Adding Widgets to the Dashy Dashboard

https://www.youtube.com/embed/dyur-NDngBc

I showed you how to setup and use a really nice homepage / dashboard for all of your self hosted services a while back called Dashy. I've stuck with it since then, and have just been super happy with it. It has active development, and tons of new features since then. One of those features is the ability to add all kinds of other widgets to the dashy interface.

Many of you have asked me how to add those widgets, so here we go.

Depending on what you're trying to view / see in your dashboard, you may need to install another bit of software on your server to provide the information for the dashboard to show. In my case, I'm going to walk you quickly through installing Glances, a really cool application that can provide you all kinds of data about your system. You can view it directly in the terminal / cli, or it can be presented as a web page (which is what we'll be using today).

I have a video on Glances and NetData that I'll link here as well for you to check out.

What You'll Need

- Dashy Installed and Ready to go - I have a video on it here, if you want to go through that first and get it installed.
- Glances running as a web server and background service (we'll go through that next)
- About 15 minutes of your time.

Installing Glances

You need to have python3 and pip3 installed. So,d epending on your distro, you may need to use a different package manager for this, than I do.

I'm using Ubuntu 20.04, so I'll be using the apt package manger. If you're running Debian / Ubuntu based distros, the commands should work for you as is.

If you're using Fedora, RedHat, or Centos (Alma, Rocky), you'll probably want to use Yum, RPM, or DNF. For Open Suse, you'll want to use Zypper, and for Arch, Pacman or PacAUR I'm guessing.

Install Python3

Open a terminal (CLI) window, and do the following: First make sure you have the lates package updates.

```
sudo apt update
```

Enter your super user password if prompted. If you're running Debian as root, you won't need the `sudo` part, just leave it off.

Next, we'll install python3 with:

```
sudo apt install python3 -y
```

After that completes, we'll install pip3 with:

```
sudo apt install python3-pip
```

Now that those are installed, we need to install Glances and Bottle (which will allow Glances to run as a web server).

Install Glances and Bottle

Use the following pip3 commands to install Glances and Bottle:

```
pip3 install bottle
```

If you get an error, try it with `sudo` like:

```
sudo pip3 install bottle
```

Now, do the same, but for Glances:

```
pip3 install glances
```

and again, if you needed sudo for Bottle, you'll need it for Glances, so do:

```
sudo pip3 install glances
```

Once those are finished installing, you can test that glances works by running the command:

```
glances
```

in your terminal. You should see a page full of information about your system show up.

You can stop glacnes with the CTRL + C key combination.

Next, you can make sure Bottle is working and run glances in web-server mode with the command:

`glances -w`

You'll see some output on the terminal, and should have something like `Glances Web User Interface started on http://0.0.0.0:61208/` on the screen.

Now, open a browser and go to

`http://localhost:61208`

or use the ip address of the machine you are working on:

`http://192.168.1.x:61208`

of course, using the correct private IP of the machine.

You should see a nice view of Glances, almos exactly as it looked in the terminal.

Now you can stop that process in the terminal with CTRL + C, an dwe need to turn that into a service that will run automatically, even after we reboot the machine.

Create the Glances Service

We'll be adding a new file to `/etc/systemd/system/` called `glancesweb.service`

So in a terminal window do the following:

`sudo nano /etc/systemd/system/glancesweb.service`

This will open a text editor in your terminal, and it should be empty.

Use the following code to start off:

```
[Unit]
Description = Glances in Web Server Mode
After = network.target

[Service]
ExecStart = /usr/local/bin/glances  -w  -t  5

[Install]
WantedBy = multi-user.target
```

If you are running as root, this should work, but we need to make sure glances is running from the path we expect, which is currently `/usr/local/bin/`. To find this out, save the file with CTRL + O, then press Enter to confirm, then use CTRL + X to exit the nano editor.

In the terminal, do the command:

```
which glances
```

You should get output like:

```
/usr/local/bin/glances
```

but, if you aren't running as root during the install you may get something like:

```
/home/<your usre>/.local/bin/glances
```

Whatever you get, highlight it, right click, select copy, and then we'll open the nano editor back up with:

```
sudo nano /etc/systemd/sysetm/glancesweb.service
```

On that line starting with `ExecStart`, make sure to remove the path (if it's different from what you got with the `which glances` command, and replace it with what you copied.

