Serverless Bookstore Application Using AWS

Reddyvari Sai Sindhu
Computer Science and Engineering
Institute of Aeronautical Engineering
Hyderabad,India
sindhureddyvari3@gmail.com

Pratapani Rithvik
Computer Science and Engineering
Institute of Aeronautical Engineering
Hyderabad,India
rithvikpratapani03@gmail.com

Satla Ravi Teja
Computer Science and Engineering
Institute of Aeronautical Engineering
Hyderabad,India
raviteja.satla08@gmail.com

T JayaSri Devi
Computer Science and
Engineering(Assistant Professor)
Institute of Aeronautical Engineering
Hyderabad,India
t.jayasridevi@iare.ac.in

***Abstract*— Recently, cloud computing has become a main standard for effective management and Internet services. Within a variety of cloud technology, without server calculation, it is distinguished from the evolution of cloud programming and reflects the range and growth of the cloud. Large Internet companies such as Amazon, Netflix, and LinkedIn are now using the cloud to develop, test, deploy, scale, operate, and update complex, multi-stage applications. Despite the flexibility and scalability, these companies face challenges in managing infrastructure costs due to increasing server load and space requirements. This is where serverless computing, introduced by AWS Lambda, offers a compelling solution. In this article, we explain how AWS Lambda works with other native AWS services through the development of a serverless chat application designed to scale seamlessly without the need for additional servers. The study also explores how AWS services such as S3, DynamoDB, and CloudWatch integrate with serverless technologies such as Lambda to improve the functionality and performance of applications in the cloud.**

Keywords— AWS Lambda, Amazon Amplify, Amazon DynamoDB, Amazon S3, AWS, Cloud Computing, Cloud Storage, Amazon AppSync, IAM..

I. INTRODUCTION

In today's world, server computing is a common practice in organizations, allowing them to work efficiently. However, this approach often comes with significant costs due to the cost of purchasing physical servers or subscribing to cloud services such as Amazon Web Services (AWS), Microsoft Azure, or Google Cloud Platform (GCP). Many organizations have a variety of applications that need to run from time to time, such as calculating employee payroll each month. These applications require servers to perform tasks such as collecting employee information (such as name, bank account number, company ID number, etc.) and calculating pay based on factors such as hours worked and vacation time taken. Serverless computing, despite its name, does not mean that it operates without servers. Instead, the responsibility of server management is transferred from the organization to the cloud service provider. With serverless infrastructure, developers only focus on writing code, typically in the form of functions (such as AWS Lambda functions), and managing the API gateways that interact with the application.

II . LITERATURE REVIEW

Safety and authentication are distinguished as an important problem in the field of incorrect calculation. This study is deepened in various AWS methods, including individual recording sessions and OTP authentication, to determine reliability and enhance safety compared to HTTP -based methods. Furthermore, the study thoroughly investigates the cloud implications associated with serverless architectures, exploring aspects such as infrastructure elasticity, load balancing, provisioning variations, and considerations regarding infrastructure and memory reservation sizes.The main contribution of this study is to propose measures for implementing abstractions within serverless architectures, facilitating efficient cross-server data management, resource allocation, and isolation, among other important features. In the field of serverless computing, extensive research has been conducted on various frameworks aligned with specific real-world application criteria. This research also includes a comprehensive comparison of various cloud platforms, exploring their features and capabilities. Furthermore, this survey will investigate the interactive application of Amazon Cloud Services and criticize the possibility of execution of AWS offer. In particular, research identifies the real paradigm shift to a cloud computing application without a server, driven by a detailed comparison of infrastructure costs that emphasize the economic advantage of the development of Web applications in AWS Lambda.

III. EXISTING RESEARCH

1. Serverless Architecture for E-Commerce Applications

Serverless architecture has been increasingly adopted in e-commerce, including e-bookstores, due to its ability to handle dynamic workloads and unpredictable traffic patterns. Research on serverless architecture in e-commerce applications highlights the following advantages:

Scalability: Serverless platforms like AWS Lambda automatically scale to accommodate varying loads, which is crucial for e-bookstores experiencing fluctuating demand.

Reduced Latency: By deploying functions closer to the end-user and leveraging AWS services like API Gateway and CloudFront, serverless architectures can minimize latency, providing a better user experience.

Cost Efficiency: Serverless models allow e-commerce platforms to reduce costs by only paying for the exact resources used, unlike traditional fixed-server models.

