

# Serveur Simple en Temps Réel (SRS) pour le Streaming RTMP

Plongeons dans l'utilisation de Simple Realtime Server (SRS) pour déployer un serveur de bord pour le streaming RTMP, en mettant l'accent sur la compréhension et la configuration efficace, y compris le fichier `rtmp.conf`. SRS est un serveur multimédia puissant et haute performance conçu pour les protocoles de streaming en temps réel comme RTMP, HLS et HTTP-FLV. Il est particulièrement bien adapté aux scénarios nécessitant une évolutivité, comme les réseaux de distribution de contenu (CDN), où un serveur de bord peut distribuer efficacement les flux à de nombreux clients tout en récupérant le contenu auprès d'un serveur d'origine. Je vais vous guider étape par étape, expliquer le concept de déploiement de bord et clarifier comment travailler avec des configurations comme `rtmp.conf`.

## Qu'est-ce que SRS et le déploiement de bord ?

SRS est un serveur open-source conçu pour gérer le streaming multimédia en temps réel avec un accent sur la simplicité et l'efficacité. Il prend en charge RTMP (Real-Time Messaging Protocol) pour le streaming en direct à faible latence, ainsi que d'autres protocoles comme HLS et WebRTC. Dans SRS, un serveur de "bord" agit comme intermédiaire entre les clients (spectateurs ou diffuseurs) et un serveur d'"origine" (d'où provient le flux). Le bord récupère les flux auprès de l'origine uniquement lorsqu'ils sont demandés par les clients, réduisant ainsi la charge sur l'origine et permettant une distribution évolutive—pensez-y comme à une couche de mise en cache de CDN adaptée pour les flux en direct.

Déployer un serveur de bord avec SRS est idéal lorsque vous avez besoin de : - Servir un grand nombre de spectateurs sans submerger l'origine. - Permettre aux diffuseurs de pousser des flux vers le bord, qui les transfère ensuite vers l'origine. - Minimiser l'utilisation de la bande passante sur les serveurs d'origine coûteux en utilisant des nœuds de bord moins chers.

## Étape par étape : Déployer un serveur de bord avec SRS pour RTMP

Voici comment configurer SRS en tant que serveur de bord pour le streaming RTMP. Je suppose que vous travaillez sur un système Linux (par exemple, Ubuntu), car SRS est optimisé pour ces environnements.

**1. Installer SRS** Tout d'abord, vous devez faire fonctionner SRS sur votre machine : - **Télécharger SRS** : Obtenez la dernière version stable à partir du dépôt GitHub officiel ([github.com/ossrs/srs](https://github.com/ossrs/srs)). À ce jour, le 26 février 2025, vous cloneriez généralement le dépôt : 

```
git clone https://github.com/ossrs/srs.git
```

`cd srs` - **Compiler SRS** : SRS utilise un processus de compilation simple avec `./configure` et `make` : `./configure` `make` Cela compile le serveur dans le répertoire `objs` (par exemple, `objs/srs`). - **Tester le binaire** : Exécutez-le avec la configuration par défaut pour vous assurer qu'il fonctionne : `./objs/srs -c conf/srs.conf` Par défaut, il écoute sur le port 1935 pour RTMP. Vérifiez la sortie de la console pour confirmation.

**2. Comprendre le concept de bord** Dans SRS, un serveur de bord fonctionne en mode “remote”, ce qui signifie qu’il ne génère pas de flux lui-même mais les récupère auprès d’un serveur d’origine lorsqu’un client les demande (pour la lecture) ou pousse des flux vers l’origine (pour la diffusion). Cette récupération à la demande est ce qui rend les serveurs de bord efficaces pour la mise à l’échelle de la distribution RTMP.

- **Serveur d’origine** : La source du flux (par exemple, là où un encodeur comme OBS pousse un flux RTMP).
- **Serveur de bord** : Un relais auquel les clients se connectent, récupérant auprès de l’origine uniquement lorsque cela est nécessaire.

Pour cet exemple, supposons que vous avez déjà un serveur d’origine exécutant SRS à 192.168.1.100:1935 (remplacez ceci par votre adresse IP d’origine réelle).

**3. Configurer le serveur de bord** SRS utilise des fichiers de configuration pour définir son comportement. Le `srs.conf` par défaut est un bon point de départ, mais pour le déploiement de bord, vous créez une configuration spécifique—appelons-la `edge.conf`. Voici comment la configurer :

- **Créer `edge.conf`** :

```
cd conf
nano edge.conf
```

- **Ajouter la configuration de bord** : Voici un `edge.conf` minimal pour le déploiement de bord RTMP :

```
listen          1935;
max_connections 1000;
srs_log_tank    file;
srs_log_file    ./objs/edge.log;
vhost __defaultVhost__ {
    cluster {
        mode      remote;
        origin    192.168.1.100:1935;
    }
}
```

- `listen 1935` : Le bord écoute les connexions RTMP sur le port 1935.
- `max_connections 1000` : Limite les connexions simultanées (ajustez en fonction de la capacité de votre serveur).
- `srs_log_file` : Journaux dans un fichier pour le débogage.
- `vhost __defaultVhost__` : La configuration de l’hôte virtuel par défaut.
- `cluster { mode remote; origin 192.168.1.100:1935; }` : Définit ce serveur comme un bord (`mode remote`) et le pointe vers le serveur d’origine.

