

# Server Gmund

Foto <https://unsplash.com/photos/black-network-switch-with-cables-ISG-rUel0Uw>

- Allgemein
- VMs und Container
  - VM 100 - Home Assistant
  - Container 102 - Caddy
  - Container 103 - Vaultwarden
  - VM 104 - MQTT
  - VM 105 - Zamma
  - VM 106 - Minecraft
  - Container 107 - Listmonk
  - Container 108 - ptrackhoki
  - Container 109 - pconnectingpeaks
  - Container 111 - bikesensor & messi
  - VM 112 - Proxmox Backup Server
  - VM 114 - Project services
  - VM 116 - Internal Services
  - VM 117 - Oberlogin
  - VM 118 - Obercloud
  - VM 119 - old Wiki
  - VM 120 - Wiki
  - VM 121 - Paperless
- Server
  - Administration
  - IP Adressen
  - Hardware
  - Software
  - Proxmox







# Allgemein

## Bei Fragen

[Joel ansprechen](#)

## Server

[Hardware](#)

[Software](#)

[Administration](#)

[Proxmox](#)

[IP Adressen](#)

## Dienste

- [Vaultwarden](#) - Passwortmanager
- [Caddy](#) - Webserver & reverse proxy
- [Home Assistant](#) - Home automation
- [Minecraft](#)
- [Mosquitto](#) - MQTT Server
- [Listmonk](#) - Mailinglisten Manager
- [Keycloak](#) - Login / Single Sign-On Manager
- [Nextcloud](#)
- [Proxmox Backup Server](#)
- [Wiki](#)

## Projekte



- [Track-Hoki](#)
- [Connecting Peaks](#)

# VMs und Container

## VMs

- [100 - Home Assistant](#)
- [104 - MQTT](#)
- [105 - Zamma](#)
- [106 - Minecraft](#)
- [112 - Proxmox Backup Server](#)
- [117 - Oberlogin / Keycloak](#)
- [118 - Obercloud](#)
- [119 - Old Wiki](#)
- [120 - Wiki](#)

## Container

- [102 - Caddy](#)
- [103 - Vaultwarden](#)
- [107 - Listmonk](#)
- [108 - ptrackhoki](#)
- [109 - pconnectingpeaks](#)
- ~~110 - pbikesensor~~
- [111 - pbikesensor2](#)



# VMs und Container



# VM 100 - Home Assistant

## Technische details

CPU	4
RAM	2G
HDD	10G
OS	Ubuntu 22.04
IP	192.168.0.250/24
Domain	hass.oberlab.de
Ports	8123

## Dienste

Auf dem Container läuft Home Assistant, eine Home Automation Software, in einem Docker-Container

Nach Außen ist der Port 8123 der Standard-Weboberfläche erreichbar. Angesteuert wird der Port vom Caddy Container, der die Umsetzung nach HTTPS vornimmt.



# Container 102 - Caddy

## Technische details

CPU	4
RAM	2G
HDD	4G
OS	Ubuntu 22.04
IP	192.168.0.253/24
Domain	-
Ports	80, 443

## Dienste

Auf dem Container läuft Caddy, ein Web Reverse-Proxy. Er schirmt die internen Dienste ab, und bietet eine allgemeine Schnittstelle für die Ports 80/443 (HTTP/HTTPS).

Die Ports 80/443 vom Internetanschluss werden von der Kabelbox an Caddy weitergeleitet, somit kümmert sich Caddy um alle einkommenden Web-Anfragen und leitet sie entsprechend weiter.

Eine weitere Aufgabe von Caddy ist die automatische Beantragung und Erneuerung (alle 3 Monate) von SSL Zertifikaten über den freien Dienst Let's Encrypt.



# Container 103 - Vaultwarden

## Technische details

CPU	1
RAM	1G
HDD	4G
OS	Ubuntu 22.04
IP	192.168.168.35/24
Domain	<a href="https://pw.oberlab.de">pw.oberlab.de</a>
Ports	80

## Dienste

Auf dem Container läuft Vaultwarden, eine freie Implementierung und ressourcenschonende Variante von Bitwarden als Docker-Container. Vaultwarden implementiert dabei die API eines Bitwarden-Servers und ist damit mit allen Bitwarden Clients kompatibel

Nach Außen ist der Port 80 erreichbar. Angesteuert wird der Port vom Caddy Container, der die Umsetzung nach HTTPS vornimmt.

