

Installer Jellyfin Media Server sur Debian

Table of Contents

- [Installer Jellyfin Media Server sur Debian](#)
 - [Caractéristiques de Jellyfin](#)
- [Media serveur Jellyfin sur Raspberry Pi](#)
 - [Préparer Raspberry Pi pour Jellyfin](#)
 - [Installer Jellyfin sur le Raspberry Pi](#)
 - [Configuration initiale de Jellyfin](#)
 - [Connexion à Jellyfin](#)
 - [Activation de l'accélération matérielle pour Jellyfin](#)
 - [Ajout de l'utilisateur jellyfin au groupe "video"](#)
 - [Augmentation de la mémoire GPU](#)
 - [Dire à Jellyfin d'utiliser l'accélération matérielle](#)
 - [Gérer les problèmes d'autorisation](#)
 - [Configuration du proxy inverse](#)
 - [Nginx](#)
 - [Apache](#)
 - [Activer HTTPS](#)
 - [Mise à jour de Jellyfin](#)

Ce tutoriel explique comment installer le serveur multimédia **Jellyfin** sur un serveur Debian 10.

Jellyfin est une application gratuite et open source qui permet d'organiser une collection de films, émissions de télévision, musique et photos dans une belle interface et de diffuser ces fichiers multimédias sur PC, tablette, téléphone, TV, Roku, etc. sur le réseau ou via l'Internet. **Jellyfin** peut être installé sur Linux, MacOS et Windows.

Caractéristiques de Jellyfin

Jellyfin est un fork du serveur multimédia **Emby**. Il contient beaucoup des mêmes fonctionnalités que **Plex** et **Emby**.

- Contrairement à **Plex** ou **Emby**, **Jellyfin** est 100% gratuit et open source. Pas de pubs. Aucune limite de lecture sur les applications mobiles. (Bien que l'application iOS ne puisse pas lire les vidéos en arrière-plan.)
- Permet de regarder la télévision en direct et définir des enregistrements automatiques pour étendre votre bibliothèque.
- Permet de récupérer automatiquement les illustrations, les métadonnées de TheTVDB, TheMovieDB, The OpenMovie Database et Rotten Tomatoes.
- Prend en charge DLNA.
- Des plugins facultatifs peuvent être installés pour fournir des fonctionnalités supplémentaires.
- Prend en charge l'accélération matérielle de l'encodage/décodage vidéo à l'aide de FFMpeg.
- Et plus.

Media serveur Jellyfin sur Raspberry Pi

Jellyfin est né en décembre 2018 de la dernière version Open Source d'Emby et avec la vocation de rester 100% gratuit pour tous. **Jellyfin** permet de scanner les répertoires à la recherche des films, séries TV ou musiques qui y sont stockés, pour permettre d'y accéder depuis n'importe quel navigateur web ou application compatible, avec une interface agréable. **Jellyfin** permet également de regarder la TV en y ajoutant un tuner TV et dispose de nombreuses extensions lui ajoutant d'autres fonctionnalités : IPTV, authentification, téléchargement automatique des sous-titres, notifications, ...

Préparer Raspberry Pi pour Jellyfin

Avant d'installer le serveur multimédia **Jellyfin** sur Raspberry Pi, il faut effectuer un travail préparatoire essentiel.

Comme **Jellyfin** est disponible via son référentiel, il faut l'ajouter pour que le gestionnaire de packages installe **Jellyfin**.

Mettre à jour le système d'exploitation de Raspberry Pi en utilisant les deux commandes suivantes.

```
sudo apt update  
sudo apt full-upgrade
```

Par défaut, le gestionnaire de packages apt ne prend pas en charge les référentiels s'exécutant derrière HTTPS il faut installer certains packages pour accéder au référentiel de packages **Jellyfin**.

Pour contourner ce problème, installer le package **apt-transport-https** en exécutant la commande suivante.

```
sudo apt install apt-transport-https
```

Ce package ajoute la prise en charge du protocole de transport HTTPS au gestionnaire de packages apt.

Ensuite, il faut importer la clé de signature GPG dans notre Pi. Utiliser la commande suivante pour rediriger la clé directement vers le gestionnaire de packages.

```
wget -O - https://repo.jellyfin.org/debian/jellyfin_team.gpg.key | sudo apt-key add -
```

Les clés GPG sont un élément essentiel de la sécurité des référentiels de packages. Les clés permettent de garantir que seuls les logiciels signés par le référentiel seront installés.

