Tandoor Recipes Organizer Interface (TROI) on a Synology Disk Station

Overview

This tutorial will inform about a couple of basic components and guide through the process of installing TROI on a Synology NAS using Docker. It will focus on employing Sonology's User Interface (UX) capabilities only.

What you will need ≡

- A Synology NAS of the Plus Series as those are enabled for virtualizing environments with their more powerful central processing units (CPUs). These are easily identified by the name DSxxxx+.
- Some understanding of DNS, Network, Ports, Firewall and Proxies to make sense of and secure things.
- Understanding of IP handling and Port forwarding for your Router.
- Understanding of the Synology Disk Station (DS) and the Disk Station Manager (DSM) and an Administrator Account for the DSM.
- A registered Domain which forwards (e.g. via DynDNS) to your Router/Server (Synology DS) and/or a Subdomain with according CNAME entry.

What is Tandoor? ≡

Tandoor is an open-source recipe manager that allows you to manage your collection of digital recipes and share them with friends and family. Documentation can be found here.

The system currently consists of three Docker Containers one for the Database, one for the Web-Interface and one Proxy Server to deliver the contents.

What is Docker? ≣

Docker is a platform written in Go and takes advantage of several features of the Linux kernel to deliver its functionality.
Docker provides the ability to package and run an application in a loosely isolated environment called a container. The isolation and security allow you to run many containers simultaneously on a given host (Synology).
Docker uses a technology called namespaces to provide the isolated workspace (the container). These namespaces provide a layer of isolation. Each aspect of a container runs in a separate namespace and its access is limited to that namespace.

Note: *Synology's DSM is a heavily customized Linux. Even though DSM already employs fo example components like ngnix and postgres, will separate containers of them use a part of the existing Linux base but not interfere with any of the Synology's components of functionality.*

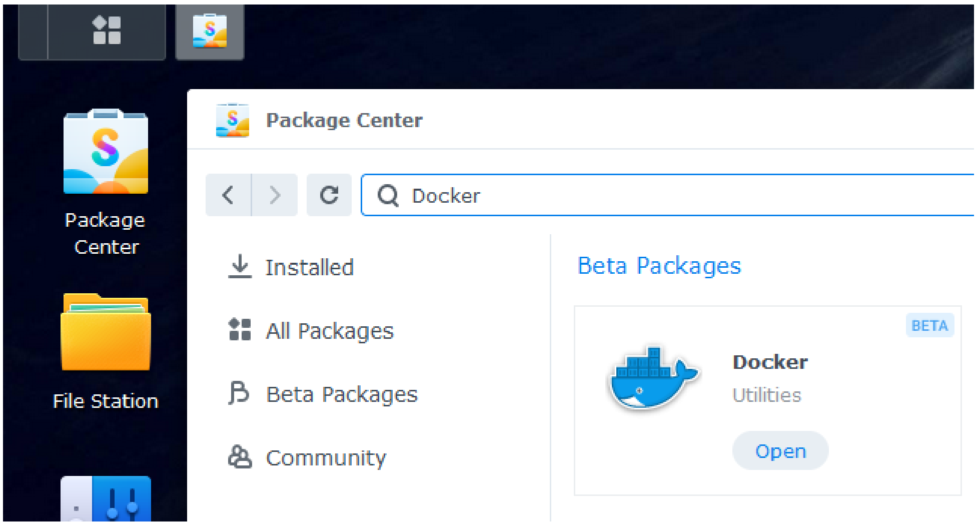
Deploy TROI ≣

1. Install the Docker Package
2. Create the folder structure and content
3. Pull the Docker Images
4. Create a Network
5. Create the Database Container
6. Create the Web Application Container
7. Create the Proxy Container
8. Start the Containers

Install the Docker Package ≣

1. Login to DSM with an Administrator Account.

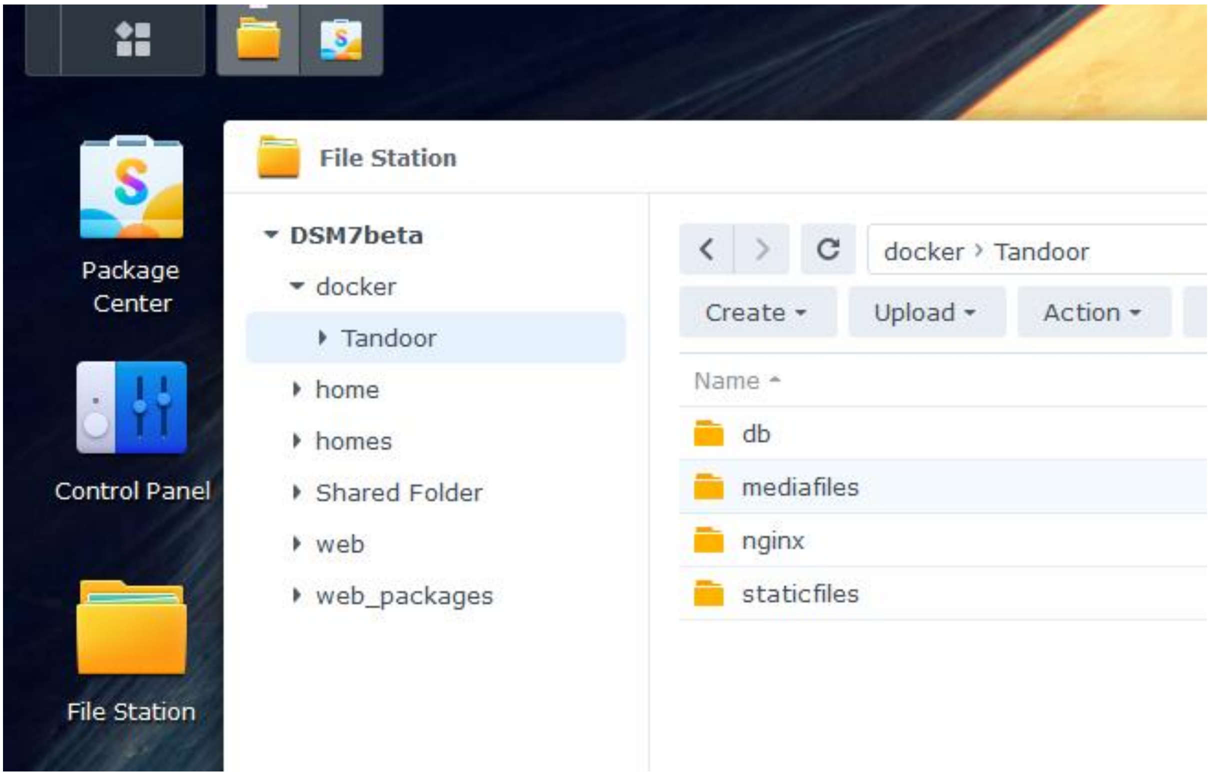
2. Proceed to **Package Center** to search for and install the Docker Package.



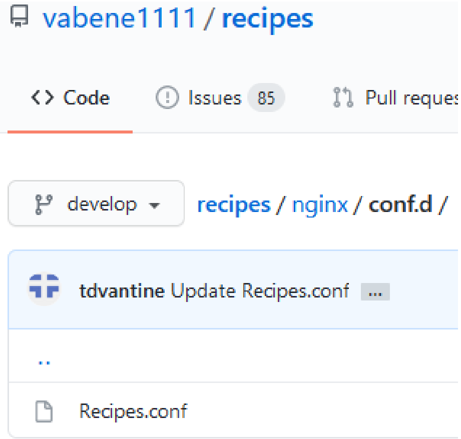
*Note: Install the **Text Editor** Package as well if not already on your DSM.*

Create the folder structure and content ≣

1. Open File Station and browse to the newly created **docker folder** to create one for this project e.g. **Tandoor**.
2. Create the following folders within the project folder. Those will be mapped
   - db
   - mediafiles
   - ngnix
   - staticfiles



3. Copy the contents of the proxy configuration from the github project page.



vabene1111 / recipes

develop

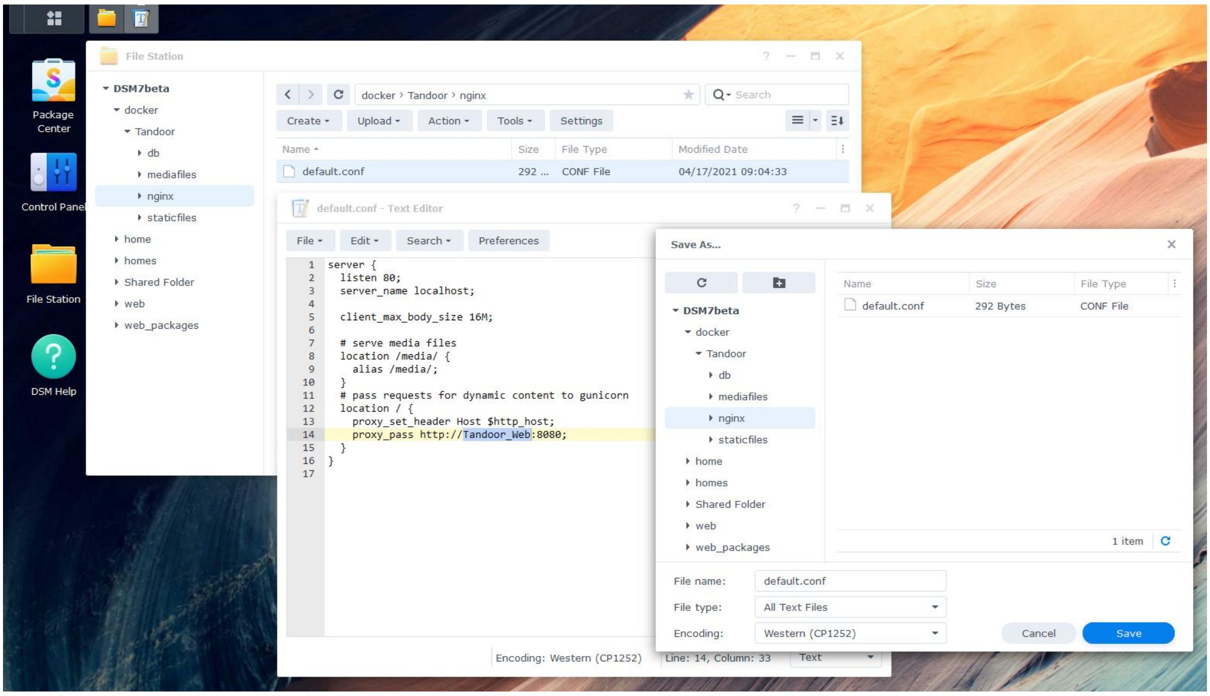
recipes / nginx / conf.d /

tdvantine Update Recipes.conf ...

..

Recipes.conf

4. Open the Text Editor on DSM and paste the contents to a new document.
   As we will later deploy our Containers under specific names, change the proxy_pass section accordingly to the Docker Container Name e.g. **Tandoor_Web**.
   Save the file as **default.conf** to **/docker/Tandoor/nginx**.



