

5.0 Setting Up NFS

Creating and using NFS with Docker

Any stack file that you deploy will have at least one service that will need to store static data. This will become a problem when stand-alone methods are used for volume creation, because the volume will not magically transport to whatever node the container gets deployed to. Enter NFS share volumes to the rescue!!

Step One - Install NFS

Make sure that you have installed NFS on the clients/nodes (including manager nodes)

```
sudo apt install nfs-common
```

On the Ubuntu NAS (gump.lan or 10.10.1.33 is our BMS file server, as of this writing), install the server components as well

```
sudo apt install nfs-kernel-server
```

Step Two - Using NFS in A Stack

Making an NFS Export

Then, as an example, we might set up a share like this:

```
sudo mkdir /pools/pool1/mysharename  
sudo chown nobody:nogroup /pools/pool1/mysharename  
sudo nano /etc/exports  
sudo systemctl restart nfs-kernel-server
```

The `/etc/exports` line to add our share and publish it would look something like this:

```
/pools/pool1/mysharename *(rw,sync,no_subtree_check)
```

Add `no_root_squash` after `no_subtree_check` if the container wants to change ownership on files during runtime.

Test The Export

To check if the export worked, use these commands. If no errors are reported - it worked!

```
sudo mount 10.10.1.33:/pools/pool1/mysharename /mnt
sudo touch /mnt/testfile
sudo rm /mnt/testfile
sudo umount /mnt
```

Step Three - Usage in a Stackfile

Add the following to the stack.yml file in the volumes definition area:

```
letsencrypt:
  driver_opts:
    type: "nfs"
    o: "addr=10.10.1.33,rw,noatime,rsize=8192,wsiz=8192,tcp,timeo=14,nfsvers=4"
    device: ":/pools/pool1/mysharename"
```

Deploy

Deploy the stack, this will create the volume and connect it as it deploys. Using Portainer, you can also browse the content.

Revision #4

Created 29 May 2025 22:01:55 by Vivian Miller

Updated 29 May 2025 22:16:00 by Vivian Miller