

Installation of Mastodon on VM Ubuntu20

Difficulté



Difficile



The installation won't work on a container!
⇒ use a VM!

<https://docs.joinmastodon.org/admin/install/> and
<https://www.scaleway.com/en/docs/installing-mastodon-community/>

Activate the user "root".

Postfix

Installation of postfix:

<https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-postfix-on-ubuntu-18-04>

```
# apt install postfix
Reading package lists... Done
Building dependency tree
Reading state information... Done
postfix is already the newest version (3.3.0-1ubuntu0.3).
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
```

Node.js

```
# apt install curl
# curl -sL https://deb.nodesource.com/setup_12.x | bash -
```

Yarn

```
# curl -sS https://dl.yarnpkg.com/debian/pubkey.gpg | apt-key add -
# echo "deb https://dl.yarnpkg.com/debian/ stable main" | tee
/etc/apt/sources.list.d/yarn.list
```

System packages

```
# apt update
# apt install -y imagemagick ffmpeg libpq-dev libxml2-dev libxslt1-dev file
git-core \
g++ libprotobuf-dev protobuf-compiler pkg-config nodejs gcc autoconf \
bison build-essential libssl-dev libyaml-dev libreadline6-dev \
zlib1g-dev libncurses5-dev libffi-dev libgdbm-dev \
nginx redis-server redis-tools postgresql postgresql-contrib \
certbot python-certbot-nginx yarn libidn11-dev libicu-dev libjemalloc-dev
```

User mastodon

```
# adduser --disabled-login mastodon
```

Change user "root" ⇒ "mastodon"

```
# su - mastodon
```

Ruby

```
$ git clone https://github.com/rbenv/rbenv.git ~/.rbenv
$ cd ~/.rbenv && src/configure && make -C src
$ echo 'export PATH="$HOME/.rbenv/bin:$PATH"' >> ~/.bashrc
$ echo 'eval "$(rbenv init -)"' >> ~/.bashrc
$ exec bash
/// to restart the users shell
$ type rbenv
/// to check if rbenv is correctly installed
$ git clone https://github.com/rbenv/ruby-build.git ~/.rbenv/plugins/ruby-
build      /// to install ruby-build as a rbenv plugin
```

```
$RUBY_CONFIGURE_OPTS=--with-jemalloc rbenv install 2.7.2
rbenv global 2.7.2
```

Install bundler and switch back to root:

```
$ gem install bundler --no-document
$ exit
```

Database

Create the database and the user

```
# sudo -u postgres psql
CREATE USER mastodon CREATEDB;
\q
```

Modify the database

```
# sudo -u postgres psql
could not change directory to "/root": Permission non accordée
psql (12.6 (Ubuntu 12.6-0ubuntu0.20.04.1))
Type "help" for help.

postgres=# update pg_database set datallowconn = TRUE where datname =
'template0';
UPDATE 1
postgres=# \c template0
You are now connected to database "template0" as user "postgres".
template0=# update pg_database set datistemplate = FALSE where datname =
'template1';
UPDATE 1
template0=# drop database template1;
DROP DATABASE
template0=# create database template1 with template = template0 encoding =
'UTF8';
CREATE DATABASE
template0=# update pg_database set datistemplate = TRUE where datname =
'template1';
UPDATE 1
template0=# \c template1
You are now connected to database "template1" as user "postgres".
template1=# update pg_database set datallowconn = FALSE where datname =
'template0';
UPDATE 1
template1=# \q
```

Mastodon

Installation

```
# su - mastodon
```

```
$ git clone https://github.com/tootsuite/mastodon.git live && cd live  
$ git checkout $(git tag -l | grep -v 'rc[0-9]*$' | sort -V | tail -n 1)
```

```
$ bundle config deployment 'true'  
$ bundle config without 'development test'  
$ bundle install -j$(getconf _NPROCESSORS_ONLN)
```

Issue!



Your bundle is locked to mimemagic (0.3.5), but that version could not be found in any of the sources listed in your Gemfile. If you haven't changed sources, that means the author of mimemagic (0.3.5) has removed it. You'll need to update your bundle to a version other than mimemagic (0.3.5) that hasn't been removed in order to install.

