ISSN: 2754-6659

Journal of Artificial Intelligence & Cloud Computing

SCIENTIFIC Research and Community

Review Article

Open Access

Integrating Security in Cloud-Native CI/CD Pipeline: A Comprehensive Review of DevSecOps Practices

Vivek Somi

Technical Account Manager at Amazon Web Services, USA

ABSTRACT

This work explores the critical issue of security vulnerabilities in the Continuous Integration and Continuous Deployment (CI/CD) pipeline, an approach rapidly embraced in modern software development to boost speed and efficiency. While CI/CD approaches accelerate software delivery, they present a range of potential security issues that must be addressed. The paper emphasizes the importance of integrating security measures throughout the CI/CD lifecycle by employing automated testing and deployment methods in code development and repository management. The primary security issues highlighted include code defects, unstable dependencies, misconfigured environments, and challenges in securing containerized applications. Addressing these risks helps underscore the necessity for businesses to implement best practices in configuration management, conduct regular security audits, and utilize automated security testing technologies. Ensuring that security is perceived as a shared responsibility rather than a secondary concern relies on fostering a security-first culture within development and operations teams. Emerging methods and technologies aimed at enhancing security within CI/CD environments include static and dynamic application security testing (SAST and DAST), Software Composition Analysis (SCA), and Infrastructure as Code (IaC) practices. This article seeks to provide companies with a comprehensive understanding and practical guidance to establish robust security policies within their CI/CD systems. Prioritizing security in the CI/CD architecture can significantly reduce the risk of data breaches and system failures, thereby increasing customer confidence in digital products.

***Corresponding author**

Vivek Somi, Technical Account Manager at Amazon Web Services, USA.

Received: November 03, 2022; **Accepted:** November 14, 2022; **Published:** November 18, 2022

Keywords: Secure Coding, Best Practices, Vulnerability, Error Control, Input Validation

Introduction

Integration security takes front stage in the development process as cycles combining program creation and deployment racing at a unique speed in the digital age of today. Businesses today build, test, and distribute software differently as cloud-native apps with pipelines for continuous integration/continuous delivery (CI/CD) become more common. But if we are to guarantee regulatory compliance and safeguard personal data, this also presents major security concerns that demand attention. For companies trying to include security controls right into the CI/CD process, DevSecOps - a methodology consisting of security concepts in a CI/CD framework has grown even more important. Among teams working on operations, development, and security, DevSecOps marks a cultural revolution encouraging group accountability and cooperation. By including security into the Continuous Integration/Continuous Deployment (CI/CD) strategy, businesses can address vulnerabilities before development, therefore reducing the chance of security breaches or the cost of corrective action. This proactive approach ensures that, rather than as a final issue, security is natural throughout the development process. When businesses adopt cloud-native technologies usually needing dynamic environments and microservices architecture, the traditional perimeter-based security solutions are insufficient. Instead of trying to control the complexity of modern software development, a more all-encompassing, ongoing safety strategy is required. Among the particular security concerns the cloud-native environment raises are poorly specified cloud services, inadequate access constraints, and container vulnerabilities. Companies employing containers and serverless designs also must have good security policies handling the specific risks related with these technologies based on their needs. More complexity for implementing container safety measures or ensuring suitable separation between workloads is added by Kubernetes and other container orchestrating systems [1-4].

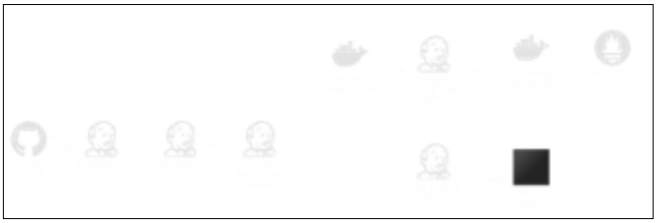

Figure 1: Cloud Native CI/CD Pipeline

Directly integrating safety measures in the CI/CD pipeline allows companies to use automation to run security audits at multiple locations, from code commits through deployment, therefore lowering the risk of exposing vulnerabilities into use. From a cloud-native standpoint, businesses ready to apply DevSecOps in a CI/CD pipeline must combine security integration technologies, tools, and approaches. Combining the two forms of security testing (DAST and SAST) techniques would help vulnerabilities to be found during pre-production. Using Infrastructure as Code (IaC) approaches also helps companies follow security guidelines

