

---

**| RESEARCH ARTICLE**

## **GitOps and Continuous Delivery in Financial Software: Best Practices for Efficient DevOps Pipelines**

**Ashmitha Nagraj**

*Senior Full Stack Engineer*

**Corresponding Author:** Ashmitha Nagraj, **E-mail:** [nagrajashmitha@gmail.com](mailto:nagrajashmitha@gmail.com)

---

**| ABSTRACT**

The combination of GitOps and Continuous Delivery (CD), utilizing DevOps pipelines to provide an effective model for the development of financial software applications that emphasize efficiency, security and regulatory compliance in high-risk and regulated environments is demonstrated in this paper. The empirical research provides a comprehensive review of previous literature to identify best practices for both methodologies in financial services, including Banking, Insurance and Fintech. Declarative configuration through GitOps has been emphasized to improve auditability and resilience while streamlining deployment using CD. In addition, the empirical research and theoretical frameworks have identified how security can be integrated into DevSecOps as well as legacy modernization and organizational strategies. Case studies from large financial institutions are used to demonstrate practical implementation examples of DevSecOps in Fintech to address the specific issues of data privacy and rapid innovation. Compliance and system integration challenges are also reviewed, along with several clear recommendations for the efficient design of pipelines.

**| KEYWORDS**

GitOps; Continuous Delivery; DevSecOps; Financial Software Engineering; FinTech; CI/CD Pipelines; Cloud-Native Architecture; Regulatory Compliance; Microservices Modernization; Secure Software Supply Chain

**| ARTICLE INFORMATION**

**ACCEPTED:** 01 May 2022

**PUBLISHED:** 25 May 2022

**DOI:** 10.32996/fcsai.2022.1.1.5x

---

### **Introduction**

In the Financial Sector there is a need to balance speed of delivery with the strict compliance and security requirements that are present within the financial industry. GitOps - using Git as the single authority for Declarative Infrastructure emerged in 2017 as an additional way to bring Consistency and Traceability into Operations [1], [2]. Continuous Delivery (CD), while building upon Continuous Integration (CI) delivers automated paths to Production ensuring Deployable Software states [3]. Together, they enable Efficient DevOps Pipelines for Applications in Finance, where Errors have the potential to cause both significant Financial and Reputational damage.

A deep focus on DevSecOps in FinTech will address security embedding in pipelines, particularly when dealing with PII and Blockchain Integrations [4] [5]. The major themes include Automation for Efficiency; Leadership for Adoption; Resilience against Threats [6] [7]. This study will provide actionable insights for practitioners.

### **Related Work**

Earlier research and literature focused on the application of DevOps within finance and the use of automation, and security as key components of that implementation. Prior work has demonstrated how Continuous Delivery (CD) can be integrated into

**Copyright:** © 2022 the Author(s). This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) 4.0 license (<https://creativecommons.org/licenses/by/4.0/>). Published by Al-Kindi Centre for Research and Development, London, United Kingdom.

regulated financial environments through compliant architectural patterns [8]. Other studies have presented case analyses of the transition from Continuous Integration (CI) to Continuous Deployment (CD), offering best practices for automation within enterprise pipelines [3]. Cloud-based DevOps implementations in insurance environments have also been evaluated to support continuous testing and feedback mechanisms [7].

Empirical investigations into CD practices within large financial institutions have analyzed pipeline efficiency improvements using survey-based evidence [6]. Frameworks tailored specifically for banking-sector DevOps have been proposed, incorporating monitoring, governance, and compliance considerations [9]. Additional research has identified adoption barriers and emphasized the importance of organizational knowledge sharing in DevOps transformations [10].

Challenges associated with implementing DevOps in financial cloud systems have been systematically analyzed [11], while leadership and organizational alignment have been identified as critical success factors in Agile-DevOps transformations in financial enterprises [12]. The adaptation of GitOps principles to high-stakes financial deployments has also been explored [1], along with containerization strategies designed to reduce operational downtime in e-commerce and financial environments [2].

Further research has examined CI/CD availability within FinTech ecosystems [13] and developed taxonomies of DevOps practices in financial contexts [14]. Studies have also evaluated team alignment in large financial organizations [15] and investigated resilience engineering approaches using GitOps principles [16]. Enterprise-level GitOps adoption within CI/CD pipelines has been analyzed [17], and success factors for Continuous Delivery implementations have been identified [18]. Security of financial data in cloud-native environments has been extensively studied [4], alongside migration roadmaps from legacy systems to microservices-based architectures [19]. Methods for ensuring integrity within financial pipelines have been proposed [20], and broader challenges in microservice development have been documented [21]. Additionally, Continuous Delivery applications in blockchain systems and service-oriented DevOps models for large organizations have been explored [5], [22]. This collective body of work provides information to assist in the development of secure, efficient financial pipelines.

