

IRC!Radio by dsc_

IRC!Radio

IRC!Radio is a radio station for IRC channels. You hang around on IRC, adding YouTube songs to the bot, listening to it with all your friends. Great fun!

Stack

IRC!Radio aims to be minimalistic/small using:

- Python >= 3.7
- SQLite
- LiquidSoap >= 1.4.3
- Icecast2
- Quart web framework

And all in a FreeBSD jail (in this case).

Command list

- “ - !np - current song
- !tune - upvote song
- !boo - downvote song
- !request - search and queue a song by title or YouTube id
- !dj+ - add a YouTube ID to the radiostream
- !dj- - remove a YouTube ID
- !ban+ - ban a YouTube ID and/or nickname
- !ban- - unban a YouTube ID and/or nickname
- !skip - skips current song
- !listeners - show current amount of listeners
- !queue - show queued up music
- !queue_user - queue a random song by user
- !search - search for a title
- !stats - stats

Installation

The following assumes you have a VPS somewhere with root access (duh). It assumes you're using `bastille` for the jail manager, and it assumes you have a `caddy` jail already set up for reverse proxy and certs.

Before doing anything else, since we're on FreeBSD, create a jail. Notice that this is a thin jail with no network interface specified, therefore it'll use the `bastille0` cloned loopback device for its network.

```
bastille create radio_jail 13.0-RELEASE 10.101.10.180
```

In my case, I have a custom `.cshrc` file to make the terminal nicer looking (and a script to copy it into place).

```
/usr/local/scripts/custom_cshrc.sh radio_jail
```

The `radio` user will be doing a bunch of the heavy lifting. It needs its own `/home` directory.

```
bastille cmd radio_jail pw adduser -n radio -m -d /home/radio -s /usr/local/bin/bash -c "radio user"
```

An `icecast` user will be needed to run `icecast`, but it doesn't need its own `/home` directory.

```
bastille cmd radio_jail pw adduser -n icecast -G wheel -d /nonexistent -s /usr/sbin/nologin -c "icecast"
```

Requirements

Part I - Everything except for `liquidsoap`, basically

Into the jail, as root:

```
bastille console radio_jail
```

First, might as well get onto the latest package repo.

```
mkdir -p /usr/local/etc/pkg/repos
```

```
echo 'FreeBSD: { url: 'pkg+http://pkg.FreeBSD.org/${ABI}/latest', enabled: yes }' >  
/usr/local/etc/pkg/repos/FreeBSD.conf
```

Do a couple rounds of package installing. First, basics. Then specifics.

```
pkg install -y bat htop git vim-console tmux
```

```
pkg install -y icecast py38-virtualenv libogg nginx ffmpeg sqlite3 py38-sqlite3 gmake bash
```

And because there is no `liquidsoap` in ports:

```
pkg install -y ocaml-opam libmad taglib libsamplerate pkgconf gavl fdk-aac
```

Part II - Use `opam` to install `liquidsoap`

```
su radio
```

Inside the jail, as the `radio` user, the majority of the rest will happen. Might as well get to the `/home` directory.

```
cd
```

```
opam init
```

Follow the instructions to make sure `.profile` is properly sourced.

```
vim .bashrc
```

And paste:

```
source /usr/home/radio/.profile
```

The compiler in the package repo is too old for what we need.

```
opam switch create 4.12.0
```

```
opam install fdkaac gavl
```

```
opam depext taglib mad lame vorbis cry samplerate liquidsoap
```

And the `install` command that won't work without the `env` vars (whether included in advance or part of the command):

```
C_INCLUDE_PATH=$C_INCLUDE_PATH:/usr/local/include  
CPLUS_INCLUDE_PATH=$CPLUS_INCLUDE_PATH:/usr/local/include LIBRARY_PATH=$LIBRARY_PATH:/usr/local/lib  
opam install taglib mad lame vorbis cry samplerate ffmpeg liquidsoap
```

Clone and Setup

Still as `radio` user, still from from `~`:

```
git clone https://git.wownero.com/dsc/ircradio.git
```

```
cd ircradio/
```

The magic commands that will need to be run more than once (here, and then farther down, at the end):

```
virtualenv -p /usr/local/bin/python3.8 venv
```

```
source venv/bin/activate
```

```
pip install -r requirements.txt
```

Adjust settings

Now that all the building blocks are in place:

```
cp settings.py_example settings.py
```

```
vim settings.py
```

Look at `settings.py` and configure it to your liking:

- Change `host` listening address at the top to internal IP given to the jail, `10.101.10.180`
- Change timezone to `America/New_York` or whatever
- Change `irc_host` from `localhost` to something like `irc.oftc.net` or `irc.libera.chat` or whatever
- If you change `irc_ssl` to `True`, change the `irc_port` accordingly.
- Change `irc_nick`, `irc_channels`, `irc_realname`, maybe `irc_command_prefix`
- Change `icecast2_hostname` to your `hostname`, i.e, `radio.example.com`
- Change the passwords under `icecast2_`
- Change the `liquidsoap_description` to whatever

Lastly, edit `ircradio/utls.py`, and comment out all of `liquidsoap_check_symlink()`, and just make it `pass`.

Alternatively, you can run the `generate` command that follows, and then run `find / -type f -name lastfm.liq`, and then as `root` put in a symlink so it will be able to find that file. But you'll need a thick jail to be able to do this. And you'll need to repeat the three magic `virtualenv` commands again.

When you are done, this will generate various initial configs (which we'll have to further edit):

```
python3.8 run.py generate
```

The generate function writes `icecast` / `liquidsoap` / `nginx` configuration files into `data/`.

