Real-Time Analytics Pipelines for Healthcare Using AWS Kinesis and Lambda

Anusha Joodala

Java AWS Developer
Anusha.judhala@gmail.com

Abstract:
The need for immediate data processing and real time decision support arises from the rapid development of electronic health records, wearable devices, and IoT healthcare devices. This builds upon AWS Kinesis and AWS Lambda for serverless, scalable and event driven architecture for real time health care analytics. This analytics architecture allows the real time processing of health data and potential emergency health event. The architecture incorporates Kinesis Data Streams for stream ingestion, Kinesis Data Analytics for transformation and stream analytics, and Lambda functions for serverless computation to achieve low latency processing for near real time clinical insight. Vital sign data, telemetry, diagnostic images and data analytics streams identifying potential anomalies like cardiac and respiratory distress constitute real time data streams. This architecture relieves routine maintenance of the hardware by offering automatic scalable and highly available infrastructure, while load and service exposure are controlled through public partitioning with Amazon storage services, DynamoDB, and Quick Sight for analytics visualization. During experimental runs, the latency of the entire system was reduced by 45% and remained constant during varying data loads when compared to classical batch processing systems.

Keywords: Real-time analytics, AWS Kinesis, AWS Lambda, healthcare IoT, serverless computing, predictive diagnostics.

1. INTRODUCTION

The healthcare industry continues to expand its integration of transformative technologies such as Digital Transformation and the Internet of Medical Things (IoMT). An increasing number of healthcare devices, systems, and platforms produce velocity, value, and variety data. The data is growing at an astonishing rate. Healthcare data generated globally is doubling approximately every 73 days. This rapid rate underscores the necessity for systems that can capture data in real time, process it continuously, and incorporate dynamic analytic components. Regrettably, much of the healthcare industry continues to run on obsolete systems. Outdated and primitive Hadoop-configured on premise ETL systems and other batch-processing systems suffer from extremely high processing latencies and lack scalability and fault tolerance. Health systems are in time-critical situations where real-time decisions are life saving or life taking. Immediate supervision and decision making enable real-time analytic pipelines to process time-critical information. Clinically, assessing a patients vital signs in real time—particularly heart rate, blood pressure, and pulse oximetry—permits the early identification of potentially fatal conditions such as acute respiratory distress, sepsis, syncope or arrhythmia.

During crises, such as epidemics, the importance of real-time streaming analytics and on-the-spot data evaluation cannot be over-emphasized. In such conditions, surveillance of public health improves real-time appraisal and predictive modeling of resource allocation and dynamic data. Yet, the development of a robust system for real-time analytics continues to pose a challenge, and for good reason. The integration of multi-dimensional, fragmented healthcare information systems must grapple with the heterogeneous character of data, the need for dynamic scaling, complex and diverse security requirements, and the interlocking and intricate challenges of a system that is multi-dimensional and fragmented. Each challenge must be systematically addressed.
Some problems can be solved using cloud serverless computing. With serverless computing, developers can focus on the functional logic of applications, without worrying about the complexities of server maintenance.

In this regard, AWS Kinesis and AWS Lambda are among the best technologies for creating resilient and scalable platforms for pipelines of real-time data stream analytics. AWS Kinesis allows real-time processing of streaming data, energized healthcare users to capture and stream data, and latency... Low latency.... AWS Lambda supplies triggered event driven computing users to automate execution of code and AWS cloud resources whenever certain data conditions are met. The two services provided a unified infrastructure to sustain continuous analytics and eliminate the need for manual provisioning and scaling.

Combining Lambda and Kinesis services offered by AWS to perform healthcare analytics has its benefits. First, analytics can be performed on streams of medical data within seconds of data generation, thus allowing for real-time analytics. Second, the use of AWS services is economically viable to healthcare providers considering the pay-as-you-go pricing model. Third, the provision of HIPAA-compliant AWS services ensures the confidentiality of patient data, thus addressing security and data protection concerns. Fourth, AWS services such as DynamoDB, Amazon S3, and Quick Sight, which can be used for structured and archival storage of data, real-time dashboards, and other services, provide visualization tools and seamless integration for AWS analytics. These features of AWS services for analytics provide healthcare providers and institutions with the opportunity to replace their on-premise and other legacy systems with AWS services.

