KodeMed

OpenShift Deployment Policy

For DevOps, Platform Engineers & IT Architects

Version 2026.5.15.59502 | KodeMed AG

Architecture — Components

Services deployed in OpenShift

Service	Technology	Port	Type
kodemed-server	Java 21, Spring Boot 3.4	8080	Deployment
kodemed-dataserver	Java 21, Spring Boot 3.4	8081	Deployment
kodemed-grouper	Java 21, Spring Boot 3.4	8082	Deployment
kodemed-ui	React 18 / nginx	3000	Deployment
PostgreSQL 16	External (not a pod)	5432	External DB

Windows Clients (not containerized):

- KodeMed.Interface — COM DLL .NET 9.0 (HIS integration)
- KodeMed.CodingClient — System tray app .NET 9.0 (WebSocket bridge)

Resource Limits per Pod

CPU and memory requests/limits for each service

Pod	CPU Req	CPU Limit	Mem Req	Mem Limit
kodemed-server	250m	1000m	512Mi	1Gi
kodemed-dataserver	200m	500m	256Mi	1Gi
kodemed-grouper	200m	500m	512Mi	1Gi
kodemed-ui (nginx)	50m	200m	32Mi	64Mi
TOTAL (x2 replicas)	1400m	4400m	2600Mi	6.1Gi

JVM: -XX:+UseContainerSupport -XX:MaxRAMPercentage=75.0 — automatically respects container memory limit.

High Availability

Minimum 2 replicas per service with rolling updates

Service	Replicas	Rolling Update	Constraints
kodemed-server	2	maxUnavail: 0, maxSurge: 1	Sticky sessions required
kodemed-dataserver	2	maxUnavail: 1, maxSurge: 1	Stateless — no constraints
kodemed-grouper	2	maxUnavail: 1, maxSurge: 1	Stateless — no constraints
kodemed-ui	2	maxUnavail: 1, maxSurge: 1	Stateless — no constraints

Sticky Sessions — kodemed-server

- WebSocket connections stored in JVM memory (ConcurrentHashMap)
- OpenShift Route MUST configure session affinity:
- haproxy.router.openshift.io/balance: source
- haproxy.router.openshift.io/timeout: 86400s
- Pod restart: clients reconnect in 60s — sessions are in DB, no data loss

Health Probes

Readiness, liveness, and startup probes for all services

Service	Endpoint	Port	Init Delay	Period
kodemed-server	/actuator/health	8080	60s	10s
kodemed-dataserver	/actuator/health	8081	60s	10s
kodemed-grouper	/actuator/health	8082	60s	10s
kodemed-ui	/	3000	5s	10s

Probe Types

- Readiness — determines if pod can receive traffic (removed from Service if failing)
- Liveness — determines if pod is alive (container restarted if failing)
- Startup — protects slow-starting apps (JVM warmup), failureThreshold: 12

Database — External PostgreSQL

No database pods allowed in OpenShift — external DB with SSL in-transit

Configuration

- No DB pod in OpenShift — compliant with cluster policy
- SSL in-transit: sslmode=require&sslrootcert=/certs/ca.crt
- SSL cert mounted via OpenShift Secret (kodemed-db-ssl-cert)
- Connection pool: HikariCP max 15 conn, min 5 idle, per service

Schema & Migrations

- Shared schema: Server prefix km_app_, DataServer prefix km_data_
- Flyway migrations run automatically on startup
- No manual SQL scripts required
- Always backup database before upgrading

Image Security & SCC

Immutable images, vulnerability scanning, and OpenShift Security Context Constraints

Immutable Images

- readOnlyRootFilesystem: true
- allowPrivilegeEscalation: false
- runAsNonRoot: true, runAsUser: 1001
- No runtime installations — config via env vars only
- No nested containers

Scan & Signatures

- Trivy scan on every CI build
- Critical CVEs = image blocked from publication
- Cosign for supply-chain integrity
- Base: eclipse-temurin:21-jre-alpine

Image	UID	SCC
kodemed-server	1001	restricted (OK)
kodemed-dataserver	1001	restricted (OK)
kodemed-grouper	1001	restricted (OK)
kodemed-ui	root*	ACTION REQUIRED

Action: migrate kodemed-ui to nginxinc/nginx-unprivileged or configure UID > 1000.

Open Questions — Answers

Responses to the meeting questions

YES

PDB (PodDisruptionBudget)?

minAvailable: 1 for each service. With 2 replicas, cluster can evict max 1 pod at a time.

NO

Kubernetes Operator?

Not needed. Standard K8s resources (Deployment, Service, Route) suffice. No custom CRDs.

MIXED

Stateless vs Stateful?

DataServer, Grouper, UI = stateless. Server = stateful (WebSocket in JVM). All as Deployment (not StatefulSet). Sticky sessions on Route.

HELM

Deployment type?

Helm charts recommended. Parameterizable per env (values-test.yaml, values-prod.yaml). Easy rollback. ArgoCD/FluxCD compatible.

YES

GitOps compatible?

Immutable images + versioned tags (YYYY.M.D.BUILD) + declarative config. Workflow: push image > update values.yaml > ArgoCD sync.

RASCI Matrix

Clearly defined responsibilities for each actor

Activity	Mieres IT	Client Platform	Client IT
Build & scan images	R	I	I
Publish images to JFrog	R	S	I
Create OpenShift namespace	C	R	A
Configure quotas/limits	C	R	A
Create Helm charts	R	C	I
Deploy (test & prod)	S	R	A
Config Routes/Ingress + SSL	C	R	A
Provision PostgreSQL + SSL	C	R	A
Manage OpenShift Secrets	C	R	A
Rolling updates	R (images)	R (deploy)	A
App support L2/L3	R	S	I
DB backup & restore	C	R	A

R = Responsible | A = Accountable | S = Supportive | C = Consulted | I = Informed

Required Actions

Action items before deployment

#	Action	Client Platform	Prio
1	Create namespace kodemed-test on OpenShift	Client	High
2	Provision PostgreSQL 16 with SSL enabled	Client	High
3	Configure Docker repo in JFrog	Client	High
4	Configure Helm repo in JFrog	Client	High
5	Provide JFrog credentials to Mieres IT	Client	High
6	Adapt kodemed-ui for non-root nginx (SCC)	Mieres IT	High
7	Create Helm charts	Mieres IT	High
8	Create OpenShift Secrets (DB, encryption, JWT)	Client	High
9	Configure Routes WebSocket + sticky sessions	Client	High
10	Validate RASCI matrix with all stakeholders	Client IT	Medium