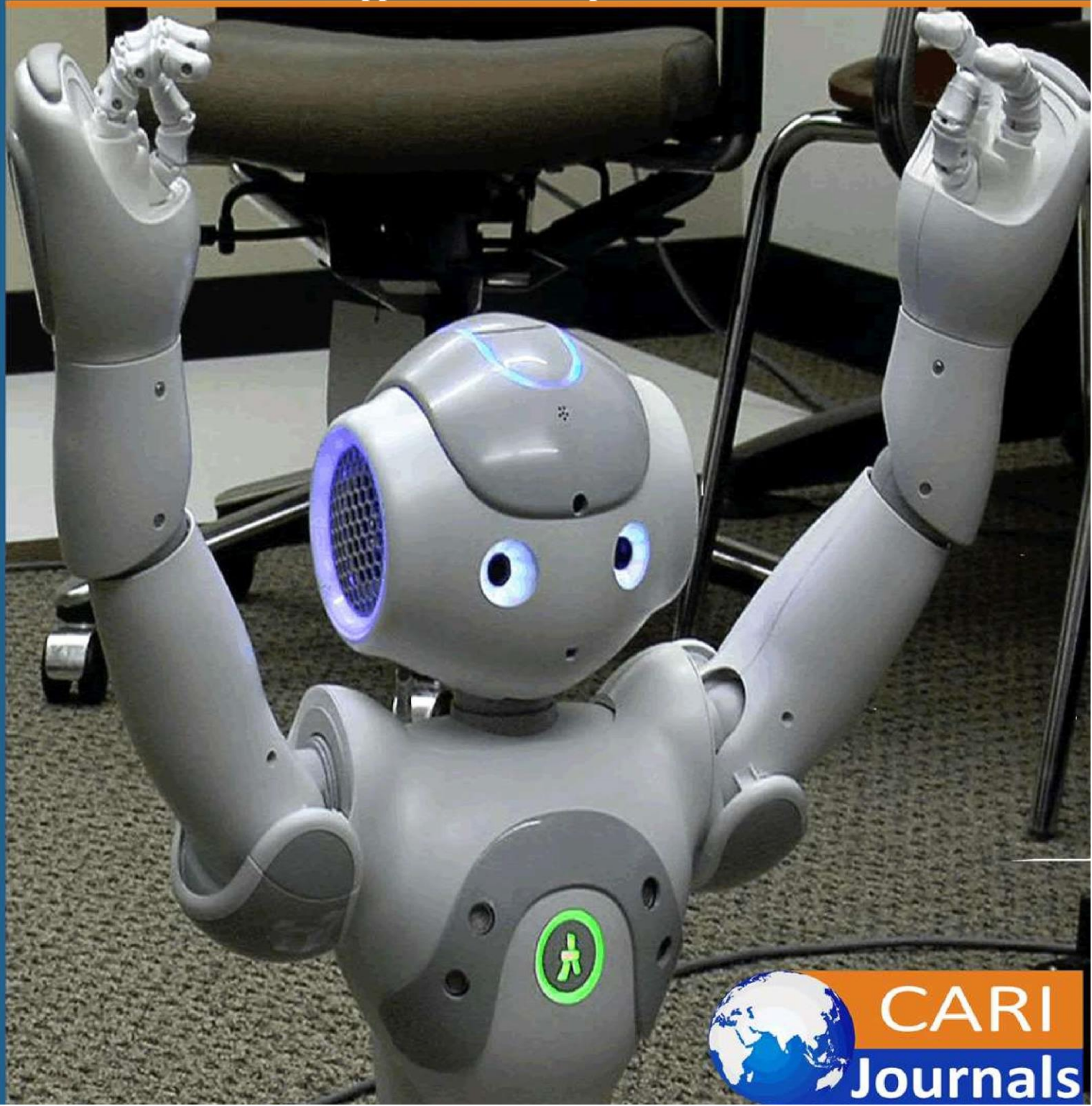
International Journal of
Computing and
Engineering
(IJCE)

DevSecOps into Multi-Cloud Environments for Resilient Application Development



CARI Journals

International Journal of Computing and Engineering
ISSN 2958-7425 (online)
Vol. 5, Issue No. 2, pp. 1 - 14, 2022
www.carijournals

DevSecOps into Multi-Cloud Environments for Resilient Application Development

Rajesh Nadipalli
Xtramile Soft LLC
https://orcid.org/0009-0009-4895-4245

Accepted: 29th April, 2022, Received in Revised Form:22nd May, 2022, Published: 18th June, 2022

