

Installationsanleitung: War Command (V73.0)

Diese Anleitung führt dich durch die Installation auf einer **Synology NAS** mit Docker (Container Manager), inkl. externer Erreichbarkeit.

Schritt 1: Ordnerstruktur vorbereiten

Erstelle auf deinem NAS (z.B. via File Station oder SSH) einen neuen Ordner für das Projekt. Die Struktur muss **exakt** so aussehen:

```
/volume1/docker/war-command/    <-- Hauptordner
└── docker-compose.yml      <-- Konfiguration der Container
└── Dockerfile            <-- Bauanleitung für den Webserver
└── src/                  <-- Unterordner für den Code
    ├── index.html        <-- Die App-Oberfläche
    ├── api.php           <-- Die Logik (Backend)
    ├── db_setup.php       <-- Datenbank-Verbindung
    ├── map_builder3.html  <-- (Optional) Map Tool
    ├── bagger.html        <-- (Optional) Alarm Tool
    └── map1.html          <-- (Optional) Map View
```

Schritt 2: Dateien anlegen & anpassen

Kopiere die bereinigten Codes in die entsprechenden Dateien. **⚠ WICHTIG:** Überall wo **CHANGE_ME...** steht, musst du deine eigenen Werte eintragen!

1. **docker-compose.yml** (im Hauptordner)

Ersetze die Passwörter.

```
version: '3.8'

services:
  web:
    build: .
    ports:
      - "8080:80" # Mapping: Port 8080 am NAS -> Port 80 im Container
    volumes:
      - ./src:/var/www/html
    depends_on:
```

```

- db
restart: always

db:
image: mariadb:10.6
environment:
  MYSQL_ROOT_PASSWORD: CHANGE_ME_ROOT_PASSWORD # <--- ÄNDERN!
  MYSQL_DATABASE: lastwar
  MYSQL_USER: planner
  MYSQL_PASSWORD: CHANGE_ME_DB_PASSWORD      # <--- ÄNDERN! (Muss
gleich sein wie in db_setup.php)
volumes:
  - db_data:/var/lib/mysql
restart: always

```

volumes:
 db_data:

2. Dockerfile (im Hauptordner)

Keine Anpassung nötig.

```

<?php
error_reporting(0);
ini_set('display_errors', 0);

// --- KONFIGURATION ---
$host = 'db';
$db   = 'lastwar';
$user = 'planner';
$pass = 'CHANGE_ME_DB_PASSWORD'; // <--- MUSS EXAKT MIT docker-compose.yml
ÜBEREINSTIMMEN!
$charset = 'utf8mb4';

// ... (Restlicher Code von oben kopieren) ...

// Webhooks ganz unten in der Datei anpassen:
$defaults = [
    'discord_webhook' => "https://discord.com/api/webhooks/DEINE_ID/DEIN_TOKEN", //
<--- ÄNDERN
    'bagger_webhook' => "https://discord.com/api/webhooks/DEINE_ID/DEIN_TOKEN"  //
<--- ÄNDERN
];
// ...
?>

```

4. src/api.php

Den Master-Key ändern (Notfall-Passwort).

```
<?php  
// ... Code Start ...  
session_start();  
  
$master_key = "CHANGE_ME_MASTER_KEY"; // <--- ÄNDERN! (Sicheres Passwort  
wählen)  
// ...  
?>
```

Schritt 3: Container starten

Verbinde dich per SSH auf dein NAS oder nutze den Aufgabenplaner (als Root), um den Container zu starten.

Per SSH (Terminal):

```
cd /volume1/docker/war-command  
docker-compose up -d --build
```

- **up**: Startet die Container.
- **-d**: Im Hintergrund (Detached).
- **--build**: Baut das Image neu (wichtig beim ersten Mal).

 **Test:** Rufe im Browser <http://IP-DEINER-NAS:8080> auf. Die Login-Seite sollte erscheinen.

Schritt 4: DynDNS einrichten (ipv64.net)

Damit dein Server von außen erreichbar ist (auch mit DS-Lite/IPv6), nutzen wir ipv64.net.

