

# Unified Log Management: Kafka Connect and Data Lakes for Advanced System Analysis and Machine Learning

Kuriens Shaji Maliekal<sup>1</sup>

<sup>1</sup>Staff Software Engineer, Walmart Global Tech, USA

\*\*\*

**Abstract** - The exponential growth in modern systems has significantly transformed how organizations monitor, optimize, and understand their operations. Logs, which record critical facts concerning system performance, user interaction, and infrastructure behavior, play a critical role in diagnosing problems, auditing activities, and facilitating advanced analyses such as predictive modeling for machine learning. Nevertheless, with the rising velocity and volume of logs coming from distributed systems and micro services, conventional log management techniques struggle to aggregate, store, and analyze logs. Previously, centralized logging systems based on relational databases or file storage were employed to collect logs. Although these systems served for small-scale applications, they turned into bottlenecks against the demands of modern distributed systems. Relational databases have scalability challenges, whereas file-based systems don't support complicated analytics or structured queries. Such restrictions hinder real-time insights and consequently influence decision-making processes within time. Integrating Apache Kafka with data lakes has proven to solve such constraints by providing real-time log streaming and affordable and scalable storage. Yet, the problems of data quality, governance, and consistency still exist.

This study proposed a unified log management framework leveraging Kafka connect to stream logs into data lakes while ensuring data quality and governances. By implementing automated de duplication and schema evolution within ETL pipelines, this approach addresses key challenges and enables both real time processing and long-term analytics. It unified real time monitoring with predictive maintenance and anomaly detection, optimizing resource utilization in cloud native environment. Future advancement in schema management, data validations and machine learning (ML) integration helps streamline real-time insights and predictive capabilities for improved decision making and operational efficiency.

**Key Words:** Log Management, Kafka, Kafka Connect, Data Lake, ML, Machine Learning, Distributed Systems

## 1. INTRODUCTION

The exponential growth of data in modern systems has transformed the way organizations understand, monitor, and optimize their operations. Logs, as a critical subset of this data, serve as the pulse of application and infrastructure, capturing important details about system behavior,

performance and user interaction. They are invaluable for diagnosing issues, auditing activities, and powering advanced analytics such as predictive modelling in Machine Learning (ML). However, the sheer volume and velocity of logs generated by distributed systems and micro services present significant challenges in aggregation, storage and analysis. Traditionally, log management depended on centralized logging system that gathered logs from multiple sources and saved in relational database or file systems. Although effective for smaller scale applications, these methods faced challenges with the requirements of contemporary distributed systems. Relational databases frequently turned into bottlenecks because of their restricted scalability and lack of capacity for large-scale real-time data ingestion. File-based systems were not advanced enough for structured queries and complex analytics. Additionally, conventional systems were poorly suited to process unstructured or semi-structured log data produced by contemporary applications. These restrictions hindered immediate insight into system performance and postponed essential decision-making processes [1].

The trend of late has been to increasingly focus on smarter, more scalable, more efficient possibilities for managing logs. One of the emerging fundamental components is Apache Kafka, which has become the cornerstone standard for real-time data streaming and log aggregation. With its distributed architecture, Kafka can handle the massive amounts of logs generated by today's applications, as it can process logs quickly at high throughput, while providing reliability, data integrity and scalable solutions for more processing workers [2]. Kafka can allow users to have logs streamed to downstream applications for processing or storage with real-time ingestion from various sources or application pods in Kubernetes clusters which have generated logs. While Kafka is optimized for real-time ingestion and streaming, it was not designed for storage or complex analytics of the ingested logs. These constraints can be addressed by the idea of combining Kafka with data lakes [3].

Data lakes allow for a cost-efficient and scalable store where raw logs can be kept in their original form. Amazon S3, Google Cloud Storage (GCS), and Hadoop based environments provide virtually unbounded capacity at comparatively affordable prices. Using Kafka Connect, a Kafka framework for external system integration, logs can be natively streamed from Kafka topics to data lakes. This not only centralizes storage but also separates ingestion from

downstream analysis processes, allowing for flexibility and scalability[4].

Existing literature shows the benefits of data lakes with Kafka. Studies show that integrating Kafka and data lakes allows organizations to retain historical logs for extended periods of time without sacrificing performance and without costing an arm and a leg [5]. There is also a wide array of use cases we can take advantage of, ranging from real time monitoring to batch analytics and machine learning model creation. While there are still challenges regarding data quality, governance and consistency between these two systems, many questions remain. Some of these include how we can protect against the data lake fragmentation which can result in records losing their integrity, or de-duplication methods that could produce logs with errors in downstream if either schema governance or de-duplication aren't in place [6] [7].

The proposed study aims to address these challenges by establishing a unified log management framework that leverages Kafka Connect to stream logs into data lakes while maintaining data quality and governance standards. By implementing schema evolution techniques and automated de-duplication processes within the ETL (Extract-Transform-Load) pipeline, this framework ensures that logs are stored in a structured and consistent manner. Additionally, the study explores how machine learning models can be trained on historical log data stored in data lakes to enable predictive analytics.

One significant contribution of this study is its focus on real-time streaming capabilities combined with long-term analytical potential. While traditional approaches often required separate pipelines for real-time monitoring and historical analysis, this framework unifies both use cases within a single architecture. Logs ingested through Kafka are immediately available for real-time processing while simultaneously being archived in the data lake for future analysis. This dual capability is particularly valuable for applications such as anomaly detection in system behavior or predictive maintenance in industrial settings.

Another key aspect of this study is its emphasis on leveraging cloud-native technologies to simplify deployment and scalability. By utilizing managed services like Amazon Managed Streaming for Apache Kafka (MSK) or Google Cloud Pub/Sub alongside cloud-based data lake solutions like S3 or GCS, organizations can reduce operational overhead while achieving high availability and fault tolerance. The study also examines best practices for optimizing resource utilization in such environments, including techniques like partitioning logs based on metadata attributes to improve query performance [8].

Furthermore, the integration of Kafka with data lakes offers a robust foundation for machine learning applications. Historical logs stored in data lakes can be used to train

models that predict system failures, detect anomalies, or optimize resource allocation. This predictive capability is crucial for maintaining high uptime and performance in modern distributed systems. Moreover, by leveraging data lakes as a centralized repository, organizations can apply machine learning across multiple applications and domains, fostering a data-driven culture that enhances decision-making at all levels.

In addition to its technical contributions, this study also explores the organizational and operational benefits of unified log management. By centralizing log data and providing real time insights, the organization can improve collaboration between development, operations, and security teams. This alignment is important for modern DevOps practices, where continuous integration and delivery requires seamless communication and feedback loops. Moreover, the ability to analyze logs in real time support agile development methodologies, allowing developers to quickly identify and rectify issues during the development cycle.

The importance of data and logs in understanding system and application behavior cannot be overstated. Logs provide a detailed record of user interaction, system events, and performance metrics, which are crucial for auditing, compliance, and security monitoring. In regulated industries such as finance and healthcare, logs serve as a legal business imperative, capturing the tamper-resistant record of the transactions and the activities. Additionally, logs are critical in machine learning application scenarios, when past data are employed to calibrate models used to forecast what will happen or identify anomalies in the future.

