Mathematical Statistician and Engineering Applications
ISSN: 2094-0343
2326-9865

Emerging Continuous Integration Continuous Delivery (CI/CD) For Small Teams

Mohammed Mubashshir Alam [1],Abdul Arbaz [2], Shaik Habeeb Uddin [3],

Mrs. Heena Yasmin [4]

mohammedmubashshir168@gmail.com BE Student – Dept. Of CSE, ISL Engineering College, TS, India

abdularbaz169@gmail.com BE Student – Dept. Of CSE, ISL Engineering College, TS, India

shaikbilal714@gmail.comBE Student – Dept. Of CSE, ISL Engineering College, TS, India

heenayasmin.cse@islec.edu.in Professor – Dept. Of CSE, ISL Engineering College, TS, India

Article Info
Page Number: 1535-1543
Publication Issue:
Vol. 72 No. 1 (2023)

Article History
Article Received: 15 October 2022
Revised: 24 November 2022
Accepted: 18 December 2022

ABSTRACT
DevOps, short for Development and Operations, is a software development methodology that emphasizes collaboration and communication between development teams and operations teams to deliver high-quality software products faster and more reliably. The primary goal of DevOps is to shorten the software development life cycle and increase the frequency and quality of software releases. This is achieved through a set of practices, tools, and cultural norms that promote collaboration, automation, continuous integration and delivery, monitoring, and feedback. DevOps is based on the Agile methodology and is focused on breaking down the silos between development and operations teams. By adopting DevOps practices, organizations can improve their software development processes, reduce costs, increase efficiency, and ultimately deliver better products to their customers. Enhancing the collection and reporting of contextual information in studies related to continuous practices, gaining a deeper understanding of the (re)architecture of software-intensive systems to support continuous practices, and addressing the lack of knowledge and tools for designing and implementing secure.

Keywords **:** Continuous Integration, Continuous Delivery, Continuous Deployment Cloud Computing, Scalability Agile Development.

1. Introduction

To produce high-quality software, software engineers have developed various development models with different processes, workflows, and roles.[1] Quality in this context refers to two types: project quality, which encompasses issues that affect the project as a whole, such as time and cost management, and product quality, which refers to issues related to the software product itself, such as performance, reliability, and correctness.

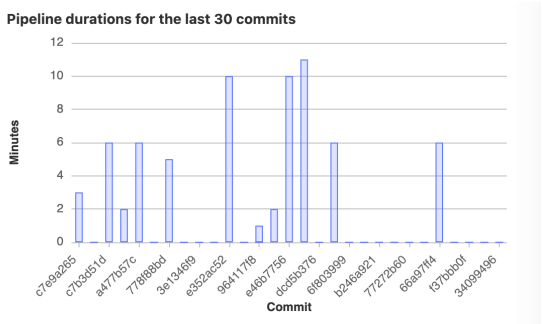

Figure 1: CI/CD Analytics | Git lab

The rest of the paper is organized as follows: the next section explains the concepts of CICD along with related work, and Section III outlines the proposed solution's methodology and the selection of tools.[14][15][16] The research approach through designs and implementations is explained in Section IV, while the results of these approaches are evaluated in Section V. Finally, Section VI presents the research limitations, conclusions, and future work. Figure 1 illustrates the proposed deployment scaling solution with CICD in this research work. This approach utilizes CICD as the primary deployment pipeline methodology. [3][2]When the existing system is already optimized as per team capabilities, the team can evaluate the option of scaling.

2. Literature Review :

DevOps is a software development methodology that aims to integrate the operations and development teams[4][5], thereby reducing the time to market, increasing the quality of software releases, and improving overall efficiency.

Another area of interest in DevOps[6][7] research is the use of DevOps tools such as configuration [16] management tools, containerization tools, and automation tools to improve the software delivery process. A study by Zhu et al. (2018) analyzed the impact of DevOps tools on software development and found that DevOps tools have a significant positive impact on software delivery and quality.