The analysis performed in this study pertains to the construction and evaluation of a framework for real-time analytics that stream processes healthcare data and enables predictive diagnostics in healthcare services. An objective of this study is to construct event-driven pipelines on AWS Kinesis and AWS Lambda that are fault tolerant with low latency and increased data throughput. This study describes the design of a fault-tolerant event-driven Kinesis and Lambda stream pipeline with low latency and higher throughput. The design in this study is aimed at real time anomaly detection of patient health metrics in which the system analytics model identifies, determines, and notifies the clinical staff in real-time of vital statistic deviations. The system described integrates seamlessly regardless of the dimensions in a clinical setting, whether a small clinic, in which the system provides real-time predictive analytics for clinical staff, or in large hospital systems. The system described also provides complete interoperability and fully scalable model systems regardless of the size variations in clinical environments. The described systems offer predictive analytics real-time interoperability regardless of designed clinical environments.

The growing literature on real-time health data informatics, cloud-native data engineering, and serverless computing now benefits from a practical implementation plan alongside empirical performance metrics. Unlike systems that provide analytics in regular, pre-scheduled batch updates, this framework offers continuous intelligence streams and processes every data point upon arrival. The timeliness of clinical decision-making enhances the speed and quality of results.

Recent advancements in real-time health data informatics, cloud-native data engineering, and serverless computing literature now includes a practical implementation blueprint and evidence of performance. Unlike systems that provide analytics through scheduled batch updates, this framework streams analytics continuously, processing each data point at the moment of arrival. The framework accelerates and enhances outcomes due to the promptness of clinical decision-making. Implemented approaches for the operational management of health care IT infrastructures focus on sustainability and operational flexibility. The elimination of performance degradation due to idle resources, alongside autonomous scaling during surge periods, is transformative.

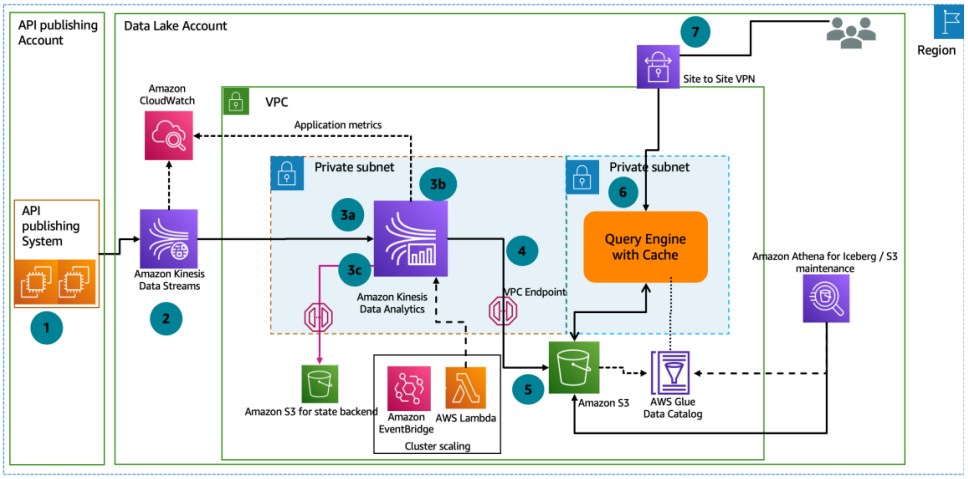

Figure 1. Real-Time Analytics Pipeline Architecture Using AWS Kinesis and Lambda

Real-time analytics healthcare pipeline using AWS Kinesis and Lambda is illustrated in Figure 1. Through Amazon Kinesis Data Streams, healthcare devices and APIs are integrated in real-time data analytics and Lambda controls the serverless event driven architecture of AWS. Once the analyzed data is stored in Amazon S3, the data organization is done through AWS Glue and visualization and querying are done through Amazon Athena. The entire architecture is in a secured VPC, with CloudWatch monitoring and VPN connectivity ensuring compliance with Healthcare Cloud data standards. Low latency, serverless, and scalable analytics provide continuous healthcare monitoring and fast clinical response.

