

# Ubuntu - Trucs et Actuces

- [Installation de Firefox via PPA sur Ubuntu 22.04 \(et versions supérieures\) et blocage de l'installation via Snap](#)
- [La structure des répertoires sous Ubuntu](#)
- [Monter un point Samba sur Ubuntu : un guide rapide](#)
- [Crontab sous Ubuntu : Planifiez vos tâches automatiquement](#)
- [La Gestion des droits des fichiers et répertoires](#)
- [apt-cacher-ng : Un cache HTTP pour les paquets APT](#)
- [Ubuntu - Encrypter son Répertoire Home](#)
- [Installer Docker sous Ubuntu](#)
- [Installer Infomaniak kDrive sous Ubuntu](#)

# Installation de Firefox via PPA sur Ubuntu 22.04 (et versions supérieures) et blocage de l'installation via Snap

L'installation de Firefox sous Ubuntu 22.04 (Jammy Jellyfish) ou versions supérieures se fait généralement via Snap, une méthode qui ne plaît pas à tout le monde en raison de la nature encapsulée des applications Snap, leur consommation de ressources et leur lenteur au démarrage. Heureusement, il est possible d'installer Firefox via un PPA (Personal Package Archive) pour obtenir une version classique basée sur les paquets **.deb**. Voici comment procéder, ainsi que la manière de bloquer l'installation via Snap.

## Étapes pour installer Firefox via PPA

### 1. Supprimer la version Snap de Firefox (si installée)

Avant d'installer Firefox via PPA, il est essentiel de supprimer la version Snap pour éviter tout conflit. Ouvrez un terminal et exécutez les commandes suivantes :

```
sudo snap remove firefox
```

### 2. Ajouter le PPA pour Firefox

Mozilla propose un PPA officiel pour distribuer Firefox via les paquets **.deb**. Pour l'ajouter, exécutez :

```
sudo add-apt-repository ppa:mozillateam/ppa
sudo apt update
```

### 3. Configurer les priorités du PPA

Il est nécessaire de configurer votre système pour privilégier l'installation de Firefox à partir du PPA plutôt que de Snap. Cela se fait via un fichier de préférences dans le dossier `/etc/apt/preferences.d/`. Créez et éditez un nouveau fichier :

```
sudo nano /etc/apt/preferences.d/mozillateamppa
```

Ajoutez-y les lignes suivantes :

```
Package: *
Pin: release o=LP-PPA-mozillateam
Pin-Priority: 1001

Package: firefox*
Pin: release o=Ubuntu*
Pin-Priority: -1
```

Ces paramètres vont forcer le système à utiliser les paquets Firefox du PPA et à ignorer ceux provenant de Snap.

### 4. Installer Firefox

Après avoir configuré les priorités, installez Firefox via le PPA :

```
sudo apt install firefox
```

Cette commande installera Firefox en tant que paquet **.deb**, et il sera mis à jour à partir du PPA de Mozilla à chaque fois que vous mettez à jour votre système.

## Blocage de l'installation automatique de Firefox via Snap

Ubuntu peut parfois tenter de réinstaller des paquets via Snap lors des mises à jour. Pour empêcher cela, il est recommandé de bloquer le Snap de Firefox.

### 1. Bloquer l'installation automatique du Snap de Firefox

Utilisez la commande suivante pour désactiver définitivement le Snap de Firefox :

```
sudo apt-mark hold firefox
```

Cela empêchera toute mise à jour ou installation automatique du Snap de Firefox à l'avenir.

### 2. Désactiver totalement Snap (optionnel)

Si vous souhaitez complètement désactiver Snap sur votre système (pas seulement pour Firefox), vous pouvez procéder ainsi :

- Désinstallez Snapd :

```
sudo apt purge snapd
```

- Supprimez également les fichiers restants de Snap :

```
sudo rm -rf /var/cache/snapd/  
rm -rf ~/snap
```

## Conclusion

En suivant ces étapes, vous aurez installé Firefox via PPA sur Ubuntu 22.04 ou supérieur, et évité l'installation via Snap. Cela vous permettra de bénéficier d'une version plus traditionnelle de Firefox, avec potentiellement de meilleures performances sur certains systèmes et une plus grande souplesse de gestion.