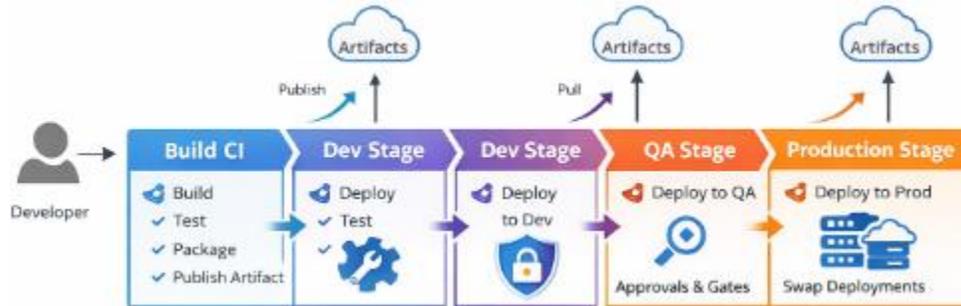


Figure 1: Continuous Releases and Continuous Deployment

### Best Practices for Efficient DevOps Pipelines

Best practices derived from the literature focus on security, modernization, integrity, and change management, clarified with structured guidelines.

### DevSecOps in FinTech

FinTech is particularly reliant on an integrated security model because of its innovative but also highly susceptible to attack applications such as mobile banking and blockchain [5] [13]. The DevSecOps security model embeds security into the entire development pipeline and shifts left compared to other traditional models of software development [4].

Defense-in-Depth, for example, is achieved by utilizing both static Application Security Testing (SAST) and Dynamic Analysis (DAST), in the CI stages [20]. With a GitOps model, versioned configurations stored in a Git repository enable automatic scanning for vulnerabilities that ensures compliance with regulatory requirements such as GDPR or PCI-DSS [16].

To implement FinTech specific practices include using Chaos Engineering to test the resiliency of the system to cyber threats, simulating failures within distributed systems [11]. A secure deployment process is maintained through pipeline integrity, preserving the consistency of deployments and preventing unauthorized tampering in high-risk environments [20].

Best Practice: Secret Management and Access Controls should be automated in the CD flow to protect PII, thereby reducing the risk of breach [4]. Clarity will be increased by clearly defined gates; code review, scan, build, test, deploy.



Figure 2: DevSecOps Pipeline

**Legacy System Modernization**

Transition Monolithic Systems into Microservice Architecture using GitOps for declarative updates [19][21]. A Hybrid Pipelines which containerize legacy components to be compatible with Continuous Deployment (CD) [2] can be created with the usage of Continuous Releases via CD to minimize disruptions to operations [3][9].

**Pipeline Integrity & Automation**

Ensure that repositories are Production-Synchronized via pull-based operators in a GitOps Environment [1]; Use Automated Testing and Monitoring for Feedback Loops [7][8]. This also provides an environment that allows for immutable deployments of applications in a Blockchain or another Fintech environment [5].

**Leadership & Organizational Change**

Provide Transformational Leadership to Foster Cultural Changes [12]; Create Standardized tools and training processes [14][18]. The implementation of service-oriented processes will facilitate Large-Scale adoption [22].

**Efficiency Metrics**

Track deployment Frequency and Recovery Time; CD has been shown to improve efficiency at financial firms by 20-30% [6][15].

Practice Area	Key Actions	Benefits
DevSecOps	Integrate SAST/DAST, chaos testing	Risk mitigation, compliance [4]
Modernization	Containerization, incremental CD	Reduced downtime [19]
Integrity	Git as truth source, automated scans	Auditability [20]
Automation	End-to-end testing, monitoring	Faster cycles [7]
Leadership	Training, tool standardization	Adoption success [12]

Table 1: Enhanced Best Practices Summary

**Case Studies**

To increase the analytical depth of this section, empirical studies are synthesized with industry case studies to detail the architectures of implementing DevOps, the operational challenges, the mitigations to these challenges, and the documented performance outcomes.

