

# OBS、SRS、および FFmpeg を使用したライブストリーミング設定

このブログ記事は、ChatGPT-4o の協力を得て整理されました。

---

ライブストリーミングは、専門的な放送から個人のビデオブログまで、幅広い用途でオンラインコミュニケーションの重要な要素となっています。強力なライブストリーミングソリューションを構築するには、さまざまなツールやプロトコルを理解する必要があります。このガイドでは、OBS、SRS、FFmpeg を使用してライブストリーミングを設定する手順を段階的に説明します。

## ライブストリーミングの主要コンポーネント

- 1. OBS (Open Broadcaster Software)** OBS は、ビデオ録画とライブストリーミングのための強力なオープンソースソフトウェアです。リアルタイムのソースとデバイスキャプチャ、シーン合成、エンコード、録画、および放送機能を提供します。
- 2. SRS (Simple Realtime Server)** SRS は高性能な RTMP、HLS、HTTP-FLV ストリーミングサーバーです。大量の同時接続をサポートし、高度に設定可能です。
- 3. FFmpeg** FFmpeg は包括的なマルチメディアフレームワークで、デコード、エンコード、トランスコード、多重化、逆多重化、ストリーミング、フィルタリング、再生など、人間や機械が作成したほぼすべてのコンテンツを扱うことができます。ストリーミング設定で広く使用されており、その多機能性と信頼性から高く評価されています。

## ライブストリーミング環境のセットアップ

### OBS 設定

- 1. OBS のインストール:** 公式サイトから OBS をダウンロードしてインストールします。
- 2. 設定の構成:** OBS を開き、設定 > ストリームに移動し、ストリームタイプをカスタム... に設定します。ストリーミングサーバーの URL (例: `rtmp://your_server_ip/live`) を入力します。
- 3. ソースの追加:** OBS でビデオとオーディオのソースを追加してシーンを作成します。これは、画面キャプチャ、カメラ、画像、テキストなどが含まれます。

## SRS サーバー設定

1. **SRS のインストール**: GitHub から SRS リポジトリをクローンし、SSL をサポートするようにコンパイルします。  
sh git clone https://github.com/ossrs/srs.git cd srs/trunk  
. ./configure --disable-all --with-ssl make
2. **SRS の設定**: conf/rtmp.conf ファイルを編集して、RTMP 設定を構成します。  
sh listen 1935; max\_connections 1000; vhost \_\_defaultVhost\_\_ { }
3. **SRS の起動**: 設定ファイルを使用して SRS サーバーを実行します。  
sh ./objs/srs -c conf/rtmp.conf

## FFmpeg を使用したストリーミング

1. **FFmpeg のインストール**: 公式サイトから、またはパッケージマネージャーを使用して FFmpeg をインストールします。
2. **FFmpeg を使用したストリーミング**: FFmpeg を使用してビデオストリームを SRS サーバーにプッシュします。  
sh ffmpeg -re -i input\_video.flv -vcodec copy -acodec copy -f flv rtmp://your\_server\_ip/live/stream\_key
3. **ストリーミングの自動化**: ビデオファイルを継続的にストリーミングするためのスクリプトを作成します。  
sh for ((;)); do ffmpeg -re -i input\_video.flv -vcodec copy -acodec copy -f flv rtmp://your\_server\_ip/live/stream\_key; sleep 1; done

## プロトコルとフォーマット

**RTMP (リアルタイムメッセージングプロトコル)** - RTMP は、低遅延で信頼性の高い伝送が可能なため、ライブストリーミングで広く使用されています。- TCP を使用し、持続的な接続を維持することで、スムーズなストリーミングを保証します。

**HLS (HTTP ライブストリーミング)** - HLS は、ビデオストリームを小さな HTTP ベースのファイルセグメントに分割し、標準的な Web サーバーを通じて簡単に配信できるようにします。- 遅延が発生する可能性がありますが、さまざまなデバイスやプラットフォームとの互換性が非常に高いです。

**HTTP-FLV** - FLV 形式を HTTP 伝送と組み合わせ、低遅延のストリーミング配信に使用されます。- 既存の HTTP インフラを活用するため、ブラウザベースのストリーミングに適しています。

## 実際の応用

**iOS および Android ストリーミング** - VideoCore や Ijkplayer などのライブラリを使用して、モバイルデバイス上で RTMP ストリーミングを実現します。- FFmpeg を統合してエンコードおよびデコードタスクを実行し、互換性とパフォーマンスを向上させます。

**Web ベースのストリーミング** - HTML5 のビデオ要素を使用して、ウェブページ上でビデオ再生を実現し、HLS または HTTP-FLV をサポートします。- WebRTC を活用して、リアルタイム通信と低遅延のインタラクションを実現します。

## ツールとリソース

- **VLC** : RTMP、HLS などのストリーミングプロトコルをサポートする多機能メディアプレーヤー。
- **SRS Player** : SRS ストリームをテストするためのオンラインプレーヤー。
- **FFmpeg ドキュメント** : さまざまなマルチメディアタスクの詳細なドキュメントを提供します。

## 結論

信頼性の高いライブストリーミングソリューションを構築するには、複数のツールやプロトコルを理解し、設定する必要があります。OBS、SRS、そして FFmpeg は強力なコンポーネントであり、これらを組み合わせることで強力なストリーミング環境を構築できます。iOS、Android、Web のいずれを対象とする場合でも、これらのツールは高品質なライブストリーミングを実現するための柔軟性とパフォーマンスを提供します。

より詳細な情報や高度な設定については、各ツールの公式ドキュメントを参照し、コミュニティフォーラムで他のヒントやサポートを探求してください。ライブストリーミングが順調でありますように！