Ceebit: GenAI & Bedrock Operations Observability for an **Agentic AI SDLC Platform**

Client Overview

Ceebit is a low-code/no-code (LCNC) application development platform provider that extended its offering with an **agentic AI software development lifecycle (SDLC) capability built on Amazon Bedrock**. Ceebit had already engaged Inadev for a foundational DevOps engagement (EKS, Terraform, CI/CD, cost optimization) and returned to extend its observability practice specifically to cover its new GenAI operations.

Engagement Overview

- **Engagement Type:** Cloud Operations – GenAI/Bedrock Monitoring & Observability Implementation
- **Scope:** Extension of an existing DevOps observability foundation to add dedicated Bedrock/ GenAI operations monitoring, correlated logging, and cost/tagging governance.
- **Technology Stack:**



Key Business Challenge

As Ceebit's agentic AI SDLC platform scaled its use of Amazon Bedrock, the engineering team had no dedicated visibility into foundation model performance, token consumption, or cost — exposing the business to unpredictable latency, budget overruns, and undetected throttling.

- **No visibility into Bedrock-specific metrics such as token usage, time-to-first-token (TTFT), and invocation latency,** despite these directly driving user-perceived performance and cost.
- **GenAI workload health and cost governance lacked defined thresholds and escalation procedures** for high latency, elevated TTFT, excessive token consumption, throttling, and invocation failures.
- **Tagging and cost-governance discipline** needed to be enforced as GenAI workloads scaled across Ceebit's AWS accounts.
- As GenAI workloads expanded across multiple AWS accounts and environments, Ceebit required a **consistent multi-account operating model for centralized observability, security, access control, logging, cost allocation, and governance**. The absence of standardized cross-account visibility and controls increased the risk of fragmented monitoring, inconsistent configurations, and unmanaged operational or financial exposure.



Engagement Objectives

- **Extend Ceebit's existing DevOps observability** foundation to cover GenAI/Bedrock operations.
- **Establish real-time visibility** into Bedrock foundation model performance, cost, and reliability signals.
- **Maintain cost governance and tagging discipline** as GenAI workloads scale.
- **Establish a standardized multi-account strategy** for GenAI and Bedrock operations, including centralized observability, cross-account access controls, security and audit integration, cost allocation, and consistent tagging governance.



Proposed Solution

Inadev built a dedicated GenAI Observability layer on top of Ceebit's existing CloudWatch/Prometheus/Grafana/Loki foundation, purpose-built to track Bedrock operations and correlate them with the rest of the platform's telemetry.

CloudWatch GenAI Observability Dashboard

- Dedicated dashboard tracking token usage, TTFT, and invocation latency for all Bedrock-based agentic workflows.
- 5 CloudWatch alarms configured: invocation latency, TTFT, daily token budget, throttling events, and server errors.

Infrastructure & Cost Observability

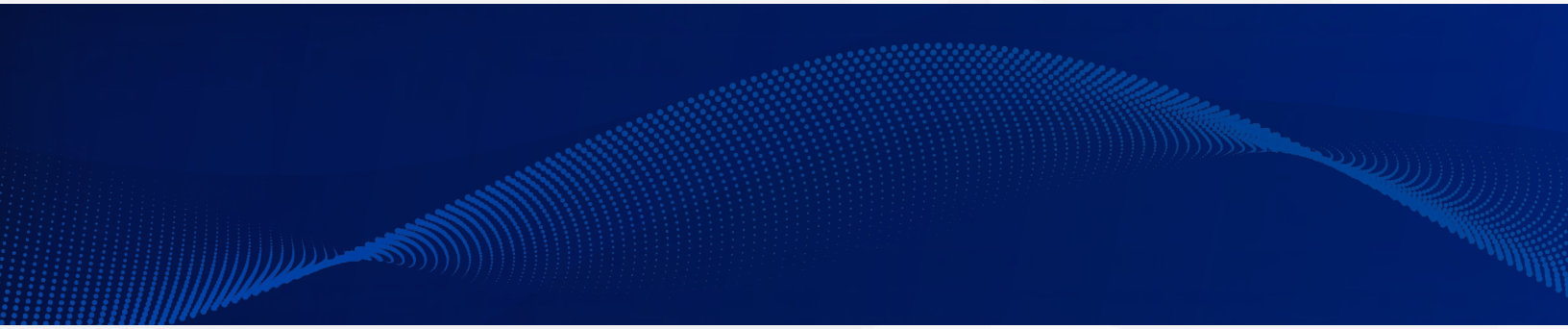
- Karpenter-managed EKS autoscaling behavior brought into the existing monitoring stack; KMS key usage (7 keys) and a 2-tag (Project, CostCenter) governance model enforced via compliance dashboards.

Architecture Overview

Ceebit's agentic AI SDLC platform runs on Amazon EKS with Karpenter-managed autoscaling, invoking Amazon Bedrock foundation models for AI-driven development workflows.

A dedicated CloudWatch GenAI Observability dashboard captures token usage, TTFT, and invocation latency directly from Bedrock, with 5 alarms covering latency, TTFT, daily token budget, throttling, and server errors. Bedrock request IDs are available in structured CloudWatch invocation logs, while Grafana and Loki separately provide application and pod-level observability.

KMS (7 keys) and a 2-tag governance model provide encryption and cost-attribution controls across the environment.



Security, Governance & Compliance

- **7 KMS keys actively monitored for encryption** key usage across the GenAI and platform infrastructure.
- **2-tag governance model** (Project, CostCenter) enforced via automated compliance dashboards.
- **Anomalous Bedrock usage patterns** alerted on as an early indicator of potential cost or security anomalies.
- **CloudTrail retained for full account-level audit visibility** alongside the GenAI-specific observability layer.

Results & Key Performance Indicators

The engagement was measured against the following key performance indicators:

KPI	Baseline	Target	Actual
Mean Time to Resolve (MTTR)	Baseline from prior DevOps engagement (ad hoc, pre-GenAI-observability)	Reduce by **30%**	**40%** reduction achieved
Deployment Failure Rate	Frequent failures pre-observability rollout	Reduce by **50%**	**80%** reduction achieved

Business Impact

Release velocity

Release cadence improved from every 4–6 weeks to weekly, enabled by faster detection and resolution of GenAI and infrastructure issues.

Cost efficiency

30% overall cost savings realized through GenAI usage visibility and tagging-driven cost governance.

Lessons Learned & Continuous Improvement

- **GenAI workloads need purpose-built observability** — generic infrastructure dashboards did not surface token, latency, or cost signals that matter most for Bedrock-based applications.
- **Tagging discipline had to be enforced proactively** (via dashboards) rather than retroactively, once GenAI workloads began scaling cost impact quickly.
- **Building on an existing DevOps observability foundation** significantly accelerated delivery of the GenAI-specific layer versus starting from scratch.