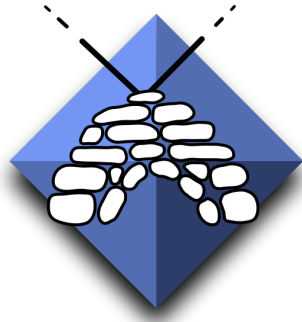
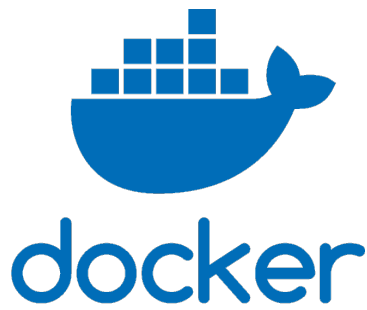


# Servidor de Streaming con Icecast2

José Luis Palencia Segura — ASTRA

31 de octubre de 2024

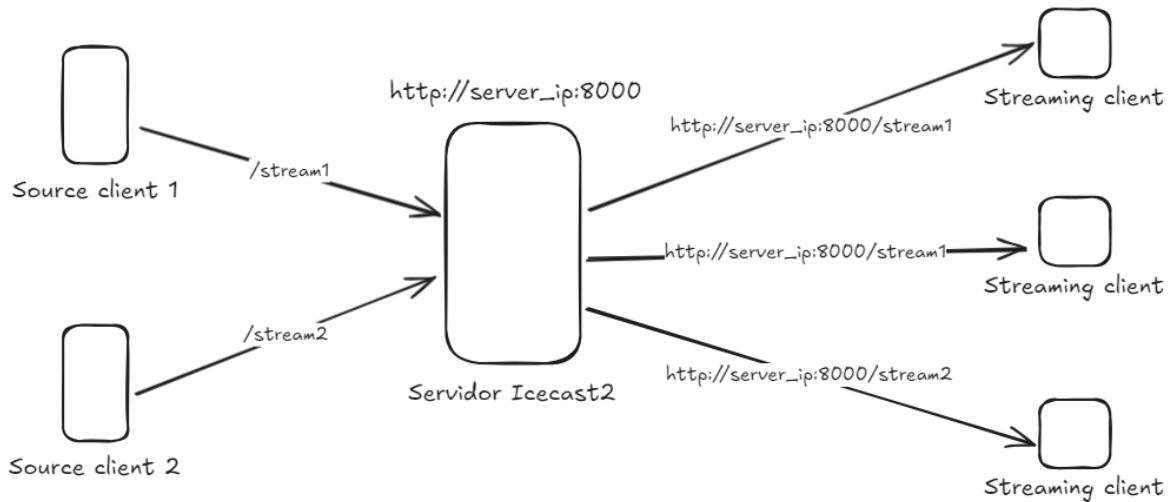


# Índice

<b>1. ¿Que es Icecast2 y como funciona?</b>	<b>3</b>
1.1. Source Clients . . . . .	3
1.2. Puntos de montaje . . . . .	3
1.3. Códecs . . . . .	3
<b>2. Instalación de servidor</b>	<b>4</b>
2.1. Servidor dedicado / Máquina virtual . . . . .	4
2.2. Contenedor Docker . . . . .	5
<b>3. Configuración de icecast2</b>	<b>5</b>
3.1. Cambiar configuración de icecast2 en Docker . . . . .	6
<b>4. Emisores de contenido / Source Clients</b>	<b>7</b>
4.1. VLC Media Player . . . . .	7
4.2. Ices2 . . . . .	8

# 1. ¿Que es Icecast2 y como funciona?

Icecast2 es un software que permite crear un servidor de streaming y distribuir transmisiones mediante un sistema de puntos de montaje. Esto quiere decir que el servidor de Icecast2 no codifica ni genera la transmisión, sino que la recibe y distribuye, montando cada flujo de datos sobre si mismo.



## 1.1. Source Clients

Como podemos observar en la imagen anterior el servidor por si mismo no emite el contenido, y necesita de clientes que envíen un flujo de datos con su codificación correspondiente. Es posible hacer esto de multiples maneras que veremos mas adelante.

