

Comment installer Jellyfin Media Server sur Ubuntu 22.04

Jellyfin est un logiciel de serveur multimédia gratuit et open source, permettant aux utilisateurs d'héberger et de gérer des bibliothèques multimédias personnelles, diffusez du contenu sur tous les appareils et donnez accès aux autres. C'est une alternative aux serveurs multimédias propriétaires, offrant personnalisation étendue et contrôle de la confidentialité. C'est une alternative aux serveurs de fichiers multimédias propriétaires comme Emby et Plex.

Dans ce guide, nous allons vous montrer comment installer le serveur multimédia Jellyfin sur Ubuntu 22.04. Vous installerez Jellyfin avec Apache comme proxy inverse et sécurisez Jellyfin avec les certificats SSL/TLS de Letsencrypt.

Conditions préalables

Avant de commencer, assurez-vous d'avoir les éléments suivants :

- Un serveur Ubuntu 22.04.
- Un utilisateur non root avec les privilèges sudo.
- Un nom de domaine pointait vers une adresse IP de serveur.

Ajout du référentiel Jellyfin

Pour commencer, exécutez la commande ci-dessous pour installer des dépendances telles que software-properties-common, apt-transport-https, ca-certificates, gnupg et curl sur votre système Ubuntu.

```
sudo apt install software-properties-common apt-transport-https ca-certificates gnupg curl -y
```

Ensuite, exécutez la commande suivante pour activer le référentiel « univers » sur votre machine Ubuntu.

```
sudo add-apt-repository universe
```

```
root@jammy64:~#  
root@jammy64:~# sudo add-apt-repository universe  
Adding component(s) 'universe' to all repositories.  
Press [ENTER] to continue or Ctrl-c to cancel.  
Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease  
Hit:2 http://security.ubuntu.com/ubuntu jammy-security InRelease  
Hit:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease  
Hit:4 http://archive.ubuntu.com/ubuntu jammy-backports InRelease  
Reading package lists... 99%
```

Exécutez maintenant la commande ci-dessous pour créer un nouveau répertoire /etc/apt/keyrings et téléchargez la clé GPG pour le Jellyfin dépôt dedans.

```
sudo mkdir -p /etc/apt/keyrings  
curl -fsSL https://repo.jellyfin.org/jellyfin_team.gpg.key | sudo gpg --dearmor -o /etc/apt/keyrings/jellyfin.gpg
```

Ensuite, exécutez la commande suivante pour ajouter le référentiel Jellyfin à votre machine Ubuntu.

```
cat <<EOF | sudo tee /etc/apt/sources.list.d/jellyfin.sources  
Types: deb  
URIs: https://repo.jellyfin.org/${ awk -F=' ' /^ID=/{ print $NF }' /etc/os-release )  
Suites: $( awk -F=' ' /^VERSION_CODENAME=/{ print $NF }' /etc/os-release )  
Components: main  
Architectures: $( dpkg --print-architecture )  
Signed-By: /etc/apt/keyrings/jellyfin.gpg  
EOF
```

```
root@jammy64:~#  
root@jammy64:~# sudo mkdir -p /etc/apt/keyrings  
root@jammy64:~# curl -fsSL https://repo.jellyfin.org/jellyfin_team.gpg.key | sudo gpg --dearmor -o /etc/apt/keyrings/jellyfin.gpg  
root@jammy64:~#  
root@jammy64:~# cat <<EOF | sudo tee /etc/apt/sources.list.d/jellyfin.sources  
Types: deb  
URIs: https://repo.jellyfin.org/${ awk -F=' ' /^ID=/{ print $NF }' /etc/os-release )  
Suites: $( awk -F=' ' /^VERSION_CODENAME=/{ print $NF }' /etc/os-release )  
Components: main  
Architectures: $( dpkg --print-architecture )  
Signed-By: /etc/apt/keyrings/jellyfin.gpg  
EOF  
Types: deb  
URIs: https://repo.jellyfin.org/ubuntu  
Suites: jammy  
Components: main  
Architectures: amd64  
Signed-By: /etc/apt/keyrings/jellyfin.gpg  
root@jammy64:~#  
root@jammy64:~#
```

Enfin, mettez à jour et actualisez l'index du package sur votre système Ubuntu à l'aide de la commande suivante. Cela récupérera

informations sur les nouveaux packages, y compris celles du référentiel Jellyfin.

```
sudo apt update
```

```
root@jammy64:~#  
root@jammy64:~# sudo apt update  
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease  
Hit:2 http://archive.ubuntu.com/ubuntu jammy InRelease  
Hit:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease  
Hit:4 http://archive.ubuntu.com/ubuntu jammy-backports InRelease  
Get:5 https://repo.jellyfin.org/ubuntu jammy InRelease [6636 B]  
Get:6 https://.mirror.jellyfin.org/ubuntu jammy/main amd64 Packages [2237 B]  
Fetched 8873 B in 2s (3676 B/s)  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done
```

Installation du serveur Jellyfin

Après avoir ajouté le référentiel Jellyfin, vous pouvez maintenant démarrer l'installation de Jellyfin sur votre serveur Ubuntu.