by means of automated inspections, therefore preventing misconfigurations during deployment. By allowing teams to version their security rules alongside the code of the application, using security as code lets them foster a culture of responsibility and openness. Usually, a solid DevSecOps plan relies largely on communication and cooperation. Working closely with the groups of development and operations, security teams help to guarantee that security needs are completely understood and implemented in the development process [5,6]. Quick identification and resolution for security concerns are made possible by constant surveillance and feedback loops. By including security measurements and key performance indicators, or KPIs, into the CI/CD process, businesses can assess their security posture and make well-informed decisions to enhance their security systems anchored on facts. Growing use of cloud-native technologies and the increasing complexity of cyberthreats highlight the importance of applying DevSecOps in modern software development. Businesses who overlook security aspects of their CI/CD systems risk severe vulnerabilities and possible breaches. Conversely, companies which successfully incorporate security technologies into their development strategies might foster a culture of safety consciousness, which would finally lead to more safe software releases and higher customer confidence. Not only a recommended habit but also a requirement for security to be embedded into the CI/CD pipeline in the fast-paced, threat-filled internet environment of today [7,8]. DevSecOps provides a framework for businesses to embed security policies all through the software development process, therefore promoting cooperation, automation, and ongoing improvement [9]. Using a DevSecOps methodology will be absolutely vital in maintaining the safety and resilience of your apps in a constantly changing threat environment as companies negotiate the complexity of cloud-native development [10].

Literature Review

Alanda 2022 et al. Emphasizing speed and efficiency via several tools and programming languages, the application development process has changed dramatically. Applying Continuous Integration (CI) with Continuous Delivery (CD) lets one quickly develop and distribute applications. Automated deployment of real-time apps on cloud infrastructure via the AWS Code Pipeline - source code resides on GitHub and Amazon S3; the AWS Code Pipeline tests automatic deployment. Based on results, all programming languages can be used successfully; their average deployment time is 60 seconds [11].

Paul 2021 et al. Delves into the adoption of Security-First approaches within CI/CD pipelines in cloud computing environments, focusing on the enhancement of DevSecOps practices. As organizations increasingly migrate their development operations to the cloud, the traditional DevOps model, which emphasizes speed and agility, has faced scrutiny due to its insufficient focus on security. This study argues that a shift toward a DevSecOps paradigm where security is embedded from the inception of the development process is imperative for mitigating risks associated with cloud-native applications [12].

Carroll 2021 et al. Canary deployments, green/blue installations and roll-back - before they are put into use, both ad hoc and system integration test traffic can be directed towards components. This design incorporates front-end code by means of server-side JavaScript bundle rendering. Pre-production deployments side by side with production deployments using suitable degrees of isolation in environments designed for integration testing. Pre-production components are known to operate with production exactly as they are following a successful integration test run. Test traffic employs daily copies of staging databases taken from production databases for isolation, therefore excluding sensitive information. Safety and security issues are addressed not monolithically but rather with specific attention. This design better fits agile business practices than conventional techniques; scales well with organization size, and is more effective for integration testing [13].

Ravi 2021 et al. Explores important ideas including automated compliance checks, shift-left security, and continuous security testing to fully appreciate how these ideas improve the security posture of cloud-native apps. The foundation of the DevSecOps methodology, shift-left security stresses early integration of security measures into the development lifecycle, therefore discovering and reducing vulnerabilities at the earliest phases of the application lifetime. Early approach helps to lower the complexity and cost connected with late-stage security updates. Including automated inspections in the Continuous Integration/ Continuous Development (CI/CD) process guarantees that every code commit and deployment is under careful inspection and helps to further complement this paradigm a lot more [9].