Name	Size	File Type	Modified Date
default.conf	292 ...	CONF File	04/17/2021 09:04:33

```
server {
  listen 80;
  server_name localhost;

  client_max_body_size 16M;

  # serve media files
  location /media/ {
    alias /media/;
  }
  # pass requests for dynamic content to gunicorn
  location / {
    proxy_set_header Host $http_host;
    proxy_pass http://Tandoor_Web:8080;
  }
}
```

Encoding: Western (CP1252) Line: 14, Column: 33 Text

Name	Size	File Type
default.conf	292 Bytes	CONF File

1 item

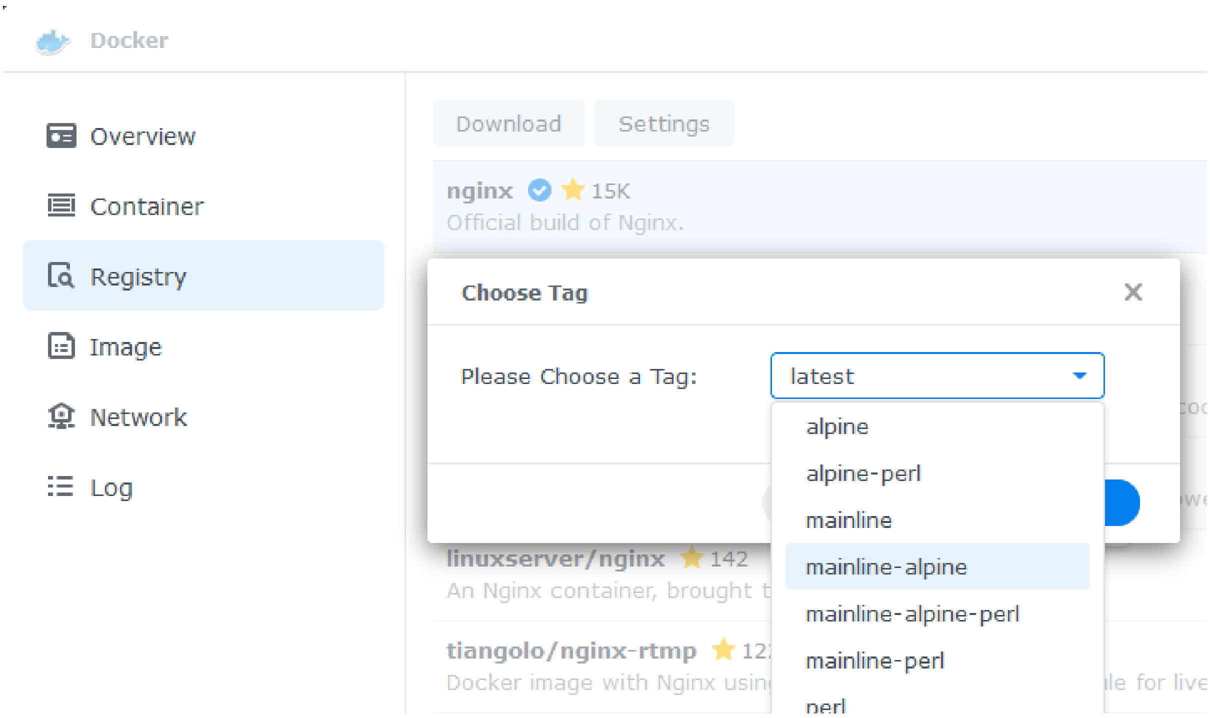
File name: default.conf

File type: All Text Files

Encoding: Western (CP1252)

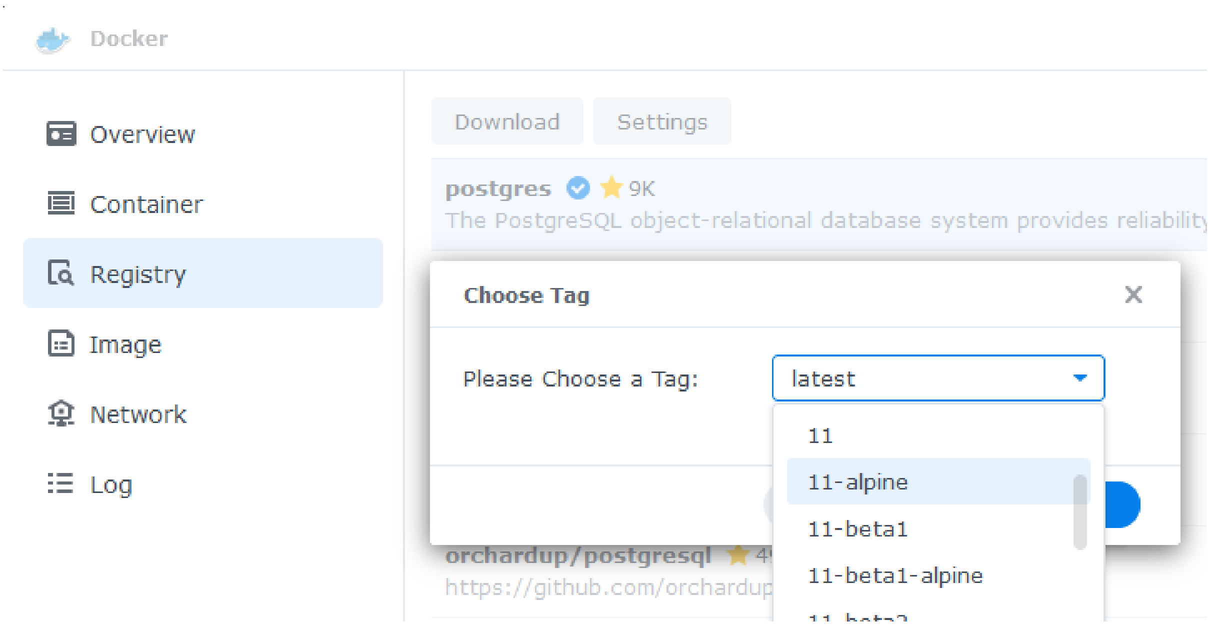
Pull the Docker Images ≡

1. Open **Docker** on DSM and choose **Registry** on the lift side menu.
   Find the below-packaged applications through the search bar and download their images choosing the according tag (version).
   - nginx (mainline-alpine)



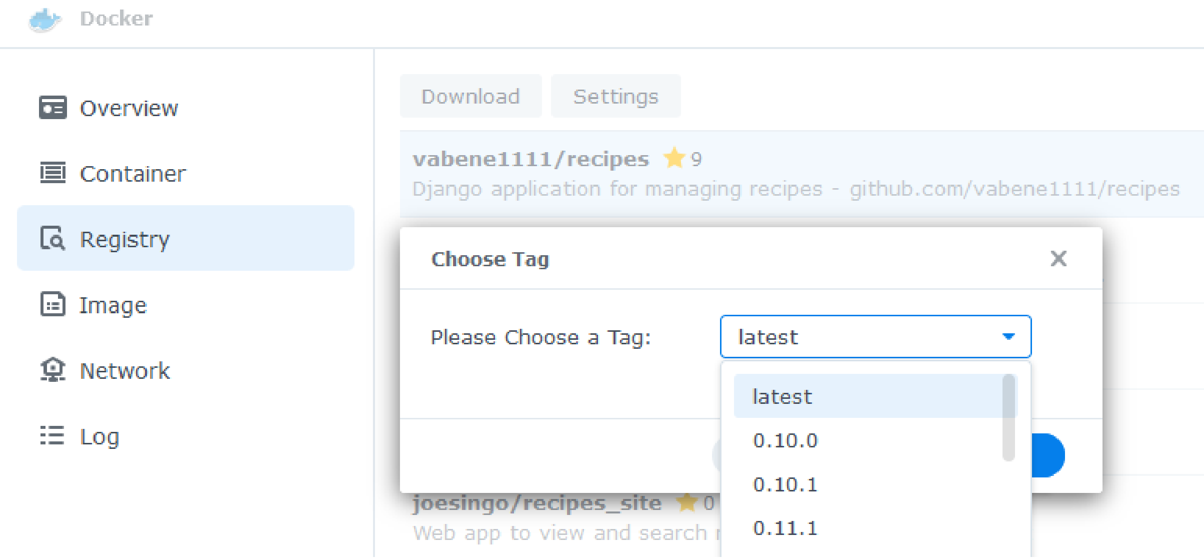
Please Choose a Tag: latest

- **postgres (11-alpine)**



Please Choose a Tag: latest

- recipes (latest)



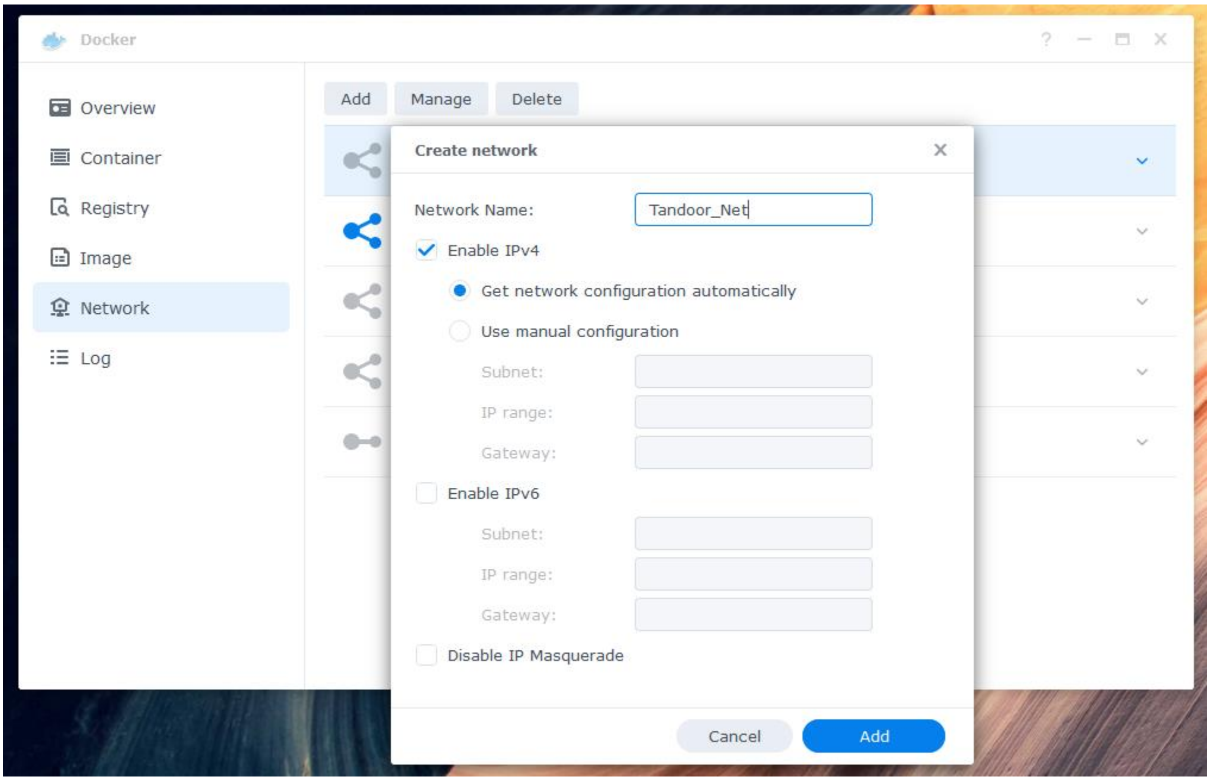
Choose Tag

Please Choose a Tag: latest

Docker
Overview
Container
Registry
Image
Network
Log
Download
Settings
vabene1111/recipes ★ 9
Django application for managing recipes - github.com/vabene1111/recipes
latest
0.10.0
0.10.1
0.11.1
joesingo/recipes_site ★ 0
Web app to view and search r

Create a Network ☰

1. For separation and ease of management e.g. Firewall rules, create a separate subnet for the project.
   In **Docker** on DSM, choose **Network** on the lift side menu and click **Add** on the top.
   Type in a **Network Name** e.g. Tandoor_Net and click **Add** on the bottom.



