QNAP NAS

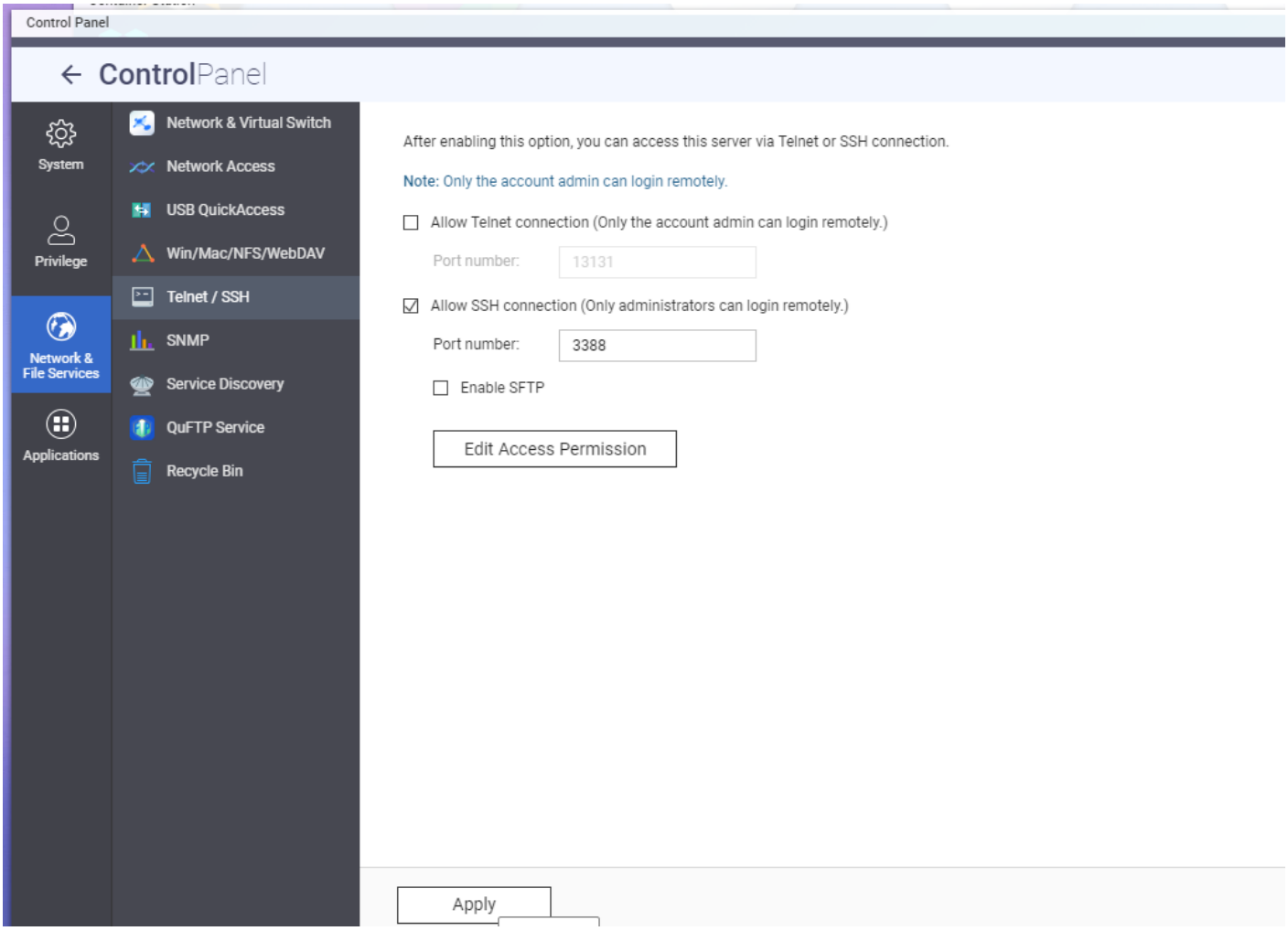
This book contains procedures on a QNAP NAS system, which might not always be so straightforward or can be forgotten as they're not often done.