1. Erstelle einen Account auf ipv64.net.
 2. Erstelle eine neue Domain (z.B. mein-warcommand.ipv64.net).
 3. Gehe zu **DynDNS Updates** und kopiere die **Update URL**.
-

ebox Schritt 5: FRITZ!Box konfigurieren

A. DynDNS eintragen

1. Logge dich in die FRITZ!Box ein.

2. Gehe zu **Internet -> Freigaben -> DynDNS**.
3. Aktiviere "DynDNS benutzen".
4. Wähle bei Anbieter: **Benutzerdefiniert**.
5. **Update-URL:** Füge die URL von ipv64.net ein.
 - *Hinweis:* Achte darauf, dass die Platzhalter <domain>, <username> und <pass> korrekt gefüllt werden (meist macht ipv64.net das schon fertig für dich in deren Copy-Paste-Vorlage).
 - Beispiel URL Schema:
`https://ipv64.net/update.php?key=<pass>&domain=<domain>&ip=<ipaddr>&ip6=<ip6addr>`
6. **Domainname:** `mein-warcommand.ipv64.net`
7. **Benutzername:** (Dein ipv64 Token oder "none")
8. **Kennwort:** (Dein ipv64 Token)
9. Speichern. Die FRITZ!Box sollte nun grün anzeigen, dass DynDNS aktiv ist.

B. Portfreigabe (Port Forwarding)

Wir leiten Anfragen aus dem Internet an dein NAS weiter.

1. Gehe zu **Internet -> Freigaben -> Portfreigaben**.
2. "Gerät für Freigaben hinzufügen" -> Wähle dein **Synology NAS** aus.
3. **Neue Freigabe:**
 - **Anwendung:** HTTP-Server
 - **Protokoll:** TCP
 - **Port an Gerät:** 80
 - **Port extern:** 80
 - *(Wichtig für Let's Encrypt Validierung)*
4. **Neue Freigabe:**
 - **Anwendung:** HTTPS-Server
 - **Protokoll:** TCP
 - **Port an Gerät:** 443
 - **Port extern:** 443
 - *(Wichtig für den sicheren Zugriff)*
5. Speichern.



Schritt 6: Reverse Proxy & SSL (Synology)

Wir sorgen dafür, dass `mein-warcommand.ipv64.net` automatisch auf deinen Docker-Container (Port 8080) umgeleitet wird und verschlüsselt ist.

A. Reverse Proxy einrichten

1. Öffne auf dem NAS: **Systemsteuerung -> Anmeldeportal -> Erweitert -> Reverse Proxy**.
2. Klicke auf **Erstellen**.

3. **Allgemein:**
 - **Beschreibung:** War Command
 - **Quelle:**
 - Protokoll: **HTTPS**
 - Hostname: **mein-warcommand.ipv64.net**
 - Port: **443**
 - HSTS aktivieren: Ja
 - **Ziel:**
 - Protokoll: **HTTP**
 - Hostname: **localhost**
 - Port: **8080** (Das ist der Docker-Port aus der docker-compose.yml)
4. Speichern.

B. Let's Encrypt Zertifikat (SSL)

1. Gehe zu **Systemsteuerung -> Sicherheit -> Zertifikat**.
 2. Klicke auf **Hinzufügen -> Neues Zertifikat hinzufügen**.
 3. Wähle **Zertifikat von Let's Encrypt abrufen**.
 4. Eintragen:
 - **Domainname:** **mein-warcommand.ipv64.net**
 - **E-Mail:** Deine E-Mail.
 5. Klicke auf "Fertig". (Das funktioniert nur, wenn Port 80 in der FRITZ!Box offen ist!).
 6. **WICHTIG:** Wähle das neue Zertifikat aus, klicke auf **Einstellungen** (oder "Konfigurieren").
 7. Suche in der Liste deinen Reverse Proxy Eintrag (**mein-warcommand.ipv64.net**) und ändere das Zertifikat von "Synology" auf dein neues **Let's Encrypt Zertifikat**.
-



Schritt 7: Erster Start & Login

1. Öffne nun **https://mein-warcommand.ipv64.net** in deinem Browser (oder **http://NAS-IP:8080** lokal).
 2. **Login:**
 - User: **admin**
 - Passwort: **CHANGE_ME_MASTER_KEY** (Den, den du in **api.php** gesetzt hast).
 3. Das System erstellt beim ersten Login automatisch den **admin** User in der Datenbank, falls er fehlt.
 4. **Sicherheitshinweis:** Gehe sofort in die Einstellungen (Zahnrad) oder nutze die "Passwort ändern" Funktion, um dem Admin-User ein individuelles Passwort zu geben.
-

Checkliste bei Problemen

- **Keine Verbindung?** Prüfe, ob die FRITZ!Box Ports 80/443 wirklich auf die IP des NAS zeigen.
- **Datenbankfehler?** Prüfe, ob das Passwort in `docker-compose.yml` und `src/db_setup.php` identisch ist.
- **Discord geht nicht?** Prüfe die Webhook-URLs in `src/db_setup.php`.
- **502 Bad Gateway?** Der Docker-Container läuft nicht. Prüfe SSH: `docker ps`.