Maintenant ajouter le référentiel **Jellyfin** à Raspberry Pi en utilisant la commande suivante:

```
echo "deb [arch=$( dpkg --print-architecture )]  
https://repo.jellyfin.org/debian $( lsb_release -c -s ) main" | sudo tee  
/etc/apt/sources.list.d/jellyfin.list
```

Cette ligne saisit automatiquement l'architecture système actuelle et sa version actuelle et remplit les blancs.

Enfin, il faut mettre à jour la liste des packages en utilisant la commande suivante:

```
sudo apt update
```



Lorsqu'on modifie le fichier sources, il faut mettre à jour la liste des packages. Sans mise à jour, le gestionnaire de packages ne sera pas au courant des modifications.



Avant de continuer, il est préférable de configurer le Raspberry Pi pour utiliser une adresse IP statique, cela facilitera la connexion au serveur multimédia **Jellyfin** lorsque l'appareil redémarrera.

Installer Jellyfin sur le Raspberry Pi

Avec le référentiel de packages ajouté, on peut maintenant installer le serveur multimédia **Jellyfin** sur le Raspberry Pi, il suffit d'exécuter la commande suivante pour installer **Jellyfin**.

```
sudo apt install jellyfin
```

Cette commande téléchargera et installera **Jellyfin** à partir du référentiel de packages ajouté.

Pendant le processus d'installation, **Jellyfin** configurera quelques éléments sur notre Raspberry Pi:

1. La première chose qu'il fait est de créer un nouvel utilisateur appelé **Jellyfin**. Cet utilisateur est utilisé pour exécuter le logiciel sur le Raspberry Pi et est également l'utilisateur qui devra accéder aux fichiers.
2. Deuxièmement, il crée également un service pour **Jellyfin**. Ce service démarrera automatiquement le serveur multimédia au démarrage et permettra de démarrer, d'arrêter et de redémarrer facilement le service.

Configuration initiale de Jellyfin

Maintenant que le serveur multimédia **Jellyfin** est installé sur Raspberry Pi, on peut accéder à son interface Web.

L'interface Web est simple à utiliser et prend en charge la plupart des principaux navigateurs Web.

Pour accéder à l'interface Web de Jellyfin, il faut connaître l'adresse IP de votre Raspberry Pi.

http: // [IPADRESSE]: 8096



Le moyen le plus rapide d'obtenir l'adresse IP locale de votre Pi est d'utiliser la commande hostname:

```
hostname -I
```

Avant de pouvoir commencer à utiliser **Jellyfin**, il faut configurer le serveur multimédia:

1. sélectionner la langue d'affichage,
2. cliquer sur le bouton Suivant -> pour continuer.
3. créer un utilisateur administrateur pour accéder à **Jellyfin** en définissant un nom d'utilisateur et un mot de passe dans l'interface
4. cliquer sur le bouton Suivant -> pour continuer.
5. ajouter des bibliothèques multimédias au serveur multimédia **Jellyfin**, en cliquant sur le bouton Ajouter une bibliothèque multimédia:
 1. sélectionner un type de contenu (Le type de contenu est ce qui dicte la façon dont **Jellyfin** récupérera les informations sur ces fichiers)
 2. ajouter les dossiers où se trouvent les fichiers multimédias, en cliquant sur le symbole plus (+) à côté de Dossiers
 3. Une fois les dossiers ajoutés, cliquez sur le bouton « Ok » situé en bas de l'écran
6. cliquer sur le bouton Suivant -> pour continuer.
7. définir la langue par défaut que **Jellyfin** utilisera pour récupérer les métadonnées (Lorsque **Jellyfin** se connecte à des services tels que TVDB, il peut récupérer les noms des épisodes et d'autres informations dans la langue locale si elles sont disponibles.)
8. cliquer sur le bouton Suivant -> pour continuer.
9. définir la langue par défaut des métadonnées
10. (optionnel) configurer l'accès à distance du serveur multimédia **Jellyfin**
11. cliquer sur le bouton Suivant -> pour confirmer vos paramètres.
12. cliquer sur le bouton Terminer pour finaliser la configuration.