# La structure des répertoires sous Ubuntu

La structure des répertoires sous **Ubuntu** (et plus généralement sous **Linux**) suit la norme **FHS** (File Hierarchy Standard). Cette organisation est conçue pour garantir la cohérence et la simplicité d'utilisation. Voici une explication des répertoires principaux que l'on trouve dans Ubuntu :

## 1. / (Root directory)

- Le répertoire racine, c'est la base de tout le système de fichiers. Tous les autres répertoires se trouvent en dessous de celui-ci.

## 2. /bin

- Contient les **binaries** ou fichiers exécutables essentiels nécessaires pour l'utilisation du système, disponibles pour tous les utilisateurs. Il inclut des commandes de base comme `ls`, `cp`, `mv`, etc.

## 3. /boot

- Contient les fichiers nécessaires au **démarrage du système**, comme le noyau Linux et les fichiers de configuration du chargeur de démarrage (GRUB).

## 4. /dev

- Contient des fichiers spéciaux représentant les **périphériques**. Dans Linux, presque tout est traité comme un fichier, y compris les périphériques matériels (disques, ports USB, etc.). Exemple : `/dev/sda` représente un disque dur.

## 5. /etc

- Contient les fichiers de **configuration** du système et des applications. Par exemple, les fichiers de configuration de réseau ou des services système.

## 6. /home

- Chaque utilisateur a son propre répertoire personnel dans **/home**, où il peut stocker ses fichiers et configurations. Par exemple, le répertoire de l'utilisateur "toto" sera `/home/toto`.

## 7. /lib et /lib64

- Contient les **bibliothèques partagées** nécessaires au fonctionnement des programmes et des fichiers exécutables qui se trouvent dans `/bin` et `/sbin`. Cela inclut des fichiers comme `libc.so.6`, utilisés par de nombreux programmes.

## 8. /media et /mnt

- /media** est utilisé pour le **montage automatique** des médias externes comme les clés USB ou les disques durs externes.
- /mnt** est utilisé pour monter temporairement des systèmes de fichiers par l'utilisateur (typiquement manuellement).

## 9. /opt

- Ce répertoire est réservé à l'installation de **logiciels supplémentaires** qui ne font pas partie du système de base, souvent des logiciels propriétaires ou commerciaux.

## 10. /proc

- Répertoire virtuel qui contient des informations sur le **système en cours d'exécution**, notamment sur les processus, les ressources système, les paramètres du noyau, etc. Par exemple, `/proc/cpuinfo` contient des informations sur le processeur.

## 11. /root

- Le répertoire personnel de l'utilisateur **root** (administrateur du système).

## 12. /run

- Contient des informations volatiles, comme des **fichiers temporaires** générés au démarrage ou par les services. Ces fichiers sont supprimés à chaque redémarrage.

## 13. /sbin

- Comme **/bin**, mais contient des **commandes système** généralement réservées à l'administrateur (root). Des commandes comme `shutdown` ou `fdisk` se trouvent ici.

## 14. /srv

- Ce répertoire contient les données spécifiques aux **services** fournis par le système, comme les serveurs web ou FTP. Par exemple, les fichiers des sites web peuvent être stockés dans `/srv/www`.

## 15. /sys

- Répertoire virtuel similaire à `/proc`, il expose des informations sur les **composants matériels** et l'état du système.

## 16. /tmp

- Contient des fichiers **temporaires** créés par les applications et le système. Ce répertoire est vidé à chaque redémarrage.

## 17. /usr

- Ce répertoire est souvent le plus grand du système. Il contient les fichiers utilisateur partagés comme les programmes et les bibliothèques qui ne sont pas essentiels au démarrage du système. Il est subdivisé en plusieurs sous-répertoires comme :
  - **/usr/bin** : contient la plupart des programmes utilisateurs.
  - **/usr/sbin** : contient les outils systèmes qui ne sont pas indispensables au démarrage.
  - **/usr/lib** : contient les bibliothèques nécessaires aux programmes dans `/usr/bin` et `/usr/sbin`.
  - **/usr/share** : contient des fichiers indépendants de l'architecture, comme de la documentation, des manuels, des polices, etc.

