

Sammlung Codes und Tricks

- [iptables](#)
- [chmod](#)
- [In History suchen mit Bild nach oben und Bild nach unten](#)
- [Vaultwarden Admin Passwort setzen](#)
- [Websocket in Nginx Proxy Manager](#)
- [Tasmota Rules / Regeln](#)
- [Raspberry Pi 5 Lüfter Steuerung](#)
- [Hetzner /etc/apt/sources.list](#)
- [Nvidia Treiber unter Ubuntu 24.01 installieren](#)
- [Anleitung: Anaconda Installation](#)
- [git clone verzeichnis](#)

iptables

Eintrag löschen nach Nummer in der Chain

```
sudo iptables -L --line-numbers
```

Output:

Chain INPUT (policy DROP)

num	target	prot opt	source	destination	
1	ACCEPT	all --	anywhere	anywhere	ctstate RELATED,ESTABLISHED
2	ACCEPT	all --	anywhere	anywhere	
3	DROP	all --	anywhere	anywhere	ctstate INVALID
4	UDP	udp --	anywhere	anywhere	ctstate NEW
5	TCP	tcp --	anywhere	anywhere	tcp flags:FIN,SYN,RST,ACK/SYN ctstate NEW
6	ICMP	icmp --	anywhere	anywhere	ctstate NEW
7	REJECT	udp --	anywhere	anywhere	reject-with icmp-port-unreachable
8	REJECT	tcp --	anywhere	anywhere	reject-with tcp-reset
9	REJECT	all --	anywhere	anywhere	reject-with icmp-proto-unreachable
10	ACCEPT	tcp --	anywhere	anywhere	tcp dpt:ssh ctstate NEW,ESTABLISHED

```
sudo iptables -D INPUT 3
```

chmod

Wert	Rechte
0	Keine
1	x
2	w
3	w+x
4	r
5	r+x
6	r+w
7	r+w+x

In History suchen mit Bild nach oben und Bild nach unten

Hierzu muss in der Datei `/etc/inputrc` folgender Inhalt aktiviert werden unter Linux:

```
# alternate mappings for "page up" and "page down" to search the history
"\e[5~": history-search-backward
"\e[6~": history-search-forward
```

Ist die Aktivierung unter `/etc/inputrc` nicht möglich, kann im Homeverzeichnis des Users folgende Datei angelegt werden `~/.inputrc` mit dem gleichen Inhalt wie oben angegeben.

Möchte man erreichen, dass sogar bei parallel geöffneten Tabs bzw. Fenstern die History aus allen Tabs nach jedem Befehl gespeichert und dadurch jedem bereits geöffneten anderen Tab sofort zur Verfügung gestellt wird, ohne dafür einen neuen Tab öffnen zu müssen, trägt man stattdessen folgende Zeile ein in `~/.bashrc` :

```
PROMPT_COMMAND="history -a; history -c; history -r"
```

Vaultwarden Admin Passwort setzen



Bei der Vaultwarden Docker Installation mit docker-compose erhält man einen Fehler wenn folgender Eintrag gemacht wird.

Die Variable "ADMIN_TOKEN" wird nicht richtig als String gelesen und Vaultwarden ist über "./admin" nicht erreichbar oder das Passwort wird als falsch erkannt.

```
services:
  bitwarden:
    image: vaultwarden/server:latest
    container_name: vaultwarden
    restart: always
    ports:
      - 8060:80
      - 3012:3012
    volumes:
      - bw2-data:/data
      - /etc/localtime:/etc/localtime:ro
      - /etc/timezone:/etc/timezone:ro
      - bw2-data:/log
      # - LOG_FILE=/home/joachim/docker/bitwarden/log/bitwarden.log
    environment:
      - WEBSOCKET_ENABLED=true
      - SIGNUPS_ALLOWED=true
      - SIGNUPS_VERIFY=true
      #- ADMIN_TOKEN:
"$argon2id$v=19$m=65540,t=3,p=4$u/fa7M86QrbBzN6eZjYOE5s5EhLXQvYcuwBHTE1/gIE$23stGPT3z5CKQcIPG
```

```
H/p77VsvMI9ZOwkk8ViP1JNq"
```