Table 1: Literature Summary

Author's name / Year	Methodology used	Problem statement	Research gap
Bernhardt/2021 [14]	Automated CI pipeline for embedded devices.	Limited wireless testing in IoT continuous integration for embedded devices.	Lack of practical CI pipelines for testing embedded IoT devices.
Sivathapandi/2021 [15]	Optimizing CI/CD pipelines for cloud-native enterprise application deployment challenges.	Challenges in optimizing CI/CD pipelines for cloud-native technologies.	Limited studies on CI/ CD pipeline optimization in cloud-native environments.
Ghimire/2020 [16]	CI/CD pipelines for cloud deployment.	Challenges in cloud-based CI/ CD pipeline deployment and optimization strategies.	Lack of simplified CI/ CD solutions for small teams in cloud.
Khan/2020 [17]	DevOps pipeline plementation on cloud.	Automating cloud DevOps pipeline with CI, CD, and monitoring.	Limited cloud-native DevOps automation integrating CI, CD, and monitoring.
Ivanov/2018 [18]	Serverless DevOps pipeline implemented using CI/ CD and monitoring practices.	Challenges in automating DevOps practices for serveries applications identified.	Limited studies on DevOps automation practices for serverless computing applications.

Key Principles of Integrating Security

Shift-Left Security

Shift-left security highlights the crucial need of implementing security techniques prior to the creation of software process, therefore changing the approach teams take to handle application security. Early-stage security guidelines should be given primary priority so that companies can find and fix problems before they become major concerns. This proactive approach assists development teams to control prospective dangers, therefore reducing risk by means of regular security reviews during the planning and development phases. Under this method, Static Application Security Testing (SAST) becomes rather relevant whenever one evaluates the code of vulnerabilities as they become written. Teams improve the general security postures for their apps by embedding security in the development process, therefore lowering the costs associated with late-stage repairs and updates. This intentional change guarantees that security is seen as a basic component of the quality of software, hence fostering a culture of security consciousness among developers and hence producing increasingly strong and safe products over time [19].

Continuous Security Testing

Strong application security mainly concentrates on adding constant security evaluations all around based on the cycle of continuous integration and continuous deployment. Safety checks ensure that every code change passes comprehensive testing prior to deployment from every pipeline point of view. Using automated methodologies, teams can evaluate efforts in real time as Interactive Application Security Testing (IAST) or Dynamic Application Security Testing (DAST), therefore allowing the detection of vulnerabilities and compliance issues as they grow. Constant security monitoring and review help to lower the possibility of using vulnerable code. This proactive approach not only improves the general safety posture of apps but also increases their resistance against different threats, thereby offering an additional safe software development environment which delivers security precedence alongside quick delivery [20].

Automated Compliance Checks

Maintaining compliance calls for the use of automated techniques verifying adherence to legal criteria or organizational security standards. By including automated compliance inspections in the Continuous Integration or Continuous Deployment (CI/CD) process, businesses can track security setups and processes in real time against set criteria including the General Data Protection Regulation (GDPR) and the Consumer Privacy Act of the State of California (CCPA). Regular infrastructure or code compliance audits enable these solutions to automatically detect any deviations or flaws that might surface during deployment and development. Also, frequent compliance audits enable businesses to promptly address any issues, therefore lowering their risk of fines. Apart from promoting adherence to rules, this proactive strategy helps the company to foster a culture for responsibility and accountability, therefore enhancing the general security posture or authenticity of the software development process [21].

Threat Modeling

Secure applications are defined early in the design process via early security risk and vulnerability assessments. By closely reviewing the architecture and components of the application through threat modelling sessions, development teams can find any attack paths prior to the project starting point. Early on application of required security rules and design changes made possible by proactively assessing risks in the early stage of application development. By means of specific tools or threat modelling systems, teams can find vulnerabilities and grade them based on their impact or probability. Dealing with these issues during the design phase not only considerably reduces dangers but also helps to minimize the labor involved to perform audits after application development. Including security concerns into the initial design process enables businesses to build a team with a culture of security awareness that benefits all stages of software development [22].

Secure Coding Practices

By following safe coding standards and supporting a culture stressing best practices, developers can reduce the likelihood of security issues during the development process. Comprehensive training courses covering safe coding techniques including effective handling of errors, comprehensive input validation, and effective authentication will teach engineers about responsible coding practices. Establishing security-oriented coding standards would help development teams regularly solve problems all through the coding process. Including automated security tools and code review techniques also helps teams find such problems early, before the code is put into use. Organizations enable developers to give security top importance by encouraging a culture of security consciousness and responsibility. This proactive strategy supports the creation of more resilient and safe applications. Encouragement of a security-centric atmosphere guarantees that security best practices are ingrained in every stage of the development life, therefore drastically lowering the chance of security breaches [23].

