Volume 1 No 3, 2025
Journal of Global Research in Multidisciplinary Studies (JGRMS)
Research Paper
Available online at https://saanvipublications.com/journals/index.php/jgrms/index
ISSN: 3049-2963

Serverless Computing and Cloud-Native Applications Trends in AWS, Kubernetes, and DevOps

Mrs. Neha Upadhyay,
Assistant Professor,
Department of Computer Science and Engineering
IES College Of Technology, Bhopal
nehaupadhyay2080@gmail.com

***Abstract*—Software development evolution lately resulted in extensive use of cloud-native with serverless computing structures, which changed the way applications get designed for deployment. This paper delivers an extensive evaluation of the latest developments in cloud-native and serverless models that focuses especially on DevOps methodology integration. Cloud-native architectures with their features of containerization and microservices, and orchestration bring agility and scalability alongside fault tolerance capacity. Through serverless computing, developers gain complete infrastructure abstraction to release code automatically with minimal operating costs when responding to system events. Through their union with DevOps practices, this technology combination speeds up software development through continuous integration and delivery, and monitoring. Discussed within the paper are the major challenges of security and performance optimization and vendor lock-in, and observability requirements. New innovation trends based on AI automation alongside platform engineering and event-driven architecture design elements, now shape the direction of DevOps practices. The research paper summarizes upcoming directions that consist of multi-cloud implementations along with improved toolchains and enhanced governance procedures.**

Keywords—Serverless Computing, Cloud-Native Applications, AWS Lambda, Kubernetes, DevOps, Cloud Computing Trends. Containerization.

I. INTRODUCTION

In recent years, the progression of cloud technologies has augmented the adoption of serverless computing and cloud-native applications. Leveraging platforms like AWS, Kubernetes, and DevOps practices, these modern approaches offer enhanced scalability, automation, and flexibility for building efficient and resilient software systems [1]. Serverless computing is characterized by actions and occurrences that lead to the completion of services, providing a partial realization of this ideal. Active database systems are suggestive of this language, and wider computer systems where operations are performed reactively to event streams have long been theorized in the event-driven domain [2]. Serverless function providers completely adopt these concepts, implementing event-processing logic across their clouds and expressing actions using simple functional abstractions.

A greater number of apps are being created specifically for the cloud architecture due to the rapid expansion of computing through the cloud [3]. Additionally, the need to move monolithic apps to cloud computing environments is growing. The design concepts of cloud native apps vary from those of application monoliths because of the disparities in equipment.

Software programs are developed. Cloud-native applications use a cloud-first approach that leverages online data processing technology from cloud providers (both on and off premise) to achieve greater quality, speed, and scalability while reducing the hazards of implementation [4]. Container-based software creation has become more popular since it empowers establishments to generate fault-tolerant and extremely resilient apps.

The cloud computing system has revolutionized the IT industry by delivering on-demand, flexible computing resources over an internet connection. Offering a vast array of platforms, software [5], and infrastructural services to corporations of all varieties, Amazon Web Services (AWS) is a top contractor of cloud computing services. The main attributes and advantages of AWS cloud computing are examined in this abstract, encompassing the effectiveness of costs, adaptability, adaptability, and dependability.

In contemporary cloud computing settings, cloud-native constructions have become a key strategy for creating scalable, robust, and economical software [6]. Among the dominant paradigms in cloud-native development, Kubernetes and Serverless Computing represent two distinct yet influential models that simplify the effective positioning, management, and scaling of submissions by organizations [7]. DevOps is a collection of procedures that blends IT operations (Ops) with the development of software (Dev) [8]. DevOps is an on company's collaborative in nature, multifunctional endeavor to ensure the accuracy and dependability of new software versions by automating their continuous delivery. The goal of DevOps and tools like the CI/CD pipeline is to finish the project on time and prevent frequent code changes while developing [9]. This approach aligns seamlessly with serverless computing and AWS, where DevOps practices enable automated deployments, scalability, and efficient management of serverless applications.

A. *Structured of the Paper*

This is how the research piece is structured Section II examines serverless computing in cloud applications. Section III outlines cloud-native application fundamentals. Section IV discusses AWS, DevOps, and Kubernetes integration. Section V literature of review. Section VI completes with insights and future directions.

