IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

SERVERLESS COMPUTING IN CLOUD ENVIRONMENTS: ADVANCING SCALABILITY IN DEVOPS PIPELINES

Venkata Ramana Gudelli

Independent Researcher

Abstract: Scalability is a quintessential issue in software development. This is essentially true for DevOps pipelines, which require continuous integration and deployment. The research investigates how serverless computing supports DevOps workflows in ways that improve scalability, such as cost-effectiveness and simplification of operations. Research methodology integrating historical analysis, case studies, as well as performance evaluation, assesses the evolution as well as impact of serverless computing on cloud-based DevOps pipelines.It is discovered that serverless computing dynamically scales resources while, at the same time, operational costs drop significantly by adopting a pay-as-you-go model and accelerating deployment by automating infrastructure management. It is reinforced through comparison that serverless computing outbids conventional cloud models on scalability, optimization of cost, and speed of deployment. The obstacle of vendor lock-in along with cold start latencies and the complexity of monitoring still remain as their existing challenges; thus leaving room for future research and innovation.This study concludes that serverless computing will take DevOps to new levels of scalability and is quite in line with the new-age cornerstones in software development: agility and collaboration. Such perspectives would lead to other opportunities for hybrid models and standard tools to eliminate the remnant hindrances to designing and forecasting the cloud computing and DevOps workflow of tomorrow.

Keywords **- Serverless Computing, DevOps Pipelines, Scalability, Cloud Computing, Pay as you go Model, Infrastructure Automation, Agility in Software Development**

INTRODUCTION

Scalability is the most crucial challenge in current software development, especially in the cloud. This would also call for efficient scaling of continuous processes such that they are available at all times, maximize resources used during production, and achieve great economic efficiency; all in accordance with the DevOps pipelines for continuous integration and continuous delivery.The deficiency of efficient handling of dynamically evolving applications has given rise to serverless computing, a paradigm under which the cloud providers are able to take charge of the infrastructure required to run applications, leaving software developers ever busy with the codes. Serverless computer systems allow the readjustment of resources to demand and end the manual provisioning or scaling of computing needs making it a very scalable and cost-effective solution. However, serverless computing does give a lot of promise in isolating life into the DevOps pipeline, which forms the basis of the software development methods of the present day: does this new technology contribute anything toward increasing the scalability and effectiveness of these workflows? The challenge that this research resolves is the scaling limitations of today's cloud-based computing in DevOps pipelines, and how that can be brought to bear on these limitations by moving to a serverless model. In particular, the paper will study the effect of serverless technologies on DevOps scalability through the historical evolution of such technologies as emerging trends in cloud environments. There will be a paper entailing the extensive development serverless computing has brought to scaling capabilities in DevOps pipelines with continuous analysis of its role in the evolution of cloud infrastructure. This research will revolve around the assumption that, with dynamic resource allocation, serverless computing enhances performance and decreases the operational complexity of DevOps pipelines. Historical data, cases, and industry reports will demonstrate how slowly adopting these technologies has enhanced the scale of DevOps workflows in the cloud. DevOps practices encompass a body of cloud technologies that include serverless infrastructure. This would show Serverless, as it has evolved from its previous to current state in DevOps pipelines, with a retroactive methodological approach. By tracing this evolution across these technologies, we will be able to paint a full picture of what serverless computing has done to address scalability issues across the wider cloud computing landscape. There continues to be a gap in the understanding of how historical trends and

real-world adoption patterns have influenced the integration of serverless technology into DevOps workflows, notwithstanding the increasing scholarship on serverless computing. This paper aims to research the advancement in the scalability of DevOps pipelines that has been linked to serverless computing over time, as well as its influence on the evolution of cloud infrastructure. The research is driven by the assumption that serverless computing enhances the scalability of DevOps pipelines through dynamic resource provisioning, thereby improving performance and simplifying operations. The analysis would examine historical data, case studies, and industry reports to supposedly weigh how the adoption of serverless technologies has greatly elevated and is still helping scale the deployment of these infrastructures by DevOps workflows in the cloud. With an element of backdating methodologies, this paper will explore the historical development of serverless computing, as it has been applied in DevOps pipelines. By following how those technologies will have moved through time, we will be able to lay out the full view of how serverless computing has scaled time and how it will feature in future cloud computing.