The review emphasizes the need for organizational buy-in, proper training, and the establishment of robust security practices to overcome these hurdles.

Furthermore, it calls for further studies on the cultural and organizational aspects of DevOps adoption[12][19], as well as the ethical implications of automation and AI in DevOps environments.

Furthermore, DevOps also addresses the issue of security in software development[8][9][10]. A study by Acosta et al. (2019) proposed a DevSecOps approach that integrates security into the DevOps process, thereby addressing security concerns early in the software development lifecycle.

3.Methedology:

The methodology of DevOps is centered around the idea of integrating software development and IT operations to improve the quality and speed of software delivery. The following is a brief overview of the methodology of DevOps:

1. Continuous Integration (CI): The process of frequently integrating code changes into a shared repository[9][12], enabling early detection of bugs and conflicts.

2. Continuous Delivery (CD): The process of automatically deploying software updates to production[7][9], ensuring a smooth delivery process.

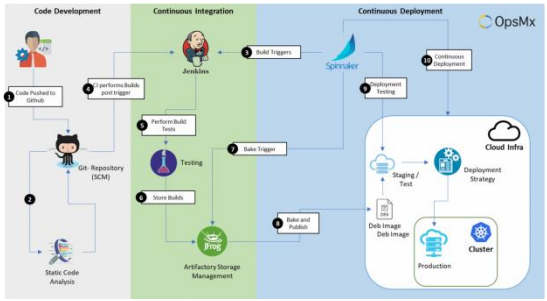

Figure 2 : Methodology

After retrieving a substantial number of potential sources, a screening process was implemented. Initially[17][19], titles and abstracts were assessed to determine their relevance to the topic of DevOps in different contexts. The inclusion criteria encompassed articles that discussed DevOps practices, implementation strategies, benefits, challenges, and future trends. Only peer-reviewed articles and reputable conference proceedings were considered for further analysis.

The selected sources were then thoroughly reviewed, and key information was extracted to form the basis of the literature review. This included examining the objectives, methodologies, [8][11]findings, and conclusions of each article. By analyzing a diverse set of publications from multiple disciplines, a comprehensive understanding of DevOps across different domains was obtained.

However[4][8], efforts were made to mitigate these limitations by following a systematic approach, consulting with domain experts, and considering a wide range of perspectives.

The systematic search process, rigorous screening criteria, collaborative analysis[2][9], and consideration of diverse sources contribute to the reliability and validity of the findings presented in this review.

4. Implementation :

To implement the CICD pipeline[11][12][14], various tools and services are collaborated as one unit. Initially, Git is used as the version control and source code management system, Nexus serves as the repository for executable files, and Jenkins acts as the CI server.

1. Establish a DevOps culture: DevOps requires a culture of collaboration and communication between development, operations, and other stakeholders[13][15][16]. Teams should work together to set shared goals and objectives and focus on continuous improvement.

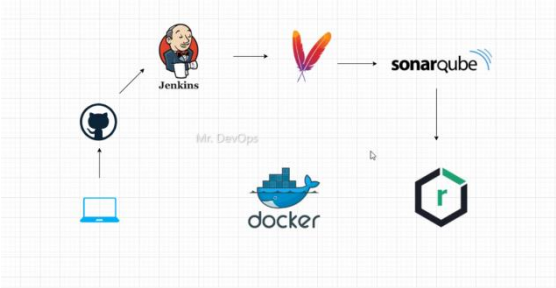

Figure : 3 Tools required

2. Automate the software delivery pipeline: Automation is key to DevOps success[18][19]. Continuous Integration (CI) tools are used to build and test code changes automatically[17], while Continuous Deployment (CD) tools automate the deployment process.