## 18. /var

- Contient des fichiers dont la taille peut **varier**, comme les fichiers journaux (`/var/log`), les fichiers de cache (`/var/cache`), ou les répertoires d'impression temporaires (`/var/spool`).

Cette structure est pensée pour être intuitive et efficace, permettant de gérer les droits, la sécurité, et l'organisation des fichiers sur le système.

# Monter un point Samba sur Ubuntu : un guide rapide

## Pourquoi monter un point Samba ?

Pour accéder facilement à des fichiers partagés sur un réseau local (comme un PC Windows).

## Les étapes clés :

### 1. Installer les outils nécessaires :

```
sudo apt install cifs-utils
```

### 2. Monter le partage :

```
sudo mount -t cifs //adresse_ip_du_serveur/partage /chemin/local -o user=ton_nom_utilisateur,password=ton_mot_de_passe
```

- Remplacer :
  - `//adresse_ip_du_serveur/partage` par l'adresse du partage Samba.
  - `/chemin/local` par le dossier où tu veux monter le partage.
  - `ton_nom_utilisateur` et `ton_mot_de_passe` par tes identifiants.

### 3. Monter automatiquement au démarrage (optionnel) : Éditer le fichier `/etc/fstab` et ajouter une ligne comme celle-ci :

```
//adresse_ip_du_serveur/partage /chemin/local cifs vers=yes,user=ton_nom_utilisateur,password=ton_mot_de_passe 0 0
```

## Exemple concret :

Pour monter le partage "Documents" d'un serveur avec l'adresse IP 192.168.1.100 dans le dossier `/mnt/partage` :

```
sudo mount -t cifs //192.168.1.100/Documents /mnt/partage -o user=john,password=secret
```

## Conseils supplémentaires :

- **Options supplémentaires :**
  - `vers=yes` : Monte le partage en mode vers.
  - `iocharset=utf8` : Pour les fichiers avec des caractères spéciaux.
- **Sécurité :**
  - Éviter de mettre les mots de passe en clair dans `/etc/fstab`. Utiliser des outils comme `secrets`.
- **Démonter le partage :**

```
sudo umount /chemin/local
```

## Pour aller plus loin :

- **Documentation officielle :** Consulter la documentation d'Ubuntu pour plus de détails.
- **Tutoriels vidéo :** De nombreuses vidéos expliquent le processus étape par étape.

## En résumé :

Monter un point Samba est un moyen simple d'accéder à des fichiers partagés sur un réseau local. Avec ces quelques commandes, tu devrais pouvoir le faire en quelques minutes.

# Crontab sous Ubuntu : Planifiez vos tâches automatiquement

## Qu'est-ce que crontab ?

Crontab est un outil puissant sous Linux, dont Ubuntu fait partie, qui permet de planifier l'exécution de commandes ou de scripts à intervalles réguliers. Cela peut être très utile pour automatiser des tâches répétitives, comme des sauvegardes, des mises à jour, ou encore l'envoi de rapports.

## Comment fonctionne crontab ?

Crontab utilise un système de cinq champs pour spécifier quand une tâche doit être exécutée :

- **Minute (0-59)**
- **Heure (0-23)**
- **Jour du mois (1-31)**
- **Mois de l'année (1-12)**
- **Jour de la semaine (0-7; 0 ou 7 est le dimanche)**

Chaque champ peut contenir des valeurs spécifiques, des listes de valeurs séparées par des virgules, des ranges (par exemple, 1-5), ou des incréments (par exemple, \*/5 pour toutes les 5 minutes).

## Comment utiliser crontab ?

Pour éditer votre crontab, utilisez la commande suivante dans votre terminal :

```
crontab -e
```

Vous serez alors invité à choisir un éditeur de texte (comme nano ou vim).

### Exemple de tâche cron:

```
***** /usr/bin/find /var/log -name "*.log" -mtime +7 -exec rm -f {} \;
```

Cette ligne exécutera chaque minute une commande qui supprimera tous les fichiers log dans le répertoire /var/log qui ont plus de 7 jours.