### **ING NL: Continuous Delivery in Banking**

A study that focused on the practice of Continuous Delivery (CD) and was conducted in ING NL company found that continuous integration and continuous delivery (CI/CD) processes were largely automated using toolsets, such as Jenkins, for supporting frequent releases [6]. The survey responses of over 100 DevOps engineers indicated that the main obstacles included the fragmentation of tools and cultural resistance at the organizational level. In response to the problems listed above, the bank established unified platforms to support its operations and established formal training programs to educate staff, which resulted in the bank being able to deploy software applications to production environments up to 20-30 percent faster than before [6]. Additionally, the inclusion of automated security scans into the build pipeline allowed for earlier detection of vulnerabilities in code when it is integrated into the source control system and aligned the application delivery process with the principles of DevSecOps, and enhanced overall transaction processing security.

### **Capital One: DevOps Team Alignment**

The researchers analyzed DevOps implementation at Capital One financial services company by using process measures that measured the efficiency of development and operations processes through performance metrics such as lead time and change failure rates [15]. The company used version-controlled GitOps oriented workflows for managing infrastructure to provide a stable CD practice environment. The main problems with implementing these new processes included dependencies on legacy systems, numerous integrations and security vulnerabilities in their current pipeline. These issues were resolved by incrementally migrating the legacy architecture into a microservice based architecture and adding automated compliance gates to the delivery pipeline [15]. This transformation provided an increase in the ability to innovate products especially in the credit service area and decreased operational risk by utilizing DevSecOps methods including automatic testing and controlled fault injection [16].

### **Insurance Industry: Cloud-Based Automation**

DevOps is used in the insurance industry, and there has been a successful use of DevOps within the insurance industry using cloud-based CI/CD pipeline for the entire process of software development, testing and deployment [7]. This organization was able to automate every single stage of their policy management system, therefore eliminating human error and increasing efficiency throughout the organization, while also providing real-time feedback loops to enable continuous improvement, and shorten the time it takes to react to changes in the marketplace. Although this automation was a success, it also created new security concerns for storing and processing sensitive financial information [7]. Therefore, the organization implemented encryption technology, access control technology, and ongoing monitoring as part of the CI/CD pipeline to mitigate those security concerns. Ultimately, the use of a scalable, automated CI/CD pipeline to process financial data provides additional evidence that the principles and technologies of DevOps and cloud native CI/CD can be applied across all industries, including the emerging FinTech industry.

### **Financial Systems-of-Systems: Cloud and DevOps Challenges**

A case study from a financial services firm has documented the integration of DevOps techniques as part of its migration to a cloud platform [11]. In implementing CI/CD pipelines across multiple microservices, one of the most difficult tasks was the integration of the multiple distributed services. Another challenge was how to integrate security controls into each component of the CI workflow. The company utilized a GitOps method to define and enforce the configuration of all applications and their infrastructure through declarative files, this enabled automation of testing and deployment for each application. This ultimately resulted in increased pipeline speed and increased consistency in deploying applications [11]. From the perspective of FinTech, by incorporating DevSecOps practices such as automated scanning for vulnerabilities on an ongoing basis during the development process, the company experienced increased resilience in very transaction intensive environments. Ultimately, the firm stated that it had achieved a significant increase in the reliability of the overall system and was able to quantify and measure reductions in financial loss due to outages.

### **FinTech Cross-Industry: DevOps Journey**

A recent study of the adoption of DevOps within FinTech companies looked at the transition of companies from using legacy development methods to CI/CD methods; specifically looking at how the high availability of financial payment systems could be supported in this new paradigm [13]. Through case study analysis, the greatest obstacles to this transition were related to the speed and scalability of rapidly growing systems as well as the risk of security breaches due to the nature of the payment process environment. These obstacles were mitigated by implementing CI/CD pipelines in containers and deploying comprehensive monitoring tools to increase the visibility and stability of these operational processes [13].

The next phase of the research showed how the practice of DevSecOps was integrated into the CD workflow of the company using automated security scanning at every point of the delivery lifecycle [13]. As a result, all newly developed APIs and financial products would have to adhere to the organization's standard for developing securely while still delivering those products quickly. Additionally, the research showed that by utilizing the methodology of GitOps, companies could help maintain operational consistency across various domains within an industry by providing a single source of truth through version controlled declarative configuration management.

Collectively, the overall analysis found that to successfully implement GitOps and Continuous Delivery in a FinTech environment, companies need to have both very extensive automation to enable greater efficiency [17], as well as significant organizational support and leadership to support the required culture transformation [12]. Overall, the case studies together provide a viable roadmap for how agile systems can be developed and deployed in a highly secure manner in a fast-moving financial system.