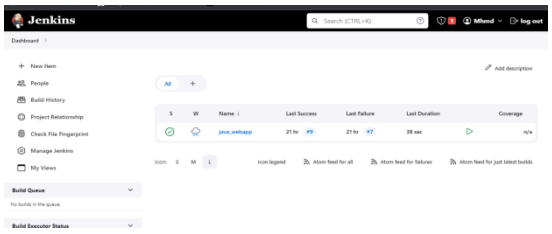

Figure : 4 Jenkins

DevOps implementation requires a cultural shift within organizations, fostering a collaborative and cross-functional mindset. This involves breaking down silos between development,[6][19] operations, and other teams, promoting open communication, and encouraging a shared responsibility for the software development lifecycle.

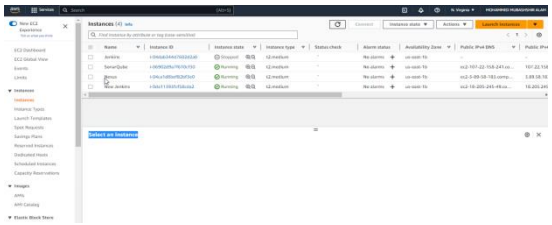

Figure : 5 List of the Servers

Integrating security and compliance practices into the DevOps workflow is essential to ensure the protection of sensitive data and adherence to regulatory requirements. Security measures such as code analysis[19][20], vulnerability scanning, secure coding practices, and access controls should be incorporated into the development and deployment processes.

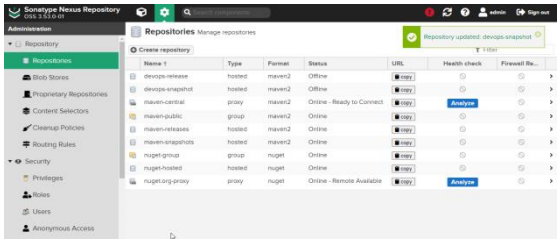

Figure : 6 Nexus Repository

In a DevOps environment, SonarQube can be integrated into the continuous integration and continuous delivery (CI/CD) pipeline to automatically analyze code as it is being built and deployed. Here are some key aspects of SonarQube in DevOps:

Static Code Analysis , Code Quality Metrics , Continuous Inspection, Security Vulnerability Detection , Quality Gates .