II. SERVERLESS COMPUTING IN CLOUD APPLICATIONS

A result in large part to the current transition of business software structures to microservices and a series of micro computing without servers, or simply put, serverless computing is emerging as a new and attractive architecture for online implementation distribution [10]

From the standpoint of a cloud service supplier, serverless IT offers an environment that promotes the use of other services throughout their ecological systems, lowers the determination needed to create and oversee cloud-based apps, and gives them more control over the entire development stack. It also lowers operating costs through the effective optimization and administration of cloud resources.

A. *Serverless VS. Traditional Cloud Computing Models*

Cloud computing has revolutionized how organizations approach infrastructure, application deployment, and service delivery. Figure 1 compares traditional and serverless computing models. In the traditional approach [11], applications run on managed virtual servers or containers, requiring manual infrastructure management for front-end logic, back-end processing, security, and databases [12]. In contrast, the serverless model offloads infrastructure management to cloud providers, using distributed services for scalability and cost efficiency. This shift enables Cloud storage to manage scalability and operations dynamically, allowing programmers to focus on the application's functionality.

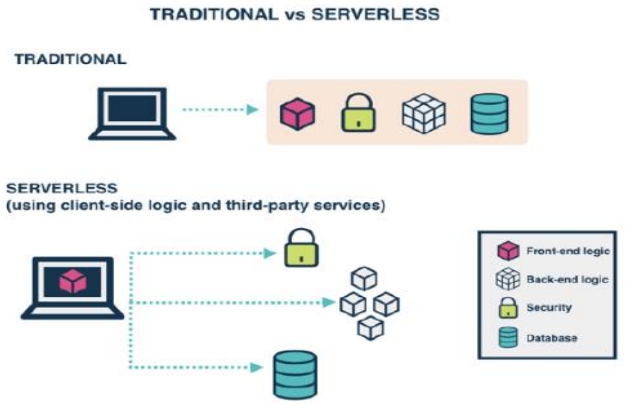

Fig. 1. Comparison of Traditional vs. Serverless Computing Models

Serverless computing automates infrastructure management, scaling, and cost optimization, unlike traditional cloud models that require manual provisioning.

- **Virtual Management:** When using traditional cloud computing, users must handle virtualized servers or containers that execute applications through manual provisioning and scaling in addition to maintenance tasks. With serverless computing, developers can work on code rather than servers since this platform abstracts the infrastructure components.
- **Infrastructure Management:** Acquiring traditional cloud services requires direct server or container management to run applications, which demands active involvement in setup operations and resource expansion and support work. The infrastructure abstraction feature of Serverless computing enables developers to work on code creation without needing server management responsibilities [13].
- **Scalability:** Event-triggered automatic scaling stands as a natural benefit of serverless architectures instead of the manual process needed in traditional architectures.
- **Cost Structure:** Organizations spend money on pre-computed resources through traditional models but frequently lose value from unused resources [14]. The cost structure of serverless computing depends on what resources customers actually use thus making it an economically desirable solution for variable workload operations.

B. *Key Benefits of Serverless Cloud Computing*

There are several benefits of serverless computing Some are as follows:

- **Improved Development Speed:** Serverless architectures accelerate the development process of software [15]. Developers reduce their development cycles substantially because they no longer need to focus on infrastructure management.
- **Scalability:** Sustainability is an intrinsic feature of serverless computing systems. Cloud service suppliers
- Perform automatic scaling of applications through demand-based operations that operate without human involvement.
- **Agility:** Microservices independence enables organizations to speed up software delivery while making better changes to meet evolving business needs. Portability Through containerization, applications function identically in different environments, which decreases deployment problems while improving their portability factor.
- **Cost Efficiency:** Serverless computing offers businesses a cost-efficient solution among its many attractive features [16].

C. *Challenges of Serverless Computing*

Numerous obstacles confront serverless computing systems with the following being among them:

- **Software engineering:** The biggest disadvantage of serverless technology involves [17] software engineering difficulties that negatively affect developer workflows.
- **Data Management:** The management of distributed service data consistency proves difficult which demands thorough design together with coordination.
- **Security:** Many security risks emerge from extensive service and API deployment which requires powerful authentication strategies and authorization protocols and continuous monitoring procedures.
- **Resource Overhead:** While containers are lightweight, the overhead of running numerous instances can accumulate, impacting resource utilization and costs.

- **System (operational) challenges:** System Challenges emerge because cloud services are very volatile [18], necessitating improvements in cloud function lifecycle, the leadership team, cost consistency, and security.

