

Servidor Simple en Tiempo Real (SRS) para Transmisión RTMP

Vamos a sumergirnos en el uso de Simple Realtime Server (SRS) para desplegar un servidor de borde para la transmisión RTMP, con un enfoque en entenderlo y configurarlo de manera efectiva, incluyendo el archivo `rtmp.conf`. SRS es un servidor de medios potente y de alto rendimiento diseñado para protocolos de transmisión en tiempo real como RTMP, HLS y HTTP-FLV. Es particularmente adecuado para escenarios que requieren escalabilidad, como redes de entrega de contenido (CDN), donde un servidor de borde puede distribuir eficientemente flujos a muchos clientes mientras recupera contenido de un servidor de origen. Te guiaré a través del proceso paso a paso, explicaré el concepto de despliegue de borde y aclararé cómo trabajar con configuraciones como `rtmp.conf`.

¿Qué es SRS y el Despliegue de Borde?

SRS es un servidor de código abierto construido para manejar la transmisión de medios en tiempo real con un enfoque en la simplicidad y la eficiencia. Soporta RTMP (Real-Time Messaging Protocol) para la transmisión en vivo de baja latencia, junto con otros protocolos como HLS y WebRTC. En SRS, un servidor de “borde” actúa como intermediario entre los clientes (espectadores o editores) y un servidor de “origen” (donde se origina el flujo). El borde recupera flujos del origen solo cuando se solicitan por los clientes, reduciendo la carga en el origen y habilitando la distribución escalable—piensa en ello como una capa de caché de CDN adaptada para flujos en vivo.

Desplegar un servidor de borde con SRS es ideal cuando necesitas:

- Servir a un gran número de espectadores sin abrumar al origen.
- Permitir que los editores envíen flujos al borde, que luego los reenvía al origen.
- Minimizar el uso de ancho de banda en costosos servidores de origen aprovechando nodos de borde más económicos.

Paso a Paso: Desplegando un Servidor de Borde con SRS para RTMP

Aquí te muestro cómo configurar SRS como un servidor de borde para la transmisión RTMP. Supongo que estás trabajando en un sistema Linux (por ejemplo, Ubuntu), ya que SRS está optimizado para tales entornos.

1. Instalar SRS Primero, necesitas ejecutar SRS en tu máquina:

- **Descargar SRS:** Obtén la última versión estable desde el repositorio oficial de GitHub (github.com/ossrs/srs). Hasta la fecha, 26 de febrero de 2025, generalmente clonarías el repositorio: `git clone https://github.com/ossrs/srs.git` `cd srs`
- **Compilar SRS:** SRS utiliza un proceso de compilación sencillo con `./configure` y `make`: `./configure` `make` Esto compila el servidor en el directorio `objs` (por ejemplo, `objs/srs`).
- **Probar el Binario:** Ejecútalo con la configuración predeterminada para asegurarte de que funcione: `./objs/srs -c conf/srs.conf` Por defecto, escucha en el puerto 1935 para RTMP. Verifica la salida de la consola para la confirmación.

2. Entender el Concepto de Borde En SRS, un servidor de borde opera en “modo remoto”, lo que significa que no genera flujos por sí mismo, sino que los recupera de un servidor de origen cuando un cliente los solicita (para la reproducción) o envía flujos al origen (para la publicación). Esta recuperación bajo demanda es lo que hace que los servidores de borde sean eficientes para escalar la entrega de RTMP.

- **Servidor de Origen:** La fuente del flujo (por ejemplo, donde un codificador como OBS envía un flujo RTMP).
- **Servidor de Borde:** Un relay al que se conectan los clientes, recuperando del origen solo cuando sea necesario.

Para este ejemplo, supongamos que ya tienes un servidor de origen ejecutando SRS en 192.168.1.100:1935 (reemplaza esto con tu IP de origen real).

3. Configurar el Servidor de Borde SRS utiliza archivos de configuración para definir su comportamiento. La `srs.conf` predeterminada es un buen punto de partida, pero para el despliegue de borde, crearás una configuración específica—llamémosla `edge.conf`. Aquí te muestro cómo configurarla:

- **Crear `edge.conf`:**

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

- **Agregar Configuración de Borde:** Aquí tienes una `edge.conf` mínima para el despliegue de borde 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`: El borde escucha conexiones RTMP en el puerto 1935.
- `max_connections 1000`: Limita las conexiones concurrentes (ajusta según la capacidad de tu servidor).
- `srs_log_file`: Registra en un archivo para la depuración.
- `vhost __defaultVhost__`: La configuración predeterminada del host virtual.

