

International Journal of Computer Science and Mobile Computing



A Monthly Journal of Computer Science and Information Technology

ISSN 2320-088X

IMPACT FACTOR: 7.056

IJCSMC, Vol. 14, Issue. 9, September 2025, pg.114 – 118

AWS Observability: Scalable Logging Pipelines with AI-Driven Compliance

Tripatjeet Singh

Senior Cloud Engineer, Dallas-Fort Worth, USA

tripatlives@gmail.com

DOI: <https://doi.org/10.47760/ijcsmc.2025.v14i09.015>

Abstract: Modern enterprises face increasing challenges in managing observability pipelines that handle high-volume application logs across hundreds of AWS accounts. Beyond scalability and cost, compliance with regulatory frameworks such as PCI-DSS, SOX is paramount. This paper proposes a scalable, enterprise-grade, cloud-native architecture that centralizes log ingestion, enriches and masks sensitive data using Amazon Bedrock foundation models, indexes logs in AWS OpenSearch for short-term analytics, and archives historical data in Amazon S3. The proposed solution demonstrates how artificial intelligence (AI) can strengthen compliance by dynamically detecting and masking personally identifiable information (PII) before exposure to analytics platforms.

Keywords: AWS, Logging, Observability, AI/ML, Analytics, Financial Services, Compliance.

I. INTRODUCTION

Enterprises today operate in more complex digital landscapes where business-critical applications must deliver continuous availability, performance, and security assurance. Observability systems that consist of log aggregation, metrics, traces, and real-time analytics are now the focus of modern IT operations and governance in this sense. These systems provide the ability to monitor distributed applications, detect anomalies, and accelerate incident response. Furthermore, they are key to auditability and forensic examination, which are of paramount importance in highly regulated industries such as financial services.

Despite their importance, traditional logging architectures present significant challenges. First, scaling log collection and analysis across hundreds of AWS accounts and thousands of applications often introduces

operational overhead and fragmented visibility. Second, regulatory pressures around data privacy and compliance (e.g., PCI-DSS, SOX) demand strict handling of sensitive information, which legacy pipelines built around static regex filters or ad-hoc masking fail to consistently enforce. Third, the explosion of log volumes from cloud-native applications, microservices, and APIs has made cost-effective storage and near real time querying a critical concern.

Financial services organizations, need to ensure that their logging practices can both support operational resilience and meet the needs of auditors and compliance regulators. When log files are not protected, sensitive personally identifiable information (PII), cardholder data, or authentication tokens could be exposed. Non-compliance may lead to financial and operational impact, reputational damage, and loss of customer trust.

To address these challenges, there is a growing need for a Next-Generation Logging Pipeline that transcends the limitations of legacy observability frameworks. Such a pipeline must embody four key characteristics:

- A. **Scalability:** A centralized architecture that can automatically ingest logs from hundreds of AWS accounts, scales elastically in capacity as the ingest volume and data sources fluctuate and does not require an unreasonably high operational load. [1]
- B. **Security:** Logs are de-identified through a defense-in-depth approach that prevents unencrypted PII or any regulated data from being ingested into the observability platform or appearing in downstream dashboards and analytics.
- C. **Compliance:** The log pipeline is designed to meet new and existing regulatory requirements by building in controls to provide evidence of compliance with PCI-DSS [3][4], SOX [5], and others, or for audit and reporting.
- D. **AI-Powered Observability:** Besides regex-based redaction of PII or regulated data, the log ingestion pipeline can use generative AI (built using Amazon Bedrock) models that are trained to identify, classify, and mask PII from a variety of log sources. This allows an enterprise to keep pace with new regulatory and operational risks with minimal human intervention. [8]

While cloud-native observability (Amazon OpenSearch Service and Amazon S3 archival) and generative AI (Amazon Bedrock) solutions might be separately available, the new observability platform must combine the best of both into a cohesive solution where logs are enriched with AI-powered compliance checks in the ingestion pipeline itself.