IV. PROPOSED WORK

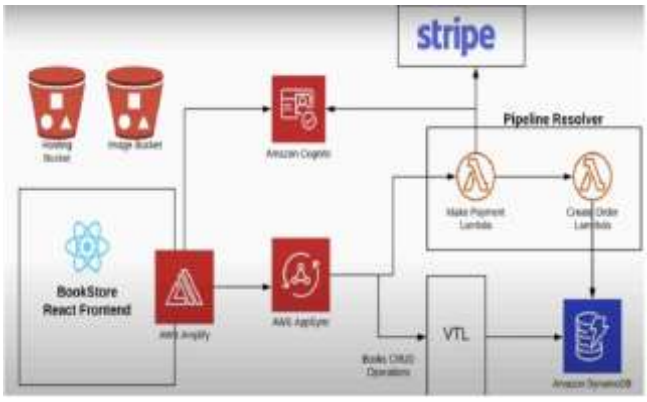
The proposed work aims to design, develop, and implement a serverless e-bookstore using AWS, leveraging serverless architecture to create a scalable, cost-effective, and efficient platform. The project will focus on using AWS services such as Lambda, DynamoDB, S3, and API Gateway to build a fully functional e-commerce application for selling books.This proposed work will demonstrate the effectiveness of serverless architecture in building a robust and scalable e-commerce platform . The use of AWS services optimizes development, reduces general operating costs, and provides practical examples of the latest cloud application design. The final product works as a future application model without a server in the field of electronic commerce.

V. METHODOLOGY

The methodology for developing a serverless eBook store using AWS Amplify, IAM, S3, DynamoDB, AppSync, and GraphQL API includes several key steps: First, in the project planning stage, define the goal and scope of your eBook store and define its features such as user authentication, book storage, purchasing, and search capabilities. The architecture design phase involves creating a high-level diagram showing the interactions between AWS services: AWS Amplify to host the interface, AWS AppSync with its GraphQL API for query processing, S3 to store e-books, DynamoDB to store user and book metadata, and IAM to manage user permissions.

Setting up the environment involves configuring AWS Amplify, initializing the project, and defining IAM roles and policies. The frontend is built using frameworks like React or Angular, with built-in Amplify libraries for authentication, storage, and API interaction. User authentication is set up using Amplify Auth, creating interfaces for browsing, purchasing, and reading eBooks.

5.1 SYSTEM ARCHITECTURE:



.

5.2 OUTCOME SCREENSHOTS

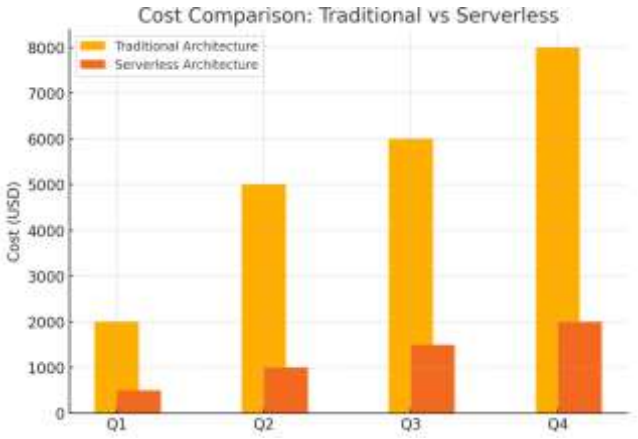
- **Home Page**: The home page introduces the project and provides an overview of its purpose. Users can navigate to different sections such as Home, Books, Cart, and Checkout.

- **Cart Page:** The cart page allows users to check out the details of the book like price of each book, quantity, and total price of the books.

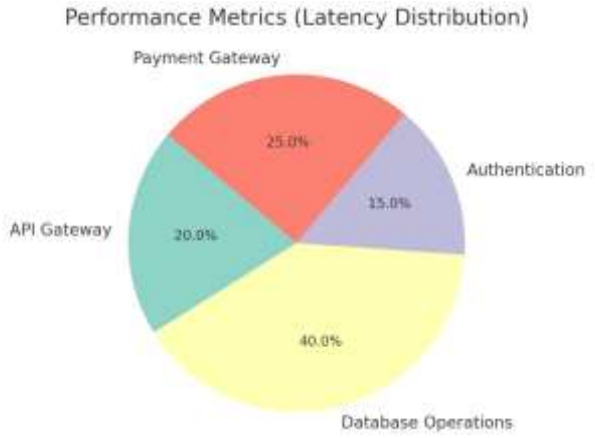
- **Checkout Page:** The checkout page allows users to select payment methods, enter card details and verify address. Once the payment is submitted then the user will be redirected to the tracking page.