Exécutez la commande apt suivante pour installer le package jellyfin. Lorsque vous y êtes invité, saisissez Y pour confirmer et continuer.

```
sudo apt install jellyfin
```

```
root@jammy64:~#  
root@jammy64:~# sudo apt install jellyfin  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  at jellyfin-ffmpeg5 jellyfin-server jellyfin-web libaac0 libass9 libb  
  libmp3lame0 libmpeg123-0 libogg0 libopenmpt0 libopus0 libpciaccess0 lib  
  libvpx7 libwebp mux3 libx11-xcb1 libx264-163 libx265-199 libxcb-dri2-0  
  libxcb-sync1 libxcb-xfixes0 libxrender1 libxshmfence1 libzvb-common  
Suggested packages:  
  default-mta | mail-transport-agent libbluray-bdj opus-tools opencl-icc  
The following NEW packages will be installed:  
  at jellyfin jellyfin-ffmpeg5 jellyfin-server jellyfin-web libaac0 lib  
  libmp3lame0 libmpeg123-0 libogg0 libopenmpt0 libopus0 libpciaccess0 lib  
  libvpx7 libwebp mux3 libx11-xcb1 libx264-163 libx265-199 libxcb-dri2-0  
  libxcb-sync1 libxcb-xfixes0 libxrender1 libxshmfence1 libzvb-common  
0 upgraded, 43 newly installed, 0 to remove and 61 not upgraded.  
Need to get 156 MB of archives.  
After this operation, 534 MB of additional disk space will be used.  
Do you want to continue? [Y/n] Y
```

Une fois l'installation terminée, exécutez la commande ci-dessous pour vérifier le service jellyfin et assurez-vous que le service est en cours d'exécution et activé.

```
sudo systemctl is-enabled jellyfin  
sudo systemctl status jellyfin
```

Dans le résultat suivant, vous pouvez voir que Jellyfin est en cours d'exécution et activé.

```
root@jammy64:~#  
root@jammy64:~# sudo systemctl is-enabled jellyfin  
enabled  
root@jammy64:~# sudo systemctl status jellyfin  
● jellyfin.service - Jellyfin Media Server  
   Loaded: loaded (/lib/systemd/system/jellyfin.service; enabled; vendor preset: enabled)  
   Drop-In: /etc/systemd/system/jellyfin.service.d  
            └─jellyfin.service.conf  
   Active: active (running) since  
   Main PID: 8087 (jellyfin)  
     Tasks: 17 (limit: 4647)  
    Memory: 90.1M  
       CPU: 12.039s
```

Vous pouvez également vérifier les ports pour Jelly à l'aide de la commande ss ci-dessous.

```
ss -tulpn | grep jellyfin
```

Vous pouvez voir différents ports 8096/tcp pour le trafic HTTP, 8920/tcp pour le trafic HTTPS, 1900/udp et 7359/udp pour le service. découverte automatique.

```
root@jammy64:~#  
root@jammy64:~# ss -tulpn | grep jellyfin  
udp UNCONN 0 0 10.0.2.15:40642 0.0.0.0:* users:(("jellyfin",pid=8087,fd=328))  
udp UNCONN 0 0 127.0.0.1:40671 0.0.0.0:* users:(("jellyfin",pid=8087,fd=330))  
udp UNCONN 0 0 0.0.0.0:1900 0.0.0.0:* users:(("jellyfin",pid=8087,fd=326))  
udp UNCONN 0 0 192.168.5.30:43064 0.0.0.0:* users:(("jellyfin",pid=8087,fd=329))  
udp UNCONN 0 0 0.0.0.0:53872 0.0.0.0:* users:(("jellyfin",pid=8087,fd=327))  
udp UNCONN 0 0 0.0.0.0:7359 0.0.0.0:* users:(("jellyfin",pid=8087,fd=318))  
tcp LISTEN 0 512 0.0.0.0:8096 0.0.0.0:* users:(("jellyfin",pid=8087,fd=310))  
root@jammy64:~#
```

Génération de certificats SSL/TLS

Dans ce guide, vous sécurisez l'installation de Jellyfin avec des certificats SSL/TLS et un proxy inverse Apache. Et maintenant, tu seras générer des certificats SSL/TLS via Certbot et Letsencrypt.

PS : Vous pouvez également utiliser un certificat auto-signé si vous êtes dans l'environnement local. Mais pour un déploiement externe, assurez-vous votre nom de domaine est préparé et pointe vers votre adresse IP Ubuntu.

