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.2026.15361

AWS LAMBDA FOR SERVERLESS MICROSERVICES ARCHITECTURE

Ponnuguru M[1], Mr. T. Pradeep[2]

III BCA, Department of Computer Applications,

Sri Ramakrishna College of Arts and Science (Autonomous), Coimbatore[1]

Assistant Professor, Department of Computer Applications,

Sri Ramakrishna College of Arts and Science (Autonomous), Coimbatore[2]

Abstract: E-commerce platforms require scalable and reliable backend infrastructures to support large volumes of user requests, transactions, and product management operations. Traditional monolithic architectures often face difficulties in handling dynamic workloads and require continuous server management. This research presents a serverless microservices-based ecommerce backend system using Amazon Web Services (AWS).

The proposed platform uses AWS Lambda for serverless computing, Amazon API Gateway for API management, DynamoDB for data storage, and messaging services such as SNS and SQS for asynchronous communication. The architecture divides the application into independent microservices including user management, product catalog, cart management, order processing, and payment services.

Infrastructure deployment is automated using AWS Serverless Application Model (SAM). The serverless approach reduces infrastructure overhead, improves scalability, and enables costefficient cloud resource utilization. The system demonstrates how cloud-native services can be integrated to build a modern, scalable, and reliable e-commerce backend platform.

Keywords: AWS Lambda, Serverless Computing, Microservices, E-commerce, DynamoDB, Cloud Computing

1. INTRODUCTION

Cloud computing has significantly transformed the development and deployment of modern applications. Organizations are increasingly migrating their systems to cloud platforms to achieve higher scalability, flexibility, and operational efficiency. Among the leading cloud service providers, Amazon Web Services (AWS) offers a wide range of services that enable developers to build highly scalable and distributed applications.

Traditional application architectures often rely on monolithic designs where all components of an application are tightly integrated. While this architecture may be suitable for small systems, it becomes difficult to maintain and scale when the application grows. A failure in one module can affect the entire system, making it less reliable.

To address these limitations, microservices architecture has emerged as a modern approach for building large-scale applications. In microservices architecture, applications are divided into smaller independent services that communicate through APIs. Each service performs a specific function and can be developed, deployed, and scaled independently.

Serverless computing further enhances this architecture by eliminating the need to manage physical servers. AWS Lambda allows developers to execute application code without provisioning or maintaining servers. The cloud provider automatically handles infrastructure management, scaling, and resource allocation.

This project proposes a Serverless Microservices E-Commerce Platform using AWS services. The system demonstrates how serverless technologies can be used to build a scalable and reliable backend infrastructure for modern online shopping platforms.

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.2026.15361

2. LITERATURE REVIEW

The rapid growth of cloud computing has encouraged organizations to adopt scalable and flexible architectures for application development. Traditional monolithic architectures, where all application components are integrated into a single system, often face challenges such as limited scalability, complex deployment processes, and high maintenance requirements. As applications grow in size and complexity, managing these monolithic systems becomes increasingly difficult.

Microservices architecture has emerged as a modern solution to overcome these challenges. In microservices architecture, applications are divided into small independent services that perform specific tasks and communicate with each other through APIs. Each service can be developed, deployed, and scaled independently, which improves system flexibility and maintainability. Researchers have highlighted that microservices architectures enhance system reliability and reduce the impact of failures since problems in one service do not directly affect the entire application.

Serverless computing is another technological advancement that simplifies application deployment and infrastructure management. Cloud providers such as Amazon Web Services (AWS) offer serverless platforms like AWS Lambda, which allow developers to run application code without managing servers. Serverless computing automatically scales applications based on demand and charges only for the actual execution time, making it a costeffective solution for cloud applications.

Several studies emphasize the benefits of combining microservices architecture with serverless computing. AWS Lambda enables developers to implement microservices as independent functions that are triggered by events such as HTTP requests, database updates, or messaging services. This event-driven architecture improves application responsiveness and supports efficient resource utilization.

Database services such as Amazon DynamoDB are commonly used in serverless applications due to their scalability and performance. DynamoDB provides a fully managed NoSQL database that automatically scales based on traffic demand. It also offers low latency data access, which is essential for real-time applications such as e-commerce platforms.

Messaging services such as Amazon SNS (Simple Notification Service) and Amazon SQS (Simple Queue Service) play an important role in enabling asynchronous communication between microservices. These services allow different components of the system to communicate without direct dependency, improving system reliability and fault tolerance.

Recent research also focuses on infrastructure automation using tools such as AWS Serverless Application Model (SAM). Infrastructure as Code (IaC) approaches simplify the deployment of cloud applications by allowing developers to define infrastructure configurations using code. This improves consistency, reduces manual errors, and accelerates application deployment.

The integration of serverless computing, microservices architecture, and managed cloud services provides an effective framework for developing modern cloud-native applications. This approach improves scalability, reduces operational complexity, and enables organizations to build highly reliable and efficient systems.

3. SYSTEM ARCHITECTURE

The proposed system uses a **serverless microservices architecture** built on Amazon Web Services (AWS). In this architecture, the application is divided into multiple independent services, each responsible for a specific function such as user management, product management, cart operations, and order processing. This approach improves scalability, flexibility, and system reliability.