Si on a fait une erreur dans la configuration initiale, on peut recommencer:

- Modifier le fichier de configuration Jellyfin. `sudo nano /etc/jellyfin/system.xml` en remplaçant
`<IsStartupWizardCompleted>true</IsStartupWizardCompleted>` par
`<IsStartupWizardCompleted>>false</IsStartupWizardCompleted>`
- Enregistrer et fermer le fichier
- Redémarrer ensuite Jellyfin `sudo systemctl restart jellyfin`
- Se reconnecter sur `localhost:8096/web` et on aura à nouveau l'assistant de configuration.

Connexion à Jellyfin

Pour accéder à l'interface Web **Jellyfin** sur le Raspberry Pi, il faut se connecter avec le nom d'utilisateur et le mot de passe créés lors de la configuration initiale de **Jellyfin**.

Une fois ces informations de connexion saisies, cliquez sur le bouton Connexion pour se connecter à l'interface Web.

Il est alors possible de regarder les médias via le serveur multimédia **Jellyfin**.

Activation de l'accélération matérielle pour Jellyfin

Il peut arriver que le périphérique sur lequel on veut lire une vidéo ne soit pas compatible avec le format stocké. C'est là qu'intervient le transcodage dans **Jellyfin**, tout comme le faisait **Plex** ou **Emby**. Ce transcodage est effectué à la volée lors de la lecture de la vidéo, et **Jellyfin** a un avantage sur ses concurrents directs, c'est qu'il peut profiter de l'accélération matérielle VAAPI (Video Acceleration API) et NVENC (Nvidia Video ENCoding), ce qui permet de réduire la charge du processeur notamment lorsqu'il faut transcoder une vidéo UltraHD/4K.

L'activation de l'accélération matérielle nécessite d'effectuer des opérations à la fois sur le Raspberry Pi et sur l'interface Web **Jellyfin**.

Ajout de l'utilisateur jellyfin au groupe "video"

Le groupe **video** est un groupe système spécial avec accès à des fonctionnalités telles que le GPU du Raspberry Pi, la commande `usermod` suivante nous permettra d'ajouter l'utilisateur **Jellyfin** au groupe vidéo:

```
sudo usermod -aG video jellyfin
```

L'argument **-a** indique qu'on veut ajouter un attribut à un utilisateur. Le **G** majuscule indique à **usermod** que l'on veut ajouter le groupe **video** à **jellyfin**.

Augmentation de la mémoire GPU

Comme l'accélération matérielle peut être assez gourmande en mémoire, il faut augmenter la quantité de mémoire disponible pour le GPU.

Utiliser l'éditeur de texte pour modifier le fichier de configuration de démarrage.

```
sudo nano /boot/config.txt
```





Il faut être prudent lorsqu'on modifie ce fichier, car on peut potentiellement empêcher le Pi de démarrer. Cependant, il est facile à réparer à partir d'un autre appareil.

Ajouter à la fin de ce fichier une ligne indiquant au système d'exploitation la quantité de mémoire qu'il doit consacrer au GPU. Plus de mémoire vidéo permet au GPU du Raspberry Pi de stocker plus de données, ce qui lui permet de gérer des tâches qui consomment une quantité considérable de mémoire vidéo.

Pour Raspberry Pi 4:

```
gpu_mem=320
```

Pour Raspberry Pi 3:

```
gpu_mem=256
```



On a donné au Raspberry Pi 4 plus de mémoire vidéo pour gérer le transcodage des fichiers HEVC 4k. Le processeur du Pi 3 ne gère pas très bien ces fichiers, donc 256 Mo de RAM vidéo devraient suffire.

On peut essayer d'augmenter la quantité de mémoire dédiée au GPU pour voir si cela améliore encore les performances.

Une fois la ligne ajoutée au fichier, enregistrer et quitter en appuyant sur CTRL + X, suivi de Y, puis de la touche ENTER.

Pour que les modifications d'allocation de mémoire prennent effet, il faut redémarrer le Raspberry Pi.

```
sudo reboot
```

Une fois le redémarrage de votre Raspberry Pi terminé, exécuter les deux commandes suivantes pour voir combien de RAM a été allouée au CPU et au GPU.

```
vcgencmd get_mem arm  
vcgencmd get_mem gpu
```

ci-dessous un exemple de ce à quoi devrait ressembler les informations retournées.

```
arm=704M  
gpu=320M
```

Dire à Jellyfin d'utiliser l'accélération matérielle

Il faut maintenant reconfigurer le serveur multimédia.