Exécutez la commande suivante pour installer Certbot sur votre système Ubuntu. Saisissez Y et appuyez sur ENTER pour continuer.

```
sudo apt install certbot
```

Exécutez maintenant la commande ci-dessous pour créer un nouveau répertoire /var/www/html.

```
mkdir -p /var/www/html
```

Enfin, exécutez la commande certbot ci-dessous pour générer des certificats SSL/TLS pour votre nom de domaine Jellyfin. Assurez-vous de modifier l'adresse e-mail et le nom de domaine avec vos coordonnées.

```
sudo certbot certonly --agree-tos --email user@email.com --no-eff-email --webroot -w /var/www/html -d media.hwdomain.io
```

Une fois tout le processus terminé, vos certificats SSL/TLS seront disponibles dans le répertoire. /etc/letsencrypt/live/domain.com

Configuration d'Apache en tant que proxy inverse

Dans cette section, vous allez installer et configurer le serveur Web Apache en tant que proxy inverse pour Jellyfin. Commençons.

Tout d'abord, exécutez la commande suivante pour installer le serveur Web Apache sur votre système.

```
sudo apt install apache2 -y
```

```
root@jammy64:~#  
root@jammy64:~# sudo apt install apache2 -y  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap  
Suggested packages:  
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser  
The following NEW packages will be installed:  
  apache2 apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap  
  mime-support  
0 upgraded, 12 newly installed, 0 to remove and 61 not upgraded.  
Need to get 2121 kB of archives.  
After this operation, 8452 kB of additional disk space will be used.  
Get:1 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libapr1 a
```

Une fois l'installation terminée, exécutez la commande a2enmod ci-dessous pour activer les modules Apache requis sur votre installation.

```
sudo a2enmod proxy proxy_http ssl proxy_wstunnel remoteip http2 headers
```

Exécutez maintenant la commande suivante de l'éditeur nano pour créer un nouveau fichier d'hôte virtuel Apache /etc/apache2/sites-available/jellyfin.conf.

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

Insérez la configuration suivante pour mettre en place un proxy inverse pour Jellyfin. Et assurez-vous de changer le nom de domaine et le chemin des certificats SSL avec vos détails.

```
<VirtualHost *:80>
  ServerName media.hwdomain.io

  # Comment to prevent HTTP to HTTPS redirect
  Redirect permanent / https://media.hwdomain.io/

  ErrorLog /var/log/apache2/media.hwdomain.io-error.log
  CustomLog /var/log/apache2/media.hwdomain.io-access.log combined
</VirtualHost>

# If you are not using an SSL certificate, replace the 'redirect'
# line above with all lines below starting with 'Proxy'
<IfModule mod_ssl.c>
<VirtualHost *:443>
  ServerName media.hwdomain.io
  # This folder exists just for certbot(You may have to create it, chown and chmod it to give apache permission to read it)
  DocumentRoot /var/www/html/jellyfin/public_html

  ProxyPreserveHost On

  # Letsencrypt's certbot will place a file in this folder when updating/verifying certs
  # This line will tell Apache to not to use the proxy for this folder.
  ProxyPass "/.well-known/" "!"

  # Tell Jellyfin to forward that requests came from TLS connections
  RequestHeader set X-Forwarded-Proto "https"
  RequestHeader set X-Forwarded-Port "443"

  ProxyPass "/socket" "ws://192.168.5.30:8096/socket"
  ProxyPassReverse "/socket" "ws://192.168.5.30:8096/socket"

  ProxyPass "/" "http://192.168.5.30:8096/"
  ProxyPassReverse "/" "http://192.168.5.30:8096/"

  SSLEngine on
  SSLCertificateFile /etc/letsencrypt/live/media.hwdomain.io/fullchain.pem
  SSLCertificateKeyFile /etc/letsencrypt/live/media.hwdomain.io/privkey.pem
  Protocols h2 http/1.1

  # Enable only strong encryption ciphers and prefer versions with Forward Secrecy
  SSLCipherSuite HIGH:RC4-SHA:AES128-SHA:!aNULL:!MD5
  SSLHonorCipherOrder on

  # Disable insecure SSL and TLS versions
  SSLProtocol all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1

  ErrorLog /var/log/apache2/media.hwdomain.io-error.log
  CustomLog /var/log/apache2/media.hwdomain.io-access.log combined
</VirtualHost>
</IfModule>
```

Enregistrez et fermez le fichier lorsque vous avez terminé.

Ensuite, exécutez la commande ci-dessous pour activer le fichier d'hôte virtuel jellyfin.conf. Ensuite, vérifiez votre configuration Apache pour vous assurer que vous avez la bonne syntaxe.

```
sudo a2ensite jellyfin.conf
sudo apachectl configtest
```

Si tout se passe bien, vous devriez voir une sortie telle que « Syntaxe OK ».

```

root@jammy64:~#
root@jammy64:~# sudo nano /etc/apache2/sites-available/jellyfin.conf
root@jammy64:~#
root@jammy64:~# sudo a2ensite jellyfin.conf
Enabling site jellyfin.
To activate the new configuration, you need to run:
    systemctl reload apache2
root@jammy64:~# sudo apachectl configtest
AH00112: Warning: DocumentRoot [/var/www/html/jellyfin/public_html] does not e
AH00558: apache2: Could not reliably determine the server's fully qualified do
suppress this message
Syntax OK
root@jammy64:~# sudo systemctl restart apache2
root@jammy64:~# █

