The Role of Test Automation in Continuous Deployment for Cloud-Based Applications

Mohnish Neelapu

Category / Domain: QA Automation

Abstract:
In contemporary software development, Continuous Deployment (CD) has become vital procedures aimed at enhancing the dependability and effectiveness of software distribution. Such procedures are the foundation of DevOps, a collection of methodologies that closes the distance among operations and development, encouraging Cooperation and delivery pipeline automation. This paper delves into the role of test automation for cloud-based applications. Also, performance measures like Mean Time toward Detect (MTTD) and Mean Time to Resolve (MTTR) as well as contrasted with different software delivery tools.

Keywords: Test automation, Continuous Deployment (CD), performance metrics, cloud-based applications.

1. INTRODUCTION

Today, traditional software development methodologies are not good enough for businesses today. Making a transition towards agile practices helps improve the software development life cycle's flexibility, efficiency, and speed, something that software development firms are looking for [1]. Due to this, several businesses choose to have Need applications and a cloud-based automated product development program. Like a remedy, Cloud collaboration with DevOps makes administrations extremely strong, since they not only make the creation of software products nonetheless too enable an application as well as management of deployment procedure. DevOps is a collection of various procedures that unites software development (Dev) as well as IT operations (Ops) [2, 3]. DevOps is able to execute hundreds of tests in a single day as well as get feedback as of customers after every delivery. This will assist organization in investigating further features to be used in the project and reduce configuration problems [4]. DevOps help to accelerate creating high-quality goods, consistent shipping, and assisting final-user by high-quality software [5].

DevOps also allow prompt reaction to the altering client needs. DevOps provide a shared environment where developers and operations can work [6]. DevOps bring in the option of eliminating organizational and cultural barriers and reducing the price of identifying problems early [7]. Due to this, numerous researchers as well as organizations seek to create their own answers in order toward produce a product that can produce also make the whole process of Continuous Delivery (CD), Continuous Integration (CI), as well as Continuous Deployment (CDT) [8]. Continuous Deployment (CD) has become an important aspect of contemporary software development practices, especially when used in Agile and DevOps frameworks [9]. This article defines CD further streamlines software delivery by deploying the process automatically, with code being deployed automatically to production once it goes through several testing stages. This automation minimizes human errors, reduces downtime, and [10] enables more frequent releases, improving an organization's capability to provide new features and upgrades to users in a timely manner. Consequently, CI/CD enables software development teams to gain quicker time-to-market and react more efficiently to customers' needs and shifting requirements.

1.1 Objectives

The research objectives are explained in the points below

- To evaluate how effective automated testing frameworks are in enhancing the reliability of cloud-based applications.

- Toward evaluate an effect of automated testing on defect detection and fix times.
- In order to evaluate the effects of automated testing on presentation and scalability of cloud applications.

2. LITERATURE REVIEW

Dakkak, A. et al., 2021 [11] A significant amount of pressure was placed on enterprises to introduce software quickly and steadily in order to compete with others as an outcome of the rising desire aimed at novel products and technology. As a result, the majority of businesses need cloud-based apps and chooses an automated approach for product creation.

In 2022 Tatineni, S. [12] concentrates on CI/CD pipeline optimization in DevOps contexts in 2022. The approach includes a thorough examination of current CI/CD procedures, a determination of the degree of automation, and an investigation of tactics including orchestration, parallelization, distribution, and containerization. Based on data gathered from industry experts,

Akbar, M.A. et al. [13] developed a framework on prioritization-based of DevOps best practices in 2022. To get industry practitioners' opinions on the best practices that were identified, a questionnaire survey study was carried out. Lastly, the best practices were ranked according to their importance for the DevOps process using the fuzzy-AHP technique. We think that the fuzzy-AHP based architecture, best practices, and their classification will assist industry professionals in updating and enhancing their approaches to make the DevOps process sustainable.

Rouf, Y. et al. [14] introduce a MAPE-K framework that uses pre-existing commercial off-the-shelf (COTS) components to communicate with one another and carry out self-adaptive tasks in multi-cloud settings 2021. By incorporating current COTS, We can effectively implement a Monitoring, Analysis, Planning, Execution- Knowledge (MAPE-K) framework to assist DevOps for apps operating in a multi-cloud setting. We demonstrate the practical practicality of our framework by a thorough case study conducted on an actual industrial platform, and we validate it with a prototype implementation.

2.1 Research gap

According to the literature analysis above, cloud apps and manual testing are limited. Cloud applications frequently have complex architectures with a large number of microservices and dependencies. It is difficult to maintain consistent performance and dependability in different situations because of this complexity. Limitations of Manual Testing: Manual testing takes a lot of time, is prone to human mistake, and isn't scalable enough to manage cloud applications' dynamic nature. Delays in feedback and a higher chance of production flaws are the outcomes of this.