- Link subdomain to docker (NGINX)
- Container station: Docker access Application
- Container station: sharing common files
- Improve SMB speeds

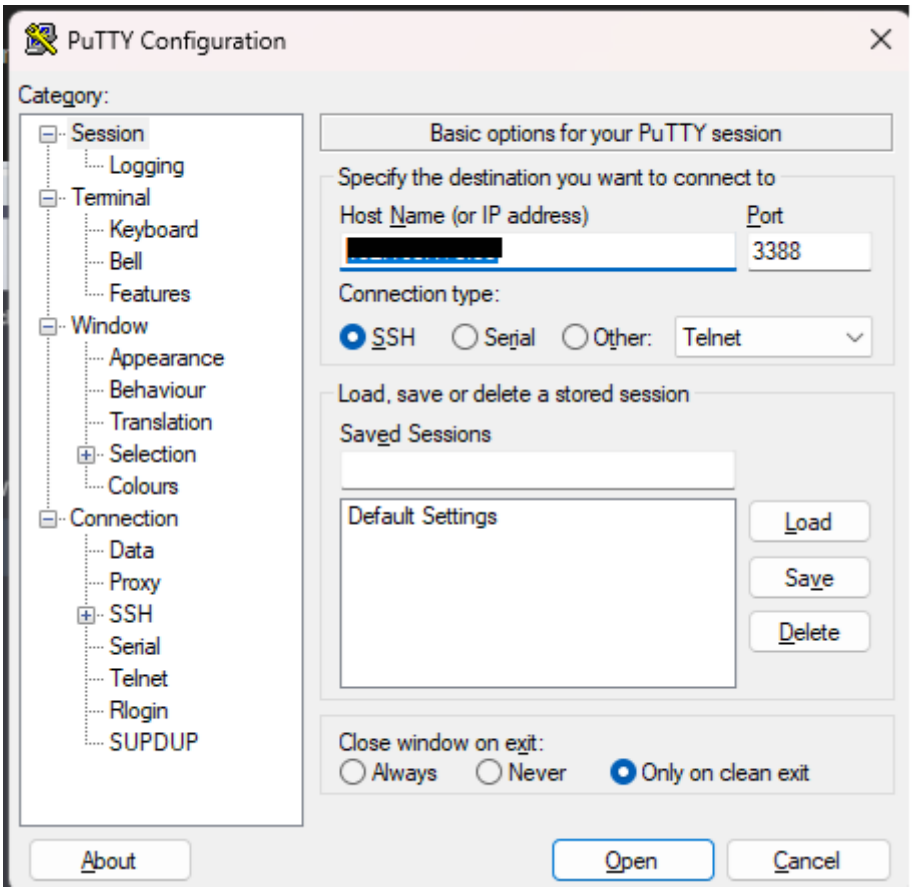
Link subdomain to docker (NGINX)

This section assumes an NGINX docker and you have setup your DNS to connect to you NAS via a wildcard.

1) Enable Telnet on the NAS in Control Panel



2) Telnet into the NAS, I prefer putty



3) cd into the folder you can easily access or edit (in my case, a network shared folder, execute, in the terminal screen "docker ps" to find the id of your container

```
docker ps
```

```
docker  cp c0568abcdb83916:/etc/nginx/nginx.conf tmp_tim.conf
```

then edit, the config file for nginx, and do the reverse:

```
docker  cp tmp_tim.conf  c0568abcdb83916:/etc/nginx/nginx.conf
```

