

Table des matières

Créer une Web Radio avec Icecast	2
Installation	2
Paramétrage	2
Icecast 2.4	2
Icecast 2.5	3
Conf NPM	4
ezstream	4
LiquidSoap	5
Préparer votre Playlist	6
Tester	7
EZSTREAM	7
LIQUIDSOAP	7
URL de stream	7
URL de Gestion	7
Script	8
Démarrage	8
Auto Script ezstream	8
Auto Script liquidsoap	9
Best Script	9
Arrêt	10
Lecture	11
Consommation	12
▣ Maintenance	12
Démontage	12
Montage	12
Ajout de MP3	13
Playlist	13
Contrôle	13
Montage	13



Créer une Web Radio avec Icecast

Créer votre Web Radio sur votre rpi.

Installation

```
sudo apt-get install icecast2 ezstream
```

Paramétrage

Icecast 2.4

```
nano /etc/icecast2/icecast.xml
```

configurer les sections `<limit> </limit>` et `<authentication> </authentication>`

```
<icecast>
  <!-- location and admin are two arbitrary strings that are e.g. visible
        on the server info page of the icecast web interface
        (server_version.xsl). -->
  <location>To the Moon</location>
  <admin>admin@mazinge.fr</admin>

  <!-- IMPORTANT!
        Especially for inexperienced users:
        Start out by ONLY changing all passwords and restarting Icecast.
        For detailed setup instructions please refer to the documentation.
        "It's also available here: http://icecast.org/docs/"
  -->

  <limits>
    <clients>100</clients>
    <sources>2</sources>
    <queue-size>524288</queue-size>
    <client-timeout>30</client-timeout>
    <header-timeout>15</header-timeout>
    <source-timeout>10</source-timeout>
```

```

    <!-- If enabled, this will provide a burst of data when a client
         first connects, thereby significantly reducing the startup
         time for listeners that do substantial buffering. However,
         it also significantly increases latency between the source
         client and listening client. For low-latency setups, you
         might want to disable this. -->
    <burst-on-connect>1</burst-on-connect>
    <!-- same as burst-on-connect, but this allows for being more
         specific on how much to burst. "Most people won't need to
         change from the default 64k. Applies to all mountpoints" -->
    <burst-size>65535</burst-size>
</limits>

<authentication>
    <!-- Sources log in with username 'source' -->
    <source-password>hackme</source-password>
    <!-- Relays log in with username 'relay' -->
    <relay-password>hackme</relay-password>

    <!-- Admin logs in with the username given below -->
    <admin-user>admin</admin-user>
    <admin-password>hackme</admin-password>
</authentication>

```

Icecast 2.5

Upgrade pour Raspbian bullseye

Derrière un Reverse Proxy.

A partir de la version 2.5, l'écoute derrière un proxy permet de renvoyer l'IP de l'internaute en face !!
Sinon c'est l'ip du reverse qui sera dans vos log accès !

```
nano /etc/icecast2/icecast.xml
```

Changer ces blocks !!

```

<listen-socket>
    <port>8000</port>
    <trusted-proxy>#vs</trusted-proxy>
</listen-socket>
<listen-socket type="virtual" id="vs">
    <client-address>IP DU REVERSE PROXY</client-address>
</listen-socket>

```

```

<security>
    <chroot>0</chroot>
    <trusted-proxy>IP DU REVERSE PROXY</trusted-proxy>
</security>

```

Conf NPM

Dans Config Nginx Personnalisée :

```
location / {
    proxy_pass http://IP SERVEUR STREAM:8000/stream;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;

    # Très important pour le streaming audio
    proxy_buffering off;
    proxy_set_header Connection "";
}
```

ezstream

- Dans un premier temps il faudra dupliquer un exemple de configuration.

```
cp /usr/share/doc/ezstream/examples/ezstream_mp3.xml
   /usr/share/doc/ezstream/examples/ezstream_Funk.xml
```