## Exemples d'utilisation de crontab

- **Sauvegardes automatiques:** Planifiez des sauvegardes incrémentales ou complètes de vos données.
- **Mises à jour automatiques:** Mettez à jour régulièrement les logiciels de votre système.
- **Envoi de rapports:** Envoyez des rapports par email ou les publiez sur un serveur.
- **Nettoyage de fichiers:** Supprimez automatiquement les fichiers temporaires ou les anciens logs.

## Conseils supplémentaires

- **Soyez précis:** Plus vos spécifications sont précises, moins votre système sera sollicité inutilement.
- **Testez vos tâches:** Avant de planifier une tâche importante, testez-la en mode manuel pour vous assurer qu'elle fonctionne correctement.
- **Utilisez des scripts:** Pour des tâches complexes, créez des scripts shell et appelez-les depuis votre crontab.
- **Vérifiez les logs:** Cron génère des logs qui peuvent vous aider à déboguer les problèmes. Le fichier principal est généralement situé dans `/var/log/cron`.

## Conclusion

Crontab est un outil indispensable pour automatiser des tâches sous Ubuntu. En maîtrisant son fonctionnement, vous gagnerez en productivité et en fiabilité. N'hésitez pas à expérimenter et à consulter la documentation officielle pour plus d'informations.

### Ressources supplémentaires:

- **Tutoriel DigitalOcean (en anglais):** <https://www.digitalocean.com/community/tutorials/how-to-use-cron-to->

[automate-tasks-ubuntu-1804-fr](#)

- **Documentation Ubuntu-fr:** <https://doc.ubuntu-fr.org/cron>



# La Gestion des droits des fichiers et répertoires

Sous Ubuntu, la gestion des droits des fichiers et répertoires est cruciale pour la sécurité et le bon fonctionnement du système. Voici un aperçu des permissions, de leur configuration et des commandes courantes pour les manipuler.

## 1. Structure des Permissions

Sous Ubuntu, chaque fichier et répertoire a des permissions associées, définies pour trois catégories d'utilisateurs :

- **Propriétaire** (user ou `u`) : la personne ayant créé le fichier/répertoire.
- **Groupe** (group ou `g`) : ensemble d'utilisateurs autorisés.
- **Autres** (other ou `o`) : tous les autres utilisateurs du système.

Les permissions sont représentées par trois types d'autorisations :

- **Lecture** (`r` pour read) : capacité de lire le contenu d'un fichier ou de lister un répertoire.
- **Écriture** (`w` pour write) : autorisation de modifier un fichier ou de créer des éléments dans un répertoire.
- **Exécution** (`x` pour execute) : permission d'exécuter un fichier (utile pour les scripts et les binaires) ou d'accéder aux sous-répertoires.

Exemple d'affichage des permissions via la commande `ls -l` :

```
-rwxr-xr-- 1 utilisateur groupe taille date fichier
```

- Le premier caractère indique le type (fichier `-`, répertoire `d`, etc.)
- Les 9 caractères suivants montrent les permissions pour le propriétaire, le groupe et les autres (`rwx`, `r-x`, `r--` ici).

## 2. Gestion des Permissions avec `chmod`

La commande `chmod` (change mode) permet de modifier les droits d'accès. Deux méthodes sont courantes :

- **Symbole** (symbolic) : utilise les lettres pour ajouter ou retirer des permissions.
- **Octal** (numérique) : chaque droit est représenté par un chiffre.

### Utilisation Symbolique

Les opérateurs sont `+` pour ajouter, `-` pour retirer, et `=` pour définir spécifiquement des permissions. Exemple :

```
chmod u+x fichier # Ajoute le droit d'exécution pour le propriétaire
chmod g-w fichier # Retire le droit d'écriture pour le groupe
chmod o=r fichier # Donne uniquement le droit de lecture aux autres
```

### Utilisation Octale

L'octal associe chaque droit (lecture, écriture, exécution) à un chiffre : `4` (lecture), `2` (écriture), `1` (exécution).

- `7` (`rwx`) pour tous les droits.
- `5` (`r-x`) pour lecture et exécution.
- `0` (`---`) pour aucun droit.

