

Vaultwarden (Bitwarden) installieren

Vorgehensweise

[Github AndresHardware](#)

Alternative implementation of the Bitwarden server API written in Rust and compatible with upstream Bitwarden clients*, perfect for self-hosted deployment where running the official resource-heavy service might not be ideal.

☐ Note: This project was known as Bitwarden_RS and has been renamed to separate itself from the official Bitwarden server in the hopes of avoiding confusion and trademark/branding issues.

This project is not associated with the Bitwarden project nor Bitwarden, Inc.

⚠**IMPORTANT**⚠: When using this server, please report any bugs or suggestions to us directly (look at the bottom of this page for ways to get in touch), regardless of whatever clients you are using (mobile, desktop, browser...). DO NOT use the official support channels.

Host your own Vaultwarden with NGINX

Diese Repository zeigt dir, wie du deinen eigenen Vaultwarden Passwort Manager Server erstellen und konfigurieren kannst. Ich nutzte hierfür Proxmox, das ganze kann aber auch mit einem RaspberryPi oder vergleichbarer Hardware gemacht werden. Eine Ausführliche Anleitung findet du auf meinem [YouTube Kanal](#).

Features

Basically full implementation of Bitwarden API is provided including:

- Organizations support
- Attachments and Send
- Vault API support
- Serving the static files for Vault interface
- Website icons API
- Authenticator and U2F support
- YubiKey and Duo support
- Emergency Access

Vorbereitungen

Verwende ein aktuelles Debian System in einem Proxmox Container

```
apt-get update && apt-get upgrade -y && apt autoremove -y
apt install curl
apt install argon2

curl -sSL https://get.docker.com | sh
apt install docker-compose -y
```

Installation

```
mkdir docker
cd docker
mkdir Vaultwarden
mkdir Vaultwarden
nano docker-compose.yml
```

Admin-Token als Hash Wert erstellen

Nach dem generieren des Admin-Token kann dieser in die Docker-Compose Datei eingefügt werden, hierbei muss vor jedem "\$"- Zeichen ein weiteres "\$" ergänzt werden!

```
# Using the Bitwarden defaults
echo -n "MySecretPassword" | argon2 "$(openssl rand -base64 32)" -e -id -k 65540 -t 3 -p 4
```

Importieren der docker-compose.yml Datei

```
version: '3'
```

```
services:
```

```
  vaultwarden:
```

```
    restart: always
```

```
    container_name: vaultwarden
```

```
    image: vaultwarden/server:latest
```

```
    volumes:
```

```
      - ./vw-data:/data/
```

```
    ports:
```

```
      - 80:80
```

```
    environment:
```

```
      - SMTP_HOST=smtp.gmail.com
```

```
      - SMTP_FROM=yourmailaddress@gmail.com
```

```
      - SMTP_FROM_NAME=Vaultwarden
```

```
      - SMTP_SECURITY=force_tls
```

```
      - SMTP_PORT=465
```

```
      - SMTP_USERNAME=yourmailaddress@gmail.com
```

```
      - SMTP_PASSWORD=YourAppPasswort
```

```
      - SMTP_TIMEOUT=30
```

```
      - SMTP_AUTH_MECHANISM="Plain"
```

```
      - LOGIN_RATELIMIT_MAX_BURST=10
```

```
      - LOGIN_RATELIMIT_SECONDS=60
```

```
      - DOMAIN=yourDomain or IP-Adress
```

```
      - INVITATION_ORG_NAME=HomeVault
```

```
      - INVITATIONS_ALLOWED=true
```

```
      - ADMIN_TOKEN=YourArgon2 key or Passwort
```

```
      - SIGNUPS_ALLOWED=true
```

```
      - SIGNUPS_DOMAINS_WHITELIST=example.com,mydomain.net,myotherdomain.org
```

```
      - SIGNUPS_VERIFY=true
```

```
      - SIGNUPS_VERIFY_RESEND_TIME=3600
```

```
      - SIGNUPS_VERIFY_RESEND_LIMIT=6
```

```
      - EMERGENCY_ACCESS_ALLOWED=true
```

```
      - SENDS_ALLOWED=true
```

```
      - WEB_VAULT_ENABLED=true
```

In der yml-Datei SMTP Einstellungen anpassen. Die Domain eintragen und den AdminToken generieren.

Kopiere nun den Inhalt aus der docker-compose.yml.txt Datei in deine eben erstellte .yml datei und passe es entsprechend an.

Anschließend kann der Container ausgerollt werden.

```
docker-compose up -d
```

Domain über Proxy Manager mit der IP Adresse verknüpfen

Als nächstes die Domain mit der IP-Adresse des Vaultwarden verknüpfen und die SSL verschlüsselung aktivieren.

Update

Noch nicht getestet

Update über docker-compose

Im Ordner in der die docker-compose.yml datei liegt folgendes eingeben:

```
docker-compose down
```

danach

```
docker-compose up
```

Alte Version

Admin Zugang

Der Adminzugang ist erreichbar über domain.co/admin. Hier muss der Token eingegeben werden. Der ist da zu finden, wo er gespeichert ist.

