ISSN: 2583-9888
DOI: doi.org/10.51219/JAIMLD/venkata-soma/265

URF PUBLISHERS
connect with research world

Journal of Artificial Intelligence, Machine Learning and Data Science

https://urfpublishers.com/journal/artificial-intelligence

Vol: 2 & Iss: 3

Research Article

Implementing Secure DevOps Practices in Monolithic and Microservices Architectures

Venkata Soma*

Citation: Soma V. Implementing Secure Devops Practices in Monolithic and Microservices Architectures. *J Artif Intell Mach Learn & Data Sci 2024*, 2(3), 1132-1136. DOI: doi.org/10.51219/JAIMLD/venkata-soma/265

Received: 02 July, 2024; **Accepted:** 18 July, 2024; **Published:** 20 July, 2024

***Corresponding author:** Venkata Soma, New York Mets, USA

ABSTRACT

This paper will investigate the effectiveness of secure DevOps practices in the microservice and monolithic architecture. DevOps incorporates development, security and operations to increase the software delivery quality and speed. This study aims to explore the distinctive security solutions and challenges for each architecture by focusing on the different practices and tools such as CI/CD containerization and automated security testing. This study collected secondary qualitative data. The findings show that while the monolithic architecture benefits from simpler security management, the microservice provides better flexibility and scalability.

Keywords: DevOps, DevSecOps, CircleCI, GitLab CI, Jenkins, monolithic vs. microservice architecture

1. Introduction

a) Project specification

DevOps is a combination of tools, practices and cultural philosophies which enhance the organization's ability to deliver service and application at a high velocity. It significantly improves and evolves products at a rapid pace compared to organizations utilizing traditional software infrastructure and development management processes[1]. A monolithic architecture is a conventional model of software programs that is developed as a unified unit which is independent and self-contained from other applications. A microservice architecture is an architectural method which depends on an independently deployable service. Whether it is a microservice or monolithic architecture, a better DevOps evaluation integrates best practice which suits the project's particular requirements and assists in efficient operation, deployment and development. Hence, this project aims to evaluate the effectiveness of secure DevOps in microservice and monolithic architecture.

b) Aim and objectives

Aims: The research aims to assess the effectiveness of secure DevOps in monolithic and microservice practices

b. Objectives:

- To implement the effectiveness of secure DevOps practices
- To compare the security challenges and solutions in both architecture
- To assess the challenges of implementing secure DevOps practices

c) Research questions

- **R1:** What is the effectiveness of secure DevOps practices?
- **R2:** What are the security challenges in security challenges and solutions in both architectures?
- **R3:** What are the challenges for implementing secure DevOps practices?

d) Research rationale

In this recent time, there is a critical requirement for robust security in the DevOps practice, especially in organizations that are transitioning from monolithic to microservice architecture. However, DevOps increases deployment speed and efficiency but there are some challenges that remain while integrating this practice in the sports industry[2]. Both microservice and monolithic architecture present different security challenges which necessitate a better approach. Therefore, this research focused on filling the knowledge gap by assessing the effectiveness of secure DevOps practice in both environments.

2. Literature Review

a) Research background

DevOps security is a philosophy which integrates three approaches which are security, operations and development. The primary goal of this practice is to remove any barriers that exist within IT operations and software development. However, DevOps security or DevSecOps is a series of cultural approaches or practices which bring together software development (Dev), IT operations (Ops) and security (Sec) to increase the organization's ability to deliver service and application at a high velocity with a better securement[3].

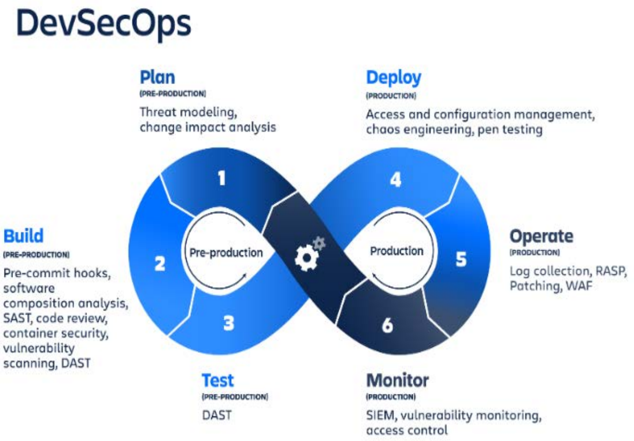
DevSecOps
Plan
(PRE-PRODUCTION)
Threat modeling, change impact analysis
Deploy
(PRODUCTION)
Access and configuration management, chaos engineering, pen testing
Build
(PRE-PRODUCTION)
Pre-commit hooks, software composition analysis, SAST, code review, container security, vulnerability scanning, DAST
Operate
(PRODUCTION)
Log collection, RASP, Patching, WAF
Test
(PRE-PRODUCTION)
DAST
Monitor
(PRODUCTION)
SIEM, vulnerability monitoring, access control
1
2
3
4
5
6
Pre-production
Production

