Partners Universal International Research Journal (PUIRJ)

Volume: 02 Issue: 02 | April-June 2023 | ISSN: 2583-5602 | www.puirj.com

Unshackled by Servers: Embracing the Serverless Revolution in Modern Computing

Dr.A.Shaji George[1], A.S.Hovan George[2], Dr.T.Baskar[3]

[1,2]Masters IT Solutions, Chennai, Tamil Nadu, India.

[3]Professor, Department of Physics, Shree Sathyam College of Engineering and Technology, Sankari Taluk, Salem District, Tamil Nadu, India.

Abstract - Over the past few years, serverless computing has emerged as a transformative paradigm in contemporary computing, supplying a brand new degree of abstraction and scalability to developers and corporations. This research survey aims to provide a comprehensive evaluation of serverless computing, exploring its foundations, platforms, use instances, challenges, and future directions. We start by analyzing the fundamental principles of serverless computing, such as its key additives and advantages. Next, we evaluate and compare the principle serverless systems and services, which include AWS Lambda, Google Cloud Functions, Microsoft Azure Functions, and IBM Cloud Functions. We also speak of actual international use instances and applications of serverless computing, which include web and cellular applications, statistics processing, IoT, and system getting to know. Furthermore, we discover the technical and organizational demands related to serverless computing, which include safety, privacy, and supplier lock-in. Finally, we discover the rising tendencies and study possibilities in serverless computing, highlighting advances in serverless technologies, integration with other cloud services, standardization efforts, and new utility domains. Our well-known survey shows that serverless computing has the potential to seriously impact the destiny of software program improvement and deployment, offering new possibilities for innovation and boom in numerous industries and domains.

Keywords: Serverless Computing, Function-as-a-Service (FaaS), Cloud Computing, Infrastructure Abstraction, Event-Driven Architecture, Automatic Scaling, Pay-Per-Use Pricing, Stateless Functions, Microservices, Cloud Providers.

1. INTRODUCTION

In recent years, the advancements in cloud computing have given rise to a new paradigm known as serverless computing. This research survey aims to provide a comprehensive understanding of serverless computing, its history, advantages, and main objectives.

1.1. Background and Motivation

Serverless computing, also known as Function-as-a-Service (FaaS), has emerged as a promising alternative to traditional cloud computing models. The term "serverless" does not imply the absence of servers but rather the abstraction of server management for developers. In this model, developers can focus on writing code while the cloud provider takes care of underlying infrastructure, server provisioning, and maintenance.

The origins of serverless computing can be traced back to the early 2010s, with the introduction of AWS Lambda by Amazon Web Services in 2014. Since then, other major cloud providers, such as Google Cloud, Microsoft Azure, and IBM, have launched their own serverless offerings.

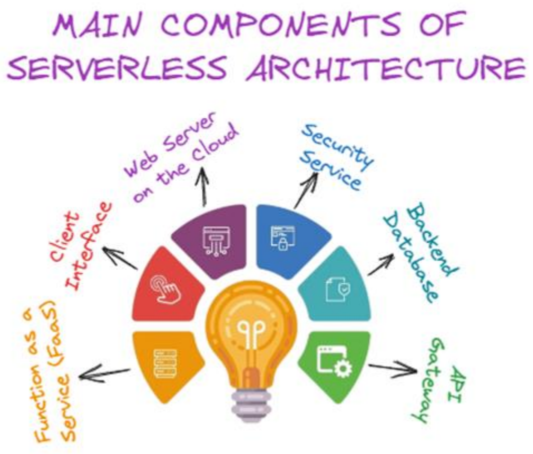

Fig -1: Serverless Architecture

The motivation for this research survey comes from the growing interest in serverless computing due to its potential benefits, such as reduced operational complexity, cost savings, and improved scalability. As more organizations adopt serverless architectures, it is crucial to understand its underlying principles, challenges, and future directions.

1.2. Objectives and Scope

The primary objectives of this research survey are to:

- Provide a comprehensive overview of serverless computing, its history, and evolution.
- Discuss the advantages and drawbacks of adopting a serverless architecture.
- Explore the state-of-the-art research and industry use cases related to serverless computing.
- Identify open challenges and future research directions in serverless computing.