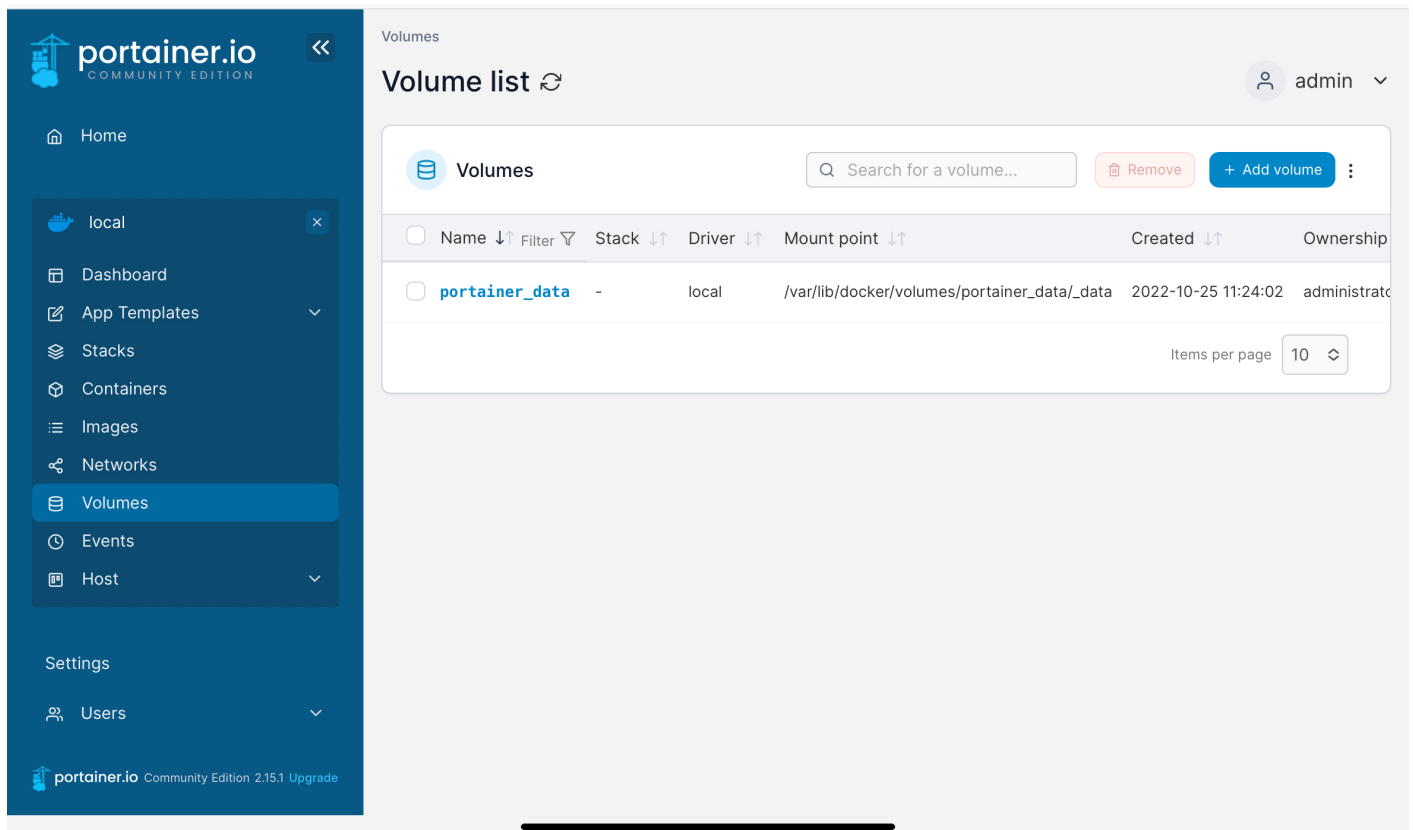
Bitwarden installieren

Protainer installieren

Bitwarden im Dockercontainer installieren mithilfe von Portainer

Volume

Volumes > Add volume



The screenshot shows the Portainer.io web interface. On the left is a dark blue sidebar with the Portainer.io logo and a navigation menu. The main content area is titled 'Volumes' and shows a 'Volume list' table. The table has columns for Name, Stack, Driver, Mount point, Created, and Ownership. One volume named 'portainer_data' is listed with a local driver and a mount point of '/var/lib/docker/volumes/portainer_data/_data'. The interface also includes a search bar, a 'Remove' button, and an '+ Add volume' button.