Abstract

The adoption of multi-cloud strategies presents significant opportunities for enhanced resilience, flexibility, and cost optimization. It also introduces substantial security and operational complexities. This article explores the integration of DevSecOps principles and practices into multi-cloud environments to foster resilient application development. I argue that a cohesive DevSecOps framework, tailored for multi-cloud intricacies, is essential for automating security, ensuring compliance, and enabling rapid, reliable application delivery. This paper examines the key challenges of managing security across disparate cloud platforms, including fragmented security controls, inconsistent identity and access management (IAM), and complex threat landscapes. I propose a unified DevSecOps pipeline that embeds security throughout the entire software development lifecycle (SDLC), from code inception to deployment and runtime. Key components of this framework include centralized security management, automated policy-as-code, continuous monitoring, and incident response across all cloud providers. Through a review of current literature and case studies, this article demonstrates how integrating DevSecOps in multi-cloud settings can significantly improve application resilience, reduce vulnerabilities, and enhance overall security posture. I conclude by offering a set of best practices and a strategic roadmap for organizations seeking to implement a successful and scalable DevSecOps model in their multi-cloud architecture.

Keywords *- DevSecOps, Multi-Cloud, Cloud Security, Application Resilience, CI/CD Security, Infrastructure as Code (IaC), Policy as Code, Automation, Cybersecurity*

JEL Codes *- O32, O33, L86, C61, D83*

1. Introduction

The paradigm of cloud computing has shifted from a single-provider strategy to a multi-cloud approach, with organizations increasingly distributing workloads across multiple cloud environments to enhance resilience, prevent vendor lock-in, and optimize costs [1]. A 2021 survey revealed that over 92% of enterprises have adopted a multi-cloud strategy [2]. While this distribution of services offers significant benefits in flexibility and availability, it concurrently introduces formidable security and operational complexities. Managing disparate security controls, ensuring consistent policy enforcement, and maintaining visibility across heterogeneous platforms create significant hurdles for security and operations teams [3]. This fragmentation of security posture directly undermines the development of resilient applications. Traditional, siloed security models are ill-equipped to handle the dynamic and ephemeral nature of cloud-native services, leading to an expanded attack surface and increased risk of vulnerabilities. The lack of an integrated security strategy results in slower development cycles and a reactive, rather than proactive, security stance.

To address these challenges, this article argues that the systematic integration of DevSecOps principles into the multi-cloud software development lifecycle is paramount for creating a secure, resilient, and efficient application ecosystem. By embedding automated security controls and collaborative practices throughout the development pipeline from code to cloud organizations can effectively manage risks in complex multi-cloud architectures. This paper will analyze the unique security challenges inherent in multi-cloud environments and propose a tailored DevSecOps framework designed to build and maintain resilient applications, thereby enabling organizations to leverage the full potential of their multi-cloud strategy securely.

2. Literature Review

The Evolution of DevOps and DevSecOps

The software development landscape has undergone a significant transformation over the last two decades, moving away from rigid, sequential models like Waterfall towards more agile and collaborative frameworks. The DevOps movement emerged as a cultural and professional practice aiming to break down the silos between development (Dev) and operations (Ops) teams. This integration sought to shorten the systems development life cycle while delivering features, fixes, and updates in close alignment with business objectives [4]. The core tenets of DevOps automation, continuous integration, and continuous delivery (CI/CD) were proven to accelerate time-to-market and improve software quality.

This includes practices like static application security testing (SAST) during coding, software composition analysis (SCA) for open-source vulnerabilities at build time, and dynamic application security testing (DAST) in testing environments. The goal is not to burden developers with the full scope of security, but to empower them with the tools and knowledge to

make secure coding a shared responsibility, thereby creating a more robust and secure product without sacrificing speed [6].

The Multi-Cloud Paradigm

The adoption of multi-cloud architectures, wherein an organization utilizes services from two or more cloud computing providers, has become a standard enterprise strategy. The primary drivers for this trend are compelling: achieving higher resilience and availability by mitigating the risks of a single provider outage, avoiding vendor lock-in to gain negotiating power and flexibility, and optimizing costs by selecting the most competitive service for a specific workload [7]. Multi-cloud strategies allow organizations to meet diverse regulatory and data sovereignty requirements by deploying applications in specific geographic regions.

