

Le logiciel Pleroma, pour communiquer sur le fédivers

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 25 avril 2019

<https://www.bortzmeyer.org/pleroma.html>

Le **fédivers** (ou "*fediverse*", ou fédiverse) est l'ensemble des systèmes qui communiquent sur l'Internet en échangeant des messages typiquement assez courts, dans une logique de "*microblogging*" et en utilisant en général le format Activity Streams et le protocole ActivityPub. Il existe plusieurs logiciels qui permettent de participer au fédivers, et cet article va parler de mon expérience avec Pleroma <<https://pleroma.social/>>.

Mon résumé du fédivers vous a semble bien vague et bien nébuleux? C'est normal, le fédivers n'a pas de définition claire. Beaucoup de gens le confondent avec Mastodon, qui n'est qu'un des logiciels permettant l'accès au fédivers. En pratique, il est vrai que « fédivers » est souvent défini "*de facto*" comme « tout ce qui peut échanger des messages avec Mastodon ».

Mon compte fédivers personnel, bortzmeyer@mastodon.gougere.fr, est sur Mastodon, mais je pense qu'il est crucial d'avoir un pluralisme des logiciels utilisés. Ce serait triste de fuir les rézosocios centralisés des GAFAs pour retrouver un monopole dans le logiciel, même si c'est un logiciel libre. Et puis, je voulais gérer ma propre **instance** (ma propre machine connectée au fédivers) et l'installation et l'administration de Mastodon me faisaient un peu peur, surtout sur une petite machine. Donc, je me suis tourné vers le logiciel des pauvres, Pleroma <<https://pleroma.social/>>.

Un mot d'abord pour les utilisateurs, et je parlerai des administrateurs système par la suite. Si vous voulez expérimenter avec Pleroma, il existe au moins une instance publique en France, . Vous pouvez y créer un compte et essayer ainsi le logiciel (j'ai d'ailleurs un compte sur cette instance, bortzmeyer@pleroma.fr, même si je m'en sers peu). Contrairement à Mastodon, où le serveur et l'interface utilisateur sont développés et distribués ensemble, Pleroma sépare les deux. L'interface Web par défaut est nommée Pleroma FE (FE pour "*FrontEnd*") mais on peut en utiliser d'autres, y compris celle de Mastodon (Mastodon FE). Bon, mais on ne va parler que de Pleroma FE.

Contrairement à Mastodon FE, tous les pouètes sont sur une seule colonne, et on utilise les liens à gauche pour afficher les messages des gens qu'on suit, les messages qui nous mentionnent, les messages privés, ou bien tous les messages connus de l'instance.

Pour écrire, on tape son pouète (son message, son "toot") dans la fenêtre (celle avec le choix par défaut « *Just landed in L.A.* »), je n'ai pas encore trouvé comment le modifier. . .) Le protocole ActivityPub laisse plein de choses non décidées. Ainsi, il n'y a aucune règle sur la taille maximale des pouètes. Mastodon la limite à 500 caractères par défaut (mais l'administratrice de l'instance peut le changer), Pleroma à 5 000 (également modifiable). Le format des pouètes n'est pas non plus normalisé et Pleroma permet (ce n'est pas activé par défaut) de mettre du gras, de l'italique. . .(Les utilisatrices les plus avancé[Caractère Unicode non montré ¹]e[Caractère Unicode non montré]s noteront qu'on peut utiliser la syntaxe Markdown.)

Comme Pleroma est par défaut mono-colonne et Mastodon par défaut multi-colonnes, cela dérouté parfois ceux et celles qui viennent de Mastodon. Quelques petites particularités de l'interface Pleroma FE, listées par lord <<https://pleroma.lord.re/users/1>> :

- Sur n[Caractère Unicode non montré]importe quel pouète qui est une réponse, tu peux survoler une petite flèche en haut à gauche pour voir le pouète d[Caractère Unicode non montré]origine, sans ouvrir tout le fil.
- En parlant de fil, le petit signe plus permet d[Caractère Unicode non montré]afficher tout le fil tout en restant là où tu en étais dans ta colonne (ça insère le contenu à partir du pouète que tu as cliqué, donc une fois refermé, tu es de nouveau là où tu étais, tu ne perds pas le fil).
- Tu peux cliquer sur l'avatar d'un utilisateur pour faire apparaître son profil sans changer la vue, juste en insérant le profil au milieu de la colonne.
- Tu peux mettre des signes distinctifs sur les gens ; par exemple, sur telle personne tu peux rajouter un liseré de la couleur que tu veux pour faire ressortir ses pouètes plus distinctement. (Je me suis amusé à mettre un fond rouge vif pour tous les gens de gauche que je suis.)

