Modern IRC

In 2019, IRC is still the best.

Yann Esposito

[2019-10-19 Sat]

After reviewing and testing many different chat solutions the clear winner (at least for my taste) is IRC. More precisely via those softwares:

- IRC
- ZNC (with playback and palaver module)
- weechat with replay script (terminal client)
- thelounge (web client)
- Palaver (iOS client).

Why IRC?

How to chat in 2019? Certainly with slack, or via a social media app in the browser or mobile phone app.

How geeks should chat in 2019?

This post is about my opinion on the subject after having tried many different chat solutions[1]. Here is a list of the features I think a modern solution should have:

1. **terminal client** or **terminal-like UI** (in emacs for example). All modern UI looks cool for screenshots, but if you are going to use it a lot, you will prefer density over good looking. Most web apps are terrible when looking at delivered information by number of pixels ratio.

[1] Here is a list of the chatting solutions I used for some time and finally abandoned (I certainly forgot a few ones):

- slack
- matrix (self-hosted)
- keybase
- discord
- gitter
- XMPP (both hosted by a 3rd party and self-hosted)
- IRC ← the winner

2. **multi-platform**: If you do not have a terminal at hand (or emacs) then, you should be able to get your message on your phone or via a web interface for portability.
3. **self-hosted**: you should control your data, your history, your logs, the encryption methods, etc...
4. **teams** and **direct messages**
5. **notifications**, I tend to control those a lot, but a small private team chat is one of few exception where you generally want to be notified.
6. **Frugal**. Really, we have a responsibility to do our best not to consume more resources than we really need. Chat should be about TEXT, not images, not videos, not presentations and PDF.
7. **No anti-features**: show when someone is typing, show when someone as read your messages, etc... Those functionalities increase your social insecurities feeling. They forces you to answer sooner instead of really take the time to answer correctly. You can read Digitial minimalism to know more about that subject[2].
8. **Free software**

I am quite disappointed by *modern* chat applications.

Their major problems are:

- **resource-heavy**; most those client applications (slack, gitter, riot, mattermost, etc...) easily consume more than 300MB of RAM. Most of the time those clients are electron applications.
- **not private**; most solution do not encrypt your conversations. Even if using encryption mechanism and you trust your client, and you will still reveal your social network topology.
- **anti-minimalist**; I want dense *text*. I do not want:
  - emojis,
  - images,
  - animations (gif or videos),
  - HTML/Markdown display
- **manipulative**; they try very hard to optimize engagement. This is generally achieved through FOMO[3] and social anxiety manipulations. A few examples:
  - show when someone is writing a message
  - show when someone has read a message,
  - get notified about missed messages,
  - get a ”top messages you missed”,
  - etc...
- **Prepare to EEE**[4]: Most of those ”modern” solution are all-in-one solutions. Server API + Clients with specific features. Doing it that way make it possible to provide specific features only via this ”all in one” solution. If

[2] http://www.calnewport.com/books/digital-minimalism/

[3] Fear Of Missing Out

[4] Embrace, extend, and extinguish.

you want to use another client, or if they deprecate some (like slack did by removing their IRC bridge) then you will have no choice to use their entire closed ecosystem.

About failed attempts

- Matrix: I've used Matrix, and in fact it was really good except; the server is written in python and is clearly not frugal at all. Also I wanted to delete most of the history in the DB, and it was impossible to find a working documentation explaining how to do that correctly and easily (I'm not even sure this is possible). You can easily remove some channels history from the DB, but doing the opposite, keeping the history only of some channel and removing all others doesn't appear to be easy.
- Mattermost: I've tried to install mattermost, to install it, there is no package, you need to start a shell script as root that will erase and change your nginx configuration.... Seriously...
- Rocket.chat, ho.... a nice word about the difference between community version and pro version... no thank you. I prefer something sustained by free software standards.
- I tried XMPP, it was OK. But the clients weren't really good, I could have used bitlbee, the installation looked more complex than IRC.

Finally, IRC + ZNC with replay module is the winner. It was the easiest and best solution.

1. it works
2. it is frugal
3. it is old and stable
4. it is both minimalist and feature complete

Tutorial

Here is how to have a great private self-hosted IRC server to share with a small group of people.

If this appear to be too much work for you, you should simply use a service that host a znc bouncer for you (I found some apparently free services doing that) and find an IRC server allowing you to create some private channel.

Self Host

You might use an external IRC server. But it is a lot safer to self-host it.

Self-hosting might not be easy if you are not familiar about how to do that.

1. buy a domain name
2. buy a machine (VPS, baremetal, host it at your home)
3. configure the DNS for your domain name go to your machine

4. configure letsencrypt to support wildcard hostnames
5. know how to create reverse proxy

I couldn't find a nice resource to link to with all those details. This is certainly a call to write such an article myself.

Create a reverse proxy with nginx

This is how I create new reverse proxies with nginx using a template: reverse-proxy-template.m4.