Sur la page d'accueil de l'interface Web:

1. passer à la page des paramètres utilisateur, en cliquant sur l'icône de la personne dans le coin supérieur gauche de la fenêtre.
 1. passer au tableau de bord des serveurs multimédias, en cliquant sur l'option **Tableau de bord**.
 1. passer à l'onglet **Lecture** (on peut trouver cette option dans la barre latérale de gauche.). L'onglet "Lecture" est l'endroit où on peut contrôler la façon dont les fichiers sont lus de notre Raspberry Pi vers les clients **Jellyfin**:
 2. Sous l'en-tête **Accélération matérielle**, on doit voir une case de sélection,
 3. cliquer sur cette case, puis sélectionnez l'option **OpenMAX OMX** (OMX est la seule de ces options à prendre en charge le matériel du Raspberry Pi).
 4. faire défiler jusqu'en bas de la fenêtre
2. enregistrer les paramètres en cliquant sur le bouton **Enregistrer**.

Gérer les problèmes d'autorisation

Un problème qu'on peut rencontrer dans un serveur multimédia tel que Jellyfin est celui des problèmes d'autorisation potentiels.

Le système d'autorisation Linux peut parfois être un peu pointilleux, en particulier lorsqu'il s'agit de disques montés.

Dans cette section, on va voir comment définir les autorisations pour un lecteur basé sur ext4.



Les types de format de lecteur alternatifs tels que NTFS ou FAT obligeront à ajuster la façon dont on monte le lecteur. Ces types de format de lecteur n'ont jamais été conçus en pensant au système d'autorisation Linux.

Déterminer le répertoire dans lequel sont stockés les fichiers multimédias (il faut de connaître le chemin exact afin que le script sache pour quel répertoire il doit modifier les autorisations). Pour exemple, on utilisera le répertoire `/home/pi/media/films`.

Passer en mode superutilisateur en exécutant la commande suivante:

```
sudo su
```

Exécuter les commandes suivantes, en remplaçant "[YOURDRIVEPATH]" par le chemin sur lequel on souhaite corriger les autorisations.

```
find [YOURDRIVEPATH] -type d -exec chmod 755 {} \;
```

```
find [YOURDRIVEPATH] -type f -exec chmod 644 {} \;
```

Ces commandes parcourront les fichiers et répertoires et réajusteront les autorisations afin que l'utilisateur **jellyfin** puisse lire les fichiers multimédias qu'il contient.

Une fois ces commandes exécutées, quitter le superutilisateur en exécutant la commande suivante:

```
exit
```

Configuration du proxy inverse

Jellyfin écoute par défaut sur le port 8096, pour pouvoir accéder à l'interface Web sur un port standard, on peut configurer un proxy inverse pour **Jellyfin** avec **Nginx** ou **Apache**.

Nginx

Nginx est un serveur Web et un proxy inverse très populaire. Pour utiliser Nginx, exécuter la commande suivante pour l'installer.

```
sudo apt install nginx
```

Créer ensuite un fichier de conf de serveur pour **Jellyfin**.

```
sudo nano /etc/nginx/conf.d/jellyfin.conf
```

Ajouter le contenu suivant à ce fichier. Remplacer **jellyfin.example.com** par l'adresse ou le nom de domaine du serveur (il faudra également ajouter un enregistrement DNS A pour ce sous-domaine).

```
server {
    listen 80;
    listen [::]:80;
    server_name jellyfin.example.com;

    access_log /var/log/nginx/jellyfin.access;
    error_log /var/log/nginx/jellyfin.error;

    set $jellyfin 127.0.0.1;

    location / {
        proxy_pass http://127.0.0.1:8096;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;

        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
```



```
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_set_header X-Forwarded-Protocol $scheme;
    proxy_set_header X-Forwarded-Host $http_host;

    # Désactiver la mise en mémoire tampon lorsque le proxy nginx
    devient très gourmand en ressources lors du streaming
    proxy_buffering off;
}

# bloc location pour /web - Ceci est purement pour l'esthétique
(/web/#!/ fonctionne au lieu d'avoir à aller à /web/index.html/#!/)
location ~ ^/web/$ {
    # Proxy main Jellyfin traffic
    proxy_pass http://$jellyfin:8096/web/index.html/;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_set_header X-Forwarded-Protocol $scheme;
    proxy_set_header X-Forwarded-Host $http_host;
}

location /socket {
    # Proxy Jellyfin Websockets traffic
    proxy_pass http://$127.0.0.1:8096;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_set_header X-Forwarded-Protocol $scheme;
    proxy_set_header X-Forwarded-Host $http_host;
}

# Security / XSS Mitigation Headers
add_header X-Frame-Options "SAMEORIGIN";
add_header X-XSS-Protection "1; mode=block";
add_header X-Content-Type-Options "nosniff";
}
```