- ADMIN_TOKEN=\${ADMIN_TOKEN}
- INVITATIONS_ALLOWED=true
- SHOW_PASSWORD_HINT=false
- DATABASE_URL=mysql://vaultwarden:xxxxxxx@192.168.XXX.XXX:3306/vaultwarden2

volumes:

bw2-data:

Abhilfe :

Erstelle im Installationsverzeichnis von Vaultwarden eine Datei ".env" mit folgendem Inhalt und ersetze in der "docker-compose.yml" die Variable "- ADMIN_TOKEN=\${ADMIN_TOKEN}" :

```
ADMIN_TOKEN="$argon2id$v=19$m=65540,t=3,p=4$u/fA7M86QrbBzN6eZjYOE5s5EhLXQvYcuwBHTE1/gIE$23s  
tGPt3z5CKQcIPGH/p77VsvMI9ZOwkk8ViP1JNq"
```

Der Token wird durch folgenden Befehl erstellt :

```
docker exec -it vaultwarden /vaultwarden hash
```

Websocket in Nginx Proxy Manager

Define location *

location

/notifications/hub

**Scheme ***

http

Forward Hostname / IP *

192.168.10.155/

Forward Port *

3012



Add a path for sub-folder forwarding.

Example: 203.0.113.25/path/

[Delete](#)**Define location ***

location

/notification/hub/negotiate

**Scheme ***

http

Forward Hostname / IP *

192.168.10.155/

Forward Port *

8060



Add a path for sub-folder forwarding.

Example: 203.0.113.25/path/

[Delete](#)[Cancel](#)[Save](#)

Einrichtung Websocket für

Vaultwarden:

location: /notifications/hub

scheme: http

forward-hostname/ip: IPvonVaultwarden

port: 3012

zahnradmenü: proxy_set_header Upgrade \$http_upgrade;
proxy_set_header Connection "upgrade";

location: /notification/hub/negotiate

scheme: http

forward-hostname/ip: IPvonVaultwarden

port: 8080

Tasmota Rules / Regeln

```
rule1 on Time#Minute|1 DO ping4 [IP-Adresse] ENDON
```

```
rule1 on Time#Minute|1 DO ping4 [IP-Adresse] ENDON ON ping#[IP-Adresse]#Success=4 DO tmsend  
Netzwerkverbindung von NAS-Speicher OK! ENDON
```

```
rule1 on Time#Minute|1 DO ping4 [IP-Adresse] ENDON ON ping#[IP-Adresse]#Success=0 DO tmsend  
Netzwerkverbindung von NAS-Speicher unterbrochen! ENDON
```

```
rule1 on Time#Minute|1 DO BACKLOG ping4 [IP-Adresse1]; ping4 [IP-Adresse2]; ping4 [IP-Adresse3] ENDON ON  
ping#[IP-Adresse1]#Success=4 DO tmsend Netzwerkverbindung [dein Gerät] ok! ENDON ON ping#[IP-  
Adresse2]#Success=4 DO tmsend Netzwerkverbindung [dein Gerät] ok! ENDON ON ping#[IP-  
Adresse3]#Success=4 DO tmsend Netzwerkverbindung [dein Gerät] ok! ENDON
```

```
rule1 on Time#Minute|1 DO BACKLOG ping4 [IP-Adresse1]; ping4 [IP-Adresse2]; ping4 [IP-Adresse3] ENDON ON  
ping#[IP-Adresse1]#Success=0 DO tmsend Netzwerkverbindung [dein Gerät] unterbrochen! ENDON ON  
ping#[IP-Adresse2]#Success=0 DO tmsend Netzwerkverbindung [dein Gerät] unterbrochen! ENDON ON  
ping#[IP-Adresse3]#Success=0 DO tmsend Netzwerkverbindung [dein Gerät] unterbrochen! ENDON
```

Raspberry Pi 5 Lüfter Steuerung

In der Datei `/boot/firmware/config.txt` müssen folgende Parameter eingetragen werden. Die Beschreibung sollte selbsterklärend sein.

