

KodeMed

Guide DevOps & Administration Serveur

Pour administrateurs IT hospitaliers, ingénieurs DevOps & opérateurs système

Version 2026.5.15.59502 | KodeMed AG

Aperçu de l'architecture

Quatre services serveur + client Windows

Service	Port	Technologie	Rôle
KodeMed Server	8080	Spring Boot 3.4	API REST, WebSocket, sessions, audit
KodeMed DataServer	8081	Spring Boot 3.4	Catalogues ICD-10, CHOP, ATC, SwissDRG
KodeMed GrouperServer	8082	Spring Boot 3.4	Groupage SwissDRG, TARPSY, ST Reha
KodeMed CodingUI	3000	React / nginx	Interface web

Base de données : PostgreSQL 16 (partagée entre Server + DataServer). Authentification : OAuth2 / OpenID Connect.

Options de déploiement

Trois modèles de déploiement pour toute infrastructure

Docker Compose

Recommandé pour serveur unique

- Tous les services dans un docker-compose.yml
- PostgreSQL inclus comme conteneur
- Configuration et maintenance les plus simples
- Idéal pour : démo, hôpitaux petits-moyens

Linux natif (systemd)

Pour politiques sans conteneur

- JARs exécutés comme services systemd
- PostgreSQL externe requis
- Support multi-instances sur le même hôte
- Idéal pour : politiques d'infrastructure strictes

Kubernetes / OpenShift

Enterprise, scalable

- Chart Helm avec tous les services
- Autoscaling horizontal des pods
- Pod Disruption Budgets, rolling updates
- Idéal pour : grands hôpitaux, multi-sites

Docker Compose — Configuration

Étape par étape sur un serveur unique

Étapes d'installation

- 1. Créer le répertoire : `mkdir -p /opt/kodemed`
- 2. Copier `docker-compose.yml`, `.env`, `runtime-config.js`
- 3. Éditer `.env` : base de données, OIDC, CORS, clés
- 4. Éditer `runtime-config.js` : URLs API pour votre domaine
- 5. `docker login harbor.miersit.com`
- 6. `docker compose pull && docker compose up -d`

Variables d'environnement clés

- `POSTGRES_PASSWORD` — mot de passe base de données
- `OIDC_ISSUER_URI` — URL du fournisseur SSO
- `CORS_ALLOWED_ORIGINS` — vos domaines publics
- `KODEMED_ENCRYPTION_KEY` — chiffrement données (RGPD)
- `KODEMED_VERSION` — tag d'image fourni par KodeMed AG

Linux natif — Installation

Installeur automatisé pour Ubuntu / Debian

Installeur interactif

- `tar -xzf kodemed-linux-<version>.tar.gz`
- `sudo bash install-kodemed.sh`
- Sélectionner les composants (Server, DataServer, Grouper, UI)
- Configurer ports, base de données, OIDC
- Services systemd créés automatiquement

Structure des répertoires

- `/opt/kodemed/` — installation (JARs, UI)
- `/etc/kodemed/` — fichiers de configuration
- `/var/log/kodemed/` — journaux applicatifs
- `/var/lib/kodemed/` — données (imports)

Gestion des services

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

Configuration du reverse proxy

Terminaison HTTPS et support WebSocket

Apache (mod_proxy)

- Activer : mod_proxy, mod_proxy_http, mod_proxy_wstunnel, mod_ssl
- ProxyPass /ws/ ws://localhost:8080/ws/ (WebSocket EN PREMIER)
- ProxyPass / http://localhost:8080/ (HTTP après)
- Un VirtualHost par sous-domaine de service

nginx

- proxy_http_version 1.1 pour WebSocket
- proxy_set_header Upgrade \$http_upgrade
- proxy_set_header Connection "upgrade"
- Un bloc server par sous-domaine de service

Le support WebSocket est obligatoire — le CodingClient et l'UI communiquent via /ws/ avec le Server.

Configuration SSO / OIDC

OpenID Connect pour l'authentification

Keycloak (Recommandé)

- Stack Keycloak préconfiguré fourni
- Caddy pour HTTPS automatique (Let's Encrypt)
- Sauvegardes quotidiennes (rétention 30 jours)
- Protection brute force : 5 échecs max.

Clients OIDC requis

- kodemed-ui — client public, flux PKCE
- kodemed-server — confidentiel, compte de service
- Rôles : admin, coder, approver, viewer

Fournisseurs alternatifs

- Azure AD, Okta, Auth0 également supportés
- Définir `OIDC_ISSUER_URI` + `OIDC_JWK_URI` dans `.env`
- Le token doit contenir les claims `sub` + rôles/groupes

Mettre à jour KodeMed

Procédures de mise à jour pour chaque modèle

Mise à jour Docker Compose

- `docker compose pull`
- `docker compose up -d`
- `curl localhost:8080/actuator/health`
- Rollback : définir la version précédente dans `.env`

Mise à jour Linux natif

- Arrêter les services : `systemctl stop kodemed-*`
- Sauvegarde : `cp -r /opt/kodemed /opt/kodemed.backup`
- Extraire nouvelle version + `sudo bash install-kodemed.sh --upgrade`
- Démarrer les services et vérifier la santé

Mise à jour Kubernetes / OpenShift

- Mettre à jour `imageTag` dans `values.yaml`
- `helm upgrade kodemed charts/kodemed/ -f values.yaml`
- `kubectl rollout status deployment/kodemed-server`
- Rollback : `helm rollback kodemed 1`

Health Checks & Sauvegarde

Points de contrôle et protection des données

Points de contrôle santé

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

Sauvegarde base de données

- PostgreSQL est le seul composant à état
- `pg_dump -U kodemed kodemed | gzip > backup.sql.gz`
- Planifier des sauvegardes quotidiennes (cron ou conteneur pg-backup)
- Tester régulièrement la procédure de restauration

Déploiement client & Dépannage

Client Windows et problèmes courants

Client Windows (MSI)

- MSI pour GPO / SCCM / Intune / Citrix
- `msiexec /i KodeMed.msi /quiet SERVERURL="https://..."`
- Pas de droits admin pour l'installateur standard
- Options : démarrage auto, langue, répertoire personnalisé

Problèmes courants

- Service ne démarre pas : vérifier logs + connectivité DB
- Erreur OIDC : vérifier que l'URL issuer est accessible
- Échec WebSocket : vérifier config reverse proxy
- Conflit de port : `ss -tlnp | grep 8080`

KodeMed

Support & Contact

support@kodemed.ch | www.kodemed.ch

info@kodemed.ch

www.kodemed.ch

KodeMed AG • Suisse