Create network

Network Name: Tandoor_Net

☑ Enable IPv4

● Get network configuration automatically

○ Use manual configuration

Subnet:

IP range:

Gateway:

□ Enable IPv6

Subnet:

IP range:

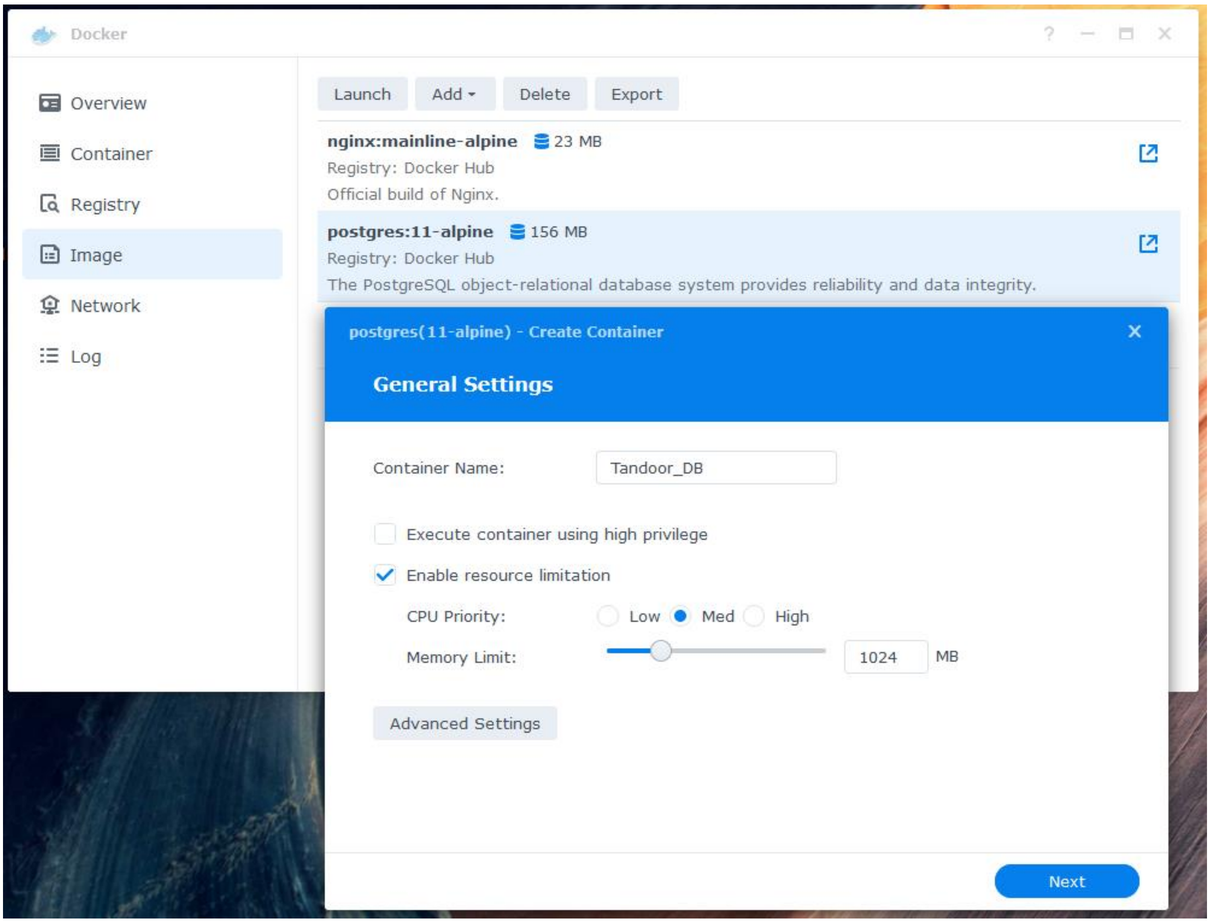
Gateway:

□ Disable IP Masquerade

Note: Docker does not require difficult Network setups. Container can simply be connected to one another using their names.

Create the Database Container ≡

1. In **Docker** on DSM, choose **Image** on the lift side menu select the **postgres:11-alpine** image and click **Launch** on the top.
   Enter in a **Container Name** e.g. Tandoor_DB and click the **Advanced Settings** button.



nginx:mainline-alpine 23 MB
Registry: Docker Hub
Official build of Nginx.

postgres:11-alpine 156 MB
Registry: Docker Hub
The PostgreSQL object-relational database system provides reliability and data integrity.

postgres(11-alpine) - Create Container

General Settings

Container Name: Tandoor_DB

☐ Execute container using high privilege

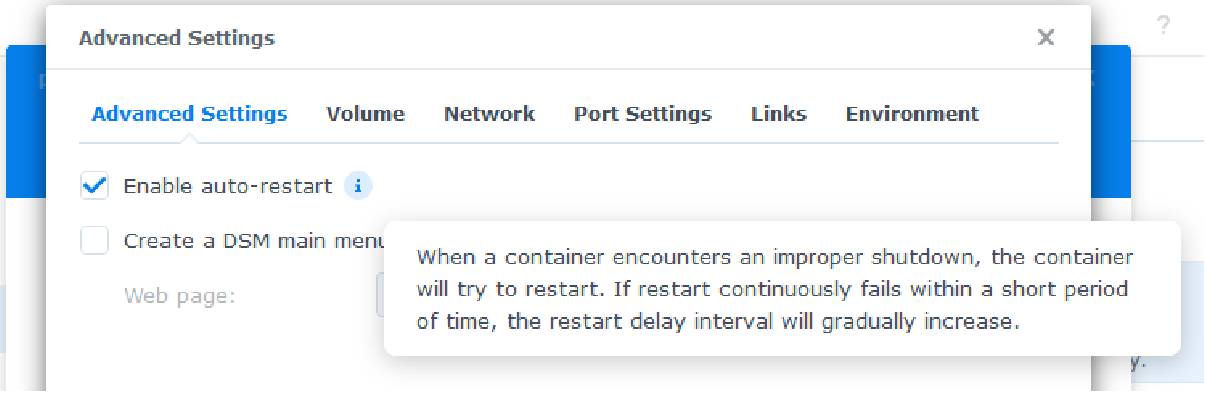
☑ Enable resource limitation

CPU Priority: ○ Low ● Med ○ High

Memory Limit: 1024 MB

Note: It is a good practice to **Enable resource limitations** up to your liking, hardware capacity and amount of containers on the host. These settings may be changed at any time.

2. Advanced Settings, Enable **auto-restart** as desired.



☑ Enable auto-restart

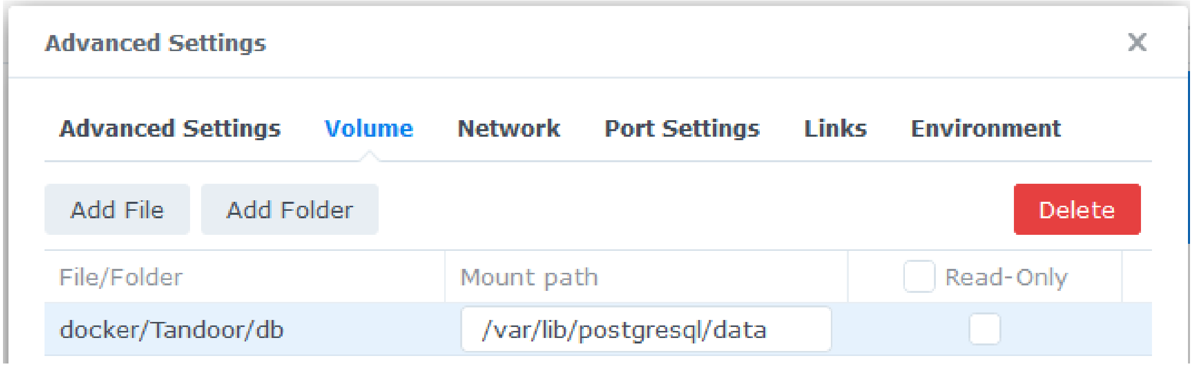
□ Create a DSM main menu

Web page:

When a container encounters an improper shutdown, the container will try to restart. If restart continuously fails within a short period of time, the restart delay interval will gradually increase.

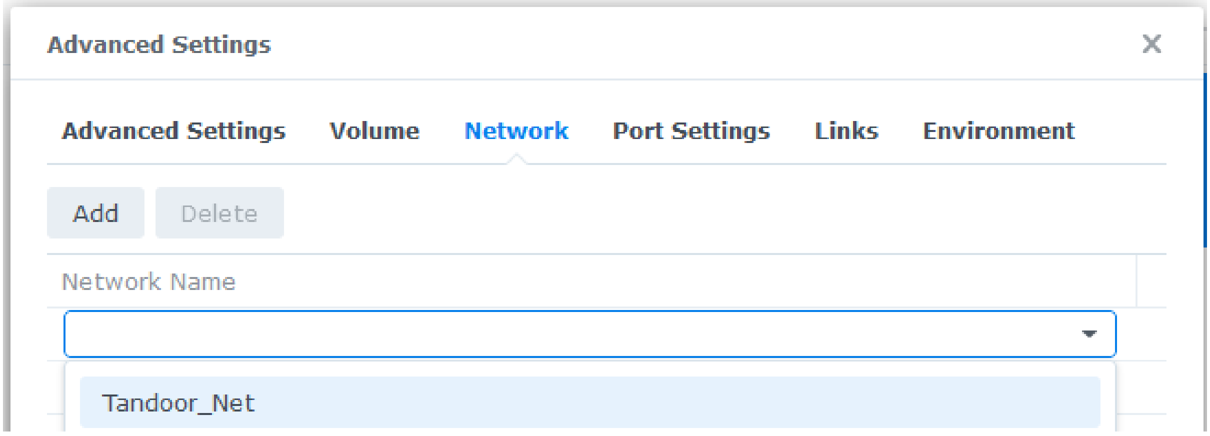
Note: It is practical to leave auto-start disabled for the beginning if logs need to be read.

3. Click **Add Folder** on the **Volume** tab and choose the **db** folder within the Tandoor project folder of the Docker tree an click Select.
   As **Mount path** enter **/var/lib/postgresql/data**



File/Folder	Mount path	□ Read-Only
docker/Tandoor/db	/var/lib/postgresql/data	□

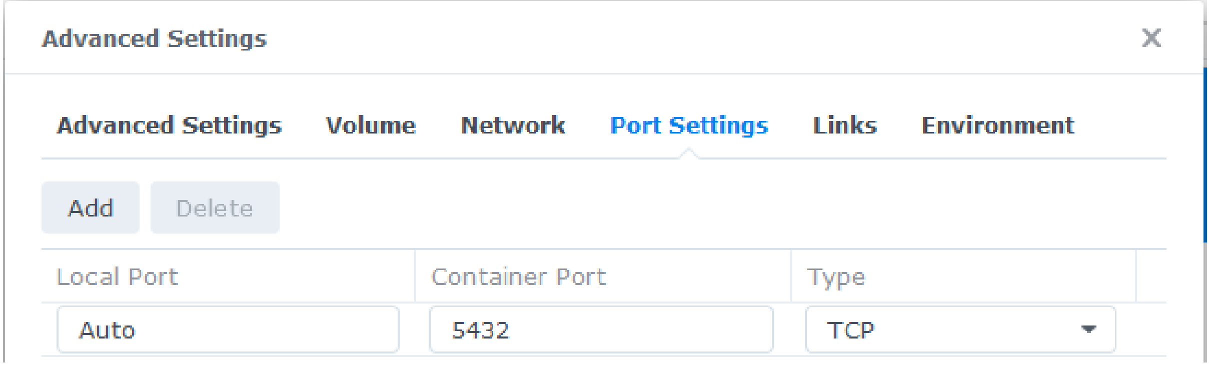
4. Click **Add** in the **Network** tab and **choose** the Network created previously e.g. Tandoor_Net.
   Once added click the **bridge** Network Name in the list and click **Delete** to limit the Container to the one dedicated Network.



