

# KodeMed

## Guida DevOps & Amministrazione Server

Per amministratori IT ospedalieri, ingegneri DevOps & operatori di sistema

Version 2026.5.15.59502 | KodeMed AG

# Panoramica architettura

Quattro servizi server + client desktop Windows

Servizio	Porta	Tecnologia	Ruolo
KodeMed Server	8080	Spring Boot 3.4	API REST, WebSocket, sessioni, audit
KodeMed DataServer	8081	Spring Boot 3.4	Cataloghi ICD-10, CHOP, ATC, SwissDRG
KodeMed GrouperServer	8082	Spring Boot 3.4	Raggruppamento SwissDRG, TARPSY, ST Reha
KodeMed CodingUI	3000	React / nginx	Interfaccia web

Database: PostgreSQL 16 (condiviso tra Server + DataServer). Autenticazione: OAuth2 / OpenID Connect.

# Opzioni di distribuzione

Tre modelli di distribuzione per ogni infrastruttura

## Docker Compose

Raccomandato per server singolo

- Tutti i servizi in un docker-compose.yml
- PostgreSQL incluso come container
- Configurazione e manutenzione più semplici
- Ideale per: demo, ospedali piccoli-medi

## Linux nativo (systemd)

Per politiche senza container

- JAR eseguiti come servizi systemd
- PostgreSQL esterno richiesto
- Supporto multi-istanza sullo stesso host
- Ideale per: politiche infrastrutturali rigide

## Kubernetes / OpenShift

Enterprise, scalabile

- Chart Helm con tutti i servizi
- Autoscaling orizzontale dei pod
- Pod Disruption Budgets, rolling update
- Ideale per: grandi ospedali, multi-sito

# Docker Compose — Configurazione

Passo dopo passo su un singolo server

## Passaggi di installazione

- 1. Creare la directory: `mkdir -p /opt/kodemed`
- 2. Copiare `docker-compose.yml`, `.env`, `runtime-config.js`
- 3. Modificare `.env`: database, OIDC, CORS, chiavi
- 4. Modificare `runtime-config.js`: URL API per il vostro dominio
- 5. `docker login harbor.miesit.com`
- 6. `docker compose pull` && `docker compose up -d`

## Variabili d'ambiente chiave

- `POSTGRES_PASSWORD` — password database
- `OIDC_ISSUER_URI` — URL provider SSO
- `CORS_ALLOWED_ORIGINS` — i vostri domini pubblici
- `KODEMED_ENCRYPTION_KEY` — cifratura dati (GDPR)
- `KODEMED_VERSION` — tag immagine da KodeMed AG

# Linux nativo — Installazione

Installatore automatizzato per Ubuntu / Debian

## Installatore interattivo

- `tar -xzf kodemed-linux-<version>.tar.gz`
- `sudo bash install-kodemed.sh`
- Selezionare i componenti (Server, DataServer, Grouper, UI)
- Configurare porte, database, OIDC
- Servizi systemd creati automaticamente

## Struttura directory

- `/opt/kodemed/` — installazione (JAR, UI)
- `/etc/kodemed/` — file di configurazione
- `/var/log/kodemed/` — log applicativi
- `/var/lib/kodemed/` — dati (importazioni)

## Gestione servizi

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

# Configurazione reverse proxy

Terminazione HTTPS e supporto WebSocket

## Apache (mod\_proxy)

- Abilitare: mod\_proxy, mod\_proxy\_http, mod\_proxy\_wstunnel, mod\_ssl
- ProxyPass /ws/ ws://localhost:8080/ws/ (WebSocket PER PRIMO)
- ProxyPass / http://localhost:8080/ (HTTP dopo)
- Un VirtualHost per sottodominio di servizio

## nginx

- proxy\_http\_version 1.1 per WebSocket
- proxy\_set\_header Upgrade \$http\_upgrade
- proxy\_set\_header Connection "upgrade"
- Un blocco server per sottodominio di servizio

**Il supporto WebSocket è obbligatorio — CodingClient e UI comunicano via /ws/ con il Server.**

# Configurazione SSO / OIDC

OpenID Connect per l'autenticazione

## Keycloak (Raccomandato)

- Stack Keycloak preconfigurato fornito
- Caddy per HTTPS automatico (Let's Encrypt)
- Backup giornalieri del database (30 giorni di retention)
- Protezione brute force: max 5 tentativi falliti

## Client OIDC richiesti

- kodemed-ui — client pubblico, flusso PKCE
- kodemed-server — confidenziale, account di servizio
- Ruoli: admin, coder, approver, viewer

## Provider alternativi

- Azure AD, Okta, Auth0 supportati
- Impostare `OIDC_ISSUER_URI` + `OIDC_JWK_URI` in `.env`
- Il token deve contenere i claim `sub` + `ruoli/gruppi`

# Aggiornare KodeMed

Procedure di aggiornamento per ogni modello

## Aggiornamento Docker Compose

- `docker compose pull`
- `docker compose up -d`
- `curl localhost:8080/actuator/health`
- Rollback: impostare la versione precedente in `.env`

## Aggiornamento Linux nativo

- Arrestare i servizi: `systemctl stop kodemed-*`
- Backup: `cp -r /opt/kodemed /opt/kodemed.backup`
- Estrarre nuova versione + `sudo bash install-kodemed.sh --upgrade`
- Avviare i servizi e verificare la salute

## Aggiornamento Kubernetes / OpenShift

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

# Health Check & Backup

Endpoint di monitoraggio e protezione dati

## Endpoint di salute

Servizio	Endpoint	Atteso
Server	/actuator/health	{"status":"UP"}
DataServer	/actuator/health	{"status":"UP"}
GrouperServer	/actuator/health	{"status":"UP"}
CodingUI	/	HTTP 200

## Backup database

- PostgreSQL è l'unico componente con stato
- `pg_dump -U kodemed kodemed | gzip > backup.sql.gz`
- Pianificare backup giornalieri (cron o container pg-backup)
- Testare regolarmente la procedura di ripristino

# Distribuzione client & Risoluzione problemi

Client Windows e problemi comuni

## Client Windows (MSI)

- MSI per GPO / SCCM / Intune / Citrix
- `msiexec /i KodeMed.msi /quiet SERVERURL="https://..."`
- Nessun diritto admin per l'installatore standard
- Opzioni: avvio automatico, lingua, directory personalizzata

## Problemi comuni

- Servizio non si avvia: controllare log + connettività DB
- Errore OIDC: verificare che l'URL issuer sia raggiungibile
- WebSocket fallisce: controllare config reverse proxy
- Conflitto porta: `ss -tlnp | grep 8080`

# KodeMed

## Supporto & Contatto

[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 • Svizzera