Update configs

First, while still in the `radio` user's shell:

```
which liquidsoap
```

Then, `exit` out of `radio` and back to `root`. We'll need `root` shell for this section and the next. And if needed:

```
cd /home/radio/ircradio
```

liquidsoap

Where is `liquidsoap`? We got that above. That needs to be the path at the top of the `data/soap.liq` file. Paste it.

```
vim data/soap.liq
```

And while in there, comment out the row starting with `full`. In the final line, change `full` to `radio`. This change will remove the crossfade function unfortunately. Maybe 1.4.4 changed that function.

TODO: figure out crossfading, cuz I want it

Then `liquidsoap` also needs an `rc` file, rather than a `system.d` file.

```
vim /usr/local/etc/rc.d/liquidsoap
```

```
#!/bin/sh

# PROVIDE: liquidsoap
# REQUIRE: DAEMON
# BEFORE: LOGIN
# KEYWORD: shutdown

# Add the following line to /etc/rc.conf to enable `liquidsoap`.
#
```

```
#liquidsoap_enable="YES"

#
# To specify a non-default script file, set liquidsoap_script
# in /etc/rc.conf:
#
#liquidsoap_script="/home/radio/ircradio/data/soap.liq"
#

. /etc/rc.subr

name="liquidsoap"
rcvar=liquidsoap_enable

#update as necessary, the command path
command="/usr/home/radio/.opam/4.12.0/bin/liquidsoap"
command_args="--daemon 1>/dev/null"
#command_args="--daemon --quiet"
extra_commands="reload"

# read configuration and set defaults
load_rc_config "$name"
: ${liquidsoap_enable="NO"}
: ${liquidsoap_script="/home/radio/ircradio/data/soap.liq"}
: ${liquidsoap_flags="${liquidsoap_script}"}
: ${liquidsoap_user:=radio}
: ${liquidsoap_group:=radio}

required_files="${liquidsoap_script}"

run_rc_command "$1"
```

And it needs to be made executable.

```
pushd /usr/local/etc/rc.d/
```

```
chmod +x liquidsoap
```

```
popd
```

Also, `liquidsoap` will want to create a `pid` near the build `dir`, and the user needs permissions... (adjust as necessary).

```
mkdir -p /usr/home/radio/.opam/4.12.0/lib/liquidsoap/var/run/liquidsoap
```

```
pushd /usr/home/radio/.opam/4.12.0/lib/liquidsoap/var/run
```

```
chown radio:radio liquidsoap/
```

```
popd
```

nginx

```
vim data/radio_nginx.conf
```

For `data/radio_nginx.conf`, there needs to be the following at the very top:

```
events {}
```

And underneath that, the whole server block needs to be wrapped in an `html {}` block.

And change the listen port to whatever you'll forward to from `caddy`, like `8040`, though `80` should be fine too.

icecast

Get into `data/icecast.xml`.

```
vim data/icecast.xml
```

First, might as well adjust the location to a fun name and admin to any old email address.

I adjusted `<burst-on-connect>` to `1` and `<hostname>` to `radio.example.come` (the actual address).

The bottom of the file needs to have the user info in the `security` section, right under `changeowner` subsection:

```
<<changeowner>
  <user>icecast</user>
  <group>icecast</group>
</changeowner>
```

Change the paths to these, since the provided ones are for Linux.

```
<paths>
❏<basedir>/usr/local/share/icecast</basedir>
    <logdir>/var/log/icecast/</logdir>
❏<webroot>/usr/local/share/icecast/web</webroot>
❏<adminroot>/usr/local/share/icecast/admin</adminroot>
</paths>
```

When starting the service, there will be an annoying "error" if we don't have this file *rolls eyes*...

```
touch /etc/mime.types
```

One more thing for this. For `icecast` to work, it needs to be able to do what you tell it, like logging...

```
pushd /var/log
```

```
mkdir /var/log/icecast
```

```
chown -R icecast:icecast /var/log/icecast
```

```
chmod -R 760 /var/log/icecast
```

```
popd
```

Final Tidying Up

Still as `root` ...

```
cp /home/radio/ircradio/data/icecast.xml /usr/local/etc/
```

```
cp /home/radio/ircradio/data/radio_nginx.conf /usr/local/etc/nginx/nginx.conf
```

And we can enable the services..

```
sysrc liquidsoap_enable="YES"
```

```
sysrc nginx_enable="YES"
```

```
sysrc icecast_enable="YES"
```

And start them (and ultimately this will hopefully illuminate if any errors were made above).

```
service icecast start
```

```
service liquidsoap start
```

```
service nginx start
```

Set Up Host & Caddyfile (and cname record)

Hopefully the host doesn't need anything, actually.

Before getting to `caddy`, hop into your domain registrar and add a `cname` for the `hostname` desired, in this case `radio`. It may take a little while for the new record to propagate.

Then hop into the `Caddyfile` and add a section for `radio.domain.tld`, and reverse proxy to the jail and the listen port from the top of `nginx.conf`.

And then we are ready to finish up.

Start It!

From the jail console, start a new `tmux` session.

```
tmux
```

Change user and get to the repo directory.

```
su radio
```

```
cd
```

```
cd ircradio/
```

Run the three magical `virtualenv` steps.

Run the thing:

```
python3.8 run.py webdev
```

Then hop onto IRC and download a few songs! Then either the music will start playing, or you can restart `liquidsoap` with `root`.

Other

There are `html` files for the webpage inside `ircradio/templates`. Perhaps you'd like to adjust the files to customize it a bit and maybe indicate that you stole this setup from from someone else and it's really their hard work that made it possible.

Resources

<https://git.wownero.com/dsc/ircradio>

Revision #1

Created 31 July 2021 04:44:31 by scoob

Updated 31 July 2021 04:50:03 by scoob