Network Name

Tandoor_Net

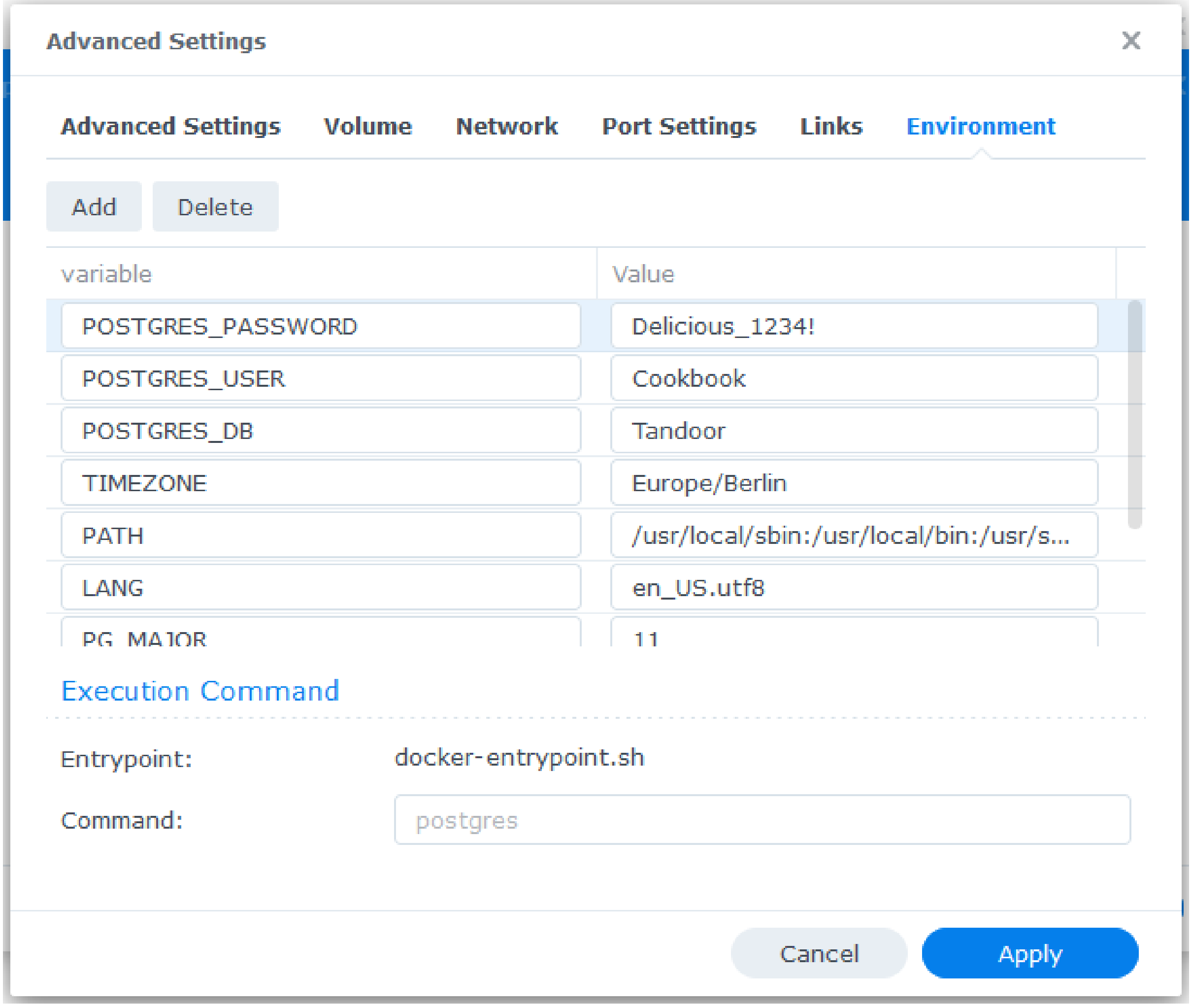
5. **Port Settings** remain as they are at Local Auto to Container 5432 as TCP.



Local Port	Container Port	Type
Auto	5432	TCP

6. **Links** settings are not required due to the use of a dedicated subnet.
7. In the **Environment** tab **Add** the below variables and values. Click the Apply button once done.

TIMEZONE	Europe/Berlin	according to your location
POSTGRES_DB	Tandoor	or another as desired
POSTGRES_USER	Cookbook	or another as desired
POSTGRES_PASSWORD	Delicious_1234!	use another secret



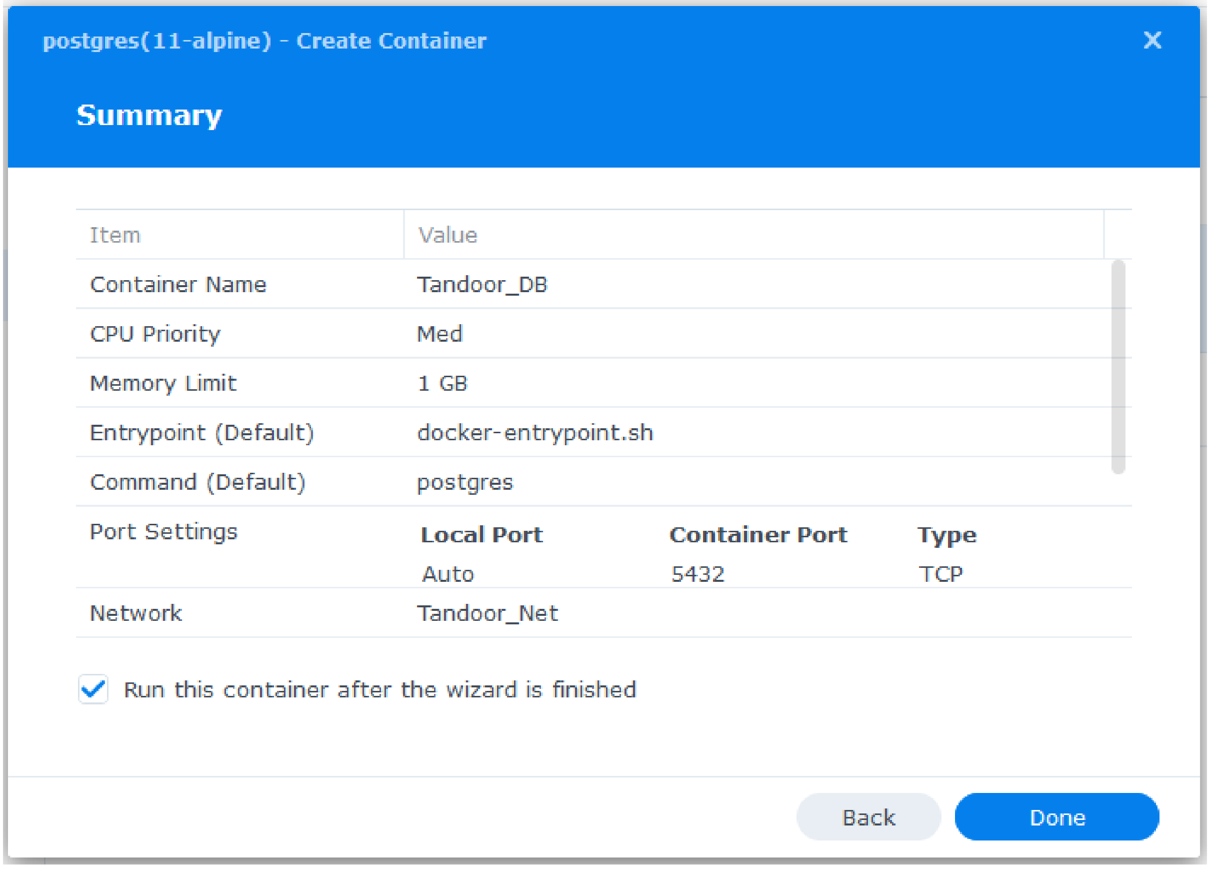
variable	Value
POSTGRES_PASSWORD	Delicious_1234!
POSTGRES_USER	Cookbook
POSTGRES_DB	Tandoor
TIMEZONE	Europe/Berlin
PATH	/usr/local/sbin:/usr/local/bin:/usr/s...
LANG	en_US.utf8
PG_MAJOR	11

Execution Command

Entrypoint: docker-entrypoint.sh

Command: postgres

8. Back at the **General Settings** click **Next**.
   In the Summary, click **Done**.



postgres(11-alpine) - Create Container

Summary

Item	Value
Container Name	Tandoor_DB
CPU Priority	Med
Memory Limit	1 GB
Entrypoint (Default)	docker-entrypoint.sh
Command (Default)	postgres
Port Settings	**Local Port** Auto · **Container Port** 5432 · **Type** TCP
Network	Tandoor_Net

☑ Run this container after the wizard is finished

Note: It is usually more practical to un-tick the run after wizard to allow starting the containers in proper order once all are created.

Create the Web Application Container ≡

1. In **Docker** on DSM, choose **Image** on the lift side menu select the **vabene1111/recipes:latest** image and click **Launch** on the top.
   Enter in a **Container Name** e.g. Tandoor_Web and click the **Advanced Settings** button.
2. Advanced Settings, Enable **auto-restart** as desired.
3. Click **Add Folder** on the **Volume** tab and map below **Folders** to below **Mount paths.**

docker/Tandoor/staticfiles	/opt/recipes/staticfiles	mind the folder name if other
docker/Tandoor/mediafiles	/opt/recipes/mediafiles	mind the folder name if other

4. Click **Add File** and map the below **File** to below **Mount path.**

docker/Tandoor/nginx/default.conf	/etc/nginx/conf.d/default.conf

5. Click **Add** in the **Network** tab and **choose** the Network created previously e.g. Tandoor_Net.
   Once added click the **bridge** Network Name in the list and click **Delete.**
6. **Port Settings** remain as they are at Local Auto to Container 8080 as TCP.
7. **Links** settings are not required due to the use of a dedicated subnet.
8. In the **Environment** tab **Add** the below variables and values. Click the Apply button once done.

DEBUG	0	
ALLOWED_HOSTS	*	
SECRET_KEY	LoremIpsumDolorDitAmet	use random string
TIMEZONE	Europe/Berlin	use same as database
DB_ENGINE	django.db.backends.postgresql	
POSTGRES_HOST	Tandoor_DB	DB Container Name
POSTGRES_PORT	5432	as DB Container
POSTGRES_DB	Tandoor	as DB Container
POSTGRES_USER	Cookbook	as DB Container
POSTGRES_PASSWORD	Delicious_1234!	as DB Container
FRACTION_PREF_DEFAULT	0	
COMMENT_PREF_DEFAULT	1	
SHOPPING_MIN_AUTOSYNC_INTERVAL	5	
GUNICORN_MEDIA	0	
REVERSE_PROXY_AUTH	0	

9. Back at the **General Settings** click **Next**.
   In the Summary, click **Done**.

Create the Proxy Container ≡

1. In **Docker** on DSM, choose **Image** on the lift side menu select the **nginx:mainline-alpine** image and click **Launch** on the top.
   Enter in a **Container Name** e.g. Tandoor_Proxy and click the **Advanced Settings** button.
2. Advanced Settings, Enable **auto-restart** as desired.
3. Click **Add Folder** on the **Volume** tab and map below **Folders** to below **Mount paths.**

docker/Tandoor/staticfiles	/staticf
docker/Tandoor/mediafiles	/media

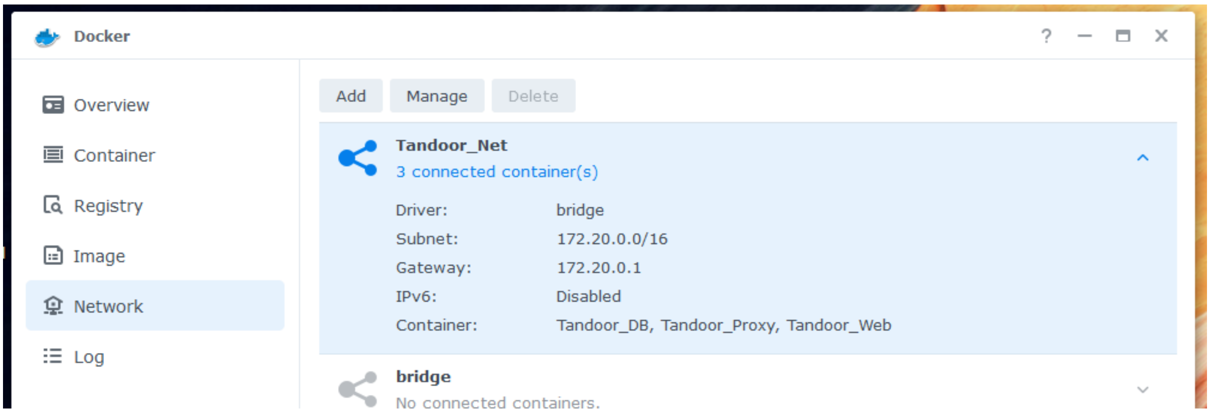
4. Click **Add File** and map the below **File** to below **Mount path.**
   Check the **Read-Only** box behind the entry.

docker/Tandoor/nginx/default.conf	/etc/nginx/conf.d/default.conf

5. Click **Add** in the **Network** tab and **choose** the Network created previously e.g. Tandoor_Net.
   Once added click the **bridge** Network Name in the list and click **Delete.**
6. For the **Port Settings** choose a Local Port which is not already used e.g 3080 on the Synology and leave the Container Port at 80 and Type as TCP.
7. **Links** settings are not required due to the use of a dedicated subnet.
8. Leave the **Environment as is** with the four default entries.
9. Back at the **General Settings** click **Next**.
   In the Summary, click **Done**.

