



OPCITO TECHNOLOGIES

Infrastructure Solutioning & Setup with Migration to Microservices Architecture

About The Customer

The customer is a technology platform provider focused on transforming how care is delivered. Their platform supports cloud-based infrastructure, AI analytics, and connected device integration to enable scalable, personalized care across acuity levels. As part of their care delivery approach, they enable remote patient management, transitions across care settings (hospital, home, post-acute), and in-home services coordination, all under a unified ecosystem. Their model aims to reduce readmissions, lower cost of care, and detect patient deterioration earlier.

Business Challenge

As the customer's technology platform expanded, their monolithic infrastructure began to limit scalability and flexibility. Deployments were becoming slower, and updates to one part of the system often affected others, making innovation cumbersome.

They needed a modern, cloud-native architecture capable of supporting independent service scaling and faster releases. The move from an on-premises environment to the AWS cloud was critical to achieving this.

At the same time, multiple CI/CD tools across teams were creating inconsistencies in deployment processes. The customer wanted a unified and automated CI/CD system to simplify service delivery and improve development velocity.

How Opcito Helped

To enable the customer's transition from a monolithic setup to a scalable, cloud-native ecosystem, Opcito's engineering team delivered a comprehensive modernization strategy. This involved designing a secure cloud architecture, implementing microservices with Kubernetes, unifying deployment pipelines, and establishing observability across the environment—all aligned to improve agility, scalability, and reliability.

- **Cloud-native infrastructure design and deployment:** Opcito's cloud experts architected and deployed a secure, cloud-native infrastructure on AWS entirely from the ground up. The team implemented best practices for network design, data protection, and resource optimization, creating a scalable and high-availability foundation that supported seamless migration from the existing on-premises systems.



This document is proprietary and confidential. No part of this document may be disclosed in any manner to a third party without the prior written consent of Opcito Technologies.

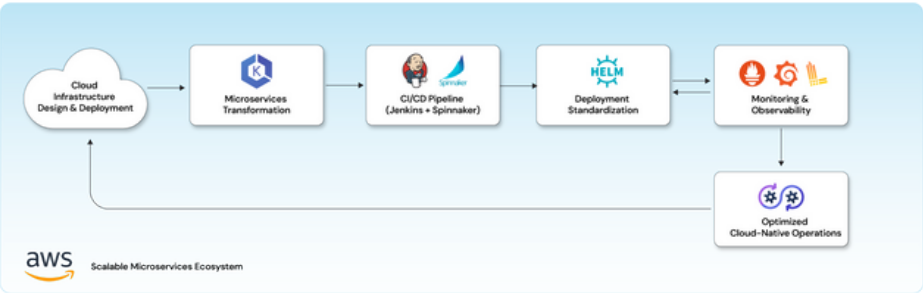
India office +91 (20) 6712 4100

US office +1 (650) 772 4442

How Opcito Helped

- **Microservices transformation with Kubernetes:** To break the constraints of the monolithic architecture, the application stack was containerized and deployed on Amazon EKS. This transition enabled independent scaling of services, improved fault isolation, and accelerated deployment cycles—allowing development teams to release features faster and with greater confidence.
- **Unified CI/CD pipeline implementation:** Opcito developed a robust, end-to-end CI/CD pipeline leveraging Jenkins and Spinnaker. The unified setup automated build, testing, and deployment workflows, replacing fragmented toolchains across teams. This standardization enhanced release speed, consistency, and overall delivery efficiency.
- **Helm-based deployment standardization:** To simplify deployment management, Opcito created a single Helm chart capable of managing multiple microservices. This reusable framework reduced configuration overhead, minimized human error, and ensured uniform deployments across environments, streamlining operations and onboarding for new services.
- **Monitoring and observability enablement:** The team integrated a comprehensive monitoring and observability stack using Prometheus, Grafana, and Loki. These tools provided actionable insights into system health and performance, enabling proactive issue detection and smoother operations across the Kubernetes ecosystem.

This complete transformation enabled the customer to shift from a monolithic architecture to a fully containerized microservices ecosystem on AWS, built for speed, scale, and stability.



Technologies, Tools, and Platforms used

- AWS (EKS, RDS)
- KUBERNETES (HELM CHARTS)
- JENKINS & SPINNAKER
- BITBUCKET
- PROMETHEUS & GRAFANA
- LOKI
- MYSQL
- MONGODB

Benefits

REDUCED MAINTENANCE COSTS	Using React cut long-term maintenance costs by boosting developer efficiency and minimizing reliance on specialized skills.
ENHANCED SCALABILITY	React's modular design allowed the app to scale easily with growing traffic and data needs.
IMPROVED PERFORMANCE	Lazy loading and code-splitting sped up load times and delivered a smoother user experience.
INCREASED PRODUCTIVITY	React's modern tools and workflows accelerated development and reduced time-to-market.