

Tips zum Betrieb eines Immich Foto Servers (C) Werner Joss 2024

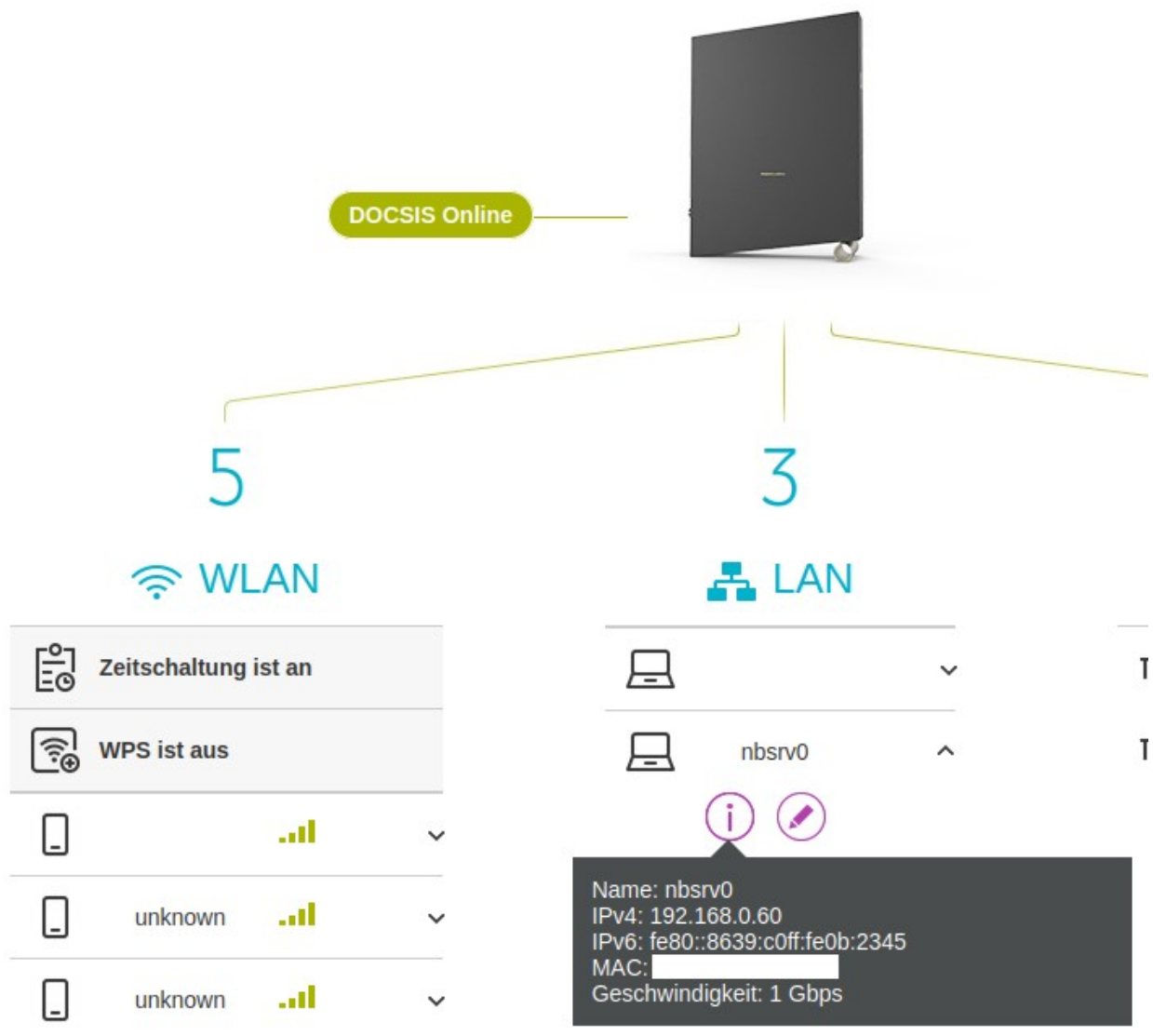
Voraussetzung ist hier eine Installation mit [docker-compose](#).

Zunächst wird er so konfiguriert, dass er bei Anschluss mit RJ45 Lan Kabel an einen Router (z.B. Fritzbox) out-of-the-box im lokalen Netzwerk funktioniert:

Das Gerät wird auf DHCP eingestellt und bezieht seine lokale IP Adresse vom Router.

Das sollte für die ersten Gehversuche ok sein, sobald die zugewiesene IP Adresse (im Folgenden als IPADDR bezeichnet) bekannt ist.