Start the Containers ≡

1. In **Docker** on DSM, choose Network and expand the network created earlier e.g. Tandoor_Net to assure the three containers are within the same. User the Manage button and add or delete containers from the chosen network if needed.



Tandoor_Net
3 connected container(s)

Driver: bridge
Subnet: 172.20.0.0/16
Gateway: 172.20.0.1
IPv6: Disabled
Container: Tandoor_DB, Tandoor_Proxy, Tandoor_Web

bridge
No connected containers.

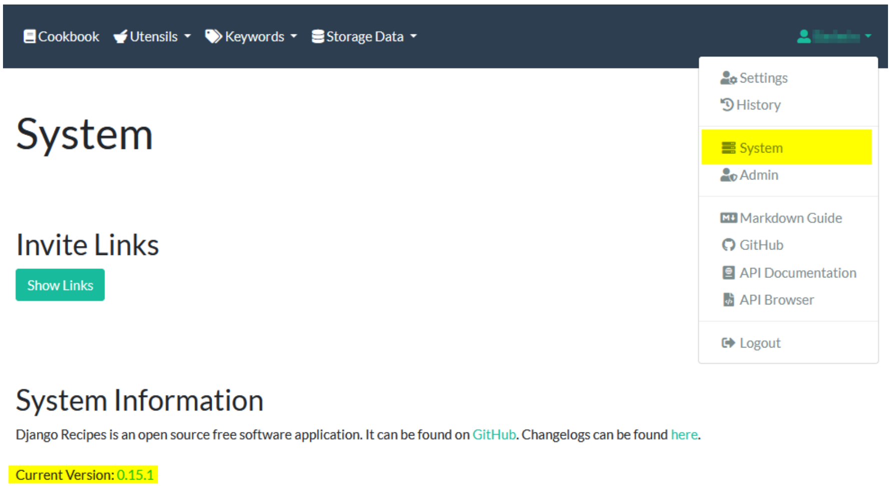
2. Start the three containers in the following order allowing a minute between starts.
   - DB Container (Tandoor_DB)
   - Web Application Container (Tandoor_Web)
   - Proxy Container (Tandoor_Proxy)

Connect to TROI ≡

Allow a couple of minutes for migration after starting above.
Connect to the Web Interface by entering the IP of your Synology followed by the Proxy Container Local Port e.g. http://169.254.0.10:3080 (substitute for your environment accordingly).
You will be asked to create a Super User upon first Login.
Make sure to keep a proper record of the credentials used for that one.

Update TROI ≡

1. Check for new Versions on Github.
2. Check for the current Version by logging in Tandoor with an **Admin** user, expanding the **User Menu** on the top right corner, selecting **System** and finding the **Current Version**.



System

Invite Links

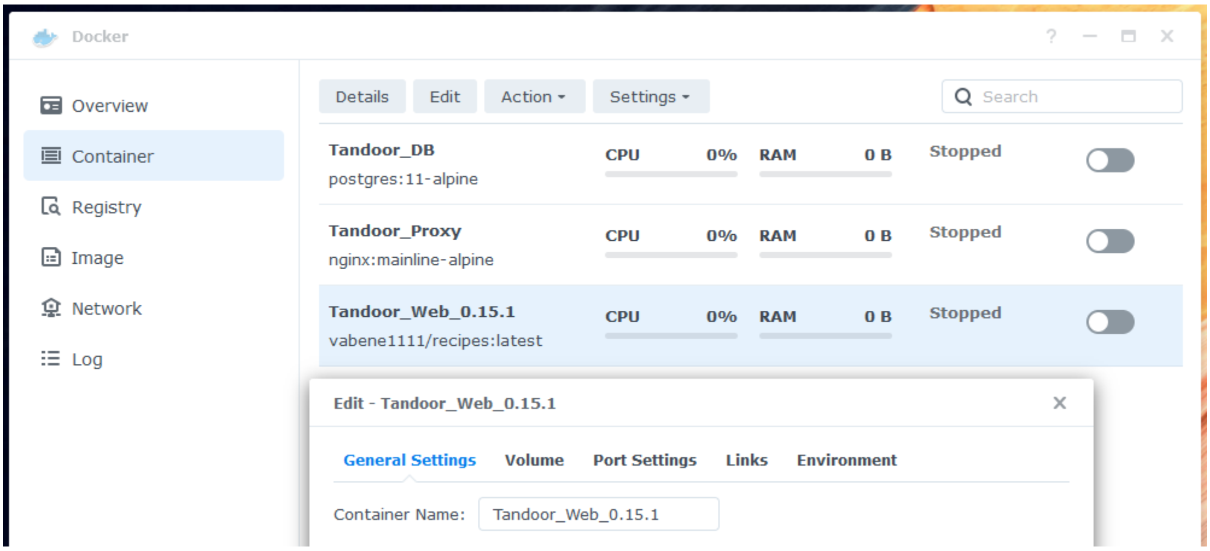
System Information

Django Recipes is an open source free software application. It can be found on GitHub. Changelogs can be found here.

Current Version: 0.15.1

3. Check the github docker compose file file for the **image:** node. The Data Base Container (postgres:11-alpine) and Proxy Container (nginx:mainline-alpine) usually should not require an update.
4. **Stop** the three Tandoor **containers**.

5. **Edit** and **rename** the Web Application Container (Tandoor_Web) by e.g amending the current version number.

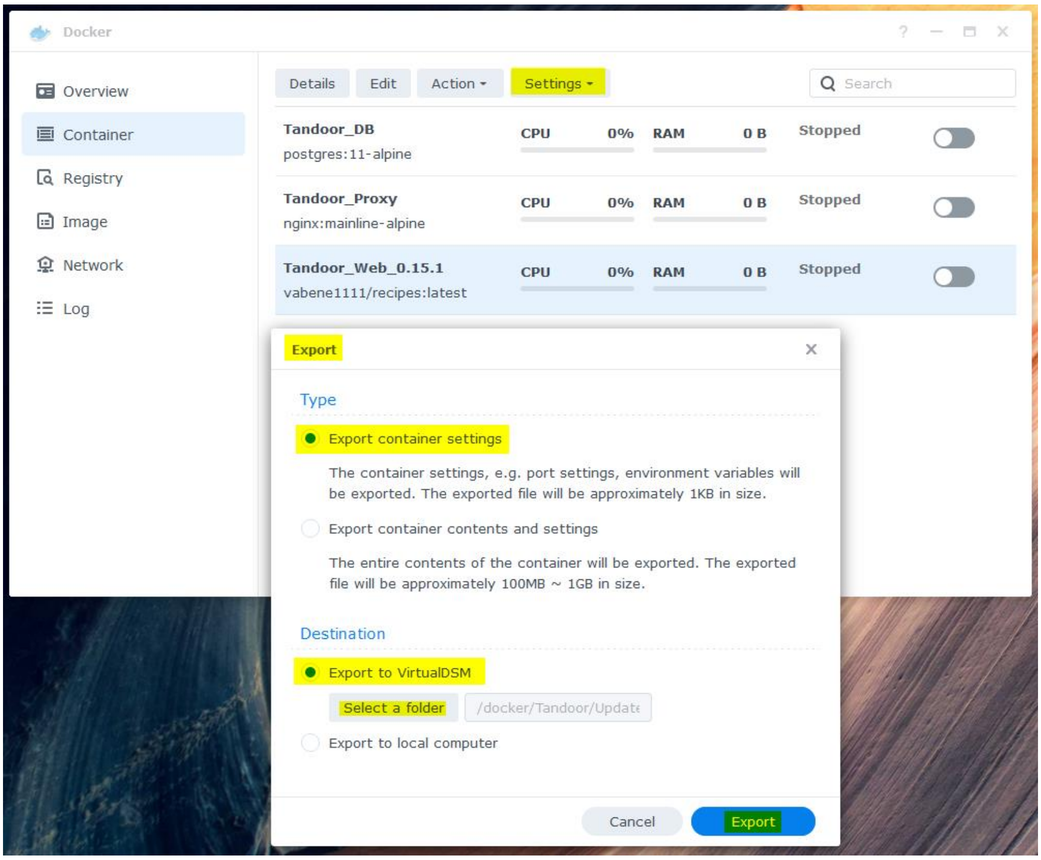


Tandoor_DB postgres:11-alpine	CPU	0%	RAM	0 B	Stopped	○
Tandoor_Proxy nginx:mainline-alpine	CPU	0%	RAM	0 B	Stopped	○
Tandoor_Web_0.15.1 vabene1111/recipes:latest	CPU	0%	RAM	0 B	Stopped	○

Edit - Tandoor_Web_0.15.1

Container Name: Tandoor_Web_0.15.1

6. **Select** the Tandoor **Web Application Container**, click **Settings**, click **Export**. In the next window make sure to select **Export container settings** and the desired location e.g. a folder on DSM and click **Export**.



Tandoor_DB postgres:11-alpine	CPU	0%	RAM	0 B	Stopped	○
Tandoor_Proxy nginx:mainline-alpine	CPU	0%	RAM	0 B	Stopped	○
Tandoor_Web_0.15.1 vabene1111/recipes:latest	CPU	0%	RAM	0 B	Stopped	○

Export

Type

● Export container settings

The container settings, e.g. port settings, environment variables will be exported. The exported file will be approximately 1KB in size.

○ Export container contents and settings

The entire contents of the container will be exported. The exported file will be approximately 100MB ~ 1GB in size.

Destination

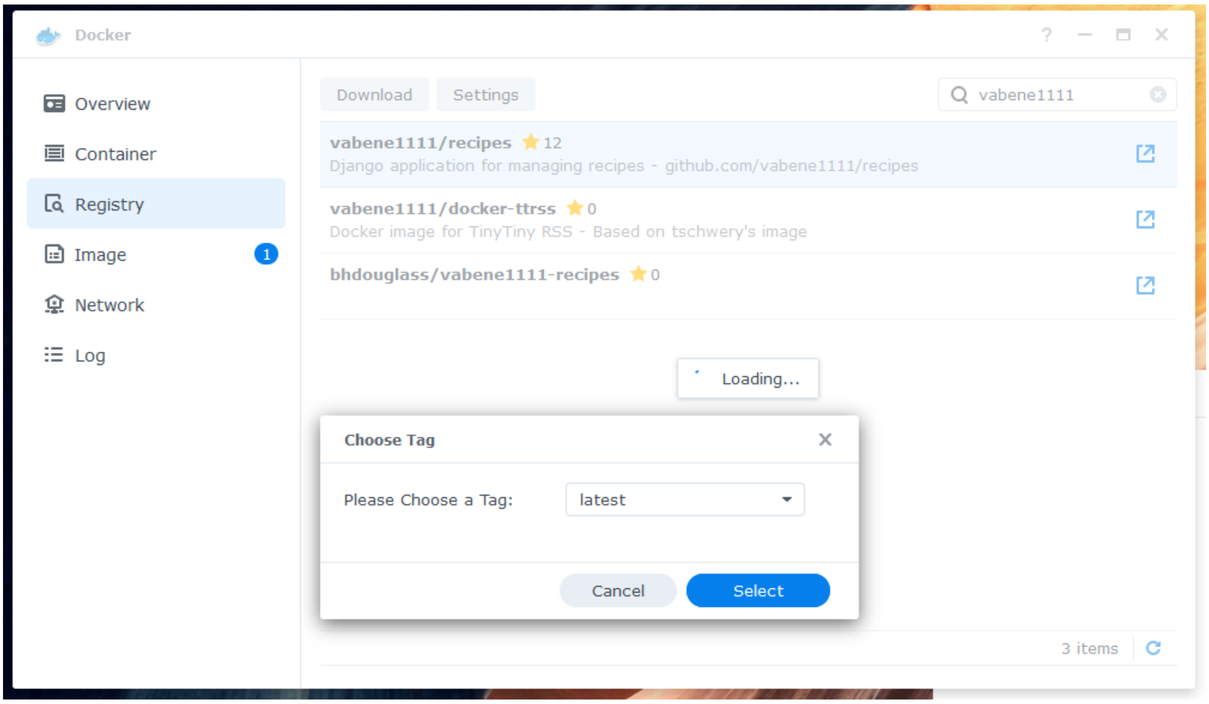
● Export to VirtualDSM

Select a folder /docker/Tandoor/Update

○ Export to local computer

7. In **Registry**, pull the latest container image by searching and double clicking the desired Registry entry **vabene1111/recipes,** choose the **latest** tag (version) and click **Select.** A number will show besides Image but allow some time to finish the

pull. DSM will show a notification once completed.