The Importance of Security in CI/CD

Strengthening Security Against Vulnerabilities

Minimizing vulnerabilities in implemented applications depends on including security into the CI/CD pipeline. Including security safeguards at every level helps businesses actively identify and manage potential hazards from first creation of code until deployment, therefore preventing significant consequences. This approach helps to find vulnerabilities in code as it is written by means of automated safety testing - static and dynamic analysis. Frequent security audits and feedback loops also enable teams to always enhance their security practices. By ensuring security is a basic component all through the construction process, organizations may greatly lower risks while improving the entire safety posture with their applications before they impact production environments [24].

Enhancing Compliance and Governance

Maintaining compliance to both internal organizational standards and legal criteria requires adding security to the CI/CD process. Automated inspections of compliance all throughout the pipeline enable businesses to monitor their infrastructure and applications for conformity to approved security criteria. Being proactive helps one lower the likelihood of non-compliance fees, which may seriously damage reputation and finances. Moreover, it promotes responsibility and governance across the development of operational teams. Encouragement of openness and teamwork helps companies create a more compliant safe workplace, therefore strengthening their total security posture [25].

Building Trust and Reliability

Strong security practices applied throughout the CI/CD pipeline are essential for building confidence or dependability among users, clients, and stakeholders. Companies that consistently show a commitment to security by way of well-defined rules

considerably enhance their market reputation. This commitment towards security ensures stakeholders that the business first gives priority to the protection of sensitive data. Clients hope for the company's ability to stop operational interruptions and data leakage when they witness proactive security measures in place. This confidence not only raises consumer loyalty but also attracts potential clients [26].

Supporting Continuous Improvement

Including safety within the CI/CD design promotes a continuous evolution in security policy. As they actively find flaws and review security incidents, development teams get a lot of knowledge that impacts their attitude to risk management. By means of this iterative process, teams can enhance tools, reinforce their security systems, and establish standards depending on actual experience. Companies can build an ecosystem whereby learnt knowledge is distributed and used to increase the flexibility of their products against changing risk. This proactive strategy over time builds a more strong security posture that helps teams to properly protect their assets and shift with times to meet new demands [27].

Identifying Security Risks in the CI/CD Pipeline

Understanding security concerns in the CI/CD pipeline allows one to maintain dependability and integrity for activities in the field of software development. Acting as a blend of Continuous Integration or Continuous Development, CI/CD automates software production, therefore allowing developers to frequently and consistently provide code updates. On the other hand, if not properly regulated, this automation could expose different security flaws. One of the main risks is code repositories. Among the vulnerable coding techniques these repositories could be prone to, improper authentication and input validation, therefore creating vulnerabilities like SQL injection and cross-site scripting (XSS). Relying on external libraries increases risk because outdated or poorly maintained source code can contain known vulnerabilities that attackers might exploit. Frequent code repository checks will help to find and resolve these problems, therefore guaranteeing the safe coding standards.

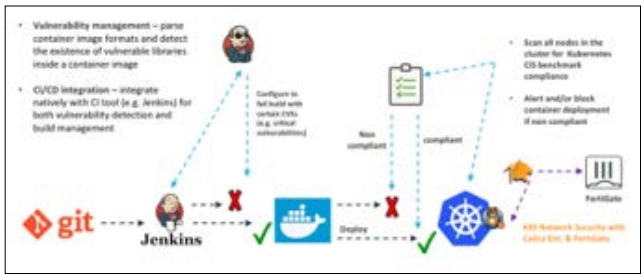

Figure 2: Security for all Stages of a CI/CD Pipeline [28].

Deployment stages are overlooked or carried out wrongly. Security methods should be included into automated systems to identify harmful code and vulnerabilities before they are used.

