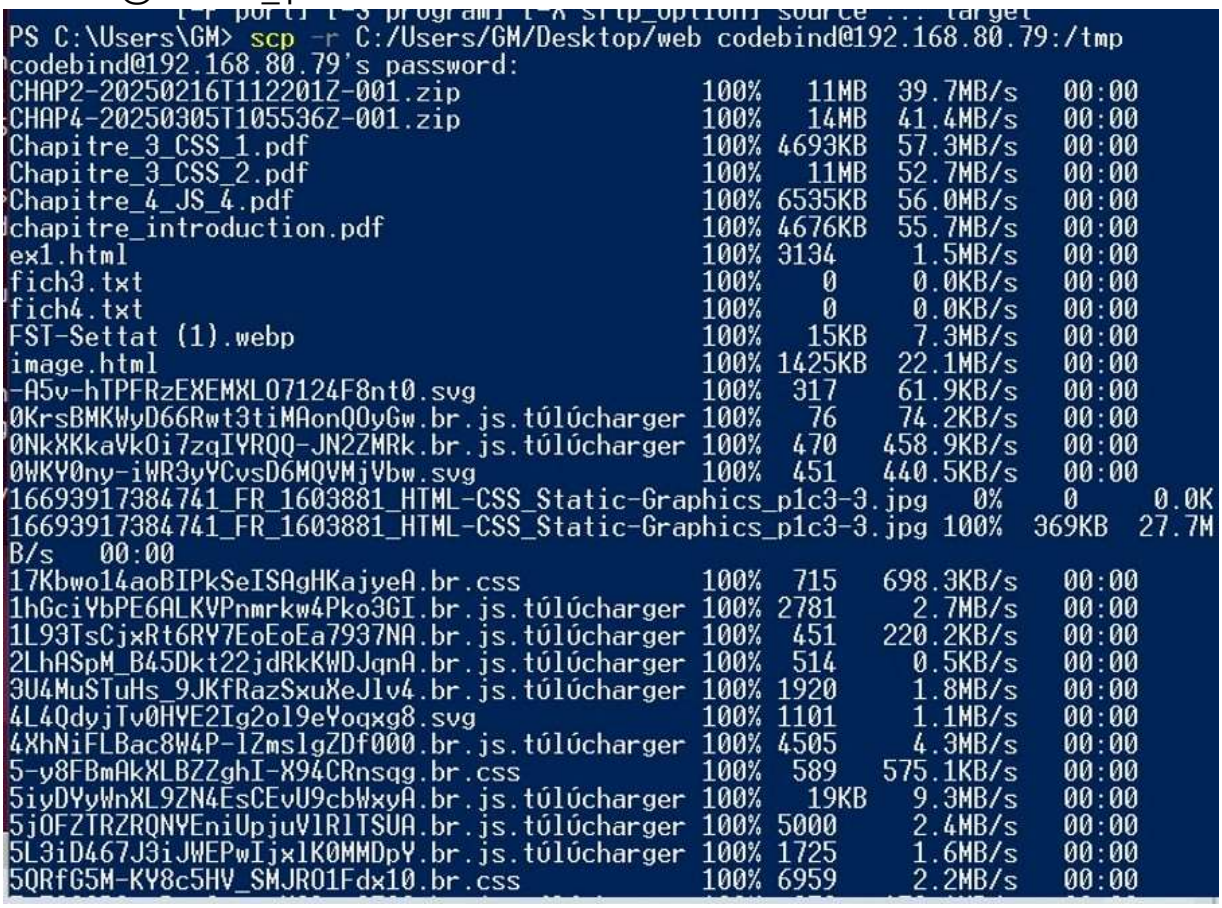
1. Connexion SSH ssh utilisateur@adresse_ip_du_serveur:



2.Mise à jour du system sudo apt update && sudo apt upgrade -y:



3.Transfert des fichiers scp -r /chemin/vers/ton/projet utilisateur@adresse_ip:/chemin/cible:

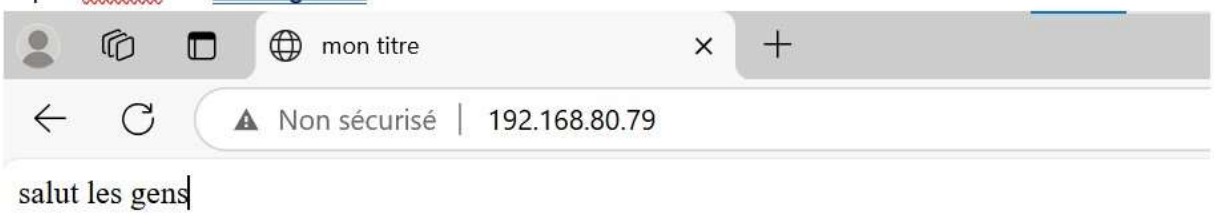


4.Configuration Placer les fichiers dans le dossier web (souvent /var/www/html) Donner les bons droits : sudo chown -R www-data:www-data /var/www/html sudo chmod -R 755 /var/www/html

