Mastodon

- Sébastien BOURDIN
- Aminata CISSE
- Thiepthy KANAGASABAI

Documentation d'installation de Mastodon sous Ubuntu DEBIAN 8

Contexte:

Dans le cadre de notre Master 2 MIAGE, nous devons réaliser un projet en Recherche et Développement sous la supervision de Mr. Didier COURTAUD. Notre projet est de développer un serveur Mastodon. Mastodon est un logiciel Open Source qui permet de créer un réseau social. Il peut être considéré comme l'analogue de Twitter. A l'inverse de Twitter, Mastodon est décentralisé. Chaque personne peut créer son propre serveur Mastodon et le relier à d'autres serveurs Mastodon. L'objectif de se projet est de:

- Développer un réseau social Mastodon pour la Miage d'Evry
- Le relier au serveur Mastodon français
- Rédiger une documentation pour permettre à d'autres Miages d'installer leurs serveurs pour le relier au nôtre.

> Ce document servira de guide à toute personne qui aimerait avoir une instance Mastodon sous Ubuntu DEBIAN 8.

Installer les dépendances

> Attention exécuter les commandes suivantes en root ou avec sudo

```
echo "deb http://httpredir.debian.org/debian jessie-backports main" >> /etc/
apt/sources.list
apt update && apt full-upgrade -y
apt-get install imagemagick ffmpeg libpq-dev libxml2-dev libxslt1-dev file  curl g
it pkg-config libprotobuf-dev protobuf-compiler
```

Installer NodeJs

```
curl -sL https://deb.nodesource.com/setup_6.x |     bash -
apt install nodejs
npm install -g yarn
```

Installer redis-server

```
sudo apt-get install redis-server redis-tools
```

Installer Postgresql

```
sudo apt-get install postgresql postgresql-contrib
```

Se connecter avec l'utilisateur *postgres* pour créer l'utilisateur *Mastodon*

```
sudo su - postgres
psql
CREATE USER mastodon CREATEDB;
\q
exit
```

> Créer un utilisateur mastodon sans mot de passe Attention! Il faut se connecter en root pour pouvoir exécuter cette commande

```
adduser mastodon
```

> se connecter avec l'utilisateur *mastodon*

```
su - mastodon
```

Installation de quelques dépendances pour éviter les bugs

```
sudo apt-get install libicu-dev
sudo apt-get install libldap2-dev
sudo apt-get install libidn11-dev
```

Installation de ruby

> Installation des dépendances On se connecte en root pour exécuter les commandes suivants

```
  apt install autoconf bison build-essential libssl-dev libyaml-dev libreadline6-d
ev zlib1g-dev libncurses5-dev libffi-dev libgdbm3 libgdbm-dev
```

> On se connecte ensuite avec l'utilisateur mastodon pour l'installation de rbenv

```
su - mastodon
git clone https://github.com/rbenv/rbenv.git ~/.rbenv
cd ~/.rbenv && src/configure && make -C src
echo 'export PATH="$HOME/.rbenv/bin:$PATH"' >> ~/.bash_profile
echo 'eval "$(rbenv init -)"' >> ~/.bash_profile
```

> Se déconnecter pour que les modifications soient prises en compte ensuite se connecter puis installer ruby-build

```
exit
su - mastodon
```

> Installation de ruby-build

```
git clone https://github.com/rbenv/ruby-build.git ~/.rbenv/plugins/ruby-build
```

> Installation de ruby-build 2.5.0 pour mastodon

```
rbenv install 2.5.0
rbenv global 2.5.0
```

> On peut connaitre la version de ruby en tapant cette commande:

```
ruby -v
```

> Après avoir installer toutes les dépendances, on peut procéder à l'installation de Mastodon

Installation de Mastodon

```
cd ~
git clone https://github.com/tootsuite/mastodon.git live
cd live

Pour avoir la dernière version stable, on tape la commande suivante
git checkout $(git tag | tail -n 1)
```

