NetBox - Open Source Network Mapping

- NetBox - Open Source Network Inventory and Mapping

NetBox - Open Source Network Inventory and Mapping

https://www.youtube.com/embed/sHvMyRHA7sY

NetBox is an Apache 2.0 licensed open source software suite that allows you to design, map, and manage your network and infrastructure systems.

Let me be clear, and you should read the NetBox docs for more detail, but NetBox is not a network monitoring system. It's not a DNS system, and it's not a RADIUS system.

NetBox was created by the amazing folks over at Digital Ocean to allow them to design out, and manager the network infrastructure, IPs, Devices, Racks, Rooms Locations, and so much more.

While this may sound like a tool that is only useful to Network Engineers, don't be fooled. It's a tool that can help you now, and will definitely be useful to you in the future as we continue to see more and more devices on our homelab and small business networks. It's so important to structure your network well, and if you can do that from the beginning while your homelab or small business network is still relatively small, you'll really win big as your network grows over time with all of the devices, and IoT objects, and everything else that wants access to the internet.

I go through a very high level overview of the NetBox user interface in the video, but I highly recommend you check the NetBox demo site to see a very loaded system, and play with it yourself to get a real feel for what it can provide.

Once again:

The NetBox Demo Site

Installation

What You'll Need

- Docker-CE and Docker-Compose
- Git, Curl, and Wget installed
- About 10 minutes of your time

- (optional) NGinX Proxy Manager (or a proxy software of your choosing)
- (optional) SMTP (Mail Sending) Information

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.

At some point, you'll 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 NetBox

Now that you have Docker and Docker-Compose installed, you'll want to setup an organized way to backup any docker applications you install besides NetBox.

My method is to create a folder in my home directory (but you can actually put it anywhere that you have permissions to access) called "docker".

First, list the contents of your home directory and make sure my script didn't already create a docker folder for you. If you installed any of the extra software like Portainer, or NGinX Proxy Manager, then it likely did create that folder for you.

So, just do the command:

```
ls
```

to list the items in your current location.

If you see "docker" as one of the items, then no need to create it. If you don't see "docker", then you'll want to create a new folder called "docker" with the command:

```
mkdir docker
```

Now, move into that folder with the command:

```
cd docker
```

Once there, whether you created it, or my scrpit created it, you'll soon have a new folder called netbox-docker, but it's created when we run a command to pull down the netbox repository.

You'll use the `git` command to do this.

```
git clone -b release https://github.com/netbox-community/netbox-docker.git
```

This will pull down the NetBox docker reository to your machine, and create a folder called "netbox-docker".

ONce it completes, move into the netbox-docker folder with

```
cd netbox-docker
```

Next, you'll want to look at the files in this folder. Use teh command:

```
ls
```

to list them out.

Notice a file called "docker-compose.override.yml.example". We need this file to be name "docker-compose.override.yml". We can do this with a single command:

```
cp docker-compose.override.yml.example docker-compose.override.yml
```

Once done, you'll want to make sure that port 8000 is open and unused on your host machine. If it's already in use, then open the new docker-compose.override.yml file we just created, and change the left side port from 8000 (left side of the colon) to some port that is not in use.

To open the file for editing, use

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

Arrow down to the line with the port mapping defined, and change the left side port ONLY to something else. If 8000 is already open on your host, then it's no issue, and just leave the file as is.

Next, we need to make some adjustments in the "env" folder, more specifically in each of the files inside that folder.

Let's start with the "netbox.env" file first.

We can do the following to see all of the files in that folder:

```
ls ./env/
```

Once you have the files listed, we can edit them in nano with

```
nano ./env/netbox.env
```

You'll see a lot of environment variables here. These variables help the NetBox container know what you want / need for your installation of Netbox to be secure, and useful.