## 1.2. Puntos de montaje

Cada punto de montaje es independiente, y se gestiona directamente desde el cliente que envía la transmisión. Estos puntos pueden ser limitados de diferentes formas en la configuración del servidor icecast2, estableciendo un limite de puntos de montaje, tamaño de lista de reproducción y ancho de banda máximo por transmisión.

## 1.3. Códecs

El *Source Client* tiene que codificar la transmisión antes de enviarla, de manera que los paquetes correspondientes deben instalarse manualmente en el cliente emisor, en caso de que la herramienta utilizada no los incluya en su instalación.

## 2. Instalación de servidor

Dependiendo de si buscamos mantener un servidor serio, o tan solo queremos transmitir momentaneamente a familiares o amigos, podemos valorar entre utilizar un servidor dedicado, una máquina virtual con un sistema completo o un contenedor.

- **Servidor dedicado:** esta opción nos dará mayor potencia para un servicio profesional, un buen control de logs y accesos, escalabilidad del hardware, etc.
- **Máquina virtual:** una máquina virtual completa también puede ser una buena opción para evitar accesos al equipo físico y permite además tomar instantaneas del estado de la máquina, que nos permitan restaurar el servidor en caso de un fallo fatal u otros inconvenientes que puedan tumbar el servicio. También ofrece buena escalabilidad y control de logs.
- **Contenedor docker:** un contenedor de docker es la mejor opción para montar algo sencillo y rápido, que nos permita en pocos minutos levantar el servicio y funcionar redireccionando un puerto del equipo principal a un puerto del contenedor (8000 en este caso). Al igual que las máquinas virtuales aísla el servidor de la máquina, pero impide la resolución de nombres, al no disponer de IP la red local (se utiliza un NAT entre contenedor y equipo físico).

### 2.1. Servidor dedicado / Máquina virtual

En sistemas basados en Debian podemos encontrar el software necesario en los repositorios de apt, aunque también podemos obtenerlo desde su [página oficial](#) e instalarlo manualmente.

```
#Desde repositorios de apt
sudo apt update -y
sudo apt install icecast2 -y

#Desde la pagina oficial
wget https://downloads.xiph.org/releases/icecast/icecast-2.4.4.tar.gz
tar -xzf icecast-2.4.4.tar.gz
sudo apt update -y
sudo apt install build-essential libxslt1-dev libxml2-dev libogg-dev -y
sudo apt install libvorbis-dev libcurl4-openssl-dev -y
cd icecast-2.4.4
./configure
make
sudo make install
```

## 2.2. Contenedor Docker

En un contenedor docker podemos instalarlo de la misma manera, aunque es mas común utilizar imágenes de alpine linux preparadas para ejecutar únicamente los programas necesarios. En este caso con el proyecto [infiniteproject/icecast](https://github.com/infiniteproject/icecast) podremos montar un servidor con el siguiente comando (*Situar todos los parámetros en la misma línea y borrar las \ en caso de utilizar Windows en el equipo de origen*):

```
docker run -dit \
  --name=icecast2 \ #Nombre distintivo del contendor
  -p 8000:8000/tcp \ #Puerto redirigido
  -e ICECAST_SOURCE_PASSWORD=<pass> \ #Para los Source Clients
  -e ICECAST_ADMIN_PASSWORD=<pass> \ #Para administracion web
  -e ICECAST_RELAY_PASSWORD=<pass> \ #Para otros servidores
  -e ICECAST_ADMIN_USERNAME=username \ #Usuario administrador
  -e ICECAST_ADMIN_EMAIL=email@gmail.com \ #Email de contacto
  -e ICECAST_LOCATION=Earth \ #Localizacion politica del servidor
  -e ICECAST_HOSTNAME=hostname \ #Nombre del equipo virtual
  -e ICECAST_MAX_CLIENTS=100 \ #Maximo de clientes de streaming
  -e ICECAST_MAX_SOURCES=2 \ #Maximo de transmisiones de source clients
  infiniteproject/icecast:latest
```

El contenedor está preparado para funcionar directamente, por lo que no hay que configurar nada mas, y solo resta transmitir contenido desde un source client.

## 3. Configuración de icecast2

La configuración de icecast se modifica mediante un archivo de nombre **icecast.xml**, que dependiendo de si la instalación se ha realizado con apt o manualmente puede encontrarse en **/etc/icecast2/icecast.xml** o **/usr/local/etc/icecast.xml**.

