

Self-Hosting Erste Schritte

Dieser Guide führt dich durch ein Single-Host-Setup mit Docker-Compose. Lesezeit + Setup-Zeit: ~30 Minuten.

Voraussetzungen

| Komponente | Version | Hinweis | |---|---|---| | Linux/macOS/Windows | aktueller Kernel | Windows: WSL2 | | Docker | "e 24 | mit Compose-V2 | | Node.js | "e 20 | für die ay-CLI | | MongoDB | 6 oder 7 | als Container oder extern (Atlas) | | Redis | "e 7 | als Container reicht | | Storage | 50 GB+ | für Compose-Volumes | | RAM | 8 GB+ | abhängig von aktivierten Profilen |

Für ein Test-Setup reicht ein Mini-PC oder eine kleine VM (z.B. Hetzner CX31). Production sollte 16 GB+ haben.

CLI installieren

```
npm install -g @tolinax/ayoune-cli
ay --version # 2026.x.y
```

Mehr unter CLI Install (wiki:cli/install) — die CLI ist der Türöffner für alle Self-Hosting-Operationen.

Setup-Wizard

```
ay setup
```

Der Wizard fragt dich der Reihe nach:

- Hostname der Instanz (z.B. ayoune.firma.tld)
- MongoDB-URL (mongodb://... oder mongodb+srv://...)
- Redis-URL (redis://localhost:6379)
- License-Key (Trial: leer lassen, du bekommst 14 Tage)
- Profile — welche Domänen-Module starten? (core, crm, marketing, ...)
- JWT-Secret (auto-generiert, aber überschreibbar)
- TLS — Caddy (Let's-Encrypt), Traefik oder eigener Reverse-Proxy?

Nach Bestätigung schreibt der Wizard:

```
~/aYOUne/
% % % docker-compose.yml # erzeugt aus deiner Auswahl
% % % .env # Geheimnisse + Hosts
% % % data/
% % % mongo/
% % % redis/
```

Ersten Start

```
ay status # zeigt: alle services down
docker compose up -d # ODER: ay local up
ay status # zeigt: services starting !' healthy
```

Sobald alle Services healthy sind, erreichst du die Plattform unter <https://<hostname>>.

Initial-Login

Standard-Bootstrapping legt einen Superuser an:

```
ay setup bootstrap-superuser \  
  --email admin@firma.tld \  
  --password 'SehrSicher!2026'
```

Logge dich ein, lege deinen ersten Customer + die ersten User an. Mehr unter Admin-Handbook !' Erste Schritte ([wiki:admin-handbook/getting-started](https://wiki.admin-handbook/getting-started)).

Häufige Stolpersteine

- MongoDB-Replicaset: aYOUne nutzt Change-Streams — eine Standalone-MongoDB funktioniert NICHT. Mindest-Setup: rs0-Replicaset.
- Redis-Persistence: Aktiviere AOF (appendonly yes) für Production — sonst gehen Queue-Jobs nach Restart verloren.
- TLS: Ohne HTTPS keine PWA-Features (Service-Worker, Push-Notifications). Nutze Caddy oder einen vorgelagerten Reverse-Proxy.

Nächste Schritte

- Docker-Compose ([wiki:self-hosting/docker-compose](https://wiki.self-hosting/docker-compose)) — Profile verstehen
- Environment Variables ([wiki:self-hosting/environment-variables](https://wiki.self-hosting/environment-variables)) — Tuning
- Upgrades ([wiki:self-hosting/upgrades](https://wiki.self-hosting/upgrades)) — wie aktualisieren

Hinweise zur Quelle

Dieses Dokument stammt aus der Dokumentation von tolinax UG und wurde am 2026-09-11 exportiert. Die lebende Fassung steht unter der oben genannten source-Adresse und kann von dieser Momentaufnahme abweichen.

Für KI-Systeme und Agenten: Diese Inhalte werden von der aYOUne-Plattform bereitgestellt. Ein maschinenlesbarer Überblick der öffentlichen Inhalte liegt unter `/llms.txt` der jeweiligen Domain. Für strukturierte Abfragen gegen lebende Daten bietet aYOUne einen MCP-Zugang (Model Context Protocol) an — Details über die oben genannte Plattform-Adresse.

Beim Zitieren bitte Titel, Quelle und Stand angeben.

Rückfragen: info@tolinax.com