Figure 1: Functions of DevSecOps[3].

b) Critical assessment

3. Monolithic and Microservice Architecture

A monolithic architecture is a conventional approach to designing software in which an entire application is built being an invisible and individual unit[4]. Within this architecture, all the various application components such as the data access layer, business logic and user interface get integrated and deployed together. On the other hand, microservice architecture applications are developed as a collection of independent and small services and each of them represents a particular business capability[5].

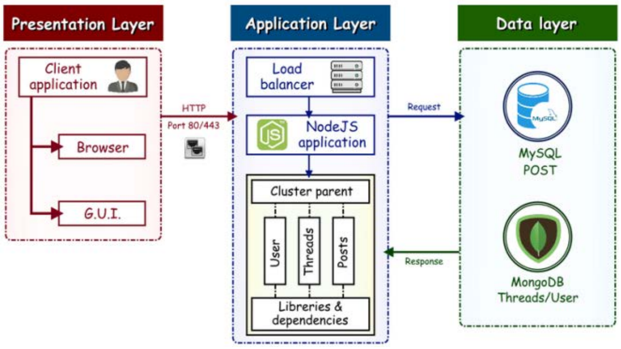

Figure 2: Monolithic system architecture[4].

4. Technologies and Tools Utilized in Secure DevOps

Secure DevOps uses a broad range of technologies and tools to ensure that security is incorporated throughout the development pipelines[6]. Continuous integration or continuous deployment (CI/CD) tools such as CircleCI, GitLab CI, and Jenkins automate the deployment and testing. Besides different configuration tools such as Puppet and Ansible effectively assist in maintaining security and consistency throughout the environment. Additionally, containerization technology such as Orchester and docker platforms such as Kubernetes facilitate secure and effective microservice management[7].

5. Impacts of DevOps Security Practices

DevSecOps mainly focuses on assuring the security aspects across the overall development process. The increased shared responsibilities and communication within the development team, security and assist to prevent critical issues cases through the conventional silo approach. It provides practice and improved security[8]. The security threats could be fixed while they would be found as early as possible. This security practice significantly provides cost-effective and quick software delivery[9]. DevOps would notably improve the sports industry by streamlining the software delivery, analytics and streamlined data management by assuring more secure and faster updates to applications. Hence, it would provide better security updates to tracking player performance, fan engagement and game statistics. This would effectively lead to better decision-making, efficient operation and better fan experiences.

c) Linkage to aim

all information in this chapter focuses on effectiveness of secure DevOps in monolithic and microservice practices. It addresses the way DevOps increased data security of a company, and the way NY Mets can use this to fetch benefits in monolithic and microservice practices. Hence, this review section is successfully linked to the research aim.

d) Literature gap

This research extensively addresses the effectiveness of security practice in DevOps, yet it lacks the in-depth exploration of different security mitigation strategies during the implementation of secure DevOps. Hence, in-depth exploration of specific security mitigation risks is the literature gap of this study. Additionally, it fails to cover the key role of different emerging technologies such as ML and AI in DevOps practices.