> On lance l'installation de blunder

```
gem install bundler
bundle install --deployment --without development test
yarn install
```

Configuration

> Ouvrons le fichier .env.production avec la commande nano

```
 cp .env.production.sample .env.production
nano .env.production
```

> Les lignes à modifier

```
REDIS_HOST=localhost
DB_HOST=/var/run/postgresql
DB_USER=mastodon
DB_NAME=mastodon_production
LOCAL_DOMAIN=domainedevotreinstance.tld
```

> On génère trois clés avec **RAILS_ENV=productionbundle exec rake secret** :

```
PAPERCLIP_SECRET=
SECRET_KEY_BASE=
OTP_SECRET=
```

> Exemple de clé générée

```
250201baf31243159865836db1826b8cf1d442ea5cc37119908cadde0d160b3f248b6e47a8035c62d7
fd9538fb3a5bcec6bc808040bc4995426a5c27b230420
```

> Ensuite on génère les clés pour les notifications Push avec **RAILS*ENV=production bundle exec rake mastodon:webpush:generate*vapid_key** On les copie ensuite dans le fichier .env.production

```
VAPID_PRIVATE_KEY=arSiUfz1Gj7c6itsvGWQ-P6owXGOynsBgZ8phEveaTk=VAPID_PUBLIC_KEY=BEX
6bhfung_oXFV6WBEiKs7-LheNmEIAIkf-gLP7x40Vrh8fise6-XduD-EclCqHnULfXeFjtSXZ0rAeAIaQD
PE=
```

> Choisir la langue par défaut

```
 DEFAULT_LOCALE=fr
```

> Passons à la mise en place de la base de données

```
RAILS_ENV=production bundle exec rails db:setup
```

> On pré-compile les fichiers CSS et JS

```
RAILS_ENV=production bundle exec rails assets:precompile
```

On met en place les scripts Systemd

> Les fichiers de configuration se trouve dans le dossier /etc qui est un dossier système. Pour pouvoir éditer ces fichiers, il faut les droits d'administrateur. On se connecte donc en root.

> Pour que Mastodon fonctionne correctement, on a besoin de trois services.

Processus web

> On ouvre le fichier **mastodon-web.service** avec la commande **nano**

```
nano /etc/systemd/system/mastodon-web.service
```

> Collez :

```
  [Unit]
 Description=mastodon-web
 After=network.target

[Service]
 Type=simple
 User=mastodon
 WorkingDirectory=/home/mastodon/live
 Environment="RAILS_ENV=production"
 Environment="PORT=3000"
 ExecStart=/home/mastodon/.rbenv/shims/bundle exec puma -C config/puma.rb
 TimeoutSec=15
 Restart=always

[Install]
 WantedBy=multi-user.target
```

Processus arrière plan

> On ouvre le fichier **mastodon-sidekiq.service**

```
nano /etc/systemd/system/mastodon-sidekiq.service
```

> Collez:

```
    [Unit]
 Description=mastodon-sidekiq
 After=network.target

[Service]
 Type=simple
 User=mastodon
 WorkingDirectory=/home/mastodon/live
 Environment="RAILS_ENV=production"
 Environment="DB_POOL=20"
 ExecStart=/home/mastodon/.rbenv/shims/bundle exec sidekiq -c 20 -q default -q mai
lers -q pull -q push
 TimeoutSec=15
 Restart=always

[Install]
 WantedBy=multi-user.target
```

Processus pour l'API

> On ouvre le fichier **mastodon-streaming.service**

```
nano /etc/systemd/system/mastodon-streaming.service
```

> Collez:

```
[Unit]
 Description=mastodon-streaming
 After=network.target

[Service]
 Type=simple
 User=mastodon
 WorkingDirectory=/home/mastodon/live
 Environment="NODE_ENV=production"
 Environment="PORT=4000"
 ExecStart=/usr/bin/npm run start
 TimeoutSec=15
 Restart=always

[Install]
 WantedBy=multi-user.target
```