Security in Multi-Cloud Environments

Despite its advantages, the multi-cloud paradigm introduces significant security challenges that are extensively documented in the literature. Each CSP such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP) offers a unique set of security tools, APIs, and configuration semantics. This heterogeneity complicates the enforcement of consistent security policies and makes it difficult to achieve a holistic view of the organization's security posture [8].

Research has also focused heavily on the complexities of Identity and Access Management (IAM) in multi-cloud settings. Managing distinct identity systems and permission models for each CSP increases the administrative overhead and the risk of misconfiguration, which remains a leading cause of data breaches in the cloud [9]. To address these issues, solutions like Cloud Security Posture Management (CSPM) have emerged, aiming to provide continuous compliance monitoring and misconfiguration detection across multiple clouds. These tools often operate at the infrastructure level and may not be fully integrated into the application development lifecycle, representing a reactive rather than a proactive security measure [10].

3. Challenges of Securing Multi-Cloud Environments

The strategic adoption of multi-cloud architectures, while beneficial for resilience and flexibility, inherently introduces a set of complex security challenges. These challenges stem from the operational disparities between cloud service providers (CSPs) and the expanded threat landscape that such environments create. Effectively securing applications requires a deep understanding of these specific hurdles.

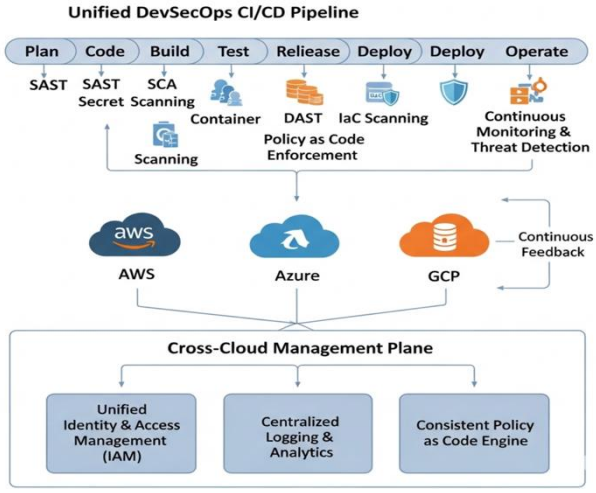


Figure 1. DevSecOps Framework for Multi-Cloud Security

Fragmented Security Controls and Visibility

Each CSP provides a proprietary suite of security tools, services, and application programming interfaces (APIs). This technological diversity results in a fragmented security landscape where policies and configurations are not directly portable. Security teams are compelled to master multiple toolsets and dashboards, making it exceedingly difficult to establish and maintain a unified security posture [11]. This lack of a single pane of glass for security monitoring and management creates blind spots, which can delay the detection of and response to security incidents.

Inconsistent Identity and Access Management (IAM)

Identity is a critical security perimeter in the cloud, yet managing it across multiple providers is a significant challenge. Each CSP has a distinct IAM framework with different definitions for roles, permissions, and groups. Attempting to manually replicate policies across providers is not only inefficient but also prone to human error, which can lead to misconfigurations and unauthorized access [9]. While federated identity solutions can centralize user authentication, they do not fully resolve the problem of authorizing access to specific cloud resources, which remains provider-dependent. This inconsistency complicates the enforcement of the principle of least privilege and increases the risk of privilege escalation attacks [12].

Regulatory and Compliance Complexities

Operating across multiple clouds often means operating across multiple geographic and legal jurisdictions, each with its own data protection and privacy regulations, such as the General Data Protection Regulation (GDPR) in Europe or the Health Insurance Portability and Accountability Act (HIPAA) in the United States. Ensuring and demonstrating compliance becomes profoundly complex when data is stored and processed across these boundaries [13]. Organizations must

navigate a tangled web of legal requirements, which dictates where certain data can reside and who can access it.

Increased Attack Surface and Sophisticated Threats

A distributed infrastructure naturally presents a broader attack surface. Every new cloud environment, network connection, and API endpoint is a potential entry point for malicious actors. The complexity of securing the interconnectivity between different cloud providers, as well as between clouds and on-premises data centers, introduces new risks that are not present in single-provider architectures [14]. Attackers can exploit inconsistencies in security configurations between providers or target the complex supply chain of services used to build modern applications. This expanded perimeter demands a more sophisticated and proactive threat detection strategy that can correlate events from disparate sources to identify coordinated attacks.

