DOI: https://doi.org/10.46632/jitl/1/1/9

Journal on Innovations in Teaching and Learning
Vol: 1(1), 2022
REST Publisher; ISSN:2583 6188
Website: http: //restpublisher.com/journals/jitl/

Optimizing Cloud Computing Application for Improved DEVOPS Automation Streamlining

***M. Kannan, K. Rajeswari, T. Nandhakumar, V. Roopa**
Mahendra Engineering College, Namakkal, Tamil Nadu, India.
*Corresponding author email: kannan@mahendra.info

Abstract: Robust growth in Software Development (SD) has become crucial for accomplishing organizational goals as well as performing more efficiently in the fiercely competitive IoT infrastructure. DevOps is a rising methodology that allows businesses to present excellent software features by automation, better team communication, as well as improve efficiency in the Software Development Life Cycle (SDLC). Cloud Computing (CC) has revolutionized the IT landscape, offering unprecedented scalability and flexibility. There are stronger ties among DevOps and CC. DevOps is interested in maximizing progress so that client demands may be quickly incorporated into system output, whilst the cloud provides automatic supply and scalability for managing application modifications. IT professionals in CC frequently commit mistakes that their expertise may simply avoid. The issue arises because recommended practices remain unclear. These sectors are extremely new, yet the issue may be more related to individuals than technology, and the obstacle is often more challenging for individuals to solve. DevOps is changing the face of IT operations, and novel features and tools are constantly being introduced by various firms to improve the DevOps pipeline. However, optimizing performance in cloud environments remains a complex challenge. The article also highlights the significance of automation and Infrastructure as a Code (IaC) in maintaining consistent, high-performance cloud environments. Through case studies of large-scale implementations, it demonstrates the practical application of these strategies, revealing significant improvements in system responsiveness, resource utilization, and cost-efficiency.

Keywords: Software development, DevOps, cloud computing, Infrastructure as Code, optimization

1. INTRODUCTION

The current fast-paced SD market, enterprises try to produce high-quality software applications that fulfill user desires and performance necessities, particularly with the advent of cloud technologies [1]. Integrating performance evaluations into DevOps pipelines is an important step toward assuring software performance. DevOps incorporates the phrases "development (dev)" as well as "operations (ops)", is a technique that unifies teams for development and operations. It has developed rapidly over the past few years because of its capacity for enhancing stakeholder communication as well as the delivery of software processes [2]. In contrast, Performance testing is critical for establishing the efficiency in which software applications perform over various conditions. During testing, the IT infrastructure is tested under real-world workloads, as well as its response time, scalability, as well as resource utilization are tracked, as well as any bottlenecks and faults identified. Conducting performance tests within the DevOps pipeline allows organizations to identify and resolve issues with performance prior to the SDLC.

DevOps can perform hundreds of tests in a single day and collect feedback from clients following every delivery. This will allow the company to investigate additional features that could be added in the project while minimizing configuration concerns. DevOps is critical for CC application and the core competencies for CC comprise deployment of automated application, IaaS, and server management. DevOps is a critical component of CC that controls deployment in infrastructure, deployment in application, and application functionality for several settings. Infrastructure providers have considered CC is an economic system of scale, as well as cloud economics is frequently employed to distinguish among private and public clouds [3]. In an environmentally constrained context, DevOps aids in producing an excellent service, ensuring Continuous Delivery (CD), and providing end users with quality software

[4]. DevOps also offers rapid responsiveness to fluctuate client requirements. DevOps enables developers and operators to collaborate in a common environment. DevOps enables the removal of organizational as well as cultural barriers, and minimizing the cost of defect identification in the beginning phases [5]. Aside from the DevOps benefits, problems include transitioning from local infrastructure to microservices, mixing multiple domain resources employing novel technologies, and having separate toolkits across dev and ops teams.