Kafka is such an important part of log streaming due to its ability in leveraging the characteristics of low latency logs with significant volumes of data. Because Kafka is distributed, it handles logs and consumes them in real-time, rather than batch processed logs. These respective needs generate an imperative for real-time actions based on system events - considering the time pressures associated with financial trading platforms and other analysis needs that are real-time in nature. Additionally, coupling Kafka with data lakes even brings the ability to launch real time analysis with a long-term storage component, since data lakes can be a long-term log ingestion and log analytics solution where the long-term diagnostic capacity supports up-to date operational needs and decision-making for the organization. Kafka Connect is also very functional for streaming logs into data lakes. As the bridge, Kafka Connect provides an interface to make it much more flexible for incorporating the use of Kafka data into other systems. In any other need for data pull from Kafka, organizations can also determine how to configure the flow of logs and in a much broader scope of data lake technologies depending on the characteristics of the existing systems and their needs. For example, organizations already invested in the Amazon Web Services (AWS) ecosystem might prefer S3, while those

using Google Cloud Platform (GCP) might opt for GCS. This choice enables organizations to leverage their existing investments while still benefiting from the scalability and flexibility of Kafka.

From the analysis and machine learning perspectives, logs amassed in data lakes present a rich dataset for model training for predictive models. With the use of methods like natural language processing (NLP) and time-series analysis, organizations can tap into log data insights that were not exploitable in the past. For example, anomaly detection models can be trained to detect unusual trends in system activity, enabling proactive maintenance and minimizing downtime. Likewise, predictive models can predict resource usage based on past trends, allowing for better resource allocation and cost savings. The suggested framework also deals with the issue of data governance and quality in log management. Through the application of strong schema management and data validation procedures, organizations can ensure that logs are stored in a structured and consistent manner. Consistency is essential for downstream analytics since it supports reliable querying and analysis across datasets. In addition, applying data governance rules to log data allows organizations to comply with regulations and preserve data integrity in the long term.

## 2. Why Logs Matter in Modern Software Systems

Logs play an important role in modern software systems by serving as a critical lens through which developers and administrators can observe, interpret, and manage application and system behavior. As these systems grow in complexity, effective logging has become essential for maintaining visibility, performance and security. Logs not only provide a detailed view into application and system behavior but also support critical functions such as troubleshooting, compliances, and advanced analytics, including ML driven insights.

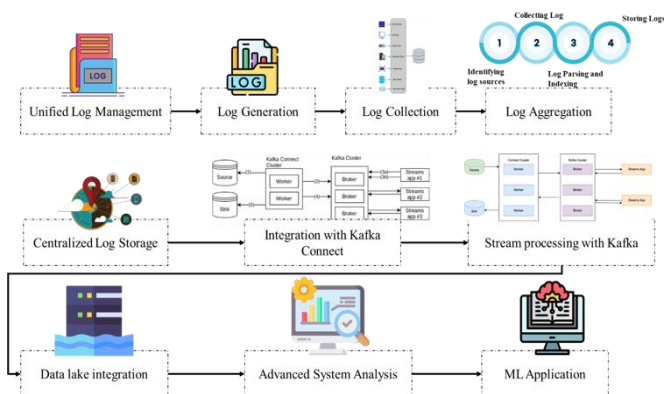


Figure 1. Unified Log Management to Insights

### 2.1. Application and System Behavior Monitoring

Logs serve as a reflective surface for the health and performance of the application, which is important in

identifying and troubleshooting problems. As the prevailing study [9], with ML applications, continuous oversights is critical in identifying unforeseen challenges occurring in production. Logs generate and collect information that can help identifying the difference made to application behavior, final anomalies, and to understand the factors affecting the performance of ML systems. This information is very important to decrease risk and to improve how the system can operate. In addition, as IoT platforms grow more complex [10], they can be augmented by log data to enable real time monitoring of distributed systems. Using tools such as elastic stack (Elastic search, Kibana beats) and Apache Kafka, IoT platforms utilize logs to monitor system performance from edge nodes to cloud servers to data streams, facilitating effective problem-solving and decision-making. Logs offer a formatted means to see platform health, so administrators and users get immediate access to important performance information.

According to [11], a key advantage of having access to large quantities of log data produced by different applications, is our ability to utilize it for real-time anomaly detection, early maintenance dispatchment, and for prediction of future anomalies. By utilizing open-source technologies, such as, Kafka, PySpark and Grafana, there are scalable and fault tolerant log analytics architectures which provide organizations capability of intuitive visualization and actionable insights within timelines. Log data is reflective of application health and has the capability to reverse track a problem to its source and support faster troubleshooting and debugging. In case of the scalable data pipeline [12], it is important to note that logs are important in keeping the performance of distributed systems that process a great deal of data. This may include using tools like Apache Kafka that allow event-driven architecture, and Apache Spark for distributed processing. Logs allow data to flow efficiently through the data pipeline from ingestion through analysis. Logs also help identify blockages or errors in the data processing that would help organizations maintain efficient data pipelines.

Likewise the study [13] discusses the architecture of multi cloud disturbed systems where logs assist in monitoring system wellness in varied platforms. Logs are employed to monitor the data fragmentation, latency, and system crashes, optimize resource utilization, identify problems, and ensure fault tolerance. Here, logs become critical in the aspect of delivering real-time measurements to guarantee the smooth functioning of multi-cloud distributed systems.

Logs are also extremely important in cyber security [14], mainly in using cyber threats intelligence (CTI) from open source data to seek and analyze logs from a variety of internet based sources in real time for potential indicators of compromise (IOCs) using Natural Language Processing (NLP) for classifying and extracting IOCs. The use of systems such as Apache Kafka for message processing guarantees efficient processing of logs from different sensors, enabling

timely response and data backup in the event of server failure. This assists with keeping operations safe and efficient in situations where real-time information is most important. In the application where monitoring in real time is critical, for example air pressure monitoring in petroleum and mining industries [15], logs enable a means of identifying anomalies in sensor readings.

Similarly, the study [16] affirms the value of logs in Information System Security (ISS) by giving decision-makers the capability of reviewing risks and maintaining software security overall. Since taskforces are using logs as data in the Fuzzy Analytic Hierarchy Process (Fuzzy-AHP) framework, they can use security logs to prioritize security risks and consider potential means of software maintenance overall, ultimately promoting system security from a holistic lens. Similarly, [17] elaborates further by stressing how software logging facilitates better software maintenance, particularly for large systems, where logging user-based activities allows prioritizing those areas needing refining. With an effective logging facility that is able to record key HTTP requests and user actions, a way can be discerned on how parts of the software are being utilized most intensively, making for more intelligent decisions regarding allocation in maintenance.

Ultimately, logs convert raw event in the system into insights that can be acted upon at all levels, from real time incident response to historical optimization and predictions about future behavior. Logs play a vital role in ensuring secure, truthful, and high performing software systems.

