Run Mesh Central In Docker

- MeshCentral - an Open Source, Self Hosted Remote Machine Management and Access tool

MeshCentral - an Open Source, Self Hosted Remote Machine Management and Access tool

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

If you have been looking for a self-hosted, open source, free way to do remote support for yourself, your family, your friends, your business, or anything else, this is an excellent tool for doing exactly that.

The best part is we can now install MeshCentral with Docker, and this makes it so much easier to get going. It was already easy, but now it's even simpler. We are going to install it, and setup NGinX Proxy Manager to give our install a Fully Qualified Domain Name and LetsEncrypt for CA Certified SSL Encryption.

Install with Docker-CE, Docker-Compose, and NGinX Proxy Manager

We'll be using an image on DockerHub from typhonragewind, and he has this setup really well for getting everything up and running pretty easily.

Before we begin, you will want to have Docker-CE (not docker.io) and Docker-Compose installed on your system. I have a post on how to do this here. Definitely go get that step done first, then come back to continue with the MeshCentral install.

Install MeshCentral

Head over to https://github.com/Typhonragewind/meshcentral-docker and scroll down to see the docker-compose.yml text on the page.

> It's always best to check the original source pages for any updates. As these articles age, the projects will progress and change over time. Make sure you get the latest and greatest!

Copy the text.

For convenience, here's my file contents for the docker-compose.yml.

```
version: '3'
services:
    meshcentral:
        restart: always
        container_name: meshcentral
        image: typhonragewind/meshcentral:latest
        ports:
            - 8086:443  #MeshCentral will moan and try everything not to use port 80, but you can also use it if you so desire, just change the config.json according to your needs
        environment:
            - HOSTNAME=meshctrl.mydomain.com     #your hostname
            - REVERSE_PROXY=192.168.21.217     #set to your reverse proxy IP if you want to put meshcentral behind a reverse proxy
            - REVERSE_PROXY_TLS_PORT=443
            - IFRAME=false    #set to true if you wish to enable iframe support
            - ALLOW_NEW_ACCOUNTS=false    #set to false if you want disable self-service creation of new accounts besides the first (admin)
            - WEBRTC=true  #set to true to enable WebRTC - per documentation it is not officially released with meshcentral, but is solid enough to work with. Use with caution
            - TZ=America/Chicago
        volumes:
            - ./meshcentral/data:/opt/meshcentral/meshcentral-data    #config.json and other important files live here. A must for data persistence
            - ./meshcentral/user_files:/opt/meshcentral/meshcentral-files    #where file uploads for users live
```

Now open a terminal window, and create a directory called "meshcentral"

`mkdir meshcentral`

and move into that directory

`cd meshcentral`

Now make a "docker-compose.yml" file

`nano docker-compose.yml`

and Paste in the docker-compose text you just copied from the GitHub page above. Use CTRL + Shift + V to paste in the terminal window.

Next, we want to make a few changes to the docker-compose text before saving.

First up, if you want a port that is different than the default 8086 set in the yaml file, then change the left side of the port mapping only. DO NOT change the right side of the port mapping. Notice, the author warns you that if you try to make it port 80, it's going to cause issues, so I suggest not using port 80 for this. Just make sure to use a port that you are not currently using on your host machine.

Next, setup a hostname for your server. If you want to setup a hostname, this is where you specify it. If you want to access this from outside your private network, set a publicly routable hostname. Make sure you own the domain you use if you use a publicly routable domain.

If you don't want to use a publicly accessible domain, then setup a hostname for your private netwrok...somthing like "meshcentral.local".

If you'll be using a reverse proxy (which in this tutorial we will be using NGinX-Proxy-Manager), then you need to enter the IP address of the host machine for your proxy. In my case it's 192.168.7.125, but make sure to use the IP of your proxy host.

I left REVERSE_PROXY_TLS_PORT as 443, and sugget you do the smae. I changed IFRAME to "false".

I am also changing ALLOW_NEW_ACCOUNTS to false, because I don't want to allow anyone else to sign up, but me. If you have an IT team, or other users you want to allow to access the system, you can leave this enabled, but you can also create other users from inside the MeshCentral UI when it's up and running.

Finally, I set WEBRTC to be "true", as I want the fastest possible connection between my machines, and this will allow that when it's available.

Volumes

We need to setup a space for our volumes to be mapped as well. This allows us to update the containers without losing data.

First save your changes with CTRL + O, then Enter, and then use CTRL + X to exit.

Now make a folder called "meshcentral" again.

```
mkdir meshcentral
```

So you should have a folder called "meshcentral" inside the folder called "meshcentral".

Start it Up

Now use the command

```
docker-compose up -d
```

to pull down the images, and start the container(s).

When you see 'done' in the terminal, give it about 2 minutes, then navigate to your hose machine's IP address with the port 8086. Make sure to use https for the URL.

for me I went to https://192.168.7.51:8086. You'll get a self signed certificate warning, but feel free to accept the risk, and move forward.

Setup NGinX Proxy Manager for Access

If you are going to be accessing this system from outside your network, then you'll want to setup a routable domain name. You can use something like DuckDNS to get a dynamic DNS address that will update when your public IP updates, or you can purchase your own domain name, and create an entry that points to your Public IP address.

Either way, once you get the domain to your network, you need a way to route that traffic more safely around your network to your various machines and containers. This is where NGinX Proxy Manager comes in.

Open up NPM and click the "Proxy Hosts" option if you aren't already on that view. Now click to Add a New Host.

In the pop-up form, enter the domain name that will be used to access your meshcentral install. for example if you are going to use "meshcentral.remotemanagement.com" (presuming you won remotemanagement.com) then you would enter "meshcentral.remotemanagement.com" into the blank, and press tab to accept the entry.

Next, change the `http` to `https` in the field below the domain name, then tab over to the Forward Hostname / IP field, and enter the IP of the host machine you have MeshCentral running on (if it's a different machine than the one your NPM is running on). If you have both NPM and MeshCentral running on one machine, then enter the docker gateway IP in this field. You can get the docker gateway IP through the terminal.

```
docker network inspect bridge
```

Check the JSON for the key 'Gateway' and make note of the IP value. Usually this is 172.17.0.1, but not always, so it's good to double check.

Next, tab to the port field in NPM, and enter the port you set on the left side of the port mapping in the yaml docker-compose file. If you left it as the default it will be 8086.

Now, Save.

In the NPM interface, click the new entry, and make sure you are taken to the proper server. You should again see the Certifiacte warning.

Let's Get Our LetsEncrypt Certificates

In NPM, click the 3-dot icon on the right end of your new proxy host entry, and select 'Edit'.

Go to the "SSL" tab in the pop-up window, and select "Request a New Certificate". Move down, and enable "Force SSL". Next, enter your email address for LetsEncrypt, and enable the "I Agree" option. Now click Save.

It may take a few seconds, but if everything works, the pop-up will close without any warnings or errors.

You can again click on the URL you set for this host entry, and it will open in a new tab. You should now see the login screen of your MeshCentral install, and you can see a verified SSL lock icon in the URL bar.

Click the 'Crate an Account' option in the login window, and create your new account for Mesh Central.

Make sure to check out the video at the top, to see how to use MeshCentral, add machines, control them, and navigate the UI.

Support my Channel and Content

Support my Channel and ongoing efforts through Patreon:

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