Reducción de la fragmentación y esfuerzo manual en arquitecturas de micro-frontends mediante procesos automatizados de actualización de librerías compartidas

Nombres y apellidos

Ing. Adrian Steven De La Rosa Altahona
Código estudiantil: 201811294376

Ing. Hector Junior Martínez Cervantes
Código estudiantil: 201621280485

Trabajo de Investigación presentado como requisito para optar el título de:
Especialista en Ingeniería de Software

Tutor(es):

MsC. Sergio Vicente Jimenez Martinez

Co-tutor(es):

MsC. Jonathan Ricardo Ruiz Rangel

RESUMEN

En este estudio se propone el diseño e implementación de un pipeline automatizado en Azure DevOps para gestionar la actualización de una librería compartida dentro de una arquitectura de micro-frontend. Mediante un enfoque basado en CI/CD, se busca evaluar y delimitar el impacto de la automatización del pipeline en la reducción del tiempo de implementación de las nuevas versiones de la librería, la disminución de errores por versiones anteriores y la mejora en la estabilidad e interacción del sistema. Se establece un eje central que cumple el rol de gestor e implementador de las versiones concreta de la librería compartida, automatizando el proceso con la mínima intervención humana.

Palabras clave: Automatización en DevOps – Métodos y herramientas para CI/CD – CI/CD en micro-frontend – Gestión de dependencias – Arquitectura micro-frontend.

ABSTRACT

This study proposes the design and implementation of an automated pipeline in Azure DevOps to manage the update of a shared library within a micro-frontend architecture. Through a CI/CD based approach, we aim to evaluate and delimit the impact of the pipeline automation in the reduction of the deployment time of new library versions, the reduction of errors due to previous versions and the improvement in the stability and interaction of the system. A central axis is then defined that acts as a version manager and executor of the shared library releases, automating the process with the minimum human intervention.

Key Words: Automation in DevOps – Methods and tools for CI/CD – CI/CD in micro-frontend – Dependency management – Micro-frontend architecture.

REFERENCIAS BIBLIOGRÁFICAS

1. Antunes, F. L. (2024). Investigating Benefits and Limitations of Migrating to a Micro-Frontends Architecture. 103–113. Retrieved from https://doi.org/10.5753/sbes.2024.3303
2. Bahiense-Junior, M. R. (2024). Micro Frontend Application Shell and Module Federation Architecture Implementation and Comparison. 166–171. Retrieved from https://doi.org/10.1109/icodse63307.2024.10829917
3. Battina, D. S. (2021). The Challenges and Mitigation Strategies of Using DevOps during Software Development. International Journal of Creative Research Thoughts (IJCRT). Retrieved from http://www.ijcrt.org/papers/IJCRT2101583.pdf
4. Bogart, C. a. (2021). When and How to Make Breaking Changes: Policies and Practices in 18 Open Source Software Ecosystems. Association for Computing Machinery, 30(4). doi:10.1145/3447245
5. Borra, P. (2024). Maximizing Efficiency and Collaboration with Microsoft Azure DevOps. International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), 4(2), 556–562. Retrieved from https://doi.org/10.48175/ijarsct-18864
6. Boyer, F., Etchevers, X., de Palma, N., & Tao, X. (2018). Architecture-Based Automated Updates of Distributed Microservices. In Service-Oriented Computing (pp. 21-36). Retrieved from https://doi.org/10.1007/978-3-030-03596-9_2
7. Chatterjee, P. S. (2024). Enhancing Operational Efficiency through the Integration of CI/CD and DevOps in Software Deployment. In 2024 Sixth International Conference on Computational Intelligence and Communication Technologies (CCICT) (pp. 173-182). doi:10.1109/CCICT62777.2024.00038
8. Chen, L. (2018). Microservices: Architecting for Continuous Delivery and DevOps. 2018 IEEE International Conference on Software Architecture (ICSA), 39-397, doi: 10.1109/ICSA.2018.00013.

9. Conte, N. S. (2025). A Catalog of Micro Frontends Anti-patterns. Retrieved from https://arxiv.org/abs/2411.19472
10. Criado, M. M. (2019). A Progressive Web Application Based on Microservices Combining Geospatial Data and the Internet of Things. IEEE Access, 7, 104577-104590. Retrieved from https://api.semanticscholar.org/CorpusID:199542477
11. DORA Research Program, G. C. (2023). 2023 Accelerate State of DevOps Report. Retrieved from https://dora.dev/research/2023/dora-report/2023-dora-accelerate-state-of-devops-report.pdf
12. Emmanni, P. S. (2020). Implementing CI / CD Pipelines for Enhanced Efficiency in IT Projects. International Journal of Science and Research (IJSR), 9(9). doi:10.21275/SR24402001528
13. Fluri, J. a. (2023). Measuring the Benefits of CI/CD Practices for Database Application Development. In 2023 IEEE/ACM International Conference on Software and System Processes (ICSSP) (pp. 46-57). doi:10.1109/ICSSP59042.2023.00015
14. García González, J. R., & Sánchez Sánchez, P. (2020). Diseño teórico de la investigación: instrucciones metodológicas para el desarrollo de propuestas y proyectos de investigación científica. Información tecnológica, 31(6), 159-170. Retrieved from http://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-07642020000600159&nrm=iso
15. Gashi, E. a. (2024). The advantages of Micro-Frontend architecture for developing web application. In 2024 13th Mediterranean Conference on Embedded Computing (MECO) (pp. 1-5). doi:10.1109/MECO62516.2024.10577836
16. Gousios, J. H. (2022). Can we trust tests to automate dependency updates? A case study of Java Projects. Journal of Systems and Software, 183. doi:https://doi.org/10.1016/j.jss.2021.111097
17. He, R. a. (2023). Automating Dependency Updates in Practice: An Exploratory Study on GitHub Dependabot. IEEE Transactions on Software Engineering, 4004-4022. Retrieved from https://doi.org/10.1109/TSE.2023.3278129
18. Kasenda, R. Y. (2024). The Role and Evolution of Frontend Developers in the Software Development Industry. Jurnal Syntax Admiration, 5(11), 5191-5196. doi:https://doi.org/10.46799/jsa.v5i11.1852
19. Kičić, J. a. (2024). Dynamic Micro-Frontends. In 2024 11th International Conference on Electrical, Electronic and Computing Engineering (IcETRAN) (pp. 1-5). doi:10.1109/IcETRAN62308.2024.10645144
20. Kula, E. a. (2022). Factors Affecting On-Time Delivery in Large-Scale Agile Software Development. IEEE Transactions on Software Engineering, 48(9), 3573-3592. doi:10.1109/TSE.2021.3101192
21. Mahboob, J. a. (2021). A Kubernetes CI/CD Pipeline with Asylo as a Trusted Execution Environment Abstraction Framework. In 2021 IEEE 11th Annual

