Ampache

- Ampache Music Streaming Server

Ampache Music Streaming Server

https://www.youtube.com/embed/PpUkgQSi_RQ

Why Run Your Own Music Streaming Server?

I've been an iTunes user for more than a decade, in fact going on two decades now. I liked it, and still like it somewhat, but with the move to split everything out, the interface is painful to me. Getting music to my local machine is a hassle. I also have become less and less pleased with the thought of someone else holding my data (in this case my music). I paid for it, I own it, and I've collected it for more than 35 years now. Why should I not have it locally?

Before I get comments, yes, I know I can download it, but that goes right back to the hassle.

I've bean looking for solutions to host and stream from my own home server. It wasn't hard to find several viable options, but Ampache kept popping up. So here are my show notes with Installation, first run wizard, and a few admin items you'll likely want to take care of early on as you setup your own Ampache streaming server.

Installation

WE will be using Docker to install Ampache from the official ampache docker image. Docker is an excellent tool for running extremely minimal virtual servers so there's not all the bloat of a full virtual OS install.

I'm using Ubuntu 18.04 (Zorin 15.2), so I'll be using the Docker-CE installation instructions found on Digital Ocean's site to install Docker. If you don't have Docker installed yet, feel free to jump over there and get it installed first. If you have a different OS (Newer or older, different Linux distro, Windows, or MacOS) just google around for the instructions to install Docker-CE on your OS. :

Installing Ampache, once Docker is installed is really quite straight-forward. A few things we want to keep in mind:

1. We want to make sure we setup proper port mapping from our host to our docker container where Ampache will be running. Ampache runs on port 80 (the common port for most web sites). In the Ampache instructions for their docker they only give information to map port 80.
2. This, however is an unsecured (no SSL) port, so we also want to map a port from our host to our docker container for port 443.
3. Finally, we need to map a volume (a storage location on our host) to `/media` on our docker container. This is where we will keep our music.

I mapped `/home/brian/Music/ampache` to `/media` and this has worked well for me. So let's begin.

First we'll setup the directory for our music storage.

Starting from my home directory on my server, `/home/brian`, I'll change directory into Music, then make a new directory called `ampache`.

```
cd Music
```

```
mkdir ampache
```

Done! Not too bad, right?

Next, we want to put some of our music in this folder. You can use the `cp` command to copy music from one folder to another, or the `mv` command to move it (cut / paste), or `scp` to copy music from a remote machine. Of course, if you are running a GUI (Graphical User Interface) on your server, you can use all of the normal methods for getting music to the `ampache` directory you just made.

Now we need to get Ampache installed. For this we use a very straight-forward one liner in Docker. Depending on whether you've added your user to the docker group or not, you may need to use `sudo` to run these commands. I'll put them with sudo, but if you are part of the docker group, feel free to leave it off.

```
sudo docker run -d --name=ampache -v /home/<your user>/Music/ampache:/media -p 8051:80 -p 8543:443 ampache/ampache
```

What we are doing above is explained like this:

`sudo` -> do this as a super user with root permissions

`docker run -d` -> fun this docker container as a daemon (a service that runscontinuously in the background)

`--name=ampache` -> call this container "ampache" so I can find it in a list of containers.

`-v /home/<your user>/Music/ampache:/media` -> map my host machine folder (and yes you need to put your actual username for the host server where it says "<your user>") to the "/media" directory in my container.

`-p 8051:80` -> map my host port 8051 to the container port 80. NOTE: this step isn't required if you aren't running anything else on this server that uses port 80 already.

`-p 8543:443` -> map my host port 8543 to the container port 443. NOTE: this step isn't required if you aren't running anything else on this server that uses port 443 already.

On the two NOTES above, keep in mind, it may still be useful to point a non-standard port to 80 and 443 in the container as it leaves you room to use NginX or another Proxy server when you want to reach your music server more easily and securely from outside your home network.

`ampache/ampache` -> pull down the official ampache image to build our container from.

Extra Flags for Docker

You may want to add one more flag and option.

`--restart=unless-stopped` -> this option will restart the container if it should crash for some reason, or if the system should do an automatic reboot during the night. It won't restart on it's own, however, if you intentionally stopped it using the `docker stop <container name>` command.

First Run Wizard

After the docker command finishes running you shoudl be able to reach your server at it's IP address and whatever port you mapped to the container port 80. For me it is http://192.168.7.125:8051 (your's will likely be different).

The first time you visit the running site, you'll run through a "First Run Wizard". This wizrd helps you setup an administrative useer, the database (mysql included in the image we pulled) connection, and just makes sure everything we need is installed.

1. Select Language
2. Make sure all checks are good - but there will be a warning about files larger than 20 MB not being allowed - don't sweat that one.
3. MySQL Setup - Leave the defaults all the way to the checkboxes. You want to check the option for "Create Database User". When you check it you'll get two new fields. You can change the username if you want, but remember it. Then add a password and remember it.
4. On the next screen, you need to fill your user you added, and the password for that user. Scroll down and choose if you want to allow transcoding. You can check all the API boxes, or just take the defautls. Then click 'Create Config' (bottom center).
5. Make your first user (admin level account). Click 'Create Account'
6. Scroll to the bottom, and click 'Update Now'
7. You should get 'No Update Needed'. Click the link for Return to Main Screen' and login with your admin user from step 5.

Add Music

Click on Admin >> Add Catalog. From here, give you catalog a name, then move down and make sure 'local' is selected. Finally, give the path of `/media` for the location of your music. Click 'Go' and the music should begin filling your Ampache system pretty quickly.

You can now navigate to 'Songs' on the main view (the little headphones icon), and see your list of music as it's being built.

Conclusion

This is a very brief overview of what Ampache can do, and how to install and get it running. So before you invest hours or days in getting your music and playlists setup, make sure to get everything else setup just like you want it in your server. That way if you bork it, you can nuke and pave an dnot lose tons of effort in the process.