The scope of this survey includes various aspects of serverless computing, such as platforms, programming models, performance, security, and cost optimization.

1.3. Organization of the Paper

The remainder of this research survey is organized as follows:

Section 2 provides an overview of serverless computing, its history, and the main concepts.

Section 3 discusses the advantages and disadvantages of serverless computing.

Section 4 presents the state-of-the-art research and industry use cases, including case studies and best practices.

Section 5 explores the challenges and future directions in serverless computing research.

Section 6 concludes the paper with a summary of the key findings and contributions.

2. FOUNDATIONS OF SERVERLESS COMPUTING

In this section, we discuss the basic concepts and principles of serverless computing, including its definition, key components, and main characteristics.

2.1. Definition of Serverless Computing

Serverless computing, also known as Function-as-a-Service (FaaS), is a cloud computing paradigm that abstracts away the underlying infrastructure and server management tasks, allowing developers to focus on writing code and implementing application logic. In a serverless architecture, the cloud provider is responsible for provisioning, scaling, and maintaining servers, while the developers only need to worry about their code and its execution.

2.2. Key Components of Serverless Computing

There are several key components in a serverless computing environment:

Functions: Functions are the fundamental building blocks in a serverless architecture. A function is a piece of code that performs a specific task and can be triggered by an event or an API call. Functions are stateless, meaning they do not store any state information between invocations.

Events: Events are the triggers that initiate the execution of a function. These can be external events, such as a user request, an API call, or a message from another service, or internal events, such as a scheduled timer.

Event Sources: Event sources are the components that generate events and send them to the serverless platform. Examples include API gateways, message queues, and database triggers.

Serverless Platform: The serverless platform is responsible for managing the execution of functions. It provisions the necessary resources, such as computing power and memory, and automatically scales the infrastructure based on demand. The platform also takes care of fault tolerance, load balancing, and other operational aspects.

State Management: Since serverless functions are stateless, state management is often handled by external services. This can include databases, object storage, or caching services.

2.3. Main Characteristics of Serverless Computing

Serverless computing exhibits several key characteristics:

Abstraction of Infrastructure: Serverless computing abstracts away the underlying infrastructure and server management tasks, allowing developers to focus on writing code and implementing application logic.

Event-Driven: Serverless architectures are typically event-driven, meaning functions are triggered by events, and the platform takes care of managing the execution of these functions.

Automatic Scaling: Serverless platforms automatically scale resources based on demand, ensuring that the infrastructure can handle varying workloads without manual intervention.

Pay-Per-Use Pricing: In a serverless environment, users only pay for the resources consumed during the execution of their functions, rather than pre-allocating resources.

Stateless Functions: Functions in a serverless architecture are stateless, meaning they do not store any state information between invocations. This enables better scalability and fault tolerance.

Microservices and Granularity: Serverless computing promotes the use of microservices, allowing developers to build applications as a collection of small, independently deployable functions. This enables better modularity, flexibility, and maintainability.

By understanding these foundational concepts, we can better appreciate the advantages and challenges associated with serverless computing, which will be discussed in the following sections.

3. SERVERLESS COMPUTING PLATFORMS AND SERVICES

In this section, we provide an overview of the main serverless computing platforms and services, including a comparison of their features, pricing, and performance.

3.1. Amazon Web Services (AWS) Lambda

Amazon Web Services (AWS) Lambda is a pioneer in the serverless computing space, launched in 2014. It supports multiple programming languages, including Node.js, Python, Java, Go, and .NET. Lambda integrates with several AWS services, such as API Gateway, S3, and DynamoDB, enabling seamless scalability and event-driven architectures.

Features:

Multiple language support

Automatic scaling

Event-driven execution

Integration with other AWS services

Built-in logging and monitoring

Pricing:

AWS Lambda follows a pay-per-use pricing model, with charges based on the number of requests and the compute time (measured in GB-seconds) consumed during function execution. The first 1 million requests and 400,000 GB-seconds of compute time are free each month.

3.2. Google Cloud Functions

Google Cloud Functions is Google's serverless offering, launched in 2016. It supports Node.js, Python, Go, Java, and .NET. Cloud Functions integrates with other Google Cloud services, such as Pub/Sub, Firestore, and Cloud Storage, to enable event-driven architectures and seamless scalability.