Autre avantage de Pleroma par rapport à Mastodon : le moteur de recherche cherche dans tous les messages, pas seulement ceux marqués avec un croisillon.

Outre l'instance citée plus haut, il existe d'autres instances du fédivers utilisant Pleroma. La mienne est . Elle n'est **pas** ouverte au public, mais si vous demandez gentiment, je peux vous y ouvrir un compte. Les CGU de cette instance sont simples : « l'administrateur [moi] fait ce qu'il veut ». Bon, c'est pareil sur toutes les autres instances mais, ici, c'est moins hypocrite. (Sur une instance qui accepte les créations de compte à tou[Caractère Unicode non montré]te[Caractère Unicode non montré]s, pensez à éditer `priv/static/static/terms-of-service.html` pour indiquer les CGU.)

Le reste de cet article concerne plutôt les administrateurs système, qui voudraient installer une instance Pleroma sur leur machine.

Un des principaux intérêts de Pleroma par rapport à Mastodon est d'être très léger et de pouvoir tourner sur une petite machine. Dans mon cas, il s'agit d'un container LXC sur une Turris Omnia <<https://www.bortzmeyer.org/turris.html>>. J'ai installé Alpine Linux dessus, car c'est un système peu consommateur de ressources (et qui n'a pas systemd, chic).

Comme je n'ai qu'une seule adresse IPv4, le container n'a pas d'adresse IPv4 publique. Donc, j'ai configuré un relais sur la machine ayant l'adresse publique. Ce relais utilise nginx et sa configuration est celle proposée par Pleroma (`installation/pleroma.nginx` dans les sources de Pleroma), notamment :

1. Car trop difficile à faire afficher par \LaTeX

```
server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;
    ssl_certificate      /var/lib/dehydrated/certs/pleroma.bortzmeyer.fr/fullchain.pem;
    ssl_certificate_key  /var/lib/dehydrated/certs/pleroma.bortzmeyer.fr/privkey.pem;
    server_name pleroma.bortzmeyer.fr;

    # Redirection vers le container qui porte Pleroma
    location / {
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_set_header Host $http_host;

        proxy_pass http://LE.CONTAINER.À.MOI:4000;
    }
}
```

Cela a également l'avantage de permettre de gérer les certificats Let's Encrypt (ce que Pleroma ne sait pas faire seul).

Pleroma n'est apparemment distribué que sous forme de code source. Il faut donc le compiler. Il est écrit en Elixir et il faut donc un compilateur Elixir (cf. mon article à propos d'Elixir <<https://www.bortzmeyer.org/programming-elixir.html>>). Sur Alpine, Elixir est dans la section community et pas main. Il faut donc ajouter l'URL de community à /etc/apk/repositories.

Ensuite, on télécharge les sources. Le site de référence est . Le téléchargement se fait via le répertoire des versions livrées <<https://git.pleroma.social/pleroma/pleroma/releases>>, ou bien avec git si on veut la version de développement :

```
% git clone https://git.pleroma.social/pleroma/pleroma.git
```

Une fois que c'est fait, il n'y a plus qu'à suivre les instructions d'installation <<https://git.pleroma.social/pleroma/pleroma>>. (Si vous vous demandez ce qu'est ce mix, c'est l'outil de gestion de tâches d'Elixir. Un peu comme make pour C. Il permet entre autres d'installer les dépendances <<https://hexdocs.pm/mix/Mix.Tasks.Deps.Get.html>>.)

```
% mix deps.get
% mix pleroma.instance gen
What domain will your instance use? (e.g pleroma.soykaf.com) [] pleroma.bortzmeyer.fr
What is the name of your instance? (e.g. Pleroma/Soykaf) [] Unsafe
What is your admin email address? [] stephane+pleroma@bortzmeyer.org
What is the hostname of your database? [localhost]
What is the name of your database? [pleroma_dev] pleroma
What is the user used to connect to your database? [pleroma]
What is the password used to connect to your database? [autogenerated]
Writing config to config/generated_config.exs. You should rename it to config/prod.secret.exs or config/dev.secret.exs
Writing config/setup_db.psql.
```

To get started:

1. Verify the contents of the generated files.
2. Run `sudo -u postgres psql -f 'config/setup_db.psql'`.
3. Run `mv 'config/generated_config.exs' 'config/prod.secret.exs'`.

Arrivé à ce stade, le SGBD PostgreSQL doit être installé, et tourner, pour faire la suite.

