

Documentation de Déploiement - Backend et Frontend Web StreamIO

1. Introduction

1.1 Objectif

Ce document explique comment déployer le **backend** et le **frontend web** de la plateforme StreamIO sur un serveur Ubuntu sans interface graphique. Le backend gère les fonctionnalités (authentification, gestion des offres), et le frontend web fournit l'interface utilisateur du site . Le déploiement inclut :

- Configuration de la base de données MySQL (backend).
- Récupération du code backend et frontend depuis GitHub.
- Configuration de Nginx pour servir le frontend web.
- Connexion entre frontend et backend.
- Services pour un démarrage automatique.

1.2 Prérequis

- **Serveur** : PC avec Ubuntu Server 20.04/22.04 LTS, accès root/sudo, connexion FTTH (Orange/SFR).
- **GitHub** : Accès aux dépôts backend et frontend web.
- **Outils** : `git`, `mysql-server`, `nginx`, `nodejs`.
- Backend déjà fonctionnel (API accessible, ex. : `http://localhost:3000`).

2. Déploiement du backend (récapitulatif)

2.1 Préparation

Mettez à jour Ubuntu :

```
sudo apt update
```

```
sudo apt upgrade -y
```

1.

Installez **git** et **curl** :

```
sudo apt install -y git curl
```

2.

2.2 Base de données (MySQL)

Installez MySQL :

```
sudo apt install -y mysql-server
```

```
sudo mysql_secure_installation
```

1.

- Définissez un mot de passe root, supprimez les utilisateurs anonymes, désactivez l'accès root distant.

Créez la base **streamio** et l'utilisateur :

```
sudo mysql -u root -p
```

```
CREATE DATABASE streamio;
```

```
CREATE USER 'streamio_user'@'localhost' IDENTIFIED BY  
'MotDePasseSecurise';
```

```
GRANT ALL PRIVILEGES ON streamio.* TO  
'streamio_user'@'localhost';
```

FLUSH PRIVILEGES;

EXIT;

2.

Importez le schéma ([schema.sql](#) depuis GitHub) :

```
mysql -u streamio_user -p streamio < schema.sql
```

3.

- Tables : [users](#) (utilisateurs), [offers](#) (offres), [messages](#) (discussions).

2.3 Code backend

Clonez le dépôt (ex. : Node.js) :

```
sudo mkdir /opt/streamio-backend
```

```
sudo chown $USER:$USER /opt/streamio-backend
```

```
cd /opt/streamio-backend
```

```
git clone
```

```
https://github.com/votre\_utilisateur/streamio-backend.git .
```

1.

Installez Node.js et dépendances :

```
curl -fsSL https://deb.nodesource.com/setup\_18.x | sudo -E  
bash -
```

```
sudo apt install -y nodejs
```

```
npm install
```

2.

Configurez `.env` :

`nano .env`

`DB_HOST=localhost`

`DB_USER=streamio_user`

`DB_PASSWORD=MotDePasseSecurise`

`DB_NAME=streamio`

`PORT=3000`

3.

2.4 Service systemd (backend)

Créez

`/etc/systemd/system/streamio-backend.service` :

`sudo nano /etc/systemd/system/streamio-backend.service`

[Unit]

`Description=Backend StreamIO`

`After=network.target mysql.service`

[Service]

`ExecStart=/usr/bin/node /opt/streamio-backend/index.js`

`WorkingDirectory=/opt/streamio-backend`

Restart=always

User=streamio_user

Environment=NODE_ENV=production

[Install]

WantedBy=multi-user.target

1.

Configurez l'utilisateur et activez :

```
sudo adduser --system --no-create-home streamio_user
```

```
sudo chown -R streamio_user:streamio_user  
/opt/streamio-backend
```

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable streamio-backend
```

```
sudo systemctl start streamio-backend
```

2.

3. Déploiement du frontend web

3.1 Préparation

Installez Nginx pour servir les fichiers statiques :

```
sudo apt install -y nginx
```

1.

Vérifiez que Nginx fonctionne :
`sudo systemctl status nginx`

2.

3.2 Récupération du code frontend

Note : Supposons un frontend React (ajustable si Vue.js, Angular, etc.).

Créez un dossier pour le frontend :
`sudo mkdir /opt/streamio-frontend`

`sudo chown $USER:$USER /opt/streamio-frontend`

1.

Clonez le dépôt GitHub (remplacez par l'URL réelle) :
`cd /opt/streamio-frontend`

`git clone`
`https://github.com/votre_utilisateur/streamio-frontend.git .`

2.

Installez Node.js pour compiler (si non installé) :
`curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E`
`bash -`

`sudo apt install -y nodejs`

3.

Installez les dépendances et compilez :
`npm install`

npm run build

4.

- Cela génère un dossier **build/** avec les fichiers statiques (HTML, CSS, JS).

3.3 Configuration de Nginx

Créez un fichier de configuration Nginx :