```

Exécutez maintenant la commande `systemctl` pour redémarrer le service Apache et appliquer vos modifications.

```
sudo systemctl restart apache2
```

Une fois redémarré, vérifiez le service Apache pour vous assurer qu'il est en cours d'exécution et activé.

```
sudo systemctl is-enabled apache2
sudo systemctl status apache2
```

Si tout se passe bien, vous devriez voir qu'Apache est activé et exécuté sur votre machine Ubuntu.

```

Syntax OK
root@jammy64:~# sudo systemctl restart apache2
root@jammy64:~#
root@jammy64:~# sudo systemctl is-enabled apache2
enabled
root@jammy64:~# sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 1808 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 1812 (apache2)
    Tasks: 105 (limit: 4647)
   Memory: 8.9M
      CPU: 76ms
   CGroup: /system.slice/apache2.service
           └─1812 /usr/sbin/apache2 -k start
             └─1813 /usr/sbin/apache2 -k start
               └─1814 /usr/sbin/apache2 -k start

```

Configuration d'UFW

Une fois Jellyfin installé, configurons UFW pour sécuriser votre installation.

Tout d'abord, exécutez la commande ci-dessous pour ouvrir les ports pour SSH, HTTP et HTTPS.

```
sudo ufw allow OpenSSH
sudo ufw allow 'Apache Full'
```

Ouvrez maintenant les ports 1900/udp et 7359/udp pour la découverte automatique à l'aide de la commande suivante.

```
sudo ufw allow 1900/udp
sudo ufw allow 7359/udp
```

Ensuite, exécutez la commande ci-dessous pour démarrer et activer UFW. Tapez Y pour confirmer.

```
sudo ufw enable
```

Enfin, vérifiez la liste des ports et services sur UFW à l'aide de la commande ci-dessous.

```
sudo ufw status
```

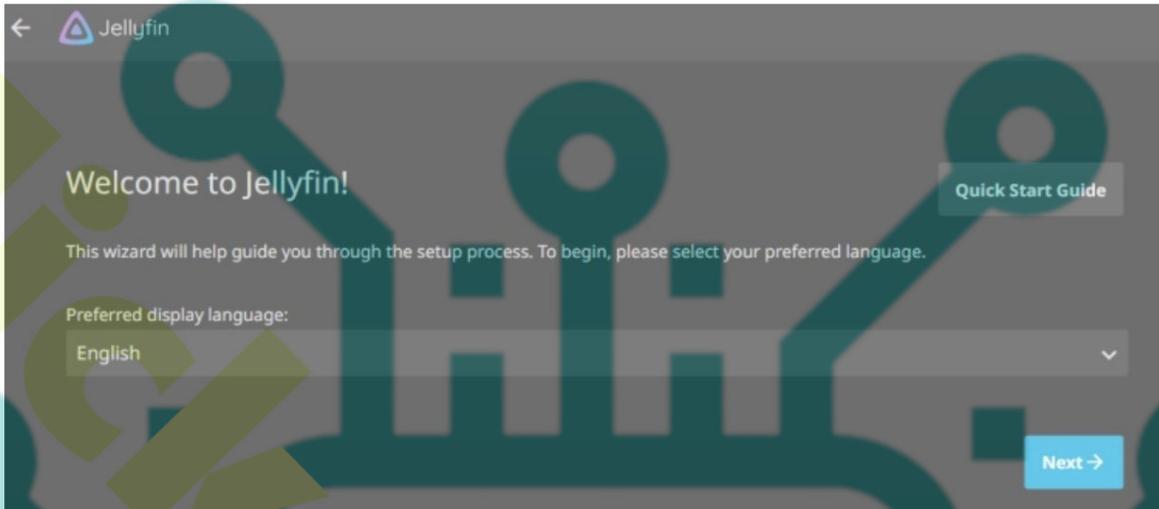
Vous verrez que les profils OpenSSH et « Apache Full » sont activés et que les ports 1900/udp et 7359/udp sont également ajoutés à UFW.

