International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 08 | Aug - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

IAC-Based Serverless Image Processor with AWS & Terraform

Nidhin K E[1], Prof. K Sharath[2]

[1] *Student, Department of MCA, Bangalore Institute of Technology, Karnataka, India*
[2]*Professor, Department of MCA, Bangalore Institute of Technology, Karnataka, India*

---***---

Abstract

The rapid growth of digital content has increased the demand for efficient, scalable, and secure systems to manage image processing tasks. Traditional server-based solutions often suffer from high operational costs, limited flexibility, and complex maintenance requirements. This work presents a "Serverless Image Processing Pipeline", designed as a cloud-native, event-driven framework leveraging Amazon Web Services (AWS). The system integrates Amazon S3 for storage, AWS Lambda for computation, and API Gateway for secure interaction with external applications. Images uploaded to a raw storage bucket automatically trigger serverless functions that perform resizing, format conversion, optimization, and other transformations. The processed outputs are stored separately to ensure organized management and secure retrieval. Monitoring and error tracking are facilitated through AWS CloudWatch, while IAM roles enforce role-based access control. The architecture demonstrates elasticity, cost-efficiency, and extensibility, making it adaptable for advanced use cases such as AI-driven classification, watermarking, or real-time analytics. The implementation validates the feasibility of serverless computing in automating image workflows while maintaining performance, reliability, and enterprise-grade security.

Keywords- Serverless Computing, Image Processing, AWS Lambda, Amazon S3, Event-Driven Architecture, Cloud-Native Applications, Cost-Efficient Pipeline, Elastic Scalability, Secure Data Management, API Integration.

.

I. INTRODUCTION

The growing reliance on digital platforms has created an ever-increasing demand for efficient and reliable systems capable of handling large volumes of image data. Industries such as e-commerce, healthcare, digital marketing, and social media require real-time processing of images for tasks like resizing, format conversion, optimization, and secure storage. Conventional server-based architectures often struggle to meet these requirements because they involve high maintenance, limited flexibility, and rising operational expenses. To overcome these limitations, cloud-native and event-driven approaches have emerged as practical alternatives.
This project introduces a Serverless Image Processing Pipeline that leverages Amazon Web Services (AWS) to provide a cost-effective, scalable, and fully automated solution. The pipeline integrates Amazon S3 for storage, AWS Lambda for computation, and API Gateway for seamless interactions with external applications. By adopting a serverless paradigm, the system eliminates the overhead of managing infrastructure and ensures that resources are consumed only when triggered by events.
The workflow begins when an image is uploaded to a raw storage bucket, which automatically invokes Lambda functions to carry out transformations such as resizing, compression, and format conversion. Processed images are then placed into a separate output bucket for organized storage and retrieval. Security is reinforced through IAM-based access control, while monitoring and debugging are handled using AWS CloudWatch.
Beyond its current functionality, the architecture is extensible to support advanced features such as watermarking, AI-powered classification, or integration with mobile and web applications. The result is a flexible, reliable, and enterprise-ready pipeline that demonstrates the practical value of serverless technologies in modern image processing.

II. LITERATURE SURVEY

The rapid rise of cloud-native applications has motivated extensive research into serverless computing and its role in data pipelines. Traditional infrastructures, while powerful, often suffer from high maintenance costs, rigid scaling, and significant management overhead. Recent studies highlight serverless computing as a paradigm capable of addressing these challenges by providing elasticity, reduced operational burden, and fine-grained cost efficiency. Researchers across domains have explored its potential in diverse areas including image analytics, video processing, deep learning, healthcare, and Internet of Things (IoT).
Mathew et al. [1] proposed a pattern-based orchestration model for serverless pipelines, focusing on function composition and workflow reliability. Their work emphasized how orchestration patterns can enhance performance while mitigating the complexity of managing multiple cloud functions. Similarly, San Juan [2] presented an early framework for image processing using serverless functions, demonstrating the feasibility of resizing and transforming images without dedicated infrastructure. This study set a foundation for applying Function-as-a-Service (FaaS) in multimedia applications.
Yang et al. [3] extended the serverless paradigm into remote sensing, introducing FaasRS, a system that processed satellite imagery through serverless functions. Their findings underscored scalability and reduced operational costs, highlighting potential applications in geospatial monitoring. Complementing this, Rad and Ghobaei-Arani [4] provided a taxonomy and comprehensive review of data pipelines in serverless computing, classifying different design approaches and identifying research trends. Their analysis helped frame

the broader context for serverless architectures in big data workflows.