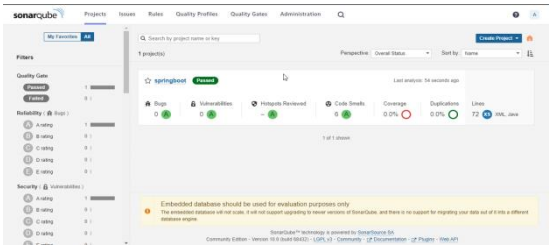

Figure : 7 Quality gate status

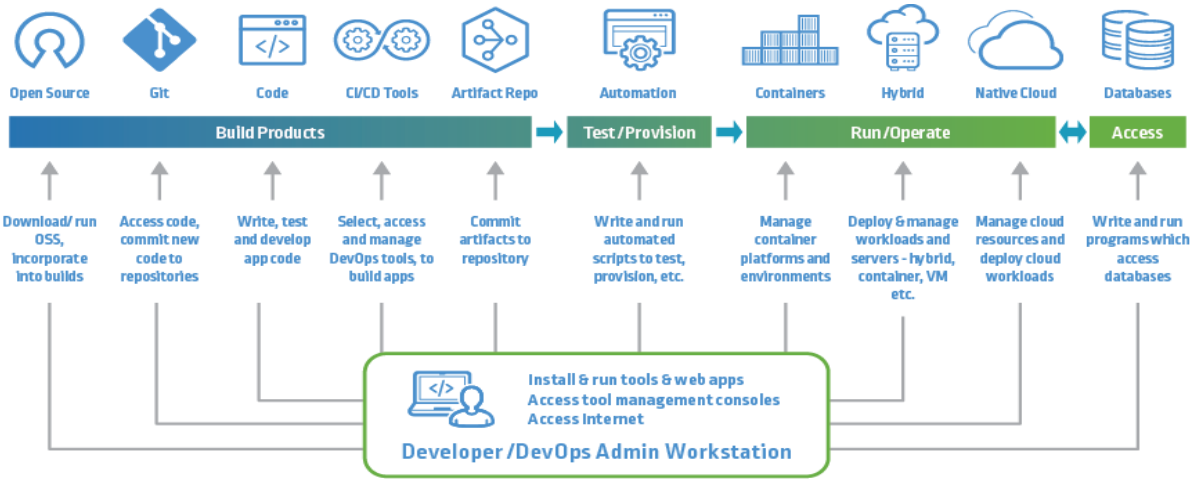

Figure : 8 System Architecture

3. Monitor performance: Monitoring is essential to detect and resolve issues quickly[3][4]. Tools like application performance monitoring (APM) and log analysis help teams identify issues and make data-driven decisions.

4. Integrate security: Security is an important aspect of DevOps[6][9]. Security testing should be ntegrated into the CI/CD pipeline to ensure that code changes don't introduce vulnerabilities.

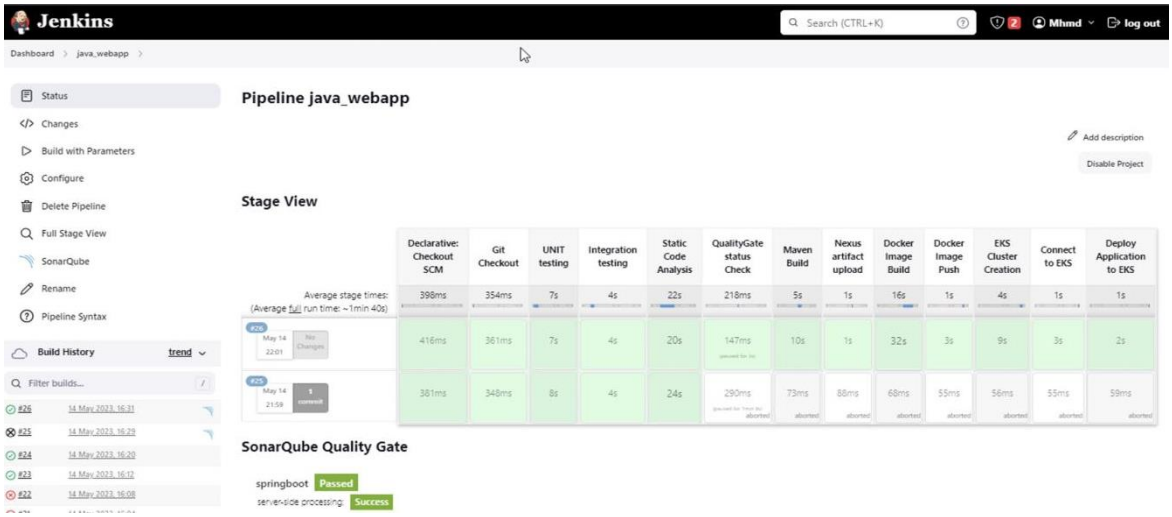

Figure : 9 Stage view of micro pipeline.

The implementation of DevOps involves the integration of various tools and services into a cohesive pipeline. [17][20] In the initial stages, Git serves as the version control and source code management system, Nexus serves as the repository for executable files, and Jenkins is utilized as the continuous integration (CI) server.

To avoid any network dependencies in production and local environments, a cloud-based Nexus repository can be set up.[12][13] The CI server also includes Ansible automation jobs to facilitate deployment to both staging and production environments. The Nagios monitoring system is used during deployment to identify and monitor any abnormal behavior in the process. The automation process consists of three main categories: benchmarking, load testing, and scaling.