<https://www.bortzmeyer.org/pleroma.html>

```
% sudo -u postgres psql -f 'config/setup_db.psql'
% mv 'config/generated_config.exs' 'config/prod.secret.exs'
```

On peut alors lancer à la main Pleroma (le code va être compilé par mix) :

```
% mix phx.server
```

Et voilà, votre Pleroma tourne. Essayez de vous connecter sur `https://MON-INSTANCE/` pour voir. Si vous n'avez pas fait proprement la redirection depuis le relais nginx, vous aurez des choses bizarres, comme des boutons présents mais inactifs.

Pour activer Pleroma sur Alpine, avec OpenRC, on copie le script proposé, `installation/init.d/pleroma` en `/etc/init.d/pleroma` et on peut alors indiquer que Pleroma doit être lancé au démarrage, avec `rc-update add pleroma`. Si on veut le démarrer tout de suite, c'est `rc-service pleroma start`.

Notez que Pleroma permet de gérer facilement deux environnements, de production et de développement, en fonction de la variable d'environnement `MIX_ENV`. Comme je n'ai qu'un environnement, j'ai défini `MIX_ENV=prod` dans mes fichiers de configuration. Sinon, il faut y penser à chaque commande :

```
% MIX_ENV=prod mix phx.server
```

Bon, Pleroma est maintenant démarré. Créons un[Caractère Unicode non montré] utilisateur :

```
% mix pleroma.user new dominique dominique@example.net --password toto
```

Et voilà, `dominique` peut désormais se connecter via l'interface Web, et envoyer des pensées profondes et des photos de chats à tout le fédivers.

Vous trouvez que le mot de passe mis est vraiment trop nul ? Laissons l'utilisateur le changer mais on peut aussi l'aider :

```
% mix pleroma.user reset_password dominique
```

Où ai-je trouvé ces commandes ? Sur l'ancien Wiki <<https://git.pleroma.social/pleroma/pleroma/wikis/Admin%20tasks>>, plus mis à jour mais qui contient des informations que je ne trouve pas facilement dans la documentation officielle <<https://docs-develop.pleroma.social/readme.html>> (Haelwenn me signale que toutes ces tâches mix sont documentées ici <<https://docs-develop.pleroma.social/api-reference.html#mix-tasks>>.) On peut aussi voir toutes les commandes avec `mix help pleroma.user`. Et il y a bien sûr l'examen du source (`lib/mix/tasks/pleroma/` qu'elle a un bon mot de passe, `dominique` peut être nommé[Caractère Unicode non montré] administrateur :

```
% mix pleroma.user set dominique --admin
```

Ou bien « modératrice » (censeur) :

```
% mix pleroma.user set dominique --moderator
```

Il n'y a pas à ma connaissance de page Web pour l'administrateur d'une instance Pleroma. On édite le fichier de configuration (`config/prod.secret.exs`, qui est du code Elixir) et on tape des commandes comme celles ci-dessus. Pour changer l'apparence de son instance Pleroma, par exemple, on regarde cette page https://docs-develop.pleroma.social/small_customizations.html. Pour contrôler le `robots.txt`, voir cette documentation <https://docs-develop.pleroma.social/Mix.Tasks.Pleroma.RobotsTxt.html>.

Qu'est-ce que Pleroma va afficher lors de son exécution? Cela se règle également dans le fichier de configuration :

```
config :logger, level: :info
```

(Si vous êtes curieu[Caractère Unicode non montré]x[Caractère Unicode non montré]se, le deux-points indique un atome Elixir.) Et si on veut tous les détails (vraiment tous) :

```
config :logger, level: :debug
```

Un des intérêts d'avoir sa propre instance, est qu'on peut fouiller dans la base de données, au-delà de ce que permet l'interface, ce qui est pratique, les capacités de recherche de pouètes de Mastodon et de Pleroma étant limitées. On se connecte à la base PostgreSQL :

```
% sudo -u postgres psql pleroma
```

Et on peut alors taper les requêtes SQL de son choix (ici, émetteur et date d'arrivée du message) :

```
pleroma=# SELECT actor,inserted_at FROM activities ORDER BY inserted_at DESC LIMIT 10;
```

actor	inserted_at
https://hostux.social/users/R1Rail	2019-04-24 15:30:38
https://hostux.social/users/R1Rail	2019-04-24 15:27:30
https://mastodon.gougere.fr/users/bortzmeyer	2019-04-24 15:25:02
https://hostux.social/users/R1Rail	2019-04-24 15:25:02
https://mastodon.social/users/jpmens	2019-04-24 15:21:20
https://mamot.fr/users/Shaft	2019-04-24 15:21:20
https://mastodon.host/users/federationbot	2019-04-24 15:18:58
https://mastodon.host/users/federationbot	2019-04-24 15:18:12
https://misskey.io/users/7rmk7gm7ew	2019-04-24 15:18:12
https://mastodon.host/users/federationbot	2019-04-24 15:16:55