5.Redémarrage du serveur web sudo systemctl restart apache2

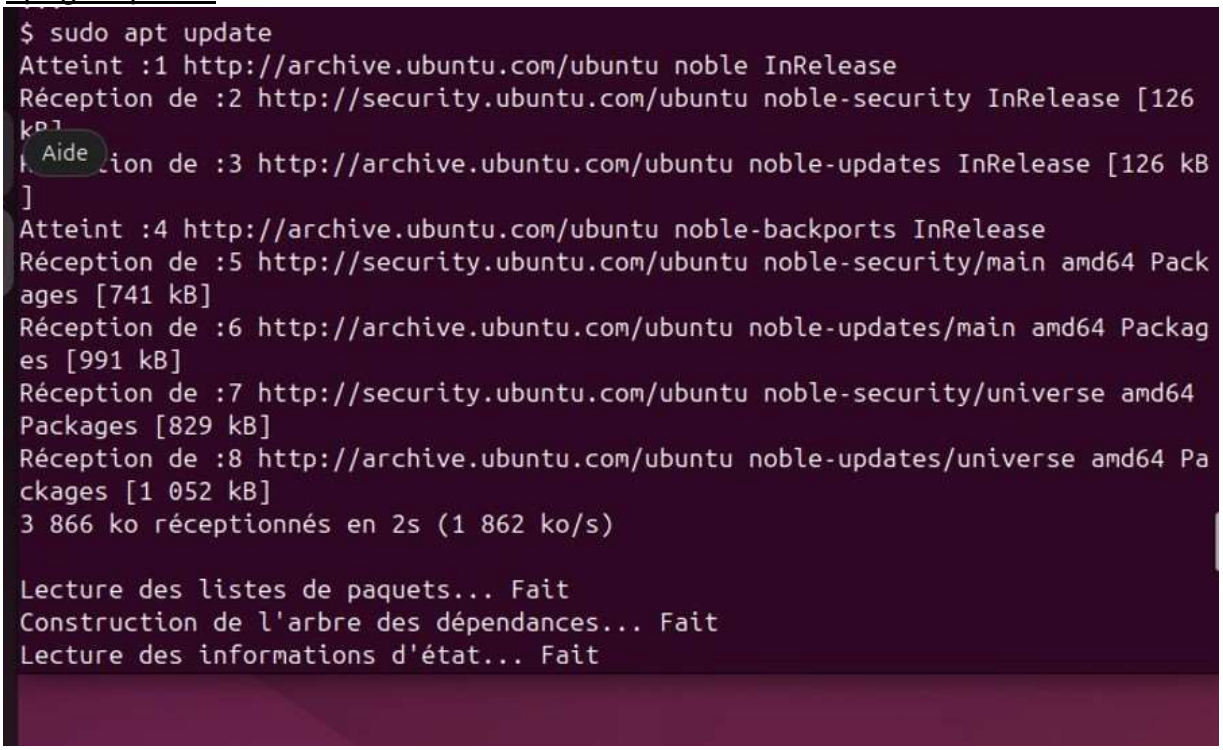
```
PS C:\Users\GM> ssh codebind@192.168.80.79
codebind@192.168.80.79's password:
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.11.0-21-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

La maintenance de sécurité étendue pour Applications n'est pas activée.

0 mise à jour peut être appliquée immédiatement.

Activez ESM Apps pour recevoir des futures mises à jour de sécurité supplémentai
res.
Visitez https://ubuntu.com/esm ou executez : sudo pro status

Last login: Fri Apr 11 01:22:06 2025 from 192.168.80.79
$ sudo cp -r /tmp/web/* /var/ww/html/
[sudo] Mot de passe de codebindá:
cp: cible '/var/ww/html/': Aucun fichier ou dossier de ce nom
$ sudo cp -r /tmp/web/* /var/www/html/
$ sudo chown -R www-data:www-data /var/www/html
$ sudo chmod -R 755 /var/www/html
$
$ sudo systemctl restart apache2
```

