

Tartalomjegyzék

1.	Általános információk	4
2.	Mappajogosultságok Dockerhez	6
2.1.	Docker felhasználó	9
3.	Konténerek	11
3.1.	Konténerek telepítése	12
3.2.	Konténerek frissítése	14
3.3.	Torrent kliensek	15
3.3.1.	Transmisison	15
3.3.2.	qbittorrent	17
3.4.	Média Server	18
3.4.1.	Emby	18
3.4.2.	Plex server	19
3.5.	Felhő szolgáltatás	20
3.5.1.	Nextcloud	20
3.5.2.	Seafile	21
3.6.	Reklámblokkoló:	21
3.6.1.	Adblocker - félkész	21
3.6.2.	Pihole - félkész	22
3.6.3.	AdguardHome – félkész a telepítés	26
3.7.	Photo Management	29
3.7.1.	PhotoPrism	29
3.8.	Konténerek netes elérése	30
3.8.1.	Dinamikus DNS frissítés – DDNS	30
3.8.2.	Dinamikus DNS frissítés – router saját címével	30
3.8.3.	Dinamikus DNS frissítés – Duckdns	30
3.8.4.	Dinamikus DNS frissítés – ddclient	31
3.8.5.	Dinamikus DNS frissítés – cloudflare	31
3.9.	Webszerver és tanúsítványkezelő	32
3.9.1.	swag – alias letsencrypt	32
3.9.2.	Nginx Proxy Manager	35
3.9.2.1.	SSL és tanúsítvány generálás	35
3.9.2.2.	Nextcloud netes elérése	36
3.9.2.3.	Qbittorrent netes elérése	37
3.9.2.4.	Jelszavas védelem konténerekhez – netdata	38

3.10.	Eszközök szinkronizálása és backupja	40
3.10.1.	Resilio : hálózati szinkronizálás	40
3.10.2.	Rsync: HDD-n belüli másolás	43
3.11.	Duplicati – tökéletes Backup	43
3.12.	Több konténer egy helyen – Multistack	43

OMV 5 + Portainer + reverse proxy – szerver mindenkinek

Ez a leírás azoknak készül, akik már nem elsőként telepítenek OMV-t vagy épp már gyakorlottak.

Előző Portaineres írásom az alapokról itt megtekinthető:

https://logout.hu/bejegyzes/stigma/portainer_alapok.html

Az OMV telepítésről sokat nem írok, mert lényegében egyszerű és hamar fel lehet tenni bármilyen gépre (ARM vagy x86-os PC) viszont az alábbi oldalak jobb és értékesebb leírásokat tartalmaznak:

1. X86/PC vonalon itt találtok leírásokat/videókat:

dbTech oldala: <https://dbtechreviews.com/>

dbTech Youtube csatornája: [Youtube](#)

2. ARM vonalon pl: Raspberry PI vonalon itt találtok leírásokat/videókat:

PcMac: <https://pcmac.biz/>

Hivatalos fórumoldal: <https://forum.openmediavault.org/>

Fontos: Telepítéskor **nem ajánlott** az adat HDD-k csatlakoztatása. Csak akkor csatoljuk fel őket és formázzuk meg, amikor a rendszer már feltelepült illetve a frissítéseket is elvégeztük.

1. Általános információk

Ha a rendszert feltelepítettük, elsőként frissítsük az OMV-t a legújabb verzióra, amit az alábbi két paranccsal tudunk megtenni SSH-n keresztül. SSH-hoz Windows alól használhatjuk a Putty nevű programot.

- apt update
- apt full-upgrade -y

Ezek után (mivel friss a rendszerünk), célszerű a felesleges csomagokat (package) törölni

- apt-get autoclean
- apt-get autoremove

A felhasználók többsége a NAS-t távolról is el akarja érni, ezért a későbbiekben leírom majd, hogy kell. Ehhez mindenképp telepíteni kell egy webszervert a NAS-ra, ami alapértelmezetten a 80-as (http) és 443-as (https) portokat használja. Ezeket meg is lehet változtatni, de nem mindig ajánlott. Ha távolról is el akarjuk érni a NAS-t, ezeket a portokat a routerben is ki kell majd nyitni.

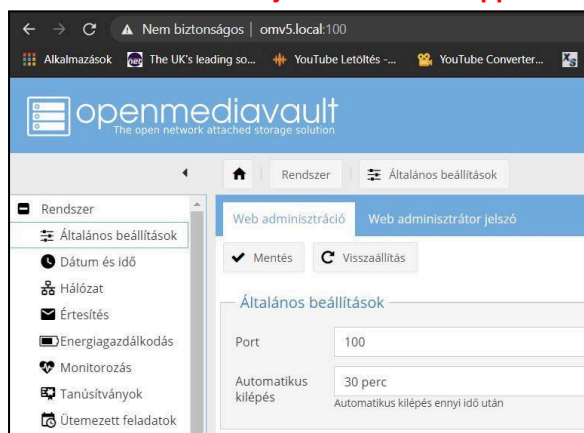
A webszerveren kívül is elérhetőek a NAS tartalmai, de akkor a domain cím után portokat kell megadni.

Ha a NAS-ra telepítünk webszervert (Treatik, swag, Nginx Proxy Managert) **akkor mindenképp meg kell változtatnunk a OMV portját** melyet a <http://NASIP-vel> érünk el, mivel mint írtam az alapértelmezett port amit az OMV Webgui használ az a 80-as port. **A portcserét mindenképp a webszerver telepítése előtt kell megtenni!**

Ezenkívül az alábbi beállítások ajánlottak:

- **Jelszócsere és port csere** : Rendszer \ Általános beállítások:
Itt a Port-ot tegyék át bármilyen más portra (pl: 90, 100, 180 vagy bármi egyéb)
Az automatikus kilépést hagyjátok alapértelmezésen vagy áttehetitek akár 30 percre is.

A Web adminisztrátor jelszót mindenképp módosítsátok meg a másik fülön!!!



- **Hálózat**:
Ha itt megadtok egy gép és tartománynevet, akkor ezt IP helyett a böngészőbe beírva eléritek az omv-t, úgy hogy nem kell tudni mire fixáltátok az ip-címet a routerben.

Erre a példa az alábbi: a NAS-t a routerben mondjuk fixáltatok 192.168.0.10-re
Gépnév: NAS
tartomány: nas.local

Ha böngészőben el akarjátok érni a NAS-t akkor elég az alábbi beírni:
<http://nas.local:portszám>
jelen esetben, ha 100-ra váltottatok, akkor <http://nasip:100>

Ez példaként egy olyan név lehet, amit ti is elnevezhettek, de mindezt a telepítés alatti felületen is megtehetitek, mikor az install-t végzitek.

o **Interface-ek:**

IPV 4 esetén ha fixáltuk a routerben az IP címét a NAS-nak, akkor az alábbiakat célszerű választani:

Eljárás: Statikus

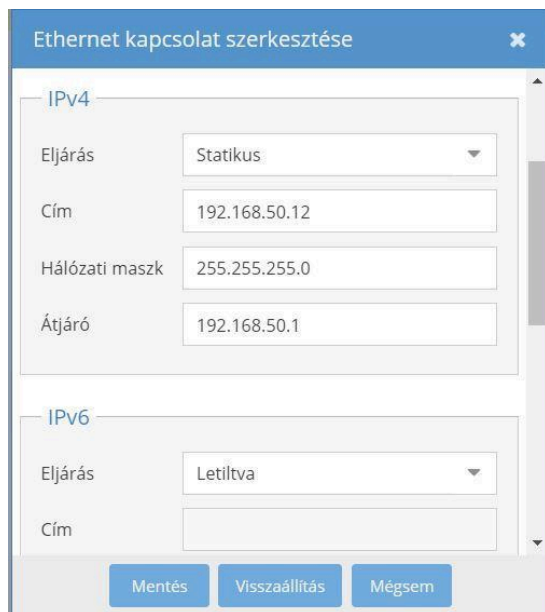
Cím: NAS IP címe, ami mondjuk 192.168.0.10

Hálózati maszk: 255.255.255.0

Átjáró: 192.168.0.1 (a router ip-jét kell itt megadni)

IPV6 => ha nem használod, akkor tiltsd le (disabled)

Haladó beállítás \ DNS => 8.8.8.8.



Nálam a NAS IP-je 192.168.50-el kezdődik ezért látjátok, hogy a NAS-t a 192.168.50.12-es ip-n érem el.

2. Mappajogosultságok Dockerhez

Vágjunk is bele egyből a közepébe. Először hozzuk létre magunknak a szükséges mappáinkat, amelyek az alábbiak lesznek:

- Docker: Bármilyen nevet adhatunk nekik, ebben fogjuk tárolni a szükséges Docker konfigurációkat, illetve egyéb adatokat.
- Torrent vagy Download mappa: Ide fogunk torrentezni.

Lépünk be az OMV-be, majd Megosztott mappák : Hozzáadás

Megosztott mappa hozzáadása

Név: Docker

Meghajtó: Datadisk [55.67 MiB (1%) used, 9.70 GiB available]

Útvonal: Docker/

A mappa útvonala a megosztáshoz A megadott mappa létre fog jönni, ha az még nem létezik.

Jogosultságok: Mindenki: olvasás/írás
Megosztott mappa útvonalának fájllelési módja

Megjegyzés

Mentés Visszaállítás Mégsem

Nagyon fontos megjegyeznem, hogy a jogosultságnak: **Mindenki (ír/olvasnak)** kell lennie.

Felül megjelenik egy sárga sáv, ahol nyomjunk az **Igen**-re.

Ugyanígy hozzuk létre a Torrent vagy épp Download mappát, melybe majd letölteni fogunk.

Ezek után nincs más feladatunk, mint elérhetővé kell tenni a hálózati eszközök számára, amit az alábbi helyen tudunk megtenni:

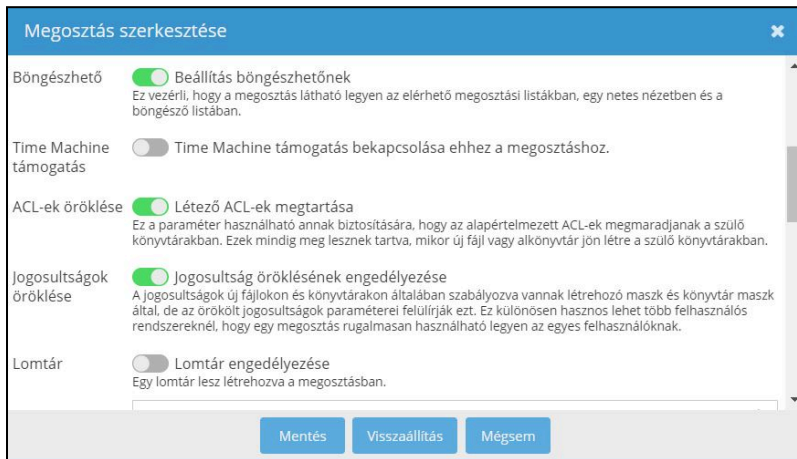
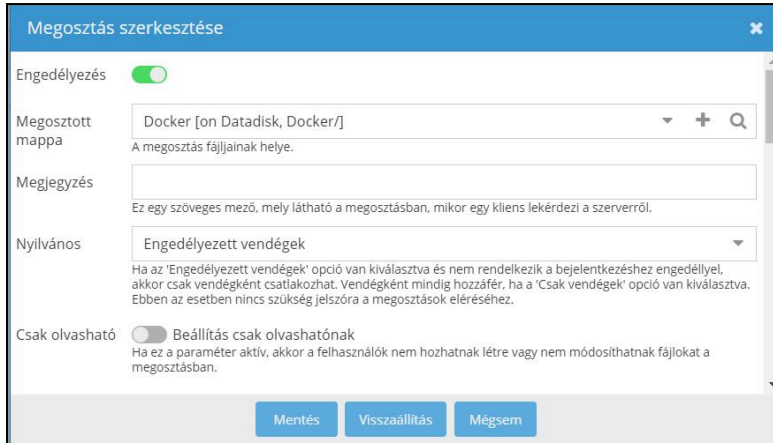
1. SMB/CIFS : Megosztás : Hozzáadás
2. Válasszuk ki a Docker mappát (vagy amit adtunk)
 - a. Nyilvános : Engedélyezett vendégeknek
 - b. Böngészhető : zöld

- c. ACL-ek öröklődés: zöld
- d. Jogosultságok öröklődése: zöld

3. Majd nyomjunk a **Mentés** gombra

Mindezt tegyük meg a Download /Torrent nevű mappánkkal melyet létrehoztunk.