Features:

Multiple language support

Automatic scaling

Event-driven execution

Integration with Google Cloud services

Built-in logging and monitoring

Pricing:

Google Cloud Functions follows a pay-per-use pricing model based on the number of invocations, compute time, and memory allocation. The first 2 million invocations, 400,000 GB-seconds, and 200,000 GHz-seconds are free each month.

3.3. Microsoft Azure Functions

Microsoft Azure Functions is Microsoft's serverless offering, launched in 2016. It supports a wide range of programming languages, including C#, Java, JavaScript, Python, and PowerShell. Azure Functions integrates with other Azure services, such as Event Hubs, Cosmos DB, and Storage, to enable event-driven architectures and seamless scalability.

Features:

Multiple language support

Automatic scaling

Event-driven execution

Integration with Azure services

Built-in logging and monitoring

Pricing:

Azure Functions follows a pay-per-use pricing model based on the number of executions, memory allocation, and execution time. The first 1 million executions and 400,000 GB-seconds of execution time are free each month.

3.4. IBM Cloud Functions

IBM Cloud Functions is IBM's serverless offering, based on the open-source Apache OpenWhisk project. It supports several programming languages, including Node.js, Python, Java, Swift, and PHP. Cloud Functions integrates with other IBM Cloud services, such as Watson, Cloudant, and Message Hub, to enable event-driven architectures and seamless scalability.

Features:

Multiple language support

Automatic scaling

Event-driven execution

Integration with IBM Cloud services

Built-in logging and monitoring

Pricing:

IBM Cloud Functions follows a pay-per-use pricing model based on the number of invocations, memory allocation, and execution time. The first 5 million invocations, 375,000 GB-seconds, and 150,000 GHz-seconds are free each month.

3.5. Other Notable Serverless Platforms

In addition to the major cloud providers, there are several other notable serverless platforms:

Alibaba Cloud Function Compute: Alibaba Cloud's serverless offering, supporting Node.js, Python, Java, and .NET.

Oracle Functions: Oracle's serverless platform, based on the open-source Fn Project, which supports multiple programming languages.

Vercel: A serverless platform focused on frontend and static site deployments, offering support for Node.js and Go.

Netlify Functions: A serverless platform integrated with Netlify's static site hosting service, supporting JavaScript and TypeScript.

Each of these platforms has its unique features, pricing models, and performance characteristics. When choosing a serverless platform, factors such as language support, integration with existing services, scalability requirements, and pricing should be considered.

4. USE CASES AND APPLICATIONS

In this section, we describe the main use cases and applications of serverless computing, including real-world examples from different industries and domains.

4.1. Web and Mobile Applications

Serverless computing has become popular in the development of web and mobile applications, as it allows for rapid development, automatic scaling, and cost-effective deployment. Example use cases include:

APIs: Developers can create serverless functions to power RESTful APIs, allowing for seamless communication between the frontend and backend services.

Authentication: Serverless functions can be used to implement secure authentication mechanisms, such as OAuth or JWT, in web and mobile applications.

Real-time data processing: Serverless architectures can handle real-time data processing for features like chat, notifications, and live updates.

Server-side rendering: Serverless functions can be used for server-side rendering of web pages, improving performance and SEO.

4.2. Data Processing and Analytics

Serverless computing is well-suited for data processing and analytics tasks due to its on-demand scalability and pay-per-use pricing model. Example use cases include:

ETL (Extract, Transform, Load) pipelines: Serverless functions can be used to clean, transform, and load data into data warehouses or storage systems.

Stream processing: Serverless architectures can be used to process and analyze data streams in real-time, such as social media feeds, log files, or IoT device data.

Big Data processing: Serverless platforms can be used to run distributed data processing frameworks like Apache Spark or Hadoop, enabling large-scale data analysis.

4.3. Internet of Things (IoT)

Serverless computing is a natural fit for IoT applications, as it can handle the variable workloads and large data volumes generated by connected devices. Example use cases include:

Device management: Serverless functions can be used to manage and control IoT devices, such as provisioning, firmware updates, and device monitoring.

