

# TiMaS Docker-Installation

---

## Ziel der Anleitung

Diese Anleitung führt Sie durch die Einrichtung des TiMaS-Systems in einer modernen Docker-Umgebung. Sie richtet sich allgemein an Linux-Systeme, zeigt jedoch beispielhaft die Schritte für einen Ubuntu-Server. Nach erfolgreicher Ausführung steht Ihnen TiMaS inklusive Datenbank, Verwaltungsoberfläche der Datenbank (phpMyAdmin) und optionalem HTTPS via Traefik zur Verfügung.

## Inhaltsverzeichnis

- [1. Voraussetzungen](#)
- [2. Docker & Docker Compose installieren](#)
- [3. Arbeitsverzeichnis der TiMaS einrichten](#)
- [4. Konfiguration: `.env` Datei anpassen](#)
- [5. Docker Registry Login \(einmalig\)](#)
- [6. TiMaS starten](#)
- [7. Tools & nützliche Befehle](#)
- [8. Speicherpfade im Container](#)
- [9. Verwendete Standard-Ports](#)
- [10. Erweiterte-Einstellungen über die `.env`-Datei](#)

## 1. Voraussetzungen

Sie benötigen:

- Ein Linux-System (beispielhaft gezeigt mit Ubuntu 20.04 oder neuer; auch andere Distributionen wie Debian geeignet)
- Root-Rechte oder Zugriff per sudo
- Eine Internetverbindung

### Hinweis:

Diese Anleitung beschreibt die Einrichtung von **Docker**, **Docker Compose** und dem **TiMaS-System** auf einem Ubuntu-Server.

## 2. Docker & Docker Compose installieren

① **Hinweis:** Dieser Schritt ist **optional**, wenn Docker und Docker Compose bereits auf dem System installiert und funktionsfähig sind. In diesem Fall können Sie direkt mit Punkt 3 „Arbeitsverzeichnis einrichten“ fortfahren.

### 2.1 System vorbereiten

In diesem Schritt wird das Linux-System vorbereitet, indem grundlegende Werkzeuge installiert und ein Schlüsselring für das Docker-Repository eingerichtet wird.

```
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o
/etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc
```

### 2.2 Docker-Repository hinzufügen

In diesem Schritt wird das offizielle Docker-Repository in das System eingebunden.

Ubuntu

```
echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \
$(. /etc/os-release && echo "${UBUNTU_CODENAME}-$VERSION_CODENAME") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

Debian

```
echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/debian \
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

## 2.3 Docker & Compose installieren

In diesem Schritt installieren Sie die wichtigsten Docker-Komponenten, mit denen Sie Container starten und verwalten können

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin  
docker-compose-plugin
```

### 3. Arbeitsverzeichnis der TiMaS einrichten

#### 3.1 Empfehlung: Separaten Benutzer für TiMaS anlegen

Erstellen Sie für den Betrieb von TiMaS einen eigenen Systembenutzer, z. B. `timas`. Dieser Benutzer sollte Mitglied der Gruppe `docker` sein, um Container verwalten zu können.

 **Hinweis:** Root- oder `sudo`-Rechte sind lediglich für die Installation erforderlich, **nicht** für den regulären Betrieb. Der Benutzer benötigt außerdem Schreibzugriff auf das TiMaS-Arbeitsverzeichnis (`/srv/timas`).

#### Beispiel: Benutzer „timas“ anlegen und zur Docker-Gruppe hinzufügen

```
sudo adduser timas && sudo usermod -aG docker timas
```

#### 3.2 Arbeitsverzeichnis für TiMaS anlegen

Erstellen Sie das Arbeitsverzeichnis unter `/srv`:

```
sudo mkdir -p /srv/timas
```

#### 3.3 Dateien übertragen

Automatischer Download per `curl` Laden Sie das Archiv mit `curl` herunter, entpacken Sie es in das Arbeitsverzeichnis, und setzen Sie anschließend die Dateiberechtigungen:

```
cd /srv
sudo curl -L -o timas.tar.gz https://dl.timas.cloud/timas.tar.gz
sudo tar -xzf timas.tar.gz -C timas
sudo rm timas.tar.gz
```

## Alternative: Lokale oder manuelle Übertragung

Wenn sich die Dateien bereits lokal auf dem System befinden, können Sie sie einfach per Datei-Explorer/[WinSCP](#) oder per [cp](#)-Befehl in das Zielverzeichnis kopieren:

Das TiMaS Docker Installations Paket können Sie von hier herunterladen:

```
https://dl.timas.cloud/timas.tar.gz
```

Das Archiv unter [/srv](#) ablegen und dann folgende Befehle ausführen:

```
cd /srv
sudo tar -xzf timas.tar.gz -C timas
sudo rm timas.tar.gz
```

## 3.4 Lizenzdatei

Legen Sie Ihre Lizenzdatei (Dateiendung [.pusl](#)) im folgenden Verzeichnis ab:

```
/srv/timas/data/license/
```

Beispiel für den Transfer per [scp](#):

```
scp pus7000.pusl root@server.xy:/srv/timas/data/license
```

**Hinweis:** Achten Sie darauf, die Datei an die korrekte Adresse und in das richtige Verzeichnis zu übertragen.

## 3.5 Berechtigung des Verzeichnisses

Der Besitz des gesamten Verzeichnisses wird abschließend auf den Benutzer [timas](#) übertragen, damit dieser später vollen Zugriff auf alle Dateien hat.

```
sudo chown -R timas:timas /srv/timas
```

## 4. Konfiguration: `.env` Datei anpassen

Bearbeiten Sie die Datei `.env` im Projektverzeichnis.

Zum Beispiel mit `nano`

```
nano .env
```

Benötigte Angaben:

- `TIMAS_LICENSE_NAME`: Name der Lizenzdatei (z. B. `pus7000.pusl`). Diese Datei erhalten Sie von dem Händler, bei dem Sie TiMaS erworben haben.
- `TIMAS_VERSION`: Gewünschte TiMaS-Version laut Lizenz
- `TIMAS_HTTP_HOST_RULE`: IP-Adresse(n) oder Domain(s) des Servers (für Traefik)

Beispiel:

```
TIMAS_LICENSE_NAME=pus7000.pusl
TIMAS_VERSION=1.149.3
TIMAS_HTTP_HOST_RULE=Host(`abc.timas.de`) || Host(`localhost`) ||
Host(`127.0.0.1`)
```

### SSL mit Let's Encrypt aktivieren

Passen Sie die folgenden Umgebungsvariablen an, um SSL über Let's Encrypt zu nutzen:

```
TIMAS_HTTPS_HOST_RULE=Host(`abc.timas.de`)
TRAEFIK_SSL=true
LETSENCRYPT_EMAIL=ihre-mail@mail.de
```

**💡 Hinweis:** Ersetzen Sie `abc.timas.de` durch Ihre eigene Domain und tragen Sie Ihre gültige E-Mail-Adresse ein. Diese wird von Let's Encrypt für Zertifikatsbenachrichtigungen verwendet.

## 5. Docker Registry Login (einmalig)

☒ Melden Sie sich anschließend als Benutzer **timas** an (z.B. per **su - timas** oder durch Ab- und Anmeldung).

Führen Sie diesen Befehl im Projektverzeichnis aus, um Zugang zur Container-Registry zu erhalten:

```
cd /srv/timas  
docker login reg.pus-gmbh.eu -u docker-fetch --password-stdin < reg.pat
```

## 6. TiMaS starten

**Hinweis:** Falls Sie der Empfehlung aus Kapitel 3 gefolgt sind und TiMaS mit einem eigenen Benutzer (z. B. `timas`) betreiben, der Mitglied der Gruppe `docker` ist, können Sie die folgende Befehle auch **ohne sudo** ausführen

### 6.1 Datenbank installieren

Dies installiert den Datenbankserver und spielt die Auslieferungs-Datenbank ein

```
docker compose up timas_db
```

Die laufende Konsolenausgabe wird angezeigt, bis Sie mit Ctrl + C abbrechen.

### 6.2 Testlauf im Vordergrund

Beim ersten Start empfiehlt es sich, die Container im Vordergrund auszuführen. So sehen Sie direkt alle Ausgaben im Terminal und erkennen, ob die Dienste korrekt starten oder ob es Fehler gibt (z. B. Lizenzproblem oder fehlende Konfiguration).

```
docker compose up
```

Die laufende Konsolenausgabe wird angezeigt, bis Sie mit Ctrl + C abbrechen.

### 6.3 Dauerbetrieb im Hintergrund

Für den produktiven Einsatz wird empfohlen, TiMaS im sogenannten "Detached Mode" zu starten. Dadurch laufen alle Container dauerhaft im Hintergrund.