Toolchain and Automation Sprawl

While DevSecOps relies heavily on a streamlined and automated toolchain, achieving this in a multi-cloud context is challenging. Security tools are often designed with a specific CSP in mind, leading to "tool sprawl" where organizations must purchase, integrate, and maintain a separate set of security solutions for each cloud platform. The lack of interoperability between tools makes it difficult to build a truly seamless and abstracted DevSecOps workflow, hindering the efficiency and scalability of secure application delivery.

4. A Proposed DevSecOps Framework for Multi-Cloud Resilience

To overcome the challenges outlined in the previous section, a structured and intentional approach is required. I propose a DevSecOps framework specifically architected for the complexities of multi-cloud environments. This framework is not merely a collection of tools, but a strategic integration of principles and technological components designed to embed security and resilience throughout the entire application lifecycle. The primary objective is to create a system where secure and resilient application delivery is the default, automated outcome, regardless of the underlying cloud provider.

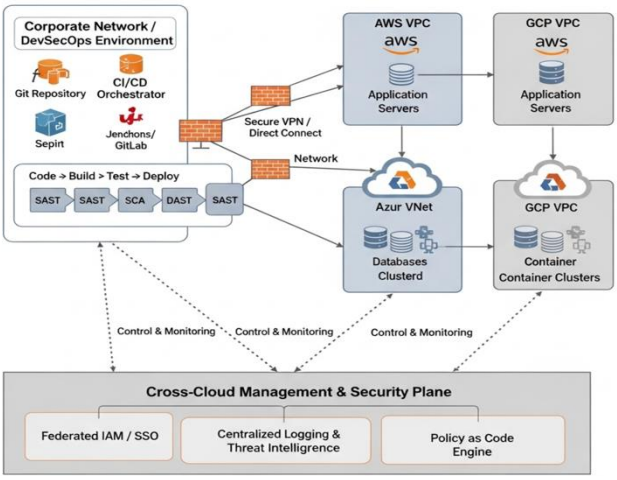


Figure 2. DevSecOps Framework for Multi-Cloud Resilience

Core Principles

Security as Code (SaC)

This principle mandates that all security configurations including network policies, access controls, and infrastructure definitions are defined, stored, and managed as code in a version control system [15]. By treating security as code, organizations can automate its deployment, apply versioning, and conduct peer reviews and automated testing on security policies just as they would with application code. This approach ensures that security measures are consistent, repeatable, and auditable across all cloud environments, significantly reducing the risk of manual misconfiguration.

Pervasive Automation

Automation is the engine of DevSecOps. In a multi-cloud context, it is essential for applying security controls consistently and at scale. This framework advocates for the automation of security scanning (SAST, DAST, SCA), compliance checks, and policy enforcement within the CI/CD pipeline [6]. Automation eliminates the need for manual intervention, which is often slow and error-prone, thereby enabling security to operate at the speed of development and deployment across multiple platforms simultaneously.

Continuous Monitoring and Feedback

A resilient system requires constant vigilance. The framework depends on continuous, real-time monitoring of applications and infrastructure across all cloud providers to detect security threats and anomalous behavior. More importantly, the data gathered from this monitoring must create a rapid feedback loop to development and operations teams [16]. This ensures that vulnerabilities or incidents identified in runtime lead to immediate corrective actions and inform future development sprints, fostering a cycle of continuous improvement.

Key Components of the Framework

Unified CI/CD Pipeline

The centerpiece of the framework is a CI/CD pipeline architected to be cloud-agnostic. This pipeline orchestrates the entire software development lifecycle (SDLC) and embeds automated security checkpoints at every stage.

Pre-Commit/Commit

Static Application Security Testing (SAST) tools and secret scanners analyze code for vulnerabilities before it is merged into the main repository.

Build

Software Composition Analysis (SCA) tools inspect dependencies for known vulnerabilities, and container images are scanned for security flaws.

Test

Dynamic Application Security Testing (DAST) tools probe running applications for vulnerabilities, while Infrastructure as Code (IaC) scanners validate configurations against security best practices [17].

Deployment & Runtime

Policies are programmatically enforced before deployment. Post-deployment, runtime security tools provide continuous threat detection and response.

Centralized Security Management and Orchestration

To overcome the challenge of fragmented controls, the framework utilizes a security management plane that provides a "single pane of glass" across all clouds. Tools such as Cloud Security Posture Management (CSPM) and Cloud-Native Application Protection Platforms (CNAPP) are critical for abstracting provider-specific details, enabling centralized visibility, misconfiguration management, and threat correlation [10].

