WriteFreely

A clean, simple blogging and journaling platform

- Install WriteFreely

Install WriteFreely

https://www.youtube.com/embed/pR4QATSRyXA

Sometimes you just need a writing tool that will allow you to get to work. Not something that will take you hours of setup and configuration, and theming, and layout, and on, and on, andon.... My friends, WriteFreely is an open source, self hosted platform that does just that. We'll set this up using a nice docker image a community member has created, and we'll make our time worthwhile once again.

Whether you are documenting the steps you use to change the oil on your car, or set the alarm in your home, or setup an entire domain with users and VLANs, and VPNs, and neighborhood hotspots, the most important part is just starting. Sometimes setting up software to help document it all can be the real hindrance to getting any project off the ground. With WriteFreely, we'll remove that hindrance.

What you'll need

- Docker and Docker Compose
- (Optional) NGinX Proxy Manager or some other reverse proxy application.
- (Optional) A Domain Name that you own and can setup DNS A-records for
- About 20 minutes of your time

Installation

Installation of Docker and Docker Compose 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.

Install and Configure Write Freely

Let's start by creating our folder structure for our installation. I like to have a parent level folder called "docker" that I then put all of my docker applications inside of. each in their own folders.

`mkdir -p docker/write-freely`

Now let's move into our new folder with

`cd docker/write-freely`

want to create our docker-compose.yml file that tells docker which image(s) to pull, and how to run our application.

`nano docker-compose.yml`

Now, use copy / paste to paste the following code block into that new file:

```
Version: '3.3'
services:
  write-freely:
    image: nephatrine/write-freely:latest
    container_name: write-freely
    environment:
      TZ: America/New_York
      PUID: 1000
      PGID: 1000
    ports:
      - "70:70/tcp"
      - "8080:8080/tcp"
    volumes:
      - ./write-freely:/mnt/config
```

In the above code block, you'll want to adjust a few items:

- Make sure you set the TZ: variable to your local timezone. In my case, I changed it to 'America/Chicago'
- on the ports, you'll likely want to change the left side port number from 8080 to some other, less common port. In my case I used 8250, so my mapping looked like this when done: '8250:8080/tcp'
- If you know your user id and group id are not 1000 on the system you are installing on, then you'll want to change those numbers to correspond to your user id and group id numbers as well, but most likely 1000 is correct.

Now save the file with CTRL + O, then press Enter to confirm, and use CTRL + X to exit the nano editor.

We are finally ready to run our application.

```
docker compose up -d
```

Once you run the above command, you'll see the image being pulled down, and the container being started up. Wait about 20 seconds or so after you are retunred to the normal terminal prompt, then open your favorite, modern, web browser, and browse to the ip address of your host machine / server where you are running the application, and the port number you used on the left side of the second port mapping above.

In my setup I went to http://192.168.10.60:8250

You should see the main default page of WriteFreely. IF so, that's great, but we still have a bit of work to do.

Create an Admin User

Now, let's create an admin user. The author gives us a nice, and simple little command to do this, but first we need to setup a special configuration file for WriteFreely. In my video I did this through the mapped local volume, but here we'll do it through the docker container bash terminal.

To enter the bash environment of your running container enter the following command. Notice that the prompt will change when you do.

```
docker exec -it write-freely /bin/bash
```

Now, let's move into the folder where we need to copy the .ini file already generated.

```
cd /mnt/config/etc
```

Next, we'll create a new folder called 'writefreely':

```
mkdir writefreely
```

And we'll copy the existing file 'writefreely.ini' to this folder, renaming it to 'config.ini'