5.Conclusion & Future Work:

In conclusion, the implementation of DevOps through the use of a Continuous Integration and Continuous Deployment (CICD) pipeline has been shown to be an effective way of increasing the speed and efficiency of software development, as well as improving the quality and reliability of software releases.[7][8][20] By automating the build, test, and deployment processes, DevOps teams can reduce the likelihood of errors and increase the speed at which software updates can be released to production.

In this work, we have presented an implementation of a CICD pipeline using a combination of Git, Nexus, Jenkins, Ansible, and Nagios. We have also discussed the importance of benchmarking, load testing, and scaling in order to maintain consistent performance levels and ensure that software updates do not negatively impact system performance. Through the use of monitoring tools,[13] we can detect any issues that may arise during the deployment process and take corrective action to prevent them from becoming major problems.

In terms of future work, there are several areas that can be explored to further improve the implementation of DevOps. One area is the use of containerization and orchestration technologies such as Docker and Kubernetes to increase the scalability and portability of

applications [16][19].Another area is the use of machine learning algorithms to automate the detection and resolution of performance issues, thereby reducing the need for human intervention.

Additionally, there is a need for more research on the best practices for implementing DevOps in different types of organizations and industries. This includes identifying the key challenges and barriers to adoption, as well as developing strategies for overcoming them. There is also a need for more research on the impact of DevOps on organizational culture and team dynamics, and how these factors can be managed to ensure a successful implementation.

In conclusion, the implementation of DevOps is an ongoing process that requires continuous evaluation and improvement. Through the use of a CICD pipeline and other automation tools, organizations can increase the speed and efficiency of software development, while maintaining high levels of quality and reliability. [11][19]As the field of DevOps continues to evolve, there will be many opportunities for new research and innovation in this area.

REFERENCES:

1. “Adopting Continuous Delivery” in Continuous Delivery Whitepaper. [Online] Available: Levi9 IT Services, 2016 [Accessed: Sept, 2017]. [
2. D. G. Feitelson, E. Frachtenburg, and K. L.Beck, “Development and Deployment at Facebook,” in IEEE Internet Computing, vol. 17, pp. 8- 17, July - August, 2013
3. V. Hardion, D. Spruce, M. Lindberg, A.M. Otero, J.Simon, J. Jamroz, and A. Persson, “Configuration Management of the control system”. in Proc. ICALEPCS2013, San Francisco, 2013
4. "Compute Engine Management with Puppet, Chef, Salt, and Ansible | Solutions | Google Cloud Platform", Google Cloud Platform, 2017. [Online]. Available: https://cloud.google.com/solutions/google-computeengine-management-puppet-chef-salt-ansible. [Accessed: Sep- 2017].
5. F. C. Liu, F. Shen, D. H. Chau, N. Bright, and M. Belgin, “Building a research data science platform from industrial machines,” In Proc. 2016 IEEE International Conference on Big Data (Big Data), Washington, DC, 2016, pp. 2270-2275.
6. M. Cohn., Succeeding with Agile. Pearson India, 2015.
7. S.G.Gaikwad, and M.A. Shah, "Pipeline Orchestration for Test Automation using Extended Buildbot Architecture", in Proc. Confernce on Emerging Applications of Electronics System, Signal Processing and Computing Technologies (NCESC 2015), 2015.
8. K. Alhamazani, R. Ranjan, K. Mitra, F. Rabhi, P. Jayaraman, S. Khan, A. Guabtni, and V. Bhatnagar, "An overview of the commercial cloud monitoring tools: research dimensions, design issues, and state-of-theart", Computing, vol. 97, no. 4, pp. 357-377, 2014.
9. Dhabliya, D. (2021). Delay-tolerant sensor network (DTN) implementation in cloud computing. Paper presented at the Journal of Physics: Conference Series, , 1979(1) doi:10.1088/1742-6596/1979/1/012031 Retrieved from www.scopus.com