Policy-as-Code Engine

A consistent policy enforcement mechanism is essential for governance. By implementing a centralized Policy-as-Code (PaC) engine, such as Open Policy Agent (OPA), organizations can define abstract security and compliance policies that are automatically translated and enforced within the native control planes of each CSP [18]. This ensures that rules regarding data residency, network access, and resource configuration are applied uniformly, regardless of where the application is deployed.

5. BEST PRACTICES FOR IMPLEMENTATION

Successfully operationalizing the proposed DevSecOps framework requires more than just deploying new tools; it demands a strategic approach to cultural change, technology selection, and process evolution. Adhering to the following best practices can guide organizations in navigating the complexities of integrating security into their multi-cloud development processes, ensuring a smoother and more effective transition.

Fostering a Security-First Culture

The most significant hurdle in any DevSecOps transformation is often cultural rather than technical. Success is contingent on shifting from a model where security is the responsibility of a siloed team to one where it is a collective, proactive effort. This begins with executive sponsorship that champions security as a core business enabler, not a roadblock [19]. Organizations should invest in continuous education and training for development and operations teams on secure coding practices, threat modeling, and the specific security nuances of each cloud platform they use. A "Security Champions" program can be highly effective, empowering designated developers within teams to act as security advocates, providing peer guidance and bridging the communication gap with central security teams [20]. This fosters a culture of shared ownership and embeds security consciousness directly into the development workflow.

Selecting the Right Tools

In a multi-cloud environment, tool selection is critical to avoiding further fragmentation. The primary criterion should be a tool's ability to operate seamlessly across different cloud providers. Organizations should prioritize solutions that offer a unified management interface and an extensible API for deep integration into a centralized CI/CD pipeline [21]. Key characteristics to look for include cloud-agnostic Infrastructure as Code (IaC) scanners, container security platforms that support various registries and orchestrators, and security monitoring tools that can ingest and correlate data from diverse sources like AWS CloudTrail, Azure Monitor, and Google Cloud's operations suite. The goal is to create a cohesive, interoperable toolchain that abstracts away provider-specific complexities, not to assemble a collection of disparate point solutions.

Gradual Implementation and Iteration

Attempting a "big bang" adoption of a comprehensive DevSecOps model across a multi-cloud enterprise is fraught with risk. A more prudent approach is a gradual, iterative implementation. Organizations should start with a single, high-impact application or project to serve as a pilot. This allows the team to learn, adapt the framework to their specific needs, and demonstrate value quickly [22]. Initial steps could include integrating SAST and SCA scanning into the CI pipeline and then progressively adding DAST, IaC scanning, and automated policy enforcement. Each successful iteration builds momentum and provides valuable lessons that can be applied as the framework is scaled across more teams and applications. This agile methodology minimizes disruption and allows the model to evolve based on real-world feedback.

Measuring Success

To justify investment and drive continuous improvement, it is essential to define and track key performance indicators (KPIs) that measure the effectiveness of the DevSecOps program. These metrics should go beyond traditional security measures and reflect both security posture and development velocity. The DORA (DevOps Research and Assessment) metrics provide a solid foundation, including Deployment Frequency and Change Failure Rate [23]. For security, crucial metrics include Mean Time to Remediate (MTTR) for identified vulnerabilities, security defect density, and the percentage of security tests automated within the pipeline. Tracking these metrics provides tangible evidence of progress, helps identify bottlenecks, and allows teams to make data-driven decisions to refine their processes and toolchains for enhanced resilience and efficiency [24].

6. Case Studies

Case Study 1: Financial Services Firm Adopting a Proactive Compliance Posture

A global financial services firm, bound by strict regulatory frameworks including PCI DSS and GDPR, embarked on a multi-cloud strategy to enhance service availability and leverage specialized analytics services from different providers. Their initial challenge was maintaining a consistent and auditable compliance posture across AWS and Azure [13]. Manual audits were slow and incapable of keeping pace with their DevOps release cycles, creating significant risk.

Implementation

The firm adopted a DevSecOps approach centered on "compliance-as-code." They implemented a centralized Policy-as-Code engine (using a tool similar to Open Policy Agent) to define security and compliance rules. These policies were stored in a central Git repository and automatically enforced within their CI/CD pipeline. For example, any Infrastructure as Code (IaC) template that provisioned a storage bucket without encryption or logging enabled was automatically rejected at the build stage [17]. They also leveraged a Cloud Security Posture Management (CSPM) tool to provide a unified dashboard for continuous monitoring and to flag any configuration drift in real-time across both cloud environments [10].