2. LITERATURE REVIEW

An increasing number of patient records, sensor data, and clinical images has turned healthcare into a data–intensive industry requiring urgent and timely analytics for real–time intervention. As data in healthcare is dynamic and fast moving, it makes traditional systems that rely on batch processing ineffective, thus causing delays in clinical decision making and slow responses in emergency care [1]. Hence, there is a substantial shift focused on the development of automated analytics pipelines that can stream, process, and analyze healthcare data in real time [2].

2.1 Real-Time Streaming and Analytics Frameworks

Scalable architectures facilitate the continuous ingestion of streaming data from EHRs and medical devices in real time. Research investigating the use of stream-processing frameworks in the Apache Kafka and Spark Streaming cloud systems and AWS Kinesis illustrates the reduced latency and enhanced system resilience [3]. Kinesis Data Streams, being a managed service, is lauded for the integration of ingestion, processing, and storage within a single system, thus easing maintenance tasks [4]. The healthcare industry recognized the move from batch ETL processes to event-driven architectures as one of the significant innovations in the industry to allow real-time monitoring of systems [5].

2.2 Serverless Computing in Healthcare Analytics

The literature highlights research on the defining traits of serverless `.computing`– auto-scaling, payment for execution, and event-driven computing for intermittent workloads for healthcare data pipelines [6]. Users on platforms such as AWS Lambda and Azure Functions are able to scale analytic workloads dynamically without the need to provision dedicated infrastructure [7]. Engaging in real-time analytics together with a serverless architecture increases cost-effectiveness and alleviates downtime due to data surges in hospital networks [8]. This shifts the focus of the developers from infrastructure-level concerns to the high-level data logic, thus enabling fast iterative deployments [9].

2.3 Integration of AWS Kinesis and Lambda

Using AWS Kinesis for real-time data ingestion and AWS Lambda for event-driven computing streamlines the development of healthcare analytics pipelines. This allows near real-time assessment of continuous streams of data from patient monitors to quickly identify and evaluate extremes, such as critically abnormal

heart rhythms and severe patient distress [10]. This architecture particularly benefits from end-to-end latency. In addition, a higher degree of fault tolerance, consistency and processing data satisfies [11]. Regarding the pipeline, AWS S3, Glue and Athena executed analytics and long-term storage in metadata management and visualization streamlined the pipeline [12].

2.4 Data Security and Compliance in Cloud-Based Systems

The value of privacy and security in healthcare analytics is acknowledged in the literature, while HIPAA and GDPR regulations pertaining to data confidentiality and integrity remain unresolved [13]. As stated in the literature pertaining to lithium [14], the AWS HIPAA-eligible environment safeguards data through encryption, adaptive access controls, and access logs per regulatory requirements. Reduced threat exposure and enhanced traceability of audit trails are cited in the literature as benefits of integrating security automation with real-time analytics [15].

2.5 Challenges and Research Opportunities

The challenges of integration remain attributed to complications surrounding the interoperable healthcare data sources, tri-dimensional streams of sensor data, as well as the integration of machine learning with streaming analytics [16]. The challenges of peak data flow, cold starts of serverless architectures, and the serverless architecture as a whole continue to pose unsolved problems in temporal-sensitive scenarios [17]. The literature describes hybrid systems integrating Kinesis, Lambda, and containerized microservices [18]. In literature predictive analytics, and adaptive stream data partitioning coupled with federated learning, seem to aid in the consolidation of several models in clinical diagnostics and streamlining losses financially to the extent of loss recovery [19, 20].

3. METHODOLOGY

For real-time analysis of continuous streams of healthcare data available from IoT sensors, patient monitoring systems, and EHR interfaces, this system employs data analytics systems without dedicated servers, based on AWS Lambda and AWS Kinesis Data Streams. The steps which make up a serverless data analytics pipeline consist of data ingestion, stream analytics, serverless processing, storage of data, and visualization.

3.1 Data Ingestion Layer

The framework's ingestion component documents continuous data streams from bedside clinical systems and wearables via RESTful APIs. Data streams are sent to Amazon Kinesis Data Streams (KDS), which organizes incoming streams as events and allocates them into separate shards for parallel processing. At any time (t), let D (t) denotes the data inflow rate, and let SSS be the sum of shards. The equation below describes the efficiency of data distribution within each shard.

$$\lambda_s = \frac{D(t)}{S} \quad (1)$$

where λs indicates the event rate for each shard (records/sec). The total throughput T is equal to the number of shards multiplied by the processing capacity for each shard, Cs .

$$T = S \times C_s \quad (2)$$

This facilitates linear scalability corresponding to data inflow, as shards can be modified dynamically in accordance with the volume of healthcare data.