Installation de Jellyfin Media Server

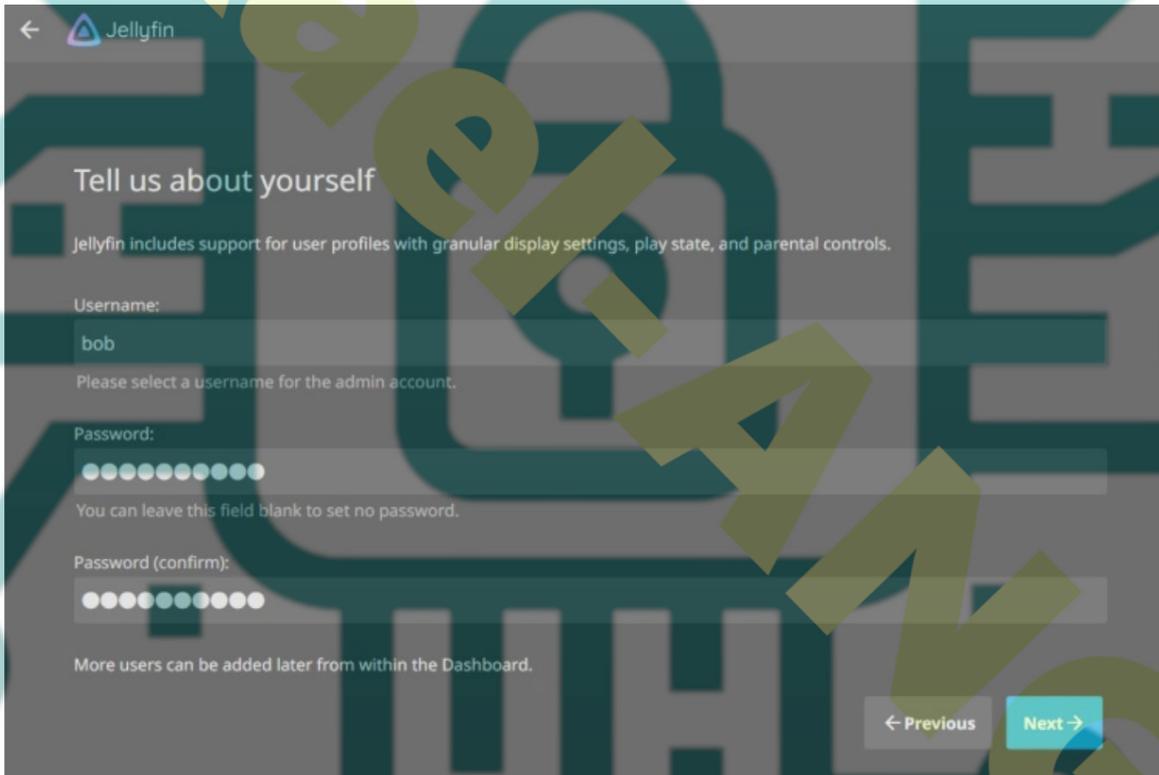
À ce stade, toutes les dépendances sont configurées. Vous pouvez maintenant accéder à votre installation Jellyfin depuis votre navigateur Web.

Visitez le nom de domaine de votre serveur Jellyfin (c'est-à-dire : <https://media.hwdomain.io/>) à l'aide de votre navigateur Web. Vous verrez le message de bienvenue de Jellyfin.

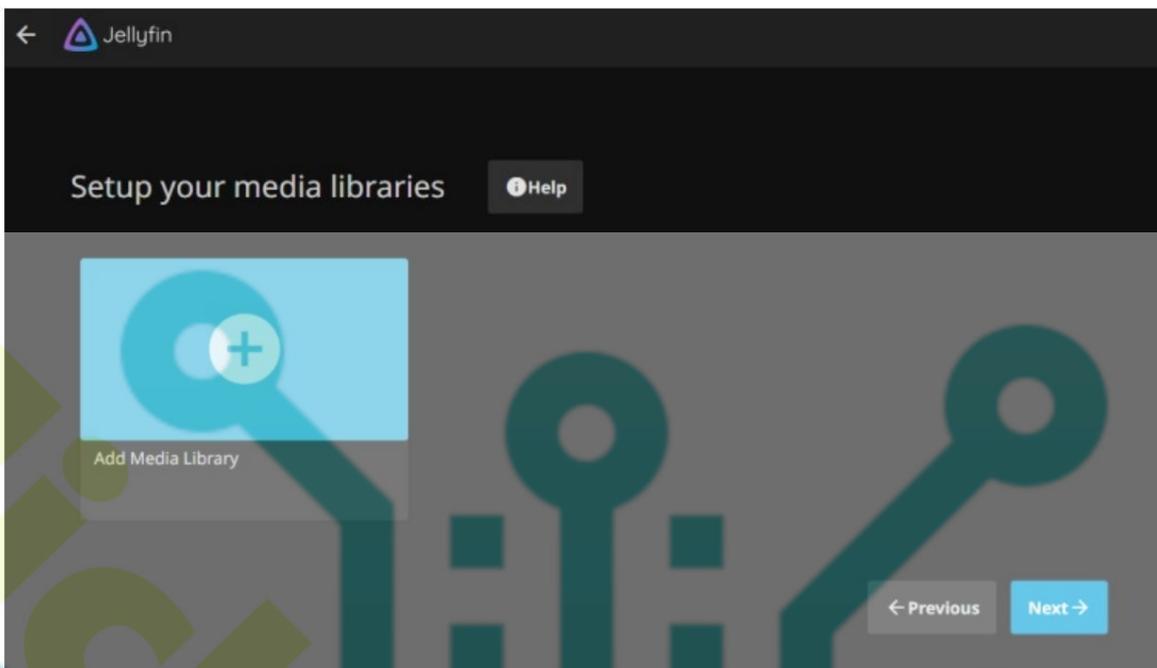
Cliquer sur Suivant pour continuer.