Outcome

By codifying compliance, the firm reduced audit preparation time by over 60% and eliminated a significant class of common misconfigurations. The automated guardrails in the pipeline allowed development teams to innovate faster, with the confidence that they were operating within pre-approved security boundaries, thus enhancing both resilience and their verifiable compliance posture [25].

Case Study 2: E-commerce Platform Enhancing Resilience for Peak Traffic

A major e-commerce platform utilized a multi-cloud architecture (GCP for data analytics, AWS for web hosting) to ensure high availability, especially during peak shopping seasons like Black

Friday. A previous outage caused by a regional provider failure highlighted the need for a more resilient and securely automated failover and deployment process.

Implementation

The platform's SRE and DevOps teams collaborated to build a unified, cloud-agnostic CI/CD pipeline. They heavily invested in containerization with Kubernetes, using managed services from both providers. Security was integrated into the container lifecycle; images were scanned for vulnerabilities at build time and a service mesh was deployed to enforce secure communication (mTLS) between microservices, regardless of which cloud they were running in [26]. Dynamic Application Security Testing (DAST) was automated within the testing phase of their pipeline, allowing for continuous security validation of their frequently updated APIs.

Outcome

The integration of automated security checks into a unified pipeline significantly improved their deployment velocity and system stability. The DORA metrics, a benchmark for DevOps performance, showed marked improvement; deployment frequency increased while the change failure rate decreased substantially [23]. During the subsequent peak season, the platform successfully handled traffic surges and automatically scaled resources across both clouds without security incidents, demonstrating a direct link between their DevSecOps maturity and enhanced application resilience.

Analysis of Findings

These cases, though from different industries, reveal a common set of findings that validate the proposed framework. Both organizations recognized that manual security processes were inadequate for multi-cloud environments and that automation was essential. The principle of treating security and compliance as code was a cornerstone of their success, enabling consistent and repeatable enforcement of policies. Both cases highlight the importance of a centralized or unified management view, whether through a CSPM tool for compliance or a unified CI/CD pipeline for deployments. This abstraction layer is critical for managing the inherent complexity of multiple providers. Finally, the empirical evidence suggests a strong correlation between mature DevSecOps practices and elite operational performance, including improved stability, availability, and faster delivery cycles [23]. These findings underscore the argument that integrating DevSecOps is not merely a security function but a critical enabler of business resilience and agility in a multi-cloud world.

7. FUTURE DIRECTIONS

While the proposed framework provides a robust model for integrating DevSecOps into multi-cloud environments, the field is continuously evolving. Several emerging areas of research and technology promise to further enhance the resilience and security of applications in these complex ecosystems. The application of Artificial Intelligence (AI) and Machine Learning (ML)

to cybersecurity represents a significant frontier. In a DevSecOps context, future research should focus on developing intelligent systems capable of automating threat detection and response in real-time across multiple clouds. These models could analyze vast datasets from logs, network traffic, and application performance to identify anomalous patterns indicative of a sophisticated attack, moving beyond signature-based detection [27]. AI can enhance the "shift-left" paradigm by providing developers with intelligent code suggestions to prevent vulnerabilities before they are even written, and by automatically prioritizing vulnerabilities based on their exploitability and potential business impact, allowing teams to focus on the most critical risks [28]. The concept of "never trust, always verify" is central to a Zero Trust Architecture (ZTA). While foundational guidance exists, significant research is needed to create practical, scalable ZTA frameworks that are inherently multi-cloud [29]. Confidential computing aims to close this gap by using hardware-based Trusted Execution Environments (TEEs) to isolate and protect data even while it is being processed in memory [30]. Future investigation should explore how to integrate TEEs seamlessly into a multi-cloud DevSecOps pipeline. This includes developing methods for attestation to verify the integrity of an enclave before use and creating cloud-agnostic tools that allow developers to build and deploy applications into secure enclaves without requiring specialized hardware expertise, thereby protecting sensitive workloads across any cloud provider. Security chaos engineering involves intentionally injecting security-focused failures such as simulating a compromised instance, revoking credentials, or disrupting a security control to identify weaknesses in a system's defenses before an attacker does [31]. Future research should focus on creating a systematic methodology and safe-to-use tooling for conducting security chaos experiments in complex multi-cloud production environments.

