Career Point International Journal of Research (CPIJR)
©2022 CPIJR | Volume 3 | Issue 4 | ISSN: 2583-1895
July-September 2025 | DOI: https://doi.org/10.5281/zenodo.17383079

A Review paper on Serverless Web Applications: Benefits and Challenges in Cloud Computing

Kashish Rangwani[1], Ms. Shalini Chawla[2]

[1]Student (BCA), School of Computer Application & Technology, Career Point University, Kota (Raj.), India
[2]Assistant Professor, School of Computer Application & Technology, Career Point University, Kota (Raj.), India

Abstract

The emergence of serverless computing has significantly transformed the development and deployment process of modern web applications. Serverless architecture, a cloud computing execution model where the cloud provider dynamically manages the allocation of machine resources, eliminates the need for traditional server management by developers. This model, often implemented through Function-as-a-Service (FaaS) platforms such as AWS Lambda, Azure Functions, and Google Cloud Functions, provides automatic scaling, high availability, and a pay-as-you-go pricing model, making it particularly attractive for startups and agile development teams.

This review paper explores how serverless web applications operate within cloud ecosystems and evaluates the benefits that have made them increasingly popular among developers and enterprises. These advantages include reduced operational complexity, rapid deployment cycles, cost efficiency, and improved scalability. Alongside these strengths, the paper also examines the key challenges that accompany serverless computing, such as cold start latency, vendor lock-in, limited debugging capabilities, execution time limits, and concerns regarding security and privacy.

By studying current literature, real-world use cases, and technological trends, this paper aims to provide a balanced perspective on the effectiveness of serverless architectures in solving modern web application development problems. The paper also identifies gaps in current research and discusses future opportunities for enhancing serverless technologies. As cloud computing continues to evolve, understanding serverless systems is critical for building efficient, resilient, and scalable applications in the digital era.

Career Point International Journal of Research (CPIJR)
©2022 CPIJR | Volume 3 | Issue 4 | ISSN: 2583-1895
July-September 2025 | DOI: https://doi.org/10.5281/zenodo.17383079

Keywords: Serverless Computing, Cloud Computing, Function-as-a-Service (FaaS), Web Applications, AWS Lambda, Azure Functions, Cold Start, Scalability, Event-driven Architecture, Vendor Lock-in, Cost Efficiency, Debugging Challenges, Stateless Functions, Execution Timeout, Security Concerns

Introduction

Serverless computing is a modern cloud-based execution model where application code is deployed as discrete functions without the need for server provisioning or management. This model is enabled by cloud providers such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud, through platforms like AWS Lambda, Azure Functions, and Google Cloud Functions. In this architecture, developers write and deploy code that is automatically triggered by events such as HTTP requests, file uploads, or database changes. The cloud provider handles infrastructure concerns like resource allocation, load balancing, and scaling. This approach has transformed how modern web applications are built, deployed, and maintained, especially for startups, microservices, and real-time services.

While serverless architecture offers significant benefits in terms of cost savings, development speed, and operational simplicity, it also introduces new challenges. These include cold start latency, limited control over backend infrastructure, complex debugging, and security concerns due to external service dependencies. Moreover, many applications become tightly coupled with specific cloud providers, leading to vendor lock-in. Although serverless adoption is growing, academic and industry research often focuses more on performance advantages and less on the limitations or unresolved issues. This report aims to provide a balanced overview by discussing both the benefits and the challenges that accompany serverless web applications.

This review focuses on the implementation, advantages, and limitations of serverless web applications within the broader context of cloud computing.

It aims to:

- Identify the core benefits of adopting serverless architecture for web development
- Explore technical and operational challenges faced in real-world deployments
- Analyse current research and technological trends shaping serverless computing

The PICOS framework applied in this review is as follows:

©2022 CPIJR | Volume 3 | Issue 4 | ISSN: 2583-1895
July-September 2025 | DOI: https://doi.org/10.5281/zenodo.17383079

Participants: Cloud-based developers, architects, and organizations using or considering serverless technologies

Interventions: Use of serverless platforms (FaaS) for building scalable web applications

Comparisons: Serverless vs. traditional cloud and on-premise architectures

Outcomes: Application performance, scalability, cost efficiency, security implications

Study Design: Review and synthesis of recent literature, technical documentation, and real-world case studies

Review of Literature

Security and Tooling Concerns: Several researchers, such as Shillaker and Pietzuch (2021), raised concerns about the security of third-party event triggers, insufficient isolation between functions, and increased attack surfaces in serverless environments. Others like Eyk et al. (2023) evaluated observability challenges and proposed improvements in tracing function flows and managing state externally via services like AWS DynamoDB or Redis.