(10 rows)

<https://www.bortzmeyer.org/pleroma.html>

Et si on veut tout le contenu en Activity Streams du pouète (ici, je restreins aux messages que j'ai envoyé) :

```
pleroma=# SELECT actor,inserted_at,data FROM activities WHERE actor LIKE 'https://pleroma.bortzmeyer.fr/%'
https://pleroma.bortzmeyer.fr/users/stephane | 2019-04-24 10:04:20 | {"cc": ["https://pleroma.bortzmeyer.fr/
...
```

La colonne data contient l'objet Activity Streams, en JSON.

J'ai indiqué que j'utilisais la version de développement de Pleroma. Il y a donc du nouveau régulièrement et, après un `git pull`, il faut penser à charger les dépendances nouvelles :

```
% mix deps.get
```

Si le schéma de la base de données a changé, il faut également mettre à jour votre base :

```
% mix ecto.migrate
```

(Ecto <<https://hexdocs.pm/ecto/>> est l'interface aux bases de données utilisé par Elixir.) Si vous avez un doute sur la nécessité de ces deux commandes, n'hésitez pas, lancez-les, elles sont inoffensives si le travail est déjà fait. Vous trouverez davantage d'informations sur la mise à jour de votre instance sur l'ancien Wiki <<https://git.pleroma.social/pleroma/pleroma/wikis/Updating%20your%20instance>> (pas encore trouvé où c'était sur la documentation officielle).

Et si je veux hacker Pleroma sérieusement? Voyez ce bon article <<https://github.com/wimvanderbauwhede/limited-systems/wiki/Hacking-the-Pleroma:-Elixir,-Phoenix-and-a-bit-of-ActivityPub>>, avec explications sur Elixir, Ecto, Mix, etc. Au passage, vous devrez apprendre Elixir donc les documentations sont là <<https://elixir-lang.org/docs.html>> (dont une pour débutants <<https://elixir-lang.org/getting-started/introduction.html>>.) Sinon, il y a le livre de Dave Thomas <<https://www.bortzmeyer.org/programming-elixir.html>>. Si vous trouvez des bogues dans Pleroma, signalez-les <<https://git.pleroma.social/pleroma/pleroma/issues>>.

Enfin, pour finir, quelques articles et informations sur l'administration d'un serveur Pleroma :

- Un article d'introduction à Pleroma <<https://blog.treyssatvincent.fr/qu-est-ce-que-pleroma/>>, plutôt pour les utilisateurs. C'est une traduction de l'article du développeur principal <<https://blog.soykaf.com/post/what-is-pleroma/>>.
- On peut avoir de l'aide sur le canal #pleroma du réseau IRC Freenode.
- Ou bien en utilisant une pièce Matrix, (accessible avec un navigateur Web, si vous n'avez pas de client Matrix).
- Je n'ai pas trouvé d'instance Pleroma dédiée à l'administration ou au développement de Pleroma, ni de mot-croisillon spécifique (à part bien sûr #Pleroma).
- L[Caractère Unicode non montré]a[Caractère Unicode non montré] Je développeur[Caractère Unicode non montré]se[Caractère Unicode non montré]r principal[Caractère Unicode non montré] Je de Pleroma est bien sûr joignable sur le fédivers : lain@pleroma.soykaf.com (mais, comme vous vous en doutez, ielle est assez occupé[Caractère Unicode non montré]e).

- Il y a la documentation officielle <<https://docs-develop.pleroma.social/>>.
- On trouve des choses utiles sur l'ancien Wiki <<https://git.pleroma.social/pleroma/pleroma/wikis/Admin%20tasks>>.
- Installation sur une Debian <<https://i2p.rocks/blog/the-magical-world-of-pleroma-setting-up.html>>, avec nginx en frontal. Également dans la documentation officielle <https://docs-develop.pleroma.social/debian_based_en.html>.
- Installation pour Alpine <<https://lord.re/posts/136-pleroma/>>. Également dans la documentation officielle <https://docs-develop.pleroma.social/alpine_linux_en.html>.
- *"If you read english and want to run Pleroma in a FreeBSD jail instead of a container, see Jacques' article."* <<https://adminblog.foucry.net//2019/02/23/pleroma-in-jail/>>