Vaultwarden benötigt zwingend eine HTTPS Verbindung, ansonsten kommt nur eine Seite mit einer Fehlermeldung ("Please access this site over HTTPS")



# VM 104 - MQTT

## Technische details

CPU	2
RAM	1G
HDD	8G
OS	Debian 11.3
IP	192.168.0.251/24
Domain	.
Ports	1883, 8883

## Dienste

Auf dem Container läuft Eclipse Mosquitto, ein open-source MQTT Server, in einem Docker-Container

Nach Außen sind die Ports 1883 (unverschlüsselt) und 8883 (TLS-Verschlüsselt) offen

Der Zugang zum MQTT Server ist nur mittels Login/Passwort möglich, anonyme Verbindung sind nicht möglich. Dies gilt sowohl für Devices, die Daten senden wollen, als auch für Dienste, die Daten abonnieren wollen.

## MQTT User anlegen

```
docker-compose exec mosquitto mosquitto_passwd -b /mosquitto/config/mosquitto.passwd username
```

## MQTT mitlesen



```
docker-compose exec mosquitto mosquitto_sub -u username -P passwort -t "#"
```

# MQTT Topics

Um einen beliebigen Wildwuchs and MQTT Topics zu verhindern sollten hier alle Topics aufgelistet und neue entsprechend des Schemas angelegt werden.

Kategorie	Topic	Beschreibung	Kontaktperson
Projekte	projects/connecting_peaks/<device>/+	Connecting Peaks	Joel
	projects/trackhoki/<device>/+	Track Hoki	Joel
Lab	gmund/hass/heizung/<device>/+	Home Automation im Gmunder Lab	Joel/John



# VM 105 - Zamma

## Technische details

CPU	12
RAM	2/4G
HDD	8G
OS	Debian 12 Bookworm
IP	192.168.168.8/24
Domain	zamma.oberlab.de
Ports	80, 1337, 1338, 3000, 5432

## Dienste

Hier laufen die Webseiten für Zamma und Zamma-Pflanzen, die jeweiligen Admin-Oberflächen, sowie die jeweiligen Postgres Datenbanken.



# VM 106 - Minecraft

## Technische details

CPU	6
RAM	4/8G
HDD	12G / 8G /home
OS	Ubuntu 22.04 Server
IP	192.168.0.249/24
Domain	minecraft.oberlab.de oberlab.chickenkiller
Ports	8000, 9000, 9443, 25565

## Dienste

Auf der VM läuft ein Minecraft von Julian/Konsti. Administriert wird der Container mittels [Portainer](#).

Nach Außen sind die Ports 8000, 9000, 9443 von Portainer erreichbar.

Minecraft verwendet den Port 25565.



# Container 107 - Listmonk

## Technische details

CPU	1
RAM	1G
HDD	8G
OS	Ubuntu 22.04
IP	192.168.168.3/24
Domain	<a href="https://listmonk.oberlab.de">listmonk.oberlab.de</a>
Ports	9000

## Dienste

Auf dem Container läuft ListMonk, eine freie Mailinglisten Verwaltungssoftware als Docker-Container.

Nach Außen ist der Port 9000 erreichbar. Angesteuert wird der Port vom Caddy Container, der die Umsetzung nach HTTPS vornimmt.



# Container 108 - ptrackhoki

## Technische details

CPU	4
RAM	2G
HDD	8G
OS	Ubuntu 22.04
IP	192.168.168.4/24
Domain	trackhoki.oberlab.de
Ports	3000, 8086

## Dienste

Auf dem Container laufen Telegraf, Influxdb und Grafana

Nach Außen ist der Port 3000 von Grafana erreichbar. Angesteuert wird der Port vom Caddy Container, der die Umsetzung nach HTTPS vornimmt.