Modern Enhancements and Use Cases (2022–2024): Recent literature focuses on extending serverless for broader use cases, including large-scale web applications and hybrid deployments combining containers with FaaS. McGrath and Brenner (2022) analyzed debugging, logging, and monitoring in serverless environments and highlighted the lack of tool integration. Case studies from Netflix, Airbnb, and Coca-Cola demonstrate real-world implementation of serverless for high-demand applications, showing benefits in reducing DevOps complexity and operational costs.

Enterprise Adoption and Performance Insights (2019–2021): As serverless matured, organizations began adopting it for web APIs, real-time analytics, and IoT backend services. Jonas et al. (2019) from UC Berkeley published a seminal paper, “Cloud .Their work introduced the idea of “serverless 2.0” to address emerging bottlenecks.

Platform Comparison and Evaluation (2017–2018): Baldini et al. (2017) performed one of the first comprehensive comparisons of serverless platforms, analysing performance, scalability, and pricing across AWS Lambda, Azure Functions, and IBM Open Whisk. Their findings showed that while serverless models were ideal for short lived, event-driven tasks, significant variability existed in cold start times, execution limits, and platform-specific constraints.

Career Point International Journal of Research (CPIJR)
©2022 CPIJR | Volume 3 | Issue 4 | ISSN: 2583-1895
July-September 2025 | DOI: https://doi.org/10.5281/zenodo.17383079

Platform Comparison and Evaluation (2017–2018): Baldini et al. (2017) performed one of the first comprehensive comparisons of serverless platforms, analysing performance, scalability, and pricing across AWS Lambda, Azure Functions, and IBM Open Whisk. Their findings showed that while serverless models were ideal for short lived, event-driven tasks, significant variability existed in cold start times, execution limits, and platform-specific constraints.

Research Gaps

Despite the growing adoption of serverless computing and its numerous advantages, several significant research and implementation gaps remain that hinder its full-scale deployment across diverse industries and applications:

1. Cold Start Latency and Performance Inconsistency

One of the most commonly cited issues in the serverless paradigm is cold start latency—when a function is invoked after being idle, there is a delay before execution begins due to container initialization. This affects user experience in latency-sensitive applications like real-time chats, payment gateways, and IoT systems. While some work has been done to mitigate this, such as provisioned concurrency in AWS Lambda, consistent solutions across platforms are still lacking.

2. Vendor Lock-in and Lack of Portability

Most serverless applications are tightly coupled to the APIs, services, and configuration formats of specific cloud providers. This makes migrating applications between providers time-consuming and complex. Research on standardizing serverless APIs or creating platform-independent deployment frameworks is still in the early stages.

3. Limited Tooling for Monitoring, Debugging, and Testing

Traditional debugging and monitoring tools do not translate well to the ephemeral, stateless environment of serverless functions. Developers often lack visibility into performance metrics, error logs, and execution flow. While commercial tools like AWS CloudWatch and

©2022 CPIJR | Volume 3 | Issue 4 | ISSN: 2583-1895
July-September 2025 | DOI: https://doi.org/10.5281/zenodo.17383079

Azure Monitor exist, comprehensive open-source alternatives and real-time tracing systems are still limited.

4. Execution Constraints and Stateless Design

Serverless functions typically have execution time limits (e.g., 15 minutes on AWS Lambda) and must remain stateless. Applications requiring long-term processes, session data, or low-latency access to stateful storage often struggle with these limitations. This restricts the kinds of applications that can be effectively built using pure serverless architecture.

5. Security and Compliance Concerns

Serverless applications often rely on third-party APIs, external services, and event-driven triggers, increasing the attack surface. Research on security models tailored to serverless, including zero-trust architecture, API throttling, and granular permission management, is still evolving. Compliance challenges like GDPR and HIPAA also raise questions about data handling and function-level access control.

Objectives of Research

The primary objective of this research is to explore the role, effectiveness, and limitations of serverless web applications in the context of modern cloud computing. As organizations increasingly move toward cloud-native solutions, understanding the full potential and drawbacks of serverless architecture is essential for developers, architects, and business decision-makers. The specific objectives of this review are as follows:

1. To Analyse the Core Benefits of Serverless Architecture

To evaluate how serverless computing enhances scalability, reduces infrastructure costs, improves development speed, and simplifies application deployment in cloud environments.

2. To Identify and Examine the Key Challenges in Serverless Computing

To study the limitations associated with serverless systems, such as cold starts, vendor lock-in, limited execution time, and the complexities of monitoring and debugging in stateless environments.

©2022 CPIJR | Volume 3 | Issue 4 | ISSN: 2583-1895
July-September 2025 | DOI: https://doi.org/10.5281/zenodo.17383079

3. To Review Existing Literature and Technological Trends

To summarize and critically evaluate existing research work, industry reports, and case studies that discuss the adoption and performance of serverless computing in real-world web application scenarios.

