Installation of Diaspora* pod on a VM Ubuntu20

Difficulté
🟩🟩🟨🟧□
Difficile

The installation won't work on a container!
Use a VM!

Sources/tutos:
https://www.howtoforge.de/anleitung/wie-man-das-dezentrale-social-media-netzwerk-diaspora-auf-debian-10-installiert/
https://angristan.fr/installer-pod-diaspora-debian-ubuntu/
https://wiki.diasporafoundation.org/Installation/Ubuntu/Focal#Configuration

Packages

```
$ sudo apt-get install postfix
$ sudo apt-get install build-essential git curl gsfonts imagemagick
libmagickwand-dev nodejs redis-server libssl-dev libcurl4-openssl-dev
libxml2-dev libxslt1-dev libpq-dev
$ sudo apt-get install postgresql
```

```
# systemctl start redis-server
# systemctl enable redis-server
# systemctl start postgresql
# systemctl enable postgresql
```

Users and preparation of the database

Database

```
$ sudo -u postgres psql
```

```
postgres=# CREATE USER diaspora WITH CREATEDB PASSWORD
'<password_of_diaspora>';
```

Modification of the database to avoid issues due to UTF8:

```
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
```

Diaspora

```
$ sudo adduser --disabled-login diaspora
```

Install RVM and Ruby

```
$ sudo -iu diaspora
$ gpg2 --recv-keys 409B6B1796C275462A1703113804BB82D39DC0E3 7D2BAF1CF37B13E2069D6956105BD0E739499BDB
$ find .gnupg/ -type d -exec chmod 750 {} \;
$ find .gnupg/ -type f -exec chmod 640 {} \;
$ curl -L https://s.diaspora.software/1t | bash
```

```
nano ~/.bashrc         /// and add at the end:
[[ -s "$HOME/.rvm/scripts/rvm" ]] && source "$HOME/.rvm/scripts/rvm"
```

Logout and re-login:

```
$ exit
$ sudo -iu diaspora
```

```
$ rvm autolibs read-fail   => install as "root" the missing dependancies
$ rvm autolibs read-fail
$ rvm install 2.6
$ ruby -v
```

Download and configure Diaspora*

```
$ cd ~
$ git clone -b master https://github.com/diaspora/diaspora.git
$ cd diaspora
$ cp config/database.yml.example config/database.yml
$ cp config/diaspora.toml.example config/diaspora.toml
```

```
$ nano config/database.yml

  port: 5432
  username: "diaspora"
  password: "the_password_of_user_diaspora"
  encoding: unicode
```

```
$ nano config/diaspora.toml     ###### let the "whitespaces" like in the
commented sections!!

[configuration.environment] ## Section
url: "https://sub_domain.the_domain.tld/"
certificate_authorities: '/etc/ssl/certs/ca-certificates.crt'
require_ssl: true

[configuration.server]  ## Section
rails_environment: 'production'

[configuration.mail]  ## Section
enable = true
sender_address = "notification@diaspora.domain.tld"
method = "sendmail"
```

bundle

```
$ gem install bundler
$ script/configure_bundler
$ bin/bundle install --full-index
  ...."Bundle complete! 142 Gemfile dependencies, 234 gems now
installed......"
```

Database setup

```
$ RAILS_ENV=production bundle exec rake db:create db:migrate
$ RAILS_ENV=production bin/rake assets:precompile
```

Starting diaspora* from tmux

```
~/diaspora$ tmux
$ pwd
/home/diaspora/diaspora
$ ./script/server
```

There should not be any error messages.