8. Conclusion

The proliferation of multi-cloud architectures offers undeniable advantages in flexibility and resilience, yet it simultaneously introduces a landscape fraught with security complexities that traditional models cannot address. This article has argued that integrating DevSecOps principles directly into the fabric of multi-cloud application development is not merely an enhancement but a strategic imperative. By treating security as code, embracing pervasive automation, and fostering a culture of shared responsibility, organizations can overcome the challenges of fragmented controls, inconsistent identity management, and complex compliance burdens.

The proposed framework, with its unified CI/CD pipeline and centralized management plane, provides a strategic roadmap for embedding security throughout the entire software lifecycle. The case studies presented demonstrate that such an approach yields tangible improvements in both security posture and development velocity, directly contributing to enhanced application resilience. As organizations continue to expand their cloud footprint, the systematic adoption of a multi-cloud DevSecOps model will be the definitive factor in their ability to innovate securely and confidently. The future of resilient application development lies not in choosing between

speed and security, but in unifying them through a cohesive, automated, and collaborative framework built for the complexities of the modern cloud ecosystem.

REFERENCES

[1] S. B. S. Prasad, R. R. S. Baig, and P. V. Kumar, "Multi-Cloud Security Issues and Solutions: A Systematic Review," in 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), May 2021, pp. 1297-1304. doi: 10.1109/ICICCS51141.2021.9432179.

[2] Flexera, 2021 State of the Cloud Report, 2021. [Online]. Available: [https://info.flexera.com/CM-REPORT-State-of-the-Cloud].

[3] K. R. R. Kumar and R. S. Prakash, "A Comprehensive Study on Security and Privacy Challenges in Multi-Cloud Environment," in 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI), June 2020, pp. 839-845. doi: 10.1109/ICOEI48184.2020.9142945.

[4] L. Lwakatare, A. Kuvaja, and P. Oivo, "DevOps in practice: A multiple case study of five companies," in Information and Software Technology, vol. 114, Oct. 2019, pp. 217-230. doi: 10.1016/j.infsof.2019.06.010.

[5] M. Rahman and F. Williams, "A conceptual framework for DevSecOps," in 2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE), Aug. 2018, pp. 150-155. doi: 10.1109/iCCECE.2018.8537877.

[6] V. N. Inukonda and R. V. B. A. M. Rao, "A Study on DevSecOps," in 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), Apr. 2021, pp. 1326-1331. doi: 10.1109/ICCMC51019.2021.9418385.

[7] P. Mell and T. Grance, The NIST Definition of Cloud Computing, NIST Special Publication 800-145, Sept. 2011. [Online]. Available: [https://csrc.nist.gov/publications/detail/sp/800-145/final].

[8] F. A. Amrollahi, M. S. Fallah, and S. A. G. G. H.pour, "A comprehensive study of security and privacy in the multi-cloud," Journal of Network and Computer Applications, vol. 198, Jan. 2022, 103278. doi: 10.1016/j.jnca.2021.103278.

[9] S. A. Al-Marridi, H. M. Al-Mardini, and M. A. Emmam, "A Survey of Identity and Access Management in Multi-Cloud Environments," in 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), Feb. 2020, pp. 418-423. doi: 10.1109/ICIoT48696.2020.9089539.

[10] Gartner, Market Guide for Cloud-Native Application Protection Platforms, May 2021. [Online]. Available: [https://www.gartner.com/en/documents/4001925].

[11] Cloud Security Alliance, The State of Cloud Security Concerns, Challenges, and Incidents, 2020. [Online]. Available: [https://cloudsecurityalliance.org/download/the-state-of-cloud-security-concerns-challenges-and-incidents/].

[12] N. A. Al-khater, R. A. M. Said, and M. H. H. Al-karkhi, "Identity and Access Management in a Multi-Cloud Environment: Issues and a Proposed Framework," in 2021 International

Conference on Information Technology (ICIT), Apr. 2021, pp. 699-704. doi: 10.1109/ICIT52682.2021.9491696.

[13] R. G. D. de Oliveira, "Data protection and the challenges of multi-cloud environments for GDPR compliance," Computer Law & Security Review, vol. 43, Nov. 2021, 105634. doi: 10.1016/j.clsr.2021.105634.