Ehhez pár képet is mellékellek:



Ha minden jól csináltunk (nekem most többet fogtok látni) akkor a fájlkezelőnkbe beírva az alábbi látvány fogad.



A létrehozott Docker mappánkba hozzunk létre a szükséges almappáinkat, melyek az egyes konténereink adatait fogják tárolni.

Nálam ez így néz ki : NAS IP-je : 192.168.50.12 mint a képen is kiderült 😊

Hálózat > 192.168.50.12 > Docker

Név	Módosítás dátuma	Típus	Méret
Cloudflare	2020. 06. 06. 11:13	Fájlmappa	
Emby	2020. 06. 06. 11:07	Fájlmappa	
Grafana	2020. 06. 09. 19:41	Fájlmappa	
Letsencrypt	2020. 06. 06. 11:17	Fájlmappa	
MagicMirror	2020. 06. 06. 10:58	Fájlmappa	
Nextcloud	2020. 06. 06. 12:43	Fájlmappa	
Nextcloudodb	2020. 06. 06. 12:33	Fájlmappa	
Pihole	2020. 06. 06. 14:10	Fájlmappa	
Prometheus	2020. 06. 09. 19:12	Fájlmappa	
Transmission	2020. 06. 06. 11:02	Fájlmappa	

A Download vagy Torrent mappában szabadon létrehozhatunk pár almappát melyekbe letöltögetünk, például Filmek, Sorozatok, Képek, etc.

Fontos megjegyezni, ha kézzel hoztál létre mappákat, akkor a OMV-ben a megosztott mappáknál az ACL jogosultságokat nézd meg, hogy a Download/Torrent mappában nem mentek-e félre. Ha igen akkor javítsd ki őket.

Szemléltetve a Docker mappa tartalmát, nálam ez így néz ki:

Docker mappa:

- | - > Emby
- | - > Pihole
- | ----- > etc
- | ----- > dnsmasqsd
- | - > Letsencrypt
- | - > Nextcloud
- | - > Nextcloudodb
- | - > Transmission
- | ----- > watch
- | ----- > config
- | - >

Ezenfelül a Windows-os gépben a fájlkezelőben fel kell venni hálózati meghajtónak a Torrent mappát, mely majd a Transmisison-hoz lesz szükséges:

Sajátgép => menüsor => hálózati meghajtó csatlakoztatása => [\\NASIP\Torrent](#)

2.1. Docker felhasználó

A dockerben használt PUID és PGID azonosít egy felhasználót, ezért a legegyszerűbb módja, ha létrehozol egy új felhasználót az OMV-ben, mert „ennek nevében futtatjuk” majd a konténereket.

Én létrehoztam egy **stigma** néven egyet: OMV \ Hozzáférési jogok \ Felhasználók \ Hozzáadás

- A névhez beírod a kívánt felhasználó nevét
- Adj meg egy jelszót neki:
- Csoportoknál add hozzá az alábbi csoportokat: root, docker

Így néz ki képpel szemléltetve:

A docker konténerek futtatása esetében ezt a felhasználót (PUID: 1000 | PGID:100) fogom használni!

The screenshot shows the OMV (OpenMediaVault) web interface. On the left is a sidebar with system settings. The main area is titled 'Hozzáférési jogok kezelése' (Access Rights Management) and 'Felhasználó' (User). Below this, there's a 'Felhasználók' (Users) tab and a 'Beállítások' (Settings) tab. A table lists users, with 'stigma' highlighted. Below the table, a terminal window shows a login session as root, followed by the command 'id stigma' which outputs the user's PUID and PGID.

Név ↑	E-mail	Megjegyzés	Csoportok
stigma			users, root, docker

```
mc [root@omv5vm.omv5vm.local]:/srv
login as: root
root@192.168.50.96's password:
Linux omv5vm.omv5vm.local 5.4.0-0.bpo.4-amd64 #1 SMP Debian 5.4.19-1~bpo1
20-03-09) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Jun 14 16:10:47 2020 from 192.168.50.21
root@omv5vm:~# mc

root@omv5vm:~# id stigma
uid=1000(stigma) gid=100(users) groups=100(users),0(root),995(docker)
root@omv5vm:~#
```

A mappajogok nálam így néznek ki:

Hozzáférési jogok kezelése

Megosztott mappák

+ Hozzáadás Szerkesztés Jogosultságok ACL Törlés

Név ↑	Meghajtó	Relatív elér...	Abszolút elérési út	Megjegyzés	Használatb...
Docker	HDD1	Docker/	/srv/dev-disk-by-label-HDD1/Docker		Igen
Nextcloud	HDD1	Nextcloud/	/srv/dev-disk-by-label-HDD1/Nextcloud		Igen

Megosztott ACL mappa módosítása

Könyvtár

- Docker
- Duckdns
- Nextcloud
- Nextclouddb
- Nginx
- Owncloud

Felhasználó és Csoport jogosultságok

Típus	Név ↑	Olvasás...	Csak ol...	Nincs h...
Felhasználói fiókok	stigma	☒	☐	☐
Rendszerfiókok	adm	☐	☐	☐

További beállítások

Tulajdonos: stigma
Tulajdonos jogosultságai: Olvasás/Írás/Végrehajtás

Csoport: users
Csoport jogosultságai: Olvasás/Írás/Végrehajtás

Egyebek: Olvasás/Írás/Végrehajtás
Mások jogosultságai (pl. anonymous FTP felhasználók):

Csere: ☒ Minden meglévő jogosultság kicserélése

Rekurzívan: ☐ Jogosultságok alkalmazása fájlokra és almappákra

Alkalmaz Bezárás

Itt a Docker mappában tárolt minden almappának a stigma felhasználó lesz a tulajdonosa, mivel ezen a felhasználó név alatt „futtatom” a konténereket Portainerben.

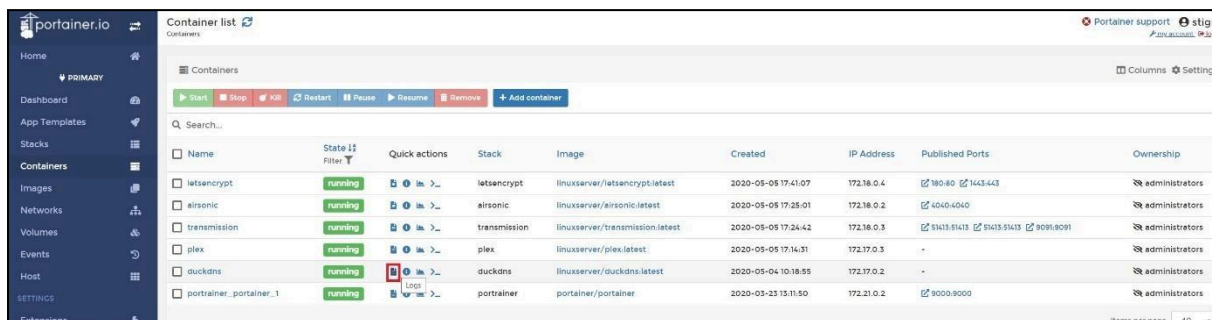
TIPP: Ha gyorsan akarod, hogy a jogosultságok a Docker almappában az adott felhasználóhoz legyenek rendelve, akkor állj a Docker mappára, állítsd be mindent a képen látható módon, természetesen cseréld ki a te felhasználódra a stigma userem, majd alul a Rekurzívan csúszka legyen zöld és Alkalmaz.

3. Konténerek

A konténerek futtatásához fel kell tennünk az alábbi elemeket:

- Rendszer \ Plugins => OMV Extrák : Engedélyezni kell az openmediavault-extra elemet
- utána az OMV extrákon belül a Docker és a Portainer telepíthetővé válik.

A Portainer-ben feltelepített konténereinket tudjuk menedzselni/telepíteni/újratelepíteni illetve a legfontosabb logolni. Ez azért fontos, mert ha valahol hiba van akkor a konténer nevére kattintva a logban könnyen ki lehet igazodni. lásd pl: az alábbi képen.



Name	State	Quick actions	Stack	Image	Created	IP Address	Published Ports	Ownership
letsencrypt	running	[Icons]	letsencrypt	linuxserver/letsencrypt:latest	2020-05-05 17:41:07	172.18.0.4	80,443	administrators
airsonic	running	[Icons]	airsonic	linuxserver/airsonic:latest	2020-05-05 17:25:01	172.18.0.2	4040-4040	administrators
transmission	running	[Icons]	transmission	linuxserver/transmission:latest	2020-05-05 17:24:42	172.18.0.3	9091-9091	administrators
plex	running	[Icons]	plex	linuxserver/plex:latest	2020-05-05 17:14:31	172.17.0.3	-	administrators
duckdns	running	[Icons]	duckdns	linuxserver/duckdns:latest	2020-05-04 10:18:55	172.17.0.2	-	administrators
portainer_portainer_1	running	[Icons]	portainer	portainer/portainer	2020-05-23 13:11:50	172.21.0.2	9000-9000	administrators

Ami még fontos, hogy minden konténert PUID, PGID esetében futtatok. Ott ahol nem adatok meg felhasználót ott **root**-ként fut a konténer.

Amikor egy konténer **Volume** elérését definiáljátok, mindenképp kell írási jognak lenni az adott mappához a felhasználónak (PUID és PGID): (OMV-ben az ACL-nél ellenőrizheted)

PUID és PGID: OMV-ben létrehozol egy új felhasználót, ami mondjuk legyen dockeruser.

Ebben az esetben SSH-n az alábbi parancsot kell kiadnod, hogy megtudd mi a PUID és PGID azonosító

id dockeruser

Nálam itt egy stigma felhasználó lett aminek PUID 1000 és PGID 100 lett az azonosítója

```
root@omv5vm:~# id stigma
uid=1000(stigma) gid=100(users) groups=100(users),0(root),995(docker)
root@omv5vm:~#
```

A konténereket az alábbi módon tudod futtatni:

Portainer => Stack => Add stack => Web editorba bemásolni, nevet adni neki, majd a **Deploy to stack** gombra nyomni.

A konténerekre itt tudsz rákeresni: <https://hub.docker.com/>

Ami még fontos lehet az a konténerek esetén a megfelelő image letöltése:

pl: qbittorrent esetén az alábbi látható

image: ghcr.io/linuxserver/qbittorrent

Itt a dockerhubos résznél amikor elfuttatod, elég a „**image: linuxserver/qbittorrent**” rész, mert a másik image (ghcr.io) ARM lapokhoz készült.

3.1. Konténerek telepítése

A konténerek telepítésének két egyszerű módja van:

1. Portainer, melyet felteszel OMV-re az alábbi címen éred el: <http://nasip:9000/>

Portainerben már 3.-al kezdődő docker-compose fájlok is futtathatóak!

Konténerek telepítésének menete:

1. Portainer => Stack => Add stack => Web editorba bemásolni a tartalmat + nevet adni neki

Cserélni kell:

Volumes: elérési utakat (Figyelni arra hogy a mappák és almappák létezzenek)
PUID vagy PGID-t, ha neked más lenne

2. **Deploy to stack** gombra nyomni.

