

# KodeMed

## DevOps- & Serveradministrations-Handbuch

Für Spital-IT-Administratoren, DevOps-Ingenieure &  
Systembetreiber

Version 2026.5.15.59502 | KodeMed AG

# Architekturübersicht

Vier serverseitige Dienste + Windows-Desktop-Client

Dienst	Port	Technologie	Funktion
KodeMed Server	8080	Spring Boot 3.4	REST API, WebSocket, Sitzungen, Audit
KodeMed DataServer	8081	Spring Boot 3.4	ICD-10-, CHOP-, ATC-, SwissDRG-Kataloge
KodeMed GrouperServer	8082	Spring Boot 3.4	SwissDRG-, TARPSY-, ST-Reha-Gruppierung
KodeMed CodingUI	3000	React / nginx	Web-Frontend

Datenbank: PostgreSQL 16 (gemeinsam für Server + DataServer). Authentifizierung: OAuth2 / OpenID Connect.

# Deployment-Optionen

Drei Bereitstellungsmodelle für jede Infrastruktur

## Docker Compose

Empfohlen für Einzelserver

- Alle Dienste in einer docker-compose.yml
- PostgreSQL als Container enthalten
- Einfachste Einrichtung und Wartung
- Ideal für: Demo, kleine-mittlere Spitäler

## Linux Nativ (systemd)

Für Container-freie Richtlinien

- JARs laufen als systemd-Dienste
- Externer PostgreSQL erforderlich
- Multi-Instanz-Unterstützung auf demselben Host
- Ideal für: strikte Infrastruktur-Richtlinien

## Kubernetes / OpenShift

Enterprise, skalierbar

- Helm Chart mit allen Diensten
- Horizontale Pod-Autoskalierung
- Pod Disruption Budgets, Rolling Updates
- Ideal für: grosse Spitäler, Multi-Site

# Docker Compose – Einrichtung

Schritt-für-Schritt-Installation auf einem Server

## Installationsschritte

- 1. Verzeichnis erstellen: `mkdir -p /opt/kodemed`
- 2. `docker-compose.yml`, `.env`, `runtime-config.js` kopieren
- 3. `.env` bearbeiten: Datenbank, OIDC, CORS, Schlüssel
- 4. `runtime-config.js`: API-URLs für Ihre Domain anpassen
- 5. `docker login harbor.mieresit.com`
- 6. `docker compose pull` && `docker compose up -d`

## Wichtige Umgebungsvariablen

- `POSTGRES_PASSWORD` — Datenbankpasswort
- `OIDC_ISSUER_URI` — SSO-Provider-URL
- `CORS_ALLOWED_ORIGINS` — öffentliche Domains
- `KODEMED_ENCRYPTION_KEY` — Datenverschlüsselung (DSGVO)
- `KODEMED_VERSION` — Image-Tag von KodeMed AG

# Linux Nativ — Installation

Automatischer Installer für Ubuntu / Debian

## Interaktiver Installer

- `tar -xzf kodemed-linux-<version>.tar.gz`
- `sudo bash install-kodemed.sh`
- Komponenten wählen (Server, DataServer, Grouper, UI)
- Ports, Datenbank, OIDC konfigurieren
- systemd-Dienste werden automatisch erstellt

## Verzeichnisstruktur

- `/opt/kodemed/` — Installation (JARs, UI)
- `/etc/kodemed/` — Konfigurationsdateien
- `/var/log/kodemed/` — Anwendungslogs
- `/var/lib/kodemed/` — Daten (Importe)

## Dienstverwaltung

- `systemctl start|stop|restart kodemed-server`
- `systemctl status kodemed-server`
- `journalctl -u kodemed-server -f`

# Reverse-Proxy-Konfiguration

HTTPS-Terminierung und WebSocket-Unterstützung

## Apache (mod\_proxy)

- Aktivieren: mod\_proxy, mod\_proxy\_http, mod\_proxy\_wstunnel, mod\_ssl
- ProxyPass /ws/ ws://localhost:8080/ws/ (WebSocket ZUERST)
- ProxyPass / http://localhost:8080/ (HTTP danach)
- Ein VirtualHost pro Dienst-Subdomain