III. Fundamentals of Cloud Native Applications (CNA)

Cloud-Native Applications (CNA) are scalable, containerized, and microservices-based software designed for efficient deployment and operation in cloud environments [19]. Durability and robustness are two of CNA's key characteristics. A CNA must be able to use the cloud's on-demand independent service, quick stretch, and measurable service features, as well as modify the amount of space available by adding or deleting funds, in order to achieve sustainability[20]. A CNA must be able to withstand the failure of virtualized materials, inexpensive hardware [21], or cloud services in order to be resilient. CNAs are organized into core and supplemental functions in order to accomplish both goals.

A. Key Features of Cloud-Native Applications

Cloud-Native Containerization, variable accompaniment, architectural design of microservices, ongoing deployment, and delivery are characteristics of proposals (CI/CD), scalability, resilience, and close alignment with DevOps practices for rapid and reliable software delivery in cloud environments [22]. The key features of Cloud-Native Applications are:

- **Performance:** The public cloud's built-in capabilities, which may outperform their non-native counterparts in terms of performance.
- **Efficiency:** Cloud-native apps should make better use of underlying resources by using cloud-native capabilities and APIs. That results in either reduced operational expenses or improved performance [23].
- **Scalability:** The native cloud APIs provide you immediate access to the platform's autoscaling and load-balancing capabilities when you develop your apps [24].

B. Best Practices for Designing Scalable and Resilient Cloud-Native Applications

These are some guidelines for creating cloud-native applications such as

- Automatic software integration is a key component of continuous integration, which typically entails constructing and testing modified source code at regular intervals. The term "frequency" refers to the regularity with which software is developed and tested; for instance, with each version control commit [25].
- Software must be constantly changeable and deployable, and Continuous Integration is a part of Continuous Delivery. The majority of the time, this need is met by implementing an automated staging environment.
- Continuous Deployment guarantees that software gets put into circulation right away as soon as it is committed to the version control system branches specific to production environments and passes the automated tests to be ready for production [10].
- Deployment and Continuous Delivery are often used interchangeably and might be confused, but it's crucial to distinguish between the terms in academic settings [26][27].

C. Benefits and Challenges of Cloud-Native Architectures

There are some benefits and challenges in CNA that are as follows [28]:

1) Benefits of CAN

There are several benefits of CAN some of them are as follows:

- Reliability enhancements. Improved observability is one of the key advantages, where the service mesh provides detailed insights into the behavior and performance of microservices.
- The decentralized nature of microservices enhances fault isolation, ensuring that the system as a whole is unaffected when a particular service malfunctions.
- Microservices may be independently developed and deployed, allowing for quicker release cycles and more immediate business change adaptability requirements.
- Its broken-down services microservices increase fault separation to stop problems from spreading throughout the system [29]

2) Challenges of CAN

However, integrating a service mesh into existing systems is not without challenges.

- Managing a multitude of microservices introduces operational complexity, requiring sophisticated orchestration and monitoring tools.
- While containers are lightweight, the overhead of running numerous instances can accumulate impacting resource utilization and costs [30].
- Many services and APIs on the system create more security points that need thorough password validation and constant monitoring.
- The operational overhead associated with running a service mesh is also notable, as it can impact system performance and resource usage if not properly configured [31].

IV. AWS, DevOps And Kubernetes For Serverless Computing

AWS, DevOps, and Kubernetes together streamline serverless computing by automating deployment, scaling, and infrastructure management [32][33]. This trio enables rapid development, efficient resource use, and high scalability for modern cloud applications.

A. AWS in Cloud Computing

AWS is a form of cloud computation infrastructure that offers a range of goods. Compute, Distribution, databases, Statistics, Communication, Mobile, Development Tools, the leadership team Tools [34], Internet of Things, and Security are some of these goods. It is not necessary to know how to code to use AWS. Without coding, many basic tasks cannot be completed. For novices who want to launch a career in technological fields, AWS is an excellent option [35]. AWS offers a range of computer services that are suited to different requirements, such as Amazon ECS for coordinated containers, AWS Lambda for serverless computation that is

triggered by events, and Amazon EC2 for expandable virtual servers on demand. AWS offers a range of storage options[36], including Amazon Glacier for affordable, secure cloud storage intended for data preservation and long-term assistance, Amazon S3 for scalability object storage, and Amazon EBS for storage volumes at block level used with EC2 instances [37].

1) AWS Fargate Serverless Containers

AWS Faregate gives developers a way to operate and manage Docker containers by letting them work with Docker instead of dealing with underlying server details [38]. AWS Faregate makes administrators focus solely on developing applications by handling all server management details for them. Geospatial data processing systems now use AWS Faregate to run container tasks when required [39].