In my case, I would make it look like

```
[Service]
ExecStart = /home/<your usre>/.local/bin/glances -w -t 5
```

Now, because it's running from my home directory, I need to add one more line just below this one. If you are running from the `/usr/local/bin` directory, you **do not** need this line.

```
User = <your user>
```

so for me, the file looks like:

```
[Service]
ExecStart = /home/brian/.local/bin/glances -w -t 5
User = brian
```

Yours should have your username of course.

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

Next, we need to start and enable our service.

We do the following commands to do this:

`sudo systemctl start glances.service`

`sudo systemctl enable glances.service`

As long as you don't get any errors after each of those, you can check the status with

`sudo systemctl status glances`

You should see a row in the output near the top that shows `active`. If you see `failed`, you need to recheck the `glancesweb.service` file, and make sure you have everything correct.

Now that it's running, you can again go to the ip address of:

`http://<local ip>:61208`

and view your glances in the web browser. As long as it shows up, we are ready to move forward with getting some widgets in Dashy.

Adding Widgets to Dashy

At the time of writing, Dashy widgets can only be added from the configuration file, and not through the UI / GUI editor. I believe it's being worked on, but the config file isn't hard to modify, so let's jump into it.

If you're running Dashy in the way I showed in my video previously, you'll want to get on the server it runs on, and navigate to the foldeer where the configuration file is located. For me it's in a folder in my home directory called `docker/dashy/public`.

So I do

`cd ~/docker/dashy/public`

Now if you do

`ls`

you should see a file called `conf.yml`. This is the configuration file you want.

First, let's copy conf.yml to a new backup version, just in case we mess something up, it's easy to bring it back to the way it is right now.

`cp conf.yml conf.bak.yml`

Now, let's modify our `conf.yml` file to add a widget.

`nano conf.yml`

You may need to use `sudo`. If you see a red bar at the bottom of your nano editor, you need to exit with CTRL + X, and re-open it with `sudo`.

`sudo nano conf.yml`

Now, move down through the file until you see the first section called `sections`.

Just below that line, create a new line. Keep in mind that yaml or `.yml` is very space specific. So mind your spacing.

Let's add a new Widget section for our server. My server's name is "Aria".

```
sections:
  - name: Aria Info
```

Next, we'll add a `widgets` indicator, and our first widget. We'll add the glances cpu usage widget.

```
sections:
  - name: Aria Info
    widgets:
      - type: gl-current-cpu
        options:
          hostname: http:192.168.10.209:61208
```

At this point, you can save with CTRL + O, and then go to your browser and open your Dashy dashboard, to see your new widget and ensure it works. You need to, of course, replace the IP in the example above with the IP address of your server that you installed glances on. You may have to refresh Dashy if you're already running it, and you may have to tell firefox to release the cache then refresh (firefox is great, but it really hates to refresh and show new sutff).

Let's add another widget. Continuing in the nano editor from where we are. we'll add:

```
sections:
  - name: Aria Info
    widgets:
      - type: gl-current-cpu
        options:
          hostname: http:192.168.10.209:61208
      - type: gl-current-mem
        options:
          hostname: http:192.168.10.209:61208
```

You can again save, and take a look at your Dashy dashboard to see the new widget. You can now just go crazy adding widgets to Dashy using this same method.

Let's say you want to add two servers data. It's now hard. We just create another section for our next server like so:

```
sections:
  - name: Aria Info
    widgets:
      - type: gl-current-cpu
        options:
          hostname: http:192.168.10.209:61208
      - type: gl-current-mem
        options:
          hostname: http:192.168.10.209:61208
  - name: Liratta Info
    widgets:
      - type: gl-current-cpu
        options:
          hostname: http:192.168.10.152:61208
      - type: gl-current-mem
        options:
          hostname: http:192.168.10.152:61208
```

Notice the different name and ip address in our second section. Also, understand that the glances / bottle install and service setup, needs to be done on each server / machine you want this information from.

Now you can turn Dashy into an incrdible tool for all kinds of great information.

Support me on Patreon

Support my Channel and ongoing efforts through Patreon:

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

Revision #1
Created 2022-09-30 16:59:18 UTC by Brian McGonagill
Updated 2022-09-30 17:01:44 UTC by Brian McGonagill