The DevOps pipeline is a collection of interconnected procedures and tools that enable the Continuous Integration (CI), delivery, and deployment of software applications [6]. It promotes collaboration between development and operations teams, resulting in smooth communication, effective feedback loops, and quick iteration cycles. The primary purpose of a DevOps pipeline is to streamline and automate SD processes in order to reduce time for market while enhancing the overall quality of the product. The DevOps pipeline was created to address traditional SD challenges like manual methods, long release cycles and segregated teams.

Businesses that employ DevOps approaches want to foster a collaborative, agile as well as continuous improvement attitude may allow that organization to generate more reliable, scalable, and successful software but simultaneously adapting quickly to market demands. Figure 1 depicts the usual DevOps cycle [7]. The integration of the DevOps pipeline and performance testing provides various benefits to SD teams. For starters, it enables teams to identify performance issues as well as bottlenecks promptly enabling it to be addressed in resolving prior to they affect clients. Identifying and dealing with performance-related challenges as early as the development process can help organizations conserve time, money, and effort on reworking and operational tuning subsequent to development [8]. Furthermore, testing for performance in the DevOps pipeline encourages ongoing performance validation including monitoring in real time. It allows teams to gain useful insights about system behavior, performance trends, as well as ability of scalability over SDLC. DevOps offers more extensive support for cloud application deployment by utilizing a variety of tools that enable automation as well as continuous improvement. For any modern firm, software has become increasingly vital. Because software and its surrounds are very volatile, managing changes to software in a DevOps context has become a vital competence for a digital firm. In reality, the ever-changing technical specifications of cloud-based software might be having a comparable impact on its economics [9]. According to the researcher, the benefits of using cloud-enabled platforms include affordable access to enormous on-demand storage, processing, and network functions, as well as the ability to category a pool of competent workers via VMs throughout all stages of development. [10] [11].

2. LITERATURE REVIEW

Jinfu Chen has conducted investigations on the performance efficiency in the quality of software, as well as the difficulty of recognizing and treating performance regression analyses. Performance objectives are namely response time and throughput which is critical for guaranteeing the effectiveness of the system. However, the performance regressions may lead for breakdown system and degrade the user satisfaction. Discovering and resolving these regressions early in the SDLC is time-consuming and difficult. Chen's paper investigates the underlying causes of performance regressions over source code and offers a method towards automatic prediction of performance regressions [12]. Cor-Paul Bezemer et al. have suggested leveraging operational data as well as unit test execution logs to analyze efficiency and find regressions. The technique includes constraints such as requiring additional testing and verification over a software applications and environments with broader range to determine the generalization of the suggested methods. To acquire a better knowledge about the performance managed in industrial DevOps contexts, an extensive study was conducted [13]. An investigation examined the performance evaluations frequency of the tools utilized, the performance data extent, and the usage of model-based methods. Acquiring replies from multiple stakeholders in many industry areas yielded significant information. The study results shed light on a major impediment to the broad acceptance of performance analysis over DevOps are the complexities of performance analysis techniques and devices. This intricacy makes it difficult for developers to easily integrate performance analysis towards their DevOps workflows. To overcome this disadvantage and encourage wider adoption, performance analysis solutions must offer an inexpensive learning curve along with simple capabilities for integration. P. Batra and A. Jatain have emphasizes the performance evaluation significance in the DevOps framework. Their research recommends for incorporating quality features and comparison analysis based on measurement for enhancing performance evaluation techniques. This study helps businesses achieve efficient and excellent in quality delivery of software by providing light on difficulties and possible solutions, thus contributing to the body of information regarding SD processes [14]. Moreover, it is important to emphasize that the research fails to completely address the possibility of challenges and limitations associated with setting up performance evaluation in the DevOps context that can involve issues with tool selection, integrating performance evaluation and selection towards complex SD environments. Using the appropriate performance evaluation tools is crucial to carrying out successful evaluations [15]. There are several tools accessible that offer powerful functionality for load generation, analysis and monitoring. These applications are useful for simulating realistic user loads, measuring reaction times, capturing system metrics as well as evaluating data performance. When selecting performance evaluation tools that take into account the application's particular requirements as well as the desired level of testing complexities. Essential factors to consider include the accessibility of monitoring and statistical capabilities, compatibility and scalability with the technology stack, and ease of usage. Figure 1 shows some of the most popular performance evaluation techniques accessible currently.