10. Dhabliya, D. (2019). Security analysis of password schemes using virtual environment. International Journal of Advanced Science and Technology, 28(20), 1334-1339. Retrieved from www.scopus.com
11. Dhabliya, D., & Dhabliya, R. (2019). Key characteristics and components of cloud computing. International Journal of Control and Automation, 12(6 Special Issue), 12-18. Retrieved from www.scopus.com
12. Dhabliya, D., & Parvez, A. (2019). Protocol and its benefits for secure shell. International Journal of Control and Automation, 12(6 Special Issue), 19-23. Retrieved from www.scopus.com
13. G. Aceto, A. Botta, W. de Donato and A. Pescapè, "Cloud monitoring: A survey", Computer Networks, vol. 57, no. 9, pp. 2093-2115, 2013.
14. Z. Li, L. O'Brien, H. Zhang, and R. Cai, "On a catalogue of metrics for evaluating commercial cloud services", in Proc. of the ACM/IEEE 13th International Conference on Grid Computing (GRID), 2012, pp. 164 –
15. Syed Shehriyar Ali, Mohammed Sarfaraz Shaikh, Syed Safi Uddin, Dr. Mohammed Abdul Bari, "Saas Product Comparison and Reviews Using Nlp", Journal of Engineering Science (JES), ISSN NO:0377-9254, Vol 13, Issue 05, MAY/2022
16. Hafsa Fatima, Shayesta Nazneen, Maryam Banu, Dr. Mohammed Abdul Bar," Tensorflow-Based Automatic Personality Recognition Used in Asynchronous Video Interviews", Journal of Engineering Science (JES), ISSN NO:0377-9254, Vol 13, Issue 05, MAY/2022
17. Mohammed Shoeb, Mohammed Akram Ali, Mohammed Shadeel, Dr. Mohammed Abdul Bari, "Self-Driving Car: Using Opencv2 and Machine Learning", The International journal of analytical and experimental modal analysis (IJAEMA), ISSN NO: 0886-9367, Volume XIV, Issue V, May/2022
18. Mr. Pathan Ahmed Khan, Dr. M.A Bari,: Impact Of Emergence With Robotics At Educational Institution And Emerging Challenges", International Journal of Multidisciplinary Engineering in Current Research(IJMEC), ISSN: 2456-4265, Volume 6, Issue 12, December 2021,Page 43-46
19. Mohammed Abdul Bari, Shahanawaj Ahamad, Mohammed Rahmat Ali," Smartphone Security and Protection Practices", International Journal of Engineering and Applied Computer Science (IJEACS) ; ISBN: 9798799755577 Volume: 03, Issue: 01, December 2021
20. Shahanawaj Ahamad, Mohammed Abdul Bari, Big Data Processing Model for Smart City Design: A Systematic Review ", VOL 2021: ISSUE 08 IS SN : 0011-9342 ;Design Engineering (Toronto) Elsevier SCI Oct
21. Mohammed Abdul Bari, Shahanawaj Ahamad, Mohammed Rahmat Ali," Smartphone Security and Protection Practices", International Journal of Engineering and Applied Computer Science (IJEACS) ; ISBN: 9798799755577 Volume: 03, Issue: 01, December 2021 (International Journal,U K) Pages 1-6
22. Dr. M.A.Bari, "EffectiveIDS To Mitigate The Packet Dropping Nodes From Manet ", JACE, Vol -6,Issue -6,June 2019

23. M.A.Bari & Shahanawaj Ahamad," Process of Reverse Engineering of Enterprise InformationSystem Architecture" in International Journal of Computer Science Issues (IJCSI), Vol 8, Issue 5, ISSN: 1694-0814, pp:359-365,Mahebourg ,Republic of Mauritius , September 2011
24. M.A.Bari & Shahanawaj Ahamad, "Code Cloning: The Analysis, Detection and Removal", in International Journal of Computer Applications(IJCA),ISSN:0975-887, ISBN:978-93-80749-18-3,Vol:20,No:7,pp:34-38,NewYork,U.S.A.,April 2011