Data processing: Serverless architectures can process and analyze data generated by IoT devices in real-time, enabling features like anomaly detection, predictive maintenance, and real-time alerts.

Integration with other services: Serverless functions can integrate with other cloud services, such as databases, analytics platforms, and messaging systems, to store and process IoT data.

4.4. Machine Learning and Artificial Intelligence

Serverless computing can be used to build and deploy machine learning (ML) and artificial intelligence (AI) applications, offering benefits such as scalability, cost savings, and simplified management. Example use cases include:

Model training: Serverless platforms can be used to train ML models on large datasets, leveraging the automatic scaling and pay-per-use pricing model.

Model deployment: Serverless functions can be used to deploy ML models as APIs, enabling real-time predictions and inferences.

Data preprocessing: Serverless architectures can be used to preprocess and clean data for ML training, such as feature extraction, data normalization, or data augmentation.

4.5. Other Emerging Use Cases

Serverless computing is continuously evolving, and new use cases are emerging across various industries and domains:

Microservices: Serverless platforms can be used to build and deploy microservices, enabling better modularity, flexibility, and maintainability in large-scale applications.

Chatbots and voice assistants: Serverless functions can be used to power chatbots and voice assistants, enabling real-time natural language processing and integration with other services.

Edge computing: Serverless platforms can be deployed at the edge of the network, enabling low-latency data processing and decision-making in applications like autonomous vehicles, smart cities, and real-time analytics.

Serverless computing has proven to be a versatile technology with a wide range of use cases across various industries and domains. Its benefits, such as automatic scaling, cost savings, and simplified management, make it an attractive option for both startups and enterprises alike.

5. CHALLENGES AND LIMITATIONS

In this section, we discuss the challenges and limitations associated with serverless computing, both from a technical and an organizational perspective.

5.1. Technical Challenges and Limitations

Serverless computing comes with several technical challenges and limitations that developers must consider:

Cold starts: When a serverless function is first invoked, there may be a delay (cold start) while the underlying infrastructure provisions resources. This can lead to increased latency for end-users, especially in applications that require low-latency responses.

Statelessness: Serverless functions are stateless by design, meaning they do not maintain any state between invocations. Developers must manage state externally, often using databases or other storage systems, which can add complexity and potential performance bottlenecks.

Resource limits: Serverless platforms typically impose limits on resources such as memory, CPU, and execution time. These limitations can affect the types of workloads that can be effectively run on serverless platforms and may require developers to optimize their code for these constraints.

Debugging and monitoring: Debugging and monitoring serverless applications can be complex due to the distributed nature of the architecture and the lack of direct access to the underlying infrastructure.

5.2. Organizational and Cultural Challenges

Adopting serverless computing can also present organizational and cultural challenges:

Skillset: Developing and managing serverless applications requires a different skillset than traditional application development. Organizations may need to invest in training and hiring to build expertise in serverless technologies.

Change management: Adopting serverless computing may require changes in development processes, team structures, and ways of working, which can be challenging to implement and manage.

5.3. Security and Privacy Concerns

Serverless computing introduces new security and privacy concerns that must be addressed:

Function-level security: Due to the granular nature of serverless functions, developers must ensure that each function is properly secured, including access controls, input validation, and error handling.

Data storage: As serverless applications often rely on external storage systems, developers must ensure that data is securely stored, transmitted, and processed. This can involve encryption, access controls, and compliance with data protection regulations.

Third-party dependencies: Serverless applications often rely on third-party libraries and services, which can introduce security vulnerabilities if not properly managed and updated.

5.4. Vendor Lock-In and Portability Issues

Serverless computing can lead to vendor lock-in and portability issues:

Vendor-specific services: Many serverless platforms offer proprietary services and integrations that can make it difficult to migrate applications between providers or to on-premises infrastructure.

Platform differences: Different serverless platforms have varying implementations, language support, and feature sets, which can make it challenging to build portable applications that can be easily migrated between providers.

Despite these challenges and limitations, serverless computing continues to grow in popularity due to its numerous benefits, such as automatic scaling, cost savings, and simplified management. By carefully considering these challenges and adopting best practices, organizations can successfully leverage serverless computing to build and deploy innovative applications.