Exemples :

```
chmod 755 fichier # Propriétaire : tous les droits, groupe et autres : lecture et exécution
chmod 644 fichier # Propriétaire : lecture et écriture, groupe et autres : lecture uniquement
```

## 3. Changer le Propriétaire et le Groupe avec `chown` et `chgrp`

Pour changer le propriétaire d'un fichier ou d'un répertoire :

```
chown nouveau_proprietaire fichier
```

Pour changer le groupe associé :

```
chgrp nouveau_groupe fichier
```

Combiner les deux dans `chown` :

```
chown nouveau_proprietaire:nouveau_groupe fichier
```

## 4. Permissions Avancées : `sudo`, ACL, et `umask`

- **Sudo** : Les utilisateurs non root peuvent utiliser `sudo` pour exécuter des commandes avec des privilèges élevés, par exemple, pour modifier des fichiers système.
- **ACL (Access Control List)** : En plus des permissions de base, ACL permet d'accorder des permissions personnalisées pour des utilisateurs spécifiques :

```
setfacl -m u:utilisateur:rwx fichier # Ajoute des permissions spécifiques  
getfacl fichier # Vérifie les permissions ACL
```

- **Umask** : Définit les permissions par défaut pour les nouveaux fichiers et répertoires. Le `umask` est soustrait des permissions maximales (777 pour répertoires et 666 pour fichiers).

```
umask 022 # Par défaut, enlève l'écriture pour le groupe et les autres
```

# apt-cacher-ng : Un cache HTTP pour les paquets APT

## Introduction

**Apt-cacher-ng** est un outil open-source permettant de mettre en place un cache HTTP pour les paquets APT. Il est spécialement conçu pour les environnements réseau où plusieurs machines utilisent les mêmes dépôts de paquets. Cet outil permet de réduire la consommation de bande passante et d'améliorer les temps de téléchargement des paquets en stockant localement une copie des paquets déjà téléchargés.

Dans cet article, nous allons explorer le fonctionnement d'apt-cacher-ng, son installation, sa configuration, et son utilisation dans un environnement Linux.

## Pourquoi utiliser apt-cacher-ng ?

1. **Économie de bande passante** : Dans des réseaux locaux avec de nombreuses machines, apt-cacher-ng permet de télécharger les paquets depuis Internet une seule fois. Les autres machines récupéreront alors les paquets depuis le cache.
2. **Amélioration des performances** : En stockant les paquets localement, apt-cacher-ng réduit le temps d'installation des paquets, surtout dans les environnements où la connexion Internet est lente.
3. **Réduction de la charge sur les serveurs distants** : En centralisant les téléchargements, apt-cacher-ng permet de réduire la charge sur les dépôts publics.

## Installation d'apt-cacher-ng

Pour installer apt-cacher-ng sur un système basé sur Debian ou Ubuntu, ouvrez un terminal et lancez les commandes suivantes :

```
sudo apt update
sudo apt install apt-cacher-ng
```

Après l'installation, apt-cacher-ng démarre automatiquement en tant que service. Vous pouvez vérifier son état avec :

```
sudo systemctl status apt-cacher-ng
```

Par défaut, apt-cacher-ng écoute sur le port 3142, mais cela peut être modifié dans le fichier de configuration.

## Configuration de apt-cacher-ng

Le fichier de configuration principal d'apt-cacher-ng se trouve dans `/etc/apt-cacher-ng/acng.conf`. Les options par défaut conviennent à la majorité des utilisateurs, mais voici quelques paramètres importants à connaître :

- **Port** : Par défaut, apt-cacher-ng utilise le port 3142. Vous pouvez changer cela en modifiant la ligne `Port: 3142`.
- **CacheDir** : Indique l'emplacement où les fichiers en cache seront stockés. Par défaut, cela est configuré sur `/var/cache/apt-cacher-ng`.
- **Remap** : Cette directive permet de rediriger les requêtes vers des dépôts spécifiques, par exemple Debian, Ubuntu, ou autres.