4. To Explore Real-World Use Cases and Applications

To understand how businesses and developers are using serverless platforms in practice across industries such as e-commerce, media, IoT, and APIs, and to assess the impact of these implementations.

5. To Highlight Gaps and Suggest Future Research Directions

To identify under-researched areas in the field of serverless computing and propose future research opportunities that can help improve its efficiency, security, and usability for a broader range of applications.

Research Methodology: -

This study adopts a qualitative, review-based methodology to systematically examine the benefits and challenges of serverless web applications in cloud computing. The methodology involves collecting, analysing, and synthesizing existing literature, technical documentation, and real-world use cases to derive insights into the current state and future potential of serverless architectures.

1. Literature Review

A comprehensive review of academic publications, industry whitepapers, and cloud provider documentation was conducted. The focus was on identifying how serverless computing is being implemented in web applications and what advantages and limitations are most frequently cited. Priority was given to research that included technical evaluations, performance metrics, and user experience data.

2. Source Selection and Data Collection

Databases and Platforms Used:

IEEE Xplore, ACM Digital Library, Google Scholar, SpringerLink, ScienceDirect, and official documentation from AWS, Azure, and Google Cloud.

Keywords for Search:

"Serverless architecture," "Function-as-a-Service (FaaS)," "serverless challenges," "cloud-native applications," "AWS Lambda," "cold start problem," "serverless scalability," and "vendor lock-in."

Inclusion Criteria:

- Research papers and articles published between 2015 and 2024
- English language only
- Focused on the use of serverless computing in cloud-based web applications
- Peer-reviewed and/or published by credible academic or industry sources

3. Model Categorization and Analysis

- Classification by Technology and Provider: Literature was categorized based on the serverless platform studied (e.g., AWS Lambda, Azure Functions, Google Cloud Functions).
- Use Case Categorization: Applications were grouped by use case—such as web hosting, API services, real-time analytics, and backend automation—to analyze performance and usability patterns.
- Evaluation Parameters: The analysis focused on parameters such as scalability, cost efficiency, cold start latency, security, and developer experience.

4.Thematic Synthesis Approach

The selected studies were reviewed thematically to draw comparisons between serverless benefits and limitations. This method enabled a structured understanding of the most common use cases, platform behaviours, and technical barriers in deploying serverless applications in a production environment.

Benefits and Challenges of Serverless Web Applications

Aspect	Benefits	Challenges

Career Point International Journal of Research (CPIJR)
©2022 CPIJR | Volume 3 | Issue 4 | ISSN: 2583-1895
July-September 2025 | DOI: https://doi.org/10.5281/zenodo.17383079

Cost	Pay-as-you-go pricing reduces infrastructure costs by charging only for execution time.	Hidden costs may occur due to overuse of functions or third-party services.
Scalability	Automatic scaling based on demand without manual configuration.	Cold starts can affect performance during sudden spikes or infrequent use.
Development Speed	Faster time to market by focusing only on writing and deploying code.	Lack of support for traditional debugging tools may slow troubleshooting.
Infrastructure Management	No need to manage or maintain servers; handled entirely by cloud provider.	Limited control over the environment; reliance on provider's infrastructure.
Availability	Built-in high availability and fault tolerance across data centers.	Service disruptions or outages from the provider can affect application uptime.
Flexibility	Easy integration with other cloud services and event triggers.	Vendor lock-in due to proprietary APIs and platform-specific services.
Execution Environment	Ideal for short-lived, stateless, event-driven applications.	Execution time limits (e.g., 15 minutes) and stateless nature restrict use cases.
Security & Compliance	Providers offer basic security features and role-based access control.	Increased attack surface; complex compliance with GDPR, HIPAA, etc.
Monitoring & Debugging	Logs and metrics tools available (e.g., CloudWatch, Azure Monitor).	Difficult to trace issues in distributed, ephemeral functions.

Career Point International Journal of Research (CPIJR)
©2022 CPIJR | Volume 3 | Issue 4 | ISSN: 2583-1895
July-September 2025 | DOI: https://doi.org/10.5281/zenodo.17383079

Discussion

Key Insights

The analysis of serverless web applications highlights a clear shift in how modern software is being developed and deployed. Serverless architecture offers numerous benefits such as reduced infrastructure management, dynamic scalability, and cost efficiency. These advantages make it highly suitable for stateless, event-driven workloads such as APIs, microservices, background jobs, and real-time data processing.

The adoption of serverless platforms like AWS Lambda, Azure Functions, and Google Cloud Functions has enabled developers to focus more on core business logic and less on

©2022 CPIJR | Volume 3 | Issue 4 | ISSN: 2583-1895
July-September 2025 | DOI: https://doi.org/10.5281/zenodo.17383079

operational concerns. Furthermore, its automatic scaling capabilities and pay-per-execution pricing model make it a cost-effective solution for startups and small-to-medium-sized enterprises.