6. Methodology

a) Research Philosophy

A research philosophy is a particular way of beliefs about a specific way in which information or data for the research topic would be collected, utilized and analyzed[10]. It primarily deals with the source, nature and development of the knowledge. However, there are four types of research philosophies positivism, pragmatism, realism and interpretivism. In this paper, positivist research philosophy has been followed which helps in getting efficient conclusions.

b) Research approach

A research approach is mainly defined as collecting various strategies and plans which would utilize for structuring the entire research process. It significantly involves collecting, assessing and interpreting the collected data for answering research questions[11]. However, there are three research approaches which include deductive, inductive and abductive research questions. This paper adopted the inductive research approach for implementing security practices and assessing the effectiveness of these practices in different software architectures.

c) Research design

A well-structured research design within a methodology section is an efficient plan for answering all the questions. There are three types of research design which involve exploratory, explanatory and descriptive research design. This research followed the descriptive research design for significantly implementing the security practices of DevOps and evaluating the effectiveness of DevOps security practices.

d) Data collection methods

The data collection method is a research methodology that is a vital process of collecting data and necessary information through relevant sources for finding the answers to research problems[12]. Data collection methods could be divided into two main categories which are primary or quantitative data and secondary or qualitative data collection methods. primary or quantitative information is a kind of data which had not been around and mainly involves the numerical information. On the other hand, secondary data involves data that already exists and is published in Journals, research articles, magazines, newspapers and books. This paper collected the secondary data, and, in this concern, it evaluated the thematic data analysis methods.

e) Ethical consideration

During the period of data collection methods, there are several codes of conduct have been maintained. Any sort of commercial implication of taken data had been strictly avoided. All the information and data had been gathered by reliable and authentic sources such as articles, news sources and authentic journals.

7. Results

a) Critical analysis

In the sports industry, evaluating secure DevOps practices increases the efficiency and security of data management systems which is vital for managing sensitive player analytics and information. For instance, automated security checks and continuous monitoring effectively protect against data breaches which assures data integrity for performance metrics and player stats. Through securing the DevOps pipeline, the sports organization could effectively innovate and foster advanced analytics. This system's real-time data processing would help to improve player performance. Hence, the main aim of this project is to assess the key security practices and the effectiveness of these practices in different architectures.

b) Findings and Discussions

Theme 1: Application of DevOps in monolithic architecture

In the monolithic architecture, the key DevOps practice aims to secure the whole application as an individual unit, this characteristic involves extensive testing for vulnerabilities, robust access control and compatible patch management for mitigating risk throughout the integrated elements[13]. The security practice mainly involves evaluating strong access control, establishing comprehensive vulnerability testing and frequent patch management. Different security measures such as firewalls beside intrusion detection systems get applied effectively throughout the overall applications. This specific nature of the monolithic security could simply be management, yet it can further lead to bottleneck.

Theme 2: Application of DevOps in microservice architecture

Consequently, within the microservice architecture, the security practice of DevOps highlights securing each individual independent service. This includes evaluating service-specific authorization and authentication, utilizing containerization with tools such as Kubernetes and Docker for isolation and assuring a secure API[14]. Aside from this, both monitoring and security testing are vital for addressing the potential vulnerabilities in the continuous deployment pipelines and integration. It would ensure that each microservice maintains high-security standards.

Theme 3: Difference between monolithic and microservice architecture and their security system

The effectiveness of the overall secure DevOps practices significantly varies within the microservice and monolithic architecture due to their inherent structural differences[15]. In the monolithic architecture, the primary focus is on effectively securing the application being a cohesive unit. This approach could effectively simplify the entire security management as all components are incorporated and employed together. Extensive security testing and patch management could efficiently reduce the vulnerability risks. Moreover, the monolithic architecture could lead to bottlenecks as an individual vulnerability can hamper the whole system. However, the complexity of a vast monolithic application could make the security updates more heavy and slower to implement.

On the contrary, microservice architecture provides a more scalable and flexible approach towards security[16]. Each individual microservice operates individually which allows for a poor quality of security measures made for the particular requirements of each service. This isolation decreases the risks of an individual's vulnerability impacting the whole system. Tools such as Kubernetes and Docker facilitate secure orchestration and containerization increasing the entire security. Consistent integration and deployment pipeline effectively enable the automated security testing, and mitigation of vulnerabilities and ensure rapid identification. However, managing security throughout several microservices could be complex.

Theme 4: Difference between security management aspects of monolithic and microservice architecture

The monolithic architecture while effective in a specific scenario poses several challenges within the DevOps domain. In monolith architecture, deploying changes sometimes involves updating the whole application which leads to higher risks and longer deployment times[17]. The principles of DevOps highlight small and frequent releases which could be challenging with the monolithic structure. This architecture also could struggle to efficiently scale as it grows, making it much harder to adopt DevOps practices[18]. Since the monolith architecture had tightly coupled elements, a significant change in one part could impact others, making a consistent integration and delivery much more complex.