The structure of the paper is followed by: Section 1 explains the introduction, section 2 explains literature survey, section 3 explains proposed methodology, and section 4 explains result analysis and conclusion part in section 5.

3. METHODOLOGY

Automated testing frameworks, when integrated into the DevOps pipeline seamlessly, provide unmatched advantages, such as faster feedback loops, better test coverage, and huge savings in manual testing efforts. In this suggested model describes several automated testing tools and techniques [15], highlighting their use in cloud-native architectures. Figure 1 describes the suggested model of automated test in continuous deployment of cloud-based applications.

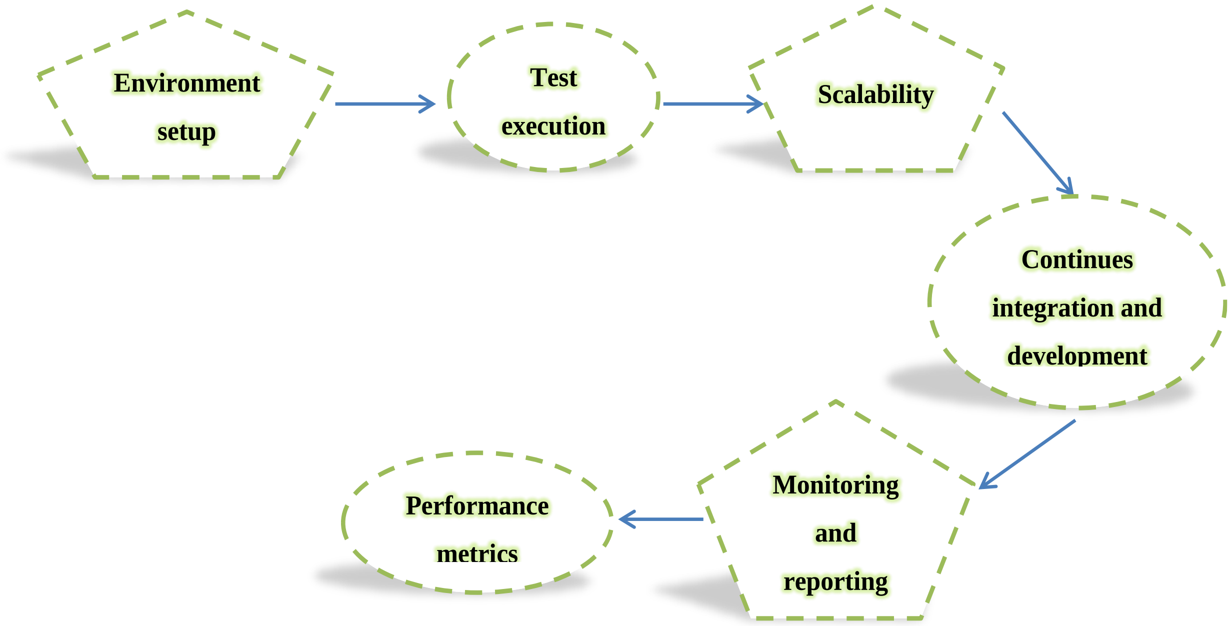

Fig 1: proposed model of test automated cloud based applications

3.1 Role of test automation in cloud-based applications

Improved Reliability and Performance

Autobahn testing suites greatly enhance cloud application reliability and performance. Automating testing leads to consistent, comprehensive testing of applications, where defects are uncovered and prevented as early as in the development life cycle, leaving little room for regressions as well as in-production issues. This yields powerful applications that have consistent performance levels under diverse environmental conditions [16]. Quicker Development Lifecycles With automated testing within the DevOps pipeline, there is faster speed in the development cycle. CI/CD workflows, when integrated with automated testing, deliver faster feedback and faster iteration. This allows growth teams to spot as well as fix problems faster creating a setting for ongoing improvement and new ideas.

Increased Development Cycles

The development process is accelerated by the DevOps pipeline's use of automated testing. Faster iterations and feedback are produced when automated testing is combined with CI/CD techniques. This promotes a culture of innovation and continuous improvement by enabling development teams to identify and address issues promptly.

Complete Test Coverage

A set of testing approaches including functional, performance, security and regression testing make up the full coverage of assessments delivered through automated testing systems [17]. The application quality improves through fixing identified vulnerabilities and performance issues when testing includes sufficient application coverage.