Inadequate security allows undetectable defects into the production line, may lead to major consequences including data leaks or system breakdowns [29,30]. Moreover, the use of obsolete or inadequate security testing techniques can impede threat detection, hence allowing the deployment of compromised or defective programs. Every CI/CD level of thorough security inspection must be done if we wish to sufficiently reduce these risks. Moreover, configuration control decides whether the CI/CD pipeline has secured stages. Two situations that might expose readily available vulnerabilities are unsecured server setups or incorrect privileges. For example, too liberal access limits can reveal sensitive information or provide illegal access to important systems [31]. Through constant configuration audits and safe setting implementation, one can help to lower these risks and guarantee that surroundings remain secure all during the development life. Containers might bring new issues even if they have benefits like scalability and isolation. For example, incorrect defaults or set container images can expose likely defects [32]. By regularly scanning for vulnerabilities using reliable base images and implementing runtime security measures to monitor unusual activities, companies prioritize the security of their container images [33]. Apart from these technical challenges, organizational culture mostly determines identification and elimination of security vulnerabilities in the CI/CD system. Rooted in a security-first perspective, encouragement of cooperation and communication should help operations, security teams, plus success all around. Including security criteria into the CI/CD flow helps businesses to guarantee that security is not only a fundamental component from the development process but also not a secondary problem [21,31,32]. This method not only points up early hazards but also encourages a responsible culture in the development team. If the development team of companies are to remain current on the most recent security concerns and best practices, they must be continuously educated on security best practices. Security awareness initiatives, seminars, and regular training courses enable teams to spot prospective hazards and implement reasonable defenses. Giving security top focus all through the CI/CD process would help companies greatly lower their attack surface and improve the general safety posture for their software development systems [33].

Case Studies and Real-World Implementations

Financial Services Organization

By including automated safety testing into its CI/CD process, a large financial services company greatly improved its security posture to satisfy strict regulatory standards. Early phases in development life vulnerabilities were found by the company applying either dynamic or statice application security testing techniques. By reducing the average time needed to handle problems, this proactive strategy permitted faster deployment without compromising security. The company therefore not only streamlined its expansion process but also enhanced its regulatory compliance, so ensuring a robust security basis for its usage [34].

E-Commerce Platform

An e-commerce platform, security issues in its online apps caused great challenges that compromised not only the business but also its customers. To address these problems, the platform built a comprehensive security plan involving Software Composition Analysis (SCA) and automated compliance checks.

The corporation uncovered flaws in external libraries using SCA and promptly addressed them. This proactive approach enhanced consumer trust in the platform's commitment to safeguard private data in addition to its general security posture. User activity in the e-commerce platform therefore increased noticeably, reflecting the positive impact of robust security measures on customer confidence and loyalty[35,36].

Healthcare Provider

Aware of the pressing need to address data privacy and compliance concerns, a healthcare provider decided to apply DevSecOps approach to enhance its security systems.

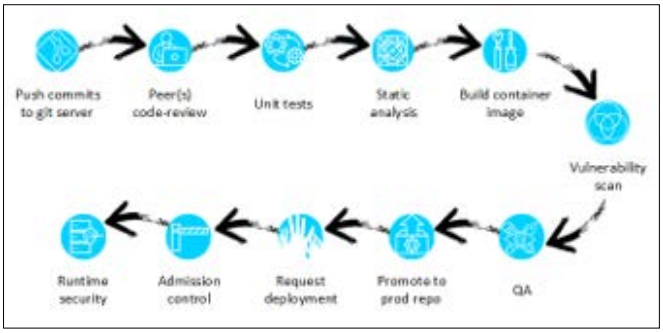

Figure 3: Modern CI/CD workflow [37].

Including security tools right into its Continuous Integration/ Continuous Development (CI/CD) process, the company made sure every code modification passed extensive security testing.

Early in the building process, this proactive approach helped them to find and fix anticipated problems and ensured compliance with Health Insurance Protection & Accountability Act (HIPAA) rules. This not only lowered the possibility of data leaks but also improved their whole security system, thereby improving their power to safeguard patient data [38].

Conclusion

In essence, one must first uncover security concerns in the CI/CD pipeline if we want to safeguard the development of software processes against an ever more complicated threat environment. When businesses embrace agile techniques and embrace automation, the likelihood of vulnerabilities appearing at various stages of development cycle becomes evident. From code development and repository management to development and deployment methods, this stresses the significance of implementing security measures at every level of the CI/CD process. Regular audits and automated security testing technologies help companies to aggressively find and fix vulnerabilities before they find their way into production. Since misconfigurations can result in exploitable vulnerabilities, configuration management and the security of containerized systems must get special attention. Encouragement of a security-first culture in development and operations teams improves cooperation and responsibility, therefore guaranteeing that security is a shared duty rather than a side effect. Awareness campaigns and ongoing education enable team members to remain current with changing risks and best practices, therefore promoting an environment where security takes the front stage. Along with lowering data breach and system failure risk, a complete approach to security in the CI/CD pipeline strengthens customer confidence in software solutions. Organizations can improve their resilience against attacks and fulfil their objectives of fast, dependable software delivery by giving security top priority as a natural component of the development process. Emphasizing security throughout the CI/CD pipeline is a critical investment that safeguards both the organization's assets and its reputation in a competitive marketplace.