On the other hand, microservice architecture also introduces several numbers of issues. In microservice architecture, there are several services communicating throughout networks, and monitoring and managing the interaction could be complex. Besides, orchestrating several services demands robust operational support and infrastructure, which might require extra time and expertise[19]. In DevOps microservice architecture effectively breaking an application within several microservices can lead to extensive management overhead and hamper the overall performance.

c) Critical evaluation

From the findings, it had been observed that, compared to the microservice and monolithic architecture, the secure practice of DevOps provides various benefits as well as challenges. Monolithic architecture gets benefits through the clarified security management for the integrated nature of the application. This also facilitates extensive patch management and vulnerability testing. Moreover, study shows that m this can lead towards slower updates and bottlenecks. Contrarily, the microservice architecture can provide greater isolation and flexibility and decrease the impacts of each and every vulnerability besides enabling significant automated security testing by the CI/CD pipeline. However, there are still complexities in managing the security throughout both architectures.

8. Conclusion

The research outlines the effectiveness of secure DevOps practice in both microservice and monolithic architecture. While the monolithic system gets benefits through simpler security management for its unified structures. Microservice architecture provides improved flexibility and scalability, by they present complexities for managing multiple services. The study shows that the secure DevOps practice involves containerization, automated security testing and continuous integration which increase the operational efficiency and security in both architectures. Hence, understanding these differences would assist in selecting a suitable DevOps strategy.

9. Research Recommendation

Business organization must adapt their security practices to align with the architectural design. In this case, for a monolithic system, an effective focus on extensive security testing and comprehensive access control would help to mitigate the vulnerabilities in a unified application. In the case of the microservice, adapting automated testing, effective containerization and service-specific security measures can manage the distributed nature of the architecture. However, adopting tools such as Docker and Kubernetes can increase the streamline and security management.

10. Future Work

The security practice need of DevOps requires more exploration regarding the advanced security mitigation strategies and the integration of merging technologies such as machine learning and AI in secure DevOps practice[20]. Hence, effectively investigating how the technologies could increase threat detection, improve overall security and automate vulnerabilities in both microservice, and monolithic architecture would give valuable insights. Moreover, examining the effects of evolving DevOps practices and tools in real-world applications, specifically in dynamic and complex atmospheres could provide practical solutions and direct future implications.