Tape l'idrress sur la navigateur

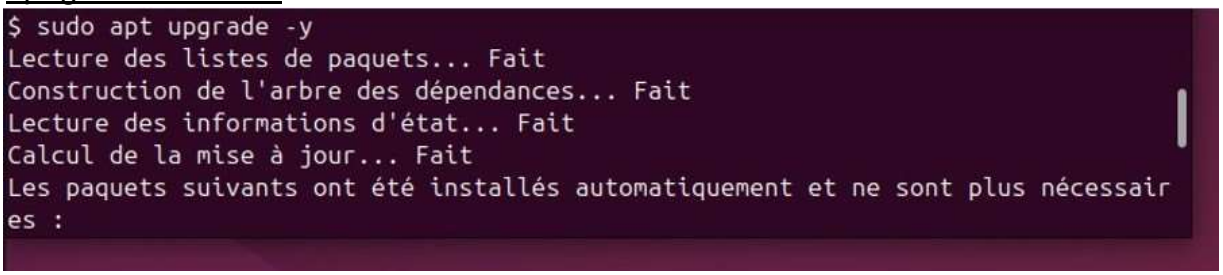


➔ **Installation côté client :**

• apt-get update:



• apt-get install sshfs:



3.Test d'installation:

• Sur le système client, créer un répertoire dans lequel va être monté le système de fichiers :

➔ **mkdir /home/utilisateur/Test**

•Monter un répertoire distant (ici ~/Public):

➔ **Sshfs utilisateur@ip_distante:/home/utilisateur /Public /home/utilisateur/Test**

•Enfin penser à démonter :

➔ **fusermount -u /home/utilisateur/Test**

4.Montage d'un dossier distant:

• Créer un point de montage local :

➔ **mkdir ~/mon_serveur**

➔ **Sshfs user@ip_du_serveur:/chemin/distant ~/**

•EXEMPLE:

➔ **sshfs alice@192.168.1.10:/var/www ~/mon_serveur**

5.Définition de FUSE(Filesystem in Userpace):

• **FUSE** est un système qui permet aux utilisateurs non root (comme toi ou moi) de créer et utiliser des systèmes de fichiers personnalisés, sans modifier le noyau (kernel) de l'OS.

- Commande d'installation:

➔ **sudo apt update**

➔ **sudo apt install fuse**

6.Fichiers de configuration:

- Fichier de config SSH utilisé par SSHFS:

➔ **~/.ssh/config (config utilisateur) ➔ nano ~/.ssh/config**

- Fichier de config FUSE (pour options avancées):

➔ **/etc/fuse.conf**

➔ **sudo nano /etc/fuse.conf**

- Config dans /etc/fstab (pour montage automatique):

Si tu veux monter SSHFS au démarrage, tu peux ajouter une ligne dans /etc/fstab : **fstab**

2.2 SCP (Secure Copy Protocol):

1. Contexte Général :

SCP(Secure Copy Protocol) : est l'outil classique pour copier en toute sécurité des fichiers de votre ordinateur local vers des serveurs distants, et inversement.

Le nom du protocole SCP est dérivé de deux technologies sous-jacentes :

- Protocole SSH (Secure Shell) : qui permet l'accès chiffré aux systèmes distants ;
- l'outil RCP (remote copy), qui copie les fichiers de manière non sécurisée, c'est-à-dire sans chiffrement, en réseau.

PARTIE 1 : Vérification de l'installation de SCP

➔ **Vérifier si scp est installé**: Scp -v

➔ **Vérifier que le service SSH tourne sur la machine:** *Sudo systemctl status ssh*