Algunas de las configuraciones mas relevantes de este archivo:

- **location**: localización política
- **admin**: email del administrador
- **clients**: maximo número de clientes de streaming conectados simultaneamente.
- **queue-size**: tamaño de cola de datos en bytes que el servidor almacena para cada cliente.
- **port**: puerto utilizado por el servicio.
- **bind-address**: dirección desde la que está permitido el acceso (0.0.0.0 para cualquier dirección).

Además de las configuraciones que afectan al rendimiento del servidor, también se configura en este archivo las contraseñas para los diferentes accesos en texto plano, por lo que los permisos referentes a este deben otorgarse con sumo cuidado:

- **source-password:** contraseña de acceso a emitir.
- **relay-password:** contraseña para redistribuir una emisión remota.
- **admin-user:** usuario administrador de la web del servicio.
- **admin-password:** contraseña del usuario administrador.

Para iniciar el servicio teniendo en cuenta los valores de configuración establecidos utilizaremos:

```
#Instalacion con apt
systemctl enable icecast2 #Al iniciar el sistema con systemd
systemctl start icecast2 #Iniciar con systemd
icecast2 -c /etc/icecast2/icecast.xml & #Inicio manual

#Instalacion manual con tar.gz
/usr/local/bin/icecast -c /usr/local/etc/icecast.xml &
```

### 3.1. Cambiar configuración de icecast2 en Docker

Si hemos iniciado el contenedor de docker con una configuración, y es necesario hacer cambios sin borrar y crear de nuevo el contenedor, podemos ejecutar una terminal interactiva e instalar nano para editar el icecast.xml.

```
docker exec -it icecast2 sh
apk add nano
nano /etc/icecast.xml #Editar y guardar con Ctrl+o y Ctrl+x
exit
```

Y se reinicia el contenedor para aplicar los cambios:

```
docker restart icecast2
```

## 4. Emisores de contenido / Source Clients

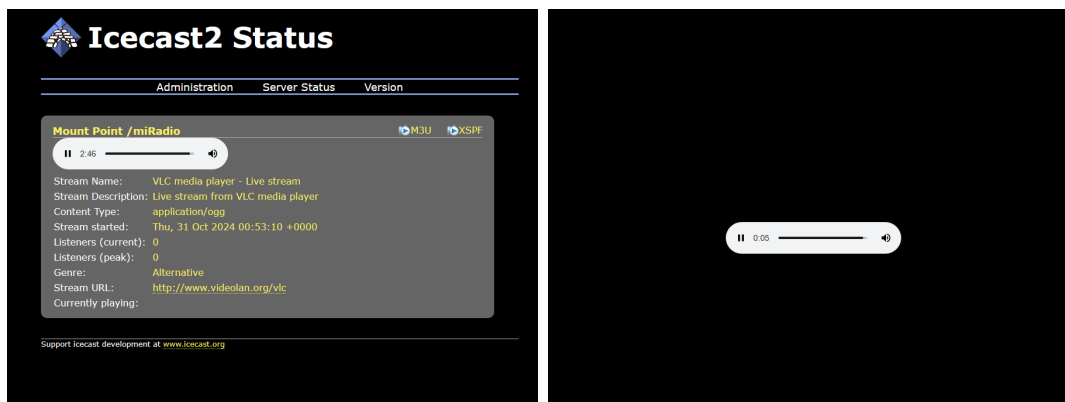
Los emisores que transmitan un flujo de datos al servidor pueden utilizar multiples códecs y programas, de entre los cuales mostraré dos:

### 4.1. VLC Media Player

Este es el mas sencillo de utilizar, debido a que trae una amplia variedad de códecs consigo, y ofrece una interfaz gráfica que facilita su uso para usuarios no tan experimentados.

