Lemmy - Reddit Alternative

- Lemmy - A Self Hosted, Open Source Alternative to Reddit

Lemmy - A Self Hosted, Open Source Alternative to Reddit

https://www.youtube.com/embed/6lQFZvCCe7U

If you've watched any number of my videos, you'll know that I really enjoy Reddit. It's my kind of social media platform. I subscribe to the content areas I like, and only get content from those areas. The sub-reddits are generally well monitored, so they keep out the junk and useless stuff I don't care about, and I get a ton of ideas for videos from the various sub-reddits I subscribe to.

In my professional life (my day job) I am a Product Manager, which means I set the roadmap for new features, and what items take priority. I've been looking for a tool to setup and get direct input from my users on what features they would like to see enhanced, or added to the system. One of the best tools I can get is a simple system for a user to enter an idea, with a bit of detail that allows other users to vote on the idea, add more details, and potential more ideas to really give the best end product.

Thta's where Lemmy comes in. I love how Reddit has a very simple threading system built in. You subscribe to the sub-reddit (sub-lemmy) you want, then enter a post. Others can vote your post up or down, and add their own flare and comments to the post.

So today, we'll setup Lemmy, and I'll walk you though the pitfalls and setup issues I came across.

What You'll Need

- Docker-CE and Docker-Compose
- (optional) NGinX Proxy Manager (or other proxy server you prefer) if you want to reach your site from an domain and / or outside your local network.
- About 20 minutes of your time.

Installation

Installing Docker-CE and Docker-Compose

If you already have Docker and Docker-Compose installed, feel free to skip down to the next section.

You may want to install some pre-requisite softwre as well:

Debian / Ubuntu

```
sudo apt install git curl wget
```

Fedora / Redhat

```
dnf install git curl wget
```

Arch

```
sudo pacman -Sy git curl wget
```

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 permissios 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.

For instance, you may want to answer 'y' to NGinX Proxy Manager, and Portainer-CE if you don't already use these in your system.

At some point, you'll 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.

Installing Lemmy

Having docker-compose setup and ready, we'll create our folder structure first. If you don't already have a "docker" folder setup, I like to create a parent "docker" where I keep all of my other docker folders so I only have to zip up and backup the parent "docker" folder to get a solid backup of my working system.

Next, create a folder called "lemmy" inside the "docker" folder.

`mkdir -p docker/lemmy`

Now move into our "lemmy" folder with:

`cd docker/lemmy`

Next, we'll pull down two files from the LemmyNet github site. The "docker-compose.yml" and the accompanying "lemmy.hjson". You may need to install `wget` if you don't already have it installed. It's usually a simple command to install it.

In Ubuntu / Debian use:

`sudo apt install wget`

In Redhat / Fedora/ Centos use:

`sudo dnf install wget`

Now download the two files with the commands:

`wget https://raw.githubusercontent.com/LemmyNet/lemmy/main/docker/docker-compose.yml`

`wget https://raw.githubusercontent.com/LemmyNet/lemmy/main/docker/lemmy.hjson`

Next, we need a folder for the picture storage of our site. We'll create it, and then change the ownership of it, to allow the container to write the files to the folder. Use the commands:

`mkdir -p volumes/pictrs`

to make the new folder. And

`sudo chown -R 991:991 volumes/pictrs`

to change the ownership of the folder as needed.

Now we'll check out our `docker-compose.yml` file for any potential changes we may need / want to make.