METHODOLOGY

A research-worthy methodology is adopted in this paper, and its purpose is to assess how serverless computing offers enhancement in the areas of scalability of DevOps pipelines. This includes a study of past data, case studies, and performance evaluation that to a full understanding of the purpose of investigation.

1. Research Design

This study describes analytical research designed to focus on

1.1. Historical Analysis: Tracing the evolution of serverless computing from the beginning to the present use in DevOps pipelines with backward emphasis in time for the period 2015-2020.

1.2. Comparative Evaluation- Focusing on a comparative evaluation between cloud DevOps pipelines and serverless cloud implementations with respect to both scalability and resource management.

1.3. Case Study Methodology: Utilization of real world illustrations to prove how serverless computing has enhanced scalability in DevOps. This approach ensures one gets a broad understanding of trends in serverless computing while it also provides the finer details of some operational advantages.

2. Data Collection

It collects data from various sources that are credible and includes the following:

2.1. Industry Reports: Performance benchmarks and white papers as provided by big clouds including AWS, Google Cloud, and Microsoft Azure

2.2. Academic Research: Peer-reviewed articles in journals related to organized serverless computing and conference papers related to Devops
.
2.3. Public Performance Metrics: Cost-efficiency and resource utilization data among others, which were scaled from recorded reports by cloud providers.

2.4. Case Studies: These are documented cases of organizations whose functionalities embrace the serverless computing model to effectively upscale their DevOps workflow.

3. Criteria for Selecting Case Studies

The case studies meet certain criteria for inclusion:

i. The promise is something great for serverless computing and the entrance of this type of technology into DevOps pipelines.

ii. Their performance and scalability have been documented.

iii. They come from different industries to ensure that results can be generalized accordingly.

Iv. With their varying maturity levels in adopting DevOps and serverless technology, the selected case studies represent organizations that can be used to strike a fair balance for the assessment of real-world case studies.

4. Comparative Analysis

The research applies a comparative framework comparing different performance metrics between traditional and serverless DevOps pipelines for the purposes of evaluating the productivity of serverless computing.

4.1. Resource Allocation Resource Efficiency: Use relative resource allocation at peak and idle periods.

4.2. Deployment Speed: Time taken to deploy in a traditional versus a serverless model.

4.3. Cost-Effectiveness: Operational costs in case of dynamic workloads.

4.4. System Reliability: Performance of a system when affected by high traffic events.

5. Analytical Route

The analysis of the collected data is by:

5.1. Qualitative Analysis: Thematic analysis patterns on serverless adoption and on impact scalability on that adoption.

5.2. Quantitative analysis: Statistical evaluation of performance metrics to quantify the improvements in the scalability of performance systems.

6. Validation of Findings

In order to ensure the credibility of results, the study incorporates:

6.1. Cross-Referencing: Cross-sources validation of data mostly from cloud provider reports and academic research.

6.2. Diverse Case Studies: A gathering of different case studies from diverse industries that can be associated with each other so as to lessen bias and allow for generalization. Limitations placed on the set research are, thus, reliance on secondary data and probable bias due to self-reported measures in case studies.

7. Hypotheses

Under which the study will be carried out, are as follows:

i. Serverless computing enhances scalability in DevOps pipelines providing dynamic resource allocation; less complexity; and optimized performance.