2) AWS Step Functions and Workflow Automation

AWS Step Functions enable workflow automation by coordinating manifold AWS facilities into Workflows without servers with visual monitoring and error handling.

- AWS Step Functions provides a container less service to control the interaction between various AWS services in workflow applications [40]. Developers can create and execute workflows that integrate various services including AWS Lambda and AWS Faregate through this system.
- Workflow trace profiles and execution times in current workflow management systems. The research demonstrates how to measure service-based system performance through trace profiling that uses verified metrics to analyze workflow time data [41].

B. Kubernetes for Serverless Computing

Automating the creation, operation, scaling, and management of containerized programs, Kubernetes, often known as K8s, is a platform built on open source [42]. It offers an operating system that is both portable and expandable, making it possible to deploy and preserve submissions across a cluster of computers in an effective manner. A pod serves as the essential deployment unit in the framework of virtualization using containers, particularly within the framework of the Kubernetes environment [43]. Kubernetes plays a crucial role in enabling and enhancing serverless computing by providing scalable, flexible, and efficient container orchestration. Below are key aspects of how Kubernetes contributes to serverless computing:

- **Serverless Frameworks:** Supports Native, OpenVAS, and Kebeles for deploying serverless applications [44].
- **Autoscaling:** Uses Horizontal Pod AutoAlert (HPA) and KEDA for dynamic function scaling.
- **Cost Efficiency:** Enables a pay-per-use model by allocating resources only when needed.
- **Multi-Cloud & Portability:** Avoids vendor lock-in and supports hybrid & multi-cloud deployments [45].
- **Event-Driven Execution:** It integrates with Kafka, NATS, and cloud events for real-time processing.
- **Security & Isolation:** It implements RBAC, network policies, and pod security for secure execution [46].

C. DevOps for Serverless Computing

DevOps plays a crucial role in managing and optimizing serverless computing by automating deployments, enhancing observability, improving security, and ensuring cost efficiency. Software deployment, regular consumption, and adaptability have undergone significant shift in the modern cloud-based systems industry. Because of the cloud's scalability and flexibility, businesses are using it more and more, which has created several possibilities. The shift to cloud-based systems is driven by a number of advantages, including increased accessibility, scalability, and resource efficiency. However, this change poses questions about how to maintain the highest standards of software dependability and quality in this evolving environment [47].

V. Literature Review

In this sector, deliver the aforementioned investigate on Serverless computing and cloud-native applications and trends in AWS, Kubernetes, and DevOps. Table I explores advancements in cloud-native apps and distributed computing, emphasizing AWS, Kubernetes, and DevOps trends. It identifies key benefits, challenges, and future directions in scalability, automation, and edge- cloud integration.

Chen (2021) describes a cutting-edge serverless architecture and extracts a number of important indicators that come together to provide a methodical approach and a benchmark known as the serverless bench. For highly productive applications, serverless computing offers flexibility and cost-effectiveness. Strong isolation within functionality occasions and minimal startup latency to guarantee user experience are two issues that the serverless technology must resolve in order to do this. Lastly, I'll discuss the potential and difficulties of serverless computer systems, covering how to safeguard shared online computing secure and efficient [42]

Alonso et al. (2023) intend to examine current definitions of multi-cloud native apps in order to define the multi-cloud notion from the standpoint of submission expansion. Although plenty of applications have found cloud computing to be an efficient means of obtaining computing power and storage as a customer service, it might not be able to handle the enormous volumes of data generated by network-connected Things appliances and substantially accommodate diverse application needs, including those that multi-cloud programs. These methods must smoothly integrate services and resources throughout the data stream, at the distributed computers' edge and base [10].

Sahana et al. (2023) highlighted the many performance measurements that impact the distribution of loads as well as the current load rebalancing strategies. A competent Elasticsearch cluster is set up on AWS in order to use it in the construction of a weight-based weight balance. Traffic is distributed via an automated canary deployment technique, which assigns weights to the various services. Software deployment, scalability, and the handling of containers may be automated using Kubernetes, a type of container orchestrate system. To divide up incoming connections among the pods, Kubernetes provides a load-balancing service [48].