Scalability and Flexibility

Test frameworks can run extensive test suites under multiple operational environments and configurations while cloud platforms work as natural growing platforms. The ability to test the application under conditions similar to real-world usage proves beneficial when testing occurs under different conditions [18]. Cloud-based testing enables flexible resource deployment at user demand to improve testing success while lowering budget costs.

Less Manual Labor and Human Error

Automated testing cuts down on the reliance on manual testing, which is usually tiresome and vulnerable to errors by humans. Automated testing of redundant and complicated test tasks allows development teams to devote their time and energy to more strategic projects [19] like creating new features and optimizing user experience. This change enhances productivity and results in more accurate test outcomes.

Better Collaboration and Integration

Automated testing within a DevOps environment encourages increased collaboration between the development, operations, and quality assurance teams. Pipeline-based integrated testing closes gaps in communication and coordination issues so that integration of code changes becomes simple and easy. In this collaborative process, all the parties are aligned on the same idea and collaborate with each other in order to ship quality software.

Cost Effectiveness

While the initial setup and upkeep of automated test framework may be costly, the long-term cost savings are significant. Automation reduces manual test resources to be utilized for lengthy periods, in addition to reducing post-production fix-up charges of bugs. Both time and money savings end up being monumental [20].

4. RESULT ANALYSIS

In this section explains the result analysis of performance metrics of various software delivery tools with proposed DevOps. Performance metrics such as MTTD and MTTR were analyzed.

4.1 Quantitative Data

Surveys and Questionnaires: Sent to IT professionals and DevOps engineers with experience working with automated testing frameworks. They ask about perceived reliability improvements, defect fix times, and overall system performance.

Performance Metrics: Gathered from cloud applications prior to and subsequent to introducing automated testing environments. Performance metrics are defect rates, Mean Time to Detect (MTTD), Mean Time to Repair (MTTR), application availability, as well as system performance during load. Figure 2 explains the performance metrics evaluation for proposed and existing software model. In that the DevOps attains the minimum time to solve the issue.

MTTD is a measurement of the duration it takes to identify a difficult or incident.

MTTR is a measure of the amount of time needed to fix a system, product, or service after it fails.

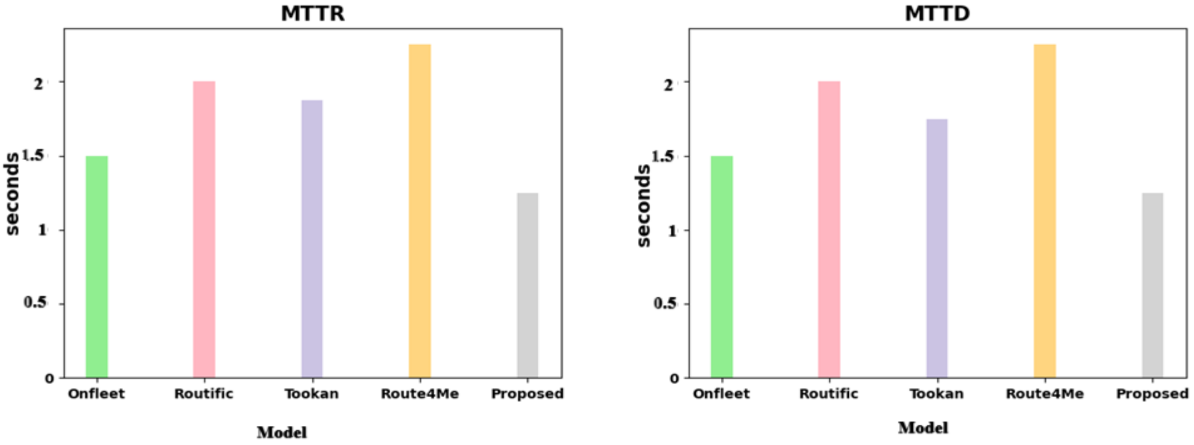

Fig 2: Performance metrics

5. CONCLUSION

Cloud-based DevOps infrastructure requires the implementation of automated testing frameworks in order to enhance the reliability along with effectiveness of cloud applications. This research study demonstrates how automated testing brings major changes to application development through performance enhancement of stability and defect detection speed and system-wide performance optimization under varied load scenarios. Automated test frameworks used in our study demonstrate they decrease failure rates of applications and time spent offline which builds cloud-based solution reliability. The research indicates that these frameworks enhance both defect identification speed and correction efficiency thus reducing MTTD and MTTR metrics. Automated testing allows organizations to manage performance through various environments which facilitates cloud applications to scale effectively. The research proves that modern DevOps methods require automated test frameworks for their cloud-based operations. Businesses benefit from automated frameworks by getting better software quality in competitive markets through their dual advantage of simplified expansion and improved application reliability. The execution of subsequent investigations will study sophisticated testing methods with modern technology to develop safer and easier automated testing methods.

