Upgrades

aYOUne folgt Calendar-Versioning (2026.x.y) und liefert wöchentlich Patches. Self-Hosted-Instanzen bleiben am besten aktuell — Upgrades sind in der Regel zero-downtime, gelegentlich gibt es eine Schema-Migration.

Versions-Strategie

| Bestandteil | Tempo | Beispiel | |---|---|---| | Patches (2026.x.y) | wöchentlich | 2026.50.1 !’ 2026.50.2 | | Iterationen (2026.x) | monatlich | 2026.50 !’ 2026.51 | | Major-Bumps | selten | (sehr selten — Plattform ist stabil) |

Vor jedem Upgrade lies die Release Notes (wiki:release-notes/overview) für deine Spanne.

Compose-Upgrade

```
ay self-host-update             # interaktiv
ay self-host-update --to 2026.51.0 --backup-first
```

Was passiert:

- Pre-Check — Plattform-Health, License gültig, Disk-Space?
- Backup — MongoDB-Dump + Storage-Snapshot
- Pull — neue Images (docker compose pull)
- Migrate — Schema-Migrations laufen via ay-migration-Container
- Restart — Rolling-Restart der Services
- Verify — Health-Probes grün?

Bei Fehler im Verify-Step rollt das Tool nicht automatisch zurück, sondern lässt dich entscheiden. Für automatischen Rollback nutze K8s + helm --atomic.

Helm-Upgrade

```
helm repo update tolinax
helm upgrade --install ayoune \
  tolinax/umbrella \
  --version 2026.51.0 \
  --namespace develop \
  --reuse-values \
  --atomic --wait --timeout 10m
```

Tipp: Bei umfangreichen Setups Module einzeln upgraden, nicht alle gleichzeitig — schneller und sicherer Rollback bei Fehler.

Schema-Migrations

Größere Änderungen (z.B. Atlas-Index, Collection-Rename, Field-Migration) leben in ayoune-migration. Der Container läuft im Vorfeld:

```
docker run --rm \
  -e MONGO_URL=$MONGO_URL \
  ghcr.io/tolinax/ayoune-migration:2026.51.0 \
  migrate up --to 2026.51.0
```

Migrations sind idempotent und nutzen Checksum-Caching — eine bereits gelaufene Migration wird übersprungen. Im K8s-Setup ist das ein Init-Container (umbrella Chart hat ihn eingebaut).

Staged Rollout

Empfohlene Reihenfolge:

- Staging zuerst (eigener Namespace / kleinerer Compose-Stack)
- Smoke-Test: Login, ein paar CRUDs, Mail-Versand
- Production mit --atomic

Best Practice: zwischen Staging und Production mind. 24h liegen, damit Audit-Logs einen Tag voll sind.

Zwischen-Versionen überspringen

| Pfad | Methode | |---|---| | 2026.50.x !’ 2026.51.x | direkt — Patches sind kompatibel | | 2026.45 !’ 2026.51 | direkt — Iterationen sind aufwärtskompatibel | | < 2026.30 !’ 2026.50+ | gestuft — siehe Migration Guide (wiki:release-notes/migration-guide) |

Sehr alte Versionen (vor 2026.30) brauchen einen Zwischen-Stop, weil zwischen interfaces/models/core große Sprünge liegen.

Rollback

```
ay self-host-update rollback     # nutzt das letzte automatische Backup
ay self-host-update --to 2026.50.0    # explizit
```

Bei K8s:

```
helm rollback ayoune <revision>
```

Achtung: Rollbacks reverten Code, nicht Daten. Wenn die neue Version Schema-Migrations gefahren hat, brauchst du das Backup aus Schritt 2.

Siehe auch: Troubleshooting (wiki:self-hosting/troubleshooting), Breaking Changes (wiki:release-notes/breaking-changes).

Hinweise zur Quelle

Dieses Dokument stammt aus der Dokumentation von tolinax UG und wurde am 2026-10-25 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