Luo et al. (2023) it will first evaluate the deployment limits of contemporary cloud-based Serverless workflow scheduling on edges before moving to develop behavior trees for Serverless workflow definition and presenting initial findings about behavior trees for Serverless workflow distribution. Serverless computing has gained popularity yet researchers have not invested sufficient effort to study Serverless workflows and particularly Serverless edge computing[49].

Jiao, Xu and Fan (2021) investigate Kubernetes-based cloud native architecture of applications to address microservice safety and effectiveness monitoring issues. Kubernetes is a micro-service-style application structure and choreography solution for cloud-native platforms. Cloud Nativity has steadily grown to be one of the most important ideas in the software business as the design of applications and cloud computing have advanced. At the same time, the aforementioned functional modules may be greatly reused and the development cost can be decreased based on the second level deploying of the container [50].

Kumar, Gupta and Bhandari (2022) introduce Developers may create and execute applications using serverless design, which is a cloud-native construction model. Although they are not used in the development of applications, servers are still present in serverless. The supply, maintenance, and growth of the server technology are often handled by a cloud provider. To deploy their code, creators just need to package it in vessels. The development of the "cloud-naive calculating prototypical" is another benefit. The phrase "cloud-naive" refers to the potential environmental benefits that internet IT services might provide to the general public [51].

Table I presents a survey of the serverless technology research and cloud-native trends, highlighting key advancements, challenges, and limitations across areas like benchmarking, multi-cloud applications, Kubernetes, and edge workflows, with future directions focusing on performance, scalability, and security improvements

TABLE I. LITERATURE REVIEW ON SERVERLESS COMPUTING AND CLOUD-NATIVE TRENDS

Reference	Focus Area	Key Findings	Challenges	Limitations & Future Work
Chen (2021)	Serverless Benchmarking	Introduces Serverless Bench to evaluate serverless platforms; highlights cost-efficiency and elasticity	Ensuring low latency and strong isolation	Future work in security and performance optimization
Alonso et al. (2023)	Native Applications for Multiple Clouds	Reviews multi-cloud trends for cloud-native app development	Handling heterogeneous requirements and IoT data	Integration of edge, core, and data path services
Sahana et al. (2023)	Kubernetes Load Balancing in AWS	Uses weighted load balancer with canary deployments on Kubernetes	Managing traffic distribution and service health	Needs real-time performance metrics and scalability testing
Luo et al. (2023)	Serverless Edge Workflow Scheduling	Proposes behavior tree model for workflow scheduling on edge	Resource constraints on edge devices	Expand work on Serverless workflows in edge environments
Jiao, Xu and Fan (2021)	CNA Architecture with Kubernetes	Discusses microservices-based cloud-native monitoring & security	Performance and security bottlenecks	Explore module reuse and development cost reduction
Kumar, Gupta and Bhandari (2022)	Serverless Architecture Overview	Describes serverless as a cloud-native development paradigm	Abstracting infrastructure for developers	Investigate environmental benefits and "cloud-naive" potential

VI. CONCLUSION AND FUTURE WORK

The development of software development has been suggestively shaped by the convergence of serverless computing, cloud-native applications, and DevOps practices. These concepts have shaped a complete transformation of application creation and release strategy within current IT system designs. The lack of Professionals needed for the administration of infrastructure focuses only on application logic because of serverless architectures. DevOps approaches facilitate rapid software cycles and improved collaboration between development and operations, whereas Kubernetes-driven cloud-native development with containerization offers scalable operations in addition to resilience and portability characteristics. Developers under serverless computing can devote their time to structure logic instead of managing infrastructure components. By using cloud-native application architecture, containerization, and Kubernetes orchestration, modern computer systems have become more dependable and flexible. DevOps implementation cuts the distance between developers and operators, thereby leading to automation and continuous delivery that produces faster market delivery.

Future studies should concentrate on improving cloud-native cybersecurity and serverless environments through zero-trust models, API protection, and real-time compliance. Integrating AI/ML into DevOps can drive intelligent automation for anomaly detection and pipeline optimization. Emphasis should also be placed on improving workload portability across hybrid/multi-cloud setups, boosting observability with advanced monitoring tools, and optimizing performance and cost. Lastly, simplifying developer workflows with better tooling and low-code/no-code solutions will support faster, scalable, and secure application delivery.

REFERENCE

[1] Z. N. Chen *et al*., "Evolution of Cloud Operating System: From Technology to Ecosystem," *J. Comput. Sci. Technol.*, 2017, doi: 10.1007/s11390-017-1717-z.