`nano docker-compose.yml`

```
version: '2.2'
```

```
services:
  postgres:
    image: postgres:14-alpine
    environment:
      - POSTGRES_USER=< lemmy >
      - POSTGRES_PASSWORD=< lemmy >
      - POSTGRES_DB=lemmy
    volumes:
      - ./volumes/postgres:/var/lib/postgresql/data
    restart: always

  lemmy:
    image: dessalines/lemmy:0.16.4
    ports:
      - "8536:8536"
      - "6669:6669"
    restart: always
    environment:
      -
RUST_LOG="warn,lemmy_server=info,lemmy_api=info,lemmy_api_common=info,lemmy_api_crud=info,lemm
y_apub=info,lemmy_db_schema=info,lemmy_db_views=info,lemmy_db_views_actor=info,lemmy_db_views_
moderator=info,lemmy_routes=info,lemmy_utils=info,lemmy_websocket=info"
    volumes:
      - ./lemmy.hjson:/config/config.hjson
    depends_on:
      - postgres
      - pictrs
□
  lemmy-ui:
    image: dessalines/lemmy-ui:0.16.5
    ports:
      - "1235:1234"
    restart: always
    environment:
      - LEMMY_INTERNAL_HOST=lemmy:8536
      - LEMMY_EXTERNAL_HOST=< your hosts ip address here >:8536
      - LEMMY_HTTPS=false
    depends_on:
      - lemmy
```

```
  pictrs:
    image: asonix/pictrs:0.3.1
    ports:
      - "127.0.0.1:8537:8080"
      - "127.0.0.1:6670:6669"
    user: 991:991
    volumes:
      - ./volumes/pictrs:/mnt
    restart: always
```

In the above file, you'll want to change the database username and password, and the host ip address of your install. I have indicated items you want to change by surrounding them with the angle-braces `<` and `>`.

Save the file with CTRL + O, press Enter to confirm, and then exit the nano text editor with CTRL + X.

Now, we'll move to the configuration file for our Lemmy install. We need to make a few changes / additions in it as well. We'll open the file called "lemmy.hjson" with our nano editor.

`nano lemmy.hjson`

```
{
  # for more info about the config, check out the documentation
  # https://join-lemmy.org/docs/en/administration/configuration.html

  setup: {
    # username for the admin user
    admin_username: < "AUserName" >
    # password for the admin user
    admin_password: < "AStr0n6LoN6Pas5w0r8!" >
    # name of the site (can be changed later)
    site_name: < "A Name You Want on your Lemmy Site" >
  }

  opentelemetry_url: "http://otel:4137"

  # the domain name of your instance (eg "lemmy.ml")
  hostname: < "lemmy.supergreat.org" >
  # address where lemmy should listen for incoming requests
  bind: "0.0.0.0"
  # port where lemmy should listen for incoming requests
```

```
  port: 8536
  # settings related to the postgresql database
  pictrs_config: {
    url: "http://pictrs:8080"
    api_key: "API_KEY"
  }
  database: {
    # name of the postgres database for lemmy
    database: "lemmy"
    # username to connect to postgres
    user: < "lemmy" >
    # password to connect to postgres
    password: < "lemmy" >
    # host where postgres is running
    host: "postgres"
    # port where postgres can be accessed
    port: 5432
    # maximum number of active sql connections
    pool_size: 5
  }
  slur_filter:
    '''

(fag(g|got|tard)?\b|cock\s?sucker(s|ing)?|ni((g{2,}|q)+|[gq]{2,})[e3r]+(s|z)?|mudslime?s?|kike
s?|\bspi(c|k)s?\b|\bchinks?|gooks?|bitch(es|ing|y)?|whor(es?|ing)|\btr(a|@)nn?(y|ies?)|\b(b|re
|r)tard(ed)?s?)
    '''
  # optional: email sending configuration
  email: {
    # hostname (smtp.example.com) and port (25, 465, 587, etc) of the smtp server
    smtp_server: < "smtp.example.com:587" >
    # login name for smtp server
    smtp_login: < "my@example.com" >
    # password to login to the smtp server
    smtp_password: < "a-R3a11y-L0n6-str0NG-pa55wORd" >
    # address to send emails from, eg "noreply@your-instance.com"
    smtp_from_address: < "my@example.com" >
    # whether or not smtp connections should use tls (true | false)
    use_tls: < true >
    # Whether or not smtp connections should use tls. Can be none, tls, or starttls
```

```
    tls_type: < "starttls" >
  }
}
```

As with the docker-compose.yml file, I have again put angle-braces `<` and `>` around items you'll likely want to change or comment out.