We'll talk about each set of environment variables by section:

```
DB_HOST=postgres
DB_NAME=netbox
DB_PASSWORD=J5brHrAXFLQSif0K
DB_USER=netbox
```

You definitely need to change the DB_PASSWORD. It's very important that you make this a long, strong password, and make it different from the default. You'll also need to change the password in the postgres.env vile to match what you change this one too.

```
EMAIL_FROM=youremail@example.com
EMAIL_PASSWORD=
EMAIL_PORT=587
EMAIL_SERVER=box.example.com
EMAIL_SSL_CERTFILE=
EMAIL_SSL_KEYFILE=
EMAIL_TIMEOUT=5
EMAIL_USERNAME=youremail@example.com
# EMAIL_USE_SSL and EMAIL_USE_TLS are mutually exclusive, i.e. they can't both be `true`!
EMAIL_USE_SSL=false
EMAIL_USE_TLS=true
```

You'll want to fill out the SMTP email details if you want the system to be able to send you any kind of emails / notifications. It's important that you understand your SMTP settings, as they can, and will, vary widely from person to person, and from service to service.

```
REDIS_CACHE_DATABASE=1
REDIS_CACHE_HOST=redis-cache
REDIS_CACHE_INSECURE_SKIP_TLS_VERIFY=false
REDIS_CACHE_PASSWORD=t4Ph722qJ5QHeQ1qfu36
REDIS_CACHE_SSL=false
```

Again, you need to chage the password for the Redis_Cache, the value for the REDIS_CACHE_PASSWORD needs to be a long, strong password that is different from this default. You'll also need to change the value int he redis-cache.env file to match this password value you set here.

```
REDIS_DATABASE=0
REDIS_HOST=redis
REDIS_INSECURE_SKIP_TLS_VERIFY=false
REDIS_PASSWORD=H733Kdjndks81
REDIS_SSL=false
```

Next, you'll want to also change the default redis database password, to a long, strong password. Again, make sure to update the value in the redis.env file to match with the password value you set here.

> Note: The following may no longer be valid. If you did not set this, or see this in your netbox.env file, it's okay. Start the services, and then navigate to the IP and port you set for netbox as follows in the final instructions. You'll create your super user with this command:

>

```
docker compose exec netbox /opt/netbox/netbox/manage.py createsuperuser
```

```
SUPERUSER_API_TOKEN=0123456789abcdef0123456789abcdef01234567
SUPERUSER_EMAIL=youruser@example.com
SUPERUSER_NAME=admin
SUPERUSER_PASSWORD=admin
```

You'll want to change the SUPERUSER_API_TOKEN value here, as this would potentially allow API access to anyone in the world if you left it as this generic default.

Additionally, you'll want to change the SUPERUSER_PASSWORD value to a long, strong password. It's very important that you update all of these values in order to make the system as scure as possible.

Once you've made the appriate changes in this file, use CTRL + O to save, then press Enter to confirm, and use CTRL + X to exit the nano editor.

Don't forget to update the values in the postgres.env, redis.env, and redis-cache.env files to match the values you set in each section respectively in the netbox.env file. AGain, use nano to make your edits, and when done, use CTRL + O to save, then press Enter to confirm, and use CTRL + X to exit the nano editor.

Now, you're ready to pull down the images. For this, the NetBox instructions say to use the command:

```
docker-compose pull
```

Once, that completes, use the command:

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

This will start the containers, and show you a trailing log of the what's going on. Be patient, as the first run will take several minutes to get started. I'd give it a solid five minutes before trying it out.

When you're ready, put your host IP and the port you setup in the override.yml file into your browser of choice. My host is at 192.168.10.112 and I left the port as 8000, so I entered:

```
http://192.168.10.112:8000
```

I was greated with the dashboard, but saw little lock icons for everything. This indicates you need to login. You can now click the Login icon in the upper right of the window. You'll be taken to the login view, and you'll enter the superuser username and password you set in the netbox.env file.

Once logged in, you are ready to start designing and mapping your network out.

On YouTube, there is a channel with a ton of NetBox content that are much more in depth, I'm linking you to it here.

I hope you enjoy NetBox, I believe it is an amazing tool that will benefit anyone who is getting into homelab, self hosting, all the way up to the corporate IT and Network Engineers.

Support Digital Ocean and this great project, by checking out Digital Ocean with a $50.00 credit - click this affiliate link. You get a credit, and if you stay with Digital Ocean, I get a credit, and that helps keep the Awesome Open Source channel going.

DigitalOcean – The developer cloud
Helping millions of developers easily build, test, manage, and scale applications of any size – faster than ever before.

Explore our products



Support my Channel and Content

Support my Channel and ongoing efforts through Patreon:

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