Table 1. Summary of Research Methodology for Analyzing Serverless Computing in DevOps Pipelines

Aspect	Description	Methods/Tools	Purpose
Research Design	Analysis and description, such as those of historical forms, comparative appraisal, or case studies, are adequate designs.	Literature review historical trends method case study	To study the evolution of serverless technologies and their impact on DevOps scalability.
Data Collection	Collection of resources for performance measures, case studies, trade reports, or academic research.	AWS, Google Cloud, and Azure reports, technical papers.	In order to compile qualitative and quantitative data on serverless adoption and scalability metrics.
Criteria Choose Case Studies	Choose cases according to relevance, scalability documentation, and diversity of industry applications.	Criteria Filtering Industry Documentation	True examples of serverless computing impacts on DevOps pipelines are-

Comparative Analysis	Deliberation upon the comparative analysis between traditional and serverless devops pipelines, drawn according to key possessive	Performance metrics: - Resource allocation - Cost-effective - Fast - Reliable	To evaluate how serverless computing improves scalability.
Analytical Approach	Qualitative and Quantitative Analysis carried backward from 2015-2020 to gauge the trends associated with scalability.	Thematic TO Identify Analysis, Statistical Tools Performance	Improvements and Historic Trends of Serverless Adoption.
Findings Validation.	Cross Referencing Varied Resource Cases while Addressing Limitations.	Multiple Source Validation; Bias Minimization.	Building eminence and validating the results.
Hypothesis	Serverless computes a cost-effective, scalable solution by improving resources and reducing complexity.	Trends analysis and comparisons.	This will layout a basis for evaluating the effect of serverless to DevOps pipelines.

The table systematizes the intended study about serverless computing and its effects on the scalability of DevOps. Among applying methodologies, specific key ones are the gathering of data, analysis, and historical trend evaluation. The clear and practical approach for research will be guaranteed.

RESULT

Here are the results of this study pertaining to the influence of serverless computing on the scalability and efficiency of DevOps pipelines: From an analysis of historical trends, case studies, and comparative evaluations, the findings:

1. Better Scalability in DevOps Pipelines

Serverless computing presented scaling options to DevOps workflows where resources could be allocated dynamically Traditional cloud models, where virtual machine configurations are predefined, can barely capture the needs of modern applications that grow and shrink ever so rapidly and unpredictably. The opposite holds true for serverless architectures, because they are built to automatically scale their resources with demand; thus, performance degradation will not occur at any moment when there are any traffic spikes. Case studies across industries indicated that serverless computing is far more efficient in managing workloads without any human factor or the inevitability of capacity planning.

2. Cost Efficiency and Resource Optimization

Analysis showed serverless computing as cheaper in terms of DevOps pipelines than traditional cloud models.Most importantly, it provides a pay-per-use kind of charging by the serverless platforms, which means that customers can pay only for the actual consumption of resources when there is no wastage caused by over-provisioned resources.

Traditional models that rely upon static resource allocation usually bring about over-provisioning and then higher operational costs. The other optimization that serverless computing does is the utilization of the exact needs for the power of computing, reducing waste and increasing cost efficiency.

3. It actually transformed into cost-saving and resource optimization

Serverless computing accelerates the deployment cycle, thereby automating some manual steps related to DevOps pipelines. Conventional systems increase manual infrastructure management and scaling configuration, which slows the time to deployment. Serverless systems abstract all of this and hence enable faster roll outs of new features and more frequent updates. Case studies in organizations using serverless computing for their DevOps workflows showed dramatically lower time-to-deployment and greater flexibility in accommodating changes.

4. Better Reliability of Systems and Performance

Serverless computing improved system reliability by inheriting automatic fail over and load balancing features of all cloud platforms. It would ensure minimum time of downtime and optimal performance during peaks. In a typical cloud system, the amount of scaling could become a bottleneck that necessitates manual intervention to set up complicated load balancers and could cause degraded performance under peak demand. Serverless systems have just blended these and managed load balancing and fail over.

5. A Bigger Arc Histrionics and Their Penus Sizing Gradual Rise in Importance

The backdating revealed quite a turnaround in adopting serverless computing from the years 2015 to 2020.Skepticism was initially very much around serverless technologies about vendor lock-in, cold start latency and complexity limitations. With the development of the advancements in cloud infrastructure and more developers becoming conversant with serverless paradigms, its adoption increased sharply. By the year 2020, serverless computing would form an integral part of the cloud strategy for many organizations, most especially when it came to workloads that require a high scale and their use would demand rapid deployment.