In the domain of machine learning and AI-driven pipelines, Arrak et al. Study [5] explored the role of serverless platforms in machine learning, mapping out existing applications and identifying emerging directions in the field.. They noted the potential for on-demand training and inference while recognizing challenges such as cold starts and resource limitations. Dubey et al. [6] further contributed by demonstrating a deep learning-based serverless image handler implemented on AWS, showing how pre-trained models could be embedded into serverless workflows for efficient image classification and transformation. Similarly, Liu et al. [11] proposed FuncPipe, a serverless solution focused on efficient and low-cost training of deep learning models.. Their results revealed how pipeline parallelism could accelerate workloads while maintaining financial sustainability.

Serverless pipelines have also been explored in IoT and fog computing environments. Poojara et al. [7] investigated serverless data pipelines for IoT data streams, showing how hybrid cloud-fog models can handle latency-sensitive tasks. Dehury et al. [8] proposed a general-purpose pipeline architecture for serverless platforms, outlining methods for optimizing execution flow and reducing redundant data movement. Mirampalli et al. [13] compared NiFi and MQTT-based pipelines in fog environments, highlighting trade-offs between throughput and resource efficiency.

A parallel stream of research has concentrated on video processing pipelines. Zhang et al. [9] introduced CharmSeeker, an automated configuration system for serverless video workflows that improved efficiency in high-demand multimedia environments. Ao et al. [10] presented Sprocket, a framework for serverless video processing, focusing on resource allocation and scalability under varying workloads. The findings proved that serverless approaches can scale from image processing to real-time multimedia workflows.

Healthcare applications of serverless data pipelines have also gained momentum. Singh et al. [12] explored a serverless framework for disease prediction using chest X-ray images, achieving efficient processing and analysis through Lambda-based pipelines. Their study highlighted the potential of serverless computing in diagnostic systems where scalability and timely execution are critical.

From an operational standpoint, the adoption of serverless pipelines requires strong integration with modern software engineering practices. Ivanov and Smolander [14] examined the implementation of DevOps pipelines for serverless applications, focusing on continuous integration and delivery (CI/CD). Their findings emphasized how automation and iterative deployment cycles can improve reliability and accelerate innovation in serverless systems.

Collectively, these works establish a rich body of knowledge demonstrating the adaptability of serverless data pipelines across domains. From foundational image processing [2,6], geospatial applications [3], and IoT workflows [7,13] to AI-driven deep learning [5,11] and video streaming [9,10], serverless computing has emerged as a unifying paradigm. Key themes across the literature include scalability, cost-efficiency, flexibility, and ease of integration with existing services. At the same time, recurring challenges such as cold starts, latency in real-time tasks, and limitations in resource-bound functions remain active areas of research.

In summary, the literature highlights the promise of serverless pipelines as a robust framework for data-intensive tasks while also stressing the need for ongoing innovations in orchestration, optimization, and cross-domain integration. This project builds upon these foundations by developing a Serverless Image Processing Pipeline that combines event-driven automation with scalability, extensibility, and security to address practical needs in image transformation and storage.

III. EXISTING SYSTEM

In traditional image processing systems, applications typically rely on dedicated servers or virtual machines to handle tasks such as uploading, transforming, and storing images. These systems operate on a fixed infrastructure where servers must be provisioned in advance, configured for peak demand, and continuously maintained regardless of workload variations. The workflow generally begins with an image uploaded by the user through a web interface or directly into a centralized server. The application then processes the image by performing operations such as resizing, compression, or format conversion. Once processed, the image is stored in a database or file system for retrieval by end users or external applications.

While this approach has been widely used in content management platforms, e-commerce systems, and healthcare applications, it has several drawbacks. Most notably, the infrastructure is tightly coupled, requiring continuous monitoring and scaling adjustments by system administrators. Additionally, servers must remain online at all times, leading to unnecessary energy and resource consumption even during low activity. As the demand for real-time image processing grows, traditional server-based systems struggle to provide elasticity and cost-efficiency, resulting in higher expenses and longer deployment cycles. These limitations make conventional systems less suitable for modern, high-volume, and dynamic workloads.

Disadvantages

- High Operational Cost – Servers must remain active 24/7, leading to wasted resources even during idle periods.
- Limited Scalability – Traditional systems struggle to handle sudden spikes in image processing workloads.
- Maintenance Overhead – Requires constant monitoring, updates, and manual scaling by administrators.
- Security Concerns – Centralized storage and processing increase the risk of breaches.
- Inefficient Resource Utilization – Underuse of infrastructure during low demand increases inefficiency.

IV. PROPOSED SYSTEM

The proposed system introduces a Serverless Image Processing Pipeline that eliminates the need for conventional server management while providing a fully automated and event-driven solution. Images uploaded to an Amazon S3 bucket automatically trigger AWS Lambda functions, which handle tasks such as resizing, compression, format conversion, and optional watermarking. The processed images are then stored in a separate output S3 bucket, ensuring proper segregation of raw and processed files for better organization and security.