- Puis éditer celui-ci avec les paramètres issue de icecast.xml

```
<ezstream>
  <url>http://localhost:8000/stream</url>
  <!--
    If a different user name than "source" should be used, set it in
    <sourceuser/>:
  -->
  <!-- <sourceuser>mr_stream</sourceuser> -->
  <sourcepassword>hackme</sourcepassword>
  <format>MP3</format>
  <filename>/home/pi/Music/Funk.m3u</filename>
  <!-- Once done streaming playlist.m3u, exit: -->
  <stream_once>0</stream_once>
  <shuffle>1</shuffle>
  <!--
    The following settings are used to describe your stream to the server.
    It's up to you to make sure that the bitrate/samplerate/channels
    information matches up with your input stream files. Note that
    <svrinfoquality /> only applies to Ogg Vorbis streams.
  -->
  <svrinfoname>My Funk Stream</svrinfoname>
  <svrinfourl>http://stream.mazinger.fr</svrinfourl>
  <svrinfogenre>Funk Music 80's</svrinfogenre>
  <svrinfodescription>Le Meilleur de la Funk!!</svrinfodescription>
  <svrinfobitrate>128</svrinfobitrate>
```

```
<svrinfochannels>2</svrinfochannels>
<svrinfosamplerate>44100</svrinfosamplerate>
<!--
  Prohibit the server to advertise the stream on a public YP directory:
-->
<svrinfopublic>0</svrinfopublic>
</ezstream>
```

LiquidSoap

Installer la source liquidsoap:

```
apt install liquidsoap
```



La Configuration est bien plus simple :

Editer un fichier.liq exemple 80.liq

```
#Pour executer le stream avec root
set("init.allow_root", true)

# FICHER DE LOG
set("log.file.path", "/tmp/ma-radio.log")

# Musique (Répertoire où se trouve les musiques)
myplaylist = playlist("/home/pi/Music/ma-zick.m3u", reload_mode="watch") #
Playlist principale

# Ajout des jingles (Répertoire où se trouve les jingles)
jingles = playlist("/home/pi/Music/jingles/")

# Si il y a un problème alors on lance la musique ci-dessous
security = single("/home/pi/Music/jingles/ERROR_MSG/Probleme-Radio.mp3")

# On créer une variable radio
radio = myplaylist

# On y ajoute les jingles 3 titres par défaut
radio = random(weights = [1, 2], [jingles, radio])

# Et la sécurité si pas de stream (playlist impossible à charger)
radio = fallback(track_sensitive = false, [radio, security])

# On crée une écoute d'un live, si il y a du son il sera diffuse sinon la
playlist reste en lecture
#live = fallback(track_sensitive=false,
#[input.http("http://localhost:8000/live"),
#radio])
```

```
# On lance une écoute pour voir si il y a un blanc pendant le live, si c'est
le cas la playlist se relance
#stream = fallback(track_sensitive=false,
#[ strip_blank(max_blank=5.,live) , radio ])

# Envoie du flux vers Icecast
output.icecast(%mp3,
  host="127.0.0.1",
  port=8000,
  password="hackme",
  mount="/stream",
  name="Best Funk Radio",
  description="Le meilleur de la Funk",
  genre="Funk Music 80's",
  url="http://stream.mazinger.fr",
  public=false,
  radio)
```

Préparer votre Playlist

- Positionnez vous dans votre répertoire "Music"

```
cd $HOME/Music
```

Si il n'existe pas créer le

```
mkdir /home/pi/Music
```

- Une fois dedans générer votre playlist en recherchant vos Mp3 depuis vos sources.

dans mon exemple je fait une recherche depuis un point de montage correspondant à mon NAS

```
find /mnt/BigHeroMp3/Funk\ Music/ALL\ FUNK/ -iname "*.mp3" >Funk.m3u
```

-iname permet d'être insensible à la casse (.mp3 et .MP3) seront valide !!

Extrait du fichier généré:

```
/mnt/BigHeroMp3/Funk Music/ALL FUNK/Bettye Lavette - I Can't Stop.mp3
/mnt/BigHeroMp3/Funk Music/ALL FUNK/Central Line - Walking Into Sunshine.mp3
/mnt/BigHeroMp3/Funk Music/ALL FUNK/COOL NOTES - LOOK WHAT YOU DONE TO
ME.mp3
/mnt/BigHeroMp3/Funk Music/ALL FUNK/David Joseph - You Can't Hide.mp3
/mnt/BigHeroMp3/Funk Music/ALL FUNK/ D Train-Keep Giving Me Love.mp3
/mnt/BigHeroMp3/Funk Music/ALL FUNK/A.W.B pick up the pieces.mp3
/mnt/BigHeroMp3/Funk Music/ALL FUNK/Fat Eddy Band - Let your body move
it.mp3
/mnt/BigHeroMp3/Funk Music/ALL FUNK/Cameo - Talking out.mp3
```

Donc mon fichier de playlist sera situé dans `/home/pi/Music/` et sera nommé **Funk.m3u**