- cluster { mode remote; origin 192.168.1.100:1935; }: Configura este servidor como un borde (mode remote) y lo apunta al servidor de origen.

- **Guardar y Salir:** Ctrl+O, Enter, Ctrl+X en nano.

4. Iniciar el Servidor de Borde

Ejecuta SRS con tu configuración de borde:

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

Verifica los registros (./objs/edge.log) para confirmar que está en funcionamiento y conectado al origen.

5. Probar la Configuración

- **Publicar un Flujo:** Usa una herramienta como OBS o FFmpeg para enviar un flujo RTMP al servidor de origen:

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

Aquí, live es el nombre de la aplicación y livestream es la clave del flujo.

- **Reproducir desde el Borde:** Usa VLC u otro cliente RTMP para reproducir el flujo desde el borde:

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

Reemplaza <edge-server-ip> con la IP de tu servidor de borde (por ejemplo, 192.168.1.101). El borde recuperará el flujo del origen y te lo servirá.

6. Explorar rtmp.conf SRS no incluye un archivo rtmp.conf por defecto, pero podrías encontrar referencias a él en tutoriales o configuraciones personalizadas. Es esencialmente una convención de nombres para un archivo de configuración específico de RTMP. Por ejemplo, la documentación de SRS (ossrs.net) proporciona una muestra rtmp.conf para la transmisión RTMP en tiempo real:

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

```

        mr            off;
    }
}

```

- **Propósito:** Esta configuración optimiza para la transmisión RTMP de baja latencia en un servidor de origen, no en un borde. Para el despliegue de borde, adaptarías agregando el bloque `cluster` del paso 3.
- **Configuraciones Clave:**
 - `tcp_nodelay on`: Reduce la latencia deshabilitando el algoritmo de Nagle.
 - `min_latency on`: Prioriza la baja latencia sobre el almacenamiento en búfer.
 - `gop_cache off`: Deshabilita la caché de Group of Pictures para la reproducción en tiempo real.
 - `mr off`: Deshabilita “merge read” para evitar retrasos en la publicación.

Para un borde, combinarías esto con las configuraciones `cluster` en lugar de usarlo de manera independiente.

Explicando Más: Mecánica del Borde y RTMP

- **Cómo Funciona el Borde:** Cuando un cliente solicita `rtmp://<edge-ip>/live/livestream`, el borde verifica si tiene el flujo. Si no, lo recupera del origen (`192.168.1.100:1935`) y lo almacena en caché localmente para servir a otros clientes. Si un editor envía al borde, este reenvía el flujo al origen.
- **Especificaciones de RTMP:** RTMP es un protocolo de baja latencia ideal para la transmisión en vivo. SRS maneja RTMP de manera eficiente, soportando características como el codificación de tiempo absoluto (ATC) para la sincronización entre servidores, aunque está desactivado por defecto en el modo borde a menos que se especifique.
- **Escalabilidad:** Agrega múltiples bordes apuntando al mismo origen para manejar a miles de clientes. SRS soporta la recuperación automática listando múltiples orígenes (por ejemplo, `origin 192.168.1.100:1935 192.168.1.200:1935;`).

Consejos y Solución de Problemas

- **Firewall:** Asegúrate de que el puerto 1935 esté abierto en ambos servidores de origen y borde.
- **Registros:** Verifica `edge.log` para errores como fallos de conexión al origen.
- **Latencia:** El borde añade una latencia mínima (generalmente $<1s$) si el origen también es de baja latencia.
- **Múltiples Bordes:** Despliega bordes adicionales con la misma configuración, ajustando puertos o IPs según sea necesario.

Conclusión

Desplegar un servidor de borde SRS para RTMP es sencillo una vez que entiendes la relación origen-borde. La `edge.conf` configura el borde para recuperar o enviar flujos dinámicamente, mientras que una configuración estilo `rtmp.conf` podría ajustar el rendimiento de RTMP si es necesario. Con esta configuración, estás listo para escalar la transmisión en vivo de manera eficiente—ya sea para unos pocos espectadores o una audiencia global. ¿Quieres ajustarlo más o integrar HLS junto con RTMP? ¡Solo házmelo saber!