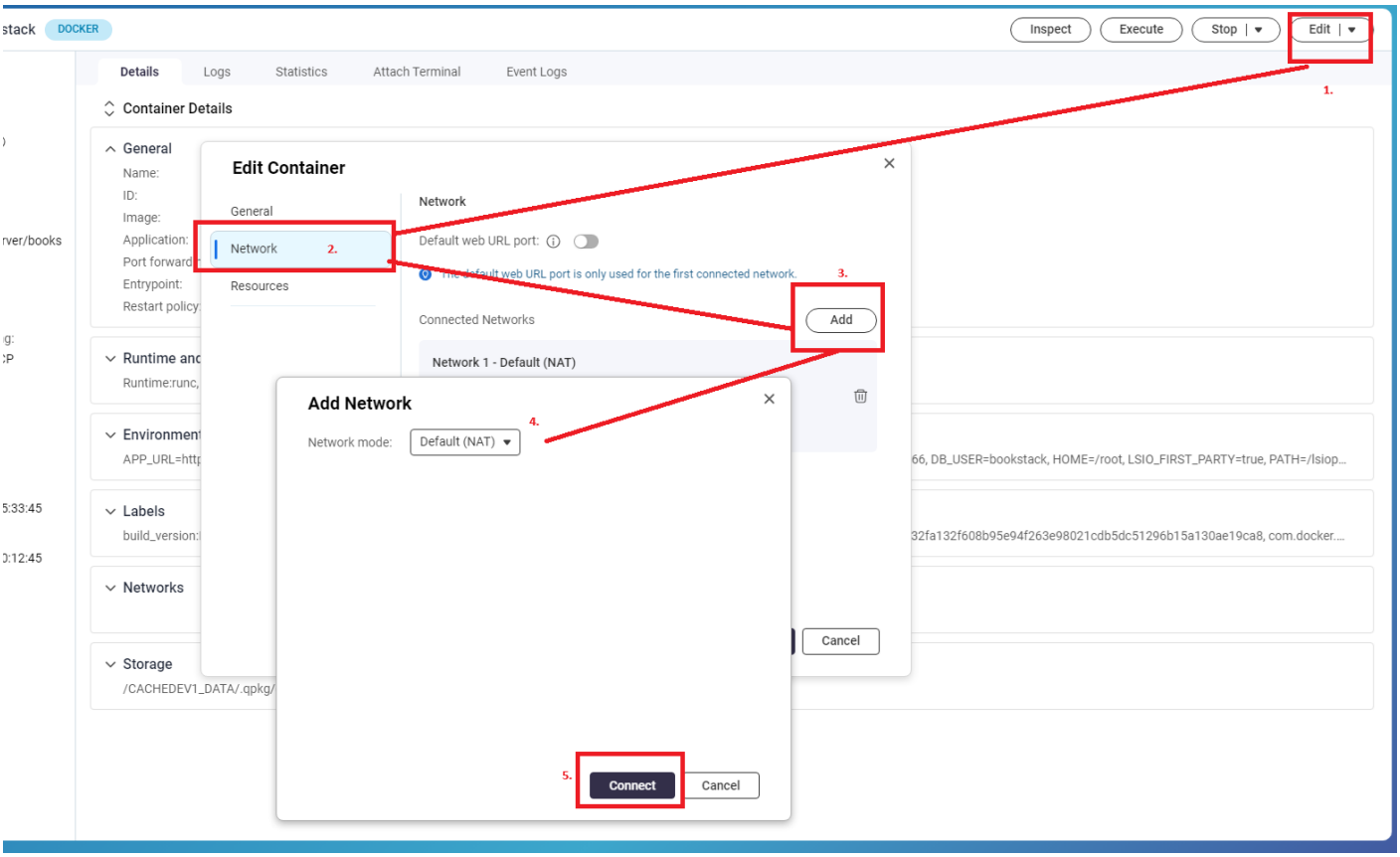
Restart the container.

Container station: Docker access Application

When you use a YAML file to create an Application in Container station, QNAP will create a private network.

Hence, you will not be able to connect to it, for example with your nginx proxy forwarder.

1. Go to your Container station
2. Select the application and the containers details, one by one
3. Click "Edit" and open network section
4. Remove the "connected networks" and add the network your other dockers are connected on fe. ***Default(NAT)***
5. Connect, and repeat this for all the containers in the application



Optional steps:

1. Get in a shell of your nginx docker and curl to verify it can communicate with the container

Container station: sharing common files

In this page we're going to create a link from in a container/image to the Host.
Inside the container is "/app/media" and bind it to a share on the QNAP host system.