vabene1111

vabene1111/recipes ★ 12
Django application for managing recipes - github.com/vabene1111/recipes

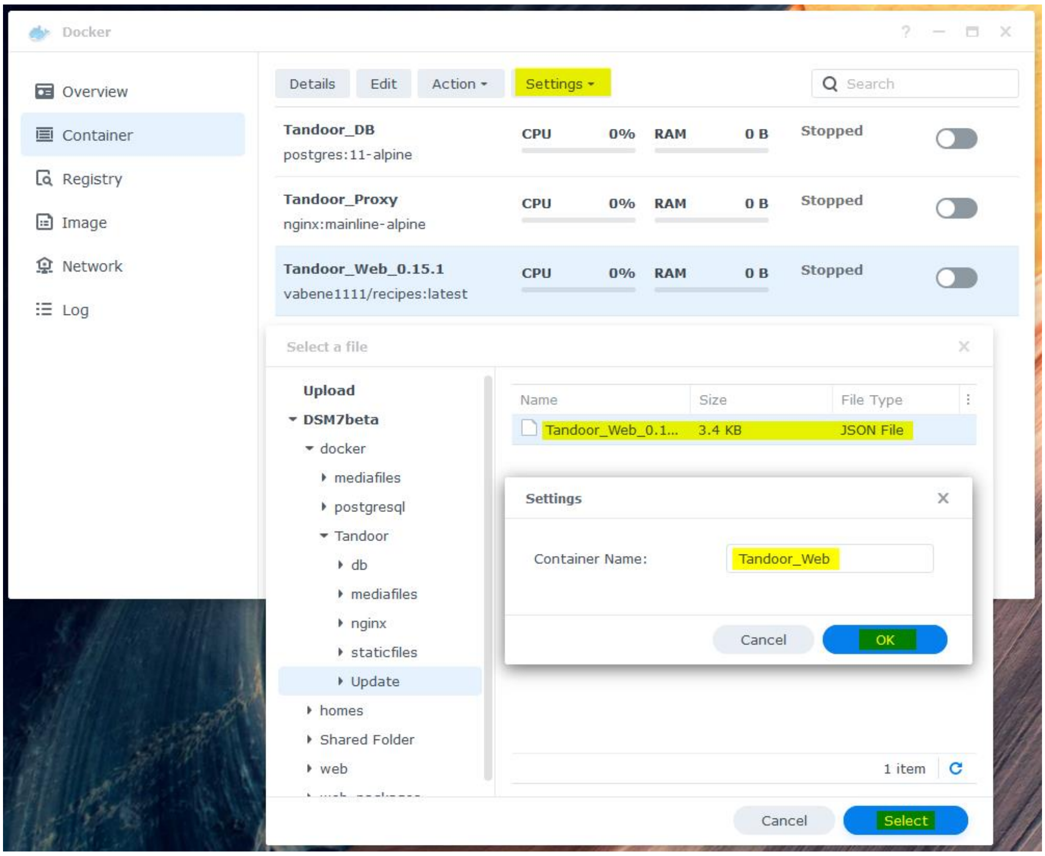
vabene1111/docker-ttrss ★ 0
Docker image for TinyTiny RSS - Based on tschwery's image

bhdouglass/vabene1111-recipes ★ 0

Choose Tag

Please Choose a Tag: latest

8. In **Container** click **Settings** and choose **Import**. Choose the **file** created in the **previous step** and click **Select**. Make sure to enter the **original Container Name** and click **OK**.



Tandoor_DB postgres:11-alpine	CPU	0%	RAM	0 B	Stopped
Tandoor_Proxy nginx:mainline-alpine	CPU	0%	RAM	0 B	Stopped
Tandoor_Web_0.15.1 vabene1111/recipes:latest	CPU	0%	RAM	0 B	Stopped

Name	Size	File Type
Tandoor_Web_0.1...	3.4 KB	JSON File

Settings

Container Name: Tandoor_Web

9. Start the three containers in the following order allowing a minute between starts.
- DB Container (Tandoor_DB)

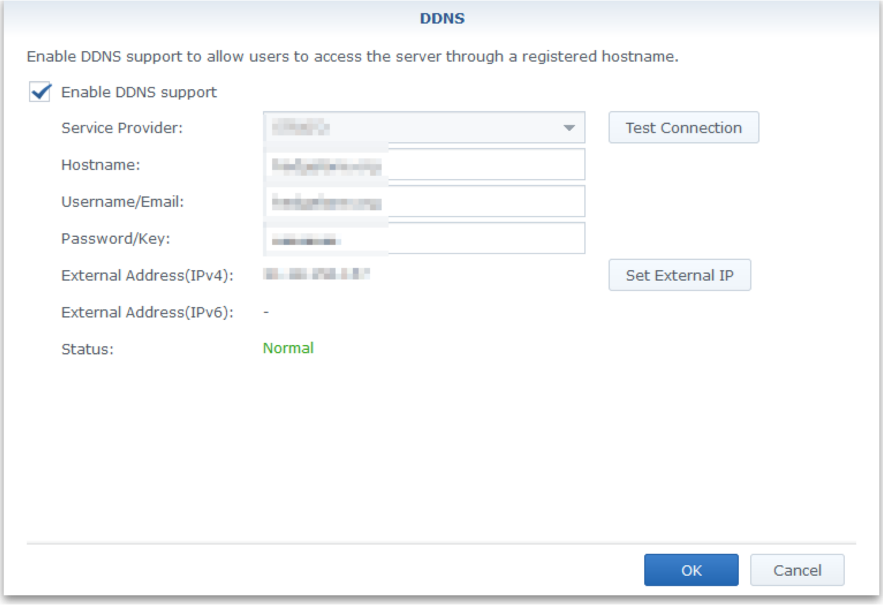
- Web Application Container (Tandoor_Web)
- Proxy Container (Tandoor_Proxy)

10. Connect to the Web Interface by entering the IP of your Synology followed by the Proxy Container Local Port e.g. http://169.254.0.10:3080 (substitute for your environment accordingly).
11. Make sure to **review the Release Notes** on Github carefully for new features and settings.

Expose TROI to the Interwebs

All roads lead to Rome but covering them all here would be out of scope. Hence, this section assumes a common setup; a non-static IP as endpoint which is broadcasted by the Synology Disk Station to a DynDNS Provider where a Subdomain leads via the Router Port 443 through the Synology Reverse Proxy and Firewall to a Let's Encrypt Certified HTTPS connected TROI.
It is also assumed, if you read this you made yourself **very aware of the dangers** and **familiar with the precautions of exposing** your Network and Server (Synology) **to the WWW**.

1. Acquire a **Domain** and activate DynDNS for it. For this tutorial **domain.tld** is used.
2. **Configure** your **Router** to pass Port 80 (HTTP) and 443 (HTTPS) to the Synology.
3. In **DSM open Control Panel**, **External Access** choose the **DDNS** tab and **add** a **Service Provider** entry accordingly. The Status should show Normal after a TestConnection and calling your domain.tld in a browser from outside of your local network lead to the Disk Station.



DDNS

Enable DDNS support to allow users to access the server through a registered hostname.

☑ Enable DDNS support

Service Provider: [redacted]

Hostname: [redacted]

Username/Email: [redacted]

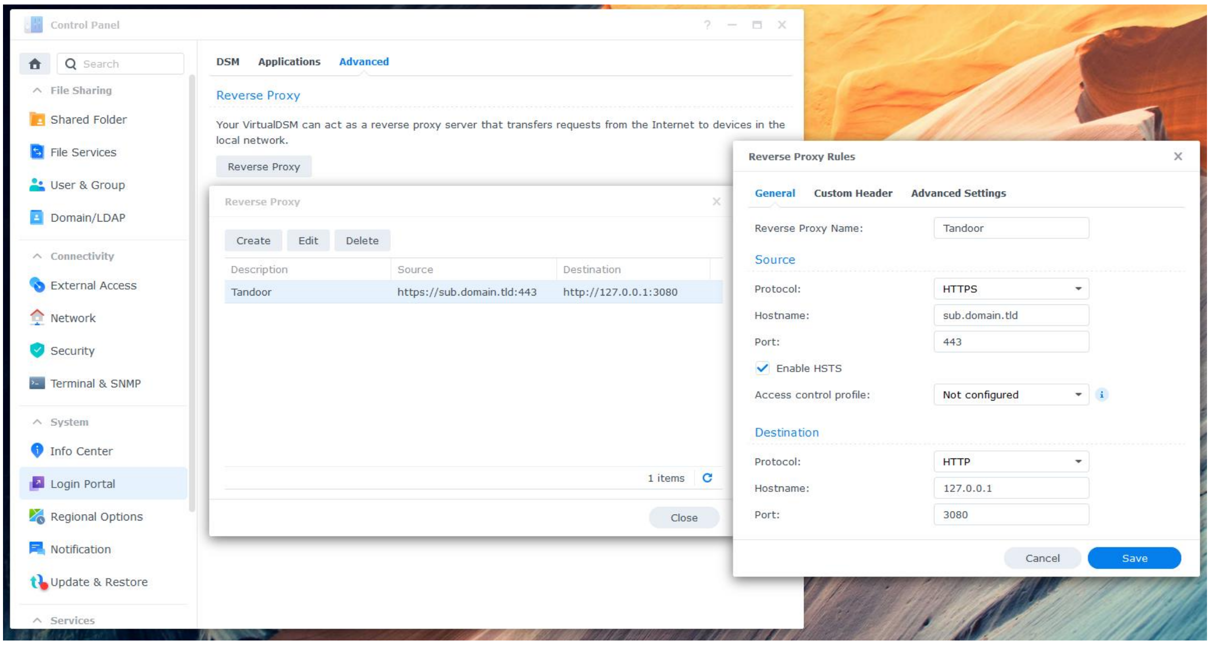
Password/Key: [redacted]

External Address(IPv4): [redacted]

External Address(IPv6): -

Status: Normal

4. Acquire a **Subdomain** with your DNS Service Provider. For this tutorial **sub.domain.tld** is used. Create a **CNAME** record for the Subdomain pointing to the Domain e.g. **domain.tld.**. Mind, the dot behind tld and calling your sub.domain.tld in a browser from outside of your local network should now lead to the Disk Station.
5. In **DSM** open **Control Panel**, **Login Portal** choose the **Advanced** tab and click **Reverse Proxy** and **Create** to enter the according settings.



Reverse Proxy

Your VirtualDSM can act as a reverse proxy server that transfers requests from the Internet to devices in the local network.