### 3.2. Audit and Compliance Use Cases

Logs are used as historical records for audits, serving an important function to ensure compliance in industries. Logs are particularly important in regulated industries, where they serve as a way to track system activity and user activities, enabling organizations to comply with industry specific regulation and standards.

The project presented in this dissertation undertook to develop an automated data ingestion system that makes use of Kafka Connect for transferring data from different types of data structures like databases, cloud storage, real-time streams into a predictive analytics system. The automated data ingestion allows flexibility and scalability through the use of connectors, which allow transformation and enrichment of data whilst data flows through the pipeline. The overall architecture enables [18] real-time processing and integration of predictive analytics into everyday business decision making that relies upon the ability of an organization to have access to predictable trends that can be informed instantly. The project outlines methods for the environment setup of a data lake, connector configuration and predictive model development, with targeted intervention for issues related to data quality and processing latency.

Data engineering became more complex and large scale with the development of modern software applications. The existing study [19], provided an extensive review on the evolution, fundamental elements, technological advancement, and future trends in data engineering in connection with software development. The study investigated the integration of AI within cloud-native systems, processing frameworks, and real-time data engineering. Issues of scale and security were addressed, together with workflow optimization techniques. The article also contained results showcased in data tables and working code snippets to provide practical advice for practitioners and researchers alike. Today's large organizations have just a tremendous challenge to manage and analyze massive quantities of financial data in order to take strategic decisions and stay competitive. Traditional data warehouse solutions frequently did not meet modern financial analytics volume, complexity, and performance needs. This research [20] examined architectural principles, technical solutions, and best practices that are necessary for developing scalable data warehouses that meet the needs of financial analytics. It looked at the data integration directions, performance optimization techniques, security structures, and regulations in compliance. By conducting thorough technological analysis of real-life case studies, this paper provides practitioners with a structural road map for designing and implementing solid, scalable, and secure data warehousing capabilities.

The explosive growth of data has necessitated scalable data pipelines to effectively manage, process, and analyze complex, large-scale data streams. [21] delved into some of the fundamental architectural principles and design patterns involved in constructing batch and real-time streaming pipelines and solved challenges such as data integration, fault tolerance, and scalability with contemporary data engineering tools. Practical case studies illustrated real-world methods for architecting pipelines to satisfy the requirements of big data environments.

Integrating Data Engineering and MLOps proved important for building efficiency, secure ML pipelines where data engineering managed data ingestion and transformation while MLOps streamlined model deployment, monitoring, and governance. The paper [24] identified architectural patterns and tool chains that addressed data management, workflow interruptions, and ethical concerns, exploring advanced techniques like pipeline design, redundancy, and server less architectures to optimize ML workflows. In parallel, a unified log management system [25] based on message queues outperformed traditional solutions in performance, security, and scalability, enabling centralized analysis and intelligent security operations for large-scale log data.

### 2.3. Logs as Fuel for Machine Learning

Logs have emerged as a foundational resource for ML in modern intelligent systems, acting as the baseline for pattern recognition, behavior modelling, and predictive analytics [26]. The rapid increase in data generated across a broad range of connected devices, sensors, and digital services has resulted in organizations having available to them unmanageable streams of log data identifying their every interaction, transaction and event in a system which is an excess of log data at our disposal [27]. This stream of log data is vital for the development of machine intelligence as an abundance of data enables models to identify intricate patterns and model the behavior of users or systems at a level of scale and granularity never experienced before [28]. As an example in smart cities and transportation systems, logs from millions of individual journeys data are used to reconstruct travel patterns, optimize routes, and predict patterns, providing direct input to support AI and ML improvements in operational activity and user experience.

The use of unified log management platforms, including those that use Apache Kafka and data lakes, have also increased the value of logs for ML use [29]. These architectures support the effortless collection, aggregation, and storage of heterogeneous log data from various sources, enabling real-time analytics and machine learning pipelines. By streaming logs into elastic data lakes, enterprises can process and convert raw data cost-effectively, rendering it feasible for sophisticated ML models for anomaly detection, predictive maintenance, and automated decision-making in industries such as transportation, manufacturing, and cyber security [30]. Real-time correlation and analysis of multi-modal log data are essential to detect latent anomalies, predict failures, and address emerging threats in advance [31].

Despite these technological advances, several limitations persist in leveraging logs for machine learning. Security and privacy concerns arise due to the sensitive nature of log data, necessitating stringent access controls and encryption mechanisms to protect information [32] [33]. Moreover, the high dimensionality and noise inherent in log data can complicate model training, leading to potential false positives or overlooked anomalies. Data governance and compliance, especially in regulated sectors like finance and transportation, impose additional constraints on data retention and usage [34]. Addressing these issues demands a combination of automated lifecycle management, integrated lineage tracking, and ethical considerations in data handling to ensure trustworthy and effective ML-driven insights. Continued research and innovation in unified log management and MLOps frameworks are vital to overcoming these challenges and unlocking the full potential of logs as a fuel for machine learning [35].

### 3. Apache Kafka for Real-Time Log Streaming

Apache Kafka is now an industry-standard technology for real-time log streaming, providing a scalable, high-throughput platform that effectively ingests and processes logs from multiple systems and applications. Contrary to logging systems that come in common use, Kafka's architecture is scalability and reliability-focused to facilitate effortless log aggregation and analysis even in complicated, containerized environments. By categorizing log data by topics and partitions, Kafka enables effective handling and retrieval of logs from many application pods or nodes, for use in both operational monitoring and analytical applications. For this reason, Kafka is an excellent backbone to use with centralized log management across today's dynamic infrastructures.

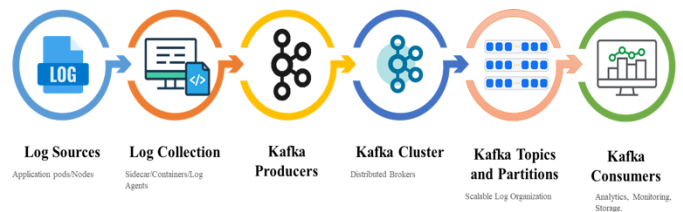


Figure 2. Apache Kafka for real time Log Streaming

#### 3.1 Role of Kafka in Log Ingestion

Apache Kafka has become a preferred option for real-time log ingestion and offering a distributed, high-throughput pipeline capable of ingesting the volume and velocity of log data that organizations interface with from applications and infrastructure today [36]. Traditional logging systems often lack the scalability and reliability that Kafka can provide, and unlike traditional logging systems, Kafka is built on the principles of topics, producers, and consumers that support a flow of data in its architecture and real-time environment [37, 38]. Although logs are published and consumed independently in Kafka's distributed architecture, Kafka can support event-driven architectures in a reliable way to ensure availability of log data for any downstream analytics or monitoring.

Kafka's distributed design offers substantial benefits over traditional logging systems, especially in terms of durability, fault tolerance, and scalability [39, 40]. Through the replication of log data across partitions and brokers, Kafka provides high availability and node failure resilience, making it well-suited for mission-critical applications. Its integration with orchestration platforms such as Kubernetes additionally boosts deployment agility and operational productivity, enabling organizations to dynamically scale their log pipelines according to the demands of workload [41]. This proves particularly useful within containerized and cloud-native environments, where applications are often being deployed and scaled across distributed clusters [42].