References

1. Gabriela P, Kaley T, Parsons CLM (2021) Effect of increased milking frequency during early lactation on bovine mammary epithelial cell differentiation. ADSA 2021 Annual Meeting Abstracts 104.
2. Lenarduzzi V, Nikkola V, Saarimäki N, Taibi D (2021) Does code quality affect pull request acceptance ? An empirical study. J Syst Softw 171: 110806.
3. Vadavalasa RM (2021) End to end CI / CD pipeline for Machine Learning. International Journal of Advance Research, Ideas and Innovations in Technology https://www.researchgate.net/publication/351022405_End_to_end_CICD_pipeline_for_Machine_Learning.
4. Gajbhiye B, Jain A, Goel O (2021) Integrating AI-Based Security Into CI/CD Pipelines. International Journal of Creative Research Thoughts (IJCRT) 9: 6203-6214.
5. Islavath N (2020) Transitioning to the Cloud : A Devops Roadmap for Migrating Legacy. IJITMIS 11: 38-44.
6. Al Kiswani, Jalal Hasan Ahmed (2019) Smart-Cloud : A Framework for Cloud Native Applications Development. University of Nevada, Reno ProQuest Dissertations & Theses https://www.proquest.com/openview/bd6b7f4e1da585bb9fb255bb4782f9c9/1?pq-origsite=gscholar&cbl=18750&diss=y.
7. Singh A, Mansotra V (2021) A Comparison on Continuous Integration and Continuous Deployment (CI / CD) on Cloud Based on Various Deployment and Testing Strategie. International Journal for Research in Applied Science and Engineering Technology https://www.academia.edu/96158637/A_Comparison_on_Continuous_Integration_and_Continuous_Deployment_CI_CD_on_Cloud_Based_on_Various_Deployment_and_Testing_Strategies.
8. Trantzas K, Tranoris C, Gallego-madrid J, Hermosilla A (2021) An automated CI / CD process for testing and deployment of Network Applications over 5G infrastructure. 2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom) https://ieeexplore.ieee.org/document/9647628.
9. Amit KR, Venkat Rama RA, Shashi T, Chetan SR,Venkata Sri M B (2021) DevSecOps : Integrating Security into the DevOps Pipeline for Cloud- Native Applications. Journal of Artificial Intelligence Research and Applications 1: 89-114.
10. Sahni V (2020) Continuous Security: Investigation of the DevOps Approach to Security. M.Sc. thesis, School of Computing, National College of Ireland, Dublin, Ireland https://norma.ncirl.ie/4555/1/conordeegan.pdf.
11. Alanda A, Mooduto HA, Hadelina R (2022) Continuous Integration and Continuous Deployment (CI / CD) for Web Applications on Cloud Infrastructures. JITCE 2: 50-55.
12. Paul D, Rajalakshmi S, Gowrisankar K (2021) Security-First Approaches to CI / CD in Cloud-Computing Platforms : Enhancing DevSecOps Practices. Australian Journal of Machine Learning Research & Applications 1: 184-225.
13. Jeremy JC, Pankaj A, David G (2021) Preproduction Deploys : Cloud-Native Integration Testing. 2021 IEEE Cloud Summit (Cloud Summit) https://ieeexplore.ieee.org/document/9658870.
14. Bernhardt AJ (2021) CI / CD Pipeline from Android to Embedded Devices with end-to- end testing based on Containers. Master's Programme, ICT Innovation https://www.diva-portal.org/smash/get/diva2:1618170/FULLTEXT01.pdf.
15. Sivathapandi P, Care H, S Corporation () Optimization of CI / CD Pipelines in Enterprise Environments : A Comparative Analysis of Deployment Strategies. 2: 228-274.
16. Ghimire R (2020) Deploying Software in the Cloud with CI / CD Pipelines. Business Information Technology https://www.theseus.fi/bitstream/handle/10024/345618/Thesis_Ramesh_Ghimire_1.pdf?sequence=2.
17. Khan MO (2020) Fast Delivery, Continuously Build, Testing and Deployment with DevOps Pipeline Techniques on Cloud. Indian J Sci Technol 13: 552-575.