2. Docker compose fájlból

Fájlos futtatásból bármilyen verzió futtatható, melyet az alábbi módon tudsz megtenni.

1. Docker mappádon belül létre kell hozni egy docker-compose.yml fájlt
Ha ssh-n keresztül csinálod, akkor a fájlt így tudod létrehozni:
nano docker-compose.yml
2. Az általam megadott txt tartalmát bemásolod, majd a szükséges adatokat megváltoztatod:
 - a. PUID vagy PGID
 - b. Volume-k elérési útjai
 - c. felhasználónév vagy jelszó
 - d. TZ (timezone)
 - e. Esetleges egyéb token azonosítók
3. A mappában, ahol létrehoztad a fájlt, kiadod az alábbi parancsot:
docker-compose up -d

Ehhez készítettem egy képet nektek, mely bemutatja egy jól letöltött és telepített konténert. Akkor csináltatok jól, mikor alul zölddel **done** felirat fogad

Itt a **Pulling from** konténerneveket letöltötte és telepített a docker, majd a végén ahol a done feliratot látjátok létrehozta a 2 konténert, mely:

- owncloud_mysql_1 : adatbázis
- owncloud_ owncloud_1: maga az alkalmazás

Szemléltetésképp itt láthatjátok, hogy az alábbi paranccsal (docker.compose up -d) ugyanúgy tudsz futtatni konténert, mint amit Portainer /Stack megadás majd Deploy után

```
root@omv5vm:/srv/dev-disk-by-label-HDD1/Docker/Owncloud# docker-compose up -d
Creating network "owncloud_default" with the default driver
Pulling owncloud (owncloud:)...
latest: Pulling from library/owncloud
177e7ef0df69: Pull complete
9bf89f2eda24: Pull complete
350207dcf1b7: Pull complete
a8a33d96b4e7: Pull complete
c0421d5b63d6: Pull complete
f76e300fbe72: Pull complete
af9ff1b9ce5b: Pull complete
d9f072d61771: Pull complete
a6c512d0c2db: Pull complete
5a99458af5f8: Pull complete
8f2842d661a0: Pull complete
3c71c5361f06: Pull complete
baeacbad0a0c: Pull complete
e60049bf081a: Pull complete
0619078e32d3: Pull complete
a8e482ee2313: Pull complete
174d1b06857d: Pull complete
4a86c437f077: Pull complete
5e9ed4c3df2d: Pull complete
8a1479477c8e: Pull complete
8ab262044e9e: Pull complete
Digest: sha256:173811cb4c40505401595a45c39a802b89fb476885b3f6e8fe327aae08d20fe8
Status: Downloaded newer image for owncloud:latest
Pulling mysql (mariadb:)...
latest: Pulling from library/mariadb
da7391352a9b: Pull complete
14428a6d4bcd: Pull complete
2c2d948710f2: Pull complete
22776aa82430: Pull complete
90e64230d63d: Pull complete
f30861f14a10: Pull complete
e8e9e6a3da24: Pull complete
420a23f08c41: Pull complete
bd73f23de482: Pull complete
a8690a3260b7: Pull complete
4202ba90333a: Pull complete
a33f860b4aa6: Pull complete
Digest: sha256:cdc553f0515a8d41264f0855120874e86761f7c69407b5cfbe49283dc195bea8
Status: Downloaded newer image for mariadb:latest
Creating owncloud_mysql_1 ... done
Creating owncloud_owncloud_1 ... done
root@omv5vm:/srv/dev-disk-by-label-HDD1/Docker/Owncloud#
```

FONTOS: Minden általam ajánlott konténer esetén találtok egy txt fájlt, mely az én Portainer stack-em(docker-compose) adatait tartalmazza. Ezt ha bemásoljátok a Portainer / Stack menüpontban egy új Stack elembe, akkor gyorsabban tudtok futtatni.

FONTOS: Stack-ben a web editorban SOHA NE HASZNÁLJ TAB-ot, mert az Editor nem szereti.

A txt fájlban általánosságban az alábbi adatokat kell cserélni:

- PUID és PGID: a felhasználó, amely nevében futtatod
- volumes adatai : /config/data, stb
- Portok : 2 alkalmazás ugyanazt a portot nem tudja használni, ilyenkor át kell irányítani másira
- jelszó vagy felhasználónevek, ha a stackben meg vannak adva.

3.2. Konténerek frissítése

A legtöbb konténer egyszerűen frissíthető Portainer alatt.

Lépésről lépésre ez így néz ki:

- nyisd meg a Portainer-t : <http://nasip:9000/>
- **Contrainert** menüpontban kattints az alkalmazásra
- Felül a **Recreate** menüpontra kattints, majd a csúszka a **Pull latest image** elemen legyen bekapcsolva. (Pull latest image = utolsó image letöltése)

Képet beszúrni

Ha viszont Portainer-t akartok frissíteni, annak más módja van:

Az alábbi oldalra menjetek fel, nézzétek meg van-e frissebb verzió:

<https://docs.portainer.io/v/ce-2.11/start/upgrade>

Jelenleg a CE 2.11-es van. CE => Community Edition

SSH alól sorban ezt a 3 parancsot kell megadni

1. Portainer megállítása : `docker stop portainer`
2. Portainer törlése: `docker rm portainer`
3. Új image letöltése: `docker pull portainer/portainer-ce:2.11.0`
Itt ha bal felül az utolsó verziót jelölitek ki, akkor mindig a legfrissebb verziót ajánlja fel
4. Új verzió telepítése:
`docker run -d -p 8000:8000 -p 9000:9000 -p 9443:9443 \`
`--name=portainer --restart=always \`
`-v /var/run/docker.sock:/var/run/docker.sock \`
`-v portainer_data:/data \`
`portainer/portainer-ce:2.11.0`

FONTOS: ha a **/data** változókat nem a rendszer meghajtóra mentitek, hanem máshová, akkor a kiemelt részt cserélnetek kell.

pl.:

```
docker run -d -p 8000:8000 -p 9000:9000 -p 9443:9443 \
--name=portainer --restart=always \
-v /var/run/docker.sock:/var/run/docker.sock \
-v /srv/dev-disk-by-label-WD6TB/Docker/Portainer:/data \
portainer/portainer-ce:2.11.0
```

-

3.3. Torrent kliensek

Az alábbi 2 torrent kliens az, amelyet ajánlok és a fórumban is sokan használnak.

Ha valakinek 200 mbit/s-nél jobb letöltési sebességgel rendelkező internete van, annak inkább ajánlott a Qbittorrent. A Transmission csak egy szálát használ, ezért megakaszthatja a NAS-t, a Qbittorrent viszont támogatja a többszálás letöltést.

A Qbittorrent beállításokhoz a prohardver megfelelő fórumában kérdeztek: [Link](#)

Az alábbi két torrenthez van Windows-os kliens, melyben könnyebben tallózhatjuk be a torrenteket:

Transmission Remote GUI: [Github](#)

Qbittorrent kliens: [Electorrent](#)

A telepítés menete: [3.1 rész](#)

3.3.1. Transmisison

Dockerhub: [Link](#)

(A beágyazott txt fájlra kettőt kattintva előjön az általam használt Stack Portainerben!)



transmission.txt

Az alábbi IP címen tudtok belépni, ha nem változtattatok port-ot:

<http://NASIP:9091>

Felhasználónév és jelszót amit megadtatok.

Windows Kliens konfigurálása:

Windows-os gépünkre az alábbi programmal tudunk a NAS-ra torrentezni:

Transmission Remote Gui: [Link](#)

Tegyétek fel, majd **Torrent / Kapcsolódás a Transmission-höz / új kapcsolat:**

Transmission fül:

Távoli kiszolgáló: NASIP

port: 9091 vagy ha változtattátok akkor azt írjátok oda

Azonosítás szükséges: pipa

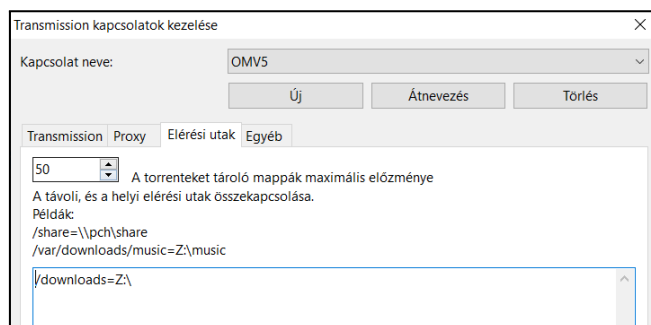
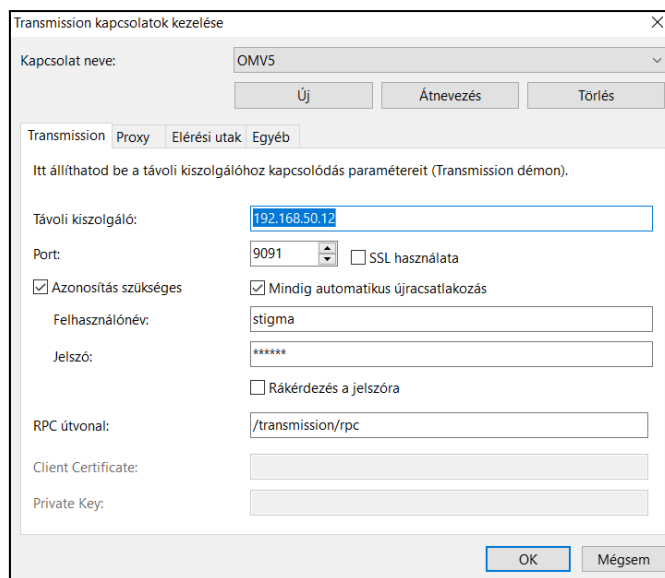
felhasználónév és jelszó: amit a konténerben megadtatok

Elérési utak fül: ide ezt írjátok

`/downloads=Z:\`

Itt a Windowsban, ha a Torrent mappát Y vagy bármilyen betűvel vettétek fel hálózati meghajtónak, akkor arra cseréljétek ki a kiemelt részt.

Képekkel szemléltetve:



Fontos: Ha új torrentet szeretnétek letölteni, akkor mindenképp azt a változót adjátok meg amit megadtatok a volumes részben a letöltésre.

Példának ez lenne:

- /srv/dev-disk-by-label-WD6TB/Torrent:/**downloads**

Tehát ha új torrentet adtok hozzá, akkor letöltéskor a /downloads megnevezést használjátok. Ha ezen belül van almappátok (pl.:Filmek,Sorozatok) akkor azt is megadhatjátok => /downloads /Filmek

A kis és nagybetűkre figyeljete, amikor megadjátok a letöltési mappát!!!

3.3.2. qbittorrent

A telepítés menete: [3.1 rész](#)

Dockerhub: [Link](#)

(A beágyazott txt fájlra kattintva előjön az általam használt Stack Portainerben!)



qbittorrent.txt

A gyári 8080-as portot lecseréltem 8081-re. Ilyenkor a WEBUI_PORT változót is át kell írni, ha portot cserélsz!

Ezek után az alábbi porton érhető el a qbittorrent kliens: <http://NASIP:8081>

Ha nem definiáltatok username és password elemeket, vagy épp kitöröltétek a stack-ből, akkor az alapértelmezett jelszó az alábbi:

felhasználónév: admin

jelszó: adminadmin

Ezt mindenféleképp belépés után változtasd meg, vagy ha netre engeded ki, akkor is!!!

Qbittorrent prohardver Topic: [Link](#)

Pontos beállítást az alábbi fórumban kérdezhetsz. Ez azért fontos, mert egy jobb net esetén eléggé leterheli a NAS-t. Nálam J4105-ITX + 8GB ram esetén ez így néz ki:

Ami fontos hogy csak TCP porton tölts le, illetve a routeren belül az alábbi portot engedélyezd.

