IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.15354

Serverless Data Processing Using AWS Lambda and Amazon Kinesis

H Darsan[1], Mr. B. Ramesh Kumar[2]

III BCA, Department of Computer Applications, Sri Ramakrishna College of Arts & Science (Autonomous),

Coimbatore – 641006, Tamil Nadu, India[1]

Associate Professor, Department of Computer Applications, Sri Ramakrishna College of Arts & Science (Autonomous),

Coimbatore – 641006, Tamil Nadu, India[2]

Abstract: Cloud computing has significantly transformed the development and deployment of modern applications. One of the most important advancements in cloud technology is serverless computing, which enables developers to build scalable applications without managing server infrastructure. This paper presents a serverless data processing architecture using Amazon Web Services (AWS), specifically utilizing AWS Lambda and Amazon Kinesis Data Streams to process real-time e-commerce order data.

The proposed system integrates multiple AWS services including Amazon API Gateway, AWS Lambda, Amazon Kinesis Data Streams, and Amazon S3. The frontend application collects order information from users and sends it to an API endpoint. API Gateway triggers a Lambda function that sends the order data to a Kinesis stream. Another Lambda function consumes the streaming records, processes the order information, and stores the processed data in Amazon S3. The system demonstrates how serverless architecture can efficiently process streaming data while eliminating the need for server management. Experimental results show that the system can handle real-time data processing effectively while providing scalability, reliability, and cost efficiency. This study highlights the potential of serverless computing for building scalable cloud-based data processing systems.

Keywords: Serverless Computing, AWS Lambda, Amazon Kinesis, Cloud Computing, Real-Time Data Processing, Event-Driven Architecture.

I. INTRODUCTION

The rapid growth of cloud computing has transformed how modern applications are developed and deployed. Traditional server-based architectures require developers to manage infrastructure, configure servers, and ensure scalability for increasing workloads. These responsibilities increase operational complexity and cost.

Serverless computing has emerged as a promising approach to simplify application deployment and management. In a serverless environment, developers focus only on application logic while the cloud provider manages infrastructure, scaling, and resource allocation. This allows applications to scale automatically based on demand without manual intervention.

Amazon Web Services (AWS) provides several serverless services that enable developers to build highly scalable and event-driven applications. AWS Lambda allows developers to run code in response to events without provisioning servers. Amazon Kinesis Data Streams provides a real-time data streaming platform that enables applications to process large volumes of streaming data.

In modern e-commerce systems, large volumes of order data must be processed in real time. Traditional architectures often struggle to handle dynamic workloads and sudden increases in user requests. By combining serverless computing with streaming technologies, it is possible to build systems that process data efficiently and scale automatically.

This paper proposes a serverless data processing system for handling e-commerce order events using AWS Lambda and Amazon Kinesis. The architecture demonstrates how streaming data can be processed asynchronously using event-driven workflows. The system improves scalability, reduces infrastructure management, and provides cost-efficient real-time data processing.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.15354

II. OBJECTIVES AND CHALLENGES

A. Objectives

The primary objectives of this research are:

1. To design a serverless architecture for real-time data processing.
2. To implement an event-driven system using AWS Lambda and Amazon Kinesis.
3. To process streaming order data efficiently in a cloud environment.
4. To reduce infrastructure management using serverless computing.
5. To demonstrate scalable data processing using cloud services.

B. Challenges

Despite the advantages of serverless computing, several challenges arise during system implementation. One of the main challenges is integrating multiple cloud services effectively. Proper configuration is required to ensure that different services can communicate securely.

Another challenge is handling real-time data streaming. The system must process incoming data without delays or data loss. Efficient management of streaming records is necessary to ensure system reliability.

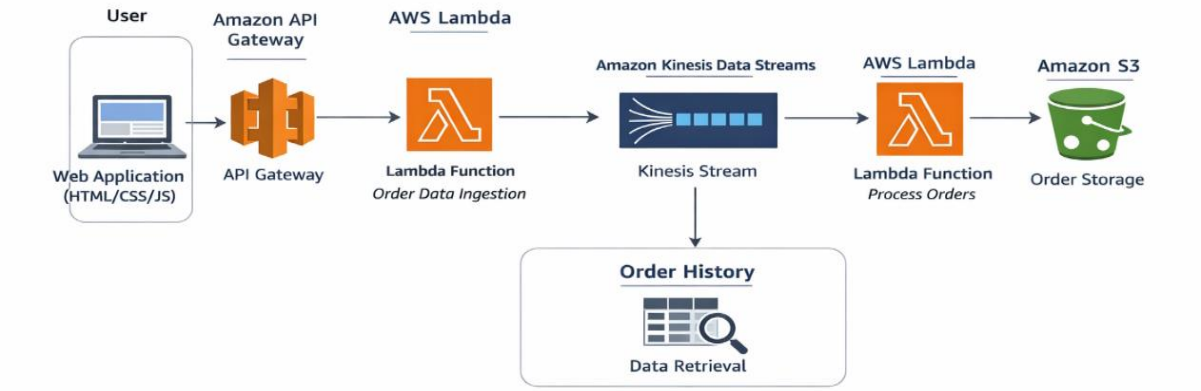
Security and access control also require careful configuration. AWS Identity and Access Management (IAM) policies must be properly defined to allow services such as Lambda, Kinesis, and S3 to interact securely.