The system is built to harness the strengths of cloud-native services, allowing it to scale easily while remaining adaptable. Since resources are provisioned only when needed, it significantly reduces operational costs compared to traditional infrastructures. The event-driven model ensures that the pipeline scales seamlessly with incoming workloads, allowing it to handle concurrent image processing without degradation in performance.

Security is enforced through IAM roles and fine-grained policies, ensuring that access to storage and processing functions is strictly controlled. In addition, CloudWatch monitoring and logging are integrated to provide visibility, error detection, and system health insights. APIs exposed via Amazon API Gateway further extend the solution's usability, enabling external systems and applications—such as e-commerce platforms or content management systems—to interact securely with the pipeline.

This approach creates a robust, cost-effective, and enterprise-ready framework that can evolve to incorporate future enhancements such as AI-driven image tagging, content moderation, and real-time analytics.

Advantages:

- Automation – Eliminates manual intervention by processing images instantly upon upload.
- Scalability – Handles varying workloads dynamically without performance loss.
- Cost-Efficiency – Pay-per-use billing minimizes infrastructure expenses.
- Security - is maintained by encrypting all data both during transfer and while stored, reinforced with strict IAM role and policy controls
- Separation of Concerns – Raw and processed images are stored in separate buckets for clear organization.
- Monitoring & Reliability – CloudWatch enables proactive error tracking and system health checks.
- Extensibility – Easily adaptable to include advanced AI, analytics, or multi-cloud deployments.

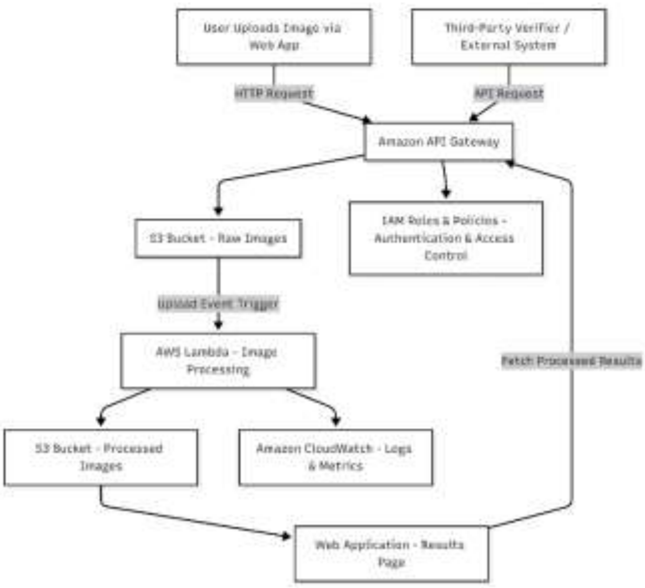

Fig 1: Proposed Model

V. IMPLEMENTATIONS

System Architecture:

The system architecture is built on AWS serverless services, primarily Amazon S3, AWS Lambda, and API Gateway. Raw images are uploaded into the input bucket, automatically triggering Lambda functions for processing. The processed files are then stored in the output bucket, ensuring separation, scalability, and seamless integration with other applications.

Authentication and User Management:

User access is governed by IAM roles and policies. Each role, such as Admin, Client, or End User, has controlled privileges, ensuring strict access boundaries and secure operations across all pipeline components.

Input Handling:

The system accepts image uploads via the web interface or directly into S3. Validation ensures supported formats (JPEG, PNG, WebP) are accepted before processing begins, maintaining efficiency and reliability.

Prompt Construction and LLM Interaction:

Although primarily designed for image processing, the architecture can support API-based interactions with LLMs for metadata tagging or automated annotations. Input prompts are structured, enabling Lambda functions to generate meaningful insights alongside traditional image transformations.

Dialogue Extraction and Post-processing:

In advanced use cases, extracted metadata or contextual annotations from processed images can be refined through post-processing steps. This involves cleaning irrelevant details, formatting results, and structuring outputs for external applications like content management systems or e-commerce platforms.

Error Handling and Security:

CloudWatch monitors all executions, enabling quick identification of failures. Automatic retries handle transient issues, while encryption and IAM policies ensure data integrity, confidentiality, and compliance with regulatory security standards.

VI. CONCLUSIONS

The Serverless Image Processing Pipeline illustrates how modern cloud-native technologies can streamline image management without the burden of traditional server infrastructure. By combining AWS services such as S3 for storage, Lambda for computation, API Gateway for communication, and CloudWatch for monitoring, the system creates an efficient flow for uploading, processing, and retrieving images. The design not only scales automatically-allowing Lambda to handle many requests in parallel-but also reduces costs through a usage-based billing approach.

