Install and configure yoru first Pleroma Server

https://www.youtube.com/embed/ufVMqBToGno

Installing Pleroma

What is it?

With the turmoil over the past few weeks at twitter, and the non-stop invasions of your privacy from facebook, advertisers, and anyone else with any half-hair-brained idea, the world is turning to social media platforms that aren't owned and operated by a single global corporation, but instead are operated by individuals or smaller groups, and whose servers are federated (connected through linking of sorts). The "fediverse" has many applications and options for people liek us, the self-hosters, the open source advocates, to run. Pleroma is a slightly lighter weight alternative to running a full -blown Mastodon instance, and isn't too difficult to get up and running in a relatively short amount of time.

The cool thing about these federated servers, is you can join one, create an account, and still follow your favorite people from around the world regardless of which server they are on. If the server you are hosted on shuts down, there are also things in place to help migrate your profile to a new server, and keep your follows and followers intact.

Additionally, you can join a server for "local" content that is of interest to you. Tech, Open Source, Art, Dance, whatever you might be interested in, there is likely a server instance running that holds that interest as a central theme. If not, then start your own. Maybe you want an instance for your family and / or friends. It's really up to you to federate, or not federate; and up to you to determine what your theme is.

What You'll Need

- Docker and Docker Compose

- NGinX Proxy Manager (or reverse proxy of your choice)
- A Domain Name that you own and can create A-Records for
- A Machine to install everything on, and run your Pleroma from (this can be a system in your home or business, or a VPS on a host like Digital Ocean, Linode, etc).
- About 20 Minutes of your Time

Installing Docker-CE, Docker-Compose, NGinX Proxy Manager, and Portainer-CE

Installation via a Simple Script

You can easily install Docker-CE, Docker-Compose, Portainer-CE, and NGinX Proxy manager by using this quick install script I created and maintain on Github. Just use the command:

`wget https://gitlab.com/bmcgonag/docker_installs/-/raw/main/install_docker_nproxyman.sh`

To download the script to your desired host.

Change the permissions to make the script executable:

`chmod +x ./install_docker_nproxyman.sh`

and then run the script with the command:

`./install_docker_nproxyman.sh`

When run, the script will prompt you to select your host operating system, then will ask you which bits of software you want to install.

Simply enter 'y' for each thing you want to install.

At some point, you may be asked for your super user (sudo) password as well.

Allow the script to complete installation.

At this point, you might want to log out and back in, as this will allow you to use the `docker` and `docker-compose` commands without the need of sudo in front of them.

ions here

Install Pleroma

You need to setup a subdomain, or domain name for your Pleroma site. Create an A-record through your DNS registrar, and put the public IP address for your server location. If this is in your home, you can use a site like http://ipchicken.com to see your public IP.

> Note, some ISPs block ports 80 and 443 to make it difficult to run servers from your home / work.

Once you've created the A Record, you'll also want to forward ports 80 and 443 in your router / firewall from the external internet (WAN) to the private (internal / LAN) IP address of the machine you're running NGinX Proxy Manager on.

Once the A-Record is updated in DNS, you should be able to browse to your selected subdomain or domain URL and get the generic "Congratulations" page for NGinX Proxy Manager.   If so, you're doing great! We'll come back to NGinX Proxy Manager in a bit. If not, please double check your setup.

> Subdomain -→ you Public IP Address -→ port 80 / 443 on your firewall -→ Private IP where NGinX Proxy Manager is running.

Be patient, as it can take anywhere from 10 minutes to 48 hours for some DNS systems to be updated.

For now, we need to clone the Pleroma repository. We can do this with the command:

`git clone` `https://git.pleroma.social/pleroma/pleroma-docker-compose.git`

If you get an error about the git command, you can install git with the commands:

Debian / Ubuntu

`sudo apt install git -y`

Fedora

`sudo dnf install git`

OpenSuse

`zypper install git`

Arch

`pacman -S git`

Once you've cloned the repository, we'll want to edit a few files inside the repository folder. Move into the folder with the command:

```
cd pleroma-docker-compose/
```

And open the pleroma.env file with the nano text editor using the command:

```
nano environments/pleroma/pleroma.env
```

In this file, you should change a few things.

```
DB_USER=pleroma
DB_PASS=<your pleroma db password here>
DB_HOST=pleroma-db
DB_NAME=pleroma
INSTANCE_NAME=Pleroma
ADMIN_EMAIL=<your email here>
NOTIFY_EMAIL=<your email here?
DOMAIN=<your pleroma site url here>
PORT=4000
```

In the above file, you need to replace the items surrounded with greater than and less than signs "<" and ">" with the information for your Pleroma site. Once done, save with CTRL + O, then Enter, and exit with CTRL + X.