3.4. Média Server

Média szerver telepítésre több is ajánlott verzió is van, melyet telepíthetsz. Plex, Emby , Jellyfin. Igazából alap felhasználási célra bármelyik megfelel. Jellyfin Emby fork, kinézetre és funkciókra 99%-ban megegyezik. A Jellyfin teljesen ingyenes, a Plex és Emby szervernek vannak fizetős részei.

A legtöbb TV-n van előre telepített Plex alkalmazás, de lehet Emby appot is rátelepíteni.

Én külön lejátszót használok (Khadax Vim3L), mely Amlogic box. Ezen Coreelec rendszer fut, amely Kodit használ.

Andoird/Coreelec boxon, ha Kodi-t használtok, akkor az alábbi 2 pluginnal tudtok egy Plex és Emby szerverhez csatlakozni:

Plex => PlexKodiConnect (PKC) : [Link](#)

Emby => EmbyCon : [Link](#)

Jelenleg én ezzel a 2 pluginnal a Emby/Plex szerverhez csatlakozva **csak az adatbázist használok**. Van 2 másik plugin is , mellyel az adatbázis + a GUI is átvárázsolható a Kodiba, és lehet belőle Plex vagy Emby felületet csinálni.

3.4.1. Emby

Ez egy filmes adatbázis, mely a letöltött filmek, sorozatok, zenékre lett kitalálva.

Ha ezt felteszitek és van több eszközötök akár TV-n, tableten, Amlogic boxon, akkor elértek ugyanazokat a tartalmakat és tudjátok szinkronizálni, mikor mit néztetek meg.

A hivatalos oldalon több /mnt/share1, mnt/share2 változót is definiálhatsz, de nálam csak 1 van, mert egy mappán belül vannak nálam a Sorozatok/Filmek egyéb elérés

A telepítés menete: [3.1 rész](#)

Dockerhub: [Link](#)



emby.txt

(A beágyazott txt fájlra kattintva előjön az általam használt Stack Portainerben!)

Ha a szerver fut, az alábbi címen éred el: <http://NASIP:8096>

Az ip-t ha beírod, és konfigurárod az emby-t, akkor itt már az adott filmek elérési útja az általad megadott volume lesz => /mnt/share1

Itt egy dbtech videó a telepítésről és a konfigurációról: [Link](#)

Én a hivatalos stackből a `/transcoding` sort kihagytam, mert nincs szükségem transzkódolásra.

Ezenfelül, amikor a videóban szereplő könyvtárakat kiválasztod, ne a videóban szereplő `/data/movies` és `/data/kids` mappákat keresd, hanem az általam megadott `/mnt/share1`-t

3.4.2. Plex server

Dockerhub: [Link \(cserélni\)](#)



plex.txt

(A beágyazott txt fájlra kettőt kattintva előjön az általam használt Stack Portainerben!)

Ez egy filmes adatbázis, mely letöltött filmekre, sorozatokra, zenékre lett kitalálva.

Ha a szerver fut az alábbi címen éred el: <http://NASIP:32400>

Én a `/movies` változóhoz a gyökér mappát társítottam, mert a Filmeket és Sorozatokat ott érem el. A mappa struktúráám az alábbi:

Torrent

| - >Filmek

| - >Soroaztok

 | - >Sorozat1

 | - >Sorozat2

| - >ETC

3.5. Felhő szolgáltatás

Cloud használatára több lehetőség is van. Én jónéhányat kipróbáltam, viszont megmaradtam a jó öreg nextcloud-nál, mert amire nekem kell, arra tökéletes:

- **Nextcloud:** Bemutatni nem kell 😊
- **Owncloud:** a 10.x-es verzió után már csak egy 30 napos trialt enged feltenni
- **Seafile:** Ez egy tökéletes cloud, ha a szinkronizált eszközöd adatait szeretnéd csak elérni
- **Pydio:** Ez egy kevésbé ismert cloud konténer. Ez inkább egyfajta Team Workspace-nek tudom elképzelni, mint a G-Suite.

3.5.1. Nextcloud

A telepítés menete: [3.1 rész](#)

Jelenleg én a hivatalos oldali Nextcloud-ot használom, pontosabban a **Base version – apache**

Ezenkívül használható a linuxserver-es változat is, de ahhoz külön fel kell tenni a linuxserver\mariadb-t és a linuxserver\nextcloud konténereket

Dockerhub: [Link](#)



nextcloud_dbtech.txt

(A beágyazott txt fájlra kattintva előjön az általam használt Stack Portainerben!)

Ha a Nextcloud-ot netre akarod kiengedni, akkor az alábbi lépések szükségesek:

Amiket cserélned kell:

- Volumes-ok esetén az elérési utakat
- Jelszavakat amiket megadtam
- Portot: én 8082:80-at adtam meg, mivel már volt más konténerem ami használta a 8080-as portot

Ha nem akarod a netre kiengedni, akkor ebben az esetben a <http://NASIP:8082-es> porton éred el a Nextcloud-ot, és be is tudsz lépni.

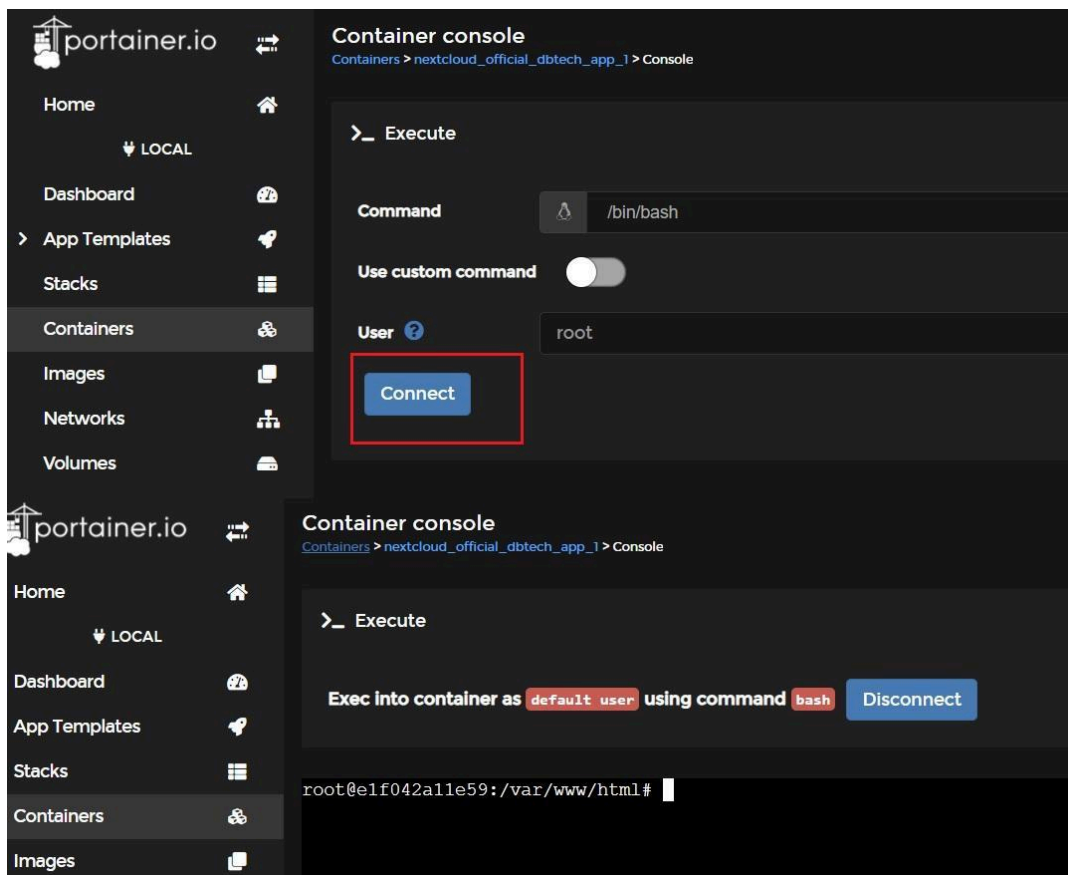
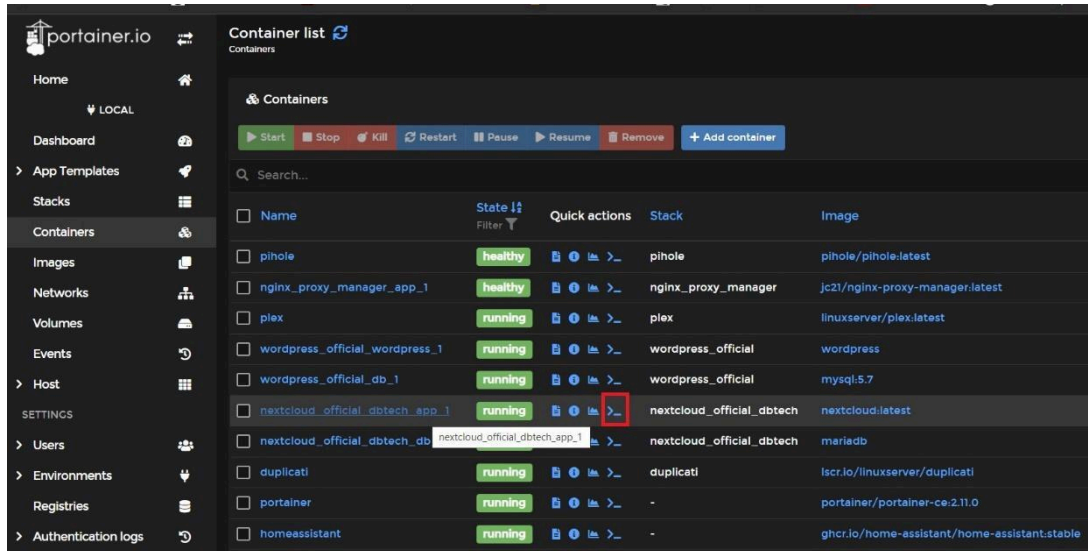
Ha ki akarod engedni a netre, akkor a konténer futtatása után, Nginx Proxy Managerrel vagy swag-al a nextcloud domain-ed [készíts egy tanúsítványt](#), és utána a netes verzióban próbálj meg belépni.

A linkelt [youtube](#) videóban az smbclient csomagot még fel kell tenned, ezáltal tudsz majd SMB /Helyi mappákat csatolni.