However, despite its strengths, Kafka based log ingestion is not without complications. Managing and scaling Kafka clusters is no easy feat and can be onerous. This is due to state management, persistent storage management, and network configuration, all of which can consume time and resources [43, 44]. In addition, validating that the data is consistent, secure, and available across distributed methodologies adds complexity, and configuration issues will lead to slowdowns and, in some instances, data loss. Many organizations may find specialized expertise needed in both Kafka and container orchestration environments, for example Kubernetes [45]. These challenges can largely be mitigated with an understanding of best practices for deployment, monitoring, and security, and the use of managed service, or automation tools, to simplify Kafka use in production.

### 3.2 Kafka in Containerized Environments

In today's containerized world, Apache Kafka is indispensable for aggregating logs across various application pods and nodes so that there is efficient and scalable data ingestion across distributed micro service architectures. Kafka's high-performance, distributed messaging system complements container orchestration platforms such as Kubernetes, where applications are deployed and scaled dynamically [46, 47]. To enable real-time log collection, sidecar containers or lightweight logging agents like Fluent Bit are typically deployed in conjunction with application containers. These agents run continuously to collect logs from their respective pods and serve as Kafka producers, sending the data into Kafka topics for centralized analysis and processing [48].

Using sidecar containers or log agents effectively loosens the coupling of log management from the core application and allows the logging facility collection be relieved from the core workloads in your application [49]. Log events are often forwarded to Kafka topics that provide the ability to introduce message streaming and partitioned degrees of parallelism to the log stream while achieving a very high throughput and low latency to deal with the large volume and velocity of log data coming from cloud-native environments. By distributing the log messages across a set of Kafka brokers, the durability, availability, and consistency of log data is maintained by leveraging Kafka's distributed architecture, and in the case of a log agent failure or deadlock, logs are replicated across multiple Kafka brokers, so you won't lose your logs and the system is reliable at this point [50].

Even with these benefits, the deployment of Kafka-based log ingestion in containerized environments has operational complications. Orchestrating sidecar container or log agent deployment and configuration across many pods is a complex automation and orchestration exercise [51]. Furthermore, ensuring uniform log structures, schema evolution, and data quality for heterogeneous sources

require robust metadata management and governance processes. However, Kafka integration with containerized logging software has emerged as a best practice cornerstone, allowing organizations to achieve unified, scalable, and robust log management across complex, dynamic application environments.

### 3.3 Kafka Topics and Partitioning for Scalability

Apache Kafka's scalability and effectiveness in dealing with high-scale log data streams are essentially predicated on its fundamental ideas of topics and partitions. A Kafka topic is essentially a logical stream or category of a particular data stream, i.e., an application's logs or a sensor network's events. But the real strength of Kafka is in the fact that each subject is split into several partitions, the major building blocks of parallelism, storage, and fault tolerance in the Kafka system. This partitioning feature allows Kafka to divide data among several brokers in a cluster to enable high throughput, scalability, and reliability in log consumption and processing [52].

From a conceptual standpoint, every partition in Kafka is essentially an ordered, append-only log of messages, which guarantees relative ordering of messages sent to that partition. Such ordering is essential for some applications where processing events in-order is required, such as transaction logs or time series. Likewise, Kafka allows multiple partitions per topic, which allows a producer to write data concurrently, but also allow consumers to read concurrently from multiple partitions. This parallelism significantly scales Kafka to process hundreds of thousands of log data and can scale horizontally by distributing workloads across additional nodes [53, 54].

The flexibility in partitioning strategies also increases Kafka's capacity for optimizing log structure across services or environments. Kafka offers several partitioning schemes, such as round-robin distribution, which distributes messages evenly across partitions; key-based partitioning, which sends all messages that share the same key to the same partition, maintaining their order; and custom partitioners, which enable organizations to apply domain-specific algorithms for distributing data. These strategies balance load, preserve data locality, and make better use of the cache, depending on the application. For instance, in multitenant or micro services deployments, partitioning can segregate log streams by customer or service to make downstream processing and access control easier [55].

Nonetheless, controlling the number and placements of partitions will take planning and continual maintenance. While partitions can provide better throughput and more parallelism for consumers, they also have overhead costs that account for more broker consumption and metadata maintenance, and possibly more network traffic. If too many partitions are present, it is not uncommon to see uneven workloads across partitions or delays if not throttled

properly. Striking that balance will require evaluating the volume of data, the concurrency of consumers, and the throughput of application processing, making the adjustments to the partition counts as the work evolves. Besides throughput and parallelism, Kafka's partitioning mechanism is also fault-tolerant and durable. Partitions are replicated on several brokers such that if a single broker fails, another copy can take over automatically without any loss of data. Replication coupled with partitioning offers a fault-tolerant framework for high-availability log streaming systems that can run reliably in distributed, cloud-native, and containerized environments.

#### 4. Kafka Connect: Bridging Kafka and Data Lakes

Logs and streaming data are critical to understanding system and application behavior. They assist in detecting operational problems, facilitate auditing for compliance, and aid machine learning applications like anomaly detection and predictive analytics [56, 57]. Through ongoing capture of detailed events and metrics, organizations can keep systems healthy, enhance troubleshooting, and create models that predict future trends or failures, making overall decision-making processes better.

Apache Kafka is a powerful, distributed platform for real-time ingestion and aggregation of logs generated by application pods and multiple nodes. Kafka's high throughput, fault-tolerant model decouples the data producer from the data consumer and allows log streaming, even during operational changes typically seen in micro services or container orchestration. As a result, Kafka fills an essential niche for capturing a continuously flowing stream of data and transferring that data into downstream systems to be acted upon or stored [58].

Kafka Connect serves as a connector between Kafka and other data lake technologies, enabling easy streaming of log data into durable storage solutions such as Amazon S3, Google Cloud Storage (GCS), and Hadoop HDFS. Its design is made up of distributed workers, connectors, and tasks that parallelize and automate data movement. Specialized sink connectors like Amazon S3 Sink Connector, GCS Connector, and Hadoop HDFS Connector facilitate integration with heterogeneous storage back ends, allowing scalable and fault-tolerant data ingestion pipelines. Kafka Connect also provides essential features such as exactly-once and at-least-once delivery semantics for ensuring data consistency, as well as schema evolution and serialization formats such as Avro and JSON, which make downstream data processing and analytics easier [59].

After logs are ingested into a data lake, they provide the flexibility and scale to pursue advanced analytics and machine learning. With the schema-on-read approach from data lakes, organizations are able to conduct exploratory data analysis, build predictive models, and perform historical audits without the upfront data transformation

[60]. The rich data of logs enables machine learning models to be trained in predictive maintenance, anomaly detection and customer behavior prediction; ultimately leading to actionable insights and operational efficiencies. When integrated with Kafka into data lakes, organizations can build unified pipelines that improve data quality, reduce machine learning deployment times and support real-time decision-making across different areas of the business.