Context

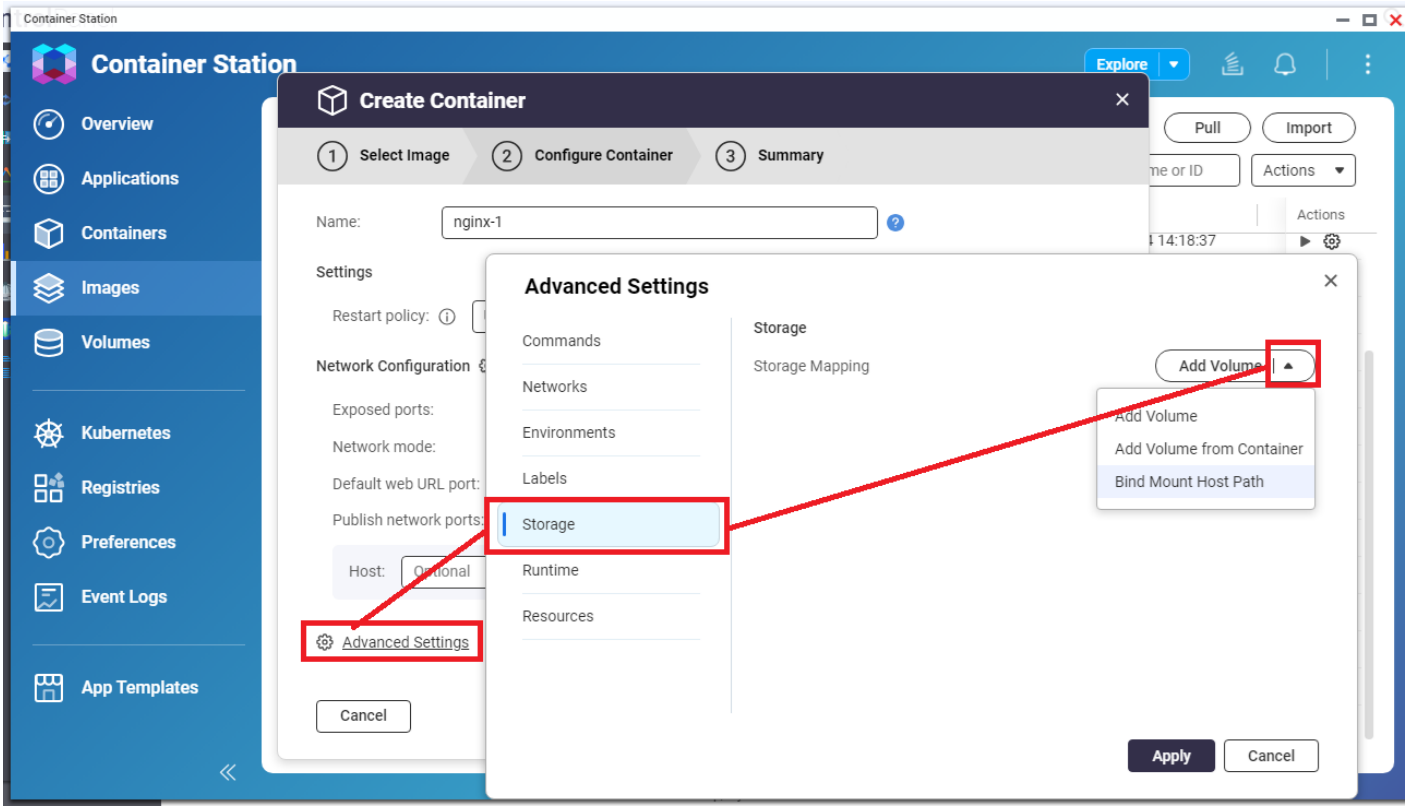
While it's convenient to spin up a new Docker image, or republish one. You'll soon come into a situation where you want to share resources over different containers. Or you realize there is data in the container, that you want to retrieve or need to copy over when you spin up a new container... Which soon becomes a daunting task and a chore, which is prone to making mistakes while doing this heavy manual work.