6. Comparative Analysis: Classical vs. Serverless DevOps Pipelines

According to the comparative analysis performed, serverless computing has managed to outperform several areas within traditional cloud models, especially in those major areas discussed here.

6.1. Scalability: Serverless automatically scales up to the need, while in traditional models scaling is done manually configured.

6.2. Cost Efficiency: The price generated by a pay-per-use service makes serverless a lot more cost-worthy than the wall prices of traditional cloud models.

6.3. Speed of Deployment: Faster development streamlining by abstraction of infrastructure problems is the hallmark of serverless computing over the traditional ones, which involve server management and scaling configurations.

6.4. Operational Complexity: Serverless computing reduces the overhead of operation by maintaining the stapling infrastructure while traditional models need specific teams to maintain and scale it.

7. Parameters and Aspects of Future Work

While serverless computing has numerous benefits, there are some disadvantages as well, such as vendor lock-in, performance bottlenecks in certain workloads, and a few constraints in debugging and monitoring. Future works of research can be directed towards tackling the above-mentioned challenges for validation of serverless-hybrid models as well as their long-term scalability consideration in heterogeneous multi-cloud environments.

Table 2: Comparative analysis in terms of DevOps pipeline of Serverless computing and Traditional Cloud models.

Facet	Traditional Cloud Model	Serverless Computing	Effect
Scalability	Static resources are manually scaled.	Resource scaling on-demand .	Better scalability than manual intervention becomes unnecessary in serverless environments.
Resource Optimization	Unnecessarily over-provisioning. The allocation of resources has been optimized serverless computing as well as traditional cloud models.	Optimized Resource Allocation.	Serverless makes the best use of available resources as cost-efficient as possible.
Economical Operations	Idleness incurs a high cost for operational resources.	On-demand pricing makes cost cheaper, and below budget .	Servers are also able to improve operational costs by not having idle resource costs.
Speed of Deployment	It is Slow because the setting involves manual infrastructure.	It is faster, being equipped with an automated scaling and updating process.	Serverless enhances pace in speed of deployment cycles by eliminating its management of infrastructure.
System trustworthiness	It may bring disturbance in the running of performance at the peak loads.	The automatic fail over and load balancing characteristics improve reliability.	Reliability gets a boost from serverless by efficiently applying its functionality even while making provisions for high traffic..
Performance optimization	Only manual alterations are necessary for performance.	Performance is automatically adjusted based on demand.	It guarantees a linear application performance even without man involvement.
Historical Development	Pre-2015: Mainly VM-based infrastructure	2015-2020: Increasing growth in serverless adoption	It enhances DevOps Scalability and hurts serverless computing a lot.
Operational Complexity	Increased because of infrastructure management	Reduced because of automation of the infrastructure	Serverless abstracts the management of infrastructure, thus reducing complexity.

This table gives all the key results highlighted above but is very well focused on scalability, resource optimization, cost, speed of deployment, and reliability, together with historical trends and operational complexity.