Helyi mappa csatolására mutatok egy példát:

Ehhez installálni kell az smbclient csomagot, amit az alábbi módon tudtok megtenni:

1. Lépj be Portainerbe, majd válassz ki a Nextcloud App-od console-ját, amit pirossal kijelöltem neked: Itt a connect-re kell lemenni, majd a 2. kép végénél látnod a fehér téglalapot, oda kell írnod az alábbi 2 sort

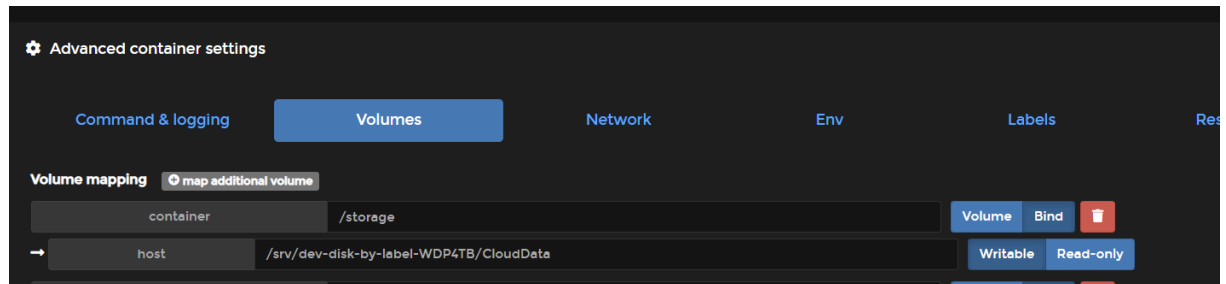


első parancs: **apt update** majd enter

második parancs: **apt install smbclient** majd enter

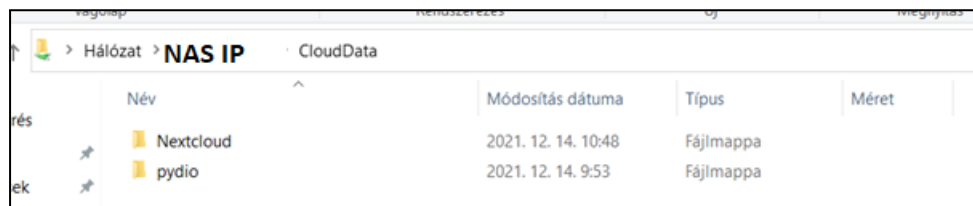
Ha ezzel megvagy, akkor nyomj egy Disconnect-et, majd indítsd újra a Nextcloud app-ot.
Ezek után így tudsz felcsatolni egy mappát:

1. lépés az OMV-ben létrehozol egy mappát, amit a NExtcloud-ba fel szeretnél csatolni.
Ugyanúgy mint docker esetében, Mindenki ír/olvas és guest felhasználók is elérjék, majd megosztod a hálózaton
2. Stack/docker-compose módosítása:
A Portainerben a Nextcloud appra még (nextcloud_official_dbtech_app_1) majd szerkeszted és hozzáadod a felcsatolni kívánt mappádat:



A képen a NASIP-t beírva neked így néz ki a mappaszerkezet, és mint láthatod a CloudData mappában több almappa is van. Ezt azért csináltam, mert több cloud konténert is teszteltem pl.: Pwncloud, Pydio, Seafile, stb.

A /storage megnevezés szabadon módosítható bármire, lehet akár /mountdata is, rád bízom.



Vagy a stack-be ha abba szerkesztenéd, akkor az alábbi sort kell hozzá adnod: Itt ugye a te elérési utadra cseréld ki:

- /srv/dev-disk-by-label-WDP4TB/CloudData:/storage

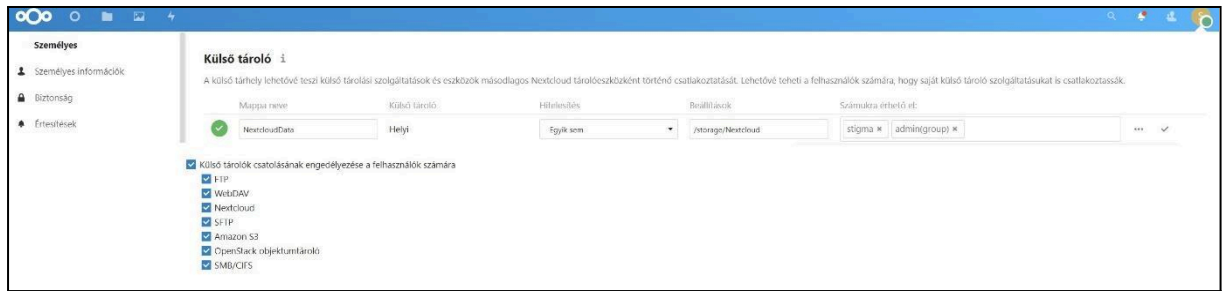
2.lépés :

Belépsz Nextcloud-ba és 1-2 apróságot még engedélyezni kell, mielőtt hozzáadjuk a mappát:

Bal felül a profilod / Alkalmazások => itt engedélyezed az alábbi: **External storage support**

Ha ezzel megvagy, akkor már csak be kell állítani a Nextcloud felületén az alábbiakat:

Bal felül a profilod / Beállítások / Adminisztrátori rész / Külső tároló



Egy kis magyarázat ehhez:

Mappa neve: Ide bármilyen nevet megadhatsz, ilyen néven fog megjelenni neked a Nextcloud felületén a fájljoknál

Külső tároló: Itt a listában nagyon sok verzió közül választhatsz, akár smb/webdav, FTP stb. Itt a helyi jelenti azt, hogy „magán a HDD-n van, ami a gépedben van.”

Hitelesítés: Mivel mindenki által elérhető és nem jelszavas, ezért ide nem kell semmit sem kitölteni. Ha SMB-s hálózati mappát vennél fel, akkor ezeket az elemeket ki kell tölteni.

Beállítások: Ide ugye a fő mappát vettük fel a docker-compose fájlba/stackbe, és ugye itt még nekem van több almappám is, ezért kell ide a /storage/Nextcloud-ot írni.

Számukra érhető el: Itt kiválaszthatod azokat a csoportokat, akik elérhetik.

Miért jobb ez a megoldás mintha csak simán a Nextcloudba húztad volna bele?

- Egyrészt ha a Nextcloud konténered tönkremegy, adataid nem vesznek el (igaz itt még nem 100% hogy elvesznek, mert újratelepítés után működhetnek)
- helyi hálózatos gépekről (laptop, asztali gép) 1 mozdulattal bele tudsz tenni új fájlokat szenvedés nélkül.

3.5.2. Seafile

Telepítés menete: [3.1 rész](#)



seafile.txt

(A beágyazott txt fájlra kettőt kattintva előjön az általam használt Stack Portainerben!)

FONTOS: Én már telepítés előtt az általam létrehozott domaint és tanúsítványt Nginx Proxy Managerrel létrehoztam és utána futtattam csak a konténert.

Amiket cserélned kell:

- Volumes-ok esetén az elérési utakat
- Jelszavakat amiket megadtam
- Portot: én 8084:80-at adtam meg, mivel már volt más konténerem, ami használta a 8080-as portot

Ha otthoni hálózaton akarsz elérni, akkor akkor itt tudod elérni: <http://NASIP:80804>

3.6. Reklámblokkoló:

Két ismertebb reklámblokkoló konténer/app van ami elterjedt mostanság. A PiHole és az Adguard. Funkcióját tekintve mindkettő egy alap felhasználónak tökéletes.

Összehasonlítás : [Link](#)

3.6.1. Adblocker - félkész

Az alábbi konténert mindenképp külön macvlan-ra kell feltenni. A Macvlan-ról a legtöbb infót a google segítségével megtaláljátok, nem térek rá ki külön.

A legjobban az alábbi két videón magyarázza el dbtech srác:

Adguard: [Link](#)

Pi-hole: [Link](#)

Amire még fontos kitérni az az, hogy ezt a routerben a DNS1 szerver helyre kell majd beírni.

A konfiguráláskor említett 192.16.50.0 vagy 192.168.50.1-es ip-t ha említekt az a saját routeretek címére kell cserélni.

A „félkész” PI-hole és Adguard esetén a macvlan-t kézzel hozd létre, majd utána a konténereket már csak futtatnod kell.

macvlan létrehozása SSH-n keresztül:

```
docker network create -d macvlan \
```

```
--subnet=192.168.50.0/24 \
```

```
--gateway=192.168.50.1 \
```

```
-o parent=en0 pihole_lan
```

Vagy ha 1 sorba kifer akkor ezt kell írnod:

```
sudo docker network create -d macvlan -o parent=enp3s0 --subnet=192.168.50.0/24
```

```
--gateway=192.168.50.1 pihole_lan
```

Pár dologra kitérek, mit is jelentenek:

parent=enp3s0 : az OMV-ben csekkold le mi a hálókártyát neve, azt írd ide.

Ez vagy eth0, vagy az amit írtam általában.

subnet=192.168.50.0/24 : Ide írd a routered első 3 maszkját, tehát ha a Tp-link ami 192.168.1.0 akkor 192.168.1.0/24

gateway=192.168.50.1: Ide a routered címét helyettesítsd be. Tp-link esetén 192.168.1.0

utolsó string/megnevezés(pihole_lan): az a Macvlan hálózatod neve lesz, bármit megadhatsz:

Ezek után már csak a szükséges konténereket kell futtatnunk, viszont vagy a stackben vagy manuálisan Portainer-ben kell majd átállítanunk, hogy használja a pihole_lan hálózatot.

3.6.2. Pihole - félkész

Dockerhub: [Link](#)

0. lépésként az alábbi fontos dolgokat el kell végezni:

OMV => hálózat => interface => Szerkesztés => DNS szerver => 1.1.1.1 vagy 8.8.8.8

Az interface nevét jegyezzétek meg, mert szükség lesz a MACVLAN telepítéskor!!
(eth0 vagy enp3s0 néven fog nagy valószínűséggel futni)

OMV esetén az alábbi fájlt át kell írnod => resolved.conf:

0. lépés: Beállítani az interface-eket és a DNS-t

1.lépés: systemctl stop systemd-resolved

2.lépés: nano /etc/systemd/resolved.conf

Itt az alábbi 3 sort kell hozzáadni:

[Resolve]

DNS=1.1.1.1

FallbackDNS=1.0.0.1

DNSStubListener=no

3. lépés: mv /etc/resolv.conf /etc/resolv.conf.backup

4. lépés: ln -s /run/systemd/resolve/resolv.conf /etc/resolv.conf

5. lépés: systemctl reload-or-restart systemd-resolved

1.lépés: MACVLAN létrehozása : [Link](#)

Kis magyarázat:

parent : enp3s0 vagy eth0 (Ezt az omv/hálózatnál csekkold le)

subnet : a routered elérésének almaszkja, de az utolsó karakternek 0-nak kell lennie

pl:192.168.50.0/24 | 192.168.1.0 | 192.168.0.

gateway: a routed címe kell: pl: 192.168.50.1 vagy 192.168.1.1 vagy 192.168.0.1

Create network
Networks > Add network

Name: MacConfig

Driver configuration

Driver: macvlan

Driver options: add driver option

Macvlan configuration

To create a MACVLAN network you need to create a configuration, then create the network from this configuration.

Configuration (selected): I want to configure a network before deploying it

Creation: I want to create a network from a configuration

Parent network card: enp3s0

IPv4 Network configuration

Subnet: 192.168.50.0/24

Gateway: 192.168.50.1

IP range: 192.168.50.80/24

add excluded IP

új macvlan létrehozás:

MyMcvlan a neve

Driver: mcvlan

Creation lett a hálózat

Configuration: MacConfig (ezt hoztam létre ezelőtt)

enable manual container attachment : ON

Name: MyMcvlan

Driver configuration

Driver: macvlan

Driver options: add driver option

Macvlan configuration

To create a MACVLAN network you need to create a configuration, then create the network from this configuration.

Configuration (selected): I want to configure a network before deploying it

Creation: I want to create a network from a configuration

Configuration: MacConfig

Advanced configuration

Labels: add label

Restrict external access to the network: ☐

Enable manual container attachment: ☒

Access control

Enable access control: ☒

Administrators (selected): I want to restrict the management of this resource to administrators only

Restricted: I want to restrict the management of this resource to a set of users and/or teams

Pihole -nál Consolra lépni, majd beírni az alábbi parancsokat:

```
apt update
```

```
apt install net-tools
```

Ezek után kilépett a Portainer-ből majd beléptem újra

pihole => network-nél váltottam MyMcvlan névre és az ipv4-hez beírtam hogy 192.168.50.99

Pihole => console => connect

ifconfig parancs => ha hibát dob akkor

```
sudo apt update
```

```
apt install net-tools
```

utána már fog működni és kiírja az ip-t amit adtam neki 192.168.50.99

2. lépés: konténer telepítése



pihole.txt

Telepítés menete: [3.1 rész](#)

A „#” jelölt portokat kikommenteltem, mert nekem nincs rá szükségem.