REFERENCES:

1. G. Nahar, and S. B. Jain, “Continuous Testing in Agile: A Game-Changer or a Bottleneck?”.
2. M. Zarour, N. Alhammad, M. Alenezi, K. Alsarayrah, “Devops Process Model Adoption in Saudi Arabia: An Empirical Study,” Jordanian J. Comput. Inf. Technol., vol. 6, no. 3, 2020.
3. M. A. Akbar, S. Rafi, A. A. Alsanad, S. F. Qadri, A. Alsanad, and A. Alothaim, “Toward Successful DevOps: A Decision-Making Framework,” IEEE Access, vol. 10, pp. 51343–51362, 2022.
4. Weaveworks. Building Continuous Delivery Pipelines. Available online: https://www.weave.works/assets/images/blta80840 30436bce24/CICD_eBook_Web.pdf (accessed on 10 January 2022).
5. R. Ghimire, “Deploying Software in the Cloud with CI / CD Pipelines,” 2020. Available online: https://www.theseus.fi/bitstream/handle/10024/345618/Thesis_Ramesh_Ghimire_1.pdf?sequence=2 (accessed on 27 October 2022).
6. M. Zarour, N. Alhammad, M. Alenezi, and K. Alsarayrah, “Devops Process Model Adoption in Saudi Arabia: An Empirical Study,” Jordanian J. Comput. Inf. Technol., vol. 6, no. 3, 2020.
7. M. A. Akbar, S. Rafi, A. A. Alsanad, S. F. Qadri, and A. Alsanad, Alothaim, A. “Toward Successful DevOps: A Decision-Making Framework,” IEEE Access, vol. 10, pp. 51343–51362, 2022.
8. Alanda, A., Mooduto, H.A. and Hadelina, R., 2022. Continuous Integration and Continuous Deployment (CI/CD) for Web Applications on Cloud Infrastructures. JITCE (Journal of Information Technology and Computer Engineering), 6(02), pp.50-55.
9. Vadde, B.C. and Munagandla, V.B., 2022. AI-Driven Automation in DevOps: Enhancing Continuous Integration and Deployment. International Journal of Advanced Engineering Technologies and Innovations, 1(3), pp.183-193.
10. GUPTA, E.V., 2022. Continuous Integration and Deployment: Utilizing Azure DevOps for Enhanced Efficiency.
11. A. Dakkak, D. I. Mattos, and J. Bosch, “Perceived benefits of continuous deployment in software-intensive embedded systems,” In 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC). IEEE, pp. 934-941, 2021, July.
12. S. Tatineni, “Optimizing Continuous Integration and Continuous Deployment Pipelines in DevOps Environments,” INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING AND TECHNOLOGY (IJCET), vol. 13, no. 3, pp. 95-101, 2022.
13. M. A. Akbar, S. Rafi, A. A. Alsanad, S. F. Qadri, A. Alsanad, and A. Alothaim, “Toward successful DevOps: a decision-making framework,” IEEE Access, vol. 10, pp. 51343-51362, 2022.
14. Rouf, Y., Mukherjee, J., Litoiu, M., Wigglesworth, J. and Mateescu, R., 2021, April. A framework for developing devops operation automation in clouds using components-off-the-shelf. In Proceedings of the ACM/SPEC International Conference on Performance Engineering (pp. 265-276).
15. A. Sharma, and A. Singh, “Comparative study of automated testing tools in DevOps,” Software Quality Journal, vol. 28, no. 3, pp. 1117-1136, 2020.
16. D. Vlasenko, and T. McElroy, “Automated testing in cloud-based applications: Best practices and case studies,” IEEE Cloud Computing, vol. 8, no. 3, pp. 45-53, 2021.
17. Y. Yang, and Y. Zhang, “Automation in cloud DevOps: Tools, techniques, and future directions,” IEEE Transactions on Cloud Computing, vol. 8, no. 2, pp. 352-367, 2020.
18. O. Götz, and J. Götz, “Automated testing in cloud computing: A review of current practices and tools,” Software Testing & Verification, vol. 30, no. 8, pp. 1057-1082, 2020.
19. M. O. Khan, A. K. Jumani, and W. A. Farhan, “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.
20. S. Karlsson, A. Čaušević, D. Sundmark, and M. Larsson, “Model-based automated testing of mobile applications: an industrial case study,” In 2021 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), IEEE, pp. 130-137, 2021, April.