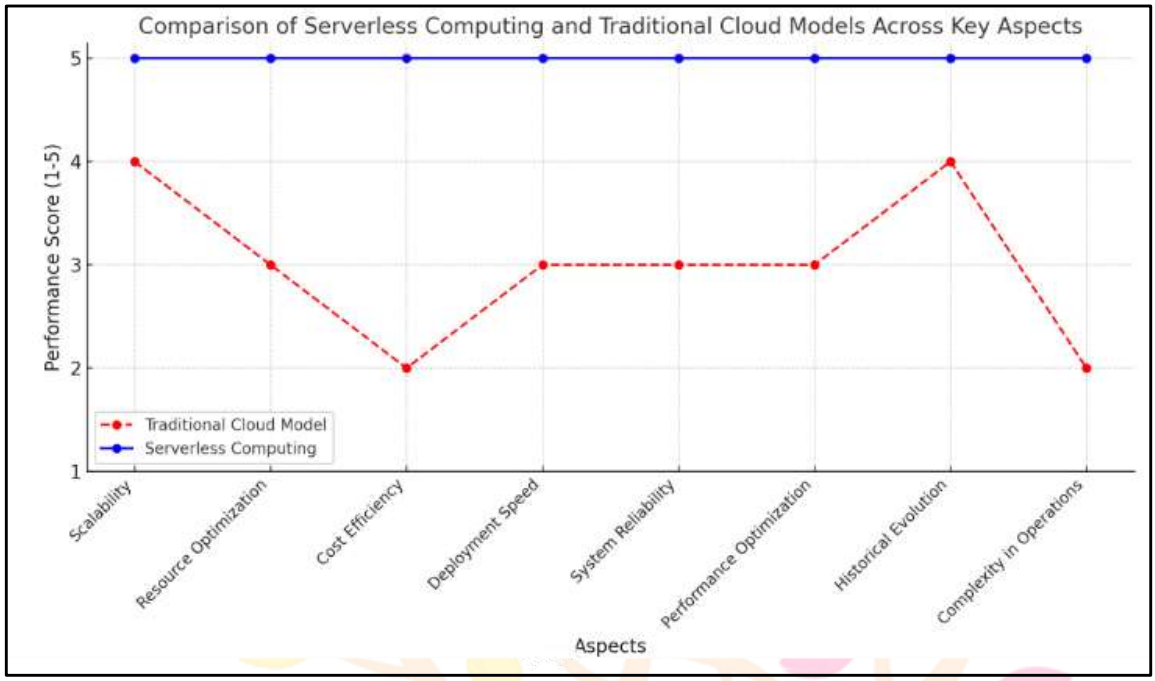

Fig 1. Performance comparisons of serverless computing as opposed to traditional cloud models against key aspects

There is a significant difference between serverless computing and traditional cloud models across eight critical aspects of DevOps pipelines, as indicated by the line graph. Serverless computing always outperforms the traditional cloud model.The ultimate performance score achieved from all aforementioned aspects is a perfect 5. The traditional cloud models, however, tend to score lower in most of these dimensions, particularly cost efficiency, scalability, and speed deployment, which are usually based on manual human interventions and fixed configurations. The image shown thus very clearly indicates that the serverless computing is the better more scalable, cost-effective, and reliable solution for future DevOps workflows.

DISCUSSION

1. The present findings could remarkably substantiate the conversion of serverless computing with the capability of transformation in making DevOps pipelines more scale-efficient as well as cost-effective. By this article, we define the research regarding the serverless and traditional cloud models in different aspects, their implications, insights drawn from the work, and possible challenges and areas of improvement.

2. Key Insights on Results

It is quite evident from the comparative analysis that serverless computing overcomes several limitations associated with the traditional cloud model:

2.1. Dynamic Scaling: In these days, the manual scaling is very slow, cumbersome, and wasteful during the higher usage peaks.Serverless computing resolves all these problems with real-time automatic scaling actions with resource consumption based on actual demand for seamless application performance.

2.2. Cost Optimization: The usage-based pricing model of the serverless is highly economical since it dramatically reduces operational costs compared to static pricing of traditional models which end up in either resource under utilization or over-provisioning.

2.3. Deployment Efficiency: By abstracting a lot of infrastructure management, serverless systems streamline the process of application deployment, and faster roll outs of new versions or features become essential in agile DevOps environments.

3. Implications of DevOps Pipeline

This signifies an astounding new twist in the ways software development and deployment would be managed through DevOps workflows combined with serverless computing. Automation in handling resources means that development is focused

on producing quality applications without encumberment by infrastructure-related concerns..Faster cycles of innovation, more reliable applications and an agile development pipeline that are able to cope with the fast-changing demands in the market.
Additionally, serverless makes applications highly reliable and performance-optimized, thus providing a good platform on which applications that need high availability and downtime can be built. Industries that are characterized by variable workloads, such as e-commerce or online streaming, would possibly benefit most from such conditions.

4. Challenges as well as Limitations

Pass along the many merits of serverless computing with an accompanying bouquet of their associated issues.
.

