Installastion ClipCascade Server

Beschreibung:

ClipCascade ist ein Selfhost-Server mit dem sich eine Zwischenablage Synchronisieren lässt.

Vorrausetzungen:

Wir brauchen wieder eine Feste IP und einen DNS-Namen

Installation:

Server installation:

```
apt install docker.io docker-compose curl
```

Verzeichnis erstellen

```
mkdir /root/clipcascade
```

nun die .env Datei erstellen

```
nano /root/clipcascade/.env
```

Variablen Tabelle

For a comprehensive list of available environment variables, refer to the Advanced Details section.

Variable	Description	Default Value	Example

`CC_MAX_MESSAGE_SIZE_IN_MiB`	Defines the maximum allowed message size in MiB. Ignored if `CC_P2P_ENABLED` is set to `true`.	`1`	`3`
`CC_ALLOWED_ORIGINS`	Specifies the allowed CORS origins for secure cross-origin access.	`*`	`https://clipcascade.example.com`
`CC_P2P_ENABLED`	Enables or disables peer-to-peer mode. When enabled, `CC_MAX_MESSAGE_SIZE_IN_MiB` is ignored.	`false`	`true`
`CC_SIGNUP_ENABLED`	Allows or restricts user self-registration.	`false`	`false`
`CC_PORT`	Specifies the port on which the server listens for incoming connections.	`8080`	`1234`

Inhalt:

WICHTIG: CC_ALLOWED_ORIGINS ist auch die Public URL für Caddy also Pflicht

```
CC_MAX_MESSAGE_SIZE_IN_MiB=100
CC_ALLOWED_ORIGINS=https://clipcascade.example.com  # Defines allowed CORS origins for security
CC_SIGNUP_ENABLED=false  # Enables or disables user self-registration
CC_P2P_ENABLED=false
```

Nun die Composer Datei

```
nano /root/clipcascade/docker-compose.yml
```

Inhalt

```
services:
  clipcascade:
    image: sathvikrao/clipcascade:latest
    #ports:
    #  - "8080:8080"  # Expose the ClipCascade server on port 8080
    restart: always  # Automatically restart the container if it stops
    volumes:
      - ./cc_users:/database  # Persistent storage for user data
    environment:
      - CC_MAX_MESSAGE_SIZE_IN_MiB=${CC_MAX_MESSAGE_SIZE_IN_MiB}  # Maximum message size in MiB
(ignored if P2P mode is enabled)
```

```
    - CC_P2P_ENABLED=${CC_P2P_ENABLED}  # Enables or disables peer-to-peer(P2P) mode
    - CC_ALLOWED_ORIGINS=${CC_ALLOWED_ORIGINS}  # Defines allowed CORS origins for security
    - CC_SIGNUP_ENABLED=${CC_SIGNUP_ENABLED}  # Enables or disables user self-registration


  caddy:
    image: caddy:latest
    restart: always
    ports:
      - "80:80"
      - "443:443"
      - "8080:80"
    command: caddy reverse-proxy --from ${CC_ALLOWED_ORIGINS} --to clipcascade:8080
    volumes:
      - ./data/caddy/data:/data
      - ./data/caddy/config:/config
```