[2] G. McGrath and P. R. Brenner, "Serverless Computing: Design, Implementation, and Performance," in *2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, IEEE, Jun. 2017, pp. 405–410. doi: 10.1109/ICDCSW.2017.36.

[3] Y. Hongyu and W. Anming, "Migrating from Monolithic Applications to Cloud Native Applications," in *2023 8th International Conference on Computer and Communication Systems (ICCCS)*, IEEE, Apr. 2023, pp. 775–779. doi: 10.1109/ICCCS57501.2023.10150977.

[4] T. Pradeep Pai and C. R. S. Kumar, "Building cloud native application - Analysis for multi-component application deployment," in *2021 International Conference on Computer Communication and Informatics, ICCCI 2021*, 2021. doi: 10.1109/ICCCI50826.2021.9402422.

[5] M. Thakkar, B. McNamara, M. Brooker, and O. Surkatty,

"Security Overview of AWS Lambda," *Amaz. Web Serv.*, 2019.

[6] S. K. Mondal, R. Pan, H. M. D. Kabir, T. Tian, and H. N. Dai, "Kubernetes in IT administration and serverless computing: An empirical study and research challenges," *J. Supercomput.*, 2022, doi: 10.1007/s11227-021-03982-3.

[7] M. Jiang, I. Nakamoto, W. Zhuang, W. Zhang, Y. Guo, and L. Ma, "Flexible Investment Strategies for Cloud-Native Architecture of Public Health Information Systems," *Wirel. Commun. Mob. Comput.*, 2021, doi: 10.1155/2021/6676428.

[8] Godavari Modalavalasa, "The Role of DevOps in Streamlining Software Delivery: Key Practices for Seamless CI/CD," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 1, no. 12, pp. 258–267, Jan. 2021, doi: 10.48175/IJARSCT-8978C.

[9] A. M. Buttar *et al.*, "Optimization of DevOps Transformation for Cloud-Based Applications," *Electronics*, vol. 12, no. 2, p. 357, Jan. 2023, doi: 10.3390/electronics12020357.

[10] B. Boddu, S. Neeli, and S. Synam, "Cloud Migration DBA Strategies for Mission-Critical Business Applications," *Int. J. Intell. Syst. Appl. Eng.*, vol. 11, no. 105, Jul. 2023, doi: 10.1186/s13677-021-00253-7.

[11] A. M. Alwakeel, "An overview of fog computing and edge computing security and privacy issues," *Sensors*, 2021, doi: 10.3390/s21248226.

[12] J. L. Deepak Dasaratha Rao, Sairam Madasu, Srinivasa Rao Gunturu, Ceres D'britto, "Cybersecurity Threat Detection Using Machine Learning in Cloud-Based Environments: A Comprehensive Study," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 12, no. 1, 2024.

[13] H. Inam, N. U. Islam, M. U. Akram, and F. Ullah, "Smart and Automated Infrastructure Management: A Deep Learning Approach for Crack Detection in Bridge Images," *Sustain.*, 2023, doi: 10.3390/su15031866.

[14] J. M. O. Candel, A. Elouali, F. J. M. Gimeno, and H. Mora, "Cloud vs Serverless Computing: A Security Point of View," in *Lecture Notes in Networks and Systems*, 2023. doi: 10.1007/978-3-031-21333-5_109.

[15] K. M. R. Seetharaman, "Internet of Things (IoT) Applications in SAP: A Survey of Trends, Challenges, and Opportunities," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 3, no. 2, 2021, doi: DOI: 10.48175/IJARSCT-6268B.

[16] A. Eivy, "Be Wary of the Economics of 'Serverless' Cloud Computing," *IEEE Cloud Comput.*, 2017, doi: 10.1109/MCC.2017.32.

[17] O. Cico, L. Jaccheri, A. Nguyen-Duc, and H. Zhang, "Exploring the intersection between software industry and Software Engineering education - A systematic mapping of Software Engineering Trends," *J. Syst. Softw.*, 2021, doi: 10.1016/j.jss.2020.110736.

[18] Y. Li, Y. Lin, Y. Wang, K. Ye, and C. Xu, "Serverless Computing: State-of-the-Art, Challenges and Opportunities," *IEEE Trans. Serv. Comput.*, 2023, doi: 10.1109/TSC.2022.3166553.

[19] N. P. Hirenkumar Mistry Kumar Shukla, "Securing the Cloud: Strategies and Innovations in Network Security for Modern Computing Environments," *Int. Res. J. Eng. Technol.*, vol. 11, no. 04, p. 11, 2024.