#### 5. Data Lake as the Foundation for Scalable Log Storage

Data Lakes offer a robust, scalable and cost-effective storage solution for logs and other types of data produced by contemporary businesses. In contrast to conventional data warehouses that need data to be pre-structured before storage, data lakes enable organizations to ingest and store data in its raw form, whether structured, semi-structured, or unstructured. This flexibility is particularly useful for logs, which tend to come in varied formats and large quantities from multiple sources like applications, IoT devices, and cloud services [61].

One of the primary benefits of data lakes is their cost efficiency and elasticity. Leveraging low-cost, cloud based object storages, data lakes can horizontally scale to petabytes or exa-bytes of log data without significant upfront investment and ongoing operational cost. With a pay-as-you-go option as an additional value, the elasticity of a data lake provides organizations with an ability to manage large log volume, even aggravating spikes of log volume, without impacting performance or needing equipment investment and upgrades [62].

Another critical advantage is the schema-on real flexibility that data lakes offer. Logs can be ingested without the need for predefined schemas, allowing organizations to store data as is and define the structure only when the data is accessed for analysis or reporting. This approach accelerates data ingestions, support rapid prototyping, and empowers data scientists and analysts to experiment with data, run advanced analytics, and build ML model directly on raw log data, unlocking deeper insights and innovation.

##### 5.1. Organizing Logs in the Lakes

Effective log organization in a data lake is crucial to achieve maximum usability and performance. Partitioning strategies are commonly employed to optimize data retrieval and processing. Logs can be partitioned based on date, application, region, or other applicable attributes, enabling faster queries and easier data sets. Partitioning by date, for instance, facilitates effective time-based analysis and makes data lifecycle management easier, while partitioning by application or region facilitates targeted troubleshooting and compliance reporting [63].

Along with partitioning, cataloging and metadata indexing are also important to make log data discoverable and actionable. Using metadata catalogs, organizations can have a catalog of all datasets in the data lake, including information about data source, format, schema, and lineage. Metadata indexing enables efficient search, access control, and data governance, allowing users to find and use the log data they require for analytics, auditing, or regulatory compliance quickly.

In short, data lakes provide a strong, elastic, and agile base for log storage that facilitates cost-effective expansion, high-speed ingestion of data, and sophisticated analytics. By adopting good partitioning and metadata management techniques, organizations can make their data lakes strategic resources for operational intelligence and innovation [64].

## 6. Analysis and Machine Learning on Log Data

Today, organizations rely heavily on extracting actionable insights from log data through querying. Query engines including Presto, Amazon Athena, and Google BigQuery enable analysts to run complex ad hoc queries directly against massive amounts of log data stored in data lakes, without needing to pre-stage or transform the data. The engines accept queries from analysts, leverage fast scalable computation, and drive dashboards that visualize trends, error rates, and operational health as they are happening. Alerting systems can be built from query results and provide automatic notification when thresholds are breached or anomalies are detected, allowing teams to respond to an incident or observe system behavior before an incident occurs [65, 66].

### 6.1. Preprocessing Logs for ML pipelines

Logs must be properly pre-processed before they can be utilized in machine learning (ML) pipelines efficiently. Pre-processing includes normalization and standardizing log formats across sources, enrichment by adding contextual information, and feature extraction to determine relevant attributes for machine learning. Pre-processing operations are often handled using tool chains like Apache Spark, Pandas, and DBT, with their scalable and flexible frameworks for log data cleaning, transformation, and structuring. Accurate pre-processing guarantees that the downstream ML models are provided with high-quality, consistent input, which is imperative for effective pattern recognition and anomaly detection [67].

### 6.2. Building Predictive Models

When machine learning is added to log analysis, the predictive capabilities expand beyond what was possible previously with rule-based monitoring. By training models from historical log data, organizations can analyze deviations in the application's behavior for anomalies and predict system load as well as find risks for a security breach before

it occurs. ML-based log analysis uses algorithms to learn automatically about what is "normal" operational performance and highlight what is not normal which may indicate a failure or an attack. The result is that operational reliability and security can be proactively improved in addition to alert fatigue as the analysis will only highlight actionable information with minimal false positives. In the end, ML-based log analysis provides a huge payback to organizations by converting raw log data from their IT systems into a potential strategic asset for predictive maintenance, risk mitigation, and continuous improvement [68].

## 7. Case Study / Example Architecture

A large manufacturing company deployed thousands of IoT sensors across its production facilities to monitor equipment's health, environmental conditions, and operational efficiency. The sensors generated massive volumes of log data, which need to be ingested, processed, and analyzed in real time to enable proactive, maintained and rapid incident response. To address these needs, the company implemented a unified log management architecture using Apache Kafka as the central events streaming platform. Each IoT device published its logs to Kafka for scalability to handle millions of events per second. Kafka connect was used to stream these logs into a cloud based data lake, enabling cost effectiveness, long term storage and supporting downstream analytics.

Once in the Data Lake, the log data was queried using distributed engines and fed into ML pipelines. This enabled real time anomaly detection flagging unusual sensor reading and predicting equipment failure before they occurred. The architecture also supported the creation of dashboards and alerting systems, providing operational teams with actionable insights and automated notification. By integrating Kafka, Kafka connect, and a data lake, the company achieved scalable, reliable log ingestion, unified governance, and advanced analytics capabilities for predictive maintenance and operational optimization [69].

Likewise, an international e-commerce service provider experienced unintegrated log management information distributed across many cloud infrastructures and micro services. The organization required a scalable log aggregation methodology that could collect logs from sources, evaluate the logs for quality, and provide advanced analysis to indicate potential fraud, monitor against customer behavior and assess operational effectiveness on the site. The company used Kafka in the architecture to centralize log collection. Logs from application across services were streamed into Kafka topics, where the high throughput and fault tolerance of Kafka guaranteed that logs were delivered reliably. Kafka Connect was used to push these logs into an Amazon S3-based data lake. The process decoupled log ingestion from analytics workloads, taking advantage of cost effectiveness and elasticity offered by

cloud storage. With the logs centrally located in the data lake, the company was able to build ETL pipelines to scrub and structure the logs in a way that was useful for machine learning and business intelligence use cases. Data scientists utilized this cleansed log data to build predictive models for fraud detection and customer segmentation. Business analysts were able to use query engines to build real-time dashboards and reports. The single architecture not only improved data governance and reduced operational overhead, but it also allowed the company to make faster decisions based on data [70].