VI. DISCUSSION

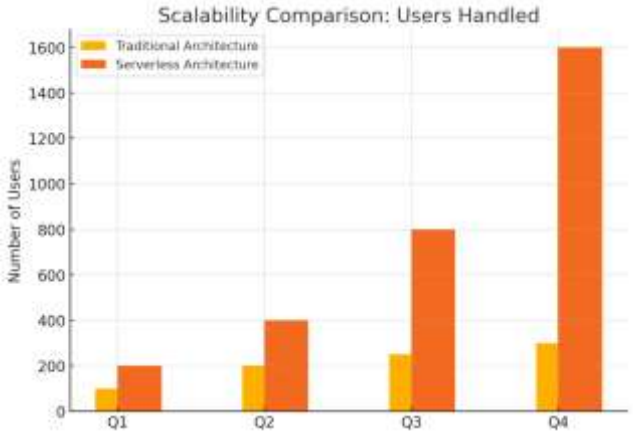
1. Cost Comparison: Traditional vs Serverless : Serverless architecture cuts costs by up to 75% compared to traditional servers, as seen in quarterly comparisons. The pay-as-you-go model makes it highly cost-efficient, especially with unpredictable traffic.



2. Performance Metrics (Latency): Latency is highest in database operations (40%) and API gateway calls (20%). Optimizing these areas can further enhance the system's responsiveness.



3.Scalability : Serverless architecture scales seamlessly, handling up to 1,600 users in peak times, far surpassing the 300-user limit in traditional setups. This automatic scaling ensures consistent performance during traffic surges.



VII. RESULT

Implementing a serverless eBook store using AWS demonstrates the benefits of serverless architecture for modern e-commerce applications. The platform enables seamless scalability by efficiently handling volatile traffic while maintaining low latency, improving user experience. AWS' pay-as-you-go model minimized operational costs and reduced resource wastage, making cost efficiency a key outcome. Robust user authentication and data protection measures ensured strict security and compliance. Overall, the project confirmed that serverless architecture is a powerful and cost-effective solution for building scalable, secure and adaptive online platforms.

VIII. CONCLUSION

The Serverless eBook Store project using AWS Amplify, IAM, S3, DynamoDB, AppSync, GraphQL API, and Cognito User Pools is an example of a modern, scalable, and secure solution for managing and distributing e-books. AWS Amplify allows this project to seamlessly integrate and deploy React interfaces. Amazon Cognito provides strong authentication and user management for improved security and ease of use, AWS AppSync combined with GraphQL enables efficient and flexible data manipulation, and DynamoDB provides fast and reliable storage of user and book metadata.

S3 is utilized for storing book files, offering scalable and cost-effective storage solutions. The integration of payment gateways, such as Stripe, through AWS Lambda functions, facilitates secure and smooth financial transactions. This architecture not only maintains high availability and performance, but also provides data consistency, security, and related standard compliance. In general, without a server, architecture guarantees that the electronic book is easily scalable along with demand, and provides users a reliable high -speed and practical platform for purchasing and reading e -books. This project demonstrates the power of AWS services to build efficient, high-performance serverless applications.

Building a serverless eBook store using AWS services like Amplify, IAM, S3, DynamoDB, AppSync, GraphQL API, and Cognito User Pool opens up opportunities for future scaling. First, AWS's scalability ensures that your eBook store can easily handle changing traffic levels and storage requirements, scaling as needed without downtime. This scalability also translates into cost-effectiveness, as serverless architectures typically only incur costs for actual resource usage, minimizing operational costs. Increased security is another important benefit, as AWS IAM and Cognito User Pool provide strong authentication and access control mechanisms to protect user and administrative access. Additionally, AWS AppSync and GraphQL APIs enable you to receive real-time updates and interactive features, increasing user engagement through dynamic content delivery. Integration with AWS Lambda further enhances functionality by automating workflows and integrating with third-party services such as payment gateways and analytics tools. AWS analytics and monitoring capabilities provide e-store operators with actionable insights into user behavior, sales trends, and operational performance to inform decision-making and continuous improvement.

IX. REFERENCES

[1] S. Mishra, M. Kumar, N. Singh and S. Dwivedi, "A Survey on AWS Cloud Computing Security Challenges & Solutions," 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2022, pp. 614-617, doi: 10.1109/ICICCS53718.2022.9788254.

[2] Philippe Abdoulaye, "Developing World-Class Digital Products and Services Using AWS," in Transforming Your Business with AWS: Getting the Most Out of Using AWS to Modernize and Innovate Your Digital Services , Wiley, 2022, pp.205-206.