`cp writefreely.ini ./writefreely/config.ini`

Finally, let's edit our new config.ini file

`nano ./writefreely/config.ini`

Again, in the video there were a couple of things I said not to change, but we will actually want to change them if you are going to access your writefreely install from a fully qualified domain name (outside your own network).

```
[server]
hidden_host          =
port                 = 8080
bind                 = 0.0.0.0
tls_cert_path        =
tls_key_path         =
autocert             = false
templates_parent_dir = /var/www/writefreely-live
static_parent_dir    = /var/www/writefreely-live
pages_parent_dir     = /var/www/writefreely-live
keys_parent_dir      = /mnt/config/data
hash_seed            =
gopher_port          = 70

[database]
type     = sqlite3
filename = /mnt/config/data/writefreely.db
username =
password =
database =
host     = localhost
port     = 3306
tls      = false

[app]
site_name             = Blog Site  # <-- Change this to your desired site title and remove
this comment
site_description      = A Blog Site # <-- Change this to your desired site subtitle and remove
this comment
host                  = http://localhost:8080/  # <-- Change localhost to your domain name and
```

```
remove this comment. e.g. http://myblog.example.com
theme                = write
editor               =
disable_js           = false
webfonts             = true
landing              =
simple_nav           = false
wf_modesty           = true
chorus               = false
forest               = false
disable_drafts       = false
single_user          = false
open_registration    = false
open_deletion        = true
min_username_len     = 3
max_blogs            = 3
federation           = true
public_stats         = true
monetization         = false
notes_only           = false
private              = false
local_timeline       = true
user_invites         = user
default_visibility   = public
update_checks        = false
disable_password_auth = false

[oauth.slack]
client_id          =
client_secret      =
team_id            =
callback_proxy     =
callback_proxy_api =

[oauth.writeas]
client_id          =
client_secret      =
auth_location      =
token_location     =
inspect_location   =
```

```
callback_proxy     =
callback_proxy_api =

[oauth.gitlab]
client_id          =
client_secret      =
host               =
display_name       =
callback_proxy     =
callback_proxy_api =

[oauth.gitea]
client_id          =
client_secret      =
host               =
display_name       =
callback_proxy     =
callback_proxy_api =

[oauth.generic]
client_id          =
client_secret      =
host               =
display_name       =
callback_proxy     =
callback_proxy_api =
token_endpoint     =
inspect_endpoint   =
auth_endpoint      =
scope              =
allow_disconnect   = false
map_user_id        =
map_username       =
map_display_name   =
map_email          =
```

I have identified the bits you'll want to change with a comment int he code block above. Make sure to make the necessary changes, then remove my comment.

You can edit in the VI editor by pressing `I` one time to go into insert mode. When done editing pres `Esc` to get back out of editing mode.

Save the changes and exit the VI editor by pressing `:wq`.

Oddly, this container wants the config.ini and the writefreely.ini files to be the same, as it seems to use both. So, let's fix that real quick as well.

`cp ./writefreely/config.ini ./writefreely.in`i

Next, we'll run the command to create our new administrative user:

```
writefreely -c /mnt/config/etc/writefreely/config.ini --create-admin [username]:[password]
```

Replace the [username] with the username you want, and the [password] with a long, strong password. My command looked like this:

`writefreely -c /mnt/config/etc/writefreely/config.ini --create-admin brian:aReallyLongStr0n6Pa55w0rd`

Once run, you should see a success message. If so, we are done in the docker container

First, exit the container with `exit`

Now, let's restart our docker compose file.

`docker compose restart`

And allow the container to restart.

Awesome! Now, refresh your browser, and click the Log In button at the top of the screen. Enter your new administrative credentials to log in.

You can now start writing in your personal blog location, or you can click around a bit. Note, you do have a draft location that allows you to write and save, but won't publish your work publicly until you move it to your personal blog location.

Setup A Reverse Proxy

If you want to access your new site from outside of your home, you'll want to setup a reverse proxy to make sure requests for the site are routed properly to this docker container. I use NGinX Proxy Manager, but you are welcome to use any reverse proxy you are more comfortable with.

In NPM (NGinX Proxy Manager) click Add Proxy Host and in the new pop up winodw enter the domain name of your writefreely install.

You need to own the main domain, and have an A Record pointing to your public IP address. This is important if you want this to function properly.

I entered ytdiy.routemehome.org. I own the domain routemehome.org, and have an A-record already setup to point to the public IP address of my server. My firewall is configured to forward all incoming requests on port 80 and / or 443 to the server running NGinX Proxy Manager. NPM then handles the routing of the requests for my different services to the proper docker container / server.

Next, enter the local IP address of your server running WriteFreely into the IP field, and the port you mapped on the left side of the 8080 port mapping into the port field. I entered 192.168.10.60 and 8250 respectively.

Next, tick the options for Block Common Exploits, and Websocket Support.

Now select the SSL tab, and choose Request a New Certificate from the drop down, then tick the options for Force SSL, HTTP/2 Support, and optionally HSTS options as well. Make sure your email is filled in, and tick the option to accept the LetsEnctrypt terms of service.

Click Save. The pop-up should just disappear if everything is setup correctly, and you should have a new entry in your proxy list for your new WriteFreely install.

You can click on the domain name in the list, and make sure it opens properly to your WriteFreely page. Login, and you're ready to start writing.

Support My Channel and Content

Support my Channel and ongoing efforts through Patreon:
https://www.patreon.com/awesomeopensource