

# Installation ClipCascade Server

## Beschreibung:

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

## Vorraussetzungen:

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
----------	-------------	---------------	---------

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

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:**

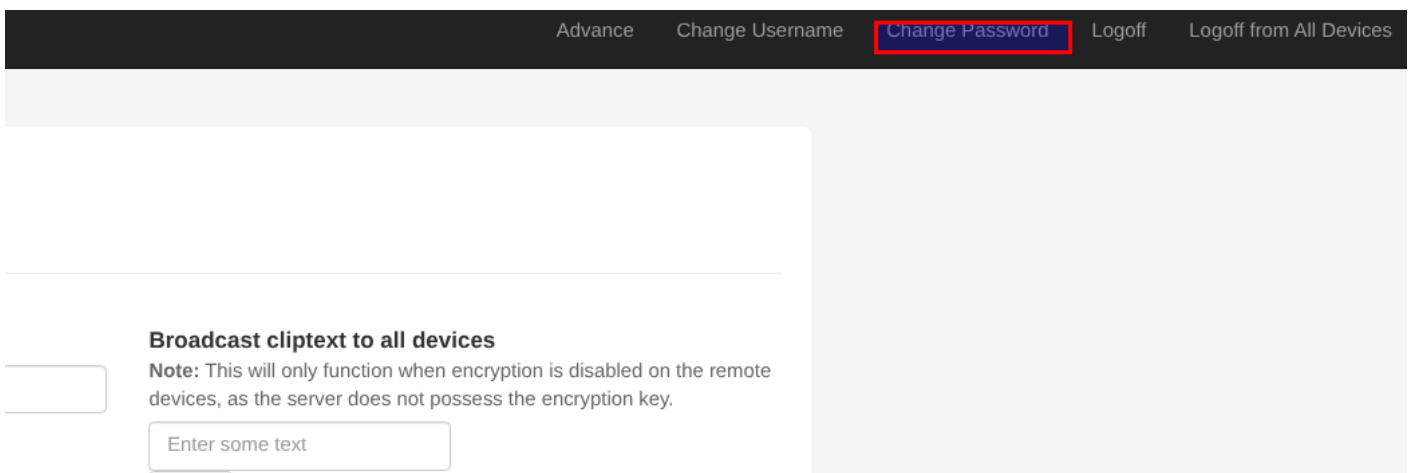
- **Password:**

# ClipCascade

Please sign in

Sign in

Nun das Kennwort ändern im Menü oben change Password



Advance Change Username **Change Password** Logoff Logoff from All Devices

**Broadcast cliptext to all devices**  
**Note:** This will only function when encryption is disabled on the remote devices, as the server does not possess the encryption key.

## Benutzer anlegen:

Dazu auf Add user klicken

### Admin Panel

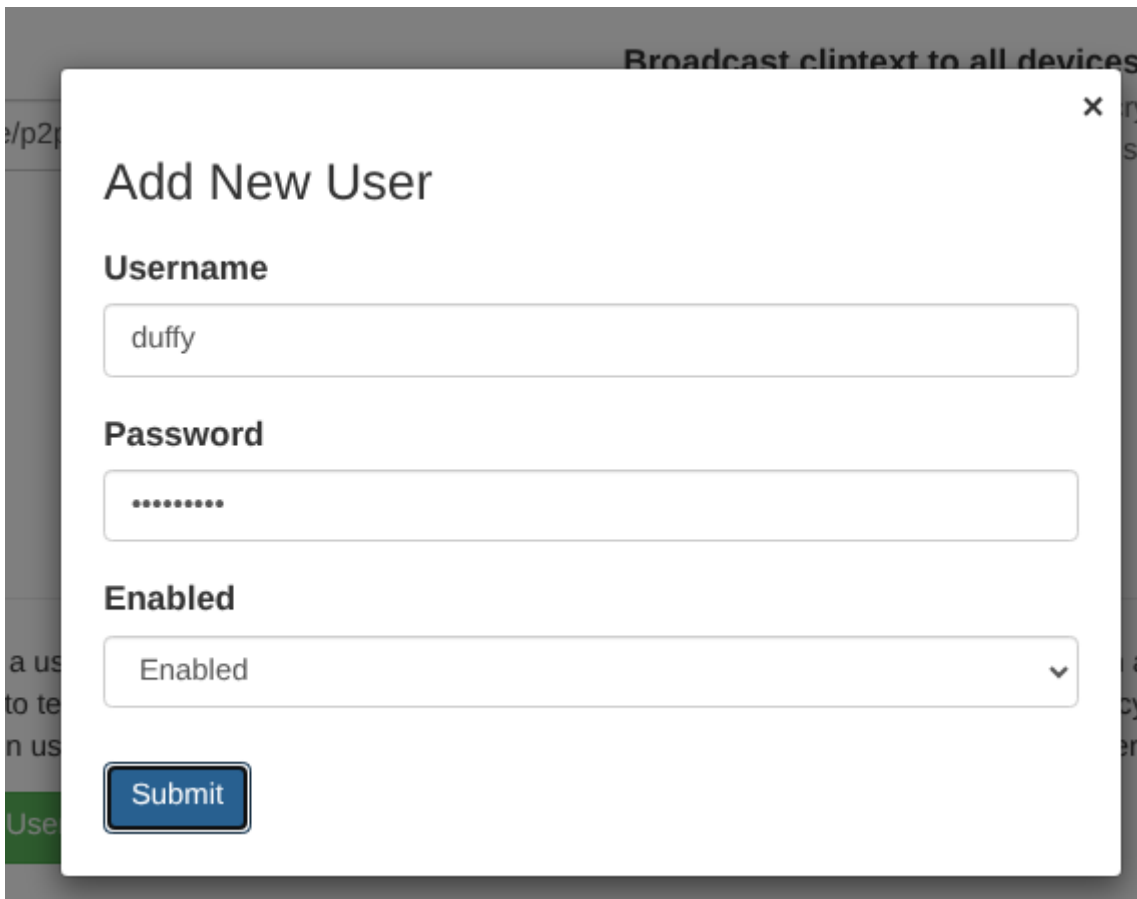
**Note:** After editing or deleting a user, the server will automatically log out the associated session. If the user has an active WebSocket connection, a server restart will be required to terminate the connection. Additionally, the server enforces a unique username policy, preventing the creation of usernames that are currently in use or have been used. These usernames will be available for reuse after the server is restarted.

Load Users

Add New User

Username	Enabled	Actions
----------	---------	---------

Daten eingeben und auf submit klicken



Broadcast clintext to all devices

### Add New User

**Username**

**Password**

**Enabled**

**Submit**

Nun erscheint der Benutzer in der Liste

## Admin Panel

**Note:** After editing or deleting a user, the server will automatically log out the associated session. If the user has an active WebSocket connection, a server restart will be required to terminate the connection. Additionally, the server enforces a unique username policy, preventing the creation of usernames that are currently in use or have been used. These usernames will be available for reuse after the server is restarted.

[Load Users](#) [Add New User](#)

Username	Enabled	Actions
duffy	Enabled	<a href="#">Edit</a> <a href="#">Delete</a>

Nun noch die ufw Firewall regeln anpassen.

Da hier der Docker container sowieso von außen erreichbar sein soll, brauchen wir hier keine Anpassung für docker.

Aber ssh soll nur auf der lokalen Netzwerkkarte zur Verfügung stehen

ufw installieren

```
apt install ufw
```

Nun die Regeln setzen, 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