18. Ivanov V (2018) Implementation of DevOps pipeline for Serverless Applications. Aalto University https://aaltodoc.aalto.fi/server/api/core/bitstreams/72a9ce0d-9ab4-45eb-b3ff-b3d1faac6d27/content.
19. Gonzalez D, Perez PP (2021) Barriers to Shift-Left Security : The Unique Pain Points of Writing Automated Tests Involving Security Controls. ESEM '21: Proceedings of the 15th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM) 1-12.
20. Koskela P (2021) Automated Security Testing Utilizing Continuous Integration and Continuous Delivery Technologies. JAMK University https://www.theseus.fi/bitstream/handle/10024/502952/Opinnaytetyto_Koskela_Pyry.pdf?sequence=2&isAllowed=y.
21. Aghera S, Researcher I (2021) Securing CI / CD Pipelines Using Automated Endpoint Security. Journal of Basic Science And Engineering 18: 168-180.
22. Anand P, Singh Y, Selwal A, Alazab M, Tanwar S, et al. (2020) IoT vulnerability assessment for sustainable computing: Threats, current solutions, and open challenges. IEEE Access 8: 168825-168853.
23. Mooghala S (2021) Advanced Secure Coding Methodologies for Payment Application Development. IJSR 10: 1727-1730.
24. Porter BSW, Iis R (2020) Assessing the Operational System Risk Imposed by the Infrastructure Deployment Pipeline Workflow. CSIAC.
25. Paul D (2022) Cloud-Native AI / ML Pipelines : Best Practices for Continuous Integration , Deployment , and Monitoring in Enterprise Applications. JAIR 2: 176-229.
26. Mamusung R, Andriani K, Nimran U, Candidate D, Brawijaya U, et al. (2019) Building Customer Loyalty through Service Quality and Customer Trust. 22: 267-273.
27. Muñoz A, Farao A, Ryan J, Correia C, Xenakis C (2021) P2ISE : Preserving Project Integrity in CI / CD Based on Secure Elements. Information 1-19.
28. (2022) Enable CI/CD Pipeline Security with DevSecOps. Fortinet Blog https://www.fortinet.com/blog/business-and-technology/ensuring-continuous-security-integration-for-devsecops.
29. Arnold B (2020) Detecting Software Security Vulnerability during an Agile Development by Testing the Changes to the Security Posture of Software Systems. 2020 International Conference on Computational Science and Computational Intelligence (CSCI) 1743-1748.
30. Koopman M (2019) A Framework for Detecting and Preventing Security Vulnerabilities in Continuous Integration/Continuous Delivery pipelines. University of Twente Student Theses https://essay.utwente.nl/78048/.
31. Saarenp J (2020) Creating an Azure CI/CD pipeline for a React web application. Laurea https://www.theseus.fi/bitstream/handle/10024/353351/Saarenpaa_Joonas.pdf.
32. Jammeh B (2020) DevSecOps : Security Expertise a Key to Automated Testing in CI / CD Pipeline. ResearchGate 3-7.
33. Voruganti KK (2021) Enhancing Cloud Security Posture through Threat Modeling and Risk Assessment Migration. Journal of Technological Innovation 2.
34. Pölöskei I (2021) MLOps approach in the cloud-native data pipeline design. Acta Technica Jaurinensis 15: 1-6.
35. Liu C, Huang T, Id PH, Huang J (2020) Machine learning-based e-commerce platform repurchase customer prediction model. Plos One 1-15.
36. Trong L, Tran T (2020) Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID- 19 . The COVID-19 resource centre is hosted on Elsevier Connect , the company ' s public news and information. Environmental Research https://www.binasss.sa.cr/feb22/56.pdf.
37. John Kinsella (2022) What Modern CI/CD Should Look Like. Theresnomon https://theresnomon.co/what-modern-ci-cd-should-look-like-e6f50594c2d2.
38. Liying J, Qingyue M, Anthony S, Beibei Y, Lu Z (2021) Payment methods for healthcare providers working in outpatient healthcare settings. Cochrane Library https://www.cochranelibrary.com/cdsr/doi/10.1002/14651858.CD011865.pub2/full.