```
sudo nano /etc/nginx/sites-available/streamio
```

1.

Ajoutez le contenu suivant :

```
server {
```

```
    listen 80;
```

```
    server_name www.streamio.ovh streamio.ovh;
```

```
    root /opt/streamio-frontend/build;
```

```
    index index.html;
```

```
    location / {
```

```
        try_files $uri $uri/ /index.html;
```

```
    }
```

```
location /api/ {  
  
    proxy_pass http://localhost:3000;  
  
    proxy_set_header Host $host;  
  
    proxy_set_header X-Real-IP $remote_addr;  
  
    proxy_set_header X-Forwarded-For  
$proxy_add_x_forwarded_for;  
  
}  
}
```

2.

- **root** : Pointe vers les fichiers compilés du frontend.
- **location /api/** : Redirige les requêtes API vers le backend (port 3000).

Activez la configuration :

```
sudo ln -s /etc/nginx/sites-available/streamio  
/etc/nginx/sites-enabled/
```

```
sudo nginx -t
```

```
sudo systemctl restart nginx
```

3.

Ajustez les permissions :

```
sudo chown -R www-data:www-data /opt/streamio-frontend
```

```
sudo chmod -R 755 /opt/streamio-frontend
```


4.

3.4 (Optionnel) Service systemd pour serveur de développement

Note : Nginx sert les fichiers statiques, donc un service n'est pas nécessaire pour la production. Cependant, pour un serveur de développement React (ex. : `npm start`), voici les étapes.

Créez

`/etc/systemd/system/streamio-frontend.service :`

`sudo nano /etc/systemd/system/streamio-frontend.service`

[Unit]

Description=Frontend StreamIO (développement)

After=network.target streamio-backend.service

[Service]

ExecStart=/usr/bin/npm start

WorkingDirectory=/opt/streamio-frontend

Restart=always

User=streamio_user

Environment=NODE_ENV=development PORT=3001

[Install]

WantedBy=multi-user.target

1.

Activez le service :

sudo systemctl daemon-reload

sudo systemctl enable streamio-frontend

sudo systemctl start streamio-frontend

2.

- Le frontend est alors accessible sur <http://localhost:3001> (pour développement).

4. Vérifications finales

Backend : Testez l'API :

curl http://localhost:3000

1.

2. **Frontend** : Accédez au site via un navigateur (depuis un autre PC) :

- URL : http://adresse_ip_serveur ou <http://www.streamio.ovh> (si DNS configuré).
- Vérifiez que l'interface charge et que les appels API fonctionnent (ex. : connexion).

Nginx : Consultez les logs si erreur :

sudo tail -f /var/log/nginx/error.log

3.

Redémarrage : Redémarrez le serveur et vérifiez :

sudo reboot

```
sudo systemctl status streamio-backend
```

```
sudo systemctl status nginx
```

4.

5. Résolution des problèmes

Nginx ne démarre pas :

```
sudo nginx -t
```

```
sudo systemctl restart nginx
```

```
journalctl -u nginx -b
```

-
- **Frontend ne charge pas :**
 - Vérifiez les permissions : `ls -l /opt/streamio-frontend/build.`
 - Vérifiez la configuration Nginx : `/etc/nginx/sites-available/streamio.`
- **Requêtes API échouent :**
 - Vérifiez que le backend fonctionne : `curl http://localhost:3000.`
 - Vérifiez le proxy : `location /api/` dans la configuration Nginx.

Code non récupéré : Vérifiez l'URL GitHub et l'accès réseau :

```
git ls-remote
```

```
https://github.com/votre_utilisateur/streamio-frontend.git
```

-

6. Maintenance

Mise à jour du frontend :

```
cd /opt/streamio-frontend
```

```
git pull
```

```
npm install
```

```
npm run build
```

```
sudo systemctl restart nginx
```

-

Sauvegardes : Sauvegardez la base de données (backend) :

```
mysqldump -u streamio_user -p streamio > backup.sql
```

-

Logs : Surveillez Nginx et le backend :

```
sudo tail -f /var/log/nginx/access.log
```

```
journalctl -u streamio-backend
```