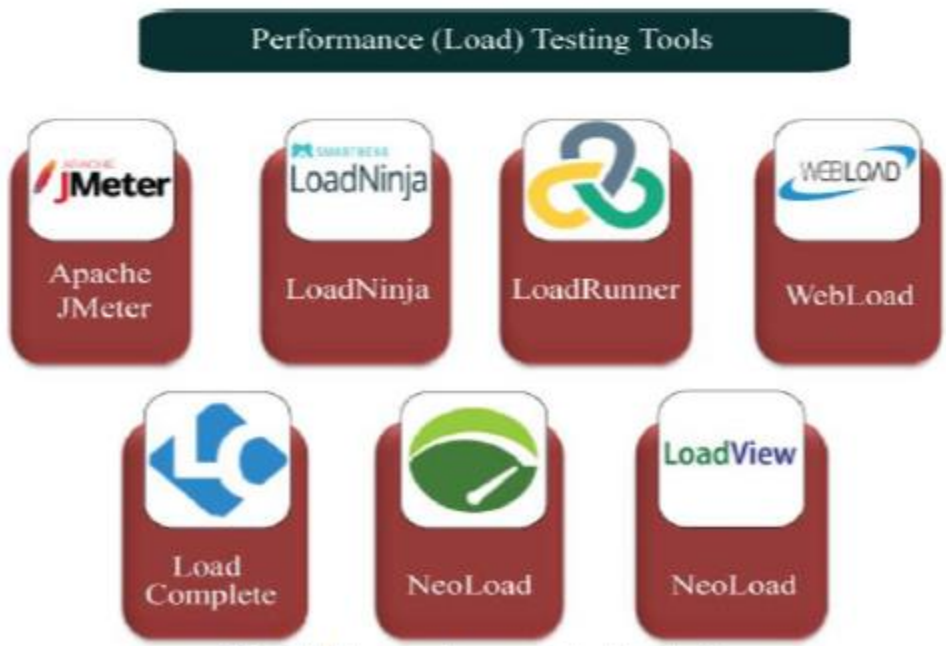

FIGURE 1. Performance evaluation technique for SD process

Automating performance tests minimizes the labor and time necessary to execute tests, allowing for more frequent testing. Automation allows teams to trigger, monitor and schedule the performance tests by default, allowing them to receive instant feedback on the application's functionality. This method improves consistency, scalability and efficiency in performance evaluation [16]. Several studies, including Akbar et al., have examined DevOps adoption in software organizations [17]. Researcher examined DevOps from the perspective of identifying DevOps success criteria, while Rafi et al. [18] presented a DevOps model of operation for tiny startups that work from home. This model is for deploying apps across hybrid clouds and selecting the best hosting match. They focused on the algorithm for determining the optimum cloud solution for the operations based on the DCs geographical location. Optimizing networking and connectivity is crucial for ensuring high performance and reliability in cloud computing environments. This section explores key techniques and strategies for enhancing network performance, reducing latency, and managing traffic efficiently in cloud infrastructures, drawing insights from comprehensive research in the field [19]. The experiments is to examine and demonstrate the effectiveness of the proposed algorithm in comparison to the conventional points by developing a multifaceted optimal strategy to generate deployment endeavors that use the ANOVA method for assessing the relationship among two groups. Thus, the findings are given in the form of an ANOVA table that divides the variation components over data towards variance among treatments and error or residual variation [20]. To determine the best answer, a Decision Maker consults with various other components. These attributes identify service providers, data hubs, applications, as well as customer footprints. The Resource Trader receives the DC id, instance type defined by memory space, the amount of CPUs, as well as storage, accessible instances, cost per hour, and assessment inputs for vendor capability.