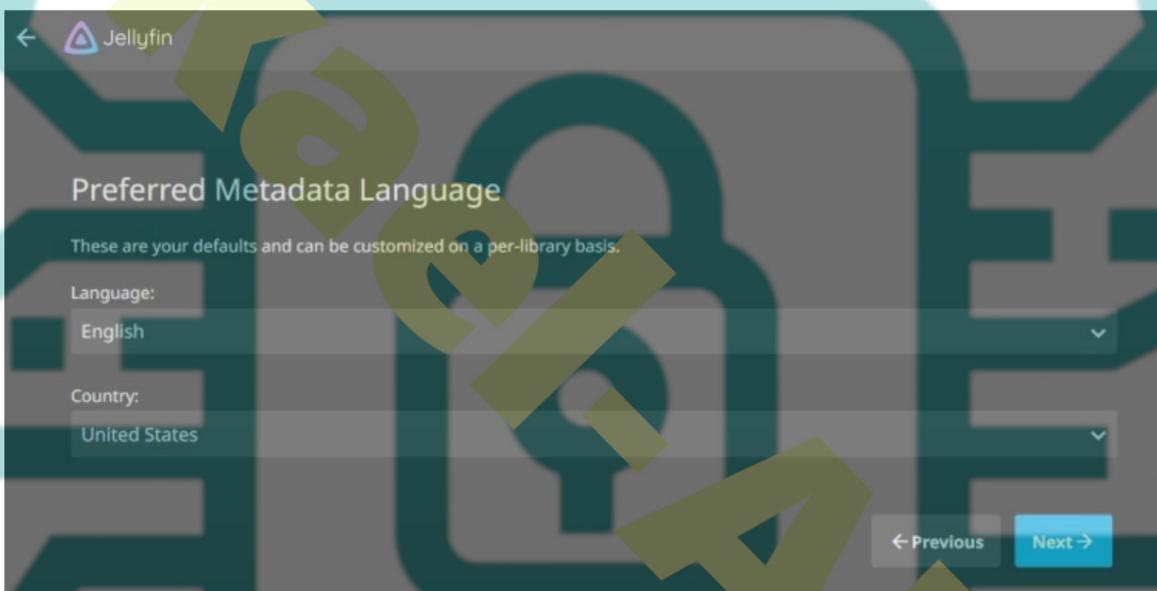
Créez maintenant votre premier utilisateur sur Jellyfin, puis cliquez sur Suivant.



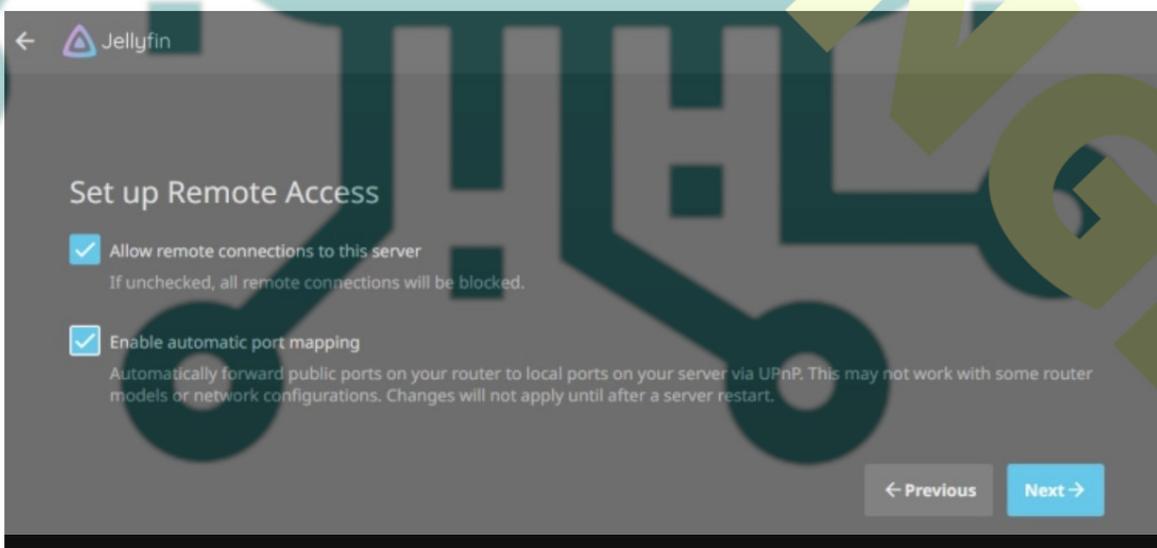
Il vous sera maintenant demandé de configurer une bibliothèque multimédia pour Jellyfin, cliquez sur Suivant pour l'ignorer. Vous pourrez le configurer plus tard, une fois la configuration terminée.