[all]

Bei Temperatur 35 Grad Celsius Lüfter auf 0-250 / 100 setzen

dtparam=fan_temp0=35000

dtparam=fan_temp0_hyst=5000

dtparam=fan_temp0_speed=100

Bei Temperatur 60 Grad Celsius Lüfter auf 0-250 / 125 setzen = Halber Speed

dtparam=fan_temp1=60000

dtparam=fan_temp1_hyst=5000

dtparam=fan_temp1_speed=125

Bei Temperatur 67,5 Grad Celsius Lüfter auf 0-250 / 175 setzen = Dreiviertel Speed

dtparam=fan_temp2=67500

dtparam=fan_temp2_hyst=5000

dtparam=fan_temp2_speed=175

Bei Temperatur 75 Grad Celsius Lüfter auf 0-250 / 250 setzen = Full Speed

dtparam=fan_temp3=75000

dtparam=fan_temp3_hyst=5000

dtparam=fan_temp3_speed=250

Hetzner /etc/apt/sources.list

Es gibt auf verschiedenen Hetzner Server Fehler mit "apt update" und "apt upgrade". Ein "--fix-missing" ist leider nicht erfolgreich.

Lösung ARM64 ist nicht eingetragen und muss auf anderen Mirror umgestellt werden:

```
deb https://mirror.hetzner.com/ubuntu-ports/packages jammy main restricted
deb https://mirror.hetzner.com/ubuntu-ports/packages jammy-updates main restricted
deb https://mirror.hetzner.com/ubuntu-ports/packages jammy universe
deb https://mirror.hetzner.com/ubuntu-ports/packages jammy-updates universe
deb https://mirror.hetzner.com/ubuntu-ports/packages jammy multiverse
deb https://mirror.hetzner.com/ubuntu-ports/packages jammy-updates multiverse
deb https://mirror.hetzner.com/ubuntu-ports/packages jammy-backports main restricted universe multiverse
```

dpkg --print-architecture. zeigt die Architektur an.

Nvidia Treiber unter Ubuntu 24.01 installieren

Add Nvidia PPA

Ensure your system has the latest packages

```
sudo apt update && sudo apt upgrade -y
```

Add the Nvidia PPA

```
sudo add-apt-repository ppa:graphics-drivers/ppa
```

When prompted hit `ENTER` to continue. Pull the latest packages

```
sudo apt update
```

Check which drivers are available. I usually go with the 'recommended' version

```
ubuntu-drivers devices
```

Install Drivers

Once you identify the driver you wish to install, use `apt` to install it

```
sudo apt install nvidia-driver-<DRIVER VERSION>
```

Reboot for the changes to take effect

```
sudo reboot now
```

Confirm Installation

Check `nvidia-smi` to see if the GPU is detected and the correct driver version is installed

```
nvidia-smi
```

I have the `nvidia-driver-560` installed at the time of this guide, and my output reflects that

Anleitung: Anaconda Installation

Anleitung: Anaconda installieren und nutzen

Anaconda ist eine Open-Source-Distribution, die vor allem für Data Science, maschinelles Lernen und wissenschaftliches Computing verwendet wird. Sie vereinfacht die Verwaltung von Python- und R-Paketen und Umgebungen.

Was ist Anaconda und wofür ist es gut?

1. Python-Umgebungen:

- Anaconda erleichtert die Verwaltung von Python-Versionen und -Bibliotheken, sodass Projekte mit unterschiedlichen Anforderungen problemlos parallel laufen können.

2. Pakete und Tools:

- Es enthält viele nützliche Tools wie **Jupyter Notebook**, **Spyder** und **conda** (Package- und Environment-Manager).
- Beliebte Bibliotheken wie NumPy, pandas, TensorFlow und Matplotlib sind vorkonfiguriert.

