

Table des matières

| | |
|--------------------------------|---|
| Machine virtuelle | 3 |
| Configuration | 3 |
| Mises à jour | 6 |

Liberta Audio est un service de diffusion audio (musique, podcasts, etc.) similaire à Spotify, Deezer, etc. basé sur le logiciel libre [Funkwhale](#).

Liberta Audio vous offre 5 Go de stockage pour votre musique et vos podcasts (attention de bien disposer des droits de diffusion de vos contenus !), organisés en chaînes et en audiOTHèques.

Le logiciel est fédéré et permet de suivre et interagir avec les utilisateurs d'autres instances de Funkwhale disséminées sur la planète, si Liberta Audio a décidé de suivre ces instance.

N'hésitez pas à consulter la [documentation de Funkwhale](#).

Machine virtuelle

- 2 CPU, 2 Go de mémoire
- Système d'exploitation : [Debian](#) stable
- Stockage des données : [ZFS](#) sur l'hyperviseur, monté en NFS.

Configuration

Funkwhale est installé via la méthode classique « Non-Docker ».

Le fichier de configuration principal `/var/www/funkwhale/config/.env` :

```
FUNKWHALE_API_IP=127.0.0.1
FUNKWHALE_API_PORT=(privé)
FUNKWHALE_WEB_WORKERS=4
FUNKWHALE_HOSTNAME=audio.liberta.vip
FUNKWHALE_PROTOCOL=https
TYPESENSE_API_KEY=(privé)
EMAIL_CONFIG=smtp+tls://libertadmin@liberta.vip:(privé)@(privé):587
ACCOUNT_EMAIL_VERIFICATION_ENFORCE=true
DEFAULT_FROM_EMAIL=libertadmin@liberta.vip
REVERSE_PROXY_TYPE=nginx
DATABASE_URL=postgresql://(privé):(privé)@192.168.10.6:5432/(privé)
CACHE_URL=redis://192.168.10.5:6379/(privé)
MEDIA_ROOT=/var/www/funkwhale/data/media
STATIC_ROOT=/var/www/funkwhale/data/static
DJANGO_SETTINGS_MODULE=config.settings.production
DJANGO_SECRET_KEY=(privé)
RAVEN_ENABLED=true
RAVEN_DSN=https://(privé):(privé)@sentry.eliotberriot.com/(privé)
MUSIC_DIRECTORY_PATH=/var/www/funkwhale/data/music
MUSIC_DIRECTORY_SERVE_PATH=/var/www/funkwhale/data/music
FUNKWHALE_FRONTEND_PATH=/var/www/funkwhale/front/dist
NGINX_MAX_BODY_SIZE=4000M
```

Le fichier de configuration du serveur HTTP Nginx `/etc/nginx/sites-enabled/funkwhale.conf` :

```
upstream funkwhale-api {
    server 127.0.0.1:(privé);
}
server {
    listen 80;
    listen [::]:80;
    server_name audio.liberta.vip;
    location / {
        return 301 https://$host$request_uri;
    }
}
map $http_upgrade $connection_upgrade {
    default upgrade;
    ''      close;
}
server {
    listen          443 ssl http2;
    listen [::]:443 ssl http2;
    charset         utf-8;
    server_name     audio.liberta.vip;
    ssl_protocols  TLSv1.2;
    ssl_ciphers     HIGH:!MEDIUM:!LOW:!aNULL:!NULL:!SHA;
    ssl_prefer_server_ciphers on;
    ssl_session_cache shared:SSL:10m;
    ssl_certificate /etc/letsencrypt/live/liberta.vip/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/liberta.vip/privkey.pem;
    add_header     Strict-Transport-Security "max-age=31536000";
    add_header     Content-Security-Policy "default-src 'self'; connect-src
https: wss: http: ws: 'self' 'unsafe-eval'; script-src 'self' 'wasm-unsafe-
eval'; style-src https: http: 'self' 'unsafe-inline'; img-src https: http:
'self' data:; font-src https: http: 'self' data:; media-src https: http:
'self' data:; object-src 'none'";
    add_header     Referrer-Policy "strict-origin-when-cross-origin";
    add_header     X-Frame-Options "SAMEORIGIN" always;
    add_header     Service-Worker-Allowed "/";
    root            /var/www/funkwhale/front/dist;
    gzip            on;
    gzip_comp_level 5;
    gzip_min_length 256;
    gzip_proxied    any;
    gzip_vary       on;
    gzip_types
        application/javascript
        application/vnd.geo+json
        application/vnd.ms-fontobject
        application/x-font-ttf
        application/x-web-app-manifest+json
        font/opentype
        image/bmp
        image/svg+xml
        image/x-icon
```

```
text/cache-manifest
text/css
text/plain
text/vcard
text/vnd.rim.location.xloc
text/vtt
text/x-component
text/x-cross-domain-policy;
location /api/ {
    include /etc/nginx/funkwhale_proxy.conf;
    client_max_body_size 4000M;
    proxy_pass http://funkwhale-api;
}
location / {
    alias /var/www/funkwhale/front/dist/;
    expires 1d;
    try_files $uri $uri/ /index.html;
}
location = /embed.html {
    add_header Content-Security-Policy "connect-src https: http: 'self';
default-src 'self'; script-src 'self' unpkg.com 'unsafe-inline' 'unsafe-
eval'; style-src https: http: 'self' 'unsafe-inline'; img-src https: http:
'self' data:; font-src https: http: 'self' data:; object-src 'none'; media-
src https: http: 'self' data:";
    add_header Referrer-Policy "strict-origin-when-cross-origin";
    alias /var/www/funkwhale/front/dist/embed.html;
    expires 1d;
}
location /federation/ {
    include /etc/nginx/funkwhale_proxy.conf;
    proxy_pass http://funkwhale-api;
}
location /rest/ {
    include /etc/nginx/funkwhale_proxy.conf;
    proxy_pass http://funkwhale-api/api/subsonic/rest/;
}
location /.well-known/ {
    include /etc/nginx/funkwhale_proxy.conf;
    proxy_pass http://funkwhale-api;
}
location /media/__sized__/ {
    alias /var/www/funkwhale/data/media/__sized__/;
    add_header Access-Control-Allow-Origin '*';
}
location /media/attachments/ {
    alias /var/www/funkwhale/data/media/attachments/;
    add_header Access-Control-Allow-Origin '*';
}
location ~ /_protected/media/(.+) {
    internal;
    add_header Access-Control-Allow-Origin '*';
}
```

```
}
location /_protected/music/ {
    internal;
    alias /var/www/funkwhale/data/music/;
    add_header Access-Control-Allow-Origin '*';
}
location /manifest.json {
    return 302 /api/v1/instance/spa-manifest.json;
}
}
```