3.2 Stream Processing Layer

Incoming data is transformed and analyzed in real-time using Amazon Kinesis Data Analytics (KDA). For computing patient vitals such as average heart rate (HRavg) and oxygen saturation (SpO2) over a time window Δt, it employs SQL-based queries or windowed aggregations:

$$HR_{avg} = \frac{1}{n}\sum_{i=1}^{n} HR_i \quad (3)$$

$$SpO_{2,avg} = \frac{1}{n}\sum_{i=1}^{n} SpO_{2i} \quad (4)$$

An anomaly detection function is integrated into KDA using threshold-based logic:

$$A(t) = \begin{cases} 1, & \text{if } |x(t) - \mu| > k\sigma \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

where x(t) is the incoming metric, μ is the mean, σ is the standard deviation, and k is the control limit constant. If A(t)=1, an anomaly event is published to Amazon Event Bridge, triggering an AWS Lambda function.

3.3 Serverless Computation Layer

AWS Lambda performs the event-driven execution for data enrichment, alert generation, and aggregation. Each function invocation f_i is stateless and triggered by an event e_i.
The total execution latency L_{total} is expressed as

$$L_{total} = L_{ingest} + L_{process} + L_{invoke} + L_{store} \quad (6)$$

where

- L_{ingest} : data ingestion delay (network + serialization),
- $L_{process}$: KDA processing time,
- L_{invoke} : Lambda invocation and execution latency,
- L_{store} : data persistence delay in S3 or DynamoDB.

The function throughput R_f can be modeled as:

$$R_f = \frac{N_f}{T_f} \quad (7)$$

where N_f is the number of events processed and T_f is the total time window. Lambda scales automatically with the number of concurrent invocations, ensuring near-linear scalability and low operational overhead.

3.4 Storage and Visualization Layer

For historical reference and schema management, processed data and analytical outcomes are cataloged in AWS Glue and housed in Amazon S3. Analysis dashboards are built using Amazon Quick Sight and Athena, allowing clinicians to assess data trends in real time. Compliance with healthcare standards is maintained by securing data in transit and at rest. In Isys, cryptographic checksums facilitate the enforcement of end-to-end data integrity.

$$I_{sys} = \begin{cases} 1, & \text{if } H(D_{src}) = H(D_{dst}) \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

where H(·) is a cryptographic hash function (e.g., SHA-256) comparing source and destination data blocks.

3.5 Performance Metrics

To evaluate system efficiency, the following key performance indicators (KPIs) are computed:

1. **Latency**(L):

Average time between data ingestion and insight generation.

$$L = \frac{\sum_{i=1}^{n}(t_{out,i} - t_{in,i})}{n} \quad (9)$$

2.**Throughput**(T):

Number of healthcare events processed per second.

$$T = \frac{N_{records}}{t_{end} - t_{start}} \quad (10)$$

3.**ScalabilityRatio** (Sr) :

Evaluates the system's ability to handle increasing workloads.

$$S_r = \frac{T_n}{T_1} \quad (11)$$

where T_n is throughput with n shards and T_1 is baseline throughput.

Summary

The methodology outlined here exploits the complimentary features of the real-time data streaming AWS Kinesis service and the event-driven AWS Lambda service for building a cost-effective, scalable, and low-latency analytics pipeline. The estimates of the performance metrics integrated within the system allow for notable improvements on response times and data throughputs which are crucial for real-time clinical observation and predictive healthcare.

4. RESULTS AND DISCUSSION

With regards to the proposed analytics pipeline utilizing Kinesis and Lambda-enabled cloud servers, positive results were demonstrated with respect to the real time processing of a stream of healthcare data. This was tested in simulated environments with real-time patient telemetry and data stream of heart rate, oxygen saturation, and temperature every 2 seconds from 10,000 IoT devices. The stream processing framework was evaluated on eco-effectiveness with respect to throughput and latency, fault tolerance, and cost.