The most convenient way, I have found, is to bind a folder in the container with the host system. And share this drive in the *private* network - so it becomes possible to edit the files easily. And share, for example a folder with images and videos, over different containers.

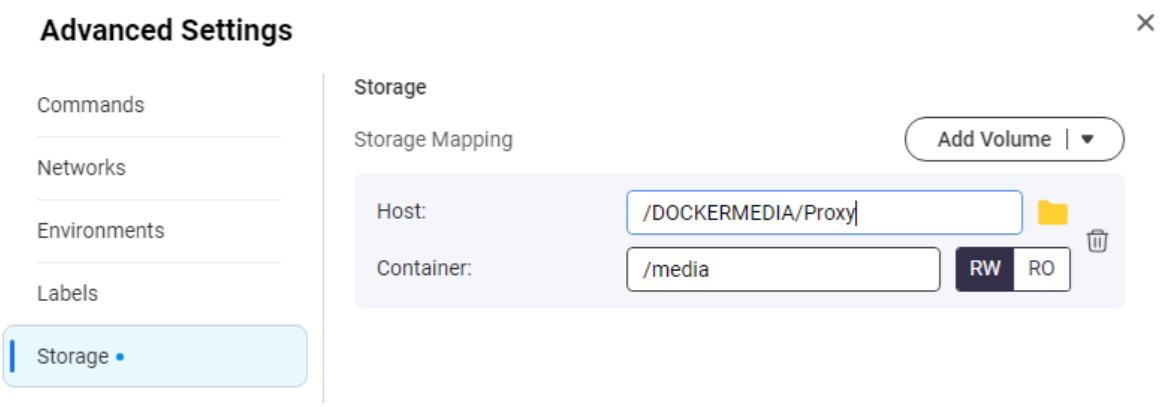
Configuration

1) Create your folders and optionally share them. (I would suggest to create a folder per project and also split up between your **Test** and **Production**.

2) Create a new container
3) While configuring, go to Advanced Settings -> Storage -> Bind Mount Host Path