Der Port 8086 von Influxdb ist verfügbar um auf die Datenbank z.B. per Skript zuzugreifen, wir aber für den Betrieb von Grafana nicht verwendet (Grafana benutzt das interne Docker-Netzwerk)



# Container 109 - pconnectingpeaks

## Technische details

CPU	4
RAM	2G
HDD	8G
OS	Ubuntu 22.04
IP	192.168.168.5/24
Domain	connectingpeaks.oberlab.de
Ports	3000, 8086

## Dienste

Auf dem Container laufen Telegraf, Influxdb und Grafana

Nach Außen ist der Port 3000 von Grafana erreichbar. Angesteuert wird der Port vom Caddy Container, der die Umsetzung nach HTTPS vornimmt.

Der Port 8086 von Influxdb ist verfügbar um auf die Datenbank z.B. per Skript zuzugreifen, wir aber für den Betrieb von Grafana nicht verwendet (Grafana benutzt das interne Docker-Netzwerk)

## Setup

### Grafana



in grafana.ini, den domain-Eintrag anpassen und Container neu starten

# Influxdb

Passwörter für admin, grafana und telegraf ändern

# Telegraf

Format der MQTT Pakete anpassen

Passwörter für MQTT & InfluxDB anpassen



# Container 111 - bikesensor & messi

## Technische details

CPU	2
RAM	1.5G
HDD	8G
OS	Ubuntu 22.04
IP	192.168.168.7/24
Domain	bikesensor.oberlab.de messi.oberlab.de messipush.oberlab.de
Ports	80, 3000, 8000, 8086

## Dienste

### Bikesensor Projekt

Der Webserver für das OpenBikeSensor Projekt läuft auf Port 8000, wird über ein Systemd-Service gestartet

Angesteuert wird der Port vom Caddy Container, der die Umsetzung nach HTTPS vornimmt.

### Messi Projekt

Auf dem Container laufen Influxdb und Grafana sowie ein cherrypy Webserver



Nach Außen ist der Port 3000 von Grafana erreichbar. Angesteuert wird der Port vom Caddy Container, der die Umsetzung nach HTTPS vornimmt.

Der Port 8086 von Influxdb ist verfügbar um auf die Datenbank z.B. per Skript zuzugreifen, wir aber für den Betrieb von Grafana nicht verwendet (Grafana benutzt das interne Docker-Netzwerk)

Der Port 80 wird von cherrypy verwendet, um die Nachrichten von TTN per WebPush zu empfangen

Die passende Software liegt unter [https://gitlab.com/oberlab/p\\_messi.git](https://gitlab.com/oberlab/p_messi.git)



# VM 112 - Proxmox Backup Server

## Technische details

CPU	4
RAM	4/8G
HDD	12G + 200G
OS	Debian ?
<u>IP</u>	192.168.0.248/24
Domain	(pbs.oberlab.de)
Ports	8007

## Dienste

Auf dem Container läuft proxmox backup server. Dessen Web-Oberfläche ist über den Port 8007 erreichbar

Nach Außen ist der Port 3000 von Grafana erreichbar. Angesteuert wird der Port vom Caddy Container, der die Umsetzung nach HTTPS vornimmt.

## Zweite Platte

Die zweite Platte beinhaltet die Backups. Sie ist unter `/mnt/datastore/oberlab` gemountet



# VM 114 - Project services

## Technische details

CPU	4
RAM	1-4G
HDD	32G
OS	Debian 12 Bookworm
IP	192.168.0.242/24
Domain	
Ports	3000, 3001, 8000, 8086, 8087

## Dienste

## Connecting peaks

Ports 3001 ([Grafana](#)) & 8087 ([Influxdb](#)). Ferner 2x [Telegraf](#) für die Abholung der Daten von [TTN](#).

Angesteuert werden die Ports vom [Caddy Container](#), der die Umsetzung nach HTTPS vornimmt.

## Bike Sensor

Port 8000 (Web-Oberfläche)

Angesteuert wird der Port vom [Caddy Container](#), der die Umsetzung nach HTTPS vornimmt.

1 systemd Service für die Web-Oberfläche

## Messis im Oberland



Ports 3000 (Grafana) für das Dashboard & 8086 (Influxdb)

Angesteuert werden die Ports vom Caddy Container, der die Umsetzung nach HTTPS vornimmt.