[20] M. S. Samarth Shah, "Deep Reinforcement Learning For Scalable Task Scheduling In Serverless Computing," *Int. Res. J. Mod. Eng. Technol. Sci.*, vol. 3, no. 12, pp. 1845–1852, 2021, doi: DOI : https://www.doi.org/10.56726/IRJMETS17782.

[21] S. Brunner, M. Blochlinger, G. Toffetti, J. Spillner, and T. M. Bohnert, "Experimental Evaluation of the Cloud-Native Application Design," *Proc. - 2015 IEEE/ACM 8th Int. Conf. Util. Cloud Comput. UCC 2015*, pp. 488–493, 2015, doi: 10.1109/UCC.2015.87.

[22] S. Arora and S. R. Thota, "Automated Data Quality Assessment And Enhancement For Saas Based Data Applications," *J. Emerg. Technol. Innov. Res.*, vol. 11, pp. i207–i218, 2024, doi: 10.6084/m9.jetir.JETIR2406822.

[23] S. R. Thota, S. Arora, and S. Gupta, "Hybrid Machine Learning Models for Predictive Maintenance in Cloud-Based Infrastructure for SaaS Applications," in *2024 International Conference on Data Science and Network Security (ICDSNS)*, IEEE, Jul. 2024, pp. 1–6. doi: 10.1109/ICDSNS62112.2024.10691295.

[24] T. Theodoropoulos *et al.*, "Security in Cloud-Native Services: A Survey," *Int. J. ofINTELLIGENT Syst. Appl. Eng.*, vol. 3, no. 4, pp. 758–793, Oct. 2023, doi: 10.3390/jcp3040034.

[25] P. M. Rajendra Prasad Sola, Nihar Malali, "Cloud Database Security: Integrating Deep Learning and Machine Learning for Threat Detection and Prevention: 0," *Notion Press*, 2025.

[26] O. C. Oyeniran, O. T. Modupe, A. A. Otitoola, O. O. Abiona, A. O. Adewusi, and O. J. Oladapo, "A Comprehensive Review of Leveraging Cloud-Native Technologies for Scalability and Resilience in Software Development," *Int. J. Sci. Res. Arch.*, vol. 11, no. 2, pp. 330–337, Mar. 2024, doi: 10.30574/ijsra.2024.11.2.0432.

[27] S. Arora, S. R. Thota, and S. Gupta, "Artificial Intelligence-Driven Big Data Analytics for Business Intelligence in SaaS Products," in *2024 First International Conference on Pioneering Developments in Computer Science & Digital Technologies (IC2SDT)*, IEEE, Aug. 2024, pp. 164–169. doi: 10.1109/IC2SDT62152.2024.10696409.

[28] S. Murri, "Data Security Environments Challenges and Solutions in Big Data," *Int. J. Curr. Eng. Technol.*, vol. 12, no. 6, pp. 565–574, 2022.

[29] N. Patel, "AI-Enhanced Zero Trust Security Architecture for Hybrid and Multi-Cloud Data Centers: Automating Trust Validation, Threat Detection, and Mitigation," *Int. J. Nov. Trends Innov.*, vol. 3, no. 1, pp. a13–a18, 2025.

[30] S. krishnan U. Advait Patel, Hariharan Ragothaman, Milavkumar Shah, Adit Sheth, Charit Upadhyay, Pravin Pandey, "AI-Powered Data Encryption Techniques: Safeguarding Cloud Infrastructure," *IEEE*, 2025.

[31] D. Bruzual, M. L. Montoya Freire, and M. Di Francesco, "Automated Assessment of Android Exercises with Cloud-native Technologies," in *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE*, 2020. doi: 10.1145/3341525.3387430.

[32] V. Prajapati, "Cloud-Based Database Management: Architecture, Security, challenges and solutions," *J. Glob. Res. Electron. Commun.*, vol. 01, no. 1, pp. 07–13, 2025, doi: https://doi.org/10.5281/zenodo.14934833.

[33] A. and P. Khare, "Cloud Security Challenges : Implementing Best Practices for Secure SaaS Application Development," *Int. J. Curr. Eng. Technol.*, vol. 11, no. 6, pp. 669–676, 2021, doi: https://doi.org/10.14741/ijcet/v.11.6.11.

[34] Abhishek Saini, Chaman Sharma, Nadeem Khan, Rohit Chauchan, and Gurjeet Singh, "A REVIEW PAPER ON AWS," *EPRA Int. J. Multidiscip. Res.*, 2024, doi: 10.36713/epra15444.