4) Write your mapping. In my case I have a share at **/DOCKERMEDIA**



How to access the files

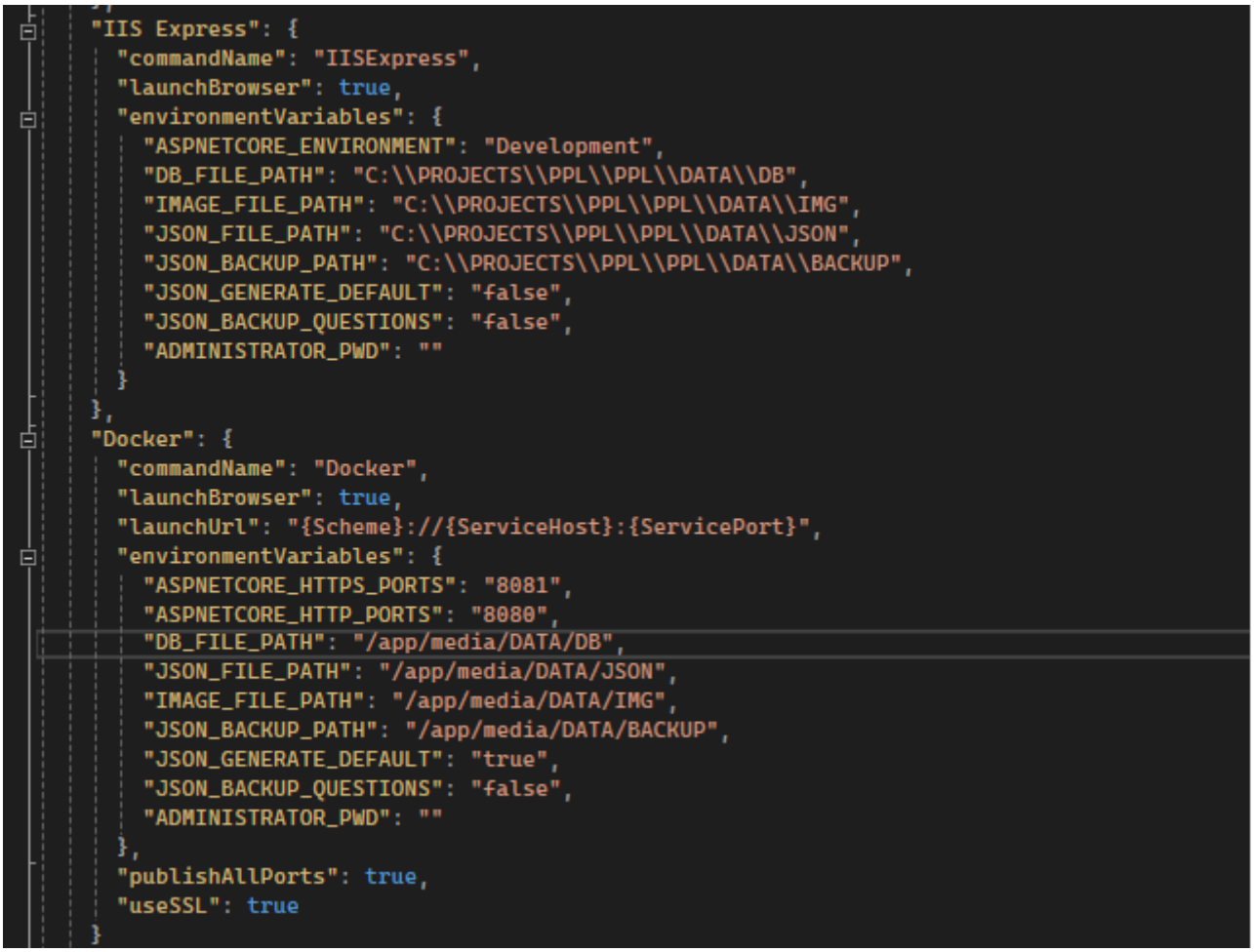
Commandline:

When I open a shell on my QNAP - it will be listed under the path **/share/DOCKERMEDIA**

```
[/share/DOCKERMEDIA] # cd
[~] # cd /share/DOCKERMEDIA
[/share/DOCKERMEDIA] #
```

From the application in the docker:

In a docker application, you would configure it as below. When you compile the solution for ISS Express, it will be on your local system.
But when you publish it to a docker, it will look at the path ***/app/media*** - which is now linked to the QNAP directory.