Am einfachsten schaut man diese im Web-Interface (Verwaltungspanel) des Routers nach:

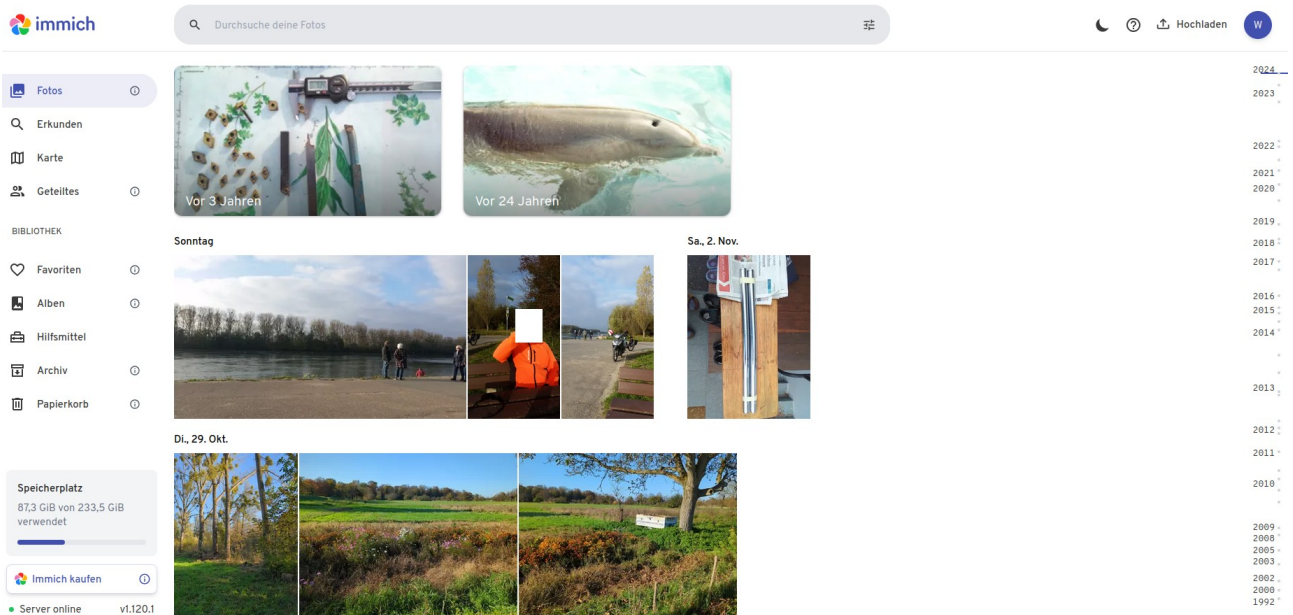


Alternativ kann auch ein Portscanner verwendet werden, z.B. nmap oder die App [Port Authority](#).

Im Browser kann man dann den Immich Server unter <http://IPADDR:2283> erreichen.

Beim ersten Start werden dann die Zugangsdaten für den Admin-User (Mailadresse + Passwort) festgelegt und mit diesen erfolgt dann der erste Login.

Was dann zu sehen ist, erinnert stark an Google Fotos :



Erste Fotos Hochladen geht direkt aus der Oberfläche (oben rechts) und ist eigentlich selbsterklärend.

Ebenso das Erstellen und Verwalten von Alben.

Anschauen ist auch intuitiv, incl. Diashow usw.

Soweit also ähnlich zu Google, aber es gibt mehr:

Eine Besonderheit ist die eingebaute Gesichtserkennung, die ist recht leistungsfähig und lohnt sich zu ergründen !

Des weiteren ist der Upload von Fotos via Commandline eine super Sache:

VIEL bequemer und schneller als via Webinterface !

Und dabei können die Bilder auch gleich einem Album zugeordnet werden.