3. lépés: Pihole beállítása:

3.6.3. AduardHome – félkész a telepítés

Dockerhub: [Link](#)

0. lépésként az alábbi fontos dolgokat el kell végezni:

OMV => hálózat => interface => Szerkesztés => DNS szerver => 1.1.1.1 vagy 8.8.8.8

Interface nevét jegyezzétek meg, mert szükség lesz a MACVLAN telepítéskor!!
(eth0 vagy enp3s0 néven fog nagy valószínűséggel futni)

OMV esetén az alábbi fájlt át kell írnod => resolved.conf:

0. lépés: Beállítani az interface-eket és a DNS-t

1.lépés: systemctl stop systemd-resolved

2.lépés: nano /etc/systemd/resolved.conf

Itt az alábbi 3 sort kell hozzáadni:

[Resolve]

DNS=1.1.1.1

FallbackDNS=1.0.0.1

DNSStubListener=no

3. lépés: mv /etc/resolv.conf /etc/resolv.conf.backup

4. lépés: ln -s /run/systemd/resolve/resolv.conf /etc/resolv.conf

5. lépés: systemctl reload-or-restart systemd-resolved

1.lépés: MACVLAN létrehozása : [Link](#)

Kis magyarázat:

parent : enp3s0 vagy eth0 (Ezt az omv/hálózatnál csekkold le)

subnet : a routered elérésének almaszkja, de az utolsó karakternek 0-nak kell lennie

pl:192.168.50.0/24 | 192.168.1.0 | 192.168.0.

gateway: a routered címe kell: pl: 192.168.50.1 vagy 192.168.1.1 vagy 192.168.0.1

ip-range -nek megadott IP címen fogod elérni az adott konténert.

pihole_net néven fog létrejönni az új hálózat, ide más nevet is megadhatasz

Ezt futtattam én, routerem elérése 192.168.50.1

```
docker network create -d macvlan \  
  --subnet=192.168.50.0/24 \  
  --gateway=192.168.50.1 \  
  --ip-range=192.168.50.99 \  
  -o parent=enp3s0 adguard_net
```

Erre pár példa ha más az ip-d hogy csináld Asus/Tp-Link, Cisco, egyéb routerekkel:

Ha ASUS routered van, aminek 192.168.1.0 az elérése

```
docker network create -d macvlan \  
  --subnet=192.168.1.0/24 \  
  --gateway=192.168.1.0 \  
  --ip-range=192.168.1.99 \  
  -o parent=enp3s0 adguard_net
```

Ha Tp-Link routered van, aminek 192.168.1.1 az elérése

```
docker network create -d macvlan \  
  --subnet=192.168.1.0/24 \  
  --gateway=192.168.1.1 \  
  --ip-range=192.168.1.99 \  
  -o parent=enp3s0 adguard_net
```

Ha egyéb (Cisco, Dlink, stb) routered van, aminek 192.168.2.1 az elérése

```
docker network create -d macvlan \  
  --subnet=192.168.2.0/24 \  
  --gateway=192.168.2.1 \  
  --ip-range=192.168.2.99 \  
  -o parent=enp3s0 adguard_net
```

2. lépés: Konténer telepítése és konfigurálása:

Telepítés menete: [3.1 rész](#)



adguardhome.txt

Ezek után ha elfuttattuk, akkor egy apróbb változást kell még beletennünk

Portainer-ben a Konténerre kattintunk, majd az alábbi 2 műveletet kell beírni:

- Network résznél kiválasztjuk az adguard_net macvlan-t amit létrehoztunk
- ??IPv4-re megadunk azt az ip-t, amelyet az ip-rangnek megadtunk

Ezek után az alábbi ip címen érjük el a konténerünket: `http://nasip:3000`

3. router konfiguráció:

3.7. Photo Management

Itt több képek tárolására szolgáló konténert is teszteltem, melyek az alábbiak.

- Photoprism: [Link](#)
- Lychee: [Link](#)
- Piwigo [Link](#)
- chevereto [Link](#)

Engem személy szerint a Photoprism győzött meg. Itt Portainer-es futtatást helyett docker-compose fájlból csináltam a telepítést.

Mindegyik alap funkciókra tökéletes : fotótárolás, menedzselés, megosztás, szinkronizálás

3.7.1. PhotoPrism

A szükséges docker-compose fájlt a [Github](#)-ról szedtem le.



PhotoPrism_docker-compose.yml

Itt inkább leírom, miket kell átírnotok a linkelt txt fájlban.

Lépésről lépésre ezeket csináltam:

1. Hozz létre egy mappát a PhotoPrism-nek. Ide kell majd egy docker-compose.yml fájlt létrehozunk
2. SSH-n lépj be a NAS felületére, majd add ki az alábbi parancsot: (ezzel hozod létre a fájlt)
nano docker-compose.yml
3. A txt-t nyisd meg amit mellékeltem, majd másold ki a docker-compse fájl tartalmát és SSH-ra visszakattintva másold be (elég ha a fekete részre bele állsz és jobb klikket nyomsz)
4. Ezek után az általam írt pár dolgot módosítsd a tartalmában, majd CTRL+X és Enter (ezzel elmentetted):
Módosítsd az „Ideerősjelszócell” megnevezéseket. Amil aposztrófban van, oda a jelszó átírás után is szükséges az aposztróf, ahol nincs ilyesmi, ott csak cseréld ki az általad megadott jelszóra.
5. Ezek után már csak a futtatáshoz szükséges parancsot kell kiadni
docker-compose up -d

Ezek után az alábbi felületen lehet elérni az alkalmazást: <http://NASIP:2342>

A docker-compose fájlban megadott belépési adatokkal tudsz belépni.

3.8. Konténerek netes elérése

Ezt többféleképpen meg tudjuk tenni, ehhez az alábbi konténerek valamelyiket használjuk

Fontos: Az alábbi 2 konténerhez megadott portokat a routerben engedélyezni kell.

1. Portok:

Általában a http(80) és HTTPS(443) portok az alapértelmezettek, nem kötelező „babrálni”.

swag esetén a port átirányítás pl.:90:80 és 450:443 tökéletesen ment, viszont az Nginx Proxy Manager esetén ez nem működött nekem, csak az alap 80:80 és 443:443-as portokkal.

IP-nek a routerben azt az eszközt kell megadni, amin fut, tehát a NAS IP címét

Ezenkívül az OMV webes portját, ami alapértelmezetten 80-as meg kell változtatni!!!

2. Domain cím:

A konténerek netes eléréséhez kell egy domain cím, mely lehet a routerhez járó

Asus/Tp-Link/egyéb domain cím is, illetve lehet egy Duckdns ingyenes cím is. Ha nincs domain-ünk akkor szerezzünk be egyet.

Dbtech régebbi videója, mely bemutatja, hogy egy domain CloudFlare alá hogyan csatolható be: [Link1](#) , [Link2](#)

3. Hálózat:

Jelenleg Nginx Proxy Managert használok, viszont régebben swag-et használtam. Ebben az esetben a netre kiengedett alkalmazásoknak létrehoztam egy külön hálózatot **web** néven, melyen azok az alkalmazásaim futottak, melyeket a netre kiengedtem.

SSH-n az alábbi parancsot kiadva elkészítheted: (nem kötelező)

docker create network web => ebben az esetben lesz egy web nevű hálózat, melyen majd manuálisan fogjuk átállítani azokat a konténereket melyeket a netre kiengedünk.

3.8.1. Dinamikus DNS frissítés – DDNS

Itt lehetőség van akár konténerrel, vagy magával a routerrel is frissíteni az IP-t.

Mindenki, akinek van internetes előfizetése vagy fix IP-vel csatlakozik a netre, vagy dinamikusan változik pl.: Digi, Telekom, stb

Minden esetben a szolgáltató IP címünkkel (<https://www.whatismyip.com/>), csatlakozhatunk az otthon kiengedett alkalmazásainkhoz, viszont a számokat megjegyezni nem olyan egyszerű, ezért inkább válasszuk az ingyenes/fizetős domain vagy routerhez kapott DDNS-t.

3.8.2. Dinamikus DNS frissítés – router saját címével

Ebben az esetben nincs szükségünk konténerre, mert a routerünk saját kezűleg frissíti az IP-t amellyel távolról elérjük a NAS-on lévő konténereinket.

Ezt a routeren belül egy DDNS nevű menüpontban találjátok meg általában.

3.8.3. Dinamikus DNS frissítés – Duckdns

Ide fájlt nem másolok/készítek mert minden fenn van a hivatalos oldalon:

Dokcerhu : [Link](#)

3.8.4. Dinamikus DNS frissítés – ddclient

Dokcerhub: [Link](#)

Telepítés menete: [3.1 rész](#)



ddclient.txt

(A beágyazott txt fájlra kattintva előjön az általam használt Stack Portainerben!)

Ezenfelül a megadott **/config** mappában van egy fájlunk (ddclient.conf), melyet szerkesztenünk kell a szükséges belépési adatokkal.

Itt a fájlban számos DDNS szolgáltatóhoz van előre definiált megoldás.

Ebben az esetben ha kiválasztottuk a szolgáltatót, akkor a sorok előtti #-et kell kieszelnünk és menteni a fájlunkat, majd a konténert újraindítani.

Én a cloudflare-es részt választottam, amit az alábbi módon írtam át (szedd ki a #-et és írd be az alábbiakat)

```
use=web
protocol=cloudflare, \
zone=domainneved.com, \
ttl=1, \
login=emailcímed, \
password=token \
domainneved.com
```

Cloudflare esetén a DNS menüpontban az ott lévő sárga felhőre kattintsunk, hogy Proxied módban lehessünk. Ez azt jelenti, hogy a mi konténerünk a root domainév IP-jét frissíti, a többi CNAME értékét a cloudflare felügyeli. Ez sokkal biztonságosabb, de ha nem menne akkor álljunk vissza **DNS only** módba. (sárga felhőre kattintani, hogy szürke felhő legyen)

3.8.5. Dinamikus DNS frissítés – cloudflare

cloudflare-es DDNS szolgáltató esetén, ha ezt használjuk akkor az alábbi videóban leírtakat célszerű használni :

[CloudFlare DDNS - Update CloudFlare with Your Dynamic IP Address](#)

FONTOS: Én egyelőre a Ddclient konténert használom csak, az onzu-t egyenlőre hanyagoltam, de így is megy a CF Proxied módban.

3.9. Webszerver és tanúsítványkezelő

A Webszerver feladata a hálózaton lévő alkalmazások menedzselése, a tanúsítványok kezelése, illetve kezelni a netre kiengedett alkalmazásokat. Erre több megoldás is van:

- Swag – alias letsencrypt: Kezdőknek ajánlott
- Nginx Proxy Manager: Kezdőknek ajánlott, kattingatós
- Traefik: Hardcore felhasználóknak ajánlott 😊

3.9.1. swag – alias letsencrypt

Dockerhub: [Link](#)

A telepítés menete: [3.1 rész](#)

(A beágyazott txt fájlra kattintva előjön az általam használt Stack Portainerben!)

Fontos: [Hivatalos oldali link](#)

A konténerben megadott portokat routerben engedélyezni kell (ezt korábban írtam)

A swag konténer többfajta DDNS csatlakozási lehetőséget is ad pl.:
duckdns,cloudflare,no-ip,stb

Ezek kitöltését a hivatalos oldalon megtaláljátok, vagy neten keressetek rá.

Ami nagyon fontos az, hogy duckdns esetén Facebook és egyéb megosztó portálokon a linket ha másnak külditek, nem fog működni, mert 3rd party dns-re hivatkozik.

Ezt E-mailben történő linkkel lehet kiküszöbölni!

Nem kötelező web hálózatot létrehozni, régebben én így csináltam!

De ha van létrehozva, akkor így kell a hálózat nevének a cseréjét megcsinálni.

Feladat: Ezek után nincs más teendőnk, mint Portainerben a swag konténerhez az általunk létrehozott **web vagy akármilyen nevű hálózatra** cseréljük.