Current Limitations

However, several challenges continue to limit its universal adoption. Cold start latency remains one of the most prominent issues, particularly in use cases requiring low-latency responses, such as financial transactions or voice-enabled interfaces. Additionally, the lack of state persistence and short execution limits restrict the development of more complex, long-running applications in a pure serverless model.

Vendor lock-in and platform dependency pose strategic risks for businesses aiming for long-term flexibility and multi-cloud support. Debugging and monitoring distributed, ephemeral functions is still a work in progress, with existing tools not offering the same level of control as in traditional architectures.

Security is another concern. The broader attack surface, along with the use of multiple cloud services and external event triggers, increases the complexity of securing serverless applications. Ensuring data privacy and compliance with regulations like GDPR or HIPAA is challenging due to reduced visibility into the backend infrastructure.

Balancing Innovation and Risk

Despite these limitations, serverless computing is advancing quickly. The concept of "serverless 2.0" is emerging, with efforts to integrate serverless with containers, improve cold start times, and offer multi-cloud compatibility. Hybrid architectures are also gaining popularity, allowing developers to combine the agility of serverless with the flexibility of containerized environments.

To fully realize the benefits of serverless, developers, researchers, and cloud providers must collaborate to address its challenges. This includes improving observability, developing standard frameworks, enhancing portability, and creating robust security protocols tailored to serverless environments.

Conclusion: -

Career Point International Journal of Research (CPIJR)
©2022 CPIJR | Volume 3 | Issue 4 | ISSN: 2583-1895
July-September 2025 | DOI: https://doi.org/10.5281/zenodo.17383079

Serverless computing has emerged as a powerful and transformative model in the evolution of cloud-based web applications. By abstracting infrastructure management and offering event-driven execution, serverless platforms enable developers to build and deploy applications with unprecedented speed, flexibility, and cost efficiency. As demonstrated throughout this review, serverless architectures are particularly beneficial for microservices, APIs, IoT solutions, and real-time data processing, where dynamic scalability and minimal operational overhead are critical.

However, the serverless model is not without its challenges. Cold start latency, limited execution time, stateless design, and vendor lock-in are persistent concerns that can hinder the performance, portability, and long-term viability of serverless applications. Furthermore, the lack of standardized debugging tools, complex security considerations, and compliance requirements pose additional risks in production environments.

Despite these issues, serverless computing continues to evolve rapidly. Ongoing research and innovation are addressing many of its current shortcomings through hybrid models, multi-cloud strategies, and improved development tools. Organizations and developers must therefore evaluate serverless architecture not as a one-size-fits-all solution, but as a strategic component within a broader application ecosystem.

In conclusion, serverless web applications represent a significant step forward in cloud computing. When implemented thoughtfully and with an understanding of both its strengths and limitations, serverless architecture can deliver scalable, efficient, and resilient applications. Future work should focus on enhancing cross-platform support, mitigating cold start issues, and strengthening security frameworks to ensure the reliable and responsible growth of serverless technologies.

©2022 CPIJR | Volume 3 | Issue 4 | ISSN: 2583-1895
July-September 2025 | DOI: https://doi.org/10.5281/zenodo.17383079

References: -

1. Roberts, M. (2016). *Serverless Architectures*. Retrieved from: https://martinfowler.com/articles/serverless.html

2. Jonas, E., Schleier-Smith, J., Sreekanti, V., Tsai, C., Khandelwal, A., Pu, Q., ... & Stoica, I. (2019). *Cloud Programming Simplified: A Berkeley View on Serverless Computing*. UC Berkeley.

3. Baldini, I., Castro, P., Chang, K., Cheng, P., Fink, S., Ishakian, V., ... & Muthusamy, V. (2017). *Serverless Computing: Current Trends and Open Problems*. In Proceedings of the 2017 Workshop on Serverless Computing (WoSC), ACM.

4. McGrath, G., & Brenner, P. (2022). *Serverless Computing: Design, Implementation, and Performance*. Journal of Cloud Computing.

5. Eyk, E. van, Iosup, A., Seif, S., & Thömmes, M. (2023). *Understanding Function-as-a-Service Platforms: Characteristics and Performance*. IEEE Internet Computing.

6. Shillaker, J., & Pietzuch, P. (2021). *Faasm: Lightweight Isolation for Efficient Stateful Serverless Computing*. USENIX Annual Technical Conference.

7. AWS Lambda Documentation. Available at: https://docs.aws.amazon.com/lambda

8. Azure Functions Documentation. Available at: https://learn.microsoft.com/en-us/azure/azure-functions/

9. Google Cloud Functions. Available at: https://cloud.google.com/functions

10. Adzic, G., & Chatley, R. (2017). *Serverless Computing: Economic and Architectural Impact*. In Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering.