Homepage

- Install Homepage

Install Homepage

https://www.youtube.com/embed/3Ux7zfCCM1A

Self hosted dashboards are incredibly useful for a myriad of reasons. They provide quick access to all of your self hosted services, quick links to bookmarks of your most used online sites and services, and can offer some very useful monitoring information about your services at a glance.

Homepage is no different in this regard, and is a fast loading piece of software to boot. One of the nicest features is that as you save changes to the configuration file, the page reloads quickly to show you how your changes have affected the page layout. This is something I've struggled with in Dashy since I started using it, having to work around caching in the browser by opening the dev tools and refreshing the page multiple times.

What you'll need

- Docker and Docker Compose
- The most general familiarity with how to open a terminal emulator (command prompt)
- About 10 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.

Installing Homepage

Now that we have docker and docker-compose installed, let's get ready to install Homepage. First we'll setup our desired folder structure. I put all of my docker applications inside of a top level (parent) folder called "docker". This makes it very easy to run a script that will backup all of my docker applications and compress them into a zipped format.

`mkdir -p docker/homepage`

This command will create the docker folder if it does not already exist, and will use the existing one if it does. Then it will create the homepage folder inside of that parent docker folder.

Now we'll move into the homepage folder and create a new file called docker-compose.yml.

`cd docker/homepage`

`nano docker-compose.yml`

In this case, I'm using the nano text editor directly in the terminal, but you can use any text editor you are more comfortable with. Make sure the editor is not a rich-text editor by default, as the yaml we will be adding is space dependent, and can be messed up with hidden rtf characters.

> Adding a new environment variable based on latest updates to Homepage. In this newer version you need to tell Homepage where it should expect to be accessed at. So if you use IP and port, you'll want o put that, but if you use a sub-domain like 'homepage.mydomain.net', then you'll want to enter that subdomain.

Copy the yaml code from the block below, and paste it into your new docker-compose.yml document.

```
---
services:
  homepage:
    image: ghcr.io/benphelps/homepage:latest
```

```
    container_name: homepage
    envrionment:
       - PUID: 1000
       - PGID: 1000
       - HOMEPAGE_ALLOWED_HOSTNAMES: "192.168.1.21:8921" # can be comma separated for multiple
entries if needed.
    ports:
      - 8921:3000
    volumes:
      - ./config:/app/config # Make sure your local config directory exists
      - /var/run/docker.sock:/var/run/docker.sock:ro # (optional) For docker integrations
    # user: 1000:1000 optional, not compatibile with direct socket see
https://gethomepage.dev/en/configs/docker/#using-socket-directly
    restart: unless-stopped
```

The changes you may want / need to make is to the port mapping (8921:3000), and to the environment variable HOMEPAGE_ALLOWED_HOSTNAMES. You can always change the left side port in a docker port mapping, just not the right side. The left side of the mapping is how you'll access the application once it's up and running. You simply want to make sure that it's not trying to use a port already in use on the host system. In this case I changed 3000 (the original left side port number) to 8921 in order to avoid a very common port used in nodejs based applications. As long as port 8921 is free on your host system, you won't need to change it.

Save the docker-compose.yml document with CTRL + O, then press Enter to confirm, and exit the nano editor with CTRL + X.

Run the Application

Now we are ready to run our application for the first time. We'll use the following command to run it:

`docker compose up -d`

If you have an older version of docker-compose, you may need to put a hyphen (-) between the words docker and compose, like this:

`docker-compose up -d`

NOTE: the un-hyphenated version is the newer version of docker-compose.

Allow docker to pull down the latest image of Homepage, and start up the container. When it's complete you'll be back at a normal terminal prompt.

Now go to your favorite modern browser and enter the IP address of your host machine (if you're running the application on the physical machine you're working on, you can use

http://localhost:8921) and the port 8921 (unless you changed the left side of the port mapping to another number, in which case you should use that port number. In my case, I went to:

http://192.168.10.154:8921

You should now be presented with the default Homepage layout screen. After loading the web page for the first time, you'll find that back in the terminal you now have multiple yaml files in the "config" folder. These are initially created after you load the page for the first time.

You can now start modifying the yaml files to customize your Homepage as you please. I highly recommend reading the documentation on the configuration options, as there are many, and you can get very detailed on what can be displayed.

https://gethomepage.dev/en/installation/

For a solid starting spot, I recommend opening the "services.yaml" and "bookmarks.yaml" files and playing with those a bit. You can add multiple service sections under each group, and you can add more groups as well in the services.yaml file. The same goes for the bookmarks.yaml file.

Make sure to check out the video for more detail on how to do that as well.

Support My Channel and Content

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