> on active ensuite les services

```
 systemctl enable /etc/systemd/system/mastodon-*.service
```

> On lance mastodon

```
 systemctl start mastodon-web.service mastodon-sidekiq.service mastodon-    stream
ing.service
```

> Pour vérifier que tout fonctionne

```
 systemctl status mastodon-web.service mastodon-sidekiq.service mastodon-streaming
.service
```

```
● mastodon-web.service - mastodon-web
   Loaded: loaded (/etc/systemd/system/mastodon-web.service; enabled)
   Active: active (running) since Tue 2017-04-04 17:09:36 CEST; 6s ago
 Main PID: 9059 (bundle)
   CGroup: /system.slice/mastodon-web.service
           └─9059 puma 3.6.0 (tcp://0.0.0.0:3000) [live]

Apr 04 17:09:36 mstdn systemd[1]: Started mastodon-web.
Apr 04 17:09:37 mstdn bundle[9059]: [9059] Puma starting in cluster mode...
Apr 04 17:09:37 mstdn bundle[9059]: [9059] * Version 3.6.0 (ruby 2.3.1-p112),
Apr 04 17:09:37 mstdn bundle[9059]: [9059] * Min threads: 5, max threads: 5
Apr 04 17:09:37 mstdn bundle[9059]: [9059] * Environment: production
Apr 04 17:09:37 mstdn bundle[9059]: [9059] * Process workers: 2
Apr 04 17:09:37 mstdn bundle[9059]: [9059] * Preloading application

● mastodon-sidekiq.service - mastodon-sidekiq
   Loaded: loaded (/etc/systemd/system/mastodon-sidekiq.service; enabled)
   Active: active (running) since Tue 2017-04-04 17:09:36 CEST; 6s ago
 Main PID: 9058 (bundle)
   CGroup: /system.slice/mastodon-sidekiq.service
           └─9058 sidekiq 4.2.7 live [0 of 5 busy]

Apr 04 17:09:36 mstdn systemd[1]: Started mastodon-sidekiq.
Apr 04 17:09:39 mstdn bundle[9058]: 2017-04-04T15:09:39.668Z 9058 TID-oruz89g
Apr 04 17:09:40 mstdn bundle[9058]: Creating scope :cache_ids. Overwriting ex
Apr 04 17:09:40 mstdn bundle[9058]: 2017-04-04T15:09:40.646Z 9058 TID-oruz89g
Apr 04 17:09:40 mstdn bundle[9058]: 2017-04-04T15:09:40.646Z 9058 TID-oruz89g
Apr 04 17:09:40 mstdn bundle[9058]: 2017-04-04T15:09:40.646Z 9058 TID-oruz89g
Apr 04 17:09:40 mstdn bundle[9058]: 2017-04-04T15:09:40.647Z 9058 TID-oruz89g

● mastodon-streaming.service - mastodon-streaming
   Loaded: loaded (/etc/systemd/system/mastodon-streaming.service; enabled)
   Active: active (running) since Tue 2017-04-04 17:09:36 CEST; 6s ago
 Main PID: 9057 (npm)
   CGroup: /system.slice/mastodon-streaming.service
           ├─9057 npm
           ├─9095 sh -c babel-node ./streaming/index.js --presets es2015,stag
           ├─9096 node /home/mastodon/live/node_modules/.bin/babel-node ./str
           └─9107 /usr/bin/nodejs /home/mastodon/live/node_modules/babel-cli/

Apr 04 17:09:36 mstdn systemd[1]: Started mastodon-streaming.
Apr 04 17:09:37 mstdn npm[9057]: > mastodon@ start /home/mastodon/live
Apr 04 17:09:37 mstdn npm[9057]: > babel-node ./streaming/index.js --presets
```