3. Vorteile:

- Kein Konflikt zwischen verschiedenen Bibliotheken.
 - Ideal für Data-Science-Projekte.
 - Erleichtert die Zusammenarbeit, da man leicht reproduzierbare Umgebungen erstellen kann.
-



ANACONDA[®]

Anleitung: Installation von Anaconda

Schritt 1: Anaconda herunterladen

1. Gehe auf die [offizielle Anaconda-Website](#).
2. Lade die Installationsdatei für dein Betriebssystem herunter (Windows, macOS oder Linux).

Schritt 2: Installation unter Windows

1. Starte den Installer.
2. Folge den Anweisungen:
 - Wähle die Option "Just Me" für eine Einzelbenutzerinstallation.
 - Empfohlen: Installiere Anaconda für deinen Benutzer (kein Admin-Zugang nötig).
 - Lass den Anaconda-Installer den Pfad setzen.
3. Aktiviere die Option: "Add Anaconda3 to my PATH environment variable" **nur**, wenn du sicher bist, dass keine andere Python-Installation Probleme verursacht.

Schritt 3: Installation unter macOS und Linux

1. Öffne ein Terminal.
2. Führe den heruntergeladenen Installer mit:

```
bash Anaconda3-*.sh
```

3. Folge den Installationsanweisungen im Terminal.

Schritt 4: Installation überprüfen

Nach der Installation kannst du prüfen, ob alles funktioniert:

```
conda --version
```

Dies sollte die installierte Version anzeigen.

Erste Schritte mit Anaconda

1. Anaconda Navigator starten

- **Windows/macOS:** Suche nach "Anaconda Navigator" und starte es.
- **Linux:** Gib im Terminal ein:

```
anaconda-navigator
```

Der Navigator bietet eine grafische Oberfläche, um Pakete zu installieren, Umgebungen zu verwalten und Tools wie Jupyter Notebook zu starten.

2. Umgebung erstellen

Du kannst separate Umgebungen für verschiedene Projekte erstellen:

```
conda create --name meine_umgebung python=3.9  
conda activate meine_umgebung
```

3. Pakete installieren

Füge Bibliotheken hinzu:

```
conda install numpy pandas matplotlib
```

4. Jupyter Notebook starten

```
jupyter notebook
```

Dies öffnet eine Weboberfläche, auf der du Python-Skripte interaktiv ausführen kannst.

Nützliche Befehle für die Arbeit mit Anaconda

Befehl	Beschreibung
<code>conda list</code>	Zeigt alle installierten Pakete in der Umgebung.
<code>conda update conda</code>	Aktualisiert den Conda-Paketmanager.
<code>conda deactivate</code>	Beendet die aktive Umgebung.
<code>conda env list</code>	Listet alle verfügbaren Umgebungen.
<code>conda remove --name meine_umgebung</code>	Löscht eine Umgebung.

Fazit

Anaconda ist ein leistungsstarkes Tool, das die Arbeit mit Python-Umgebungen erheblich erleichtert. Es ist besonders nützlich für Entwickler, die an Projekten im Bereich Data Science oder maschinelles Lernen arbeiten. Dank der integrierten Tools wie Jupyter Notebook und der einfachen Paketverwaltung ist es sowohl für Anfänger als auch für Experten geeignet.

git clone verzeichnis

Wenn ich nur ein bestimmtes Verzeichnis aus einem git repository clonen möchte sind folgende Schritte nötig:

```
git clone --no-checkout git@git.local:gitbenutzer/clone-directory.git
```

Sparse Checkout aktivieren: Aktiviere Sparse Checkout:

```
git sparse-checkout init
```

Unterverzeichnis auswählen: Gib das Unterverzeichnis an, das du herunterladen möchtest. Zum Beispiel, wenn das Unterverzeichnis `releases/` heißt:

```
git sparse-checkout set releases/1/
```

```
project/
├─ releases/
│   └─ file1.txt
│       └─ file2.txt
├─ anotherdir/
└─ README.md
```

Sparse Checkout Änderungen: Falls du später ein weiteres Verzeichnis hinzufügen möchtest, kannst du den Befehl `git sparse-checkout set` erneut ausführen:

```
git sparse-checkout set releases/1 releases/2
```