Computing and Communication Workshop and Conference (CCWC) (pp. 0529-0535). doi:10.1109/CCWC51732.2021.9376148

22. Mohayeji, H. a. (2023). Investigating the Resolution of Vulnerable Dependencies with Dependabot Security Updates. In 2023 IEEE/ACM 20th International Conference on Mining Software Repositories (MSR) (pp. 234-246). doi:10.1109/MSR59073.2023.00042
23. Moraes, F., & Affonso, F. (2024). A New Integration Approach to support the Development of Build-time Micro Frontend Architecture Applications. In Anais do XXXVIII Simpósio Brasileiro de Engenharia de Software (pp. 637-643). doi:10.5753/sbes.2024.3585
24. Mowad, A. M. (2022). Effect of Using Continuous Integration (CI) and Continuous Delivery (CD) Deployment in DevOps to reduce the Gap between Developer and Operation. In 2022 International Arab Conference on Information Technology (ACIT) (pp. 1-8). doi:10.1109/ACIT57182.2022.9994139
25. Patchkaew, P. a. (n.d.). Adaptive CI/CD: A Flexible Architecture for Software Development},. In 2024 8th International Conference on Information Technology (InCIT) (pp. 763-768). doi:10.1109/InCIT63192.2024.10810646
26. Peltonen, S. M. (2020). Motivations, Benefits, and Issues for Adopting Micro-Frontends: A Multivocal Literature Review. arXiv: Software Engineering. Retrieved from https://arxiv.org/abs/2007.00293
27. Perlin, R., Ebling, D., Maran, V., Descovi, G., & Machado, A. (2023). An Approach to Follow Microservices Principles in Frontend. IEEE 17th International Conference on Application of Information and Communication Technologies (AICT), pp. 1-6, doi: 10.1109/AICT59525.2023.10313208.
28. Raheem, A., Osilaja, A. M., Kolawole, I., & Essien, V. E. (2024). Exploring continuous integration and deployment strategies for streamlined DevOps processes in software engineering practices. World Journal Of Advanced Research and Reviews, 2813–2830. Retrieved from https://doi.org/10.30574/wjarr.2024.24.3.3988
29. S. Ferdian, T. K. (2021). Continuous Integration and Continuous Delivery Platform Development of Software Engineering and Software Project Management in Higher Education. JuTISI, 7(1). Retrieved from https://doi.org/10.28932/jutisi.v7i1.3254
30. Shriram, K. M. (2025). Engineering efficiency through CI/CD pipeline optimization. International Journal of Science and Research Archive, 908-916. doi:10.30574/ijsra.2025.14.1.0107
31. Su, C. Y. (2019). Research and Application of Micro Frontends. IOP Conference Series: Materials Science and Engineering, 490. https://doi.org/10.1088/1757-899x/490/6/062082
32. Taibi, V. L. (2019). Does migrating a monolithic system to microservices decrease the technical debt? J. Syst. Softw., 110710. Retrieved from https://api.semanticscholar.org/CorpusID:221593764
33. Ugochukwu Francis, I., Okeoma, O., Samuel Jesupelumi, O., & Uzoka, A. (2024). Revolutionizing library systems with advanced automation: A

blueprint for efficiency in academic resource management. International Journal of Scholarly Research in Multidisciplinary Studies, 019–040. Retrieved from https://doi.org/10.56781/ijsrms.2024.5.2.0045

34. Upadhyay, N. .. (2024). Micro Frontends in React.js: A Comprehensive Analysis and Implementation Guide. Indian Scientific Journal Of Research In Engineering And Management, 8(12), 1-6. Retrieved from https://doi.org/10.55041/ijsrem39990
35. Weeraddana, N. R. (2024). Dependency-Induced Waste in Continuous Integration: An Empirical Study of Unused Dependencies in the npm Ecosystem. Association for Computing Machinery. doi:10.1145/3660823
36. Wyrich, M. a. (2021). Bots Don’t Mind Waiting, Do They? Comparing the Interaction With Automatically and Manually Created Pull Requests. In 2021 IEEE/ACM Third International Workshop on Bots in Software Engineering (BotSE) (pp. 6-10). doi:10.1109/BotSE52550.2021.00009