Once you've completed your changes, again save the file with CTRL + O, press Enter to confirm, and then exit the nano text editor with CTRL + X.

Now, let's bring up our new Lemmy container and watch the logs for erros with the command:

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

Hopefully, you see no errors. You can get out of the logs with the hotkey combination `CTRL + C`.

You should now be able to navigate to your host system's IP address on port 1235 to access your new lemmy site.

I went to `http://192.168.10.112:1235`

If your site loads, well done!

Reverse Proxy

You may want to setup a reverse proxy to your site as well before you begin through the first run wizard. If you intend to access your invoicing software from outside your network, or via a domain / subdomain, definitely consider the need for a reverse proxy entry.

i use NGinX Proxy Manager for my setup. The steps you need to take are:

1. Get NGinX Proxy Manager installed in docker (my script can do this for you if you used it to install docker and docker-compose earlier.)
2. Port forward 80 and 443 from outside your network (the Internet / WAN) to inside your network to the host IP where you have NGinX Proxy Manager running (or the LAN).
3. Get a domain name from a good provider that gives you access to create A-Records and CNAME records in DNS for your domain.
4. Create a sub-domain of *.yourdomain.<org,com,net,biz,etc>, and point it with an A-record to your pulbic IP address.
5. Now wait the preset amount of time for DNS to update (usually about 10 minutes, but sometimes longer), then test to make sure *.yourdomain.<org,com,net,biz,etc> (if your domain is supergreat.org, then you can try test.supergreat.org for example... really any sub-domain should work if you've pointed *.supergreat.org at your public IP) can reach your NGinX Proxy Manager host. You'll get a generic 'Congratulations' page if it's working.
6. Create an NGinX Proxy Manager entry to handle requests for your Lemmy install at the subdomain of your choice. Let's say you want it to be "lemmy.supergreat.org".

In NPM add a new proxy host, and put that subdomain in the URL field, then press Tab or Enter to save that entry as a chip.

In the IP field, you can enter the local docker0 IP of the container for crater NGinX, or you can use localhost. If the crater app is running on a different machine than your NGinX Proxy Manager, you'll use the private IP of the host where Crater is running.

Next enter the port number for your NGinX container. We used 1235 in our example today, so if you followed that example, that's the port you want to enter.

Now tick the box for Websocket support, and Block common exploits, then Save.

7. Check that your Lemmy site opens, by clicking the URL in the NPM listing for proxy hosts.
8. If it does, let's edit the entry, and make it secure with SSL. Click the 3-dots at the right side of the entry, and select 'Edit' from the pop-up menu.

   Go to the 'SSL' tab, and select the drop-down that says 'None'. Now select 'Request a New Certificate' from the drop down, and then select the 'Force SSL' tick box, and the 'HTTP/2' tick box. Tick the box to accept the TOS of LetsEncrypt, and make sure your email is in the box.

   Now click Save. If all goes well, your edit window will close without errors, and you can now test again. You should now be accessing your Crater install via SSL.

Now, for a final test that your install is working with no issues, you'll want to login with the credentials you set as the admin user in the `lemmy.hjson` file. Now, click your username in the upper right, and select "Settings". Go through and setup your profile, then click Save. If it saves with no issues, you're good to go. If you get an error, you'll want to check your .hjson setup (particularly the email section) and make sure everything is correct. Make any changes, then do the following commands to bring down the app, and then bring it back up with your changes.

`docker-compose down`

Wait for it to stop, then do

`docker-compose up -d` again.

Once you can successuflly save changes to your profiel, you should be set and ready to start getting everything setup on your new Lemmy site.

Don't forget to say thanks to the team that makes Lemmy! They are super folks, and remeber to look for ways you can support the project.

Support my Channel and Content

Support my Channel and ongoing efforts through Patreon:

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