Tester

EZSTREAM

Voici la commande à saisir pour lancer votre Web Radio:

```
ezstream -c /usr/share/doc/ezstream/examples/ezstream_Funk.xml
```



Vous pouvez choisir un autre emplacement lors de la copie du fichier exemple. (ezstream_mp3.xml)
Faire la copie dans un autre répertoire, /home/pi/EZconf
La commande sera donc "ezstream -c /home/pi/EZconf/ezstream_Funk.xml"

Si tout se passe bien vous devriez voir ceci dans votre terminal:

```
ezstream: Connected to http://localhost:8000/stream  
ezstream: Streaming ``Lionel Richie - Lionel Richie-Say You Say Me''
```

Cela indique que ezstream est bien en fonctionnement et est en train de lire le titre ``Lionel Richie - Lionel Richie-Say You Say Me''

LIQUIDSOAP

Tester 60s de stream pour voir si tout est ok !!

```
timeout 60s liquidsoap /media/USB/80.liq
```

URL de stream

```
http://localhost:8000/stream ou http://ipDuServeur:8000/stream
```

URL de Gestion

```
http://localhost:8000 ou http://ipDuServeur:8000
```

Icecast2 Status

Administration Server Status Version

Mount Point /stream

Stream Name: My Funk Stream
 Stream Description: Le Meilleur de la Funk!!
 Content Type: audio/mpeg
 Stream started: Fri, 19 Aug 2022 20:04:59 +0200
 Bitrate: 128
 Listeners (current): 0
 Listeners (peak): 2
 Genre: Funk Music 80's
 Stream URL: <http://stream.mazinger.fr>
 Currently playing: Prince Charles & The City BeatFunk 'N' Soul 1983 - Prince Charles-In the Streets

Support Icecast development at www.icecast.org

Icecast2 Admin

Admin Home Mountpoint List Public Home

Listener Stats

Mountpoint /stream

List Clients Move Listeners Update Metadata Kill Source

IP	Sec. connected	User Agent	Action
10.200.40.30	389	VLC/3.0.17.4 LibVLC/3.0.17.4	Kick

Support Icecast development at www.icecast.org

Script

Démarrage

Afin d'automatiser le lancement vous pouvez créer un script qui exécutera ezstream.

Ex: Funk.sh

```
#!/bin/sh
ezstream -c /usr/share/doc/ezstream/examples/ezstream_Funk.xml
```

Afin de pouvoir le faire exécuter en tâche de fond:

```
./Funk.sh &
```

Auto Script ezstream

```
#!/bin/sh
#Lancement de la playlist via ezstream
ezstream -c /home/pi/Music/ezstream_Best-Funk.xml &
#Récupération du PID du process de ezstream dans une variable (varPID)
varPID=$(ps -aux | grep ezstream | awk {'print $2'} | sed '2d')
```

```
#Timer réglé sur 4h de diffusion
sleep 14400
#Arrêt de la diffusion du stream
kill $varPID

exit;
```

Auto Script liquidsoap

```
#!/bin/bash
#Lancement de la playlist via ezstream
liquidsoap /home/pi/Music/Funk.liq &
#Récupération du PID du process de liquidsoap dans une variable (varPID)
varPID=$(ps -aux | grep liquidsoap | awk {'print $2'} | sed '2d')
#Timer réglé sur 4h de diffusion
sleep 14400
#Arrêt de la diffusion du stream
kill $varPID

exit;
```

Best Script

Voici une version plus propre pour une mise en prod plus sure avec log !!

```
#!/bin/bash

# === CONFIG ===
LIQ_SCRIPT="/home/pi/Music/Funk.liq"
LOG_FILE="/home/pi/Music/stream.log"
RUNTIME=14400 # Durée en secondes (4h)

# === ENVIRONNEMENT ===
export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

# === START LOG ===
echo "=== $(date '+%Y-%m-%d %H:%M:%S') - DÉBUT DU STREAM ===" >> "$LOG_FILE"

# === LANCEMENT DU STREAM ===
liquidsoap "$LIQ_SCRIPT" &
PID=$!
echo "Liquidsoap lancé avec le PID $PID" >> "$LOG_FILE"

# === VÉRIFICATION ===
sleep 5
if ! ps -p "$PID" > /dev/null; then
```

```

    echo "ERREUR : Liquidsoap ne s'est pas lancé correctement." >>
"$LOG_FILE"
    exit 1
fi

# === TEMPS DE DIFFUSION ===
echo "Streaming pendant $((RUNTIME/3600))h..." >> "$LOG_FILE"
sleep "$RUNTIME"

# === ARRÊT DU STREAM ===
if ps -p "$PID" > /dev/null; then
    kill "$PID"
    echo "Liquidsoap (PID $PID) stoppé proprement." >> "$LOG_FILE"
else
    echo "Avertissement : le processus Liquidsoap n'était plus actif à
l'arrêt." >> "$LOG_FILE"
fi

echo "=== $(date '+%Y-%m-%d %H:%M:%S') - FIN DU STREAM ===" >> "$LOG_FILE"