Solution:

<https://discourse.joinmastodon.org/t/mimemagic-0-3-5-no-longer-available/3497/19>



```
$ gem install rake  
$ bundle update mimemagic --minor  
$ bundle config deployment false  
$ bundle update mimemagic --minor  
$ bundle config deployment true
```

```
$ yarn install --pure-lockfile
```

Configuration

```
$ RAILS_ENV=production bundle exec rake mastodon:setup  
///set "mastodon" as the db user!!  
$ exit
```

Nginx

```
# cp /home/mastodon/live/dist/nginx.conf /etc/nginx/sites-available/mastodon
# ln -s /etc/nginx/sites-available/mastodon /etc/nginx/sites-enabled/mastodon
```

Create a self signed SSL-cert

Give the FQDN of the mastodon installation as the name of the cert! (here: masto.domain.tld)

```
# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
/etc/ssl/private/mastodon_key.key -out /etc/ssl/certs/mastodon_cert.crt
```

Adjust the nginx conf file

```
# nano /etc/nginx/sites-available/mastodon
///  
map $http_upgrade $connection_upgrade {  
    default upgrade;  
    ''         close;  
}  
  
upstream backend {  
    server 127.0.0.1:3000 fail_timeout=0;  
}  
  
upstream streaming {  
    server 127.0.0.1:4000 fail_timeout=0;  
}  
  
proxy_cache_path /var/cache/nginx levels=1:2 keys_zone=CACHE:10m inactive=7d  
max_size=1g;  
  
server {  
    listen 80;  
    listen [::]:80;  
    server_name masto.domain.tld;                               <= adapt  
the domain name  
    root /home/mastodon/live/public;  
    location /.well-known/acme-challenge/ { allow all; }  
    location / { return 301 https://$host$request_uri; }  
}
```

```
server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;
    server_name masto.domain.tld;                                     <= adapt
the domain name

    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers HIGH:!MEDIUM:!LOW:!aNULL:!NULL:!SHA;
    ssl_prefer_server_ciphers on;
    ssl_session_cache shared:SSL:10m;

    # Uncomment these lines once you acquire a certificate:
    ssl_certificate      /etc/ssl/certs/mastodon.crt;               <=
adapt the file name
    ssl_certificate_key /etc/ssl/private/mastodon_key.key;         <=
adapt the file name

    keepalive_timeout    70;
    sendfile              on;
    client_max_body_size 80m;

    root /home/mastodon/live/public;

    gzip on;
    gzip_disable "msie6";
    gzip_vary on;
    gzip_proxied any;
    gzip_comp_level 6;
    gzip_buffers 16 8k;
    gzip_http_version 1.1;
    gzip_types text/plain text/css application/json application/javascript
text/xml application/xml application/xml+rss text/javascript;

    add_header Strict-Transport-Security "max-age=31536000";

    location / {
        try_files $uri @proxy;
    }

    location ~
^(emoji|packs|system/accounts/avatars|system/media_attachments/files) {
        add_header Cache-Control "public, max-age=31536000, immutable";
        add_header Strict-Transport-Security "max-age=31536000";
        try_files $uri @proxy;
    }

    location /sw.js {
        add_header Cache-Control "public, max-age=0";
        add_header Strict-Transport-Security "max-age=31536000";
        try_files $uri @proxy;
    }
}
```

```
location @proxy {
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto https;
    proxy_set_header Proxy "";
    proxy_pass_header Server;

    proxy_pass http://backend;
    proxy_buffering on;
    proxy_redirect off;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection $connection_upgrade;

    proxy_cache CACHE;
    proxy_cache_valid 200 7d;
    proxy_cache_valid 410 24h;
    proxy_cache_use_stale error timeout updating http_500 http_502 http_503
http_504;
    add_header X-Cached $upstream_cache_status;
    add_header Strict-Transport-Security "max-age=31536000";

    tcp_nodelay on;
}

location /api/v1/streaming {
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto https;
    proxy_set_header Proxy "";

    proxy_pass http://streaming;
    proxy_buffering off;
    proxy_redirect off;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection $connection_upgrade;

    tcp_nodelay on;
}

error_page 500 501 502 503 504 /500.html;
}
```

```
# systemctl restart nginx
```

Setting up systemd services

```
# cp /home/mastodon/live/dist/mastodon-*.service /etc/systemd/system/  
# systemctl daemon-reload  
# systemctl start mastodon-web mastodon-sidekiq mastodon-streaming  
# systemctl enable --now mastodon-web mastodon-sidekiq mastodon-streaming  
# systemctl status mastodon-*.service
```

Mastodon should now be available at <https://masto.domain.tld>

From:
<https://wiki.guedel.eu/> - **Wiki-Guedel**

Permanent link:
<https://wiki.guedel.eu/doku.php?id=welcome:ubuntu:mastodon>

Last update: **2021/05/14 12:56**