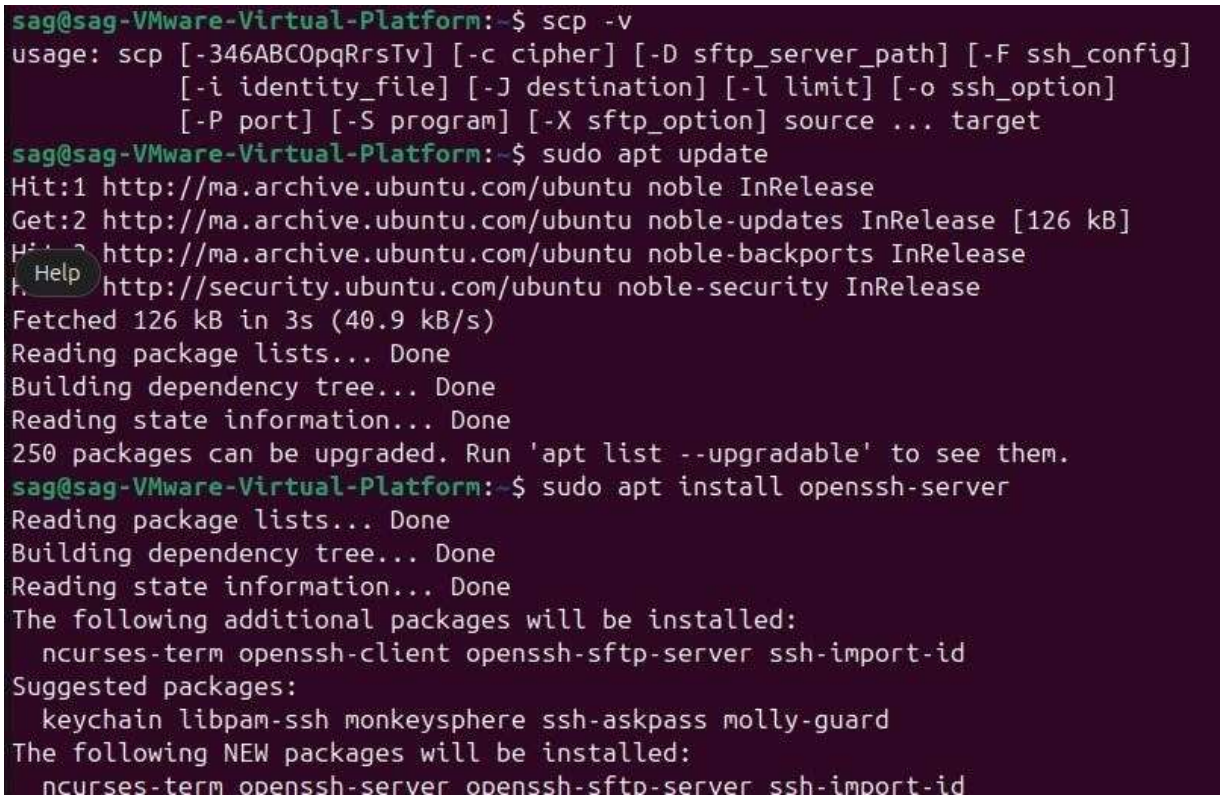
➔ **Au cas ssh not found :***Sudo apt update*
Sudo apt install openssh-server

➔ **Démarrer:** Sudo systemctl start ssh
Sudo systemctl enable ssh

➔ **Vérifier le status:** *Sudo systemctl status ssh*

➔ **Récupérer adresse ip de la machine:** *ip a*

❖ ***Capture d'écran:***



```
Created symlink /etc/systemd/system/ssh.service.requires/ssh.socket → /usr/lib/s
ystemd/system/ssh.socket.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for ufw (0.36.2-6) ...
sag@sag-VMware-Virtual-Platform:~$ sudo systemctl start ssh
sag@sag-VMware-Virtual-Platform:~$ sudo systemstl enable ssh
sudo: systemstl: command not found
sag@sag-VMware-Virtual-Platform:~$ sudo systemctl enable ssh
Synchronizing state of ssh.service with SysV service script with /usr/lib/system
d/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh
Created symlink /etc/systemd/system/sshd.service → /usr/lib/systemd/system/ssh.s
ervice.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /usr/l
ib/systemd/system/ssh.service.
sag@sag-VMware-Virtual-Platform:~$ sudo systemctl status ssh
○ ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enab>
     Active: inactive (dead)
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 4213 (sshd)
      Tasks: 1 (limit: 4551)
```

```
sag@sag-VMware-Virtual-Platform:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP g
roup default qlen 1000
    link/ether 00:0c:29:97:ea:3e brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.163.130/24 brd 192.168.163.255 scope global dynamic noprefixrou
te ens33
       valid_lft 1356sec preferred_lft 1356sec
    inet6 fe80::20c:29ff:fe97:ea3e/64 scope link
       valid_lft forever preferred_lft forever
```

PARTIE 2 : COPIER UN FICHIER VERS UNE AUTRE MACHINE DISTANTE

➔ **Dans client_SCP :**

1-créer fichier :

echo ‘’Fichier transféré avec scp’’ > fichier_scp.txt

2-transférer ce fichier vers le serveur :

scp fichier_scp.txt userHYPERLINK "mailto:salma@192.168.163.131:/home/"@HYPERLINK "mailto:salma@192.168.163.131:/home/"ip_srvHYPERLINK "mailto:salma@192.168.163.131:/home/":/home/user/ ➔

Dans srv_scp :

1-vérifier fichier est bien arrivé :

ls /home/user/

❖ ***Capture d’écran:***

```
sag@sag-VMware-Virtual-Platform:~$ echo "Fichier transfere avec SCP" > fichier_s
cp.txt
```

```
sag@sag-VMware-Virtual-Platform:~$ scp fichier_scp.txt salma@192.168.163.131:/ho
me/
```

```
sag@sag-VMware-Virtual-Platform:~$ ls
Desktop     dossier_scp      Music      share.txt  Videos
Documents   Downloads        Pictures   snap       vsftpd.conf.orig
dos_scp     fichier_scp.txt  Public     Templates
```