Benötigte Packages: `apt install python3-cherrypy3 python3-influxdb`

1 Systemd Service für den Empfang der Daten von TTN



# VM 116 - Internal Services

## Technische details

CPU	2
RAM	1-2G
HDD	12G
OS	Debian 12 Bookworm
IP	192.168.168.15/24
Domain	
Ports	80, 9000

## Dienste

### Vaultwarden

Auf dem Container läuft Vaultwarden, eine freie Implementierung und ressourcenschonende Variante von Bitwarden als Docker-Container. Vaultwarden implementiert dabei die API eines Bitwarden-Servers und ist damit mit allen Bitwarden Clients kompatibel

Nach Außen ist der Port 80 erreichbar. Angesteuert wird der Port vom Caddy Container, der die Umsetzung nach HTTPS vornimmt.

Vaultwarden benötigt zwingend eine HTTPS Verbindung, ansonsten kommt nur eine Seite mit einer Fehlermeldung ("Please access this site over HTTPS")

### Listmonk



Auf dem Container läuft ListMonk, eine freie Mailinglisten Verwaltungssoftware als Docker-Container.

Nach Außen ist der Port 9000 erreichbar. Angesteuert wird der Port vom Caddy Container, der die Umsetzung nach HTTPS vornimmt.



# VM 117 - Oberlogin

## Technische details

CPU	4
RAM	2G
HDD	8G
OS	Debian 11
<u>IP</u>	192.168.0.247/24
Domain	oberlogin.oberlab.de
Ports	8080

## Dienste

Auf dem Container läuft Keycloak, eine User Management Software, die für das SSO zuständig ist.

Nach Außen ist der Port 8080 der Standard-Weboberfläche erreichbar. Angesteuert wird der Port vom Caddy Container, der die Umsetzung nach HTTPS vornimmt.



# VM 118 - Obercloud

## Technische details

CPU	8
RAM	8G
HDD	16G+100G
OS	Debian 11
<u>IP</u>	192.168.168.10/24
Domain	files.oberlab.de docs.oberlab.de
Ports	80 9980

## Dienste

Auf dem Container läuft Nextcloud, eine Cloud Software. Passend dazu ebenfalls Collabora Office um

Nach Außen sind die Ports 80 und 9980 der Standard-Weboberfläche erreichbar. Angesteuert wird der Port vom Caddy Container, der die Umsetzung nach HTTPS vornimmt.



# VM 119 - old Wiki

## Technische details

CPU	4
RAM	2/3G
HDD	8G+8G
OS	Debian 11
<u>IP</u>	192.168.168.11/24
Domain	wikiold.oberlab.de
Ports	3000

## Dienste

Auf dem Container läuft Wiki.js, eine Wiki Software.

Nach Außen ist der Port 3000 der Standard-Weboberfläche erreichbar. Angesteuert wird der Port vom Caddy Container, der die Umsetzung nach HTTPS vornimmt.



# VM 120 - Wiki

## Technische details

CPU	4
RAM	4/4G
HDD	8G
OS	Debian 12
<u>IP</u>	192.168.168.13/24
Domain	wiki.oberlab.de
Ports	80

## Dienste

Auf dem Container läuft BookStack, eine Wiki Software.

Nach Außen ist der Port 80 der Standard-Weboberfläche erreichbar. Angesteuert wird der Port vom Caddy Container, der die Umsetzung nach HTTPS vornimmt.



# VM 121 - Paperless

## Technische details

CPU	8
RAM	3/6G
HDD	16G+8G
OS	Debian 12
<u>IP</u>	192.168.0.9/24 192.168.168.14/24
Domain	paperless.oberlab.de
Ports	80

## Dienste

Auf dem Container läuft Paperless-ngx, ein Dokument Mangement System.

Ferner läuft ein Samba-Server auf Ports 139/445, womit ein Upload von Dokumenten (zB direkt aus dem Scanner/Drucker im Lab) möglich ist.

Nach Außen ist der Port 80 der Standard-Weboberfläche erreichbar. Angesteuert wird der Port vom Caddy Container, der die Umsetzung nach HTTPS vornimmt.