Next, we need to edit the Pleroma-db environment file: We will open it in nano with the command:

```
nano environments/pleroma-db/postgres.env
```

And, again, change the value for your Pleroma db to be the exact same as the value you entered in the pleroma environment variable file, and save with CTRL + O, then Enter, and exit with CTRL + X.

Now, let's edit our "docker-compose.yml" file slightly. We'll open it with nano again using the command:

```
nano docker-compose.yml
```

Edit the ports for the Pleroma web page if needed, but if you want to reach the site from a machine other than the one you are creating Pleroma on, you need to remove the

```
127.0.0.1:
```

from the mapping of the port for the "pleroma" service only. Leave the "127.0.0.1" on the mapping for the "pleroma-db" service, as this tells the web application how and where to connect to the Postgres database.

Finally, we are ready to bring up our Pleroma site with the commands:

```
docker-compose up -d && docker-compose logs -f
```

"docker-compose up -d" tells the Pleroma service to start, and continue running int he background.

"docker-compose logs -f" tells the shell to show the logs of the service as it starts after the initial container is up and running.

You can stop the logs at any time with the CTRL + C hotkey combination, and the service will continue to run in the background.

Now you need to add an admin user:

`docker exec -it pleroma /bin/sh`

`cd`

`./bin/pleroma_ctl user new <your username> <your-email> --password <your-users-password-here> --admin -y`

You can now exit the shell in the container with the `exit` command, and move to your NGinX Proxy Manager (or reverse proxy software of your choice).

Setting up the Reverse Proxy

Go into your NGinX proxy Manager (or reverse proxy of your choice). Create a new Proxy Host, and give the proxy the domain name you have chosen for your site.

> You should have already pointed this domain name to the public IP address where your server is being hosted.

Make sure to press Tab or Enter to make the domain entry a chip in NGinX Proxy Manager. Next, move to the IP Address field and enter the private IP address where your Pleroma server is running. If it is running on the same machine as your NGinX Proxy Manager, you can enter "localhost". Now move to the Port number field, and enter 4000 (unless you changed the port number during the setup process, in which case you should use that port number here.)

Finally, move to the SSL tab, and select "Request a New Certificate" from the drop-down, tick the options for Force SSL, and HTTP/2 support, then enter your email address in the assigned field, and tick the option to accept the LetsEncrypt Terms of Service. Click 'Save'.

The pop-up box should close with no errors. You should now be able to reach your Pleroma server via SSL, and login.

Set Pleroma to allow Configuration via the Web UI

From the folder where you docker-compose.yml file is located, use the following command to edit the configuration file:

`nano volumes/pleroma/config.exs`

In the file, you need to add the text below after the

`import config`

line.

```
config :pleroma, Pleroma.Web.Endpoint,
url: [host: "your-pleroma-url-here", scheme: "https", port: 443]


config :pleroma, Pleroma.Repo,
adapter: Ecto.Adapters.Postgres,
username: "pleroma",
password: "your-db-password-here",
database: "pleroma",
hostname: "pleroma-db"


config :pleroma, configurable_from_database: true
```

Make sure to replace the db-password, and pleroma-url with your Postgres db password, and your Pleroma domain name.

Save and close with CTRL + X, then Y to confirm, then enter the docker container shell with the command:

`docker exec -it pleroma /bin/sh`

> You can alternatively use Portainer to access the shell via your web browser - check out the video on how to do it.

and run the commands:

`cd`

`./bin/pleroma_ctl config migrate_to_db`

Give it time to complete.

Now exit the shell, and once again we will edit the configuration file to allow registration to your server, and to federate your server. If you don't want to allow registration, or federation, change the 'true' to 'false' in each place.

Enter the command:

`nano volumes/pleroma/config.exs`

and copy / paste the following below the other text in the configuration file.

```
config :pleroma, :instance,
  registrations_open: true,
  federating: true
```

Save your changes wiht CTRL + X, then Y to confirm.

Now restart your Pleroma install with `docker-compose restart`.

Give it a couple of minutes to come fully up, and then go back to your Pleroma site, and refresh. You should now be able to start editing the configuration from the Web UI.

Check out the video for a full guide on getting started.

Support my Channel and Content

Support my Channel and ongoing efforts through Patreon:

https://www.patreon.com/bePatron?u=234177

Revision #2
Created 3 January 2023 16:10:19 by Brian McGonagill
Updated 3 January 2023 16:57:53 by Brian McGonagill