PARTIE 3 : Copier un fichier depuis une machine distante

➔ **Machine locale :**

scp user@ip_distante:/home/user/fichier_scp.txt ./copie_locale.txt

➔ **Vérifier que tu as bien une copie locale :** Cat copie_locale.txt

❖ ***Capture d’écran:***

```
sag@sag-VMware-Virtual-Platform:~$ scp salma@192.168.163.131:/home/fichier_scp.t
xt ./copie_locale.txt
```

PARTIE 4 : Copier un dossier entier avec l'option -r

➔ **créer un dossier :** mkdir dossier_scp echo « doc1 » > dossier_scp/1.txt echo « doc2 » > dossier_scp/2.txt

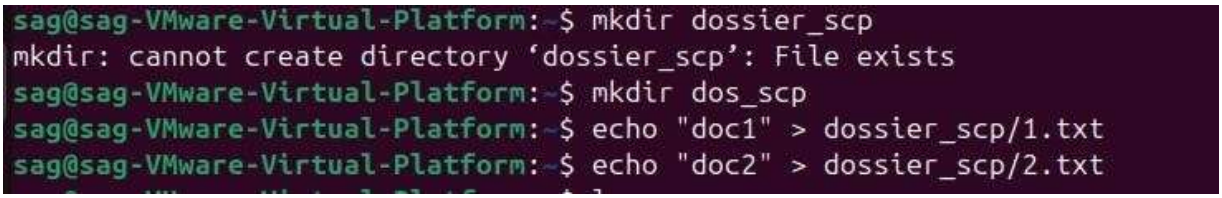
➔ **copier le dossier sur vers la machine distante:**

scp -r dossier_scp user@ip_distante:/home/user/

➔ **Vérifier sur la machine distante :**

Ls /home/user/dossier_scp

❖ ***Capture d'écran:***



Partie 5: Option utiles de SCP

Option	Description
-r	Copie récursif pour les dossiers
-p	Spécifie le port SSH
-c	Compression des données pendant le transfert
-i	Utiliser une clé privée SSH pour l'authentification

Partie 6: Transfert via un port personnalisé

Scp -p 2222 fichier.txt user@ip_srv :home/user/

PARTIE 7: Utiliser une clé SSH pour se connecter(sans mot de passe)

➔ **Générer une paire de clés :**

Ssh-keygen

➔ **Copier la clé publique vers la machine distante:**

ssh-copy-id user@ip_srv

➔ **Transférer un fichier dans mot de passe :**

Scp fichier.txt user@ip_srv/home/user/

2.3 FTPS (File Transfer Protocol Secure) :

1. Contexte Général :

FTPS (FTP Secure) est une extension du protocole FTP (File Transfer Protocol) qui ajoute une couche de sécurité grâce au protocole **TLS/SSL**. Contrairement à FTP simple, qui transmet les données (y compris les mots de passe) en clair, FTPS chiffre les connexions, assurant ainsi la **confidentialité** et **l'intégrité** des données transférées. Il est particulièrement utile dans les environnements professionnels où la sécurité des transferts de fichiers est cruciale.

2.L'intérêt d'utiliser FTPS est de :

- Protéger les identifiants et les données sensibles en transit
- Prévenir les attaques de type "man-in-the-middle"
- Répondre aux exigences de conformité en entreprise (ex. : RGPD, ISO)