```

Dans Crontab



Crontab -e

```
15 20 * * * /home/pi/Music/Best-Funk.sh >/dev/null 2>&1
```

Reload du cron, pour prendre les modif en compte.

```
/etc/init.d/cron reload
```



Chaque soir à partir de 20h15 le script sera exécuté par cron, et s'arrêtera 4h plus tard. Pour changer l'heure d'arrêt modifier le script dans la partie #Timer (sleep xxxx en sec).

Arrêt

ezstream

```

ps -aux | grep ezstream
root      28529  0.2  0.7  50704  7412 pts/2    S+   20:04   0:24  ezstream -c
/usr/share/doc/ezstream/examples/ezstream_Funk.xml

kill 28529

```

liquidsoap

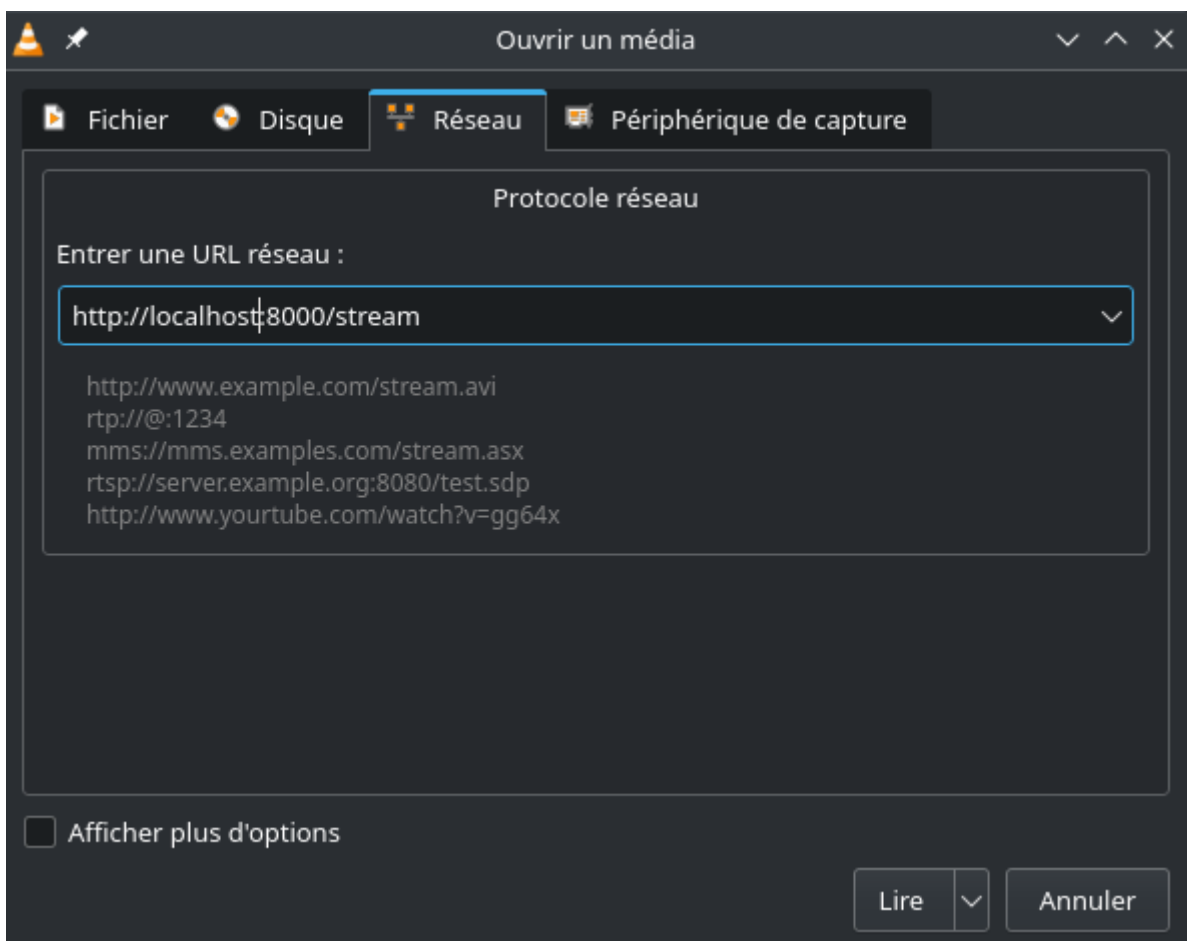
```
ps -aux | grep liquidsoap
root      28729  0.2  0.7 50704  7412 pts/2    S+   20:24   0:36 liquidsoap
/home/pi/Music/Funk.liq
kill 28729
```

