FreeScout - Support starting with Email

https://www.youtube.com/embed/vw9boxStxyc

FreeScout is an open source help desk suite that starts with email communication. It provides a way to connect your team to email. You create emails that your clients and users can utilize to get in touch with your staff, and provide your staff with a centrally managed web interface to these messages.

Imagine a company where you have a group email for:

- support@yourcompany.com
- careers@yourcompany.com
- premium-support@yourcompany.com
- sales@yourcompany.com

Any email you can create, you can tie to FreeScout. Next, you assign your team members to any email in FreeScout they may need access to. Finally, when email comes in, you can assign it to specific team member(s), and keep track of progress, ticket status and so on, all through email.

It's really a very novel idea with a simple objective to improve communication. You can add modules onto the system in order to expand your capabilities as your team and needs grow. The modules to have a small cost depending on which modules you want, but this helps support the open source aspects of FreeScout.

What You'll Need

- Docker-CE and Docker-Compose installed on a host
- NGinX Proxy Manager (or another web proxy of your choosing)
- An Email Server to which you have IMAP and SMTP credentials, and preferably the ability to create any email addresses you want / need for your business. (WARNING: do not try to use an existing email address where you already receive email. All Emails will be imported to FreeScout from that address)
- 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. In this case, if you don't have a different / preferred proxy, you should answer 'y' to NginX Proxy Manager.

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

Allow the script to complete installation.

Do the following commands:

`newgrp`

`newgrp docker`

When the install completes, do the following:

`cd docker/nginx-proxy-manager`

`docker-compose down`

`nano docker-compose.yml`

In the text editor, change the database user and password from the defaults.

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

`docker-compose up -d`

Installing FreeScout

Let's create a new folder inside of our docker folder.

`mkdir -p docker/freescout`

Move into that folder with

`cd docker/freescout`

And open a new text editor file called "docker-compose.yml"

`nano docker-compose.yml`

Now paste in the code block below, and we'll make a few changes to the file.

```
version: '2'

services:
  freescout-app:
    image: tiredofit/freescout
    container_name: freescout-app
    ports:
    - 80:80
    links:
```

```
    - freescout-db
    volumes:
    ### If you want to perform customizations to the source and have access to it, then uncomment this line - This includes modules
    #- ./data:/www/html
    ### Or, if you just want to use Stock Freescout and hold onto persistent files like cache and session use this, one or the other.
    - ./data:/data
    ### If you want to just keep the original source and add additional modules uncomment this line
    - ./modules:/www/html/Modules
    - ./logs/:/www/logs
    environment:
    - CONTAINER_NAME=freescout-app
    - DB_HOST=freescout-db
    - DB_NAME=freescout
    - DB_USER=freescout # feel free to change this to a different user.
    - DB_PASS=< some long strong password >
    - SITE_URL=http://< local ip of yoru machine >:< port on left above >
    - ADMIN_EMAIL=< your preferred email address >
    - ADMIN_PASS=< a long strong password again >
    - ENABLE_SSL_PROXY=FALSE
    - DISPLAY_ERRORS=FALSE
    - TIMEZONE=America/Chicago # change to your timezone
    restart: always

  freescout-db:
    image: tiredofit/mariadb
    container_name: freescout-db
    volumes:
      - ./db:/var/lib/mysql
    environment:
      - ROOT_PASS=AReallyLongStrongPasswordWithNum3r5andComplex1ty
      - DB_NAME=freescout
      - DB_USER=freescout  # must match the DB_USER in section above
      - DB_PASS=< same as in section above >
      - CONTAINER_NAME=freescout-db
    restart: always

  freescout-db-backup:
```

```
    container_name: freescout-db-backup
    image: tiredofit/db-backup
    links:
     - freescout-db
    volumes:
      - ./dbbackup:/backup
    environment:
      - CONTAINER_NAME=freescout-db-backup
      - DB_HOST=freescout-db
      - DB_TYPE=mariadb
      - DB_NAME=freescout
      - DB_USER=freescout # must match the two sections above
      - DB_PASS=< same as in sections above >
      - DB_DUMP_FREQ=1440
      - DB_DUMP_BEGIN=0000
      - DB_CLEANUP_TIME=8640
      - COMPRESSION=BZ
      - MD5=TRUE
    restart: always
```

I have marked the lines where you'll likely want to make a change with the less than andn greater than signs `<` and `>`, or with a comment `#`, telling you what should be put there.