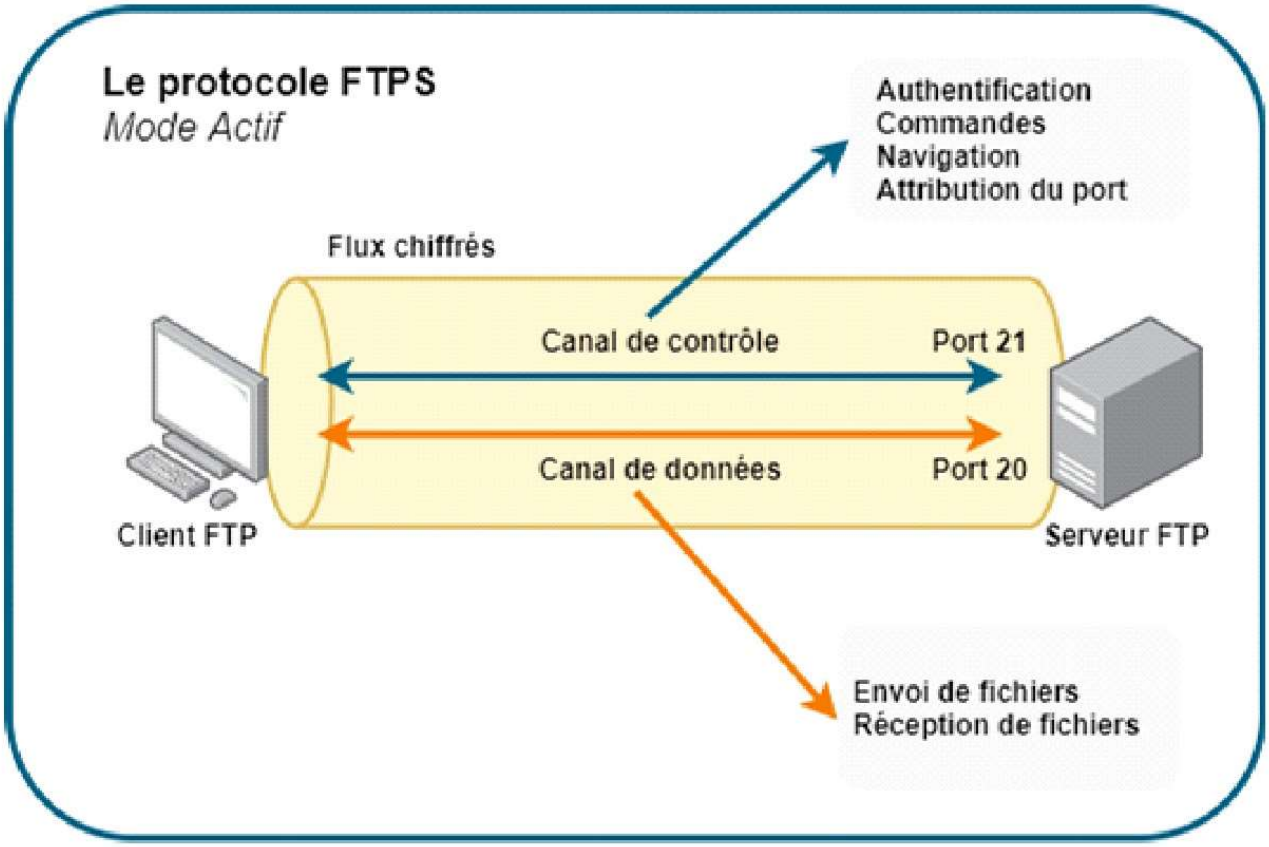
➢ **Modes de fonctionnement :**

- **FTP explicite :**

La connexion s'établit sur le port 21, et le client FTP négocie avec le serveur pour chiffrer la connexion. Commandes utilisées : `AUTH TLS` ou `AUTH SSL` et `PROT P`.

- **FTP implicite :**

Le client commence directement la connexion avec le chiffrement SSL et utilise le port 990.



➢ **Ports utilisés par FTPS :**

- Port 990 (canal de contrôle).
- Port 989 (canal de données).
- Le port 21 peut également être utilisé.

2. Objectifs

➔ **Objectifs :**

- Mettre en place un serveur FTP sécurisé (FTPS) avec vsFTPd.
- Configurer une communication sécurisée via SSL/TLS.

- Tester la connexion sécurisée depuis une machine cliente Ubuntu et une machine Windows.

3. Prérequis

➔ **Environnement :**

- Utilisation de Ubuntu pour créer deux machines virtuelles Ubuntu 24.04 :
  - **srv_ubuntu (serveur)** • **client_ubuntu (client)**

- Les deux VM configurées en **mode réseau bridgé** pour permettre la communication entre elles et avec la machine hôte

- Accès internet

➔ **Logiciels :**

- Client FTP graphique (comme FileZilla sur Windows)

4. Étapes de configuration de FTPS avec vsFTPd:

Étape 1 **: Mise à jour les deux machines Serveur et client**

- sudo apt update
- sudo apt upgrade

Étape 2 : **Installation de vsFTPd sur serveur:**

➔ ***Vérifier que vsFTPd n'est pas installé :*** dpkg -l | grep vsFTPd

sudo apt install vsftpd -y

Étape 3 **: Génération du certificat SSL/TLS et du clé privée:**

Créer le dossier ssl/vsftpd pour stocker le certificat TLS/SSL

```
# mkdir -p /etc/ssl/vsftpd
```

Installer openssh :programme qui génère les clè privés et certificat

```
sudo apt install vsftpd openssl -y
```

Génerer le certificat TLS/SSL auto-signé et la clé privée avec la commande suivante :

