

Hosted Applications

Info about my locally hosted applications

- [Nextcloud](#)
- [Jellyseer](#)
- [Jellyfin](#)
- [Tdarr](#)
- [RustDesk](#)
- [ControlD](#)

Nextcloud

Nextcloud is installed as an Add-On in Home Assistant. The Data dir is linked to a TerraMaster NAS folder named nextcloud using Samba/SMB.

Running OCC Commands: Not yet setup

1. Configure debugging: <https://developers.home-assistant.io/docs/operating-system/debugging/>
2. Remotely Login to HA through SSH
3. Run command: `docker exec addon_db21ed7f_nextcloud_ocr sudo -u abc php /app/www/public/occ SOME_COMMAND`
 1. For an interactive terminal start with `docker exec -ti ...`

Also, crontab is probably located here: /data/config/crontab

Useful Nextcloud OCC Commands:

https://docs.nextcloud.com/server/latest/admin_manual/occ_command.html

- Scan for new files added from outside NextCloud: [files:scan](#)
- Pre-Generate image previews: [preview:pre-generate](#)
- Cleanup and mimetype migrations: `maintenance:repair --include-expensive`
- Add missing DB indexes: `db:add-missing-indices`

Nextcloud DB: Using MariaDB Add-On in Home Assistant

- homeassistant.local:3306
- database: nextcloud
- username: service

Nextcloud config in HA:

```
Full_Text_Search: false
OCR: false
OCRLANG: eng
PGID: 1000
PUID: 1000
additional_apps: inotify-tools
certfile: fullchain.pem
```

```
elasticsearch_server: <ip:port>
enable_thumbnails: true
keyfile: privkey.pem
trusted_domains: >-
    home.freakycowbot.com,home.freakycowbot.com:8099,homeassistant.local,192.168.235.211
use_own_certs: true
default_phone_region: EN
networkdisks: //192.168.100.101/nextcloud
cifsusername: nextcloud_user
cifspassword: **REMOVED**
```

Initial Nextcloud Setup page:

DB credentials originally come from the Nextcloud logs on initial boot.

Using mounted SMB drive for storage (See config above). Not mounted in Home Assistant.



Create an admin account

New admin account name

New admin password



Storage & database ▾

Data folder

Configure the database

SQLite

MySQL/MariaDB

PostgreSQL

Database account

Database password



Database name

Database host

Jellyseer

Installed on server: Media PC, port 5055 using Docker

URL: <http://192.168.100.100:5055>, <http://jaggedjax.chickenkiller.com:5055/>

Jellyseer install docs: <https://docs.jellyseerr.dev/getting-started/docker?docker-methods=docker-cli>

Location: /opt/jellyseerr

Install command:

```
sudo docker run -d \  
  --name jellyseerr \  
  --user=1000:1001 \  
  -e LOG_LEVEL=debug \  
  -e TZ=America/Los_Angeles \  
  -e PORT=5055 `#optional` \  
  -p 5055:5055 \  
  -v /opt/jellyseerr:/app/config \  
  --restart unless-stopped \  
  fallenbagel/jellyseerr
```

Update Command:

```
sudo docker stop jellyseerr && sudo docker rm jellyseerr  
sudo docker pull fallenbagel/jellyseerr
```

And then run the above install command again

Jellyfin

Logs: `/var/log/jellyfin`

Plugin dir: `/var/lib/jellyfin/plugins/`

Other/Config dir: `/etc/jellyfin/`

Tdarr

Automatic video transcoding to H.265 - A more modern and space efficient format

Tdarr server

Running as a Home Assistant addon:

https://home.freakycowbot.com/hassio/addon/db21ed7f_tdarr/info

UI: <http://homeassistant.local:8265>

Tdarr nodes

- Home Assistant - This node can only use CPU transcoding and is trash. It's running alongside the server
 - Ideally it should be using VA-API, but it either is not, or this CPU doesn't support H.265
- MediaPC - `/opt/Tdarr/Tdarr_Node`
- Pop_OS Dell - `/opt/Tdarr/Tdarr_Node`
 - Laptop. Can stop service when needed with: `sudo service tdarr stop`

Node Service

To create a linux service so the Tdarr node automatically starts/runs:

```
sudo vim /etc/systemd/system/tdarr.service
```

```
[Unit]
```

```
Description=Tdarr - Audio/Video Library Analytics & Transcode/Remux Automation
```

```
Documentation=https://docs.tdarr.io/docs
```

```
[Service]
```

```
ExecStart=/opt/Tdarr/Tdarr_Node/Tdarr_Node
```

```
Restart=on-failure
```

```
[Install]
```

```
WantedBy=multi-user.target
```

```
sudo systemctl enable tdarr.service
```

```
sudo service tdarr start
```

Nodes should auto-update. See `/opt/Tdarr/logs` for update log files

You can manually update nodes by running `/opt/Tdarr/Tdarr_Updater`

RustDesk

Installed on server: Media PC as a service. Ports forwarded in router per RustDesk install directions.

- ID Server: `jaggedjax.chickenkiller.com`
- Public Key: `fKM0a7UF+m6HZkxNuqVbkaaPzGKY8RAGjjVUmwsGoPc=`

Client Setup

You can access your install scripts for clients by going to <http://jaggedjax.chickenkiller.com:8000> or just finding them like you would normally.

Once installed:

1. Go to Settings > Network > ID/Relay Server
 1. ID Server: `jaggedjax.chickenkiller.com`
 2. Public Key: `fKM0a7UF+m6HZkxNuqVbkaaPzGKY8RAGjjVUmwsGoPc=`
 3. The rest can be blank.
 1. Note: Relay Server may also need `jaggedjax.chickenkiller.com` for some external access, but probably not.

Update RustDesk Server:

1. Connect to Media PC
2. CD to `/home/william`
3. Run: `./rustdesk_update.sh`
 1. Update script from: <https://github.com/techahold/rustdeskinstall?tab=readme-ov-file#how-to-update-the-server>

You can restart RustDesk server with the following command:

```
sudo systemctl restart rustdesk signal rustdeskrelay gohttpserver
```

Installed Clients:

- Media PC: 358 816 069
- Dell PopOS Work: 290 658 077
- System76 Darter Pro: 459 844 153

ControlD

Hosted on: UniFi Cloud Gateway Ultra

IP: 192.168.1.1

The [ControlD application](#) is installed on this router to manage DNS.

Install

1. SSH into router
 1. Docs: <https://help.ui.com/hc/en-us/articles/204909374-Connecting-to-UniFi-with-Debug-Tools-SSH>
2. Run command:

```
sh -c 'sh -c "$(curl -sL https://api.controlld.com/dl)" -s cuayo73hcv forced'
```

 1. The resolver ID here "cuayo73hcv" comes from the ControlD Endpoint dashboard

Instructions from here: <https://docs.controlld.com/docs/ctrlld#quick-install>

Upgrade

1. SSH into router
2. Run command:

```
/data/controlld/ctrlld upgrade
```

 1. Can also run the install command again leaving off the "-s XXX forced" arguments