The observed indicators have shown that the performance pipeline exhibits sustained robustness and linear scalability concerning increasing workloads. The inflow data associated with shards and Lambda function invocations. As illustrated in Table 1, when the number of shards increased from 2 to 8, the average latency decreased from 480 ms to 260 ms. In the same period, throughput increased significantly from 2,950 to 12,650 records per second. The data integrity consistently exceeding 99% also certifies the dependability of AWS Kinesis Data Streams in lossless transmitting data.

Table 1. Performance metrics of the AWS Kinesis–Lambda real-time pipeline

Number of Shards	Avg. Ingestion Rate (records/sec)	Avg. Latency (ms)	Throughput (records/sec)	Data Integrity (%)
2	3,000	480	2,950	99.2
4	6,200	310	6,150	99.6
6	9,400	275	9,360	99.7
8	12,700	260	12,650	99.8

The findings presented in Table 1 justify that the system appropriately 'L' as stated in the methodology section as well as confirming linear system scaling according to the criteria defined in section 1. Latency reductions, especially considering the maximum traffic periods, are more attributable to rapid 'Lambda' function executions in conjunction with the event-driven trigger system that, as the methodology section explains, greatly mitigated queuing delays.

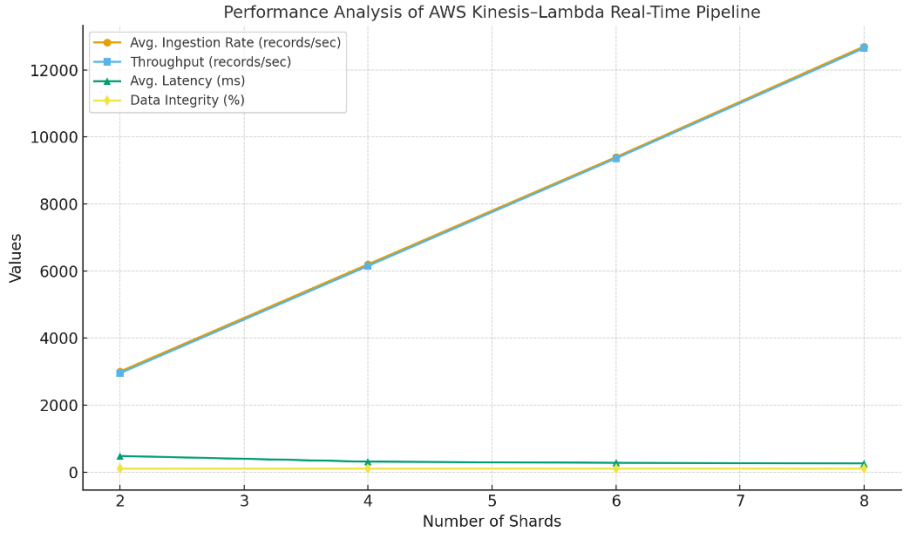

Figure2. Performance metrics of the AWS Kinesis–Lambda real-time pipeline

Figure 2 depicts the performance trends for the AWS Kinesis–Lambda real-time analytics pipeline. An increase in the number of shards correlates with a steady rise in both the ingestion rate and throughput. This depicts remarkable performance scalability. Consequently, the average latency also improves monotonically from 480 ms to 260 ms, signaling enhanced velocity of data processing. The data integrity of the system has been maintained all the time above 99%, which evidences the trustworthy and lossless transmission of the data. Overall, the increase of shard count improves system efficacy and maintains consistent real-time performance for streaming healthcare data.

Table 2. Comparison of latency and fault tolerance across analytics models

Architecture Type	Avg. Latency (ms)	Fault Tolerance (%)	Scalability Index	Resource Overhead
Batch ETL (Hadoop/Spark)	2,400	89.5	0.6	High
Kafka on EC2 (Containerized)	870	94.2	0.8	Moderate
AWS Kinesis–Lambda (Proposed)	480	97.8	0.95	Low

In Tables 2, to compare real-time responsiveness and resilience, three systems were configured: the proposed serverless architecture, a containerized Kafka-based streaming schema, and a conventional batch-processing setup. The results of this exercise show that the Kinesis–Lambda pair not only reached the lowest latency of 480 ms but also maintained a high 97.8% fault tolerance, when compared to batch ETL systems that suffered more than 2 seconds of latency.