```
docker compose up -d
```

Dieser Befehl startet alle definierten Dienste im Hintergrund. Bei einem Neustart des Systems werden die Container automatisch wieder gestartet,

## 7. Tools & nützliche Befehle

### Logs anzeigen

```
# Traefik Logs
docker logs timas_db

# Ausgabe der TiMaS Logs
docker logs timas_server

# Traefik Logs
docker logs traefik
```

### Live-Log verfolgen

```
docker logs -f timas_server
```

### Datenbank-Backup der TiMaS manuell ausführen

 **Hinweis:** Verfügbar ab TiMaS Version 1.150.0.

Mit folgendem Befehl starten Sie ein manuelles Backup der Datenbank aus dem laufenden Container heraus:

```
docker exec timas_server backup
```

Die Sicherung wird dabei automatisch in das Verzeichnis [/srv/timas/data/files/backups/](#) auf dem Host geschrieben.

### In den Container wechseln

Manchmal ist es sinnvoll oder notwendig, direkt in den laufenden TiMaS-Container zu wechseln – etwa zur Fehleranalyse, zum manuellen Ausführen von Befehlen oder zur Kontrolle von Dateien. Dies geschieht über eine interaktive Shell mit folgendem Befehl:

```
docker exec -it timas_server sh

# Beispiel: Backup ausführen
sh cronjobs/backup_database.sh

# Container wieder verlassen
exit
```

## 8. Speicherpfade im Container

| Pfad          | Bedeutung   |
|---------------|---|
| /data/license | Lizenzdatei von TiMaS                                   |
| /data/logs    | Protokolle von TiMaS                                    |
| /data/files   | Von TiMaS generierte Dateien                            |
| /data/mysql   | MariaDB-Datenbankdaten                                  |
| /data/traefik | Traefik mit z.B. Zertifikatspeicher für (Let's Encrypt) |

## 9. Verwendete Standard-Ports

 **Hinweis:** Die Ports können in der .env-Datei angepasst werden.

| Port | Dienst     | Beschreibung               | Konfigurierbar über |
|------|------------|----------------------------|---------------------|
| 80   | HTTP       | TiMaS ohne Verschlüsselung | TIMAS_HTTP_PORT     |
| 443  | HTTPS      | TiMaS mit Verschlüsselung  | TIMAS_HTTPS_PORT    |
| 8085 | phpMyAdmin | Datenbank-Verwaltung       | TIMAS_PMA_PORT      |

## 10. Erweiterte-Einstellungen über die `.env`-Datei

Bearbeiten Sie die Datei `.env` im Projektverzeichnis. Dort legen Sie unter anderem Lizenzinformationen, Netzwerkeinstellungen, Ressourcenbegrenzungen und Zugangsdaten fest.

| Variable                            | Beschreibung   |
|-------------------------------------|--|
| <code>TIMAS_LICENSE_NAME</code>     | Name der Lizenzdatei (z. B. <code>pus7000.pus1</code> ). Bereitgestellt durch den Händler. |
| <code>TIMAS_VERSION</code>          | Freigegebene Version, die durch Ihre Lizenz aktiviert ist.                                 |
| <code>TIMAS_MEMORY_MIN</code>       | Minimale Menge an zugewiesenen RAM (z. B. <code>512m</code> ).                             |
| <code>TIMAS_MEMORY_LIMIT</code>     | Maximale RAM-Zuweisung (z. B. <code>2g</code> ).   |
| <code>TIMAS_CONTAINER_ARCH</code>   | Zielarchitektur des Containers: <code>amd64</code> (x86_64) oder <code>arm64</code> .      |
| <code>TIMAS_HTTP_PORT</code>        | Port für HTTP-Zugriff (Standard: 80).  |
| <code>TIMAS_HTTPS_PORT</code>       | Port für HTTPS-Zugriff (Standard: 443).  |
| <code>TIMAS_HTTP_HOST_RULE</code>   | Hostregel für Zugriff über HTTP, z. B. IP-Adressen oder Domains.                           |
| <code>TIMAS_HTTPS_HOST_RULE</code>  | Hostregel für Zugriff über HTTPS.  |
| <code>TRAEFIK_SSL</code>            | Aktiviert SSL/TLS über Traefik ( <code>true/false</code> ).                                |
| <code>LETSENCRYPT_EMAIL</code>      | E-Mail für Let's Encrypt Zertifikatsbeantragung (bei SSL aktiv).                           |
| <code>TIMAS_DB_PASSWORD</code>      | Benutzerpasswort der MariaDB-Datenbank.  |
| <code>TIMAS_DB_ROOT_PASSWORD</code> | Root-Passwort der MariaDB-Datenbank.   |
| <code>TIMAS_ADMIN_USER</code>       | Benutzername für das Admin-Webinterface.   |
| <code>TIMAS_ADMIN_PASSWORD</code>   | Passwort für das Admin-Webinterface.   |
| <code>TIMAS_WEBTERMINAL_TZ</code>   | Zeitzone für das Web-Terminal (z. B. <code>Europe/Berlin</code> ).                         |