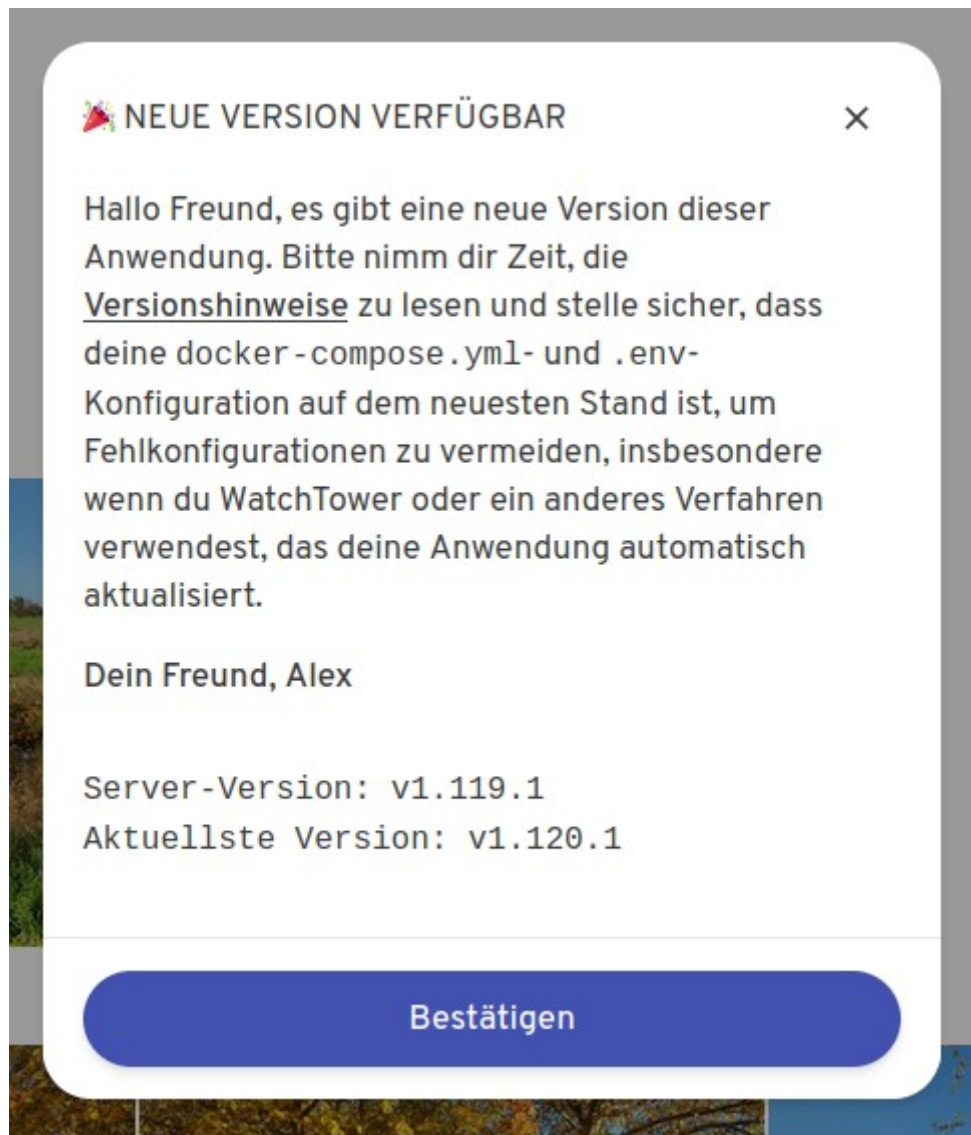
Der Haken dabei: man braucht dazu Kenntnisse im Umgang mit dem Terminal und muss auch die aktuelle Version von node.js und das Immich CLI auf dem Client von dem aus das erfolgen soll, installieren, die Anleitung dazu findet sich [hier](#) – sicher ein gewisser Aufwand, aber es lohnt sich ! Einfacher ist da der Upload bzw. Sicherung der Fotos vom Handy/Tablet via der entsprechenden Immich-App - die gibt es für Android und IOS.

Der Server selber benötigt hin und wieder etwas Zuwendung:

die immich Entwickler sind sehr aktiv und liefern alle 1 bis 2 Wochen eine neue Version, auf diese gilt es jeweils upzudaten (dasselbe gilt für die mobilen Apps, dort eben via App-Store).

