# Service Observability Stack for Kubernetes

*(Istio + OpenTelemetry + Tempo + Loki + Prometheus + Grafana)*

## Executive Summary

To enhance visibility, reliability, and performance of our microservices-based applications running on Kubernetes, we are deploying an open-source observability stack tailored for service mesh debugging, metrics, logs, and tracing. This stack will be installed on our on-premise infrastructure to reduce cloud costs and ensure compliance with data security requirements critical for our healthcare domain.

## Why This Matters for Us (Health Insurance Organization)

- Regulatory Compliance: Full control over data via on-premise deployment ensures HIPAA-aligned observability.  
- Cost Efficiency: Eliminates dependency on costly third-party SaaS solutions by using fully open-source tooling.  
- Improved MTTR (Mean Time to Resolution): Enables fast root cause detection during service degradation or outages.  
- 360° Observability: Unifies logs, metrics, and traces across services—critical for complex, distributed health applications.

## Stack Components and Benefits

|  |  |  |
| --- | --- | --- |
| **Tool** | **Purpose** | **Benefits to Org** |
| Istio | Service Mesh | Secure service-to-service communication (already deployed) |
| OpenTelemetry | Collects traces, logs, metrics | Vendor-neutral; integrates with multiple backends |
| Prometheus | Metrics collection and alerting | Visibility into application and infra performance |
| Tempo | Distributed tracing backend | Trace user journeys and pinpoint performance bottlenecks |
| Loki | Log aggregation | Scalable, low-cost centralized logging |
| Grafana | Dashboards for metrics/traces/logs | Single-pane-of-glass for health of apps and clusters |

## Deployment Strategy

- Platform: Kubernetes (existing clusters on on-site servers)  
- Storage Backend: Using MinIO (S3-compatible) deployed locally  
- Configuration: Helm-based installs, automated upgrades  
- Dashboards: Istio and app-specific dashboards in Grafana

## Expected Outcomes

* Proactive Monitoring: Immediate alerts and historical insights into degraded service performance.
* Faster Debugging: Correlate logs, traces, and metrics for a single request across services.
* Infrastructure Insight: Monitor K8s node, pod, and container health in real-time.
* Cost Control: Avoid recurring SaaS platform charges by leveraging free, open-source software and existing hardware.

## All Open-Source, No Licensing Costs

All tools in this stack are Apache 2.0 or MIT licensed, ensuring no licensing burden and complete ownership over our observability pipeline.