Please go through and change the following as needed:

- port mapping - change the left side of the colon `:` only. I recommend a non 80 port. In my video I used 8252, so my port mapping looked like `- 8252:80`
- DB_USER - ideally can be changed in all three sections, just make sure the value you enter is identical in all three sections.
- DB_PASS - needs to be made a long, strong password, and must match in all the sections.
- ADMIN_EMAIL - should be changed to your preferred email address.
- ADMIN_PASS - needs to be made a different long strong password from the db password. This is the password you'll use to log into the web user interface as an administrative user.
- SITE_URL - We'll be changing this in steps as we go. First we want to get the application running on a local network, so enter the host machine's local IP Address with `http://` at the beginning. NOTE: If you did indeed change the port mapping from 80, then include the port after the IP address (e.g. `http://192.168.1.45:8252`.

Once you've made all of these changes, save the file with CTRL + O, press Enter to confirm, and then exit the nano text editor with CTRL + X.

Run Our Application

We'll now run our application with the command:

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

The `docker-compose up -d` pulls down the images, and starts our containers to run in the background (as a daemon `-d`). The `docker-compose logs -f` portion tells the system to show us the logs after the containers have started.

Watch the logs until you see a message that says you can access the application at the ip you specified.

Open your browser, and go to that address and port. You should see the initial login screen for FreeScout.

Feel free to login using the ADMIN_EMAIL and ADMIN_PASS informaiton you provided in your docker-compose file.

Setting Up a Domain / Sub-domain

If you want FreeScout to be really useful, you'll likely want your users to be able to reach it across the internet. To do this, we want to setup a somain (or URL) that the users can go to. You'll need to own the domain you want to use, and you'll need to be able to set an A-Record in the DNS settings of your domain to point to your **public** ip address. If you have a dynamic public IP, you can use a service like DuckDNS to give you a domain that will follow your changing IP.

In my case, I own the domain "routemehome.org". I setup my subdomain to be freescout.routemehome.org, and I used NGinX Proxy Manager to handle where to send a request for that subdomain.

You need to

1. Setup an A-record to point to your public IP, or a DuckDNS domain that will follow a dynamic public IP.
2. Setup NGinX Proxy Manager in docker (you can do this with the script from the beginning of the article).
3. Port forward ports 80 and 443 through your firewall to the host machine where NGinX Proxy Manager is running.
4. Make a proxy host entry in NGinX Proxy Manager that will direct requests for your freescout install to the freescout app.
5. Make a couple of iterative changes to our SITE_URL in our docker-compose.yml file.

I'll assume you've taken care of 1, 2, and 3 on your own, and continue from there.

Log into NGinX Proxy Manager, and go into the proxy hosts setup. Click to add a new proxy host, and in the pop-up window, enter your preferred domain / subdomain name that you've pointed to your Pulbic IP.

I entered "freescout.routemehome.org". Presss TAB or ENTER to make this entry a 'chip'.

Next, enter the IP of your freescout host machine. Mine is 192.168.10.112, so that's what I entered for the IP. Now, enter the Port number from the left side of the port mapping in your docker-compose. I used 8252, so that's what I entered.

Tick the options for Block Common Exploits, and Websocket Support. then click 'Save'.

This should show you a new row in your proxy host setup. Click the domain name in the left column to open a new tab and make sure your FreeScout site opens. If you entered everything correctly, and have your A-record and port-forwarding done properly, you should see your FreeScout site open, but you'll notice the domain is replaced by the IP in the browser URL bar.

To fix this, we need to go back into our docker-compose.yml file and edit our SITE_URL.

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

Move down, and remove the IP and port from the SITE_URL and replace it with the domain name you setup.

In my case my SITE_URL now looks like: `http://freescout.routemehome.org`

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

Now do the command:

```
docker-compose up -d
```

Wait a couple of minutes for the update to take effect. Then go back into NGinX Proxy Manager, and click your domain name again. Now it should open and the domain name should remain.

In NGinX Proxy Manager, on your freescout row, click the 3-dot icon on the right end of the row, then select 'Edit' from the pop-up menu.

Go to the "SSL" tab, and click the 'none' dropdown and select 'Request a New Certificate'.

Tick the option for "Force SSL" and "HTTP/2 Support", make sure your Email is entered, then tick the option to accept the LetsEncrypt terms of service, and click 'Save'.

If the Window goes away with no errors, you should see an item in the row that says LetsEncrypt for your freescout entry.

Now, we need to make one final change to our docker-compose.yml file.

`nano docker-compose.yml`

Change the SITE_URL to have `https` instead of just `http`.

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

Once again run

`docker-compose up -d`

and wait for the update to be done, then give it a minute or so to get the application updated.

Now navigate to your domain for freescout, and you should be hitting it with SSL encryption.

Congratulations, you are done, and up and running with FreeScout.

Make sure to check out the video to see how to setup Email with Freescout and start getting the best communication flows for your employees and clients.

Support my Channel and Content

Support my Channel and ongoing efforts through Patreon:

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

Revision #1
Created 2022-09-30 16:04:41 UTC by Brian McGonagill
Updated 2022-09-30 16:06:02 UTC by Brian McGonagill