Portainer => Containers => tranmission klikk => Duplicate /edit =>Network : web

Akkor csináltátok jól ha server ready-t kaptok, mint a képen:

```
□ DHLEVEL=2048
VALIDATION=dns
DNSPLUGIN=cloudflare
EMAIL=
STAGING=
2048 bit DH parameters present
SUBDOMAINS entered, processing
Wildcard cert for stigmattech.com will be requested
No e-mail address entered or address invalid
dns validation via cloudflare plugin is selected
Certificate exists; parameters unchanged; starting nginx
Starting 2019/12/30, GeoIP2 databases require personal license key to download. Please retrieve a free
and add a new env variable "MAXMINDDB_LICENSE_KEY", set to your license key.
[cont-init.d] 50-config: exited 0.
[cont-init.d] 60-renew: executing...
The cert does not expire within the next day. Letting the cron script handle the renewal attempts overn
[cont-init.d] 60-renew: exited 0.
[cont-init.d] 99-custom-files: executing...
[custom-init] no custom files found exiting...
[cont-init.d] 99-custom-files: exited 0.
[cont-init.d] done.
[services.d] starting services
[services.d] done.
□ Server ready
nginx: [alert] detected a LuaJIT version which is not OpenResty's; many optimizations will be disabled
s from https://openresty.org/en/download.html)
No MaxMind license key found; exiting. Please enter your license key into /etc/conf.d/libmaxminddb
run-parts: /etc/periodic/weekly/libmaxminddb: exit status 1
```

Ha Cloudflare-t használsz, akkor a dns-conf mappában lesz egy cloudflare.ini fájl melyben az email címed és a API token-t kell megadni. Ha duckdns vagy mást használsz ezt hagyd figyelmen kívül.

Ezek után már csak annyi feladatotok van, hogy azokat a konténereket, melyeket netre akartok kiengedni, a **swag** mappán belül egy fájlt kell „aktiválni” vagyis pontosabban létrehozni.

Ezt valószínűleg SSH-n vagy akár hálózaton is tudjátok szerkeszteni.

Sorban az alábbiakat kell megtenned:

1. Leállítod Portainerben a swag konténert
2. A swag mappáján belül megkeresed a **nginx/proxy-confs** mappát és belemész vagy kiadod az alábbi parancsot

cd /srv/dev-disk-by-label-WDP4TB/Docker/swag/nginx/proxy-confs

3. lépés: Az itt található conf.sample fájlokból .conf-ot kell csinálnod, amelyeket a netre akarod kiengedni:

Hálózaton:

pl: qbittorrent.subdomain.conf.sample fájlból csinálsz qbittorrent.subdomain.conf fájlt

paranccsal: Belemész SSH-n a **cd /srv/.../swag/nginx/proxy-confs** mappába

cp qbittorrent.subdomain.conf.sample qbittorrent.subdomain.conf

Ezzel a paranccsal létre is hozod a fájlot.

4.lépés: A fájlt nyisd meg és ellenőrizd hogy a portod a helyén van. Nálam a 8081-es porton jön be a qbittorrent, ezért a fájlban át kell írnom.

nyisd meg az új conf kiterjesztésű fájlt: nano **qbittorrent**.subdomain.conf

```
#include /config/nginx/authelia-location.conf;

include /config/nginx/proxy.conf;
resolver 127.0.0.11 valid=30s;
set $upstream_app qbittorrent;
set $upstream_port 8080;
set $upstream_proto http;
proxy_pass $upstream_proto://$upstream_app:$upstream_port;

proxy_set_header Referer '';
proxy_set_header Host $upstream_app:$upstream_port;
}
```

Az fájlban található összes helyen a 8080-as portot ki kell cserélni 8081-re.

5. lépés: Mivel a qbittorrent-et engedem ki a netre, ezért qbittorrent Network hálózatát is át kell állítanom a web hálózatra, hogy egy hálózaton legyen a titkosításért felelős konténeremmel (swag)

A biztonság növelését vagy olyan konténerek szabályozását, amely nem kér jelszót, az alábbi módon tudjuk szabályozni:

- htpasswd fájlal: [Link](#) => .htpasswd részénél látjátok
- Authelia: [Link](#)

Egyéb más konténer netre engedésében is ugyanígy kell eljárni pl:

nextcloud, plex, emby, netdata, transmission

viszont lehet, hogy ezeknél még a konténereken belül szükséges más konfiguráció is hogy elérjétek.

3.9.2. Nginx Proxy Manager

A konténerhez tartozó stacket/sorokat itt találod: <https://nginxproxymanager.com/setup/>

A telepítés menete: [3.1 rész](#)



nginxproxymanager.txt

A konténerben megadott portokat a routerben engedélyezni kell (ezt korábban írtam)

Itt mivel 3-as verzió csak docker-compose fájlal tudod futtatni.

Ha futtatni nem tudod Portainerből, akkor docker-compose fájlként futtasd, mint a [PhotoPrism](#) konténert.

Ha Portainerből futtatod, akkor a Volume esetén a `.`-al megadott részeket cserélned kell!

Docker compose fájlban így néz ki:

- `./data:/data`

Portainerben erre cseréld:

- `/srv/dev-disk-by-label-WDP4TB/Docker/NPM/data:/data`

kb 1 perc után a konténert el tudod érni az alábbi címen: `http://NASIP:81`

Bejelentkezési adatok az alábbiak lesznek:

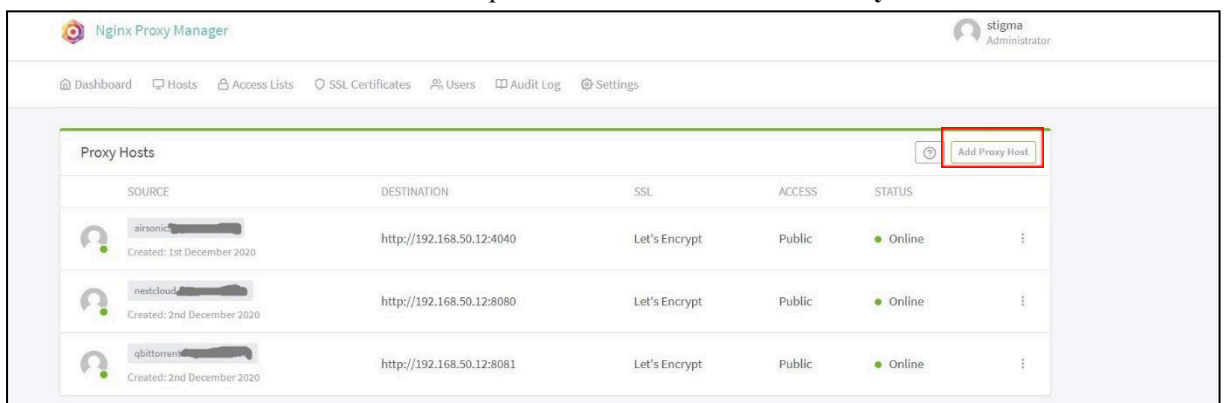
Email: `admin@example.com`

Password: `changeme`

Ezt mindenképp változtasd meg első bejelentkezés után!!

3.9.2.1. SSL és tanúsítvány generálás

A netes eléréshez az alábbi menüpontban éred el: **HOST => Proxy Host**



Itt láthatjátok már hogy nekem az alábbi pár konténerem van kiengedve a netre

3.9.2.2. Nextcloud netes elérése

A fenti képen látható **Add Proxy Host**-ra kell kattintani és kitölteni az alábbiakat:

The image displays two screenshots of the 'Edit Proxy Host' configuration window. The left screenshot shows the 'Details' tab, and the right screenshot shows the 'SSL' tab.

Details Tab:

- Domain Names ***: nextcloud
- Scheme ***: http
- Forward Hostname / IP ***: 192.168.50.12
- Forward Port ***: 8080
- Cache Assets**: ☐
- Block Common Exploits**: ☒
- Websockets Support**: ☐
- Access List**: Publicly Accessible

SSL Tab:

- SSL Certificate**: Request a new SSL Certificate
- Force SSL**: ☒
- HTTP/2 Support**: ☐
- HSTS Enabled**: ☐
- HSTS Subdomains**: ☐
- Use a DNS Challenge**: ☒
- DNS Provider ***: Cloudflare
- Credentials File Content ***:

```
# Cloudflare API token
dns_cloudflare_api_token = 0123456789abcdef0123456789abcdef01234567
```
- Propagation Seconds**: (empty field)
- Email Address for Let's Encrypt ***: @gmail.com
- I Agree to the Let's Encrypt Terms of Service ***: ☒

Amit itt át kell írnok:

Detail-s fül:

- **Domain Names** : nextcloud.domaineved.com
- **Foward Hostname/IP**: NAS Ip címe , amin fut a nextcloud
- **Foward Port**: a megadott portod amin eléred (ez lehet 8080 és 443 is)
- **Block common Exploits**: ezt kapcsolod be, legyen zöld

SSL fül

- **Force SSL**: legyen bekapcsolva:
- **???HTTP/2**: ingyenes Cloudflare esetén nem támogatott más DDNS esetén próbáld meg bekapcsolni.
- **Use DNS challenge**: routeres DDNs esetén ne kapcsolod be, Duckdns és Cloudflare esetén érdemes bekapcsolni.
- **DNS Provider**: Az általad használt DDNS szolgáltatót válaszd pl.: cloudflare, duckdns
- **Token**: Az alatta látható részhez a [cloudflare tokened](#) (ami a videoban lesz), vagy ducdksn tokened másold be.

- **Email:** ki kell tölteni majd

Ezek után a SAVE-re kell rányomni.

Itt ha nem kapsz hibát, akkor a tanúsítvány legyártódott, viszont ha 500-as vagy bármilyen más ERROR-t kapsz az azért van, mert:

- Nginx Proxy Manager portjaid nincsenek kiengedve a routerben
- http/2 engedélyezése bekavar, akkor próbálj meg anélkül legenerálni

Ha a belső porton a netxtcloud 8080 helyett 443-as porton használod (vagy más portra irányítod át), akkor a **Detail** fülön a **scheme**-nél **https**-t írd be/válassz ki.

3.9.2.3. Qbittorrent netes elérése

A fenti képen látható **Add Proxy Host**-ra kell kattintani és kitölteni az alábbiakat:

Én a qbittorrent-et a 8081-es porton használom, ha neked másik porton van, írd át akkor.

The image displays two screenshots of the Nginx Proxy Manager 'Edit Proxy Host' interface. The left screenshot shows the 'Details' tab with the following configuration:

- Domain Names ***: qbittorrent
- Scheme ***: http
- Forward Hostname / IP ***: 192.168.50.12
- Forward Port ***: 8081
- Cache Assets**: ☐
- Block Common Exploits**: ☒
- Websockets Support**: ☐
- Access List**: Publicly Accessible

The right screenshot shows the 'SSL' tab with the following configuration:

- SSL Certificate**: Request a new SSL Certificate
- Force SSL**: ☒
- HTTP/2 Support**: ☐
- HSTS Enabled**: ☐
- HSTS Subdomains**: ☐
- Use a DNS Challenge**: ☒
- DNS Provider ***: Cloudflare
- Credentials File Content ***: # Cloudflare API token
dns_cloudflare_api_token = 0123456789abcdef0123456789abcdef01234567
- Propagation Seconds**: (empty field)
- Email Address for Let's Encrypt ***: @gmail.com
- I Agree to the Let's Encrypt Terms of Service ***: ☒

Amit itt át kell írnotok:

Detail-s fül:

- **Domain Names** : qbittorrent.domaineved.com
- **Foward Hostname/IP**: NAS IP címe , amin fut a qbittorrent

- **Foward Port:** a megadott portod, amin eléred (ez lehet 8080 és 443 is)
- **Block common Exploits:** ezt kapsold be, legyen zöld

SSL fül

- **Force SSL:** legyen bekapcsolva:
- **???HTTP/2:** ingyenes Cloudflare esetén nem támogatott más DDNS esetén próbáld meg bekapcsolni.
- **Use DNS challange:** routeres DDNs esetén ne kapsold be, Duckdns és Cloudflare esetén érdemes bekapcsolni.
- **DNS Provider:** Az általad használt DDNS szolgáltatót válaszd pl.: cloudflare, duckdns
- **Token:** Az alatta látható részhez a [cloudflare tokened](#) (ami a videoban lesz), vagy duckdns tokened másold be.
- **Email:** ki kell tölteni majd

Ezek után a SAVE-re kell rányomni.