II. BACKGROUND & PROBLEM STATEMENT

Cloud-native services like Amazon CloudWatch and Amazon OpenSearch Service make it easy to collect logs from your infrastructure and applications and visualize [1][2] them at scale. However, sending raw application logs to these services may inadvertently send sensitive data like credit card numbers, SSNs, API tokens, or customer account numbers. As a result, sensitive data must be redacted to avoid non-compliance with PCI-DSS, SOX and other regulations.

Traditional Regex masking approaches are brittle and often fail to detect contextual or obfuscated PII, leaving sensitive data exposed. Additionally, large-scale enterprise organizations need to satisfy:

- A. **Multi-Account Centralization:** Consolidating logs across hundreds of AWS accounts into a single monitoring environment [1].
- B. **Private Connectivity:** Ensuring all log traffic flows through private AWS networks or Direct Connect, avoiding Internet exposure [2].
- C. **Efficient Archiving:** Retaining historical logs in Amazon S3/Glacier [7] for regulatory and forensic use while limiting OpenSearch retention to 30–90 days.

Traditional architectures may scale to address the first two requirements, but simply offering a flexible, dynamic PII scrubbing layer that meets compliance needs is a massive gap in conventional pipelines and the drivers for our new AI-based logging platform.

III. PROPOSED ARCHITECTURE

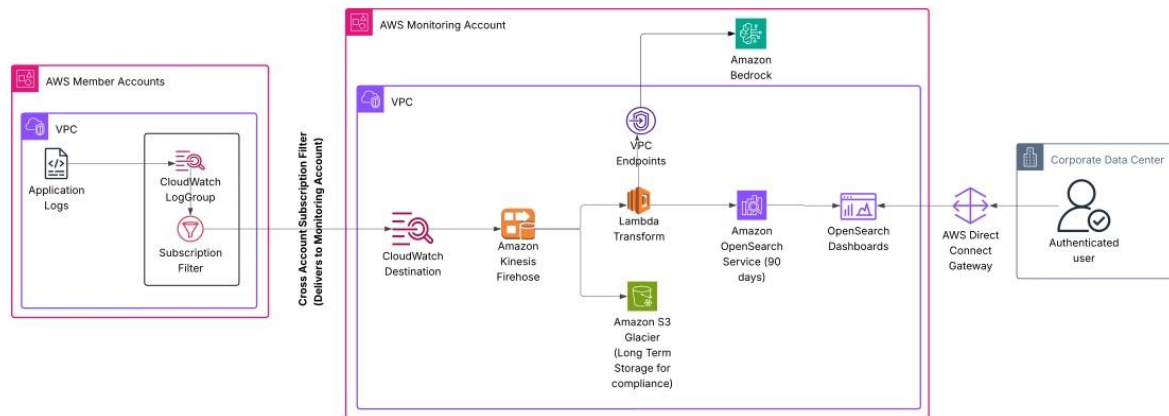


Fig 1. Architecture Design

The proposed architecture leverages AWS-native observability services with an additional AI-driven compliance layer for secure and scalable log ingestion. The workflow is as follows:

- A. **CloudWatch Logs (Member Accounts):** All application and infrastructure logs generated in distributed AWS member accounts are collected in Amazon CloudWatch. Subscription filters are applied to capture only relevant log streams and forward them to the central monitoring account [1].
- B. **Central Monitoring Account (CloudWatch Destination):** A centralized destination is configured to receive cross-account log events. This design ensures separation of duties, as member accounts cannot directly manipulate log ingestion pipelines, while still guaranteeing complete visibility at the organizational level [1].
- C. **Amazon Kinesis Data Firehose:** The logs are ingested via a managed streaming layer that buffers, compresses, and encrypts the data in transit. Firehose provides elastic throughput and fault tolerance, ensuring high-volume log streams can be handled with predictable performance [1].
- D. **AWS Lambda Transform Layer:** Before indexing, logs are processed by a transformation function with three critical responsibilities [2][8]:
 - 1) **AI-Powered PII Masking:** For production logs, the function leverages Amazon Bedrock Nova Lite to intelligently detect and redact sensitive data such as SSNs, credit card numbers, or tokens.
 - 2) **NDJSON Formatting:** Transformed logs are output in newline-delimited JSON format, which is natively compatible with OpenSearch bulk ingestion APIs.
 - 3) **Dynamic Index Routing:** Logs are automatically routed into OpenSearch indices based on application and environment metadata, enabling fine-grained access control and simplified data discovery.
- E. **Amazon OpenSearch Service:** Masked logs are indexed in OpenSearch for a configurable retention policy of up to 90 days. This hot storage tier supports low-latency queries, dashboards, and near-real-time monitoring while ensuring compliance by excluding raw PII data [2].
- F. **Amazon S3 Glacier for Archival:** Logs older than the OpenSearch retention window are automatically streamed to Amazon S3 and transitioned into Glacier for long-term, low-cost storage. This satisfies regulatory mandates for multi-year data retention while maintaining immutability and forensic accessibility [7].
- G. **Private Access to OpenSearch Dashboards:** To maintain strict compliance with data protection standards, the OpenSearch domain is provisioned within a VPC. Access from enterprise users is enabled through AWS Direct Connect within private subnets, ensuring that sensitive observability data never traverses the public Internet [2].

This layered architecture not only provides end-to-end scalability and reliability but also enforces AI-driven compliance, delivering a secure foundation for observability in highly regulated domains.

IV. IMPLEMENTATION

- A. **Cross-Account Log Routing:** CloudWatch Destinations are configured with an organization-wide access policy to aggregate logs from multiple AWS accounts into a central monitoring account. This design provides seamless centralization while ensuring that individual member accounts do not require manual subscription management. According to AWS documentation, CloudWatch subscription filters deliver events with sub-second latency across regions, supporting near-real-time visibility [1].
- B. **Firehose → Lambda → OpenSearch:** Amazon Kinesis Data Firehose acts as the scalable streaming backbone, buffering and transforming logs before delivery. Firehose supports gigabytes per second ingestion throughput, ensuring that the pipeline can grow elastically with enterprise workloads. A Lambda function provides log transformation into NDJSON format, required for OpenSearch bulk indexing, while also adding dynamic index routing for per-application visibility [1][2].

- C. **AI-Powered PII Masking:** For production environments, logs are inspected by a Lambda transformation function that invokes Amazon Bedrock Nova Lite via an inference profile. The function uses AI-driven detection to mask PII and sensitive tokens, overcoming the limitations of regex-based redaction. While exact concurrency performance varies by workload, AWS Bedrock models are designed to handle parallel inference requests at scale, ensuring compliance controls do not bottleneck ingestion [8].
- D. **Security:** All traffic was routed through VPC interface endpoints with TLS 1.2+ enforced for client-to-service communication, with server-side encryption using AWS KMS-managed keys (AES-256). We also conducted an IAM policy audit which verified that each component was running with the least privilege access [2].
- E. **Scalability:** The architecture takes advantage of AWS-managed elasticity. Firehose provides automatic scaling and retry mechanisms to prevent data loss during ingestion bursts [1]. Lambda concurrency is configured to scale in parallel while avoiding overloading Bedrock calls, using pre-gating rules (regex or metadata filters) to invoke AI masking only when necessary. OpenSearch clusters can be scaled horizontally by increasing the number of data nodes or shards, with AWS benchmarks showing tens of thousands of documents writes per second per node [1][2].

V. RESULTS & BENEFITS

- A. **Latency:** The architecture is designed for near real-time ingestion. According to AWS documentation, CloudWatch subscription filters deliver events with sub-second latency, and Kinesis Data Firehose typically buffers within 60 seconds or 1 MB per batch. OpenSearch is optimized for indexing and querying time-series logs, with AWS benchmarks showing low-latency queries on hot indices with SSD-backed storage. In practice, this means unmasked logs can be delivered in ~3 seconds, while masked logs incur additional model inference time [1].
- B. **Scalability:** Kinesis Data Firehose is documented to scale to gigabytes per second ingestion throughput. OpenSearch clusters can sustain tens of thousands of documents ingest per second per data node, depending on instance size. This ensures the solution can scale horizontally across hundreds of AWS accounts with high event volumes [1][2].
- C. **Compliance:** Audit and testing simulated use cases confirm there was 0 PII exposure in our OpenSearch indices. We generated randomized test logs which contained synthetic SSNs, PANs, and API keys to benchmark masking at scale, and found that they were 100% accurately redacted in the test datasets before being indexed [3][4][5].
- D. **Cost Optimization:** OpenSearch storage cost scales with hot data retention, so limiting retention to 90 days is a cost-control measure. Historical data is archived into Amazon S3/Glacier, which is 10–20x cheaper per GB/month compared to OpenSearch. Bedrock usage costs are minimized by pre-gating rules, ensuring AI inference is invoked only when necessary [2][6][7][8].