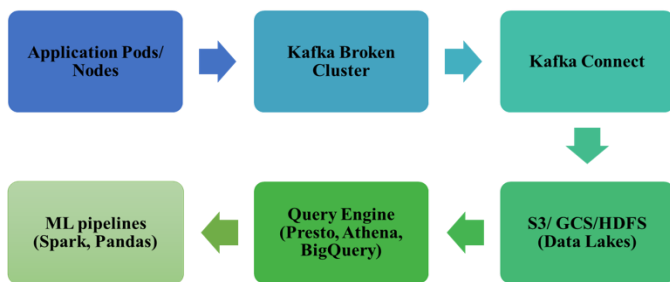


Figure 3. End-to-End Log Architecture

## 8. Challenges and strategies to overcome

Unified log management in modern IT systems, particularly when utilizing Kafka connect to stream to data lakes for sophisticated analysis and machine learning is confronted with several issues. The amount of log data produced by distributed architectures, micro services, and cloud-native apps grow exponentially, creating a massive issue with data overwhelm. Each application, system, and network device generate logs that are frequently important to comprehend system and application behavior, debug issues, audit, and fuel machine learning applications. Yet, the volume of logs can be so large that it overwhelms teams, making it hard to collect, store, and analyze logs effectively. A second significant challenge lies in the complexity and heterogeneity of log sources. Logs are generated in diverse formats and from various endpoints, such as various application pods, nodes, and cloud services. This heterogeneity requires strong standardization and conversion processes that guarantee logs are normalized and search-friendly, a prerequisite for efficient aggregation and analysis in a data lake architecture. Without standardization, it is difficult to relate events among distributed systems, which makes root causes and actionable information harder to identify.

Operational difficulties also stem from the dynamic nature of contemporary IT infrastructure. Containers and short-lived cloud instances are continuously spun up and torn down, rendering it difficult to guarantee thorough and persistent log gathering. Volatility has the potential to cause gaps in log information, making it difficult to troubleshoot and reduce the mean time to resolution (MTTR). In addition, keeping and handling enormous log data sets in data lakes like S3,

GCS, or Hadoop raises issues regarding scalability, expense, and long-term preservation, particularly when there are regulatory requirements for long storage periods. Noise in large log datasets is another crucial challenge. The process of sifting through irrelevant, or duplicate, log entries while trying to derive value is time-consuming and can considerably delay incident response. The most pressing problem may be real-time or near real-time log ingestion and analysis since waiting for useful insights could affect reliability and security of the system.

### 8.1. Future Recommendation

To ensure the log completeness and integrity, constant monitoring for log agents and Kafka lag should be done as well as early alerting about any ingestion errors to ensure healthy data flow. Security and access management are utmost priorities, comprising strict data lake access policies as well as in-transit and at-rest encryption of sensitive data to ensure protection against data breaches. To maximize performance and cost, implementing lifecycle policies for archiving or deleting stale logs efficiently manages storage. Compressing and data tiering also minimize storage costs while preserving fast access to most-used data.

Log management is moving in a direction of real-time log querying using streaming data lake technologies, such as Apache Hudi and Iceberg, which allows for instantaneous insights and faster decisions. An example of the operational changes already being realized are server less processing and auto-scaling ingest pipelines allowing an organization to configure its log ingestion and processing infrastructure, easily scale it as needed without requiring any manual input, thereby reducing complexity and cost. Where many organizations still heavily rely upon human input to react to log events, machine learning and AI powered alerting and diagnostics have changed the way we think about logs and alerting. They are now able to automatically detect anomalies and predict an impending issue, while providing timely situational awareness and actionable intelligence with great impact on the value, security, and reliability of the log data. These advancements allow organizations to respond to operational and security events in real-time, leverage available resources more effectively, and gain greater insight into deeper business intelligence through their log management systems.

## 9. CONCLUSIONS

The proposed review offers an end-to-end log management system that combines Apache Kafka with data lakes to meet the challenges of contemporary distributed systems in managing large-scale logs. Using Kafka's real-time streaming features and the cost-efficient, scalable storage capabilities of data lakes, this system provides effective log aggregation, storage, and analysis. The main contribution of the study is the dual approach that integrates real-time monitoring with long-term analytical capabilities, without the requirement

for independent pipelines for real-time and historic data. The use of Kafka Connect within the framework simplifies the integration process, allowing for flexible and efficient log ingestion into cloud-based storage solutions like Amazon S3 or Google Cloud Storage. By incorporating schema evolution methods and automated de-duplication in the ETL process, the system provides data consistency, quality, and governance while avoiding fragmentation or loss of log records. The study also highlights the value of predictive analytics powered by ML models trained on historical log data, offering organization the ability to predict system failure, detect anomalies, and optimize resource allocation. These capabilities are critical for maintaining high system performance and the uptime. The finding underscores the potential of unifying real time data ingestion with long term data storage, significantly improving operational efficiency, collaborating among DevOps teams, and fostering a data-driven culture. Moving forward, future directions should focus on further enhancing data governance, improving the scalability of the framework for even larger datasets, and developing more sophisticated machine learning models to deepen insights into system performance. Additionally, integrating more advanced anomaly detection methods and exploring other cloud-native technologies will further strengthen the framework's applicability in diverse organizational contexts.

## REFERENCES

- [1] A. Enemosah, "Enhancing DevOps efficiency through AI-driven predictive models for continuous integration and deployment pipelines," *International Journal of Research Publication and Reviews*, vol. 6, no. 1, pp. 871-887, 2025.
- [2] N. Joy, "Scalable Data Pipelines for Real-Time Analytics: Innovations in Streaming Data Architectures," *International Journal of Emerging Research in Engineering and Technology*, vol. 5, no. 1, pp. 8-15, 2024.
- [3] H. P. Salim, "A Comparative Study of Delta Lake as a Preferred ETL and Analytics Database," *International Journal of Computer Trends and Technology*, vol. 73, no. 1, pp. 65-71, 2025.
- [4] B. S. Pillarisetty, "The Role of Data Streaming in Modern E-commerce and Digital Platforms," *IJSAT-International Journal on Science and Technology*, vol. 16, no. 1, 2025.
- [5] N. Gupta and J. Yip, "Data Ingestion in Lakehouse," in *Databricks Data Intelligence Platform: Unlocking the GenAI Revolution*: Springer, 2024, pp. 45-60.
- [6] K. Singh and A. S. Kushwaha, "Data Lake vs Data Warehouse: Strategic Implementation with Snowflake," *International Journal of Computer Science and Engineering (IJCSE)*, vol. 13, no. 2, pp. 805-824, 2024.
- [7] S. D. Kuznetsov, P. E. Velikhov, and Q. Fu, "Real-time analytics: benefits, limitations, and tradeoffs," *Programming and Computer Software*, vol. 49, no. 1, pp. 1-25, 2023.
- [8] T. M. de Menezes and A. C. Salgado, "Using Logs to Mitigate Process Variability and Dependence on Practitioners in Traditional Business Process Automation Software," *Preprints*, 2024.
- [9] D. Protschky, L. Lämmermann, P. Hofmann, and N. Urbach, "What Gets Measured Gets Improved: Monitoring Machine Learning Applications in their Production Environments," *IEEE Access*, 2025.
- [10] G. Calderon, G. del Campo, E. Saavedra, and A. Santamaría, "Monitoring framework for the performance evaluation of an IoT platform with Elasticsearch and Apache Kafka," *Information systems frontiers*, vol. 26, no. 6, pp. 2373-2389, 2024.
- [11] M. B. D. Reddy, J. Jaisri, K. R. Kanagaladharani, K. V. Kumar, and K. Madhavi, "REAL-TIME PREDICTIVE LOG ANALYTICS: A SCALABLE JAVA DRIVEN PIPELINE FOR DYNAMIC INSIGHTS IN MORDEN DATA ENVIRONMENTS," *Journal of Nonlinear Analysis and Optimization*, vol. 15, no. 1, 2024.
- [12] P. B. Desai and O. Goel, "Scalable Data Pipelines for Enterprise Data Analytics," *International Journal of Research in All Subjects in Multi Languages*, vol. 13, no. 1, p. 174, 2025.
- [13] S. C. Rajesh and L. Goel, "Architecting Distributed Systems for Real-Time Data Processing in Multi-Cloud Environments," *Journal of Emerging Technology and Innovative Research (JETIR)*, 2025.
- [14] P. Balasubramanian, S. Nazari, D. K. Kholgh, A. Mahmoodi, J. Seby, and P. Kostakos, "A cognitive platform for collecting cyber threat intelligence and real-time detection using cloud computing," *Decision Analytics Journal*, vol. 14, p. 100545, 2025.
- [15] Z. Zhou, L. Zhou, and Z. Chen, "A Distributed Real-Time Monitoring Scheme for Air Pressure Stream Data Based on Kafka," *Applied Sciences*, vol. 14, no. 12, p. 4967, 2024.
- [16] R. A. Khan, I. Keshta, H. A. Al Hashimi, A. O. Almagrabi, H. S. Alwageed, and M. Alzahrani, "A Fuzzy-AHP Decision-Making Framework for Optimizing Software Maintenance and Deployment in Information Security Systems," *Journal of*