Enregistrer et fermer ce fichier. Tester ensuite la configuration de Nginx.

```
sudo nginx -t
```

Si le test réussit, recharger Nginx pour que la modification prenne effet.

```
sudo systemctl reload nginx
```

On peut désormais accéder au serveur multimédia Jellyfin via jellyfin.example.com.

Apache

Installer le serveur Web **Apache** à l'aide de la commande suivante.

```
sudo apt install apache2
```

Pour utiliser **Apache** comme proxy inverse, il faut activer les modules proxy et le module header.

```
sudo a2enmod proxy proxy_http headers proxy_wstunnel
```

Créer ensuite un fichier hôte virtuel pour **Jellyfin**.

```
sudo nano /etc/apache2/sites-available/jellyfin.conf
```

Mettre les configurations suivantes dans le fichier. Remplacer **jellyfin.example.com** par l'adresse ip ou le nom de domaine.

```
<VirtualHost *:80>
    ServerName jellyfin.example.com
    ErrorDocument 404 /404.html

    #HTTP proxy
    ProxyPass / http://localhost:8096/
    ProxyPassReverse / http://localhost:8096/

    #Websocket proxy
    SSLProxyEngine on
    <Location /:/websockets/notifications>
        ProxyPass wss://localhost:8096/:/websockets/notifications
        ProxyPassReverse wss://localhost:8096/:/websockets/notifications
    </Location>

    Header always unset X-Frame-Options
</VirtualHost>
```

Enregistrer et fermer le fichier. Activer ensuite cet hôte virtuel.

```
sudo a2ensite jellyfin.conf
```

Redémarrer Apache

```
sudo systemctl restart apache2
```

On peut désormais accéder au serveur multimédia Jellyfin en utilisant le nom de domaine `jellyfin.example.com`.

Activer HTTPS

Pour crypter le trafic HTTP pour accéder au serveur Jellyfin depuis l'extérieur, on peut activer HTTPS en installant un certificat TLS gratuit émis par Let's Encrypt. Exécuter la commande suivante pour installer le client Let's Encrypt (certbot).

```
sudo apt install certbot
```

Pour **Nginx**, il faut également installer le plugin **Certbot Nginx**.

```
sudo apt install python3-certbot-nginx
```

Pour **Apache**, il faut installer le plugin Certbot **Apache**.

```
sudo apt install python3-certbot-apache
```

Ensuite, exécuter la commande suivante pour obtenir et installer le certificat TLS.

```
sudo certbot --nginx --agree-tos --redirect --hsts --staple-ocsp --email  
you@example.com -d jellyfin.example.com
```

ou

```
sudo certbot --apache --agree-tos --redirect --hsts --staple-ocsp --email  
you@example.com -d jellyfin.example.com
```

Où:

- **-nginx**: utiliser le plug-in nginx.
- **-apache**: utiliser le plug-in Apache.
- **-agree-tos**: accepter les conditions d'utilisation.
- **-redirect**: force HTTPS par redirection 301.
- **-hsts**: ajouter l'en-tête **Strict-Transport-Security** à chaque réponse HTTP. Forcer le navigateur à toujours utiliser TLS pour le domaine. Se défend contre le stripping SSL/TLS.
- **-staple-ocsp**: active l'agrafage OCSP. Une réponse OCSP valide est agrafée au certificat que le serveur propose pendant TLS.

Le certificat devrait maintenant être obtenu et installé automatiquement.

Et on peut accéder à l'interface Web de Jellyfin via HTTPS : `https://jellyfin.example.com`.

Mise à jour de Jellyfin

Lorsqu'une nouvelle version du serveur multimédia **Jellyfin** sort, on peut la mettre à niveau en exécutant les commandes suivantes.

```
sudo apt update  
sudo apt upgrade
```

Redémarrer ensuite **Jellyfin**.

```
sudo systemctl restart jellyfin
```

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=prive:rpi-jellyfin>

Last update: **2025/02/19 10:59**