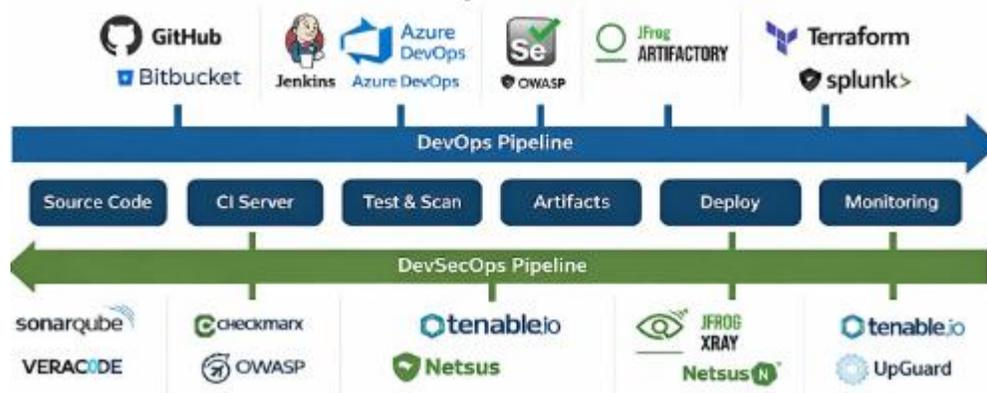


Figure 3: CI/CD Pipeline in Financial Organizations

**Challenges and Solutions**

The key challenges to adopting Financial DevOps are that there is a legacy of siloed systems, existing security vulnerabilities, and regulatory compliance [21], [4]. The solutions are Hybrid GitOps as a method for migrating from traditional systems [19] and DevSecOps with testing [16]. For FinTech the addition of blockchain creates additional complexity, thus CD (Continuous Deployment) is used for immutable deployments [5]. Leadership is required to overcome the organizational barriers to implementing Financial DevOps [12]. Phased implementation and monitoring will help to mitigate these issues [10], [22].

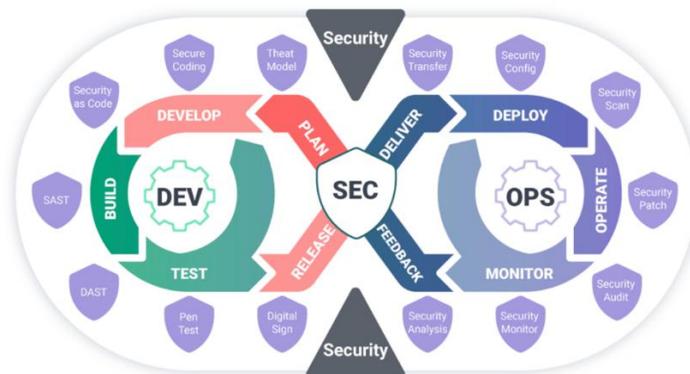


Figure 4: Secure DevOps Architecture

## Conclusion

GitOps and CD, augmented by DevSecOps, enable efficient, secure pipelines in financial software. Expanded cases and clarified practices offer practical guidance, particularly for FinTech. Future work could deepen blockchain integrations [5]. Implementing these fosters innovation and resilience.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