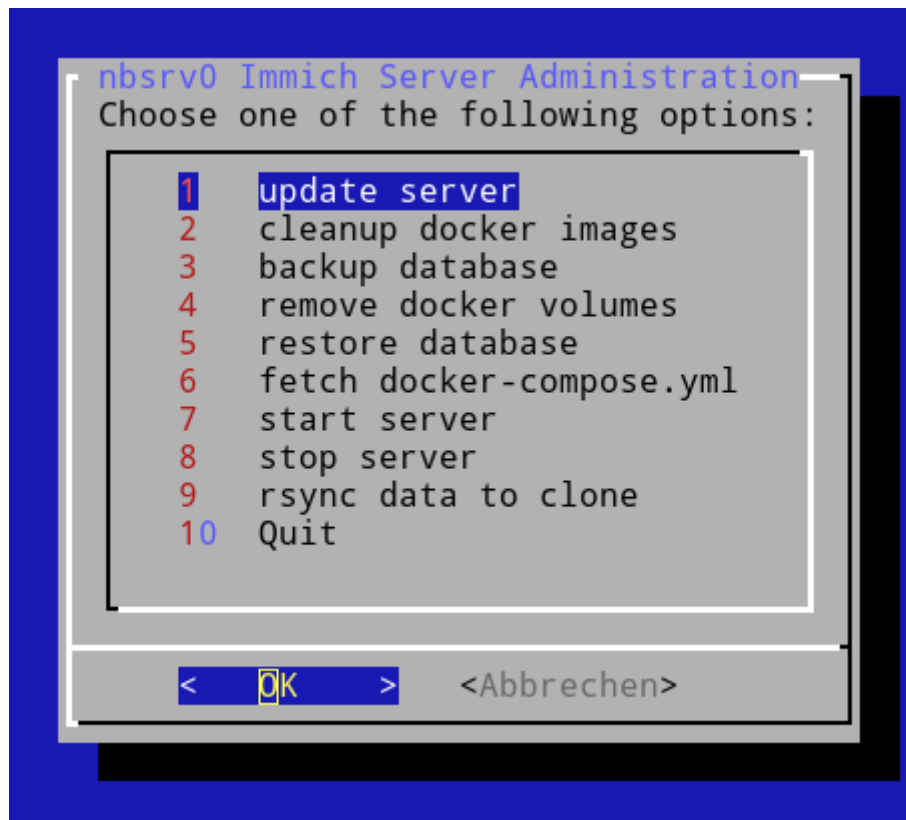
Wenn es soweit ist, erscheint beim Login ins Web-Interface ein Hinweis mit Link zum Changelog (Versionshinweise).

Da sollte man vor dem Update auf jeden Fall rein schauen und prüfen, ob es sogenannte 'breaking Changes' gibt.



In diesem (eher seltenen) Fall ist nämlich meist etwas Handarbeit angesagt, z.B. durch Änderung eines Eintrags in der Konfigurationsdatei (.env) oder durch Update / Neu Laden der Datei docker-compose.yml.

Das alles muss in einem Terminal auf dem Server gemacht werden, das via ssh erreichbar ist. Also im aktuellen Fall 'ssh -l username IPADDR' (IPADDR siehe oben). Die meisten Wartungsarbeiten lassen sich dort mittels [dieses Bash Scripts](#) erledigen:



Das Script muss auf dem Server installiert (am besten in einem Verzeichnis in \$PATH) und ausführbar sein, dazu (und zur Verwendung des Scripts) genügt ein ssh Zugang (ohne GUI) . Vor der ersten Anwendung muss die Einstellung IMMICH_HOME im Abschnitt settings geprüft und ggf. angepasst werden !

In den meisten Fällen genügt das schon, es kann aber auch noch die Variable DB_BACKUP_LOCATION angepasst werden, falls die Vorgabe nicht passt.

Anmerkungen zu den im Script gelisteten Aufgaben:

Die meisten Optionen sind selbsterklärend, jedoch nicht alle .

“rsync data to clone” wird z.B. nur dann gebraucht, wenn die Daten (=Fotos+Videos) 1:1 auf einem Klon des Servers repliziert werden sollen.

Ich habe einen solchen Klon eingerichtet, der nicht immer online ist, aber eben zur Not ein Fallback darstellt, sollte mein produktiver Server ausfallen und nicht (einfach) wieder herstellbar sein.

Die Vorgehensweise ist dabei folgende:

1. Backup der Datenbank auf dem Produktiv Server (wird auch gleich zum Klon kopiert, falls der online ist)
2. Stop Immich Server auf Klon
3. rsync Produktiv Server Daten und DB Dump (s.o.) zum Klon
4. restore Produktiv Dump auf dem Klon und starte dessen Server neu

Alle diese Aufgaben können mit je einer Instanz des Verwaltungs-Scripts auf dem Produktiv-System und dem Klon erledigt werden.

- Die Option “cleanup docker images” sollte von Zeit zu Zeit angewendet werden, da bei jedem Update neue Images gezogen werden, die alten bleiben aber als ‘Leichen’ stehen und belegen mit der Zeit ganz schön viel Speicherplatz.
- Die Option “remove docker volumes” habe ich bisher nur ein Mal gebraucht, nachdem der Server nach einem Update nicht mehr zu starten war (warum auch immer). Hier ist zu beachten, dass dabei ALLE Volumes gelöscht werden, also auch die von ggf. vorhandenen anderen docker Instanzen !
- deswegen wird hier auch nur ein extra Script erzeugt und nicht direkt ausgeführt, so dass darin ggf. enthaltene, unerwünschte Lösch-Aktionen vorher entfernt werden können.