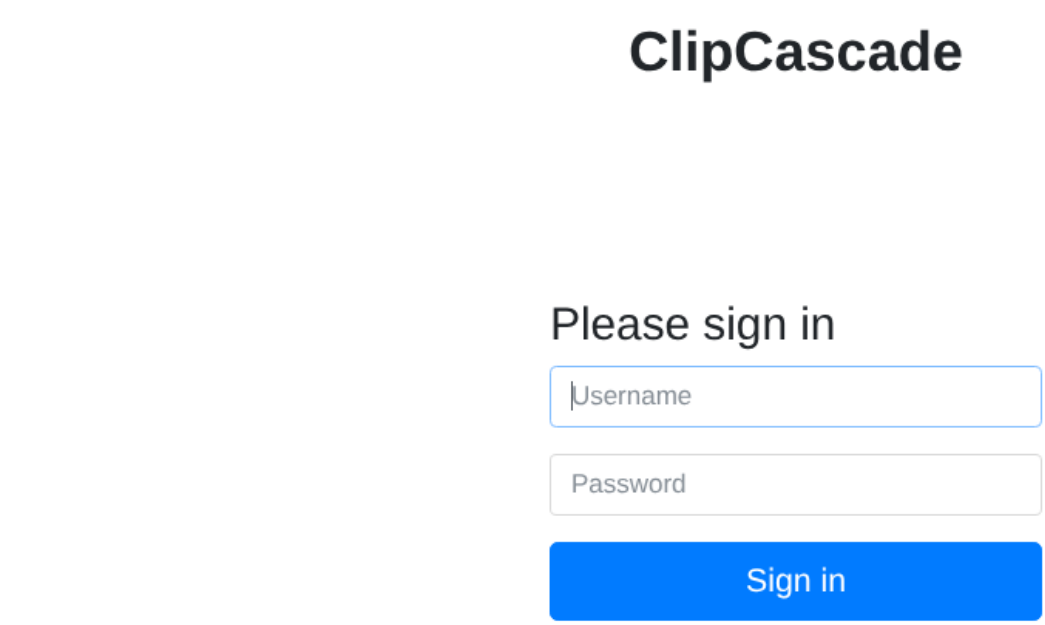
Nun den container starten

```
cd /root/clipcascade/
docker-compose up -d
```

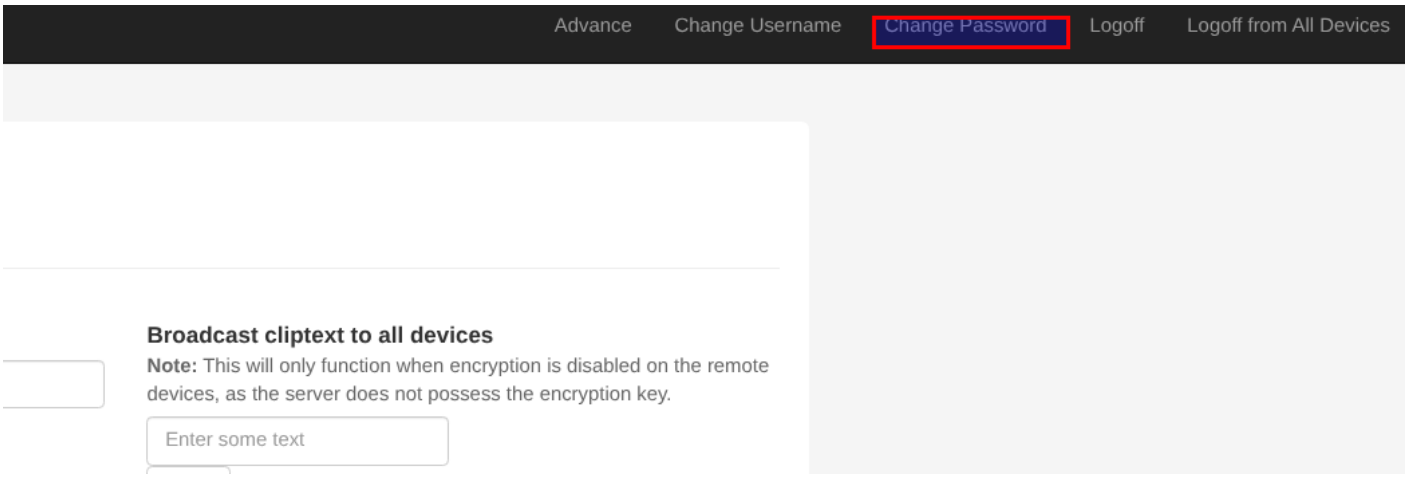
Konfiguration:

Kennwort ändern:

- **Default Credentials:**
  - **Username:** `admin`
  - **Password:** `admin123`