The system uses **Amazon API Gateway** to receive requests from client applications and route them to **AWS Lambda** functions. Lambda functions execute the business logic of each service without requiring server management. Application data such as users, products, and orders are stored in **Amazon DynamoDB**, which provides fast and scalable data storage.

For order processing, the system uses **Amazon SNS and Amazon SQS** to enable asynchronous communication between services. This ensures that orders are processed reliably without slowing down the user experience. The entire

DOI: 10.17148/IJARCCE.2026.15361

infrastructure is deployed using **AWS Serverless Application Model (SAM)**, which simplifies cloud resource configuration and deployment.

4. SYSTEM DESIGN

The system design focuses on building a scalable serverless architecture for handling ecommerce operations efficiently. The design follows a layered structure where each layer performs a specific function in managing user requests and processing transactions.

Request Handling Layer

Amazon API Gateway acts as the request handling layer. It receives HTTP requests from client applications and routes them to the appropriate AWS Lambda functions.

Data Storage Layer

Application data such as users, products, carts, and orders are stored in **Amazon DynamoDB**. DynamoDB provides high scalability, low latency access, and secure data storage.

Processing Layer

AWS Lambda functions perform the core processing tasks of the system. Each microservice such as user service, product service, cart service, and order service executes its business logic through Lambda functions.

Messaging Layer

Amazon SNS and Amazon SQS are used for asynchronous communication between services.

These messaging services ensure reliable order processing and improve system performance.

Monitoring Layer

AWS CloudWatch is used for monitoring system performance, logging events, and tracking application activities.

5. IMPLEMENTATION

The system implementation involves configuring various AWS services and integrating them to build a serverless e-commerce platform.

Step 1: Create DynamoDB Tables

Separate DynamoDB tables are created to store data related to users, products, shopping carts, and orders.

Step 2: Develop Lambda Functions

AWS Lambda functions are developed to implement the business logic for each microservice such as user management, product management, and order processing.

Step 3: Configure API Gateway

Amazon API Gateway is configured to expose REST APIs that allow client applications to interact with the backend services.

Step 4: Setup Messaging Services

Amazon SNS and Amazon SQS are configured to enable asynchronous communication between services during order processing.

Step 5: Deploy the Application

DOI: 10.17148/IJARCCE.2026.15361

The complete serverless infrastructure is deployed using AWS Serverless Application Model (SAM), which simplifies configuration and deployment of cloud resources.

6. RESULTS AND ANALYSIS

The implemented system successfully demonstrates the functionality of a serverless ecommerce platform using AWS services. The platform allows users to perform operations such as user registration, product browsing, cart management, and order processing.

Testing shows that the system efficiently handles user requests through **API Gateway and AWS Lambda**, while **Amazon DynamoDB** ensures fast and reliable data storage. The use of **SNS and SQS** enables asynchronous order processing, improving system performance and reliability.

The results indicate that the serverless architecture provides **automatic scalability, reduced infrastructure management, and efficient handling of application workloads**, making it suitable for modern cloud-based e-commerce systems.

7. ADVANTAGES OF THE SYSTEM

The proposed system offers several advantages compared to traditional e-commerce architectures.

- **Scalability:** The serverless architecture automatically scales based on user demand.
- **Cost Efficiency:** Users pay only for the resources used, reducing infrastructure costs.
- **Reduced Server Management:** AWS Lambda eliminates the need to manage physical servers.
- **High Availability:** Cloud services ensure reliable and continuous system operation.
- **Faster Development:** Microservices architecture allows independent development and deployment of services.
- **Improved Performance:** Asynchronous processing using SNS and SQS improves system efficiency.

8.CONCLUSION

This project presents a **serverless microservices-based e-commerce platform** developed using Amazon Web Services (AWS). By utilizing services such as **AWS Lambda, API Gateway, DynamoDB, SNS, and SQS**, the system provides a scalable, reliable, and costefficient solution for handling e-commerce operations.

The serverless architecture eliminates the need for server management and allows automatic scaling based on application demand. The microservices approach improves system flexibility and simplifies maintenance by separating different functionalities into independent services.

Overall, the proposed system demonstrates how cloud-native technologies can be effectively used to build modern, scalable, and efficient e-commerce applications.

9.FUTURE ENHANCEMENTS

The system can be further improved by adding new features and advanced technologies. Future enhancements may include:

- **Integration of real-time analytics dashboards** to monitor user activity and sales performance.
- **Machine learning-based recommendation systems** to suggest products to users.
- **Multi-region deployment** to improve system availability and reduce latency.
- **Integration with secure online payment gateways** for real-world transactions.
- **Enhanced security and monitoring mechanisms** for better protection of user data and transactions.

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.2026.15361

REFERENCES

[1]. Amazon Web Services Documentation – **AWS Lambda**
[2]. Amazon Web Services Documentation – **Amazon API Gateway**
[3]. Amazon Web Services Documentation – **Amazon DynamoDB**
[4]. Amazon Web Services Documentation – **Amazon SNS and Amazon SQS**
[5]. AWS Serverless Application Model (SAM) Documentation
[6]. Research papers and articles on **Serverless Computing and Microservices Architecture**.