Monitoring and debugging serverless applications can also be challenging because the system components are distributed across multiple cloud services. Logging and monitoring tools such as Amazon CloudWatch are essential for diagnosing system issues.

III. SYSTEM ARCHITECTURE

The proposed system follows a serverless event-driven architecture designed to process e-commerce order data in real time. The architecture integrates multiple AWS services to create a scalable data processing pipeline.
The frontend interface is developed using HTML, CSS, and JavaScript and hosted using Amazon S3 static website hosting. The interface allows users to submit order details such as order ID, product name, and price.
When a user submits an order, the request is sent to Amazon API Gateway, which acts as the communication layer between the frontend and backend services. API Gateway forwards the request to an AWS Lambda function.
The first Lambda function acts as a producer that receives order data and sends it to an Amazon Kinesis Data Stream. Kinesis stores incoming records in a real-time streaming pipeline, enabling asynchronous processing of order data.
A second Lambda function is configured as a consumer of the Kinesis stream. Whenever new records are added to the stream, the Lambda function is automatically triggered. This function processes the order information and updates the order status.
The processed order data is then stored in Amazon S3 in JSON format. The frontend application retrieves this stored data and displays it in the order history section of the application.

This architecture provides scalability, fault tolerance, and efficient real-time data processing.



IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.15354

IV. IMPLEMENTATION

The system was implemented using Amazon Web Services to create a serverless data processing pipeline. The implementation process involved configuring several AWS services and integrating them to enable real-time data processing.

First, an Amazon S3 bucket was created to host the frontend application using static website hosting. The website allows users to submit order details through a web interface.

Next, an Amazon Kinesis Data Stream was created to handle real-time streaming of order events. The stream was configured using on-demand capacity mode to automatically scale based on incoming data traffic.

Two AWS Lambda functions were developed using Python. The first Lambda function receives order data from API Gateway and sends the data to the Kinesis stream. The second Lambda function processes streaming records from Kinesis and stores processed data in an S3 bucket.

Amazon API Gateway was configured to provide an HTTP API endpoint that allows the frontend application to send order data to the backend system. Cross-Origin Resource Sharing (CORS) settings were enabled to allow communication between the frontend website and the API endpoint.

AWS Identity and Access Management (IAM) roles were configured to provide secure access permissions for Lambda functions to interact with Kinesis and S3 services.

Finally, Amazon CloudWatch was used for monitoring system logs and verifying successful execution of Lambda functions.

V. RESULTS AND DISCUSSION

The implemented system successfully demonstrates real-time data processing using serverless cloud technologies. The system allows users to submit order details through a web interface and processes the data automatically using AWS Lambda and Amazon Kinesis.

Testing results show that the system can process multiple order requests efficiently without requiring manual server management. The serverless architecture automatically scales based on incoming requests, ensuring consistent system performance.

The use of Amazon Kinesis enables asynchronous data processing, allowing the system to handle streaming data efficiently. This approach improves system reliability and reduces the chances of performance bottlenecks.

Another important advantage observed during testing is cost efficiency. Since AWS Lambda follows a pay-per-use model, the system consumes resources only when processing requests. This significantly reduces infrastructure costs compared to traditional server-based architectures.

Overall, the results demonstrate that serverless architecture provides an effective solution for real-time data processing in modern cloud applications.

VI. CONCLUSION

This paper presented a serverless data processing system using AWS Lambda and Amazon Kinesis. The proposed architecture demonstrates how cloud-based serverless technologies can be used to process real-time e-commerce order data efficiently.

The system integrates multiple AWS services including API Gateway, Lambda, Kinesis, and S3 to create a scalable data processing pipeline. The implementation shows that serverless architecture simplifies infrastructure management while providing automatic scaling and high availability.

The results confirm that serverless computing combined with real-time streaming services is a powerful approach for building scalable and efficient data processing systems.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026
DOI: 10.17148/IJARCCE.2026.15354

VII. FUTURE ENHANCEMENTS

The proposed system can be further improved by integrating additional cloud services. Amazon DynamoDB can be used as a database for faster querying and improved data management.

User authentication can be implemented using Amazon Cognito to enhance system security and allow only authorized users to access the system.

Real-time notifications can also be added using Amazon Simple Notification Service (SNS) to send alerts when orders are processed.

Additionally, system monitoring dashboards can be developed using Amazon CloudWatch to analyze performance metrics and optimize system efficiency.

REFERENCES

[1]. Amazon Web Services, *AWS Lambda Developer Guide*, https://docs.aws.amazon.com/lambda/
[2]. Amazon Web Services, *Amazon Kinesis Data Streams Developer Guide*, https://docs.aws.amazon.com/kinesis/
[3]. Amazon Web Services, *Amazon API Gateway Documentation*, https://docs.aws.amazon.com/apigateway/
[4]. Amazon Web Services, *Amazon S3 Documentation*, https://docs.aws.amazon.com/s3/
[5]. M. Wittig and A. Wittig, *Amazon Web Services in Action*, Manning Publications.
[6]. K. G. Boulton, *Serverless Architectures on AWS*, Packt Publishing.