6. FUTURE DIRECTIONS AND RESEARCH OPPORTUNITIES

In this section, we explore the future directions and research opportunities in serverless computing, highlighting emerging trends and technologies that could shape its evolution.

6.1. Advances in Serverless Technologies

As serverless computing continues to evolve, we can expect advances in the underlying technologies that address existing challenges and limitations:

Improved cold start performance: Researchers and vendors are working to minimize the impact of cold starts on serverless applications, through techniques such as predictive provisioning, function instance pooling, and other optimizations.

Support for stateful functions: Efforts are underway to develop serverless platforms that allow for stateful functions, enabling developers to more easily manage state in their applications without relying on external storage systems.

Enhanced debugging and monitoring tools: As serverless computing matures, we can expect more advanced debugging and monitoring tools that provide better visibility into the performance and behavior of serverless applications.

6.2. Integration with Other Cloud Services and Technologies

Serverless computing will likely become more tightly integrated with other cloud services and technologies, enabling new capabilities and use cases:

Hybrid and multi-cloud architectures: As organizations adopt hybrid and multi-cloud strategies, serverless platforms will need to support seamless integration with on-premises infrastructure and other cloud providers.

Edge computing: With the growing importance of edge computing, we can expect serverless platforms to be deployed closer to the edge of the network, enabling low-latency data processing and decision-making in applications like autonomous vehicles, smart cities, and real-time analytics.

Containers and microservices: As serverless computing and containerization continue to converge, we can expect more seamless integration between the two technologies, enabling developers to build and deploy applications using a mix of serverless functions and containerized microservices.

6.3. Standardization and Interoperability Efforts

Efforts towards standardization and interoperability in serverless computing will help address challenges related to vendor lock-in and portability:

Open standards: The development of open standards for serverless platforms, such as the CloudEvents specification and the Function as a Service (FaaS) Runtime API, will help promote interoperability between different providers and reduce the risk of vendor lock-in.

Open-source platforms: The growth of open-source serverless platforms, such as OpenFaaS, Kubeless, and Knative, will enable greater flexibility and control for organizations looking to build and deploy serverless applications.

6.4. New Application Areas and Business Models

As serverless computing continues to mature, we can expect new application areas and business models to emerge:

Emerging industries: Serverless computing will likely be adopted in new industries and domains, such as healthcare, finance, and transportation, enabling innovative applications and services that leverage its unique benefits.

New business models: The pay-per-use pricing model of serverless computing may give rise to new business models and revenue streams, such as usage-based billing for APIs or tiered pricing for data processing services.

In conclusion, serverless computing is a rapidly evolving field with numerous future directions and research opportunities. Through advances in technology, integration with other cloud services, standardization efforts, and the exploration of new application areas, serverless computing will continue to shape the future of cloud computing and application development.

7. CONCLUSION

Serverless computing has emerged as a transformative technology in modern computing, offering numerous benefits such as automatic scaling, cost savings, and simplified management. By abstracting away the underlying infrastructure, serverless computing enables developers to focus on writing code and delivering value, while the cloud provider handles the operational aspects of managing and scaling the application. This research survey has highlighted the key concepts, use cases, challenges, and future directions in serverless computing. We have explored the fundamental principles of serverless computing, including Function as a Service (FaaS) and Backend as a Service (BaaS), and discussed how these technologies enable a wide range of applications across various industries and domains, such as web and mobile applications, data processing and analytics, IoT, machine learning, and artificial intelligence. However, serverless computing is not without its challenges and limitations. We have examined the technical, organizational, and security-related challenges associated with serverless computing, as well as issues related to vendor lock-in and portability. By addressing these challenges and adopting best practices, organizations can successfully leverage serverless computing to build and deploy innovative applications. Looking ahead, there are numerous opportunities for research and innovation in serverless computing. Advances in serverless technologies, integration with other cloud services and technologies, standardization and interoperability efforts, and the exploration of new application areas and business models will all contribute to the continued growth and impact of serverless computing in modern computing. In conclusion, serverless computing has the potential to significantly shape the future of cloud computing and application development. As researchers and practitioners continue to explore and harness the power of serverless computing, we can expect to see new and innovative applications that leverage its unique benefits and capabilities.