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
  -keyout /etc/ssl/vsftpd/vsftpd.key \
  -out /etc/ssl/vsftpd/vsftpd.crt
sudo chmod 600 /etc/ssl/vsftpd/vsftpd.*
```

➔ **Explication des options :**

- `req` : Gestion des demandes de signature de certificat X.509 (CSR).
- `x509` : Norme pour générer des certificats TLS.
- `days` : Durée de validité du certificat (365 jours).
- `newkey` : Algorithme de cryptage asymétrique (RSA:2048).
- `keyout` : Fichier de stockage de la clé privée.
- `out` : Fichier de stockage du certificat.

Étape 4 **: Sauvegarde et édition du fichier de configuration:**

sudo cp /etc/vsftpd.conf /etc/vsftpd.conf.bak

sudo nano /etc/vsftpd.conf

Étape 5 **: Configuration du fichier vsftpd.conf**

 # SSL/TLS ssl_enable=YES
ssl_tlsv1_2=YES
ssl_sslv2=NO
ssl_sslv3=NO
allow_anon_ssl=NO
force_local_logins_ssl=YE
S
force_local_data_ssl=YES

 # Certificats

rsa_cert_file=/etc/ssl/vsftpd/vsftpd.crt
rsa_private_key_file=/etc/ssl/vsftpd/vsftpd.key

 # Sécurité
require_ssl_reuse=NO
ssl_ciphers=HIGH
pasv_min_port=40000
pasv_max_port=50000
chroot_local_user=YES
allow_writeable_chroot=YES

➔ Cette configuration du serveur FTP (vsftpd) force toutes les connexions (anonymes et locales) à utiliser **SSL en mode implicite** via le port **990**.

```
# To force anonymous users to use SSL
force_anon_data_ssl=YES
force_anon_logins_ssl=YES

# To force local users to use SSL
force_local_data_ssl=YES
force_local_logins_ssl=YES
implicit_ssl=YES
listen_port=990
```

Ajoutez les paramètres de configuration suivants pour activer SSL, puis sélectionnez la version de SSL et TLS à utiliser, à la fin du fichier.

```
ssl_enable=YES
ssl_tlsv1_2=YES
ssl_sslv2=NO
ssl_sslv3=NO
```

Ensuite, ajoutez les options rsa_cert_file et rsa_private_key_file pour spécifier respectivement l'emplacement du certificat SSL et du fichier de clé privée.

```
rsa_cert_file=/etc/ssl/vsftpd/vsftpd.pem
rsa_private_key_file=/etc/ssl/vsftpd/vsftpd.pem
```

Ensuite, ajoutez ces options pour désactiver toute réutilisation des connexions de données SSL et définissez les chiffrements SSL ÉLEVÉS pour autoriser les connexions SSL cryptées.

```
require_ssl_reuse=NO
ssl_ciphers=HIGH
```

Vous devez également spécifier la plage de ports (port min et max) des ports passifs à utiliser par vsftpd pour les connexions sécurisées, en utilisant respectivement les paramètres pasv_min_port et pasv_max_port. En outre, vous pouvez éventuellement activer le débogage SSL à des fins de dépannage, à l'aide de l'option debug_ssl.