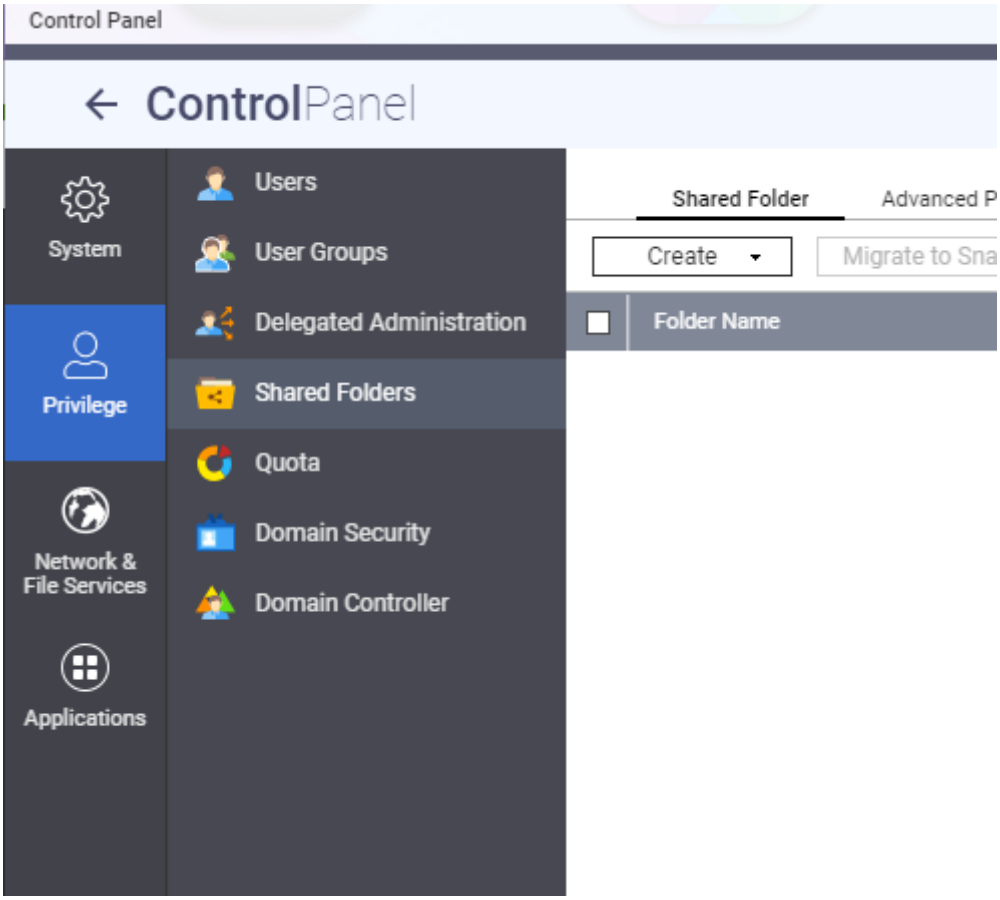
```
"IIS Express": {
  "commandName": "IISExpress",
  "launchBrowser": true,
  "environmentVariables": {
    "ASPNETCORE_ENVIRONMENT": "Development",
    "DB_FILE_PATH": "C:\\PROJECTS\\PPL\\PPL\\DATA\\DB",
    "IMAGE_FILE_PATH": "C:\\PROJECTS\\PPL\\PPL\\DATA\\IMG",
    "JSON_FILE_PATH": "C:\\PROJECTS\\PPL\\PPL\\DATA\\JSON",
    "JSON_BACKUP_PATH": "C:\\PROJECTS\\PPL\\PPL\\DATA\\BACKUP",
    "JSON_GENERATE_DEFAULT": "false",
    "JSON_BACKUP_QUESTIONS": "false",
    "ADMINISTRATOR_PWD": ""
  }
},
"Docker": {
  "commandName": "Docker",
  "launchBrowser": true,
  "launchUrl": "{Scheme}://{ServiceHost}:{ServicePort}",
  "environmentVariables": {
    "ASPNETCORE_HTTPS_PORTS": "8081",
    "ASPNETCORE_HTTP_PORTS": "8080",
    "DB_FILE_PATH": "/app/media/DATA/DB",
    "JSON_FILE_PATH": "/app/media/DATA/JSON",
    "IMAGE_FILE_PATH": "/app/media/DATA/IMG",
    "JSON_BACKUP_PATH": "/app/media/DATA/BACKUP",
    "JSON_GENERATE_DEFAULT": "true",
    "JSON_BACKUP_QUESTIONS": "false",
    "ADMINISTRATOR_PWD": ""
  },
  "publishAllPorts": true,
  "useSSL": true
}
```

Windows share:

Open window explorer and access the share:

\\192.168.111.222\dockermedia\

Ofcourse, this isn't automatically shared and we assume you've set up the share in ControlPanel.


Control Panel
ControlPanel
System
Privilege
Network & File Services
Applications
Users
User Groups
Delegated Administration
Shared Folders
Quota
Domain Security
Domain Controller
Shared Folder
Advanced P
Create
Migrate to Sna
Folder Name

Improve SMB speeds

Enable multichannel
Enable async operations I/O
Enable kernel mode SMB
Set minimum version to SMB 2 (or higher)