3.9.2.4. Jelszavas védelem konténerekhez – netdata

Ilyen konténer például a netdata, de szabadon más konténerekhez is alkalmazható.

Dockerhub: [link](#)

A linken szereplő részt bemásoltam egy stack-be majd lefuttattam, utána az alábbi konfigurációk voltak szükségesek.

Felhasználók definiálása: Access List => Add Access List

Name: ide bármit megadhatok én test-et adtam meg. Mikor a netre kiengedett konténert definiálok akkor fogom majd ezt választani.

Satisfy Any: ez legyen bekapcsolva

Az itt megadott felhasználónevekkel fogsz tudni belépni a konténerekhez, melyeket a netre engedek ki

Edit Access List

Details
Authorization
Access

allow

allow

allow

allow

deny

all

Note that the `allow` and `deny` directives will be applied in the order they are defined.

Cancel
Save

Itt én nem korlátoztam semmit sem, viszont ide meg lehet adni akár belső ip-s címeket is mellyel az eléréseket akard korlátozni (pl.: 192.168.XX.XX)

Host =>Proxy Host => **Add Proxy Host**

Edit Proxy Host

Details
Custom locations
SSL
Advanced

Domain Names *

netdata.

Scheme *

Forward Hostname / IP *

Forward Port *

http

192.168.50.12

19999

Cache Assets

Block Common Exploits

Websockets Support

Access List

test

Cancel
Save

Itt a kitöltés majdnem ugyanaz, mint eddig, viszont egy dologban eltér, mégpedig alul az Access List!

Access list-nél kiválasztod az általad generált mostani elérési listát (nekem test volt) majd az SSL-es fület a már megadottak alapján kitöltöd.

Így az én Netdata-s netes elérésű konténerem az általam a megadott stigma felhasználóval tudok belépni.

3.10. Eszközök szinkronizálása és backupja

3.10.1. Resilio : hálózati szinkronizálás

Youtube video : [Link](#)

Dockerhub: [Resilio](#)



Nekem a felállítás az alábbi:

- Fő NAS (Host) : OMV5
- Backup NAS: Xpenology DMS 6.3.2

Mindkét NAS-ra feltettem a Relisio nevű programot. OMV5-be Dockerbe melynek stackjét mellékeltem txt-be, DMS-re pedig az alkalmazások között megtaláltam.

Itt nemcsak NAS-NAS, hanem NAS-PC és más eszközökkel is összehozható a kommunikáció

Ezek után egy jelszót és felhasználónevet kell megadni

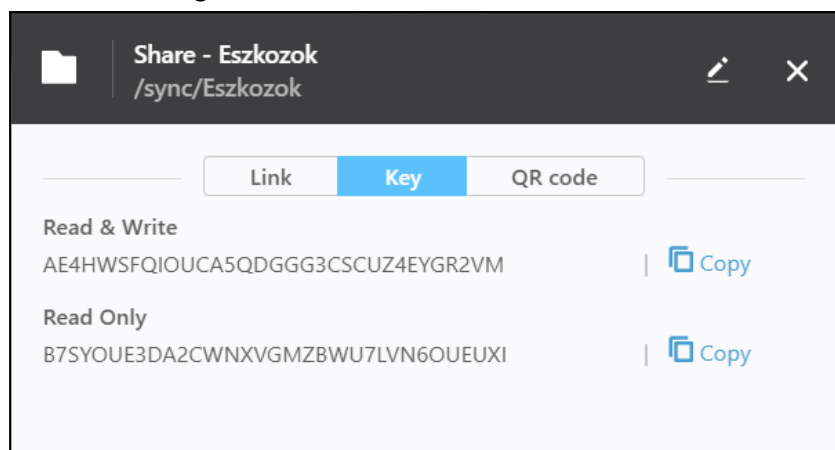
Ezek után a konfigurálás:

- OMV5: (Host)
 - Felvenni a mappákat: => Bal felül pluszjel / Standard folder
Amit felvett és szinkronizált, ott pipa van, amit még nem osztottam meg ,
hogy szinkronizáljon ott az „emberke jel”

+			
All	Connected	Disconnected	Local
	Name	Status	Peers online
	 Eszkozok	✓	1 of 1
	 Kepek		0 of 0
	 Teszt	✓	1 of 1

- DMS(Xpenology): Backup NAS

A kapcsolathoz egyfajta kódra szükség lesz, amivel a 2 NAS 2 különböző Mappája szinkronizálni fog:

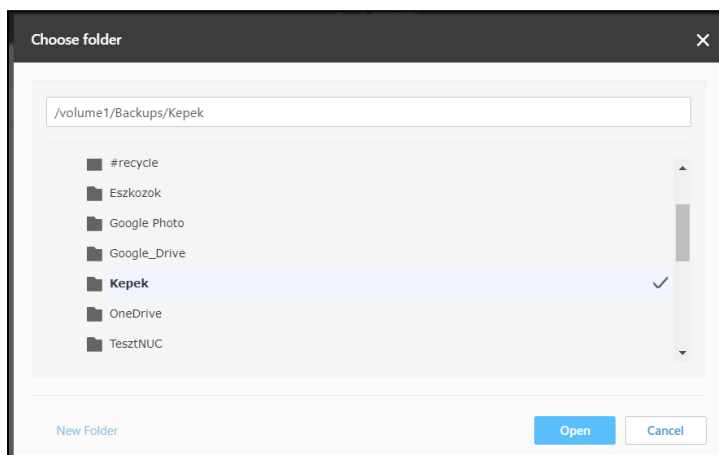
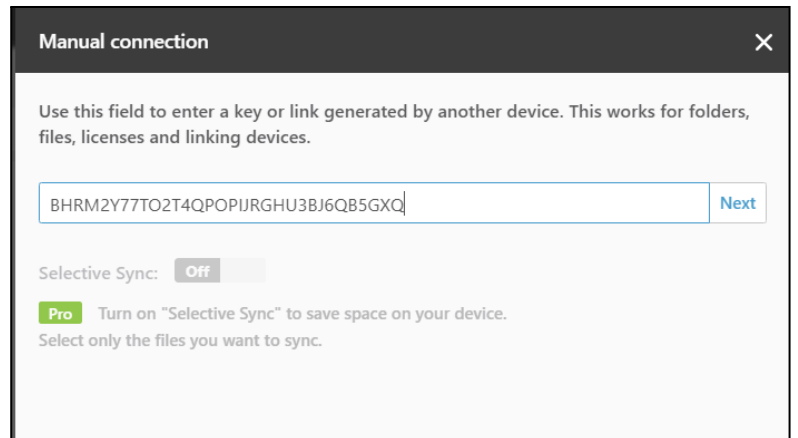
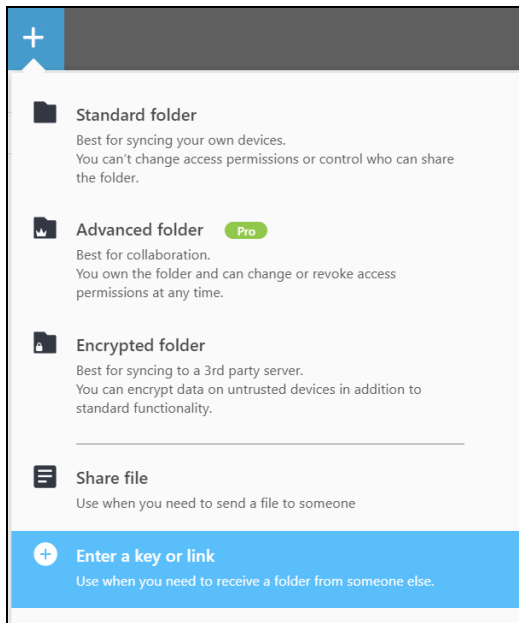


Itt az alábbi jelenti a 2 megnevezés:

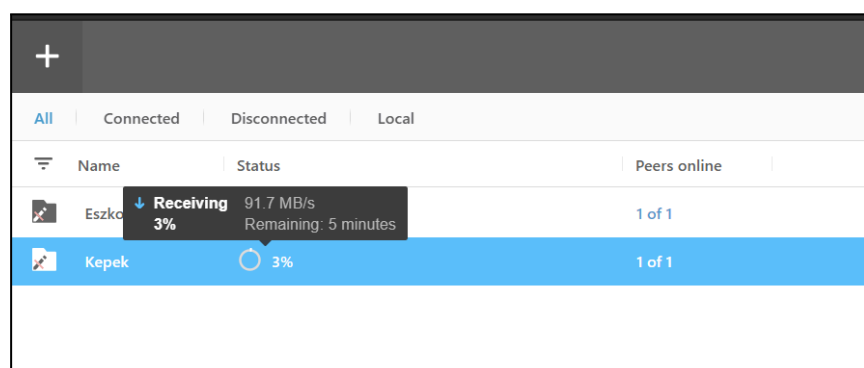
- **Read Only:** Itt csak egyirányú a kommunikáció. Ha a Host gépről, ahonnan a backupot készítem, akkor törlődik a Backupról is, de fordítva nem.
- **Read & Write:** Itt bármelyik gépről törlök, mindkét helyről törlődik.

Mappák összekapcsolása, hogy szinkronizáljon:

- Host gépen (OMV-n) kiválasztasz egy mappát /Jobb klikk / Share : majd a megfelelő kódot kimásoljuk, hogy 1 vagy 2 irányú legyen a kommunikáció
- Backup gépen: Plusz jel / Enter a key or Link / bemásoljuk a kódot majd kiválasztjuk a mappát hova akarjuk szinkronizálni:
- Képekkel mellékelve:



Hozzáadva így néz ki a szinkronizáció



Itt lehetőségünk van más eszközt pl: PC-t is hozzáadni.

3.10.2. Rsync: HDD-n belüli másolás

Ezt nem fejtem ki, mert nagyon sok Youtube videó van fenn a neten.

3.11. Duplicati – tökéletes Backup

Dockerhub: [Linuxserver](#) verziót használtam



duplicati.txt

(A beágyazott txt fájlra kattintva előjön az általam használt Stack Portainerben!)

dbtech Youtube video: [Link](#)

Én ezek alapján állítottam be. Nekem csak a Docker konténerjeim adatait menti ki egy másik HDD-re, mely a NAS-ra van 2 naponta. Maximum 3 verzió van amit eltárolok, ha több van automatikusan törölni fog.

- /config mappa : Ide menti a konténer futtatásához szükséges konfigurációs fájlokat
- /source : Ez az a forrás mappa, amiről szeretnék backup-ot készíteni
- /backup : Ez az a mappa, ahova a mentést elkészíti a duplicati

3.12. Több konténer egy helyen – Multistack

Ezt a megnevezés lehet nem létezik, én így definiáltam.

Ennek az az előnye, hogy 1 stackben vagy docker compose fájlban több konténert is definiálsz melyek azonos hálózatot használnak, vagy épp összetartoznak.

Erre 1-2 példát írok:

- Nextcloud: linuxserver\nextcloud + linuxserver + mariadb
- Automatikus letöltések (filmek, sorozatok, feliratok)
Jackett+Sonarr (sorozatok) + Radarr (Filmek) + Bazarr (feliratok)
- Vagy akár 1 stackben az összes használt

Automatikus letöltésre az alábbi konténereket tettem egybe:



autodownloads.txt

Ezáltal nem kell külön-külön futtatni hanem egy kattintásra mind települ és máris konfigurálható.