Description	Source	Destination
Tandoor	https://sub.domain.tld:443	http://127.0.0.1:3080

Reverse Proxy Name: Tandoor

Source

Protocol: HTTPS

Hostname: sub.domain.tld

Port: 443

☑ Enable HSTS

Access control profile: Not configured

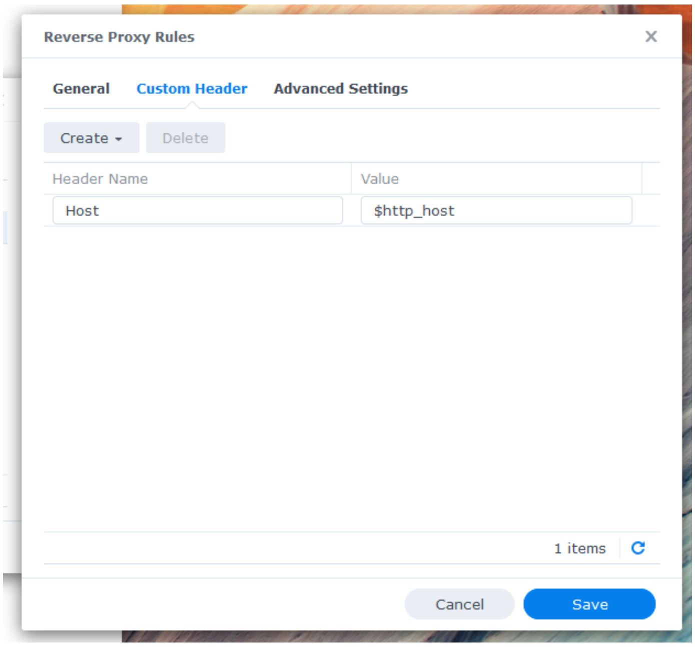
Destination

Protocol: HTTP

Hostname: 127.0.0.1

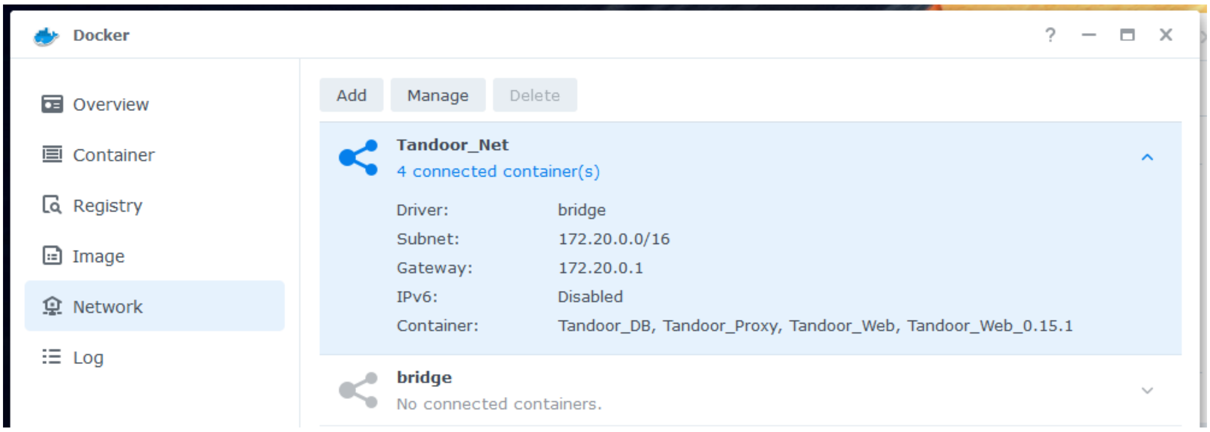
Port: 3080

6. Choose the **Custom Header** tab in the Reverse Proxy Rules entry and click Create. Enter **Host** as Header Name and **$http_host** as Value and click Save.



Header Name	Value
Host	$http_host

7. In **DSM** open **Docker**, choose **Network, expand the Network** for the Tandoor Containers and note the **Gateway IP** e.g. 172.20.0.1.

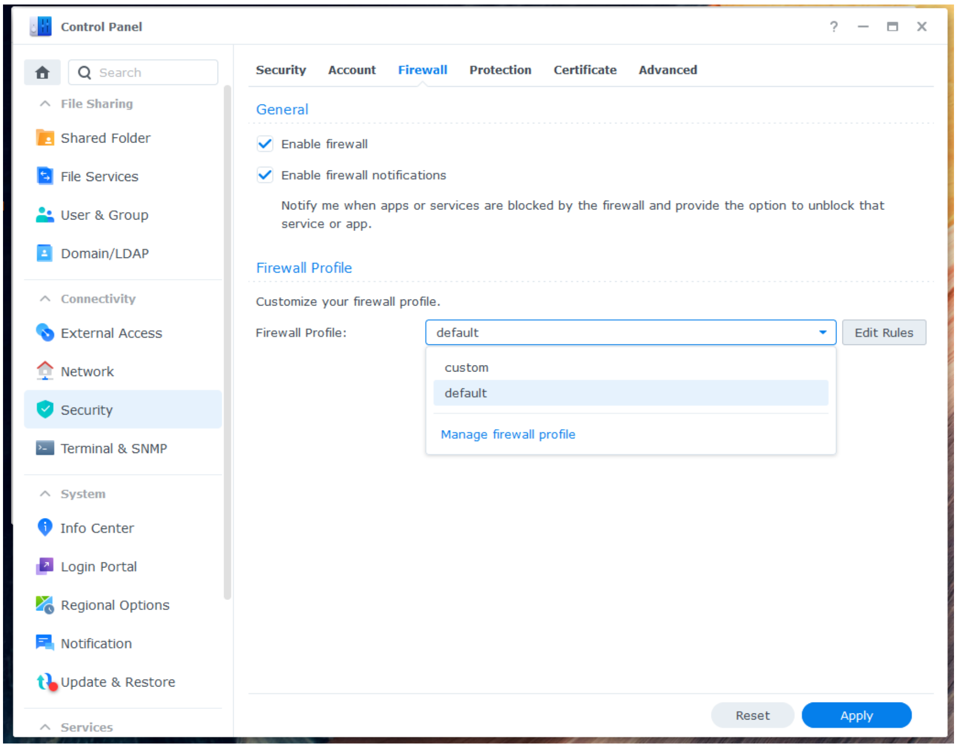


Tandoor_Net
4 connected container(s)

Driver:	bridge
Subnet:	172.20.0.0/16
Gateway:	172.20.0.1
IPv6:	Disabled
Container:	Tandoor_DB, Tandoor_Proxy, Tandoor_Web, Tandoor_Web_0.15.1

bridge
No connected containers.

8. In **DSM** open **Control Panel**, choose **Security** and the **Firewall tab**. **Enable** the **Firewall** and **Notifications**. Choose a profile and Click **Edit Rules**.



General

☑ Enable firewall

☑ Enable firewall notifications

Notify me when apps or services are blocked by the firewall and provide the option to unblock that service or app.

Firewall Profile

Customize your firewall profile.

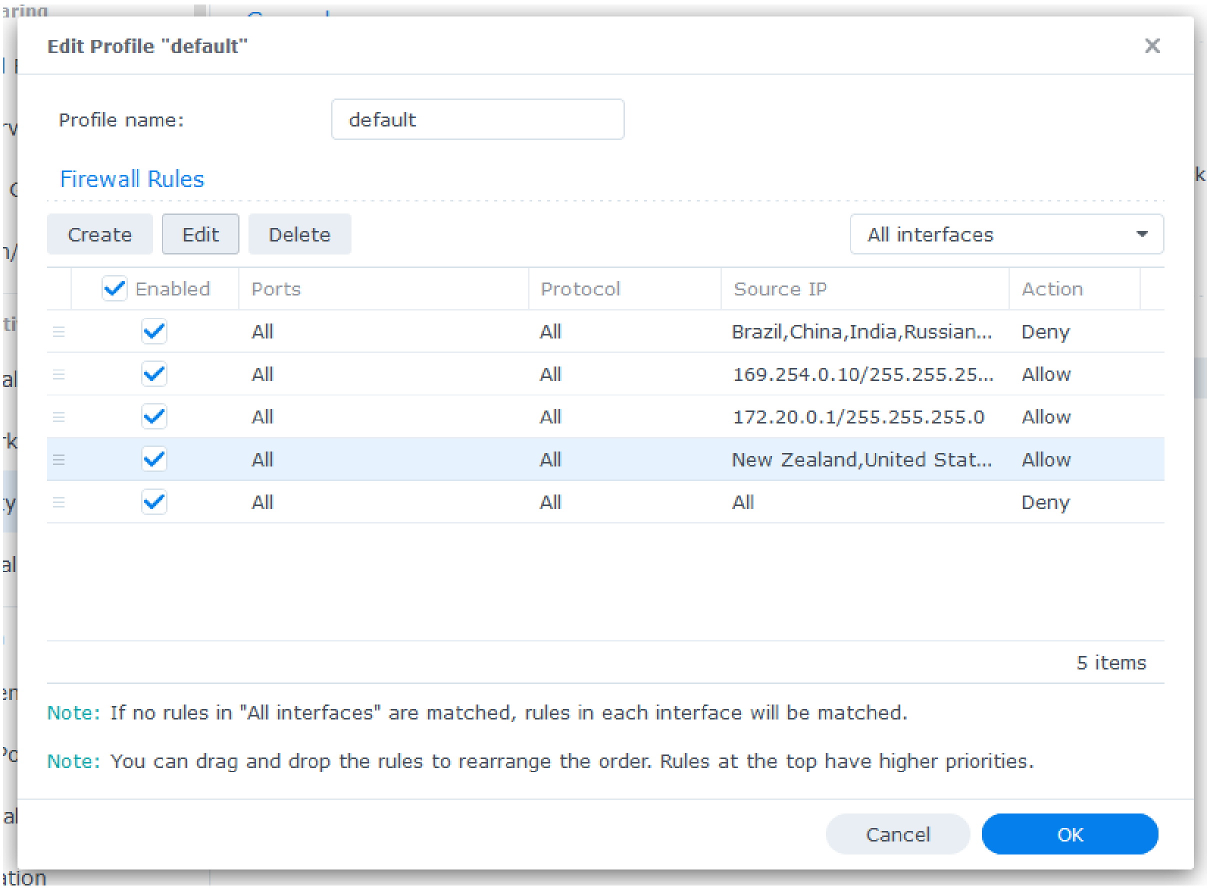
Firewall Profile: default

custom
default
Manage firewall profile

9. **Create** the below **rules** (at a minimum) replacing the DS local IP (e.g. 169.254.0.10), Docker Network IP (e.g. 172.20.0.1) and your Source Country (e.g. NZ,USA) accordingly and Click OK.

   Note: It is good practice to **restrict Ports, Protocol and Source** as much as possible as well as having the **BRIC Country's** as the **first** and an **All Deny Rule**

as the **last** entry. Also consider the **security** of your **Router/Network**.



Edit Profile "default"

Profile name: default

Firewall Rules

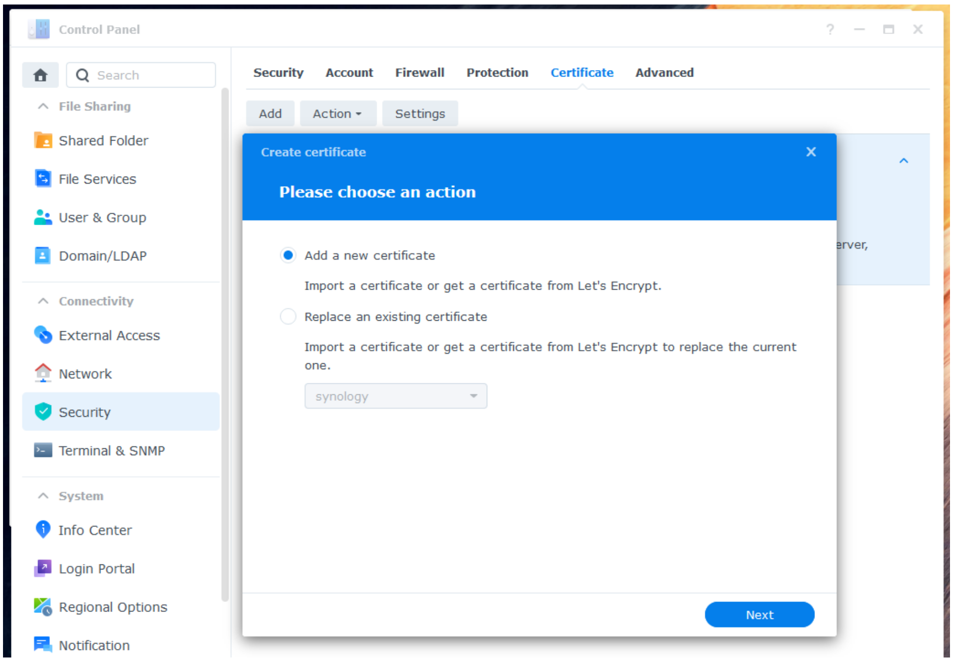
	Enabled	Ports	Protocol	Source IP	Action
≡	☑	All	All	Brazil,China,India,Russian...	Deny
≡	☑	All	All	169.254.0.10/255.255.25...	Allow
≡	☑	All	All	172.20.0.1/255.255.255.0	Allow
≡	☑	All	All	New Zealand,United Stat...	Allow
≡	☑	All	All	All	Deny

5 items

Note: If no rules in "All interfaces" are matched, rules in each interface will be matched.