## Exemple de configuration

Voici un extrait de configuration typique dans `/etc/apt-cacher-ng/acng.conf` :

```
# Port utilisé par apt-cacher-ng
Port: 3142

# Répertoire de cache
CacheDir: /var/cache/apt-cacher-ng
```

```
# Directive de redirection pour les dépôts
Remap-debrep: file:deb_mirror*.gz /debian ; http://deb.debian.org/debian
Remap-uburep: file:ubuntu_mirror*.gz /ubuntu ; http://archive.ubuntu.com/ubuntu
```

Une fois les modifications effectuées, redémarrez le service pour appliquer les changements :

```
sudo systemctl restart apt-cacher-ng
```

# Configuration des clients pour utiliser apt-cacher-ng

Pour que les machines clientes utilisent le cache, il faut modifier leurs sources de paquets APT pour rediriger les requêtes HTTP vers le serveur apt-cacher-ng.

## 1. Méthode 1 : Modification du fichier APT

Vous pouvez ajouter la configuration suivante dans chaque fichier `/etc/apt/apt.conf.d/` de chaque client pour rediriger les requêtes vers apt-cacher-ng :

```
Acquire::http::Proxy "http://[IP_du_serveur]:3142";
```

## 2. Méthode 2 : Modifier les sources de dépôts

Vous pouvez également modifier les fichiers `sources.list` pour inclure directement l'adresse de votre cache :

```
deb http://[IP_du_serveur]:3142/debian buster main
deb http://[IP_du_serveur]:3142/ubuntu focal main
```

Remplacez `[IP_du_serveur]` par l'adresse IP de la machine où apt-cacher-ng est installé.

# Gestion du cache

Avec le temps, le cache peut occuper beaucoup d'espace disque. Apt-cacher-ng fournit plusieurs méthodes pour gérer ce cache :

- **Purge des paquets obsolètes** : Pour supprimer les paquets non utilisés, utilisez la commande :

```
sudo apt-cacher-ng -d
```

- **Configuration de la taille du cache** : Dans le fichier de configuration, vous pouvez spécifier des options pour limiter la taille du cache, comme `MaxUsedSize` pour restreindre l'espace disque utilisé.

# Surveillance et logs

Les fichiers de log d'apt-cacher-ng se trouvent par défaut dans `/var/log/apt-cacher-ng/`. Il existe plusieurs fichiers de log, comme `access.log` et `error.log`, qui permettent de suivre l'activité du cache et de détecter d'éventuels problèmes.

Vous pouvez également surveiller les statistiques en temps réel sur l'interface Web d'apt-cacher-ng, accessible via `http://[IP_du_serveur]:3142/acng-report.html`.

# Sécurisation de apt-cacher-ng

Si votre cache est accessible depuis un réseau étendu, vous pouvez vouloir restreindre son accès. Dans le fichier de configuration, vous pouvez utiliser des options comme `BindAddress` pour limiter l'accès à des adresses IP spécifiques.

# Conclusion

Apt-cacher-ng est un outil puissant pour les administrateurs réseau et les utilisateurs qui gèrent plusieurs systèmes Linux. En centralisant les téléchargements de paquets, il permet d'optimiser l'utilisation de la bande passante et d'améliorer les performances des installations.

# Ubuntu - Encrypter son Répertoire Home



Il est possible d'encrypter votre répertoire personnel (homedir) après l'installation d'Ubuntu, mais cela peut être un processus délicat. Voici les deux principales options pour le faire après coup :

## Option 1 : Utilisation de `ecryptfs` pour chiffrer le répertoire personnel

`ecryptfs` est un système de fichiers qui permet de chiffrer facilement un répertoire personnel. Ubuntu propose un outil intégré pour cela, mais gardez à l'esprit que cela nécessite de déplacer temporairement vos fichiers et peut entraîner des complications si vous ne suivez pas correctement les étapes.

### Étapes :

1. **Installer les paquets nécessaires** : Si `ecryptfs` n'est pas encore installé, vous pouvez l'installer en exécutant :

```
sudo apt update
sudo apt install ecryptfs-utils
```

2. **Créer une sauvegarde de vos données** : Avant de procéder, il est fortement recommandé de sauvegarder votre répertoire personnel pour éviter toute perte de données.
3. **Monter et chiffrer votre répertoire personnel** : Vous pouvez maintenant configurer l'encryptage de votre répertoire personnel avec la commande suivante :

```
sudo ecryptfs-migrate-home -u <votre_nom_utilisateur>
```

Remplacez `<votre_nom_utilisateur>` par votre nom d'utilisateur. Cette commande va déplacer vos fichiers vers un répertoire chiffré tout en maintenant votre homedir accessible.