REFERENCES

[1] Følstad, Asbjørn, et al. "Future Directions for Chatbot Research: An Interdisciplinary Research Agenda - Computing." SpringerLink, 19 Oct. 2021, https://doi.org/10.1007/s00607-021-01016-7.

[2] A.SHAJI GEORGE, & BASHIRU AREMU. (2021). USER REVOCATION USING ADVANCED KEY GENERATION IN CLOUD ARCHITECTURE. Ciencia and Engenharia/ Science and Engineering Journal, 25(03), 36–45. https://doi.org/10.5281/zenodo.7028244

[3] "Future AI Applications for Businesses and Their Benefits." PixelPlex, 31 July 2019, pixelplex.io/blog/future-ai-applications-benefits-impact-on-business.

[4] Witekio, Master. "The Future of Serverless Computing: 5 Trends | Witekio." Witekio - Experten Für Embedded Systems & IoT, 4 Aug. 2021, witekio.com/de/blog-de/the-future-of-serverless-computing-5-trends.

[5] Dr. A.SHAJI GEORGE, & A.S.HOVAN GEORGE. (2021). Serverless Computing: the Next Stage in Cloud Computing's Evolution and an Empowerment of a New Generation of Developers. International Journal of All Research Education and Scientific Methods (IJARESM), 9(4), 21–35. https://doi.org/10.5281/zenodo.7027409

[6] "Serverless Computing 101: How Business Can Use Serverless and Benefit From It." Serverless Computing 101: How Business Can Use Serverless and Benefit From It, www.agilevision.io/blog/serverless-computing-101-how-business-can-use-serverless-and-benefit-from-it.
[7] "The Future of Serverless Computing and Its Use Cases." The Future of Serverless Computing and Its Use Cases, 15 June 2023, www.xenonstack.com/blog/serverless-computing-use-cases.
[8] The Web3 Revolution: Navigating the Future of Decentralized Networks , Book Rivers , ISBN-13 :978-93-5515-978-6 , 2023.
[9] "An Overview of the Serverless Computing Architecture Trend." Data Center Knowledge | News and Analysis for the Data Center Industry, 25 Mar. 2021, www.datacenterknowledge.com/cloud/overview-serverless-computing-architecture-trend.
[10] "13 Best Serverless Computing Platforms and Database Providers." Prisma's Data Guide, www.prisma.io/dataguide/serverless/serverless-comparison.
[11] The Future of Understanding Human Behavior: Exploring Predictive Analysis Trends , Book Rivers, ISBN: 978-93-5515-955-7, A.Shaji George, A.S Hovan George 2023.
[12] Data Lakes vs Data Wells: Which is the Best Fit for Your FMCG Business , Book Rivers, ISBN: 9355159226 , A.Shaji George , Sagayarajan Fernando ,Yazeed AlMatroudi, A.S Hovan George 2023.
[13] Community, Alibaba Cloud. "What Is Serverless Computing." Alibaba Cloud Community, www.alibabacloud.com/blog/what-is-serverless-computing_597254.
[14] A.S.HOVAN GEORGE, MASCHIO FERNANDO, Dr. A.SHAJI GEORGE, Dr. T. BASKAR, & DIGVIJAY PANDEY. (2021). Metaverse: The Next Stage of Human Culture and the Internet. International Journal of Advanced Research Trends in Engineering and Technology (IJARTET), 8(12), 1–10. https://doi.org/10.5281/zenodo.6548172
[15] "What Is Serverless Computing? Definition From TechTarget.com." IT Operations, 1 Nov. 2022, www.techtarget.com/searchitoperations/definition/serverless-computing.
[16] Kumar, Chandan. "8 Serverless Computing Platforms to Run Your Application Code." Geekflare, 13 Sept. 2018, geekflare.com/serverless-computing-platform.
[17] Marin, Eduard, et al. "Serverless Computing: A Security Perspective - Journal of Cloud Computing." SpringerOpen, 23 Oct. 2022, https://doi.org/10.1186/s13677-022-00347-w.
[18] Dr. A.SHAJI GEORGE, & A.S.HOVAN GEORGE. (2022). Potential Risk: Hosting Cloud Services Outside the Country. International Journal of Advanced Research in Computer and Communication Engineering, 11(4), 5–11. https://doi.org/10.5281/zenodo.6548114
[19] "Top 10 Serverless Computing Platforms and Databases." Top 10 Serverless Computing Platforms and Databases, 19 Apr. 2023, www.xenonstack.com/blog/serverless-computing-platforms.
[20] "Serverless Computing - the Complete Guide." Confluent, www.confluent.io/learn/serverless-computing.
[21] Dr. A. Shaji George, & A.S. Hovan George. (2022). Data Sharing Made Easy by Technology Trends: New Data Sharing and Privacy Preserving Technologies that Bring in a New Era of Data Monetization. Partners Universal International Research Journal (PUIRJ), 01(03), 13–19. https://doi.org/10.5281/zenodo.7111123
[22] keivan. "Serverless Computing I - Cryptiot." Cryptiot, 20 Sept. 2018, cryptiot.de/cloud-computing/serverless-computing-1.
[23] Castro, Paul, et al. "The Server Is Dead, Long Live the Server: Rise of Serverless Computing, Overview of Current State and Future Trends in Research and Industry." arXiv.org, 7 June 2019, arxiv.org/abs/1906.02888v1.
[24] "What Serverless Computing Really Means, and Everything Else You Need to Know." ZDNET.
[25] "What Is Serverless Computing? Applications and Architecture." Prisma's Data Guide.
[26] Dr. A.Shaji George, & A.S.Hovan George. (2022). A Review of Moonlighting in the IT Sector And its Impact. Partners Universal International Research Journal (PUIRJ), 01(03), 64–73. https://doi.org/10.5281/zenodo.7114049
[27] "What Is Serverless Computing and FaaS? | a Full Guide Into Serverless | What Is Serverless Computing." What Is Serverless Computing and FaaS? | a Full Guide Into Serverless | What Is Serverless Computing, www.alpacked.io/blog/what-is-serverless-computing.
[28] Das, Anirban, and Supriyo Biswas. "Serverless Computing, Explained." Boolean World, 27 June 2017, www.booleanworld.com/serverless-computing-explained.
[29] "Introduction to Serverless Computing | a Comprehensive Guide." Introduction to Serverless Computing | a Comprehensive Guide, 17 Apr. 2023, www.xenonstack.com/insights/serverless-computing.
[30] Dr.A. Shaji George, A.S.HOVAN GEORGE, Dr.T. Baskar, & A.S.Gabrio Martin. (2023). Human Insight AI: An Innovative Technology Bridging The Gap Between Humans And Machines For a Safe, Sustainable Future.