## nginx

- proxy\_http\_version 1.1 für WebSocket
- proxy\_set\_header Upgrade \$http\_upgrade
- proxy\_set\_header Connection "upgrade"
- Ein Server-Block pro Dienst-Subdomain

**WebSocket-Unterstützung ist obligatorisch — CodingClient und UI kommunizieren über /ws/ mit dem Server.**

# SSO / OIDC-Konfiguration

OpenID Connect für Authentifizierung

## Keycloak (Empfohlen)

- Vorkonfigurierter Keycloak-Stack bereitgestellt
- Caddy für automatisches HTTPS (Let's Encrypt)
- Tägliche Datenbanksicherung (30 Tage Aufbewahrung)
- Brute-Force-Schutz: max. 5 Fehlversuche

## Erforderliche OIDC-Clients

- kodemed-ui — öffentlicher Client, PKCE-Flow
- kodemed-server — vertraulich, Service-Account
- Rollen: admin, coder, approver, viewer

## Alternative Anbieter

- Azure AD, Okta, Auth0 ebenfalls unterstützt
- `OIDC_ISSUER_URI` + `OIDC_JWK_URI` in `.env` setzen
- Token muss `sub` + Rollen/Gruppen-Claims enthalten

# KodeMed aktualisieren

Upgrade-Verfahren für jedes Bereitstellungsmodell

## Docker Compose Update

- `docker compose pull`
- `docker compose up -d`
- `curl localhost:8080/actuator/health`
- Rollback: vorherige Version in `.env` setzen

## Linux Nativ Update

- Dienste stoppen: `systemctl stop kodemed-*`
- Backup: `cp -r /opt/kodemed /opt/kodemed.backup`
- Neue Version extrahieren + `sudo bash install-kodemed.sh --upgrade`
- Dienste starten und Gesundheit prüfen

## Kubernetes / OpenShift Update

- `imageTag` in `values.yaml` aktualisieren
- `helm upgrade kodemed charts/kodemed/ -f values.yaml`
- `kubectl rollout status deployment/kodemed-server`
- Rollback: `helm rollback kodemed 1`

# Health Checks & Backup

Überwachungsendpunkte und Datensicherung

## Health-Endpunkte

Dienst	Endpunkt	Erwartet
Server	/actuator/health	{"status":"UP"}
DataServer	/actuator/health	{"status":"UP"}
GrouperServer	/actuator/health	{"status":"UP"}
CodingUI	/	HTTP 200

## Datenbanksicherung

- PostgreSQL ist die einzige zustandsbehaftete Komponente
- `pg_dump -U kodemed kodemed | gzip > backup.sql.gz`
- Tägliche Sicherungen planen (cron oder pg-backup-Container)
- Wiederherstellungsverfahren regelmässig testen

# Client-Bereitstellung & Fehlerbehebung

Windows-Client und häufige Probleme

## Windows-Client (MSI)

- MSI für GPO / SCCM / Intune / Citrix
- `msiexec /i KodeMed.msi /quiet SERVERURL="https://..."`
- Kein Admin für Standard-Installer nötig
- Autostart, Sprache, Installationsverzeichnis-Optionen

## Häufige Probleme

- Dienst startet nicht: Logs + DB-Verbindung prüfen
- OIDC-Fehler: Issuer-URL erreichbar prüfen
- WebSocket fehlgeschlagen: Reverse-Proxy-Konfiguration prüfen
- Port-Konflikt: `ss -tlnp | grep 8080`

# KodeMed

## Support & Kontakt

[support@kodemed.ch](mailto:support@kodemed.ch) | [www.kodemed.ch](http://www.kodemed.ch)

[info@kodemed.ch](mailto:info@kodemed.ch)

[www.kodemed.ch](http://www.kodemed.ch)

KodeMed AG • Schweiz