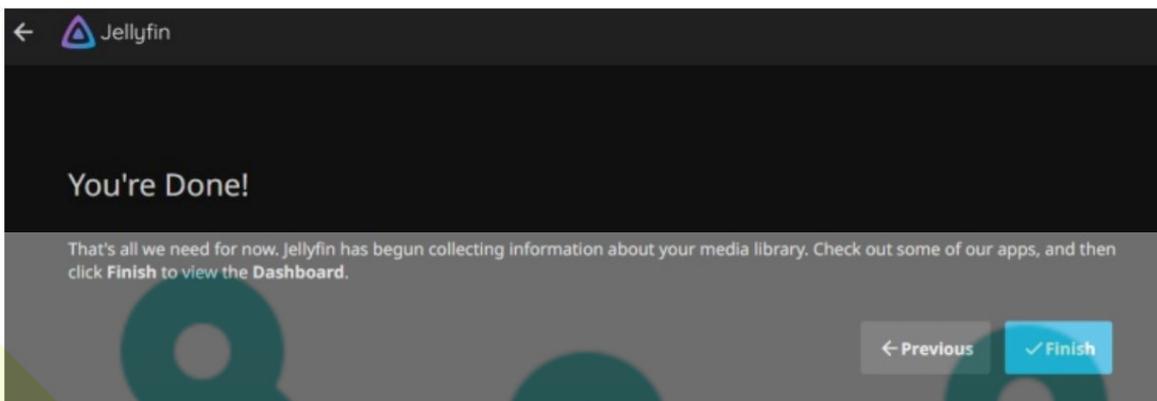
Sélectionnez votre langue de métadonnées préférée et cliquez à nouveau sur Suivant .



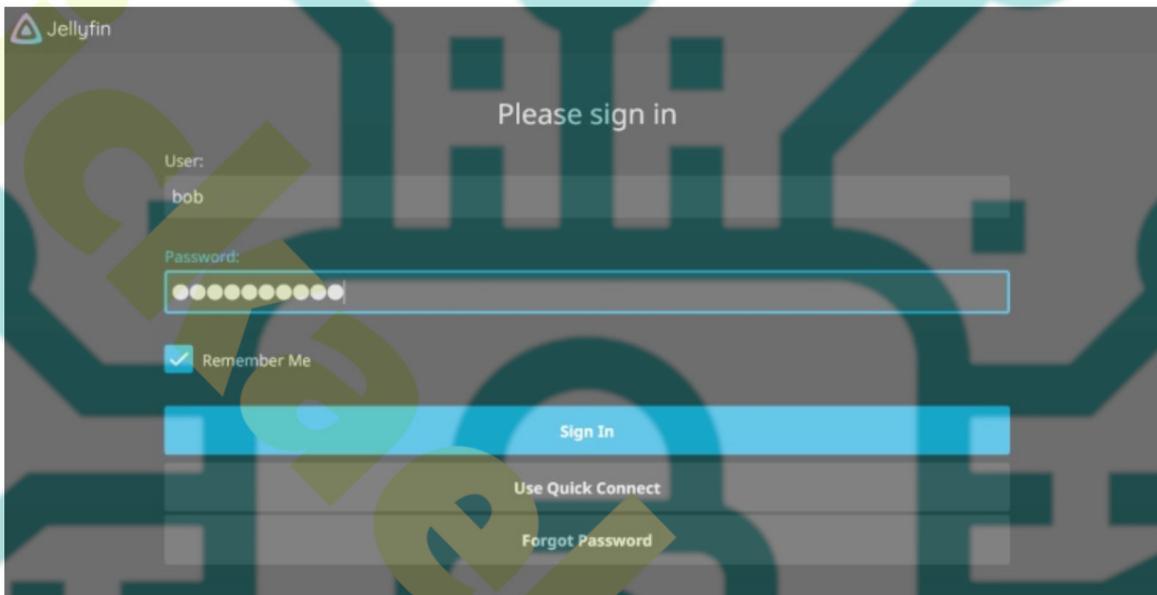
Pour la section Accès à distance, cochez les cases Autoriser la connexion à distance et Mappage automatique des ports. Ensuite, cliquez sur Suivant pour confirmer.



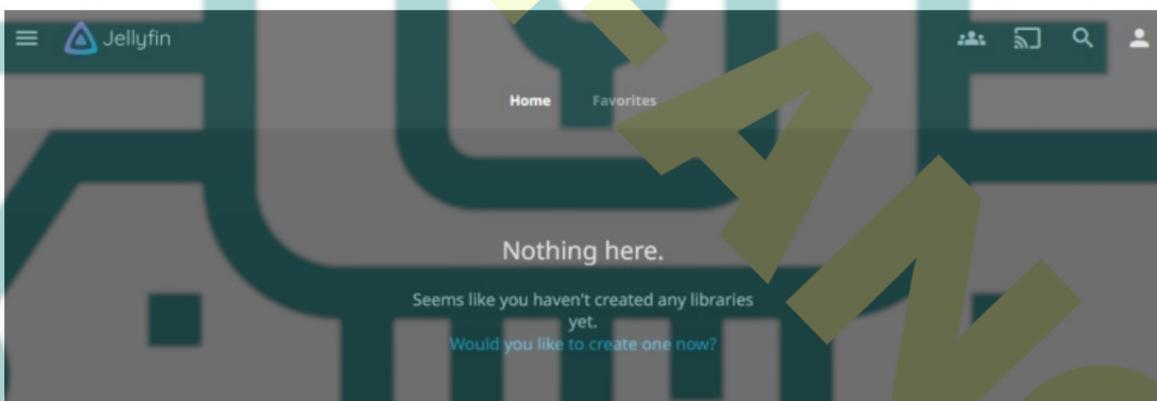
Enfin, cliquez sur Terminer pour terminer l'installation de Jellyfin.