- Añade a una lista varias canciones o videos a reproducir, y la guardamos como archivo .xspf (sin pasarse en cantidad o el servidor nos lanzará un error)
- Dirigete a el panel superior y accede a **Medio** y **Emitir...**
- Añade en el apartado **Archivo** la lista .xspf y haz clic en **Emitir**
- Omite la primera ventan y en Nuevo destino indica la opción **Icecast** y pulsa en Añadir (si quieres escuchar la transmisión localmente también, marca la casilla **Mostrar en local**).
- Indica la dirección IP del servidor de Icecast2, el puerto configurado, inventa un punto de montaje (por ejemplo: miRadio), y el usuario y contraseña (source:source-password).
- Elige un perfil (configuraciones proporcionadas por VLC) que utilice el códec que mejor se adapte a nuestro caso de uso (VLC también trae unos cuantos códecs preinstalados).
- Finalmente nos pregunta si queremos Emitir todas las emisiones elementales (si se activa, el cliente de streaming podrá elegir entre las diferentes pistas de audio, subtítulos... que se encuentren disponibles. Deshabilitarlo simplificará la transmisión), que podremos marcar opcionalmente si lo necesitamos.

Con esto comenzará la transmisión y se podrá acceder via web, desde la interfaz de icecast2, o desde el propio punto de montaje que únicamente proporcionará el stream.



## 4.2. Ices2

Si bien VLC es la opción mas sencilla y rápida, cuando se trata de streaming de audio Ices2 tiene algunas ventajas que lo hacen interesante. Ices2 está diseñado específicamente para icecast2 (aunque también es compatible con otros programas similares) y permite automatizar por completo las transmisiones, permitiendo un uso prolongado sin intervención alguna.

Este software está pensado principalmente para el streaming de audio en formato OGG o MP3, no obstante no los incluye, por lo que tendremos que instalarlos por separado:

```
sudo apt install ices2 libogg-dev vorbis-tools lame libmp3lame-dev -y
```

El archivo de configuración es ices.conf (o cualquier otro nombre .conf) y puede ser creado en cualquier parte del sistema. Es recomendable utilizar una plantilla de configuración y modificarla con los datos de nuestro servidor icecast2, puntos de montaje deseados, bitrate de la transmisión, playlist .m3u o .xspf, etc.

```
<ices>
  <!-- Configuracion del servidor Icecast -->
  <background>1</background> <!-- Poner en segundo plano al iniciar -->
  <logpath>/var/log/ices</logpath>
  <logfile>ices.log</logfile>
  <loglevel>3</loglevel> <!-- Nivel de log 0-4 de menos a mas detallado -->
  <stream>
    <!-- Datos de conexion al servidor Icecast -->
    <server>localhost</server> <!-- Direccion del servidor Icecast -->
    <port>8000</port> <!-- Puerto del servidor -->
    <password>pass</password> <!-- Password del servidor Icecast -->
    <mount>/stream.ogg</mount> <!-- Punto de montaje en el servidor -->
    <name>Mi Radio en Linea</name> <!-- Nombre de la transmision -->
    <genre>Musica</genre> <!-- Genero de la transmision -->
    <description>Estacion de prueba con Ices2</description>
    <url>http://mi-radio-en-linea.com</url> <!-- URL transmision -->
    <public>1</public> <!-- 1 para hacer la transmision publica -->
    <!-- Opciones de codificacion -->
    <bitrate>128</bitrate> <!-- Bitrate de la transmision en kbps -->
    <quality>3</quality> <!-- Calidad de transmision 0-10 -->
    <samplerate>44100</samplerate> <!-- Frecuencia de muestreo -->
    <channels>2</channels> <!-- Canales de audio: 1 mono, 2 estereo -->
    <!-- Lista de reproduccion -->
    <playlist>
      <file>/ruta/a/mi_lista_de_reproduccion.m3u</file>
      <randomize>1</randomize> <!-- reproduccion aleatoria -->
      <type>m3u</type> <!-- Tipo de lista de reproduccion m3u o xml -->
    </playlist>
    <!-- Configuracion de reconexion -->
    <reconnectdelay>2</reconnectdelay> <!-- Retardo entre intentos -->
  </stream>
</ices>
```



Con una plantilla como la anterior, ejecutamos el archivo con ices2 y automáticamente comenzará el streaming sobre el servidor y punto de montaje indicados en el archivo. Esto nos permite disponer de varias listas de reproducción y archivos de configuración, que utilicen puntos de montaje y parámetros específicos que podemos lanzar en cualquier momento.

```
ices2 /etc/ices.conf
```

También puede ser conveniente lanzar este último comando como ejecución en segundo plano para no perder la operabilidad de la línea de comandos, y así poder monitorizar otras cosas, como el rendimiento y consumo de recursos de la máquina, o los logs del propio ices2 en tiempo real.