- Software: Evolution and Process*, vol. 37, no. 1, p. e2758, 2025.
- [17] C. Scheepers, "User-based activity logging and analysis to improve system maintenance," North-West University (South Africa). 2024.
- [18] O. Emma and P. Peace, "Building an Automated Data Ingestion System: Leveraging Kafka Connect for Predictive Analytics," *Researchgate.net*, 2023.
- [19] S. Bussa and E. Hegde, "Evolution of Data Engineering in Modern Software Development," *Journal of Sustainable Solutions*, vol. 1, no. 4, pp. 116-130, 2024.
- [20] V. N. Edapurath, "Building Scalable Data Warehouses for Financial Analytics in Large Enterprises," *IJIRCT*, 2024.
- [21] S. Chundru and P. K. Maraju, "INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING," *International Journal of Intelligent Systems and Applications in Engineering*, 2024.
- [22] H. P. Kothandapani, "A Systematic Framework for Data Lake Curation and Regulatory Compliance in Financial Institutions: Architecture, Implementation, and Best Practices," *Emerging Trends in ML and Big data*, 2024.
- [23] T. P. Raptis, C. Cicconetti, M. Falelakis, G. Kalogiannis, T. Kanellos, and T. P. Lobo, "Engineering resource-efficient data management for smart cities with Apache Kafka," *Future Internet*, vol. 15, no. 2, p. 43, 2023.
- [24] S. Jain and J. Das, "Integrating data engineering and MLOps for scalable and resilient machine learning pipelines: frameworks, challenges, and future trends," 2025.
- [25] Z. Fan, B. Yang, J. Peng, B. Pei, C. Zheng, and X. Li, "Dynamic Adaptive Mechanism Design and Implementation in VSS for Large-Scale Unified Log Data Collection," *International Journal of Information Security and Privacy (IJISP)*, vol. 18, no. 1, pp. 1-26, 2024.
- [26] A. Luckow and K. Kennedy, "Data infrastructure for connected transport systems," in *Data Analytics for Intelligent Transportation Systems*: Elsevier, 2025, pp. 121-139.
- [27] P. Vlacheas *et al.*, "Deliverable 5.1 Software platform Architecture," *Horizon*, no. 30, 2026.
- [28] H. Kull and M. Hujic, "Secure log-management for an Apache Kafka-based data-streaming service," ed, 2023.
- [29] E. Zagan and M. Danubianu, "Data lake architecture for storing and transforming web server access log files," *IEEE Access*, vol. 11, pp. 40916-40929, 2023.
- [30] W.-C. Shih, C.-T. Yang, C.-T. Jiang, and E. Kristiani, "Implementation and visualization of a netflow log data lake system for cyberattack detection using distributed deep learning," *The Journal of Supercomputing*, vol. 79, no. 5, pp. 4983-5012, 2023.
- [31] H. P. Kothandapani, "Emerging trends and technological advancements in data lakes for the financial sector: An in-depth analysis of data processing, analytics, and infrastructure innovations," *Quarterly Journal of Emerging Technologies and Innovations*, vol. 8, no. 2, pp. 62-75, 2023.
- [32] N. Boyko, "Evaluating Binary Classification Algorithms on Data Lakes Using Machine Learning," *Revue d'Intelligence Artificielle*, vol. 37, no. 6, 2023.
- [33] N. Su, S. Huang, and C. Su, "Elevating Smart Manufacturing with a Unified Predictive Maintenance Platform: The Synergy between Data Warehousing, Apache Spark, and Machine Learning," *Sensors*, vol. 24, no. 13, p. 4237, 2024.
- [34] M. Gashi, B. Mutlu, and S. Thalmann, "Impact of interdependencies: Multi-component system perspective toward predictive maintenance based on machine learning and XAI," *Applied Sciences*, vol. 13, no. 5, p. 3088, 2023.
- [35] C.-H. Chang, H.-T. Chiao, H.-C. Chang, E. Kristiani, and C.-T. Yang, "A predictive maintenance architecture for TFT-LCD manufacturing using machine learning on the cloud service," *Internet of Things*, vol. 31, p. 101541, 2025.
- [36] S. Kul, S. Kumcu, and A. Sayar, "Docker Container-Based Framework of Apache Kafka Node Ecosystem: Vehicle Tracking System by License Plate Recognition on Surveillance Camera Feeds," *International Journal of Intelligent Transportation Systems Research*, vol. 22, no. 2, pp. 290-297, 2024.
- [37] N. Karpiuk, H. Klym, and T. Tkachuk, "USAGE OF APACHE KAFKA FOR LOW-LATENCY IMAGE PROCESSING," *Electronics and information technologies/Електроніка та інформаційні технології*, no. 26, 2024.
- [38] J. Oza, A. Patil, C. Maniyath, R. More, G. Kambli, and A. Maity, "Harnessing Insights from Streams: Unlocking Real-Time Data Flow with Docker and Cassandra in the Apache Ecosystem," in *2024 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, 2024: IEEE, pp. 1-6.