3. RESEARCH METHODOLOGY

DevOps practices and automation play a crucial role in continuously optimizing cloud environments. By integrating development and operations teams and leveraging automation tools, organizations can achieve faster deployment cycles, improved reliability, and enhanced performance. This section explores key DevOps practices and automation techniques that contribute to continuous optimization in cloud computing, drawing insights from recent research in the field. IaC is a fundamental DevOps strategy which includes controlling and supplying computer infrastructure via readable by machines specification files instead of physical device setup or interaction setup applications. IaC is fundamental concept in DevOps, enabling consistent and repeatable infrastructure deployment. Furthermore, there is a requirement for a reliable and methodical approach to obtaining a system for version control which can be used for management, Docker operations, and library provision. AWS services are employed for deploying an application to the cloud. Jenkins serves as a CI and CD pipeline for overseeing multiple stages of development and keeps the SD process continuously. The ELK stack monitors and visualizes code execution. According to the outcomes, DevOps is an effective approach for the application with cloud deployment and selecting resources according to the relative significance of each optimized target with respect to the value parameters including cost, CPU capacity, memory and the technique can be customized to meet particular software requirements. Certain guidelines have arisen for optimizing DevOps pipelines by effectively incorporating performance evaluations. These techniques seek to assure precise and comprehensive performance evaluations, allowing businesses to identify and deal with problems with performance promptly, resulting in exceptional SD. The subsequent guidelines have highlighted key strategies to integrate performance evaluation towards DevOps practices are

Performance evaluation of Shift-Left

Incorporating performance evaluations at the start of the SDLC is a critical step towards optimizing DevOps workflows. Performance evaluations ought to begin early in the SD process in order to firms may discover and resolve issues with performance earlier they develop into major difficulties. Employing this strategy, teams can identify problems with performance and begin trying to resolve them earlier than they worsen and develop more challenging and costly to handle.

Implementing Test Environments with Resemble Production

Generate test settings that directly resemble the production environment is critical for obtaining reliable outcome of performance tests. Companies can mimic real-life situations and evaluate the performance of applications underneath realistic settings by accurately recreating the production circumstances, encompassing hardware configurations, connectivity to the network, and software requirements. This method guarantees that performance evaluation mimics the real-life manufacturing environment, which improves the test outcomes validity and dependability.

Implementing Continuous Performance evaluation

CI and CD processes are essential to DevOps pipelines, and including continuous performance evaluation into this process is extremely useful. Businesses may constantly track and assess the performance of their applications during the SDLC by incorporating performance evaluations into their CI and CD workflows. This method aids in the early detection of performance regressions enabling teams to resolve problems and begin implementing performance enhancements more quickly and reliably. Figure 2 illustrates the integration of performance evaluation in a usual CI and CD process.