# Server



# Administration

## Remote Zugriff

Geht über [Tailscale](#). Damit muss kein dedizierter Port nach Außen freigegeben werden. Der Hauptserver hat die IP 100.91.45.38 (siehe [Obermox](#))

Zugriff auf die Container/VMs geht auch remote über SSH wenn man den Proxmox Hauptserver als Jumphost verwendet:

```
ssh -J root@100.91.45.38 root@host
```

Ebenso kann man Dateien direkt auf die Container/VMs übertragen wenn man den Proxmox Hauptserver als Jumphost verwendet:

```
scp -rp -J root@100.91.45.38 file root@host:/path
```

## Backups

Täglich um 3:00 werden alle Container und VMs eingefroren, ein Backup gezogen, und wieder aufgetaut. Die Konfiguration des Backups erfolgt in Proxmox. Es kommt Proxmox Backup Server (PBS) zum Einsatz.

Die Backups werden lokal auf der dedizierten 750GB SATA Festplatte im Server gespiegelt.

Remote-Kopien werden zusätzlich täglich um 5:00 remote zu Joel's Server daheim gespiegelt.

## DynDNS

Per [ddclient](#) wird die IP Adresse des Internet-Anschlusses überwacht, und bei Bedarf der DNS-Eintrag für gmund.oberlab.de aktualisiert. Dieser Eintrag hat eine TTL von 5 Minuten, das wäre die Zeit wo ggf noch die alte IP Adresse vom DNS Server zurückgeliefert würde.

Ein evtl. DynDNS Update seitens der Kabelbox ist davon komplett unabhängig.



# Installation / Konfiguration

Die meisten Aufgaben nach der Grundinstallation von Proxmox (mittels ISO Image der Distribution) werden über Ansible-Skripte gesteuert. Sie sind im Gitlab abgelegt.

Die Skripte können ebenfalls eine Grundkonfiguration einzelner Container oder VMs vornehmen.

Ferner sind im Gitlab die Docker Konfigurationsdaten für die verschiedenen Services abgelegt, die in den Container/VMs betrieben werden.

Weitere Anpassungen bleiben unabhängig davon möglich. Man sollte allerdings vermeiden Dateien zu ändern, die von Ansible verwaltet werden - sie werden beim nächsten Lauf zurückgesetzt !



# IP Adressen

## Bereich 192.168.0.x - routebare Adressen

**!!! IP Adressen 192.168.0.10-212 sind für DHCP reserviert, 192.168.0.1-10 für Infrastruktur !!! - siehe Kabelbox**

IP	Container	Beschreibung	Im Internet sichtbar
192.168.0.254	-	Obermox	
192.168.0.253	<u>CT 102</u>	Caddy	Ja
192.168.0.252			
192.168.0.251	<u>VM 104</u>	MQTT	Ja
192.168.0.250	<u>VM 100</u>	Home Assistant	Ja
192.168.0.249	<u>VM 106</u>	Minecraft	Ja
192.168.0.248	<u>VM 112</u>	Proxmox Backup Server	Nein
192.168.0.247	<u>VM 117</u>	Oberlogin - Keycloak	Ja
192.168.0.246	VM_113	Monitoring	Nein
192.168.0.245	VM_115	Packet Proxy	Nein



IP	Container	Beschreibung	Im Internet sichtbar
192.168.0.244	VM_102	grafana-influx	Ja
192.168.0.243	VM_110	Lama	Nein
192.168.0.242	<u>VM_114</u>	Project-Services (prj-svc)	Ja
192.168.0.221	-	Shelly Temperatursensor	Nein
192.168.0.220	-	Shelly Türkontakt	Nein
192.168.0.219	-	ShellyPlug-05	Nein
192.168.0.218	-	ShellyPlug-04	Nein
192.168.0.217	-	ShellyPlug-03	Nein
192.168.0.216	-	ShellyPlug-02	Nein
192.168.0.215	-	ShellyPlug-01	Nein
192.168.0.9	<u>VM_121</u>	Paperless Direktzugang	Nein

## Bereich 192.168.168.x - Adressen für interne Dienste

IP	Container	Beschreibung	Im Internet sichtbar (über Caddy)
192.168.168.1	-	Obermox	
192.168.168.2	<u>CT_102</u>	Caddy	Ja
<del>192.168.168.3</del>	<u>CT_107</u>	<del>Listmonk</del>	Ja
<del>192.168.168.4</del>	<u>CT_108</u>	<del>p_trackhoki</del>	Ja