Name	Stack	Driver	Mount point	Created	Ownership
portainer_data	-	local	/var/lib/docker/volumes/portainer_data/_data	2022-10-25 11:24:02	administrat

Name: vaultwarden

Container

Add Container

- Name: Vaultwarden
- image: vaultwarden/server:latest

+publish a new network port

host: 8080 container: 80

Advanced Container settings

Volume

+map additional volume

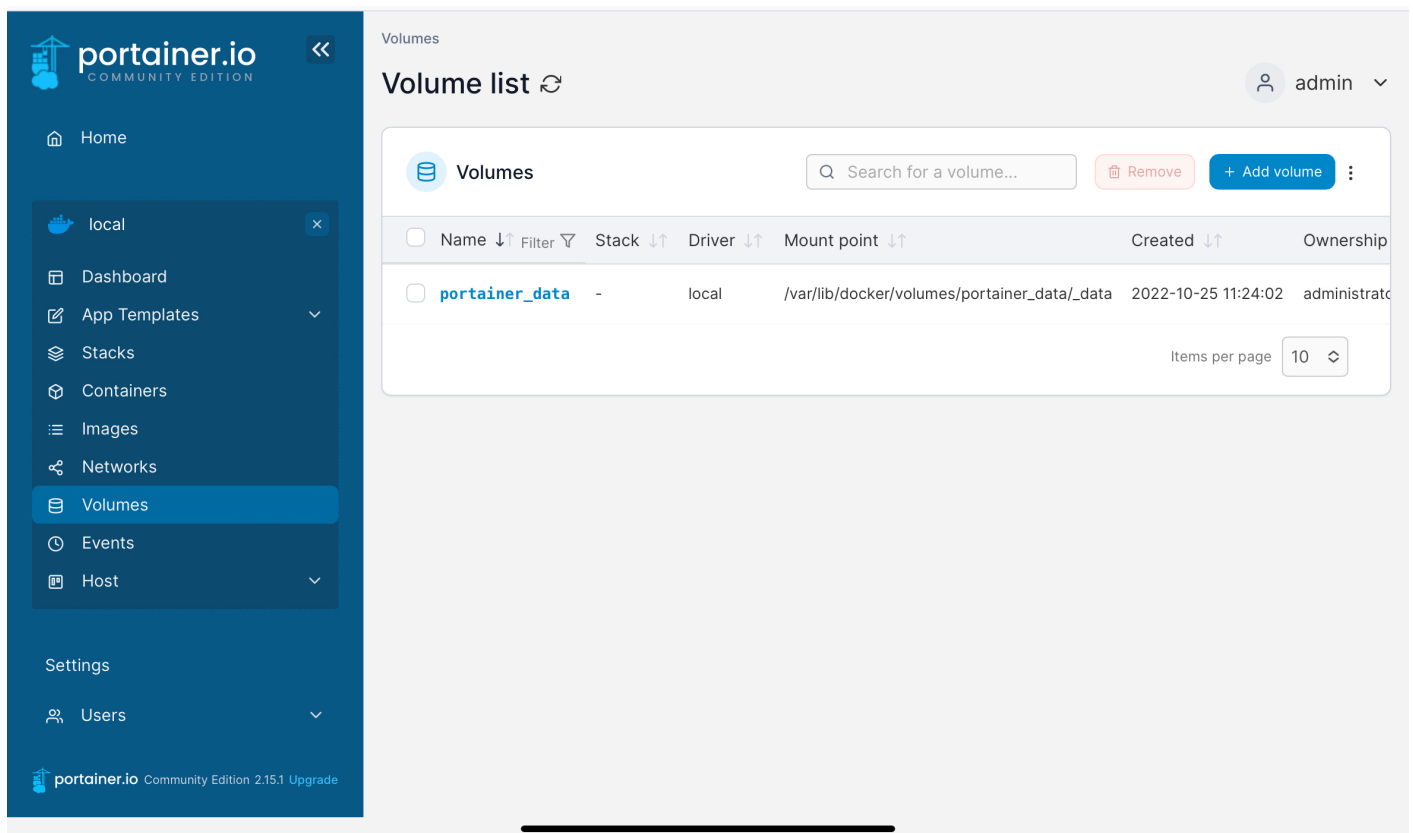
Container:

Select a Volume (vorher erstelltes Volume):

Restart policy

Always

Deploy the Container



The screenshot shows the Portainer.io web interface. On the left is a dark blue sidebar with the Portainer.io logo and a navigation menu. The main content area is titled 'Volumes' and 'Volume list'. It features a search bar, a 'Remove' button, and an '+ Add volume' button. Below this is a table with columns for Name, Stack, Driver, Mount point, Created, and Ownership. One volume is listed: 'portainer_data' with a local driver and a mount point of '/var/lib/docker/volumes/portainer_data/_data'. The interface is for a user named 'admin'.

Name	Stack	Driver	Mount point	Created	Ownership
portainer_data	-	local	/var/lib/docker/volumes/portainer_data/_data	2022-10-25 11:24:02	administrat

Nginx einrichten

Um immer den aktuellsten Vaultwarden zu verwenden am besten Watchtower einrichten

Admin-Token

der wird ebenfalls in der Config-Json Datei gespeichert

Version #3

Erstellt: 25 October 2022 09:28:12 von hermann

Zuletzt aktualisiert: 25 October 2022 11:02:23 von hermann

Revision #9

Created 20 March 2023 21:41:51 by Hermann

Updated 11 November 2025 06:36:34 by Hermann