Partners Universal International Research Journal (PUIRJ) ISSN: 2583-5602, 02(01), 1–15. https://doi.org/10.5281/zenodo.7723117

[31] "What Is Serverless Computing? | a Comprehensive Serverless Guide." What Is Serverless Computing? | a Comprehensive Serverless Guide | Elastic, www.elastic.co/what-is/serverless-computing.

[32] Dr.A.Shaji George, & A.S.Hovan George. (2023). Revolutionizing Manufacturing: Exploring the Promises and Challenges of Industry 5.0. Partners Universal International Innovation Journal (PUIIJ), 01(02), 22–38. https://doi.org/10.5281/zenodo.7852124

[33] "A Survey on Serverless Computing." DeepAI, 20 June 2021, deepai.org/publication/a-survey-on-serverless-computing.

[34] Maayan, Gilad David. "How The Cloud Has Evolved Over the Past 10 Years - DATAVERSITY." DATAVERSITY, 6 Apr. 2021, www.dataversity.net/how-the-cloud-has-evolved-over-the-past-10-years.

[35] A.SHAJI GEORGE, & BASHIRU AREMU. (2021). USER REVOCATION USING ADVANCED KEY GENERATION IN CLOUD ARCHITECTURE. Ciencia and Engenharia/ Science and Engineering Journal, 25(03), 36–45. https://doi.org/10.5281/zenodo.7028244

[36] Makhov, Vitaly. "Server Vs. Serverless: Benefits and Downsides | Nordic APIs |." Nordic APIs, 25 Nov. 2021, nordicapis.com/server-vs-serverless-benefits-and-downsides.

[37] "Serverless Architecture." MongoDB, www.mongodb.com/serverless.