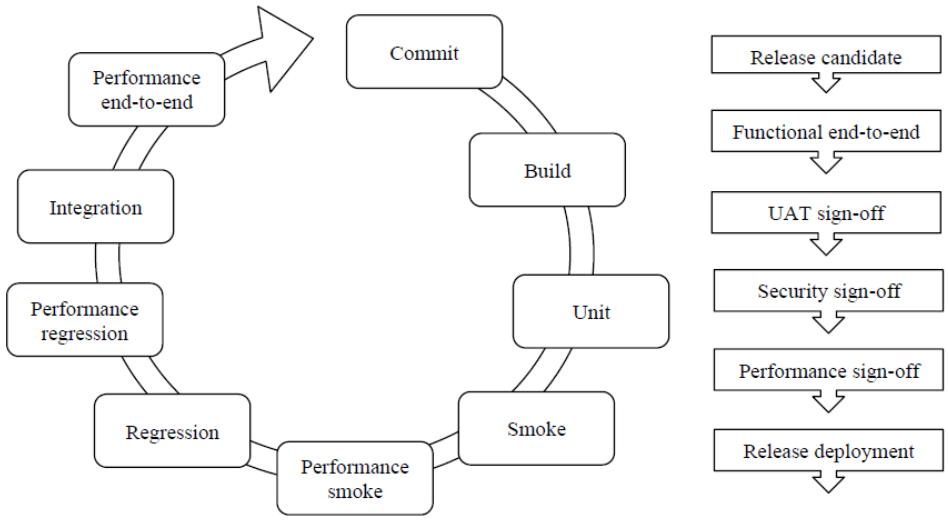

FIGURE 2. Performance evaluations for CI and CD cycle

Automation in Performance Tests

Automation technologies are crucial for simplifying the execution of performance tests and integrating them into the DevOps pipeline. The automation of performance tests minimizes the labor and time necessary to execute tests, allowing for greater frequency of testing. Automation allows organizations to trigger, monitor and schedule performance tests immediately, allowing them to receive immediate feedback concerning performance an application's. This method increases speed, scalability and consistency in performance evaluation activities.

Establishing Performance Baselines

Establishing performance baselines is critical in evaluating performance enhancement and making certain the application fulfills established performance standards. Providing preliminary benchmarks to important performance measures like response time, resource utilization and throughput that allows businesses for comparing successive outcomes of tests to monitor progress. Performance baselines are used as points of reference for assessing the effectiveness of optimization attempts and for assessing whether the software application fulfills the intended performance goals.

4. RESULT AND DISCUSSION

This section summarizes the findings from a study on enhancing DevOps processes through effective performance evaluation. The goal was to assess the effectiveness of performance evaluation techniques and procedures in developing high-quality, effective SDs. The findings show the advantages and disadvantages of applying performance evaluation in DevOps systems, highlighting the importance of ongoing performance verification, real-time monitoring, as well as continuous optimization. The findings indicate that incorporating performance evaluations early on in the SDLC has various advantages. When switching performance evaluations to the left, companies may identify and deal with problems with performance promptly, thereby minimizing the risk that these problems will impair the overall user experience. Continuous performance evaluation enables teams to detect performance regressions, verify performance upgrades, and assure consistent system efficiency through the SDLC. This technique leads to higher software quality, more users, and more confidence about the efficient application. Real-time monitoring has developed as an important approach for improving performance in DevOps systems. Companies can gain helpful insights towards a system's behavior during varying demands by implementing real-time monitoring frameworks.

Benefits of IaC include

- **Consistency:** IaC ensures that the same environment is reproduced every time, reducing configuration drift and related performance issues.

- **Version Control:** Infrastructure configurations can be version-controlled, allowing for easy rollbacks and auditing.
- **Scalability:** IaC facilitates rapid scaling of infrastructure to meet demand, crucial for performance optimization.
- **Documentation:** The code itself serves as documentation, improving knowledge sharing and reducing errors.

Popular IaC tools include Terraform, AWS CloudFormation, and Ansible. These tools allow for the definition of complex infrastructure setups in a declarative manner, enabling rapid and consistent deployment across different environments.