- [39] M. Usman, "Effortless Lifecycle Management for Experimental IIoT Workloads in Containerized Environments," in *2025 28th Conference on Innovation in Clouds, Internet and Networks (ICIN)*, 2025: IEEE, pp. 202-206.
- [40] M. Pacella, A. Papa, G. Papadia, and E. Fedeli, "A Scalable Framework for Sensor Data Ingestion and Real-Time Processing in Cloud Manufacturing," *Algorithms*, vol. 18, no. 1, p. 22, 2025.
- [41] J. M. I. Arockiasamy, "DevOps-Driven Real-Time Health Analytics: A Scalable Framework for Wearable IoT Data," *International Journal for Multidisciplinary Research*, vol. 7, p. 10.36948, 2025.
- [42] J.-C. Liu, C.-H. Hsu, J.-H. Zhang, E. Kristiani, and C.-T. Yang, "An event-based data processing system using Kafka container cluster on Kubernetes environment," *Neural Computing and Applications*, pp. 1-18, 2023.
- [43] C. Lekkala, "Deploying and Managing Containerized Data Workloads on Amazon EKS," *J Arti Inte & Cloud Comp*, vol. 2, no. 2, pp. 1-5, 2023.
- [44] T. P. Raptis and A. Passarella, "A survey on networked data streaming with apache kafka," *IEEE access*, vol. 11, pp. 85333-85350, 2023.
- [45] O. Alhammadi and O. Abul, "Real-time Web Server Log Processing with Big Data Technologies," in *2024 Innovations in Intelligent Systems and Applications Conference (ASYU)*, 2024: IEEE, pp. 1-8.
- [46] Š. Šprem, N. Tomažin, J. Matečić, and M. Horvat, "Building Advanced Web Applications Using Data Ingestion and Data Processing Tools," *Electronics*, vol. 13, no. 4, p. 709, 2024.
- [47] H. Sreepathy, B. D. Rao, M. K. Jaysubramanian, and B. D. Rao, "Data Ingestions as a Service (DlaaS): A Unified interface for Heterogeneous Data Ingestion, Transformation, and Metadata Management for Data Lake," *IEEE Access*, 2024.
- [48] S. Adhikari, "Real-Time Big Data Processing for Intelligent Transportation Systems: A Framework for Scalability," *Journal of Digital Transformation, Cyber Resilience, and Infrastructure Security*, vol. 10, no. 1, pp. 1-10, 2025.
- [49] N. Perera, "Design of Cloud-Facilitated Data Repositories for Large-Scale Traffic Pattern Analyses," *Northern Reviews on Algorithmic Research, Theoretical Computation, and Complexity*, vol. 10, no. 2, pp. 1-10, 2025.
- [50] Y. Fan and X. Mi, "Design and Efficacy of a Data Lake Architecture for Multimodal Emotion Feature Extraction in Social Media," *IET Software*, vol. 2024, no. 1, p. 6819714, 2024.
- [51] J. Chen, "Leveraging Scalable Cloud Infrastructure for Autonomous Driving Data Lakes and Real-Time Decision Making," 2025.
- [52] J. Kim, D. Lee, and U. Lee, "Affectstream: Kafka-Based Real-Time Affect Monitoring System Using Wearable Sensors," *Available at SSRN 5193660*.
- [53] A. Raza, "Real-time Machine Learning Pipelines for Big Data in Cloud Environments: Implementing Streaming Algorithms on Apache Kafka," *Open Journal of Robotics, Autonomous Decision-Making, and Human-Machine Interaction*, vol. 8, no. 6, pp. 1-11, 2023.
- [54] K. Peddireddy, "Streamlining enterprise data processing, reporting and realtime alerting using apache kafka," in *2023 11th International Symposium on Digital Forensics and Security (ISDFS)*, 2023: IEEE, pp. 1-4.
- [55] T. P. Raptis, C. Cicconetti, and A. Passarella, "Efficient topic partitioning of Apache Kafka for high-reliability real-time data streaming applications," *Future Generation Computer Systems*, vol. 154, pp. 173-188, 2024.
- [56] K. Peddireddy, "Kafka-based architecture in building data lakes for real-time data streams," *International Journal of Computer Applications*, vol. 185, no. 9, pp. 1-3, 2023.
- [57] C. Martín, P. Langendoerfer, P. S. Zarrin, M. Díaz, and B. Rubio, "Kafka-ML: Connecting the data stream with ML/AI frameworks," *Future Generation Computer Systems*, vol. 126, pp. 15-33, 2022.
- [58] B. J. Mary, "Unified Data Quality Frameworks for Real-Time ML Pipelines: Bridging DataOps, MLOps, and Streaming Architectures," *Researchgate.net*, 2025.
- [59] S. Javaherhaghighi and O. Oloruntoba, "Advanced Database Management and Data Mining for Optimizing Supervised E-Commerce Customer Behavior Prediction," *International Journal of Computer Applications Technology and Research*, 2021.
- [60] B. P. R. Rella, "Comparative Analysis of Data Lakes and Data Warehouses for Machine Learning."

- [61] A. Lamer, C. Saint-Dizier, N. Paris, and E. Chazard, "Data Lake, Data Warehouse, Datamart, and Feature Store: Their Contributions to the Complete Data Reuse Pipeline," *JMIR medical informatics*, vol. 12, p. e54590, 2024.
- [62] R. Magesh, U. Ilakkiyaa, R. Shanthini, and R. Charanya, "Unlocking the Potential of Data Lakes: Organizing and Storing Marketing Data for Analysis," in *Data Engineering for Data-driven Marketing*: Emerald Publishing Limited, 2025, pp. 199-216.
- [63] R. Marri, S. Varanasi, and S. V. K. Chaitanya, "Integrating Next-Generation SIEM with Data Lakes and AI: Advancing Threat Detection and Response," *Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023*, vol. 3, no. 1, pp. 446-465, 2024.
- [64] A. Katari, "Integrating Machine Learning with Financial Data Lakes for Predictive Analytics," *journal of Artificial Intelligence* 2024.
- [65] S. Azzabi, Z. Alfughi, and A. Ouda, "Data lakes: A survey of concepts and architectures," *Computers*, vol. 13, no. 7, p. 183, 2024.
- [66] P. L. Foalem, F. Khomh, and H. Li, "Studying logging practice in machine learning-based applications," *Information and Software Technology*, vol. 170, p. 107450, 2024.
- [67] R. Mukhamediev *et al.*, "Classification of logging data using machine learning algorithms," *Applied Sciences*, vol. 14, no. 17, p. 7779, 2024.
- [68] F. Ekundayo, I. Atoyebi, A. Soyele, and E. Ogunwobi, "Predictive Analytics for Cyber Threat Intelligence in Fintech Using Big Data and Machine Learning," *Int J Res Publ Rev*, vol. 5, no. 11, pp. 1-15, 2024.
- [69] R. R. Pasala, M. R. Pulicharla, and V. Premani, "Optimizing Real-Time Data Pipelines for Machine Learning: A Comparative Study of Stream Processing Architectures," *World Journal of Advanced Research and Reviews*, vol. 13, 2024.
- [70] S. R. Nelluri and F. A. A. Saldanha, "Mastering Big Data Formats: ORC, Parquet, Avro, Iceberg, and the Strategy of Selection," *International Journal of Computer Trends and Technology*, 2025.

## BIOGRAPHIES



**Kuriens Shaji Maliekal** has over 12 years of experience in DevOps Engineering, Machine Learning (ML) Ops, Cloud and Data Engineering, Continuous Integration and Delivery (CI/CD) automation and Quality Assurance (QA) Engineering