```
pasv_min_port=40000
pasv_max_port=50000
debug_ssl=YES
```

Une autre tâche critique à effectuer avant de pouvoir accéder en toute sécurité au serveur FTP consiste à ouvrir les ports 990 et 40000-50000 dans le pare-feu du système. Cela permettra les connexions TLS au service vsftpd et ouvrira la plage de ports passifs définis dans le fichier de configuration VSFTPD respectivement, comme suit.

sudo ufw allow 990/tcp sudo
ufw allow 40000:50000/tcp
sudo ufw reload

Étape 6 : Redémarrer le service

sudo systemctl restart vsftpd
sudo systemctl enable vsftpd

Vérifier que vsftpd est activé et démarré : sudo systemctl status vsftpd --no-pager -l

Étape 7 : Créer un fichier à partager

Sur le dossier personnel de l'utilisateur normal (créer un utilisateur normal sinon), créer un fichier share.txt touch share.txt sur /home/user

touch /home/user/share.txt

5. Connexion depuis la VM client

- **Installation du client FTP**

sudo apt install lftp

Connexion FTPS depuis le terminal

lftp -u testftp,password ftps://IP_SERVEUR

6. Connexion depuis une machine Windows (FileZilla):

- Hôte : ftps://IP_SERVEUR
- Port : 990
- Protocole : FTP – SSL/TLS implicite
- Type de connexion : **Normal**
- Identifiants : username / password

7. Transfert de fichier (tests):

➔ Sur client :

put share.txt
get share.txt

➔ Sur serveur :

Vérifier que le fichier est bien reçu dans /home/user

❖ Comparaison entre SSHFS, SCP et FTPS:

- **Critères de comparaison**

Pour évaluer et comparer les trois technologies étudiées (SSHFS, SCP et FTPS), nous nous basons sur plusieurs critères essentiels qui permettent de déterminer la solution la plus adaptée selon les besoins :

1. Sécurité

Les trois technologies offrent un haut niveau de sécurité basé sur des protocoles de chiffrement :

- SSHFS et SCP utilisent le protocole SSH (Secure Shell) pour assurer la confidentialité et l'intégrité des données.
- FTPS utilise SSL/TLS pour chiffrer les connexions et protéger les transferts.

2. Vitesse

- SCP est souvent le plus rapide car il effectue un simple transfert de fichiers sans montée de système de fichiers distants.
- SSHFS est un peu moins rapide car il monte un système de fichiers distants et dépend des performances du réseau.
- FTPS a des performances correctes mais peut être ralenti par l'établissement des connexions TLS et les vérifications des certificats.

3. Facilité d'utilisation

- SCP est très simple à utiliser via une seule commande en ligne.
- SSHFS nécessite une installation mais offre un accès direct et pratique aux fichiers distants.
- FTPS demande une configuration serveur spécifique et l'ouverture de plusieurs ports, mais il est très convivial avec des interfaces graphiques (ex : FileZilla).

4. Cas d'usage :

Technologie	Cas d’usage recommandé
SSHFS	Montage de répertoires distants pour une utilisation régulière comme un disque local.
SCP	Transfert rapide de fichiers ponctuels entre deux machines.
FTPS	Environnements professionnels nécessitant un transfert sécurisé avec interface graphique et respect des normes de sécurité

- Tableau comparatif :

Critères	SSHFS	SCP	FTPS
Sécurité	Chiffrement via SSH	Chiffrement via SSH	Chiffrement via SSL/TLS
Vitesse de transfert	Moyenne(dépend du réseau)	Rapide	Moyenne à bonne
Facilité d'utilisation	Nécessite un montage initial	Commande simple en ligne	Configuration serveur nécessaire/Client graphiques
Reprise transfert	Oui(accès direct aux fichiers)	Non	Oui(dépend du client utilisé)
Ports utilisés	Port SSH(22)	Port SSH(22)	Ports 21,990,989+ports passifs
Cas d'usage idéal	Travail quotidien sur fichiers distants	Copie rapide de fichiers ponctuels	Transfert sécurisé en entreprise

5. **Conclusion:**

➔ **Résumé des points importants:**

Dans ce projet, nous avons étudié trois technologies essentielles pour le transfert sécurisé de fichiers : SSHFS, SCP et FTPS.

Chaque solution répond à des besoins spécifiques en fonction de plusieurs critères tels que la sécurité, la vitesse, la facilité d'utilisation et le contexte d'utilisation.

Technologie	Fonction	Sécurité	Utilisation recommandé
SSHFS	Monter un répertoire distant	Chiffrement via SSH	Travail quotidien sur des fichiers distants comme en local
SCP	Transfert rapide de fichiers	Chiffrement via SSH	Transfert rapide et ponctuel de fichiers entre machines
FTPS	Transfert sécurisé basé sur FTP	Chiffrement via SSL/TLS	Environnements professionnels avec des exigences de sécurité élevées

➔ **Avantages de chaque solution**

➢ **SSHFS:**

- Permet d'accéder à un répertoire distant comme s'il était local.
- Utilisation simple après installation.
- Connexion sécurisée par SSH.
- Idéal pour un travail en temps réel sur des fichiers distants.

➢ **SCP:**

- Outil très rapide pour le transfert ponctuel de fichiers.

- Très facile à utiliser (commande simple).
- Sécurisé via SSH.
- Ne nécessite pas de configuration spécifique côté serveur (SSH suffisant).

➢ **FTPS :**

- Assure une haute sécurité grâce au protocole SSL/TLS.
- Conforme aux normes de sécurité professionnelles (RGPD, ISO).
- Compatible avec des interfaces graphiques conviviales (FileZilla).
- Recommandé pour les transferts sécurisés en entreprise.

En résumé, le choix entre SSHFS, SCP et FTPS dépend principalement du contexte d'utilisation et des besoins de l'utilisateur ou de l'entreprise :

- SSHFS est parfait pour un accès distant continu aux fichiers.
- SCP est idéal pour les transferts rapides et simples.
- FTPS est recommandé pour des environnements professionnels nécessitant une sécurité renforcée et une conformité aux normes.