[3] R. C. Pushpaleela, S. Sankar, K. Viswanathan and

S. A. Kumar, "Application Modernization Strategies for AWS Cloud," 2022 1st International Conference on Computational Science and Technology (ICCST), CHENNAI, India, 2022, pp.

[4] N. Mahmoudi and H. Khazaei, "Performance Modeling of Metric-Based Server less Computing Platforms," in IEEE Transactions on Cloud Computing, doi: 10.1109/TCC.2022.3169619.

[5] Hai, T., Zhou, J., Jawawi, D. et al. Task scheduling in cloud environment: optimization, security prioritization and processor selection schemes. J Cloud Comp 12, 15 (2023)

[6] Hassan, H.B., Barakat, S.A. & Sarhan, Q. Survey on server less computing. J Cloud Comp 10, 39 (2021)

[7] D. Zhou, H. Chen, G. Cheng, W. He and L. Li, "SecIngress: An API gateway framework to secure cloud applications based on N-variant system," in China Communications, vol. 18, no. 8, pp. 17-34, Aug. 2021, doi: 10.23919/JCC.2021.08.002.

[8] E. Rinta-Jaskari, C. Allen, T. Meghla and D. Taibi, "Testing Approaches And Tools For AWS Lambda Serverless-Based Applications," 2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops), Pisa, Italy, 2022, pp. 686-692, doi: 10.1109/PerComWorkshops53856.2022.9767473.

[9] D. Bardsley, L. Ryan and J. Howard, "Serverless Performance and Optimization Strategies," 2018 IEEE International Conference on Smart Cloud (SmartCloud), New York, NY, USA, 2018, pp. 19-26, doi: 10.1109/SmartCloud.2018.00012.

[10] J. Dantas, H. Khazaei and M. Litoiu, "Application Deployment Strategies for Reducing the Cold Start Delay of AWS Lambda," 2022 IEEE 15th International Conference on Cloud Computing (CLOUD), Barcelona, Spain, 2022, pp. 1-10, doi: 10.1109/CLOUD55607.2022.00016.

[11] S. Roy, S. Kolanu and K. S, "Gaffer: Cloud Computing based Serverless Orchestration Framework for Unprecedented Workflow," 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2021, pp. 1054-1060, doi: 10.1109/ICIRCA51532.2021.9544528.

[12] A. Alalawi, A. Mohsin and A. Jassim, "A survey for AWS cloud development tools and services," 3rd Smart Cities Symposium (SCS 2020), Online Conference, 2020, pp. 17-23, doi: 10.1049/icp.2021.0898.

[13] P. H M, S. Shankaraiah and S. R, "Patient Health Information Framework Using AWS S3 Service," 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon), Mysuru, India, 2022, pp. 1-5, doi: 10.1109/MysuruCon55714.2022.9972361.

[14] S. S. Chawathe, "Data Modeling for a NoSQL Database Service," 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 2019, pp. 0234-0240, doi: 10.1109/UEMCON47517.2019.8992924.

[15] H. Chen, "Low-latency Serverless Computing: Characterization, Optimization and Outlooking: JCC 2021 Invited Keynote," 2021 IEEE International Conference on Joint Cloud Computing (JCC), Oxford, United Kingdom, 2021, pp. xii-xii, doi: 10.1109/JCC53141.2021.00008.

[16] Zhang, Q., Cheng, L. & Boutaba, R. Cloud computing: state-of-the-art and research challenges. J Internet Serv Appl 1, 7–18 (2010). https://doi.org/10.1007/s13174-010-0007-6

[17] Höfer, C.N., Karagiannis, G. Cloud computing services: taxonomy and comparison. J Internet Serv

Appl 2, 81–94 (2011). https://doi.org/10.1007/s13174-011-0027-x

[18] Nagaraju, S., Parthiban, L. Trusted framework for online banking in public cloud using multi-factor authentication and privacy protection gateway. J Cloud Comp 4, 22 (2015). https://doi.org/10.1186/s13677-015-0046

[19] D. Geethika et al., "Anomaly Detection in High-Performance API Gateways," 2019 International Conference on High Performance Computing & Simulation (HPCS), Dublin, Ireland, 2019, pp. 995-1001, doi: 10.1109/HPCS48598.2019.9188100.

[20] S. S. Chawathe, "Data Modeling for a NoSQL Database Service," 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 2019, pp. 0234-0240, doi: 10.1109/UEMCON47517.2019.8992924.