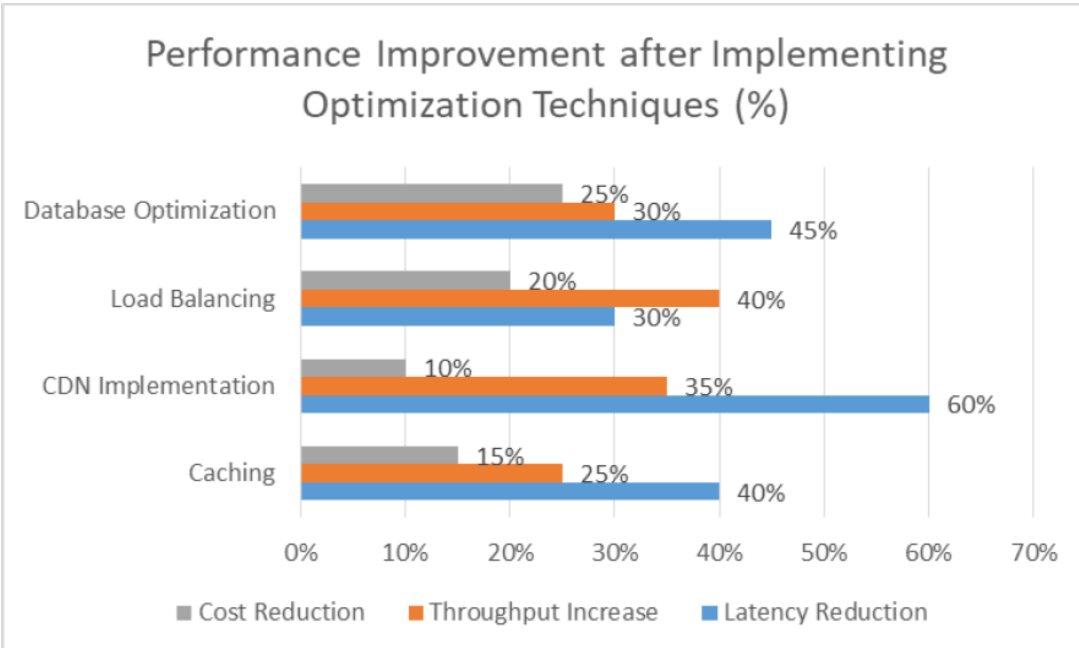

Figure 3 performance improvement using optimization technique

By implementing these performance monitoring and optimization techniques, organizations can ensure their cloud-based applications deliver optimal performance and a superior user experience. Regular monitoring, analysis, and optimization should be an ongoing process to maintain performance as applications evolve and usage patterns change. The work of Jennings and Stadler also underscores the importance of workload prediction and dynamic resource allocation in cloud environments. By leveraging advanced analytics and machine learning techniques, cloud systems can anticipate resource needs and automatically adjust allocations, further enhancing performance and efficiency.

5. CONCLUSION

In conclusion, optimizing performance in cloud computing environments is a multifaceted challenge that requires a holistic approach encompassing various aspects of cloud architecture, resource management, security, and governance. Throughout this article, we've explored key strategies for enhancing cloud performance, from designing scalable and resilient architectures to implementing effective monitoring and automation tools. This research report provides a thorough approach to enhancing DevOps pipelines by the effective usage of performance evaluation. The study emphasized the necessity of validating continuous performance, real-time monitoring, and iterative improvement in developing robust and resilient SDs. Organizations may deal with bottlenecks in performance, scalability difficulties, and potential issues influencing user experience as well as system reliability by implementing performance evaluation methods and processes across the SDLC. Based on the study's outcomes, shifting-left is a technique which might be utilized to incorporate performance evaluations early on in the SD process. The investigation also stressed the value of optimization by iteration in developing sturdy and resilient SD. Iterative optimization entails assessing performance evaluation results, determining areas for enhancement, and applying changes depending on the outcomes of the tests. This incremental approach eventually results in enhanced system performance, a capacity to accommodate increased user loads, as well as a consistent and dependable applications performance. Although the advantages of incorporating performance evaluation in DevOps are clear, the research also noted barriers to its adoption.

REFERENCES

[1]. Vivek Basavegowda Ramu, “PerfDetectiveAI - Performance Gap Analysis and Recommendation in Software Applications,” SSRG International Journal of Computer Science and Engineering, vol. 10, no. 5, pp. 40-46, 2023.

[2]. Mayank Gokarna, and Raju Singh, “DevOps: A Historical Review and Future Works,” 2021 International Conference on Computing, Communication, and Intelligent Systems, 2021.

[3]. Rafi, S.; Yu, W.; Akbar, M.A. RMDevOps: A road map for improvement in DevOps activities in context of software organi-zations. In Proceedings of the Evaluation and Assessment in Software Engineering, Trondheim, Norway, 15–17 April 2020; pp. 413–418.

