LibreSpeed and OpenSpeedTest

https://www.youtube.com/embed/7S8f_xEpCaQ

In the past we've gone over some great tools for checking your internet and network speeds from the command line, but there is really nothing quite like a really nice GUI (Graphical User Interface) for doing something like a speed test.

Today, I wantt o introduce you to not one, but two really great, really stratight-forward GUI's for running network and internet speedtests on your own hardware or VPS.

No more running a speedtest provided by your ISP when you are experiencing speed issues, and letting them hang up with the presumption that the problem is all on you.

What You'll Need

- Docker-CE
- Docker-Compose
- (Optional) NGinX Proxy Manager
- (Optional) a domain or subdomain for your speed test site(s).
- (Optional) Portainer-CE

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

Install and Setup

OpenSpeedTest

First, we'll setup OpenSpeedTest. It's a single `docker run` command, and really fast to get going. As always, I like to begin with a bit of organization. If you're just starting out, or a docker pro, it never hurts to keep your docker installs organized. I like to create a "docker" directory, then place folders inside that directory for each docker application / stack I run.

If you're new to the command line, then here are some simple commands to get you setup, if you're a pro, and already have your docker setup organized, feel free to skip to the "docker run" section below.

First let's create our main docker folder:

`mkdir docker`

Now let's move into that folder, and create our folders for both of the speedtest apps we want to install today.

`cd docker`

`mkdir {openspeedtest, librespeed}`

That command will create two folders. You can verify they were created with command:

`ls`

Make sure you see each folder listed. If so, let's move on to setting up our docker run command for OpenSpeedTest.

Docker Run

We'll move into our openspeedtest directory, and create a text file for our docker run command. I like to do this so I can reference it later (in case I need to re-build or re-install for any reason). This also helps me keep track of changes I may make as I go.

`cd openspeedtest`

Now we'll create our text file with

`nano docker-run.txt`

Inside that file copy the text below and paste it in:

```
docker run --restart=unless-stopped \
--name openspeedtest -d \
-p 8191:3000 \
-p 0192:3001 \
openspeedtest/latest
```

Once pasted, you can save the file with CTRL + O, then Enter to confirm, and exit the nano text editor with CTRL + X.

Before exiting, however, make sure you change the ports listed after the `-p` arguments to ports that are open on your host machine. You can change the number on the left side of the colon ':', but don't change the right side.

Once you've made changes, saved, and exited the file, do the following command to list the file contents to your terminal:

`cat docker-run.txt`

Highlight the listed `docker run` command fully, all the way through the word "latest", then paste it at the terminal prompt.

Press enter, and the command should run. This will pull down the openspeedtest image, and start the container on the ports you specify.

As long as you dont' see any error messages, you can go to the IP and port of your host machine in the web browser, and you should see the Open Speed Test GUI displayed. Click the start button to run your test.

For instance, I installed on my host with an ip of 192.168.10.26, so I went to `http://192.168.10.26:8192`

LibreSpeed

For LibreSpeed, we'll be using docker-compose, an excellent tool for providing a little bit more order to the docker-run command, and it has some great tooling built around it for updates in place, restarts, and so on.

first, let's back out of the "openspeedtest" folder, and get into the librespeed folder we created above.

`cd ..` will take you back one level in the folders, then you can use

`cd librespeed` to get into the librespeed folder.

Now we want to create a new "docker-compose.yml" file and put some yaml code in it that will tell docker how to run the application.

`nano docker-compose.yml`

Once the file is open, paste the following code into it:

```
---
version: "2.1"
services:
  librespeed:
    image: lscr.io/linuxserver/librespeed
    container_name: librespeed
    environment:
      - PUID=1000  # change this to your user id
      - PGID=1000  # change this to your group id
      - TZ=America/Chicago # change this to your timezone
```

```
      - PASSWORD=PASSWORD
    volumes:
      - /home/<your user name>/docker/librespeed/config:/config
    ports:
      - 8190:80
    restart: unless-stopped
```

Once you've pasted the above text into the file, you'll want to change a few things potentially. First, let's just save what you have with CTRL + O, then enter to confirm. Now, let's briefly exit nano with CTRL + X.

Now, let's check your user's UID and GID in the command line. Type the "id" command:

`id`

From the output identify your UID and GID numbers, and make note of them.

Next enter the 'pwd' command

`pwd`

and highlight / copy your current directory. This is the path we'll want for our volume mapping.

Now go back into your docker-compose file

`nano docker-compose.yml`

Replace the PUID with your UID number, and the PGID with yoru GID numbeer if needed.

Next, make sure to change the timezone to your timezone.

Finally, in the volume mapping make sure to paste the path to your librespeed directory, and add /config to it.

My path from the `pwd` command was /home/brina/docker/librespeed, andn I added /config to it to make the full path:

`/home/brian/docker/librespeed/config`

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

Now we'll run our docker-compose file to pull down LibreSpeed and start it running.

`docker-compose up -d`

Give it time to pull down, and when you see the "done" message, you can navigate to your IP and the port specified in the compose file on your host machine. You should see the LibreSpeed GUI, and you can start a speed test.

NGinX Proxy Manager

If you want to run your speedtest(s) on a VPS, you can absolutely do so using the IP address alone, but if you'd prefer to set it up with a domian / subdomain name, you may want to use a tool like NGinX Proxy Manager. You can then proxy the traffic to your speedtest(s). In particular, a proxy manager is handy if you are wanting to run more than one speed test app on a single server.

I have several videos and how to's on using NGinX Proxy Manager, so definitely get out there and give it a try if you'd like to.

Support my Channel and Content

Support my Channel and ongoing efforts through Patreon:

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

Revision #1
Created 2023-01-23 18:56:15 UTC by Brian McGonagill
Updated 2023-02-13 17:50:06 UTC by Brian McGonagill