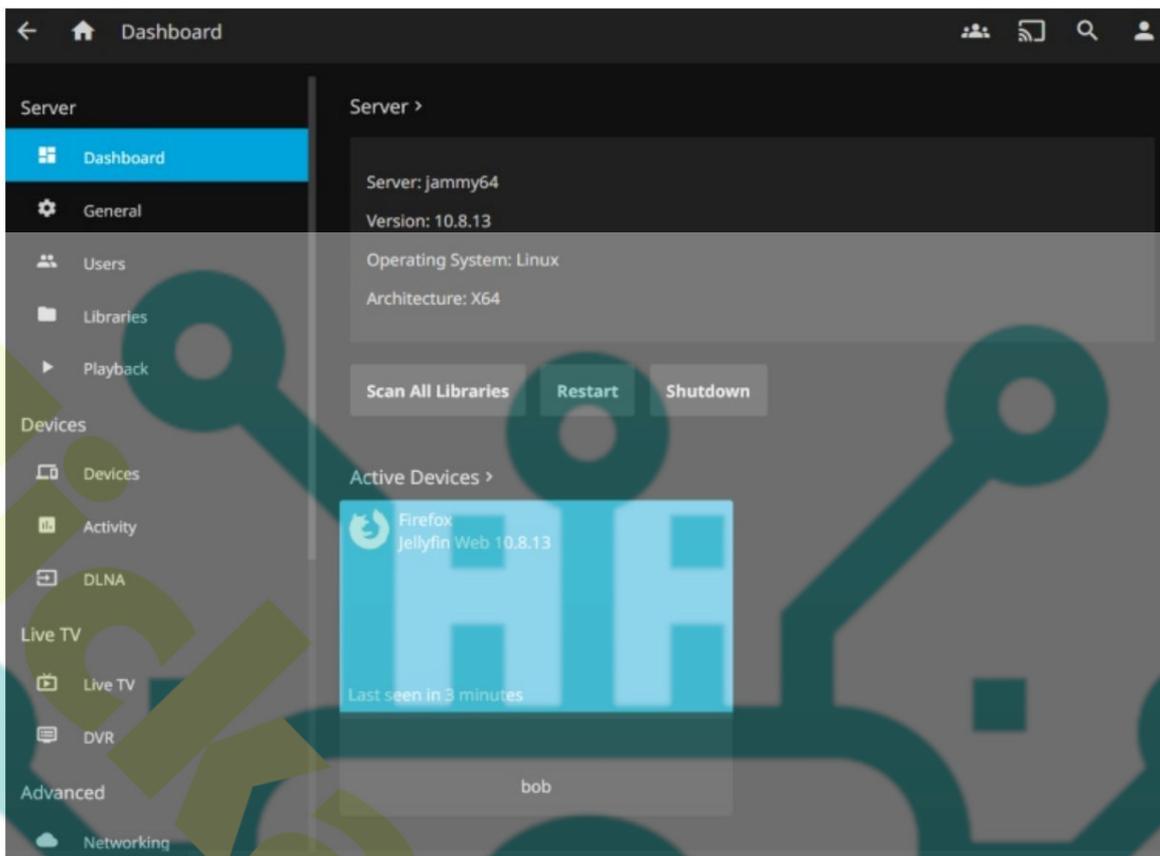
Et vous serez redirigé vers la page de connexion de Jellyfin. Saisissez votre nom d'utilisateur et votre mot de passe, puis cliquez sur Se connecter.



Vous verrez maintenant le tableau de bord Jellyfin comme suit :



Si vous cliquez sur le menu Administration, vous verrez des informations détaillées sur votre serveur Jellyfin. Dans cet exemple, vous pouvez voir que Jellyfin xxx est installé sur le système Ubuntu.



Conclusion

Toutes nos félicitations! Vous avez maintenant installé avec succès Jellyfin Media Server sur le serveur Ubuntu 22.04. Vous avez également sécurisé Jellyfin avec Apache en tant que proxy inverse et SSL/TLS de Letsencrypt. Désormais, vous pouvez ajouter une bibliothèque de fichiers multimédias et stocker votre collection de bibliothèques sur Jellyfin.