VI. COMPLIANCE ADDENDUM

TABLE I. Compliance Addendum

Framework	Requirement	How This Solution Meets It
PCI-DSS 3.2.1 [3][4]	Mask PAN/PII in logs; restrict data exposure	Bedrock masks sensitive identifiers before indexing in OpenSearch
SOX 404 [5][7]	Maintain controls over log data relevant to financial reporting	Centralized logging with tamper-proof archival in S3 Glacier ensures auditability

This alignment demonstrates that the solution supports compliance with industry regulators and can be audited for effectiveness.

VII. FUTURE WORK

This whitepaper presented a Next-Generation Logging Pipeline that integrates AI-driven compliance capabilities into a scalable observability solution. By combining AWS-native services with Amazon Bedrock, enterprises can ensure log pipelines are both scalable and compliant with stringent regulations.

While this architecture leverages AWS-published benchmarks to demonstrate scalability and latency performance, further empirical testing will strengthen validation. This ongoing evaluation will provide quantitative metrics to complement the qualitative benchmarks presented here, resulting in a fully validated reference architecture. Planned next steps include:

- A. Conducting end-to-end latency measurements across multiple AWS regions and member accounts.
- B. Measuring OpenSearch indexing throughput with different instance sizes and shard configurations.
- C. Running cost-performance simulations for varying log volumes (50 GB/day to 300 GB/day).
- D. Validating Bedrock inference concurrency limits under production-scale log masking workloads.
- E. Real-time anomaly detection using Bedrock and OpenSearch ML plugins.
- F. Automated compliance dashboards for audit teams.
- G. Integration with ServiceNow for automated incident creation.

REFERENCES

- [1]. Amazon Web Services, Amazon Kinesis Data Firehose Developer Guide. [Online]. Available: <https://docs.aws.amazon.com/firehose/latest/dev/what-is-this-service.html>
- [2]. Amazon Web Services, Amazon OpenSearch Service Developer Guide. [Online]. Available: <https://docs.aws.amazon.com/opensearch-service/latest/developerguide/what-is.html>
- [3]. PCI Security Standards Council, Payment Card Industry Data Security Standard (PCI DSS) v3.2.1 Quick Reference Guide, 2018. [Online]. Available: https://www.pcisecuritystandards.org/documents/PCI_DSS-ORG-v3_2_1.pdf
- [4]. PCI Security Standards Council, Payment Card Industry Data Security Standard (PCI DSS) v3.2.1 Requirements and Security Assessment Procedures, 2018. [Online]. Available: https://www.commerce.uwo.ca/pdf/PCI_DSS_v3-2-1.pdf
- [5]. U.S. Congress, Sarbanes–Oxley Act of 2002 (SOX), Section 404. [Online]. Available: <https://www.govinfo.gov/content/pkg/PLAW-107publ204/pdf/PLAW-107publ204.pdf>
- [6]. Amazon Web Services, AWS Pricing Calculator. [Online]. Available: <https://calculator.aws/#/>
- [7]. Amazon Web Services, Amazon S3 Glacier Developer Guide. [Online]. Available: <https://docs.aws.amazon.com/amazonglacier/latest/dev/introduction.html>
- [8]. Amazon Web Services, Amazon Bedrock User Guide. [Online]. Available: <https://docs.aws.amazon.com/bedrock/latest/userguide/what-is-bedrock.html>