

Dashy - Powerful, Informative, Configurable Self-Hosting Dashboard

<https://www.youtube.com/embed/QsQUzutGarA>

Along the same lines as Homer, Heimdall, and Monitorr; I wanted to continue our look into personal web dashboards. I've posted about those few, and talked about others as well.

Sometimes the question comes up, and it's a valid one, "Why would I use this over the bookmarks toolbar in (Chrome, firefox, etc)?"

The thing these dashboards give you, is a much lovelier UI, but also many of them provide some extra information. Heimdall and Homer can be setup to provide details about Pi-hole, qBittorrent, Sonarr, etc. Dashy gives you up / down indications at a glance, and as it's an active project, will hopefully get some more status features moving forward.

The one thing I really like about Dashy is it's various methods for configuration. You can configure it through the terminal directly in the configuration file, or you can go through the User Interface with a nicely laid out configuration tool, an interactive (WYSIWYG) configuration editor, and / or just update the .yaml right there in the browser window.

For me, a person who updates their dashboard semi-regularly, it's a nice convenience to not have to leave the Web User Interface and SSH into another machine to make changes / updates.

What You'll Need

- Docker-CE
- Docker-Compose
- (Optional) NGinX Proxy Manager
- (Optional) Portainer-CE
- About 10 minutes of your time (not including adding items to the Dashboard)

Installation

[Check out this video on installing Docker, Docker-Compose, NGinX Proxy Manager, and Portainer-CE with a single script in under 5 minutes.](#)

For organizational purposes, you should run your docker containers (docker or docker-compose) from a folder called docker. Inside that folder you should create sub-folders for each application you run. So, inside the "docker" folder, we'll create a new folder called "dashy".

```
cd docker
```

```
mkdir dashy
```

```
cd dashy
```

Now, we want to make a couple of sub-folders inside of our new "dashy" folder. They'll be called "public" and "icons".

```
mkdir {public,icons}
```

NOTE: You can create your "dashy" sub-folder, and your "public" and "icons" sub-folders in a single command with:

```
mkdir -p dashy/{public,icons}
```

You can do a quick listing of your "dashy" directory with the command:

```
ls
```

Make sure both folders show up.

Now we'll create a simple text file called "docker-run.txt"

```
nano docker-run.txt
```

Inside of the file, you'll want to paste the code block below, and then we'll make any necessary modifications for your system.

```
docker run -d \  
  -p 8295:8080 \  
  --volume </path/to/your>/docker/dashy/public/conf.yml:/app/public/conf.yml \  
  --volume </path/to/your>/docker/dashy/icons:/app/public/item-icons/icons \  
  --name dashy \  
  --restart=unless-stopped \  
  lissy93/dashy:latest
```

or you can use this docker compose file.

```
name: dashy
services:
  dashy:
    ports:
      - 8295:8080
    volumes:
      - </path/to/your>/docker/dashy/public/conf.yml:/app/public/conf.yml
      - </path/to/your>/docker/dashy/icons:/app/public/item-icons/icons
    container_name: dashy
    restart: unless-stopped
    image: lissy93/dashy:latest
```

and save it as "docker-compose.yml".

Now, we'll make a couple of modifications as needed. First, if your host machine has port 8290 in use, change the port on the left side of the colon ':' to one that is not in use on your host.

Next, change the portion in the two volume mappings with '< >' around it to be the proper path of your docker folder.

In my case I have

```
/home/brian/docker/dashy/public/conf.yml
```

and

```
/home/brian/docker/dashy/public/icons
```

Again, only change the left side of the colon.

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

Our Initial Configuration File

For us to start with a fairly clean Dashy install, we'll want to use a fairly small conf.yml file.

So make a conf.yml text file in the public sub-folder with the command:

```
nano public/conf.yml
```

Inside this file, you'll paste the following code block:

```
appConfig:
  theme: colorful
  layout: auto
  iconSize: medium
  language: en
pageInfo:
  title: Home Lab
  description: Welcome to your Home Lab!
  navLinks:
    - title: GitHub
      path: https://github.com/Lissy93/dashy
    - title: Documentation
      path: https://dashy.to/docs
  footerText: "
sections:
  - name: Starter Only
    icon: fas fa-server
    items:
      - title: Google
        description: Search
        url: https://google.com
```

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

Authentication

If you want to add some Authentication on top of Dashy, I highly recommend Authelia, but there is a built in authentication piece as well.

You can modify the above block as follows:

```
appConfig:
  theme: colorful
  layout: auto
  iconSize: medium
  language: en
  auth:
    users:
      - user: your-preferred-username
```

```
hash: hash-of-a-password-you-choose-using-sha256-hashing
```

```
type: admin
```

```
pageInfo:
```

```
title: Home Lab
```

```
description: Welcome to your Home Lab!
```

```
navLinks:
```

```
- title: GitHub
```

```
path: https://github.com/Lissy93/dashy
```

```
- title: Documentation
```

```
path: https://dashy.to/docs
```

```
footerText: "
```

```
sections:
```

```
- name: Starter Only
```

```
icon: fas fa-server
```

```
items:
```

```
- title: Google
```

```
description: Search
```

```
url: https://google.com
```

Getting Icons for your Dashboard

We'll be pulling a really great github repo down that has tons of icons for self-hosted applications, and this just really pulls together any dashboard, regardless of which one you use.

So, move into the "icons" sub-folder we create earlier, and use the command:

```
git clone https://github.com/walkxcode/dashboard-icons.git
```

Clone the github repository.

Once pulled down, do an `ls` and you'll see a new folder called "dashboard-icons". Inside that you'll find several folders, but you'll likely be most interested in the "png" folder, which holds the png files for all of the icons in the repository.

Now, move back one step into your "dashy" folder with

```
cd ..
```

Pull and Run Dashy

Let's get our `docker run` command by running the command:

```
cat docker-run.txt
```

This will list out your docker-run command in the terminal. Highlight it, copy it, and paste it into the waiting terminal prompt. Press Enter, and docker will pull down the latest dashy image, and start it running using our starter conf.yml file.

Once it runs, give it about 1 minute, then go to your browser and enter the host machine's IP address, and enter the port you set on the left side of the port mapping in the docker run command.

For instance, my host is at 192.168.10.26, and I used port 8295, so I went to

```
http://192.168.10.26:8295
```

If all went according to plan, you should see your Dashy starter page load up.

Now you can start configuring, theming, and making your own special dashy dashboard page. Check out the video for more on how to navigate the Dashy User Interface, and how to configure your ultimate dashboard!

Revision #2

Created 6 April 2023 14:52:45 by Brian McGonagill

Updated 3 June 2024 19:48:21 by Brian McGonagill