11. References

1. J. Díaz, D. López-Fernández, J. Pérez, and Á. González-Prieto, “Why are many businesses instilling a DevOps culture into their organization?”., *Empirical Software Engineering*, vol. 26, pp.1-50, 2021.https://doi.org/10.1007/s10664-020-09919-3
2. F. Almeida, Simões, J. and S.Lopes, “Exploring the benefits of combining DevOps and agile”. *Future Internet*, Vol. 14, no. 2, pp.63, 2022.https://doi.org/10.3390/fi14020063
3. A.V. Jha, R. Teri, S. Verma, S. Tarafder, W. Bhowmik, S. Kumar Mishra, B. Appasani, A. Srinivasulu, and N. Philibert, “From theory to practice: Understanding DevOps culture and mindset”. *Cogent Engineering*, Vol. 10, no. 1, pp.2251758, 2023.https://doi.org/10.1080/23311916.2023.2251758
4. G. Blinowski, A. Ojdowska, and A. Przybyłek, “Monolithic vs. microservice architecture: A performance and scalability evaluation”. *IEEE Access*, Vol. 10, pp.20357-20374, 2022. https://doi.org/10.1109/ACCESS.2022.3152803
5. Y. Abgaz, A. McCarren, P. Elger, D. Solan, N. Lapuz, M. Bivol, G. Jackson, M. Yilmaz, J. Buckley, and P. Clarke, “Decomposition of monolith applications into microservices architectures: *A systematic review”, IEEE Transactions on Software Engineering*, vol. 49, no. 8, pp.4213-4242,2023.https://doi.org/10.1109/TSE.2023.3287297
6. C. Woody, T. Chick, A. Reffett, S. Pavetti, R. Laughlin, B. Frye, and M. Bandor, “DevSecOps Pipeline for Complex Software-Intensive Systems: Addressing Cybersecurity Challenges”, *The Journal on Systemics, Cybernetics and Informatics: JSCI*, vol. 18, no. 5, pp.31-36, 2020.https://apps.dtic.mil/sti/citations/AD1110434
7. M. Waseem, A. Ahmad, P. Liang, M.A. Akbar, A.A. Khan, I. Ahmad, M. Setälä and T. Mikkonen, “Containerization in Multi-Cloud Environment: Roles, Strategies, Challenges, and Solutions for Effective Implementation”, *arXiv preprint arXiv:2403*, pp.12980, 2024. https://doi.org/10.48550/arXiv.2403.12980
8. K. Pelluru, “Integrate security practices and compliance requirements into DevOps processes”. *MZ Computing Journal*, vol. 2, no. 2, pp.1-19, 2021.http://mzjournal.com/index.php/MZCJ/article/view/139
9. R. Desai and T.N. Nisha, “Best practices for ensuring security in devops: A case study approach. In Journal of Physics”, *Conference Series*, *IOP Publishing*, Vol. 1964, No. 4, pp. 042045. July, 2021. DOI 10.1088/1742-6596/1964/4/042045
10. Kirongo and C. Odoyo, “Research philosophy design and methodologies: A systematic review of research paradigms in information technology”, 2020.http://41.89.229.23/handle/123456789/329
11. P. Pandey and M.M. Pandey, “Research methodology tools and techniques”. *Bridge Center.* 2021. http://dspace.vnbrims.org:13000/jspui/bitstream/123456789/4666/1/RESEARCH%20METHODOLOGY%20TOOLS%20AND%20TECHNIQUES.pdf
12. R. Coe, M. Waring, L.V. Hedges and L.D. Ashley, eds., “Research methods and methodologies in education”. *Sage*, 2021.https://books.google.com/books?hl=en&lr=&id=pFMIEAAAQBAJ&oi=fnd&pg=PP1&dq=Data+collection+methods+in+research+methodology&ots=_0U18ZiPHN&sig=FBMk6-5MGtcjCexOdd-FiU--mlE
13. M. Shahin, A. Rezaei Nasab and M. Ali Babar, “A qualitative study of architectural design issues in DevOps”. *Journal of Software: Evolution and Process*, vol. 35, no. 5, pp.e2379, 2023.https://doi.org/10.1002/smr.2379
14. D. Berardi, S. Giallorenzo, J. Mauro, A. Melis, F. Montesi and M. Prandini, “Microservice security: a systematic literature review”. *PeerJ Computer Science*, vol. 8, pp.e779, 2022.https://doi.org/10.7717/peerj-cs.779

15. J. Fritzsch, J. Bogner, M. Haug, A.C. Franco da Silva, C. Rubner, M. Saft, H. Sauer and S. Wagner, “Adopting microservices and DevOps in the cyber-physical systems domain: a rapid review and case study”. *Software: Practice and Experience*, vol. 53, no 3, pp.790-810, 2023. https://doi.org/10.1002/spe.3169

16. Hannousse and S. Yahiouche, “Securing microservices and microservice architectures: A systematic mapping study”. *Computer Science Review*, vol. 41, pp.100415, 2021.https://doi.org/10.1016/j.cosrev.2021.100415

17. R.N. Rajapakse, M. Zahedi, M.A. Babar and H. Shen, “Challenges and solutions when adopting DevSecOps: A systematic review”. *Information and software technology*, vol. 141, pp.106700, 2022.https://doi.org/10.1016/j.infsof.2021.106700

18. V. Velepucha and P. Flores, “Monoliths to microservices-migration problems and challenges: A SMS”. *In 2021 Second International Conference on Information Systems and Software Technologies (ICI2ST)*, pp. 135-142. IEEE, March, 2021.https://doi.org/10.1109/ICI2ST51859.2021.00027

19. S. Baškarada, V. Nguyen and A. Koronios, “Architecting microservices: Practical opportunities and challenges”. *Journal of Computer Information Systems.* 2020.https://doi.org/10.1080/08874417.2018.1520056

20. http://mzjournal.com/index.php/MZCJ/article/view/139