

VIII. CONCLUSION

According to the report, numerous IT companies have demonstrated that releases and deployments shouldn't be high-risk, tightly connected, or necessitate the deployment of hundreds or even thousands of engineers. Instead, it ought to be carried out in a way that becomes commonplace and a necessary component of daily operations. By doing this, we may further reduce the lead times in seconds and limit the impact of botched deployments and pandemonium at the production level. Our project makes this possible by reducing the time and labour that is required to develop, test, and compile the application at Testing and Production environment. Numerous deployment strategies like A/B Testing, Blue-Green, Shadow, Ramped etc. are assisted by providing automated development for multiple programming languages. In order to reduce the release time, Continuous Integration and Continuous Deployment Pipelines are developed and executed. The scalability of the application is vast according to user requirement requirements and traffic while integrating Kubernetes and multi-cloud platforms. This not only encourages the isolation between different applications but also makes it possible for per-second billing approaches to earn revenue from the resources based on usage per second.

IX. FUTURE SCOPE

For future enhancement, the effect of using CI/CD can be analysed in various contexts towards finding a generalized result. The proposed solution can be extended to be made compatible to work differently on different types of systems

and platforms. The system can also be improved in order to reduce downtime and analysed to find best cloud platform for it.

REFERENCES

- [1] Floris Erich, Chintan Amrit and Maya Daneva, "A Qualitative Study of DevOps Usage in Practice," *Journal of Software: Evolution and Process*, Vol. 29, No. 6, pp. 1-20, 2017.
- [2] L. E. Lwakatara, P. Kuvaja and M. Oivo, "Dimensions of DevOps," In: Lassenius, C., Dingsøyr, T., Paasivaara, M. (eds) *Agile Processes in Software Engineering and Extreme Programming. XP 2015. Lecture Notes in Business Information Processing, Springer, Cham*, Vol. 212, 2015. https://doi.org/10.1007/978-3-319-18612-2_19.
- [3] G. Blinowski, A. Ojdowska and A. Przybyłek, "Monolithic vs. Microservice Architecture: A Performance and Scalability Evaluation," in *IEEE Access*, Vol. 10, pp. 20357-20374, 2022, DOI: 10.1109/ACCESS.2022.3152803.
- [4] Valentin Adamescu, *Analysing monolithic and microservices software architecture for SME web services/applications*, 2020.
- [5] Tesliuk, S. Bobkov, V. Ilyin, A. Novikov, A. Poyda and V. Velikhov, "Kubernetes Container Orchestration as a Framework for Flexible and Effective Scientific Data Analysis," *Ivannikov Ispras Open Conference (ISPRAS)*, Moscow, Russia, pp. 67-71, 2019, DOI: 10.1109/ISPRAS47671.2019.00016.
- [6] S. K. Mondal, R. Pan, H. M. D. Kabir, *et al.*, "Kubernetes in IT administration and serverless computing: An empirical study and research challenges," *J Supercomput*, Vol. 78, pp. 2937-2987, 2022. DOI: <https://doi.org/10.1007/s11227-021-03982-3>.
- [7] Mojtaba Shahin, Ali Babar, Muhammad, Liming Zhu, "Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices," *IEEE Access*, Vol. 5, pp. 3909-3943, 2017.
- [8] S. A. I. B. S. Arachchi and I. Perera, "Continuous Integration and Continuous Delivery Pipeline Automation for Agile Software Project Management," *Moratuwa Engineering Research Conference (MERCCon)*, Moratuwa, Sri Lanka, pp. 156-161, 2018, DOI:10.1109/MERCCon.2018.8421965.