- **Enregistrer et quitter** : Ctrl+O, Entrée, Ctrl+X dans nano.

#### 4. Démarrer le serveur de bord

Exécutez SRS avec votre configuration de bord :

```
./objs/srs -c conf/edge.conf
```

Vérifiez les journaux (`./objs/edge.log`) pour confirmer qu'il fonctionne et est connecté à l'origine.

#### 5. Tester la configuration

- **Publier un flux** : Utilisez un outil comme OBS ou FFmpeg pour pousser un flux RTMP vers le serveur d'origine :

```
ffmpeg -re -i input.mp4 -c copy -f flv rtmp://192.168.1.100/live/livestream
```

Ici, `live` est le nom de l'application et `livestream` est la clé de flux.

- **Lire depuis le bord** : Utilisez VLC ou un autre client RTMP pour lire le flux depuis le bord :

```
rtmp://<edge-server-ip>/live/livestream
```

Remplacez `<edge-server-ip>` par l'IP de votre serveur de bord (par exemple, `192.168.1.101`). Le bord récupérera le flux auprès de l'origine et le servira.

**6. Explorer `rtmp.conf`** SRS ne fournit pas de fichier `rtmp.conf` par défaut, mais vous pourriez rencontrer des références à celui-ci dans des tutoriels ou des configurations personnalisées. Il s'agit essentiellement d'une convention de nommage pour un fichier de configuration spécifique à RTMP. Par exemple, la documentation SRS ([ossrs.net](http://ossrs.net)) fournit un exemple de `rtmp.conf` pour le streaming RTMP en temps réel :

```
listen          1935;
max_connections 1000;
vhost __defaultVhost__ {
    tcp_nodelay  on;
    min_latency  on;
    play {
        gop_cache  off;
        queue_length 10;
    }
    publish {
        mr          off;
    }
}
```

- **But** : Cette configuration optimise pour le streaming RTMP à faible latence sur un serveur d'origine, pas un bord. Pour le déploiement de bord, vous l'adapteriez en ajoutant le bloc `cluster` de l'étape 3.

- **Paramètres clés :**

- `tcp_nodelay on` : Réduit la latence en désactivant l’algorithme de Nagle.
- `min_latency on` : Priorise la faible latence sur la mise en mémoire tampon.
- `gop_cache off` : Désactive la mise en cache du Groupe d’Images pour la lecture en temps réel.
- `mr off` : Désactive “merge read” pour éviter les retards dans la publication.

Pour un bord, vous combineriez cela avec les paramètres `cluster` plutôt que de l’utiliser seul.

## Expliquer plus : Mécanismes de bord et RTMP

- **Comment fonctionne le bord** : Lorsqu’un client demande `rtmp://<edge-ip>/live/livestream`, le bord vérifie s’il dispose du flux. Si ce n’est pas le cas, il le récupère auprès de l’origine (`192.168.1.100:1935`) et le met en cache localement pour servir d’autres clients. Si un diffuseur pousse vers le bord, il transfère le flux vers l’origine.
- **Spécificités RTMP** : RTMP est un protocole à faible latence idéal pour le streaming en direct. SRS gère RTMP efficacement, prenant en charge des fonctionnalités comme le codage temporel absolu (ATC) pour la synchronisation entre serveurs, bien que cela soit désactivé par défaut en mode bord sauf spécification.
- **Évolutivité** : Ajoutez plusieurs bords pointant vers la même origine pour gérer des milliers de clients. SRS prend en charge la bascule en listant plusieurs origines (par exemple, `origin 192.168.1.100:1935 192.168.1.200:1935;`).

## Conseils et dépannage

- **Pare-feu** : Assurez-vous que le port 1935 est ouvert sur les serveurs d’origine et de bord.
- **Journaux** : Vérifiez `edge.log` pour des erreurs comme des échecs de connexion à l’origine.
- **Latence** : Le bord ajoute une latence minimale (généralement <1s) si l’origine est également à faible latence.
- **Multiplés bords** : Déployez des bords supplémentaires avec la même configuration, ajustant les ports `listen` ou les IPs si nécessaire.

## Conclusion

Déployer un serveur de bord SRS pour RTMP est simple une fois que vous comprenez la relation origine-bord. Le `edge.conf` configure le bord pour récupérer ou pousser des flux dynamiquement, tandis qu’une configuration de type `rtmp.conf` pourrait affiner les performances RTMP si nécessaire. Avec cette configuration, vous êtes prêt à évoluer efficacement le streaming en direct—que ce soit pour quelques spectateurs ou un public mondial. Vous souhaitez l’ajuster davantage ou intégrer HLS en plus de RTMP ? Faites-le moi savoir !