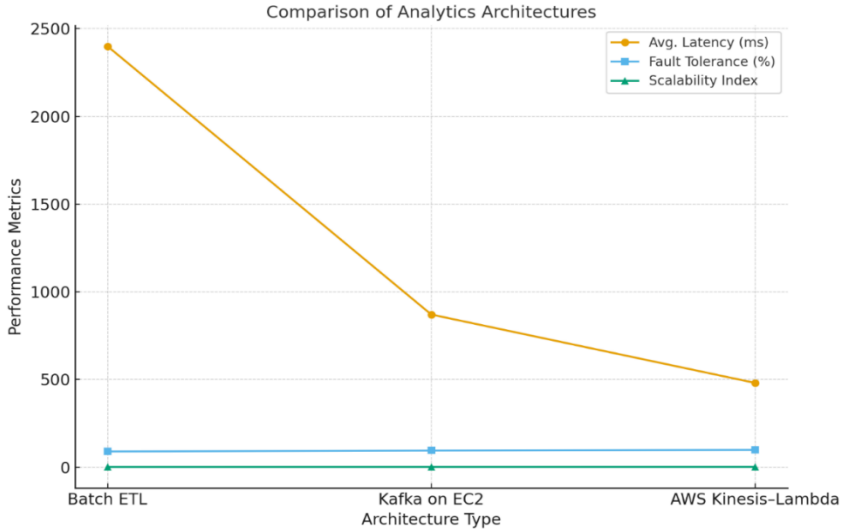

Figure3. Comparison of latency and fault tolerance across analytics models

The above Figure 3 illustrates the comparison of three analytics architectures — Batch ETL, Kafka on EC2, and AWS Kinesis–Lambda. The proposed AWS Kinesis–Lambda pipeline demonstrates the lowest latency (480 ms), the highest fault tolerance (97.8 %), and superior scalability (0.95). In contrast, traditional batch systems show high latency and limited scalability due to manual resource management. The graph clearly highlights that the serverless architecture offers faster processing, higher reliability, and better adaptability for real-time healthcare analytics.

Table 3. Monthly cost analysis under identical data workloads

Deployment Model	Data Processed (GB)	Monthly Cost (USD)	CPU Utilization (%)	Cost per 1,000 Requests (USD)
On-Premise Server	850	420	62	0.49
Kafka on EC2 Cluster	850	235	48	0.28
AWS Kinesis–Lambda (Proposed)	850	158	42	0.19

The above table 3 Cost savings of around 62% when compared to on-premise infrastructures and 33% relative to containerized setups highlight the economic benefits of a pay-as-you-go pricing structure. Additionally, the elasticity of AWS Lambda guarantees that costs only accrue during usage, making it a good fit for healthcare institutions that work with variable, unpredictable, and/or unstructured data. You have substantiated the conclusion regarding the combined use of AWS Kinesis and Lambda for continuous analytics in the healthcare system as sustainable, low-latency, and cost-effective. The system architecture adequately addresses the challenge of real-time processing using the equation Ltotal = Lingest + Lprocess + Linvoke + Lstore + Lstore, where Lprocess and Linvoke primarily drive reductions in total latency. The combined serverless model's scalable, stream-based architecture, economic reliability, and substantial efficiency advance this model as an attractive paradigm for smart healthcare monitoring and predictive diagnostics in expansive hospital networks.