4. **Vérification** : Après avoir effectué cette opération, vous pouvez vérifier si le répertoire est bien chiffré en consultant le contenu du répertoire. Les fichiers devraient maintenant être stockés dans un format chiffré.
5. **Redémarrage et vérification** : Redémarrez votre machine et vérifiez si l'encryptage fonctionne en vous reconnectant. Le répertoire devrait être monté de manière transparente lors de la connexion.

---

## Option 2 : Utilisation de `LUKS` pour chiffrer une partition

Si vous préférez un chiffrement de niveau disque, vous pouvez utiliser LUKS pour chiffrer une partition entière qui contiendra votre homedir. Cela nécessite de créer une nouvelle partition ou de réutiliser une partition existante.

### Étapes :

1. **Sauvegarde** : Avant tout, faites une sauvegarde de vos données actuelles, car cette opération risque de tout effacer sur la partition cible.

2. **Créer une nouvelle partition (si nécessaire)** : Utilisez `gparted` ou `fdisk` pour créer une nouvelle partition ou redimensionner une partition existante pour allouer de l'espace pour le répertoire personnel.
3. **Chiffrer la partition avec LUKS** : Utilisez `cryptsetup` pour chiffrer la partition :

```
sudo cryptsetup luksFormat /dev/sdX
```

Remplacez `/dev/sdX` par l'identifiant de votre partition. Vous devrez entrer une passphrase pour le chiffrement.

4. **Ouvrir la partition chiffrée** : Après avoir chiffré la partition, vous devez l'ouvrir avec :

```
sudo cryptsetup luksOpen /dev/sdX my_encrypted_home
```

5. **Créer un système de fichiers sur la partition** : Créez un système de fichiers sur la partition chiffrée :

```
sudo mkfs.ext4 /dev/mapper/my_encrypted_home
```

6. **Monter la partition et transférer vos fichiers** : Montez la partition chiffrée et copiez vos fichiers personnels dedans :

```
sudo mount /dev/mapper/my_encrypted_home /mnt  
rsync -av /home/<votre_nom_utilisateur> /mnt/
```

7. **Mettre à jour /etc/fstab et /etc/crypttab** : Ajoutez la partition chiffrée à `/etc/crypttab` pour qu'elle soit ouverte automatiquement au démarrage :

```
my_encrypted_home /dev/sdX none luks
```

Ensuite, modifiez `/etc/fstab` pour ajouter la partition dans le système de fichiers :

```
/dev/mapper/my_encrypted_home /home ext4 defaults 0 2
```

8. **Redémarrage et vérification** : Redémarrez votre machine pour vérifier que tout fonctionne correctement. Vous devrez entrer la passphrase pour ouvrir la partition chiffrée au démarrage.

---

## Conclusion

- Si vous souhaitez une méthode plus simple, `ecryptfs` est une bonne option, mais elle n'est plus aussi largement recommandée pour les nouvelles installations.
- Si vous souhaitez un chiffrement de niveau disque, `LUKS` offre une solution plus robuste mais nécessite plus d'étapes et peut impliquer des risques si les données ne sont pas correctement sauvegardées avant.

Dans les deux cas, il est important de bien comprendre les implications de la gestion des clés et des mots de passe, car perdre la clé de chiffrement peut rendre vos données irrécupérables.

# Installer Docker sous Ubuntu

## Mettre à jour le système

Ouvrez un terminal et mettez à jour la liste des paquets :

```
sudo apt update  
sudo apt upgrade -y
```

## Installer les dépendances nécessaires

Docker nécessite quelques outils pour fonctionner correctement :

```
sudo apt install -y ca-certificates curl gnupg lsb-release
```

## Ajouter le dépôt Docker officiel

Ajoutez la clé GPG officielle de Docker :

```
sudo mkdir -p /etc/apt/keyrings  
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
```

Ajoutez le dépôt Docker :

```
echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu  
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

## Installer Docker

Mettez à jour la liste des paquets et installez Docker :

```
sudo apt update  
sudo apt install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

## Vérifier l'installation

Vérifiez que Docker est installé et fonctionne correctement :

```
sudo systemctl status docker  
docker --version
```

Vous devriez voir une sortie confirmant que Docker est actif.