**Publisher's Note:** All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

## References

- [1] Enemosah, Aliyu. (2019). Implementing DevOps Pipelines to Accelerate Software Deployment in Oil and Gas Operational Technology Environments. 08. 501 - 515.
- [2] Biener, Adam & Crawford, Andrea. (2019). DevOps for Containerized Applications. 10.1007/978-3-319-94229-2\_4.
- [3] M. Virmani, "Understanding DevOps & bridging the gap from continuous integration to continuous delivery," in Proc. 5th Int. Conf. Innov. Comput. Technol. (INTECH), May 2015, pp. 78–82, doi: 10.1109/INTECH.2015.7173368.
- [4] Pankti Desai, Thaier Hamid . Best Practices for Securing Financial Data and PII in Public Cloud. International Journal of Computer Applications. 183, 40 ( Dec 2021), 1-6. DOI=10.5120/ijca2021921737.
- [5] Górski, Tomasz. (2021). Towards Continuous Deployment for Blockchain. Applied Sciences. 11. 11745. 10.3390/app112411745.
- [6] Vassallo, Carmine & Zampetti, Fiorella & Romano, Daniele & Beller, Moritz & Panichella, Annibale & Di Penta, Massimiliano & Zaidman, Andy. (2016). Continuous Delivery Practices in a Large Financial Organization. 519-528. 10.1109/ICSME.2016.72.
- [7] Soni, Mitesh. (2015). End to End Automation on Cloud with Build Pipeline: The Case for DevOps in Insurance Industry, Continuous Integration, Continuous Testing, and Continuous Delivery. 85-89. 10.1109/CCEM.2015.29.
- [8] Steffens, A., Lichter, H., & Döring, J.S. (2018). Designing a Next-Generation Continuous Software Delivery System: Concepts and Architecture. 2018 IEEE/ACM 4th International Workshop on Rapid Continuous Software Engineering (RCoSE), 1-7.
- [9] Nagarajan, Anitha & Overbeek, Sietse. (2018). A DevOps Implementation Framework for Large Agile-Based Financial Organizations: Confederated International Conferences: CoopIS, C&TC, and ODBASE 2018, Valletta, Malta, October 22-26, 2018, Proceedings, Part I. 10.1007/978-3-030-02610-3\_10.
- [10] Nagarajan, A.D., Overbeek, S.J. (2018). A DevOps Implementation Framework for Large Agile-Based Financial Organizations. In: Panetto, H., Debruyne, C., Proper, H., Ardagna, C., Roman, D., Meersman, R. (eds) On the Move to Meaningful Internet Systems. OTM 2018 Conferences. OTM 2018. Lecture Notes in Computer Science(), vol 11229. Springer, Cham. [https://doi.org/10.1007/978-3-030-02610-3\\_10](https://doi.org/10.1007/978-3-030-02610-3_10)
- [11] Carturan, Sara & Goya, Denise. (2019). Major Challenges of Systems-of-Systems with Cloud and DevOps – A Financial Experience Report. 10-17. 10.1109/SESoS/WDES.2019.00010.
- [12] Kuiper, Christopher J., "Relationship of Transformational Leadership and Organizational Change During Enterprise Agile and DevOps Initiatives In Financial Service Firms" (2019). *Doctoral Dissertations and Projects*. 2277.
- [13] Boda, Vishnu. (2020). From FinTech to Healthcare: A DevOps Journey across Industries. International Journal of AI, BigData, Computational and Management Studies. 1. 30-40. 10.63282/3050-9416.IJAIBDCMS-V112P104.
- [14] Macarthy, Ruth & Bass, Julian. (2020). An Empirical Taxonomy of DevOps in Practice. 221-228. 10.1109/SEAA51224.2020.00046.
- [15] Alluri, Venkat Rama Raju & Ahmad, Tanzeem & Pal, Dheeraj Kumar & Yellepeddi, Sai Manoj & Thota, Shashi. (2020). DevOps Project Management: Aligning Development and Operations Teams. Journal of Science & Technology. 1. 464-87.
- [16] Ramaswamy, Yogesh. (2020). Resilience Engineering in DevOps: Fault Injection and Chaos Testing for Distributed Systems. NeuroQuantology. 18. 337-347. 10.48047/nq.2020.18.12.NQ20269.
- [17] Arugula, B. (2021). Implementing DevOps and CI/CD Pipelines in Large-Scale Enterprises. *International Journal of Emerging Research in Engineering and Technology*, 2(4), 39-47.
- [18] Senapathi, Mali & Buchan, Jim & Osman, Hady. (2019). DevOps Capabilities, Practices, and Challenges: Insights from a Case Study. 10.48550/arXiv.1907.10201.
- [19] Waseem, Muhammad & Liang, Peng & Shahin, Mojtaba. (2020). A Systematic Mapping Study on Microservices Architecture in DevOps. Journal of Systems and Software. 170. 110798. 10.1016/j.jss.2020.110798.
- [20] Myrbakken, H. (2017). DevSecOps: A Multivocal Literature Review. Communications in Computer and Information Science. [https://doi.org/10.1007/978-3-319-67383-7\\_2](https://doi.org/10.1007/978-3-319-67383-7_2).
- [21] Jansen, Anton & Bosch, Jan. (2005). Software Architecture as a Set of Architectural Design Decisions. Proceedings - 5th Working IEEE/IFIP Conference on Software Architecture, WICSA 2005. 2005. 109-120. 10.1109/WICSA.2005.61.
- [22] Erich, Floris & Amrit, Chintan & Daneva, Maya. (2017). A Qualitative Study of DevOps Usage in Practice. Journal of Software: Evolution and Process. 00. 10.1002/smr.1885.