A significant achievement of the work is the complete removal of manual steps, resulting in an automated and dependable pipeline. The separation of raw and processed images ensures data clarity, while IAM-based permissions and encryption provide strong safeguards for secure access. Early challenges, including role misconfigurations and function timeout limits, shaped valuable lessons that ultimately improved the system's reliability and performance. Beyond the current implementation, the architecture leaves room for growth. Features such as AI-powered tagging, automated moderation, or video handling can be added without disrupting the core workflow, making it suitable for industries ranging from healthcare to digital marketing and online retail.

In summary, the project highlights how event-driven and serverless architectures can transform conventional workflows into adaptive, resilient, and cost-conscious solutions that meet today's needs while remaining open to tomorrow's innovations.

VII. FUTURE ENHANCEMENTS

While the current implementation of the Serverless Image Processing Pipeline successfully addresses automation, scalability, and cost efficiency, there are several opportunities to strengthen its capabilities and broaden its applicability in real-world environments.

One promising enhancement is the integration of artificial intelligence and machine learning models into the pipeline. Beyond basic transformations such as resizing or format conversion, advanced features like object detection, face recognition, and automated image classification can be embedded to deliver intelligent analytics. This would open up new possibilities for applications in healthcare, security, and digital media. Similarly, incorporating content moderation capabilities could ensure compliance and safe usage in public-facing systems.

Expanding the pipeline to handle video processing is another natural step forward. Features like video transcoding, frame extraction, and automated thumbnail generation could be implemented using services such as AWS Elastic Transcoder or Rekognition Video, thereby enhancing versatility for industries like e-learning, media streaming, and surveillance.

From an operational standpoint, adding a custom dashboard would provide real-time monitoring of processed images, error rates, and performance metrics, offering transparency to both administrators and users. Strengthening security through end-to-end encryption, compliance with data regulations, and multi-factor authentication could further reinforce trust in the system.

Finally, enabling multi-cloud deployment and supporting integration with third-party applications would increase flexibility and adoption. These enhancements would collectively ensure that the pipeline remains future-ready, intelligent, and adaptable to evolving digital needs.

VIII. REFERENCES

[1] A. Mathew, et al., "Pattern-based Serverless Data Processing Pipelines for Function-as-a-Service Orchestration Systems," *Future Generation Computer Systems*, vol. 154, pp. 87–100, Jan. 2024.

[2] J. D. Q. San Juan, *Image Processing Using Serverless Functions*. Department of Computer Science and Engineering, University of Waterloo, Apr. 2020.

[3] G. Yang, et al., "FaasRS: Remote Sensing Image Processing System on Serverless Platform," in *Proc. IEEE 46th Annual Computers, Software, and Applications Conf. (COMPSAC)*, pp. 258–267, July 2021.

[4] Z. S. Rad and M. Ghobaei-Arani, "Data Pipeline Approaches in Serverless Computing: A Taxonomy, Review, and Research Trends," *Journal of Big Data*, vol. 11, no. 1, June 2024.

[5] A. Arrak, et al., "Serverless on Machine Learning: A Systematic Mapping Study," *IEEE Access*, vol. 10, pp. 99337–99352, Jan. 2022.

[6] P. Dubey, et al., "Deep Learning-Based Serverless Image Handler Using Amazon Web Services," *Routledge eBooks*, pp. 25–41, 2023.

[7] S. R. Poojara, et al., "Serverless Data Pipeline Approaches for IoT Data in Fog and Cloud Computing," *Future Generation Computer Systems*, vol. 130, pp. 91–105, Dec. 2021.

[8] C. Dehury, et al., "Data Pipeline Architecture for Serverless Platform," *Communications in Computer and Information Science*, pp. 241–246, 2020.

[9] M. Zhang, et al., "CharmSeeker: Automated Pipeline Configuration for Serverless Video Processing," *IEEE/ACM Transactions on Networking*, vol. 30, no. 6, pp. 2730–2743, June 2022.

[10] L. Ao, et al., "Sprocket: A Serverless Video Processing Framework," in *Proc. ACM Symposium*, Sept. 2018.

[11] Y. Liu, et al., "FuncPipe: A Pipelined Serverless Framework for Fast and Cost-Efficient Training of Deep Learning Models," *Proc. ACM on Measurement and Analysis of Computing Systems*, vol. 6, no. 3, pp. 1–30, Dec. 2022.

[12] V. Singh, et al., "Disease Prediction using Chest X-ray Images in Serverless Data Pipeline Framework," in *Proc. IEEE CCGRID Workshops*, pp. 184–191, May 2023.

[13] S. Mirampalli, et al., "Evaluating NiFi and MQTT Based Serverless Data Pipelines in Fog Computing Environments," *Future Generation Computer Systems*, vol. 150, pp. 341–353, Sept. 2023.

[14] V. Ivanov and K. Smolander, "Implementation of a DevOps Pipeline for Serverless Applications," *Lecture Notes in Computer Science*, pp. 48–64, 2018.