Soweit ist das alles auf das lokale LAN beschränkt und funktioniert mit allen dort befindlichen Geräten / Computern.

Der nächste Schritt ist also, den Server von aussen zugänglich zu machen.
Dazu gibt es mehrere Möglichkeiten, zwei davon möchte ich hier kurz vorstellen:

1. via DynDns + Port-Forwarding

Hier muss der eigene Internet Router in der Lage sein, diese beiden Funktionen einzurichten. Ausserdem benötigt man einen Account bei einem DynDns Anbieter.

Ich selber verwende dazu [Dynu](#) - es gibt aber noch viele andere.

Bei diesem Anbieter muss nun ein DNS Name für die eigene externe IP Adresse eingerichtet werden (diese findet man z.B. auf <https://www.whatismyip.com/>), sowie ggf. eine Regel, falls diese Adresse vom eigenen Internet Provider regelmässig geändert wird.

Früher war das ständig der Fall, heutzutage ist das oft nicht mehr so, ich bekomme z.B. seit Jahren von Vodafone dieselbe externe IP Adresse.

Deshalb kann ich mir die automatische DNS Zuordnung (=DynDNS) sparen.



Also reicht es hier, am Router Port-Forwarding zu aktivieren, und die Ports 80 (http) und 443 (https) an die interne IP des Immich Servers weiterzuleiten.

Der sollte natürlich immer die selbe IP haben, auch das lässt sich am Router anhand der MAC Adresse einstellen:

Lokaler DHCPv4 Server



Statisches DHCP - Heimnetzwerk

Gerätename	MAC-Adresse	IP-Adresse	
	38:2C:4A:6C:C1:F7	192.168.0.3	 

Was dann noch fehlt, ist ein sogenannter Reverse Proxy, das heisst ein Webserver, der Anfragen von aussen an die interne Adresse `http://IPADDR:2283`

weiterleitet.

Das geht am besten mit [Caddy](#), eine dafür vorgefertigte Config-Datei sieht etwa so aus:

```
# The Caddyfile is an easy way to configure your Caddy web server.
#
# Unless the file starts with a global options block, the first
# uncommented line is always the address of your site.
#
# To use your own domain name (with automatic HTTPS), first make
# sure your domain's A/AAAA DNS records are properly pointed to
# this machine's public IP, then replace ":80" below with your
# domain name.

immich.nbsrv.ddnsfree.com {
    reverse_proxy http://localhost:2283
}
```

In diesem Beispiel muss die Domain nbsrv.ddnsfree.com beim DynDns Provider registriert sein, die Subdomain immich ist die des Immich Servers, der dann von Aussen unter <https://immich.nbsrv.ddnsfree.com/> erreichbar ist, sofern das Port-Forwarding am Router richtig eingerichtet ist.

Der Clou dabei ist, dass Caddy mit dieser einfachen Konfiguration nicht nur die Reverse Proxy Funktion übernimmt, sondern sich auch um die komplette automatische Einrichtung eines LetsEncrypt Zertifikats für den Server kümmert !

(Ein SSL Zertifikat ist für einen von aussen erreichbaren Server mit Login Möglichkeit ein absolutes **MUSS** !)

2. via Zerotier VPN

Diese Variante hat den Charme dass hier keinerlei zusätzliche Sicherheitsvorkehrungen (SSL Zertifikate...) notwendig sind da es sich hier um ein privates VPN handelt.

Der Nachteil ist allerdings, dass der Server dann nur von Geräten aus zu erreichen ist, die in dem VPN registriert sind – man kann also nicht ‚mal eben schnell‘ Bilder und Alben den Freunden und Bekannten zeigen.

Zur Einrichtung von Zerotier (wenn auch zu anderen Zwecken als hier) verweise ich deswegen auf [diesen Pro-Linux Artikel](#) , der auch sonst interessante Einblicke in die VPN Anwendung bietet.

Im aktuellen Fall (Immich Server) genügt es, den Server und die eigenen (auch mobilen) Clients in das eigene Zerotier Netz einzubinden, er ist dann unter http://ZT_SERVER_IP:2283/ von jedem Client aus erreichbar, ganz ohne Caddy, Reverse Proxy und auch ohne https – die Sicherheit wird hier durch das VPN erreicht !