Mises à jour

Un simple script Shell est utilisé pour « automatiser » le processus :

Le fichier funkwhale_upgrade.sh :

```
#!/bin/sh
export FUNKWHALE_VERSION="1.3.0"
cd /var/www/funkwhale
sudo -u (privé) curl -L -o front.zip
"https://dev.funkwhale.audio/funkwhale/funkwhale/builds/artifacts/$FUNKWHALE_VERSION/download?job=build_front"
sudo -u (privé) unzip -o front.zip
sudo -u (privé) rm front.zip
sudo -u (privé) curl -L -o "api-$FUNKWHALE_VERSION.zip"
"https://dev.funkwhale.audio/funkwhale/funkwhale/-/jobs/artifacts/$FUNKWHALE_VERSION/download?job=build_api"
sudo -u (privé) unzip "api-$FUNKWHALE_VERSION.zip" -d extracted
sudo -u (privé) rm -rf api/
sudo -u (privé) mv extracted/api .
sudo -u (privé) rm -rf extracted
api/install_os_dependencies.sh install
sudo -u (privé) -H -E /var/www/funkwhale/virtualenv/bin/pip install --
upgrade pip wheel
sudo -u (privé) -H -E /var/www/funkwhale/virtualenv/bin/pip install --
editable ./api
sudo -u (privé) -H -E /var/www/funkwhale/virtualenv/bin/funkwhale-manage
collectstatic --no-input
systemctl stop funkwhale-beat funkwhale-worker funkwhale-server
curl -L -o "/etc/systemd/system/funkwhale.target"
"https://dev.funkwhale.audio/funkwhale/funkwhale/raw/$FUNKWHALE_VERSION/depl
oy/funkwhale.target"
curl -L -o "/etc/systemd/system/funkwhale-server.service"
"https://dev.funkwhale.audio/funkwhale/funkwhale/raw/$FUNKWHALE_VERSION/depl
oy/funkwhale-server.service"
curl -L -o "/etc/systemd/system/funkwhale-worker.service"
"https://dev.funkwhale.audio/funkwhale/funkwhale/raw/$FUNKWHALE_VERSION/depl
oy/funkwhale-worker.service"
```

```
curl -L -o "/etc/systemd/system/funkwhale-beat.service"  
"https://dev.funkwhale.audio/funkwhale/funkwhale/raw/$FUNKWHALE_VERSION/depl  
oy/funkwhale-beat.service"  
sed -i -e 's@srv/funkwhale@var/www/funkwhale@g' -e  
's@User=.*$@User=(privé)@g' -e 's@venv/bin@virtualenv/bin@g' -e  
'/^After=.*$/d' /etc/systemd/system/funkwhale-*.service  
systemctl daemon-reload  
sudo -u (privé) -H -E /var/www/funkwhale/virtualenv/bin/funkwhale-manage  
migrate  
systemctl start funkwhale-server funkwhale-worker funkwhale-beat  
systemctl restart nginx
```

From:
<https://doc.liberta.vip/> - **Documentation Liberta**

Permanent link:
<https://doc.liberta.vip/support/audio>

Last update: **12/06/2023 17:23**