[4]. Zarour, M.; Alhammad, N.; Alenezi, M.; Alsarayrah, K. Devops Process Model Adoption in Saudi Arabia: An Empirical Study. Jordanian J. Comput. Inf. Technol. 2020, 6, 3. [CrossRef]

[5]. Akbar, M.A.; Rafi, S.; Alsanad, A.A.; Qadri, S.F.; Alsanad, A.; Alothaim, A. Toward Successful DevOps: A Decision-Making Framework. IEEE Access 2022, 10, 51343–51362. [CrossRef]

[6]. Muhammad Owais Khan et al., “Fast Delivery, Continuously Build, Testing and Deployment with DevOps Pipeline Techniques on Cloud,” Indian Journal of Science and Technology, vol. 13, no. 5, pp. 552–575, 2020.

[7]. Saif Gunja, What is DevOps? Unpacking the Purpose and Importance of an IT Cultural Revolution, Dynatrace News, 2023.

[8]. Catia Trubiani et al., “Performance Issues? Hey DevOps, Mind the Uncertainty,” IEEE Software, vol. 36, no. 2, pp. 110–117, 2019.

[9]. Ghari, S. Devops for digital business: Optimizing the performance and economic efficiency of software products for digital business. In Proceedings of the 17th Symposium on Software Engineering for Adaptive and Self-Managing Systems, Pittsburgh, PA, USA, 18–23 May 2022; pp. 53–57.

[10]. Tsilionis, K.; Sassenus, S.; Wautelet, Y. Determining the Benefits and Drawbacks of Agile (Scrum) and DevOps in Addressing the Development Challenges of Cloud Applications. In Proceedings of the International Research & Innovation Forum, Athens, Greece, 15–17 April 2021; pp. 109–123.

[11]. Akbar, M.A.; Smolander, K.; Mahmood, S.; Alsanad, A. Toward successful DevSecOps in software development organizations: A decision-making framework. Inf. Softw. Technol. 2022, 147, 106894.

[12]. Jinfu Chen, “Performance Regression Detection in DevOps,” Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Companion Proceedings, pp. 206–209, 2020.

[13]. Cor-Paul Bezemer et al., “How is Performance Addressed in DevOps?,” Proceedings of the 2019 ACM/SPEC International Conference on Performance Engineering, pp. 45–50, 2019.

[14]. Pooja Batra, and Aman Jatain, “Measurement Based Performance Evaluation of DevOps,” 2020 International Conference on Computational Performance Evaluation, 2020.

[15]. Agata Kołtun, and Beata Pańczyk, “Comparative Analysis of web Application Performance Testing Tools,” Journal of Computer Sciences Institute, vol. 17, 2020.

[16]. Mohammad Rizky Pratama, and Dana Sulistiyo Kusumo, “Implementation of Continuous Integration and Continuous Delivery (CI/CD) on Automatic Performance Testing,” 2021 9th International Conference on Information and Communication Technology, 2021.

[17]. Akbar, M.A.; Mahmood, S.; Shafiq, M.; Alsanad, A.; Alsanad, A.A.A.; Gumaei, A. Identification and prioritization of DevOps success factors using fuzzy-AHP approach. Soft Comput. 2020. [CrossRef]

[18]. Rafi, S.; Akbar, M.A.; Manzoor, A. DevOps Business Model: Work from Home Environment. In Proceedings of the International Conference on Evaluation and Assessment in Software Engineering, Gothenburg, Sweden, 13 June 2022; pp. 408–412.

[19]. M. S. Aslanpour, S. S. Gill, and A. N. Toosi, "Performance evaluation metrics for cloud, fog and edge computing: A review, taxonomy, benchmarks and standards for future research," Internet of Things, vol. 12, p. 100273, 2020. [Online]. Available: https://doi.org/10.1016/j.iot.2020.100273

[20]. Padmanaban, S.; Khalili, M.; Nasab, M.A.; Zand, M.; Shamim, A.G.; Khan, B. Determination of Power Trans-formers Health Index Using Parameters Affecting the Transformer’s Life. IETE J. Res. 2022.