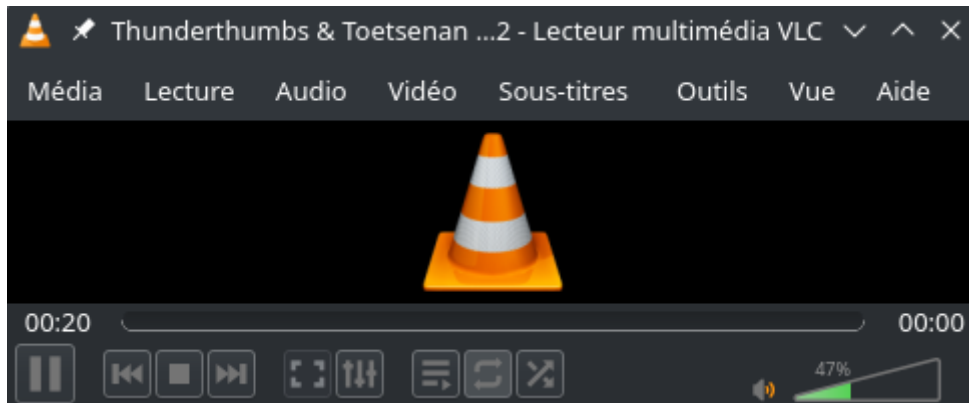
Icecast

```
systemctl stop icecast2.service
```

Lecture

- Avec VLC





Consommation

Niveau consommation de ressources cette solutions est peu gourmande:

```
1 [ 0.0%] Tasks: 55, 72 thr, 97 kthr; 1 running
2 [||||] 4.5% Load average: 0.31 0.21 0.16
3 [ | ] 1.3% Uptime: 93 days, 04:27:56
4 [ 0.0%]
Mem [|||||] 206M/923M
Swp [|||||] 96.1M/100.0M
```

Après reste à voir la charge en fonction du nombre de Client(s)

☐ Maintenance

Ajout de titres et régénérer la playlist 😎

Démontage

Si je travaille avec un support USB !!

```
umount -f /media/USB
```

Si bloqué par un processus fantôme et même un "kill -9" n'y fait rien !

```
fuser -k /media/USB/
```

Montage

Pour m'affranchir des ACL sur le point de montage d'origine pour www-data, je remonte mon volume ailleurs.

```
mount UUID=52D5-74A4 /mnt/USB
```

Commande pour avoir le "UUID"

```
lsblk -f
```

Ajout de MP3

```
cp -R /source_MP3/*.* /mnt/USB/Destination/
```

Playlist

Je renomme l'ancienne playlist !!

```
mv /mnt/USB/myplaylist.m3u /mnt/USB/myplaylist.m3u.old
```

Je génère la nouvelle playlist.

```
find /mnt/USB/Destination/ -iname "*.mp3" >/mnt/USB/myplaylist.m3u
```

Contrôle

Le nombre de fichiers

```
ll /mnt/USB/Destination/ | wc -l
```

Le nombre de lignes dans le fichier playlist.

```
cat /mnt/USB/myplaylist.m3u | wc -l
```

Les nombres devraient être identiques

Penser à démonter le point de montage de travail

```
umount /mnt/USB
```

Montage

Réappliquer le montage issu du fichier fstab

```
mount -a
```

Conf possible fstab :

```
#USB Disk Stream
```

```
UUID=52D5-74A4 /media/USB vfat ro,suid,dev,auto,async,uid=33,umask=007  
0 2
```

From:

<https://wiki.mazinger.fr/wiki/> - **My Personal Wiki**

Permanent link:

<https://wiki.mazinger.fr/wiki/doku.php?id=web:server:icecast:index>

Last update: **2026/04/27 23:24**

