Running an IRC server

Amolith

2020-12-05T16:55:00-04:00

Contents

Many people will disagree but I think IRC is still one of the best chat platforms there is for a number of reasons. However, the documentation surrounding it is sometimes lacking, commands are esoteric and can differ from server to server, some networks have stupid requirements/defaults, etc. But who says you have to join them? IRC is very easy to set up, use, and maintain and this post should be a decent guide on doing just that.

Picking an IRCd

First, `ircd` is short for *IRC daemon*; it's just a server running in the background. Second, there are a *ton* of choices, from charybdis and ngIRCd to UnrealIRCd, InspIRCd, and many others. The ircd this guide will focus on is Oragono because it's one of the simpler options yet has support for IRCv3 and comes with services out of the box.

Setup

While we could run Oragono as root or under whatever account you use, that's a stupid idea. We're going to create an `oragono` user to make sure things are properly separated.

```
1 adduser --disabled-login oragono
```

Press the enter key a bunch of times and you're good. After that, run `sudo su - oragono` to log into that user account then head to the GitHub releases page and download the latest gzipped tarball[1] for your architecture. Decompress it

---

[1] A *gzipped tarball* is just a compressed archive like a zip file

with `tar xvf oragono-*.tar.gz`. I *don't* recommend renaming the folder to something else; having the version name right there will make it easy to figure out when you need to upgrade in the future.

Copy the default config to the production config with `cp default.yaml ircd.yaml`, open it in your favourite TUI editor, and start exploring the options! The file is commented very well but I'll list a few specific options and values I recommend.

Configuration

- Obtain a TLS cert from Let's Encrypt and use it if possible. If not, use Oragono's self-signed certificates. Don't enable plaintext use.
- Consider setting Tor up and allowing users to connect through it.
- If you're using a TLS cert from Let's Encrypt (like you should be), set `sts.enabled` and `sts.preload` to `true`.
- Unless you specifically want IRC cloaks to indicate position, status, or affiliation, set `lookup-hostnames: true` and `forward-confirm-hostnames: false`.
- Always make a cool MoTD. It's essential for any IRC server.[2] I recommend using something like TAAG to come up with it.
- You *may* want to enable email authentication but it's a pain to set up properly. I haven't bothered.
- I recommend setting `channels.default-modes` to `+nts`. The `+s` flag just indicates that it's a secret channel and won't show up in the global list when someone runs `/list`. After creating a channel, if you want it publicly listed, just run `/mode -s`.
- You *may* want to uncomment `opers.admin.modes` but it can get very spammy when there are a lot of people on your server.
- If you plan to leave `history.enabled: true` and store channel message history server-side, I highly recommend setting `datastore.mysql.enabled` to `true` and going through that configuration.
- Roleplay can be fun so it's a good idea to look through that section.
- If you enabled `datastore.mysql`, also enable `history.persistent`
- For privacy reasons, I *highly* recommend setting `history.persistent.registered-channels` to `"opt-out"`. Message history will not be stored by default but the channel owner can decide to enable it if they wish. Same goes for `history.persistent.direct-messages`.
- I recommend enabling both of the options under `history.retention`.

Running the server

If you're using self-signed certs, run `./oragono mkcerts` and make sure the paths are correct in your `ircd.yaml`. Assuming your config is valid, you should

---

[2]You should definitely join mine and look at our awesome MoTD

be able to run `./oragono run` and connect to it. If you can't, make sure port 6697 is open, your credentials are correct, and nothing is wrong in the config.

You can always run Oragono in [tmux](#) or something but it would be much better to do it with [systemd, OpenRC, runit,](#) etc. I personally use systemd and this service file should go in `/etc/systemd/system/oragono.service` or something similar.

```
1  [Unit]
2  Description=oragono
3  After=network.target
4  Wants=mysql.service
5  After=network.target mysql.service
6
7  [Service]
8  Type=simple
9  User=oragono
10 WorkingDirectory=/home/oragono/oragono
11 ExecStart=/home/oragono/oragono/oragono run --conf
        /home/oragono/oragono/ircd.yaml
12 ExecReload=/bin/kill -HUP $MAINPID
13 Restart=on-failure
14 LimitNOFILE=1048576
15
16 [Install]
17 WantedBy=multi-user.target
```

Run the following commands to ensure Oragono starts when your server boots.

```
1  systemctl daemon-reload
2  systemctl enable --now oragono
```

Commands

As I said in the first section, IRC has a lot of commands and they can be confusing to work with. I still don't know everything I should. That said, here are a (very) few of the essentials:

Command	Example	Operation
/quote	/quote helpop	Send argument as a command to the server
/join	/join #channel	Join a channel
/leave	/leave	Leave a channel
/me	/me yawns	Result will look like * Amolith yawns
/query	/query amolith	Open a direct message buffer with amolith
/mode	/mode -s	Add/remove flags from a channel[3]

---

[3] See `/quote help cmodes` for all the flag options

Command	Example	Operation
/op	/op amolith	Make amolith a channel operator
/voice	/voice amolith	Let amolith speak when channel set to +M

The one command *every* Oragono oper[4] should know is /quote helpop index. It will list all the available commands so you can read through them and discover what each does.

You've reached the end of the post. You are now disallowed from telling anyone that IRC is too complicated. If you want to test a server you're setting up, feel free to use my instance of The Lounge; it's at irc.nixnet.services and any IRCd details can be entered but mine are default. Speaking of my IRC server, you should join #secluded and mention this post :D

---

[4]Short for IRC operator