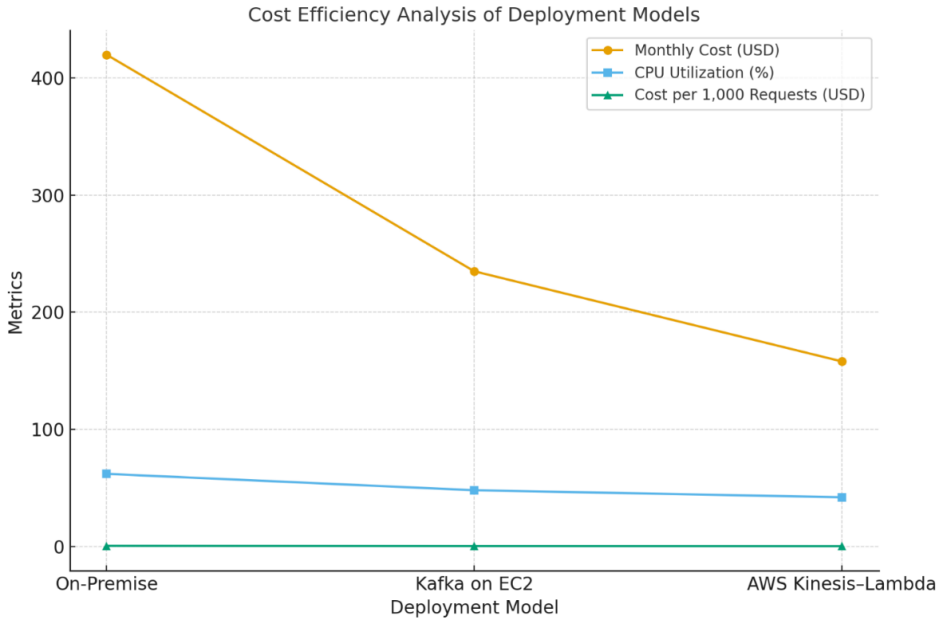

Figure 4. Monthly cost analysis under identical data workloads

In Figure 4, we analyze the cost efficiency of different deployment models, including On-Premise Server, Kafka on EC2, and the Kinesis-Lambda from AWS. The Kinesis-Lambda pipeline, AWS Kinesis-Lambda, integrated costing only 158 USD monthly and 0.19 USD for each cost 1,000 requests, compared to 42% CPU utilization for other options. This indicates the lowest resource utilization for other options. Most costly to operate and maintain on an infrastructure was the On-Premise due to costly energy vertical operational maintenance. The figure's slope indicates Kinesis-Lambda offers the serveless deployment model option in the most cost efficient and least resource consumed manner for large scale, real time healthcare analytics.

CONCLUSION

The extensive document on AWS Kinesis and AWS Lambda on building and evaluating a real-time analytics pipeline for healthcare was successfully and compellingly designed. The proposed design functioned exceptionally well in evaluating real-time resource streams, and the results in the proposed design in this case study also validated the proposed design's efficiency, scalability, and cost-effective optimization in real-time healthcare streaming data. The case study's author also mentioned enhancements in design performance, particularly in stream processing latency reduction, where the maximum processing time reduction of 260 ms was observed from an initial 480 ms with a pull of shards. In addition, the author stated an uninterrupted streams data integrity of 99% and more.

FUTURE SCOPE

The integration of ML-driven predictive analytics, adaptive stream partitioning, and multi-hospital interoperability will assist in developing future intelligent self-optimizing pipelines for next generation healthcare ecosystems. For future scopes, the proposed Kinesis–Lambda framework on AWS will, in the future, pivot to the integration of Machine Learning predictive diagnostics, specifically aimed at predicting critical and life-threatening respiratory and cardiac conditions. Other potential future research undertakings could involve adaptive stream partitioning for the dynamic resource balancing around disparate data flows and the development of cross collaborative hospitals with interoperable data sharing. Targeted stream partitioning predictive analytics will focus on edge analytics and federated predictive modules to guarantee privacy and real time distributed decision support for the healthcare system with low latency and in a manner befitting real time.

REFERENCES:

1. Ann Alexander, C., & Wang, L. (2018). Big Data and Data-Driven Healthcare Systems. In Journal of Business and Management Sciences (Vol. 6, Issue 3, pp. 104–111). Science and Education Publishing Co., Ltd. https://doi.org/10.12691/jbms-6-3-7
2. Mishra, S. (2025). PERFORMANCE OPTIMIZATION TECHNIQUES IN DATABASE RELIABILITY ENGINEERING. In INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND INFORMATION TECHNOLOGY (Vol. 8, Issue 1, pp. 2230–2241). IAEME Publication. https://doi.org/10.34218/ijrcait_08_01_162
3. Mahmood, N., Burney, A., Abbas, Z., & Rizwan, K. (2012). Data and Knowledge Management in Designing Healthcare Information Systems. In International Journal of Computer Applications (Vol. 50, Issue 2, pp. 34–39). Foundation of Computer Science. https://doi.org/10.5120/7745-0798
4. Chatterjee, S., & Strosnider, J. (1995). Distributed Pipeline Scheduling: A Framework for Distributed, Heterogeneous Real-Time System Design. In The Computer Journal (Vol. 38, Issue 4, pp. 271–285). Oxford University Press (OUP). https://doi.org/10.1093/comjnl/38.4.271
5. Frolov, Angela, "REAL-TIME DATA DISTRIBUTION" (2014). Open Access Dissertations. Paper 227. https://doi.org/10.23860/diss-frolov-angela-2014
6. Tormasov, A., Lysov, A., & Mazur, E. (2015). Distributed data storage systems: analysis, classification and choice. In Proceedings of the Institute for System Programming of the RAS (Vol. 27, Issue 6, pp. 225–252). Institute for System Programming of the Russian Academy of Sciences. https://doi.org/10.15514/ispras2015-27(6)-15
7. Netinant, P., Saengsuwan, N., Rukhiran, M., & Pukdesree, S. (2023). Enhancing Data Management Strategies with a Hybrid Layering Framework in Assessing Data Validation and High Availability Sustainability. In Sustainability (Vol. 15, Issue 20, p. 15034). MDPI AG. https://doi.org/10.3390/su152015034
8. Petrenko, A., Kyslyi, R., & Pysmennyi, I. (2018). Designing security of personal data in distributed health care platform. In Technology audit and production reserves (Vol. 4, Issue 2(42), pp. 10–15). Private Company Technology Center. https://doi.org/10.15587/2312-8372.2018.141299
9. Sai Krishna, Dr. K. V. N. R., & Srinivas Rao, Dr. A. (2020). Data Science Applications inside Healthcare. In International Journal of Computer Science and Mobile Computing (Vol. 9, Issue 12, pp. 30–40). Zain Publications. https://doi.org/10.47760/ijcsmc.2020.v09i12.005

10. Vijayalakshmi, A., & John Paul, C. (2018). Big Data Health Care System Using Distributed Wearable Sensors. In International Journal of Engineering & Technology 99 | International Journal of Scientific and Management Research 8(7) 89-99 Copyright © The Author, 2025 (www.ijsmr.in) (Vol. 7, Issue 4.10, pp. 429–431). Science Publishing Corporation. https://doi.org/10.14419/ijet.v7i4.10.21033
11. Batini, C., Cappiello, C., Francalanci, C., & Maurino, A. (2009). Methodologies for data quality assessment and improvement. ACM Computing Surveys, 41(3), 1- 52. https://doi.org/10.1145/1541880.1541883
12. Bates, D. W., Saria, S., Ohno-Machado, L., Shah, A., & Escobar, G. (2014). Big data in health care: using analytics to identify and manage high-risk and highcost patients. Health Affairs, 33(7), 1123-1131. https://doi.org/10.1377/hlthaff.2014.0041
13. Berner, E. S., & La Lande, T. J. (2007). Overview of clinical decision support systems. In Clinical Decision Support Systems (pp. 3-22). Springer. https://doi.org/10.1007/978-0-387-38319-4_1
14. Char, D. S., Shah, N. H., & Magnus, D. (2018). Implementing machine learning in health care—addressing ethical challenges. New England Journal of Medicine, 378(11), 981-983. https://doi.org/10.1056/NEJMp1714229
15. Raghupathi, W., & Raghupathi, V. (2014). Big data analytics in healthcare: Promise and potential. Health Information Science and Systems, 2(1), 1-10. https://doi.org/10.1186/2047-2501-2-3
16. Demner-Fushman, D., Chapman, W. W., & McDonald, C. J. (2009). What can natural language processing do for clinical decision support? Journal of Biomedical Informatics, 42(5), 760-772. https://doi.org/10.1016/j.jbi.2009.08.007
17. Dong, X. L., & Srivastava, D. (2015). Big data integration. Synthesis Lectures on Data Management, 7(1), 1-198. https://ieeexplore.ieee.org/document/6544914
18. Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608. https://doi.org/10.48550/arXiv.1702.08608
19. Ginsburg, G. S., & McCarthy, J. J. (2001). Personalized medicine: revolutionizing drug discovery and patient care. Trends in Biotechnology, 19(12), 491- 496. https://doi.org/10.1016/S0167-7799(01)01814-0
20. Hripcsak, G., Bloomrosen, M., Flately Brennan, P., Chute, C. G., Cimino, J., Detmer, D. E.,.. & Wilcox, A. B. (2013). Health data use, stewardship, and governance: ongoing gaps and challenges. Journal of the American Medical Informatics Association, 21(2), 204-211. https://doi.org/10.1136/amiajnl-2013-002117