Note: You can drag and drop the rules to rearrange the order. Rules at the top have higher priorities.

10. In **DSM, Control Panel**, choose **Security** and the **Certificate** tab and click **Add**. Choose to **add a new Certificate** and click **Next**.



Create certificate

Please choose an action

● Add a new certificate

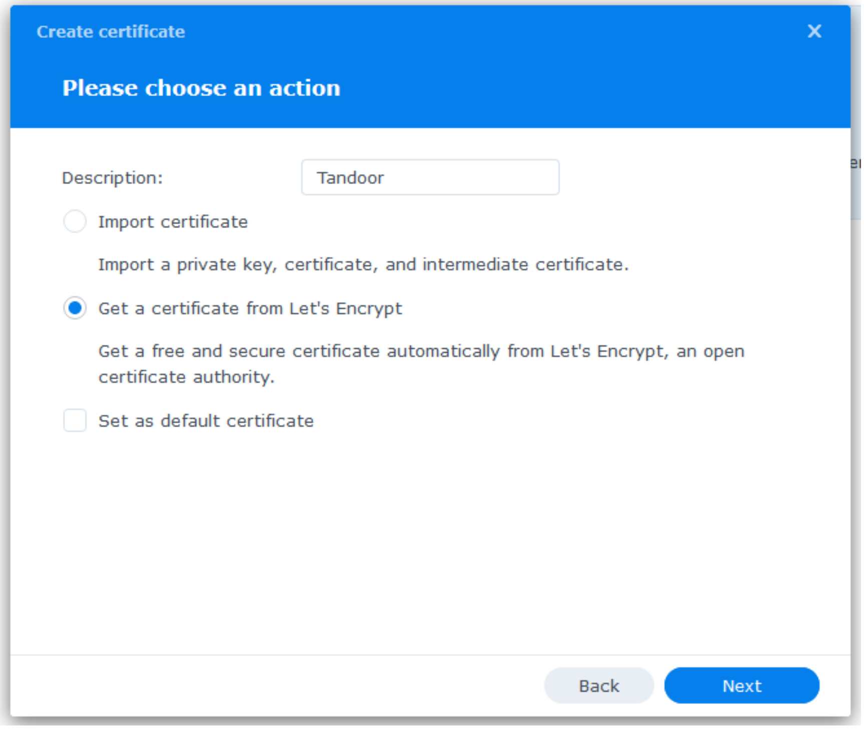
Import a certificate or get a certificate from Let's Encrypt.

○ Replace an existing certificate

Import a certificate or get a certificate from Let's Encrypt to replace the current one.

synology

11. Enter a **Description** e.g. Tandoor and choose to **get a certificate from Let's Encrypt**.



Create certificate

Please choose an action

Description: Tandoor

○ Import certificate

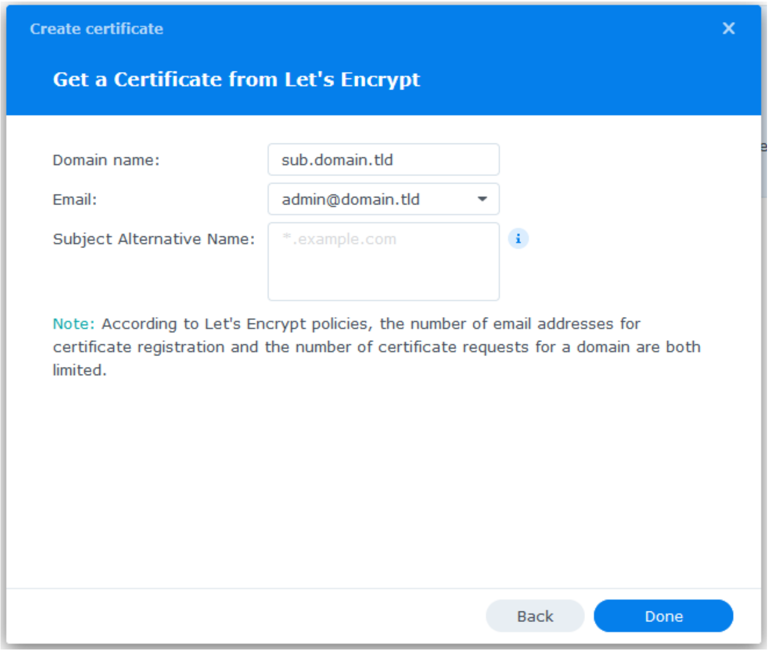
Import a private key, certificate, and intermediate certificate.

● Get a certificate from Let's Encrypt

Get a free and secure certificate automatically from Let's Encrypt, an open certificate authority.

□ Set as default certificate

12. Enter your **Subdomain** and **Email** accordingly and click **Done** to trigger the Certificate Signing Request (CSR). Your Disk Station must be reachable through Port 80 (soon only 443 required) by Let's Encrypt for a successful Certificate generation. Note: You may also use your Domain (domain.tld) as **Domain name** entry and amend the Subdomain (sub.domain.tld) to it by filling it in as **Subject Alternative Name**.



Create certificate

Get a Certificate from Let's Encrypt

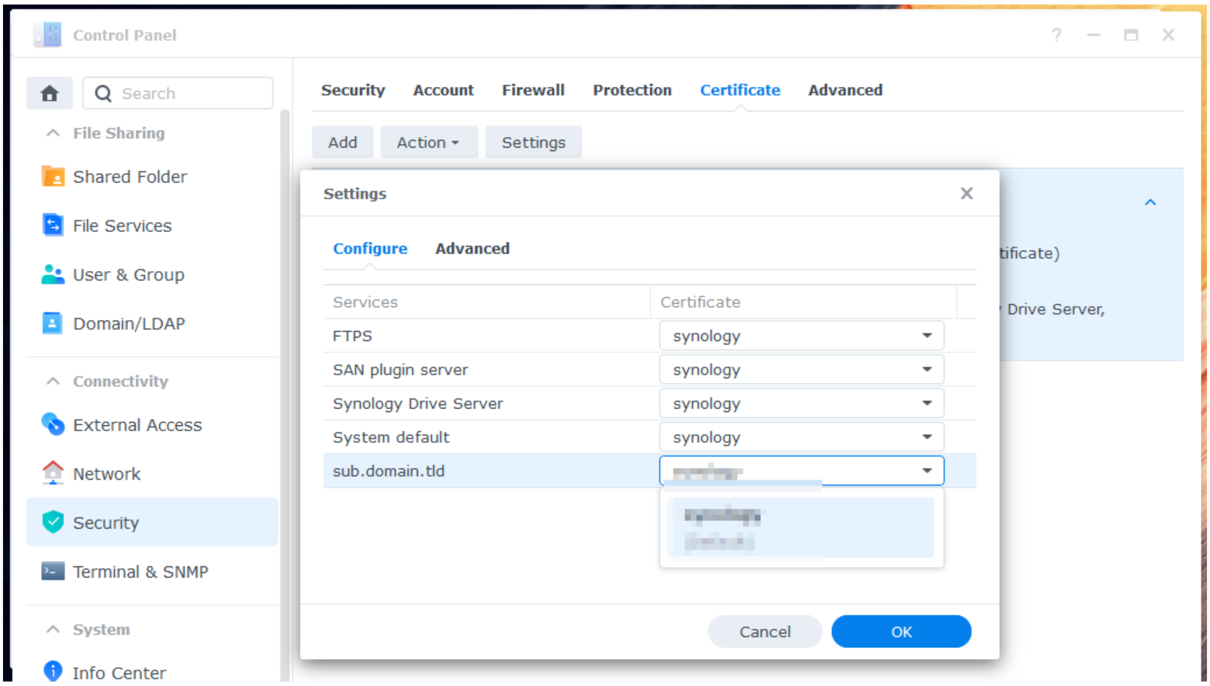
Domain name: sub.domain.tld

Email: admin@domain.tld

Subject Alternative Name: *.example.com

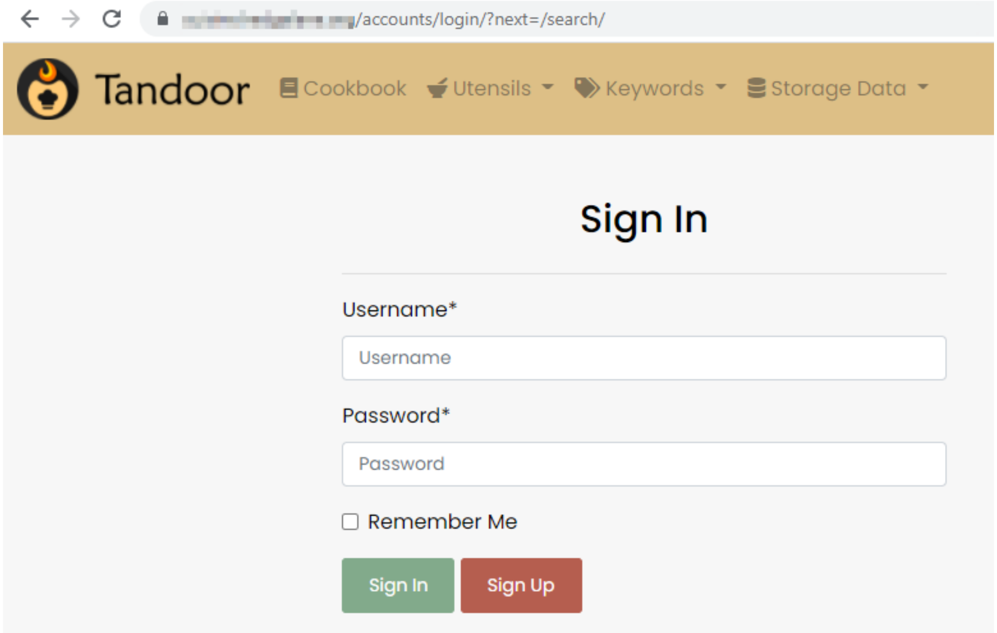
Note: According to Let's Encrypt policies, the number of email addresses for certificate registration and the number of certificate requests for a domain are both limited.

13. In **DSM**, **Control Panel**, choose **Security** and the **Certificate** tab and click **Settings**. Choose the **previously created certificate** for use with the sub.domain.tld and click **OK**.



Services	Certificate
FTPS	synology
SAN plugin server	synology
Synology Drive Server	synology
System default	synology
sub.domain.tld	[redacted]

14. You are now able to browse to e.g. https://sub.domain.tld and enjoy Tandoor.



Sign In

Username*

Username

Password*

Password

□ Remember Me

Sign In | Sign Up

Conclusion ≡

This tutorial showed you a Synology UX only approach to get a working environment of Tandoor running and update it in a couple of easy steps as well as connecting to it locally.

It also provided an example to expose Tandoor on the internet through a Subdomain using a Let’s Encrypt Certificate which will be automatically renewed by the Disk Station. Though, your caution and commonsense is required to secure you environment accordingly.

This should be a good starting point to try the approaches yourself and get your recipes in order, more often used and easily shared with the awesome Tandoor Application.

A more automated but more technical way for deployment through the Command Line Interface (CLI) and Secure Shell (SSH) would be to facilitate Docker Compose which may be found here.

Coming soon is a view of a back-up solution for the database using another Container.

Learn more ≡

- GitHub Tandoor
- Synology
- What can I do to enhance the security of my Synology NAS?
- Docker
- ngnix
- Xpenology