- CTRL+Z to stop diaspora*
- `exit` to close the tmux session

Nginx

Commands done from the user with `sudo` permissions

Installation

```
$ sudo apt-get install nginx
```

SSL certificate

Create a self-signed SSL certificate. Give the FQDN of the Diaspora* installation as the name of the cert! (here: diaspo.domain.tld)

```
$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
/etc/ssl/private/diaspora_key.key -out /etc/ssl/certs/diaspora_crt.crt
```

Configuration of nginx

```
$ sudo nano /etc/nginx/sites-available/diaspora
```

Adapt the domain name (“diaspo.domain.tld” here) and the file names of the cert+key:

```
upstream diaspora_server {
 server unix:/home/diaspora/diaspora/tmp/diaspora.sock;
}

server {
  listen 80;
  listen [::]:80;
  server_name diaspo.domain.tld;
/// adapt domain name
  return 301 https://diaspo.domain.tld$request_uri;
/// adapt domain name

  access_log /dev/null;
  error_log /dev/null;
}

server {
  listen 443 ssl http2;
  listen [::]:443 ssl http2;
  server_name diaspo.domain.tld;
/// adapt domain name

  access_log /var/log/nginx/dspr-access.log;
  error_log /var/log/nginx/dspr-error.log;

  ssl_certificate /etc/ssl/certs/diaspora_crt.crt;
/// adapt file name
  ssl_certificate_key /etc/ssl/private/diaspora_key.key;
/// adapt file name

  ssl_protocols TLSv1.2;
  ssl_ciphers EECDH+CHACHA20:EECDH+AESGCM:EECDH+AES;
  ssl_ecdh_curve X25519:P-521:P-384:P-256;
  ssl_prefer_server_ciphers on;
#  ssl_stapling on;
#  ssl_stapling_verify on;
  resolver 80.67.169.40 80.67.169.12 valid=300s;
  resolver_timeout 5s;
  ssl_session_cache shared:SSL:10m;

  root /home/diaspora/diaspora/public;

  client_max_body_size 5M;
  client_body_buffer_size 256K;

  try_files $uri @diaspora;

  location /assets/ {
    expires max;
    add_header Cache-Control public;
  }

  location @diaspora {
    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 Host $http_host;
    proxy_redirect off;
    proxy_pass http://diaspora_server;
  }
}
```

```
$ sudo ln -s /etc/nginx/sites-available/diaspora /etc/nginx/sites-enabled/diaspora
$ sudo systemctl restart nginx
$ sudo systemctl status nginx
```

Services

Running diaspora* over services is for me more comfortable than running over a tmux console.

```
$ sudo nano /etc/systemd/system/diaspora.target
and enter:

[Unit]
Description=Diaspora social network
Wants=postgresql.service
Wants=redis-server.service
After=redis-server.service
After=postgresql.service

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

```
$ sudo nano /etc/systemd/system/diaspora-web.service
and enter:

[Unit]
Description=Diaspora social network (unicorn)
PartOf=diaspora.target
StopWhenUnneeded=true

[Service]
User=diaspora
Environment=RAILS_ENV=production
WorkingDirectory=/home/diaspora/diaspora
ExecStart=/bin/bash -lc "bin/bundle exec unicorn -c config/unicorn.rb -E production"
Restart=always

[Install]
WantedBy=diaspora.target
```

```
$ sudo nano /etc/systemd/system/diaspora-sidekiq.service
and enter:

```

```
[Unit]
Description=Diaspora social network (sidekiq)
PartOf=diaspora.target
StopWhenUnneeded=true

[Service]
User=diaspora
Environment=RAILS_ENV=production
WorkingDirectory=/home/diaspora/diaspora
ExecStart=/bin/bash -lc "bin/bundle exec sidekiq"
Restart=always

[Install]
WantedBy=diaspora.target
```

```
$ sudo systemctl daemon-reload
$ sudo systemctl enable diaspora.target diaspora-sidekiq.service diaspora-web.service
$ sudo systemctl status diaspora.target diaspora-sidekiq.service diaspora-web.service
```

Reboot.

Diaspora* should now be available over the web browser at https://diaspo.domain.tld“

From:
https://wiki.guedel.eu/ - **Wiki-GuedeL**

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

Last update: **2021/05/15 17:47**