## (Optionnel) Exécuter Docker sans `sudo`

Si vous souhaitez exécuter Docker en tant qu'utilisateur non root :

```
sudo usermod -aG docker $USER
```

Déconnectez-vous puis reconnectez-vous pour que les modifications prennent effet. Testez ensuite avec :

```
docker run hello-world
```

# (Optionnel) Activer Docker au démarrage

Assurez-vous que Docker démarre automatiquement avec le système :

```
sudo systemctl enable docker
```

---



# Installer Infomaniak kDrive sous Ubuntu

KDrive est une solution de stockage en cloud développée par [Infomaniak](#), qui permet de collaborer, partager et accéder à vos données depuis tous vos appareils. Voici un aperçu de son fonctionnement :

1. **Stockage et Accessibilité** : KDrive permet de stocker jusqu'à 106 To de données, accessibles depuis n'importe où, que ce soit au bureau, à la maison ou en déplacement. Vos données sont encryptées et hébergées exclusivement en Suisse, garantissant ainsi une sécurité et une confidentialité optimales .
2. **Compatibilité et Synchronisation** : KDrive est compatible avec macOS, Windows, Linux, iOS et Android, ce qui permet de gérer directement vos données dans votre gestionnaire de fichiers. La fonction de synchronisation sélective des dossiers permet de personnaliser les éléments à synchroniser entre vos appareils .
3. **Édition de Documents** : KDrive permet de travailler en ligne sur vos fichiers Word, Excel et PowerPoint. Il est également compatible avec LibreOffice et OpenOffice, offrant ainsi une flexibilité dans le choix des outils de productivité .
4. **Sécurité et Gestion** : KDrive offre des fonctionnalités avancées de gestion des utilisateurs, des droits d'accès, des statistiques d'utilisation, et permet la récupération des données après le départ d'un utilisateur. Une fonctionnalité de coffre-fort est également prévue pour chiffrer les dossiers avec une clé privée, ajoutant une couche supplémentaire de sécurité pour les données sensibles .
5. **Intégration avec d'autres Services** : KDrive peut être intégré avec des NAS comme Synology pour la sauvegarde et la synchronisation des données, offrant une solution complète pour la gestion des fichiers .
6. **Offres et Tarifs** : KDrive propose plusieurs formules adaptées aux besoins des particuliers et des professionnels, avec des options de stockage allant jusqu'à 18 To. Les tarifs sont compétitifs et offrent des fonctionnalités similaires à celles des géants du marché comme Dropbox et Google Drive .

En résumé, KDrive est une solution de cloud sécurisée et respectueuse de la vie privée, offrant une multitude de fonctionnalités pour la gestion et le partage de données.

## 1. Téléchargement du fichier ApplImage :

- Rendez-vous sur le [site officiel d'Infomaniak](#) et téléchargez le fichier `.ApplImage` de kDrive.

## 2. Autorisation d'exécution :

- Une fois le fichier téléchargé, faites un clic droit dessus et sélectionnez "Propriétés".
- Allez dans l'onglet "Permissions" et cochez la case "Autoriser l'exécution du fichier comme un programme".

## 3. Installation de dépendances :

- Assurez-vous d'avoir installé `libfuse2` pour que les ApplImages fonctionnent correctement. Vous pouvez l'installer via la commande suivante dans le terminal :

```
sudo apt-get install libfuse2
```

## 4. Exécution de kDrive :

- Double-cliquez sur le fichier `.ApplImage` pour lancer kDrive. Si cela ne fonctionne pas, vous pouvez également l'exécuter via le terminal en naviguant dans le répertoire où se trouve le fichier et en tapant :

```
./kDrive-x.x.x.ApplImage
```

- Remplacez `x.x.x` par la version téléchargée.

## 5. Ajout au démarrage (optionnel) :

- Pour que kDrive se lance automatiquement au démarrage, vous pouvez l'ajouter aux applications de démarrage. Allez dans "Paramètres système" > "Applications au démarrage" et ajoutez kDrive à la liste.