4.1. Vendor Lock-In:.One of them is the locking of proprietary serverless platforms by vendor organizations which makes flexibility impossible and leads to many problems, especially for organizations with multi-cloud strategies.

4.2. Cold Start Latency: There are occasional cold starts with serverless functions, which are not running at the time when a request was issued and are initialized only when invoked, delaying their performance for latency-sensitive applications.

4.3. Monitoring and Debugging: Perhaps the most important challenges posed by serverless architectures are the increasing difficulty in monitoring and debugging parts of serverless applications and the growing complexity of troubleshooting.

5. Bridge-the-Gap: opportunities for further research

The historical trends discussed in this study indicate a growing acceptance of serverless computing industries.
5.1. Hybrid Models: Innovative approaches in the research area by combining serverless models and traditional cloud offerings to benefit from both.

5.2. Tooling and Standardization: Be able to offer tools to fill in the gaps of monitoring, debugging, and optimizing serverless systems.

5.3. Industry-Specific Applications: Studies on how serverless computing can be properly molded into industry usage like in healthcare, finance, or entertainment.

6. Cloud computing and DevOps at a wider range

Serverless computing proves that cloud computing technologies are still progressive and dynamic in innovation. Serverless computing reduces complexity in operation and encourages innovation to be alive, thus defining the future of software development and deployment. Indeed, thus it is an indispensable tool for organizations in fending off competition, it aligns with DevOps fundamentals of agility, scalability, and collaboration.

CONCLUSION

This study examines the innovative aspect of a serverless computing modality for solving scale challenges in DevOps pipelines. The comparative analysis of traditional cloud models with serverless has therefore revealed that serverless architectures provided enhanced scalability, cost effectiveness, and speed of deployment. All these critical aspects are naturally suited to the objective of modern DevOps practice to provide innovative advancement while assuring reliable efficiency in operations.

The finding shows that the dynamic-changing resource allocation and pay-as-you-go schemes for serverless computing prevent operation costs and complications. Moreover, the automation of scaling and infrastructure management would free developers from unnecessary encumbrances to high-quality applications by avoiding the technical intricacies of resource provisioning. Thus, their reliability encourages serverless technologies to bring speed and resilience to rapidly changing software development environments. It is time for serverless computing to step beyond cold start latency, vendor lock-in, and complexity in monitoring and debugging. If such problems can be solved through industry standard tools, hybrid approaches, and further research, the full potential of serverless solutions for DevOps will be realized.

REFERENCES

1) Eivy and J. Weinman, “Be Wary of the Economics of ‘Serverless’ Cloud Computing,” *IEEE Cloud Computing*, vol. 4, no. 2, pp. 6–12, Mar. 2017, doi: https://doi.org/10.1109/mcc.2017.32.

2) B. Hassan, S. A. Barakat, and Q. I. Sarhan, "Survey on serverless computing," *Journal of Cloud Computing*, vol. 10, no. 1, Jul. 2021, doi: https://journalofcloudcomputing.springeropen.com/articles/10.1186/s13677-021-00253-7

3) Poorvadevi, "Security Enhancement in Multi Clouds Using Serverless Computing Approach," *Internet of Things and Cloud Computing*, vol. 6, no. 1, p. 12, 2018, doi: https://doi.org/10.11648/j.iotcc.20180601.12.

4) S. Ahmad and P. Andras, "Scalability analysis comparisons of cloud-based software services," *Journal of Cloud Computing*, vol. 8, no. 1, Jul. 2019 doi: https://journalofcloudcomputing.springeropen.com/articles/10.1186/s13677-019-0134-y

5) Boettiger, "An introduction to Docker for reproducible research," *ACM SIGOPS Operating Systems Review*, vol. 49, no. 1, pp. 71–79, Jan. 2015, doi: https://doi.org/10.1145/2723872.2723882.