That I use with the following script: new-reverse-proxy.sh

```
#!/usr/bin/env zsh

(($#<3)) && {
  print "usage: $0:t SUB DOMAIN PORT"
  exit 1
} >&2

SUB="$1"
DOMAIN="$2"
PORT="$3"

m4 -D SUB=$SUB -D DOMAIN=$DOMAIN -D PORT=$PORT reverse-proxy-template.m4 > $SUB.$DOMAIN
```

The script will generate a reverse proxy nginx conf that I put in `/etc/nginx/sites-available/` and I link it in `/etc/nginx/sites-enabled`.

Install/configure ngircd

There are multiple IRC server. I gave my preference to ngircd because it appeared lightweight, easy to install and configure.

So use your preferred package manager to install it:

```
sudo apt-get install ngircd
```

Configure the `/etc/ngircd/ngircd.conf` file. I only show the really interesting lines for a private small IRC for a team.

```
[Global]
  Name = irc.your.domain
  Info = My Incredible IRC
  Password = privateIRCSecretPassword

[Options]
  # prevent channel creation
  AllowedChannelTypes =
[SSL]
  Certfile = /etc/letsencrypt/live/your.domain/fullchain.pem
  Keyfile = /etc/letsencrypt/live/your.domain/privkey.pem
  Ports = 6667,9999

[Channel]
  # predefined channel
  Name = #general
  Topic = Main team channel
  MaxUsers = 23

[Channel]
  Name = #status
  Topic = Status update channel
  MaxUsers = 23
```

Install/configure ZNC

Install ZNC from sources or via your package manager. I choose sources. Choose the latest version if you can.

```
> wget https://znc.in/releases/archive/znc-1.7.5.tar.gz
> tar xzf znc-1.7.5.tar.gz
> cd znc-1.7.5
> mkdir build
> cd build
> make
> make install
```

Then create your config file for example via:

```
> znc --makeconf
```

For the question, keep buffers after replay, you should certainly answer yes. To use znc web interface behind an nginx reverse proxy:

```
 <Listener listener0>
        AllowIRC = false
        AllowWeb = true
        Host = localhost
        IPv4 = true
        IPv6 = false
        Port = 10001
        SSL = false
        URIPrefix = /
</Listener>

<Listener listener1>
```

```
        AllowIRC = true
        AllowWeb = false
        IPv4 = true
        IPv6 = true
        Port = 10002
        SSL = true
        URIPrefix = /
</Listener>
```

Now you can put your ZNC behind a reverse proxy.

In order not to miss any message in your clients you should keep a bouncer running for you that will keep all IRC messages. But in order to sync this history correctly among all your different IRC clients you should install the playback module. And if you wish to receive push notification you should also add a module for your application (in my case palaver).

Playback module

In order not to miss any messages in all your clients you should add this ZNC playback module.

```
> cd ~/.znc/modules
> wget https://raw.githubusercontent.com/jpnurmi/znc-playback/master/playback.cpp
> znc-buildmod playback.cpp
```

Should create a `playback.so` in `~/.znc/modules`.

Palaver push module

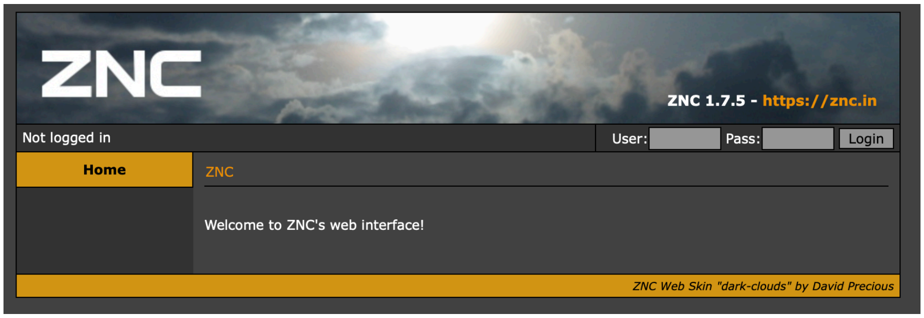
You should find the ZNC push palaver module here:

https://github.com/cocodelabs/znc-palaver

```
> git clone https://github.com/cocodelabs/znc-palaver znc-palaver
> cd znc-palaver
> znc-buildmod palaver.cpp
> cp palaver.so ~/.znc/modules/
```

Configure your IRC servers

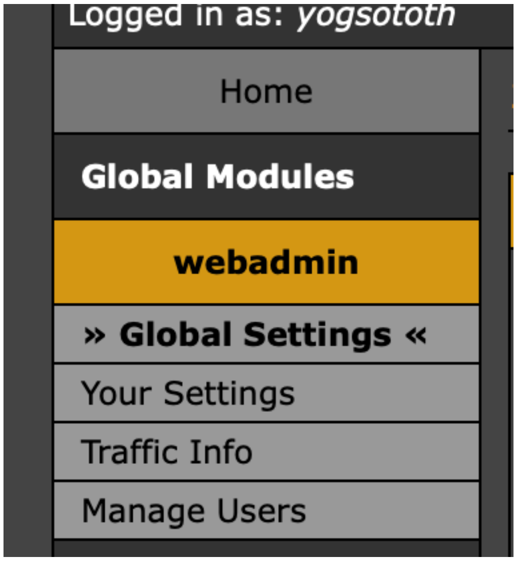
Now you should be able to reach `znc.my.domain`. You should see something like



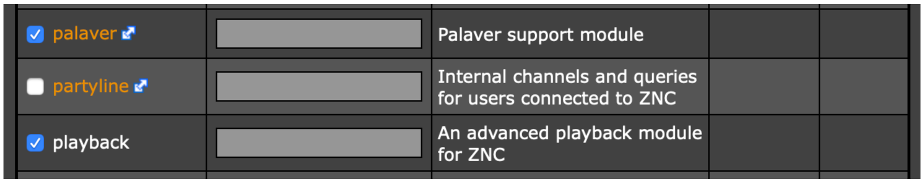
Welcome to ZNC's web interface!