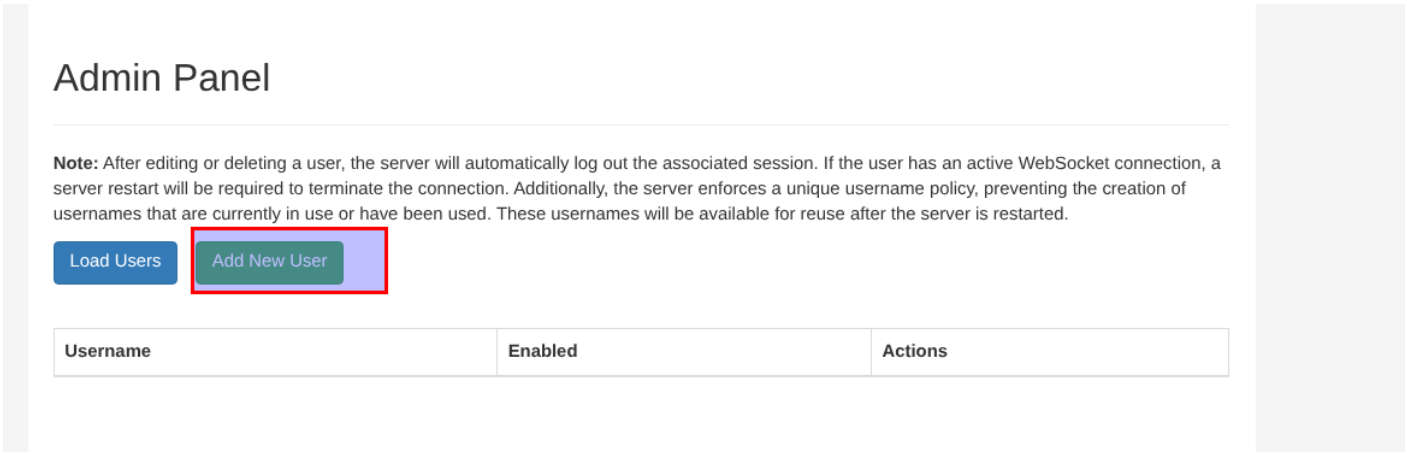


Nun das Kennwort ändern im Menü oben change Password

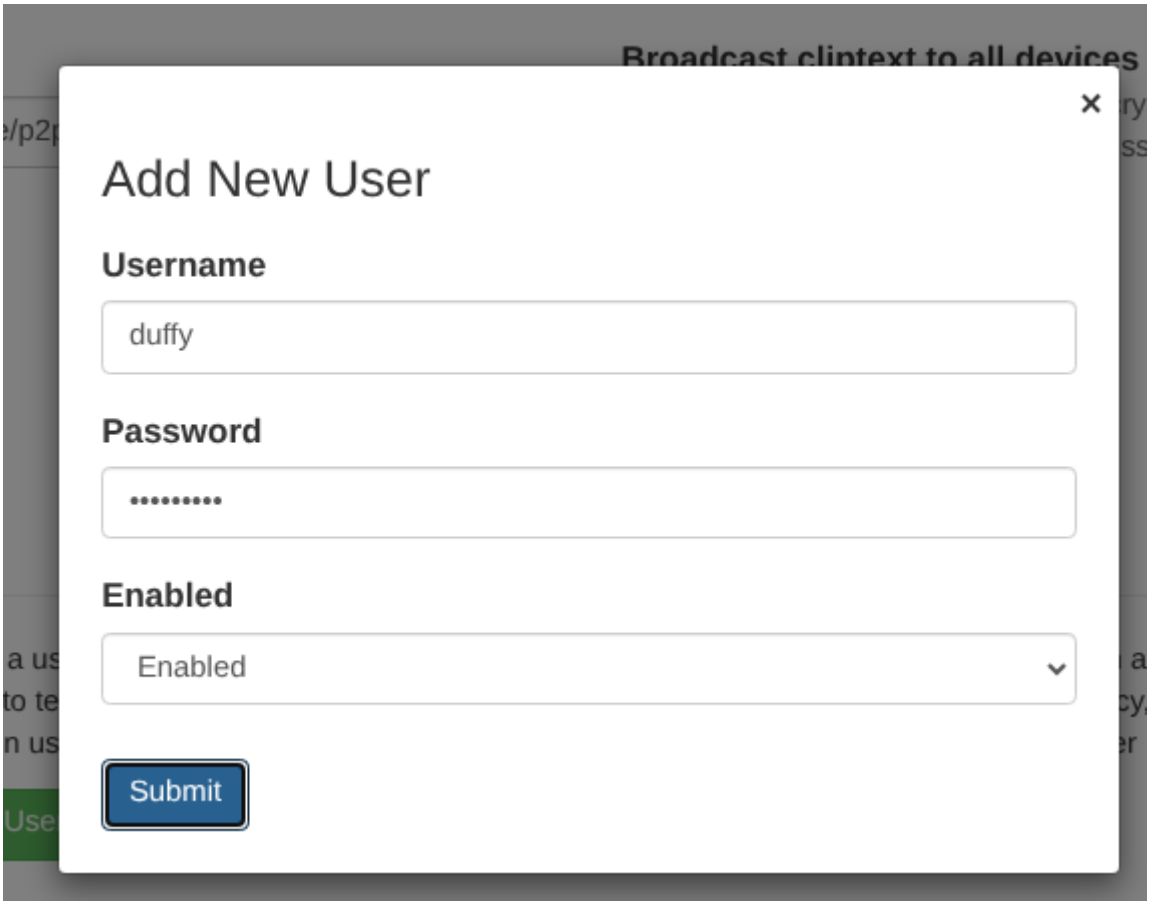


Benutzer anlegen:

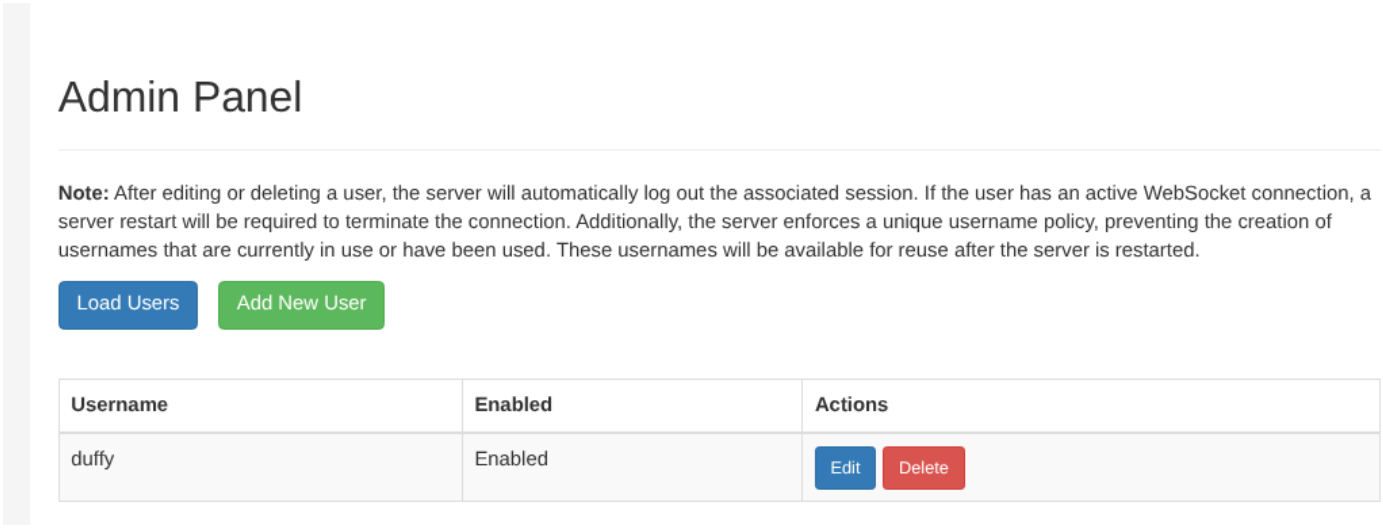
Dazu auf Add user klicken



Daten eingeben und auf submit klicken



Nun erscheint der Benutzer in der Liste



Nun noch die ufw Firewall regeln anpassen.

Da hier der Docker container sowieso von außen erreichbar sein soll, brauchen wir hier keine Anspassung für docker.
Aber ssh soll nur auf der lokalen Netzwerkkarte zur Verfügung stehen

ufw installieren

```
apt install ufw
```

Nun die Regeln setzten, darauf achten das enp6s18 die interne Karte ist ansonsten anpassen

```
ufw allow in on enp6s18 to any port 22
ufw enable
```

Version #9
Erstellt: 12 Juli 2025 10:21:17 von Admin
Zuletzt aktualisiert: 15 Juli 2025 13:25:39 von Admin