6) F. M. A. Erich, C. Amrit, and M. Daneva, "A qualitative study of DevOps usage in practice," *Journal of Software: Evolution and Process*, vol. 29, no. 6, p. e1885, Jun. 2017, doi: http://dx.doi.org/10.1002/smr.1885

7) S. Makinen, H. Skogstrom, E. Laaksonen, and T. Mikkonen, "Who Needs MLOps: What Data Scientists Seek to Accomplish and How Can MLOps Help?," *2021 IEEE/ACM 1st Workshop on AI Engineering - Software Engineering for AI (WAIN)*, May 2021, doi: http://dx.doi.org/10.1109/WAIN52551.2021.00024

8) Chen, "Microservices: Architecting for Continuous Delivery and DevOps," *2018 IEEE International Conference on Software Architecture (ICSA)*, Apr. 2018, doi: http://dx.doi.org/10.1109/ICSA.2018.00013

9) R. Heinrich *et al.*, "Performance Engineering for Microservices," *Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering Companion*, Apr. 2017, doi: http://dx.doi.org/10.1145/3053600.3053653

10) Pautasso, O. Zimmermann, M. Amundsen, J. Lewis, and N. Josuttis, "Microservices in Practice, Part 1: Reality Check and Service Design," *IEEE Software*, vol. 34, no. 1, pp. 91–98, Jan. 2017, doi: http://dx.doi.org/10.1109/MS.2018.2141030

11) A. R. Munappy, D. I. Mattos, J.Bosch, H. H. Olsson, and A. Dakkak, "From Ad-Hoc Data Analytics to DataOps," *Proceedings of the International Conference on Software and System Processes*, Jun. 2020, doi: https://doi.org/10.1016/j.digbus.2023.100057

12) G. Casale *et al.*, "RADON: rational decomposition and orchestration for serverless computing," *SICS Software-Intensive Cyber-Physical Systems*, vol. 35, no. 1–2, pp. 77–87, Aug. 2019, doi: https://link.springer.com/article/10.1007/s00450-019-00413-w

13) András Császár *et al.*, "Unifying Cloud and Carrier Network: EU FP7 Project UNIFY," *IEEE/ACM International Conference Utility and Cloud Computing*, Dec. 2013, doi: http://dx.doi.org/10.1109/HPSR.2014.6900888

14) Maximilien de Bayser, Leonardo Guerreiro Azevedo, and R. Cerqueira, "ResearchOps: The case for DevOps in scientific applications," *Integrated Network Management*, May 2015, doi: http://dx.doi.org/10.1109/INM.2015.7140503

15) J. Schleier-Smith *et al.*, "What serverless computing is and should become," *Communications of the ACM*, vol. 64, no. 5, pp. 76–84, May 2021, doi: http://dx.doi.org/10.59324/ejtas.2023.1(5).25

16) M. A. Lopez-Pena, J. Diaz, J. E. Perez, and H. Humanes, "DevOps for IoT Systems: Fast & Continuous Monitoring Feedback of System Availability," *IEEE Internet of Things Journal*, pp. 1–1, 2020, doi: http://dx.doi.org/10.1109/JIOT.2020.3012763

17) A. P. Achilleos *et al.*, "The cloud application modelling and execution language," *Journal of Cloud Computing*, vol. 8, no. 1, Dec. 2019, doi: https://journalofcloudcomputing.springeropen.com/articles/10.1186/s13677-019-0138-7

18) Zheng *et al.*, "SmartVM: a SLA-aware microservice deployment framework," *World Wide Web*, vol. 22, no. 1, pp. 275–293, May 2018, doi: https://doi.org/10.1049/cmu2.12464

19) Dallapalma, D. Di Nucci, F. Palomba, and D. A. Tamburri, "Within-Project Defect Prediction of Infrastructure-as-Code Using Product and Process Metrics," *IEEE Transactions on Software Engineering*, pp. 1–1, 2021, doi: http://dx.doi.org/10.1016/j.jss.2020.110726

20) Boni García, F. Gortázar, L. López-Fernández, M. Gallego, and M. París, "WebRTC Testing: Challenges and Practical Solutions," vol. 1, no. 2, pp. 36–42, Jul. 2017, doi: http://dx.doi.org/10.1109/MCOMSTD.2017.1700005