ZNC Web Skin "dark-clouds" by David Precious

Login with your admin user (set during the configuration or znc). Then go to your Global settings

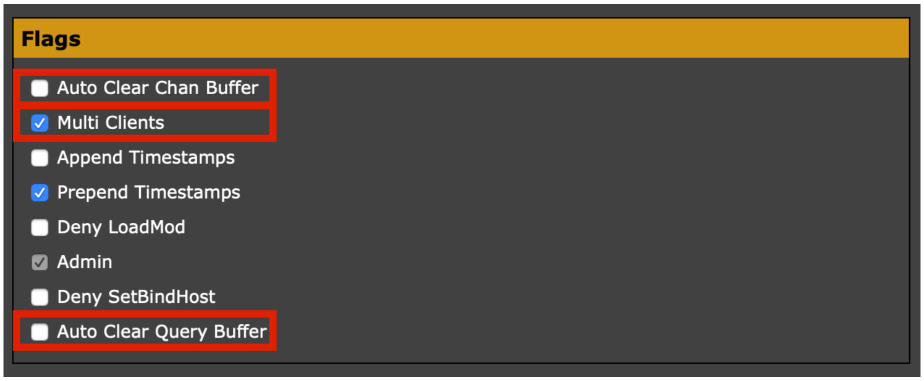


And if you scroll down you should see a list of modules. Select the playback and palaver modules and save your preferences.



☑ palaver		Palaver support module		
☐ partyline		Internal channels and queries for users connected to ZNC		
☑ playback		An advanced playback module for ZNC		

Then under the global settings, go to your User settings and scroll down to see the Flags:

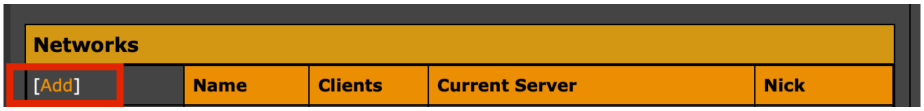


Flags

- ☐ Auto Clear Chan Buffer
- ☑ Multi Clients
- ☐ Append Timestamps
- ☑ Prepend Timestamps
- ☐ Deny LoadMod
- ☑ Admin
- ☐ Deny SetBindHost
- ☐ Auto Clear Query Buffer

Take care to unselect the ”Auto Clear Chan Buffer”, ”Auto Clear Query Buffer” and to select ”Multi Clients”. If you forget to do that, the playback plugin will not work as expected.

Finally add your IRC server to via the Network block (in your User Settings):



Networks

[Add]	Name	Clients	Current Server	Nick

From now on, you should always appear as a connected user to your IRC server. This is your ZNC bouncer reading all the messages for you even when you are not here.

Install/configure clients

weechat

Weechat the IRC client I use the most. It is terminal based, use very few resources, it is fast, dense and very nice to use.

1. add the weechat znc playback script
2. in weechat, set server capabilities

   ```
   /set irc.server_default.capabilities "account-notify,away-notify,cap-notify,multi-prefix,server-time,znc.in/server-time-iso,znc.in/self-message,znc.in/playback
   ```

3. add your server

   ```
   /server add zncnetwork znc.my.domain/6697 -ssl -username=username/zncnetwork -password=password -autoconnect
   /connect zncnetwork
   ```

4. save your confi with `/save`

More details here: https://wiki.znc.in/Weechat

thelounge

Here are the infos for installing it.

https://thelounge.chat/docs/install-and-upgrade

You can use my reverse proxy scripts to put the lounge behind a reverse proxy from your host. So you'll be able to reach `thelounge.my.domain`. Of course, connect the lounge via ZNC not directly to your IRC server.

Palaver

Using palaver should be straightfoward. There is a very clear ZNC configuration choice.

Here is its website: https://palaverapp.com

I previously used the app mutter, but it stopped to work after the iOS 13 update.

Bonus

No brainer upload file

Quite often you want to share images/files in your chat. Instead of using a public channel, I preferred to create a minimalist (223 lines of haskell) private server for this purpose only.

It is highly inspired from the image uploader example of the Yesod web framework. It is a single self-executable file + one css and jquery. The only dependency is stack.

So to install it:

1. install stack
2. `git clone https://gitlab.esy.fun/yogsototh/ymgur .`
3. follow the README instructions to launch it
4. create an nginx reverse proxy protected with basic-auth
5. share the creds to your group members
6. enjoy