IP	Container	Beschreibung	Im Internet sichtbar (über Caddy)
<del>192.168.168.5</del>	<u>CT_109</u>	<del>p_connectingpeaks</del>	Ja
<del>192.168.168.6</del>	CT_110	p_bikesensor	
192.168.168.7	<u>CT_111</u>	p_bikesensor2	Ja
192.168.168.8	<u>VM_105</u>	Zamma Hoki	Ja
192.168.168.10	<u>VM_118</u>	Nextcloud	Ja
192.168.168.11	<u>VM_119</u>	Wiki	Ja
192.168.168.12	VM_113	Monitoring	Ja
192.168.168.13	VM_120	Wiki2	Ja
192.168.168.14	VM_121	Paperless	Ja
192.168.168.15	VM_116	Internal-Services (int-svc)	Ja
192.168.168.35	<u>Container 103 - Vaultwarden</u>	Vaultwarden	Ja



Server

# Hardware

## Server

DELL Poweredge T320

Bedienungsanleitung

## CPU

Intel(R) Xeon(R) CPU E5-2430 0 @ 2.20GHz

6 Kerne, 12 Threads

## RAM

36GB :

3x 8GB ECC RAM 10600 9-11-E2 (Micron)

3x 4GB ECC RAM 10600 9-10-E1 (Hynix)

## HDD

8-Port SAS RAID Controller + 2-Port SATA Controller

(+ 2 SD Kartenslots... war wohl bei VMWare das Mittel zum Booten, sollte man jetzt nicht mehr benutzen)

## Physische Platten

- Slot 0 : 600GB SAS 15k 2.5" DELL
- Slot 1 : 600GB SAS 15k 2.5" DELL
- Slot 2 : 600GB SAS 15k 2.5" DELL



- SATA0 : 750GB 2.5" Platte

# Logische Platten

- RAID5 1.2TB : Platten #0,1,2
- RAID0 120GB : Platte #7
- Standalone 750GB Platte

# Netzwerk

2x Gigabit Ports "1" und "2"

LAN auf Port "2"

Port "1" unbenutzt

# iDrac - integrated DELL Remote Admin Console

Dedizierter Netzwerk Port

Oberfläche nur über HTTPS verfügbar - mit unbekannten DELL SSL Zertifikat -> Ausnahme im Browser bestätigen



Server

# Software

## Intro

Auf dem Server läuft die Virtualisierungssoftware Proxmox. Sie basiert auf Debian Linux, und kann sowohl Virtuelle Maschinen (VMs) als auch LXC Container verwalten. Damit lassen sich getrennte Dienste betreiben, und beliebige Test-Umgebungen aufsetzen.

## Netzwerk

IP 192.168.0.254/24

Gateway 192.168.0.1

DNS 192.168.0.1

## Proxmox Web-Oberfläche

Im Lab : <https://192.168.0.254:8006>

Über Tailscale : <https://100.91.45.38:8006>

## Tailscale

Tailscale ist eine VPN Lösung um remote auf den Server zugreifen zu können

Der Server hat die IP 100.91.45.38

Nur Mitglieder des Tailscale Rings des Oberlabs und eingeladene Gäste können auf die IP Adresse zugreifen



Server

# Proxmox

## Login

Erfolgt über Login/Passwort + **TOTP** (liegen im Vaultwarden)

## Netze

### vmbr0

Ist mit der Netzwerkschnittstelle eno2 (Port 2 am Case) verbunden.

IP 192.168.0.254/24

### vmbr168

Ist eine virtuelle Schnittstelle innerhalb vom Server, für Dienste die keine direkte Verbindung nach Außen haben und somit abgeschottet sind.

IP 192.168.168.1/24

### eno1

Nicht verwendet, ist aber vor-konfiguriert

IP 192.168.24.240/24

## Storage



# local

Root-Filesystem von Proxmox auf dem RAID5 Cluster → /var/lib/vz

ISO-Images und Container-Templates

Backups (über gemountete 750G Festplatte in /etc/fstab - transparent für Proxmox) in /var/lib/vz/dump

# local-lvm

LVM-Thin auf dem RAID5-Cluster

Storage für die VMs und Container

# storage2

140G Platte aus dem Raid0

Unbenutzt (wird ggf gelöscht/ersetzt)