[35] M. S. S Shah, "Kubernetes in the Cloud: A Guide to Observability," *DZone*, 2025.

[36] S. Mishra, M. Kumar, N. Singh, and S. Dwivedi, "A Survey on AWS Cloud Computing Security Challenges & Solutions," in *Proceedings - 2022 6th International Conference on Intelligent Computing and Control Systems, ICICCS 2022*, 2022. doi: 10.1109/ICICCS53718.2022.9788254.

[37] K. Rajchandar, M. Ramesh, A. Tyagi, S. Prabhu, D. S. Babu, and A. Roniboss, "Edge Computing in Network-based Systems: Enhancing Latency-Sensitive Applications," in *2024 7th International Conference on Contemporary Computing and Informatics (IC3I)*, 2024, pp. 462–467. doi: 10.1109/IC3I61595.2024.10828607.

[38] K. Burkat *et al.*, "Serverless containers - Rising viable approach to scientific workflows," in *Proceedings - IEEE 17th International Conference on eScience, eScience 2021*, 2021. doi: 10.1109/eScience51609.2021.00014.

[39] S. S. S. Neeli, "Leveraging Docker and Kubernetes for Enhanced Database Management," *J. Artif. Intell. Mach. Learn. Data Sci.*,

vol. 1, no. 1, p. 5, 2022.

[40] J. Thomas, "The Effect and Challenges of the Internet of Things (IoT) on the Management of Supply Chains," *Int. J. Res. Anal. Rev.*, vol. 8, no. 3, pp. 874–878, 2021.

[41] N. Shahidi, J. R. Gunasekaran, and M. T. Kandemir, "Cross-Platform Performance Evaluation of Stateful Serverless Workflows," in *Proceedings - 2021 IEEE International Symposium on Workload Characterization, IISWC 2021*, 2021. doi: 10.1109/IISWC53511.2021.00017.

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

[43] A. Poniszewska-Marańda and E. Czechowska, "Kubernetes cluster for automating software production environment," *Sensors*, 2021, doi: 10.3390/s21051910.

[44] K. Kritikos and P. Skrzypek, "A review of serverless frameworks," in *Proceedings - 11th IEEE/ACM International Conference on Utility and Cloud Computing Companion, UCC Companion 2018*, 2018. doi: 10.1109/UCC-Companion.2018.00051.

[45] O. Tomarchio, D. Calcaterra, and G. Di Modica, "Cloud resource orchestration in the multi-cloud landscape: a systematic review of existing frameworks," *J. Cloud Comput.*, 2020, doi: 10.1186/s13677-020-00194-7.

[46] S. Risco, C. Alarcón, S. Langarita, M. Caballer, and G. Moltó, "Rescheduling serverless workloads across the cloud-to-edge continuum," *Futur. Gener. Comput. Syst.*, 2024, doi: 10.1016/j.future.2023.12.015.

[47] A. Mishra and Z. Otaiwi, "DevOps and software quality: A systematic mapping," *Computer Science Review*. 2020. doi: 10.1016/j.cosrev.2020.100308.

[48] B. Sahana, T. Kumaraswamy, R. G. Nachiketh, S. Navadeep, and J. Noronha, "Weight based Load Balancing in Kubernetes using AWS," in *IDCIoT 2023 - International Conference on Intelligent Data Communication Technologies and Internet of Things, Proceedings*, 2023. doi: 10.1109/IDCIoT56793.2023.10053466.

[49] K. Luo, T. Ouyang, Z. Zhou, and X. Chen, "Behavior Tree-based Workflow Modeling and Scheduling for Serverless Edge Computing," in *Proceedings - International Conference on Distributed Computing Systems*, 2023. doi: 10.1109/ICDCS57875.2023.00100.

[50] Q. Jiao, B. Xu, and Y. Fan, "Design of Cloud Native Application Architecture Based on Kubernetes," in *2021 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, IEEE, Oct. 2021, pp. 494–499. doi: 10.1109/DASC-PICom-CBDCom-CyberSciTech52372.2021.00088.

[51] A. Kumar, R. Gupta, and R. Bhandari, "WoS Bibliometric-based Review on Serverless Computing model," in *PDGC 2022 - 2022 7th International Conference on Parallel, Distributed and Grid Computing*, 2022. doi: 10.1109/PDGC56933.2022.10053142.