[14] A. Al-hazmi, A. Al-qerem, and A. Al-smadi, "Multi-Cloud-Based Security: A Survey," in 2021 International Conference on Information Technology (ICIT), Apr. 2021, pp. 883-888. doi: 10.1109/ICIT52682.2021.9491754.

[15] T. T. T. Nguyen and P. C. K. Hung, "A Framework for Security as Code in DevOps," in 2019 IEEE World Congress on Services (SERVICES), Jul. 2019, pp. 115-120. doi: 10.1109/SERVICES.2019.00037.

[16] F. A. Cabrero, D. R. Alonso, and A. B. C. Moral, "Continuous Monitoring in DevOps: A Systematic Mapping Study," IEEE Access, vol. 8, pp. 228221-228236, Dec. 2020. doi: 10.1109/ACCESS.2020.3045618.

[17] S. Hazra, S. K. Lo, and E. S. K. Yu, "A Systematic Review of Security in Infrastructure as Code," in 2021 IEEE International Conference on Web Services (ICWS), Sep. 2021, pp. 236-240. doi: 10.1109/ICWS53863.2021.00040.

[18] T. Pshakin, J. Walter, and J. H. J. D. I. de. L. Cruz, "Policy as Code: A Case Study with Open Policy Agent," in 2021 IEEE 22nd International Conference on Information Reuse and Integration for Data Science (IRI), Aug. 2021, pp. 241-247. doi: 10.1109/IRI51335.2021.00041.

[19] A. D. Hilton, “From DevOps to DevSecOps: A cultural journey,” in 2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC), Jun. 2019, pp. 1-8. doi: 10.1109/ICE.2019.8792612.

[20] T. Myrbakken and S. A. F. G. S. Stålhane, “Implementing security champions in a software development organization: An experience report,” in 2017 IEEE International Conference on Software Architecture Workshops (ICSAW), Apr. 2017, pp. 62-65. doi: 10.1109/ICSAW.2017.30.

[21] S. K. A. Islam, “A review of the state-of-the-art of DevSecOps,” IEEE Access, vol. 9, pp. 152341-152354, 2021. doi: 10.1109/ACCESS.2021.3126230.

[22] E. P. W. T. R. de Feitas, A. de Almeida, and V. C. Garcia, “A maturity model for DevSecOps,” in Proceedings of the 15th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE), 2020, pp. 645-652. doi: 10.5220/0009420806450652.

[23] N. Forsgren, J. Humble, and G. Kim, Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations. IT Revolution Press, 2018.

[24] M. Siponen and T. V. T. Le, “A framework for measuring the effectiveness of an information security program,” Information & Management, vol. 56, no. 4, pp. 520-532, Jun. 2019. doi: 10.1016/j.im.2018.11.006.

[25] D. C. D. R. de Oliveira and A. L. de Medeiros, "Automated compliance audits of cloud infrastructure with DevSecOps practices," in 2020 IEEE/ACM 5th International Workshop on a Test-driven Approach for Systems and Software Processes (DATA), Oct. 2020, pp. 27-33. doi: 10.1145/3412841.3418578.

[26] A. Singh and N. Singh, "Security of Microservices in a Containerized Environment: A Review of the State-of-the-Art," IEEE Access, vol. 8, pp. 138618-138640, 2020. doi: 10.1109/ACCESS.2020.3012297.

[27] H. Yasar, "DevSecOps Speeds Artificial Intelligence and Machine Learning Capability," Software Engineering Institute, Carnegie Mellon University, 2021 Year in Review, Aug. 2021. [Online]. Available: [https://insights.sei.cmu.edu/annual-reviews/2020-year-in-review/devsecops-speeds-artificial-intelligence-and-machine-learning-capability/].

[28] I. Ahmad, S. A. G. G. H.pour, M. S. Fallah, and F. A. Amrollahi, “AI-Driven DevSecOps: A Systematic Literature Review,” in 2021 International Conference on Computer, Information and Telecommunication Systems (CITS), Nov. 2021, pp. 1-5. doi: 10.1109/CITS52676.2021.9618585.

[29] R. Chandramouli, "Zero Trust Architecture," NIST Special Publication 800-207, National Institute of Standards and Technology, Gaithersburg, MD, Aug. 2020. doi: 10.6028/NIST.SP.800-207.

[30] Confidential Computing Consortium, "Confidential Computing: A Technical Analysis of the Threat Model," The Linux Foundation, White Paper, Oct. 2021. [Online]. Available: [https://confidentialcomputing.io/wp-content/uploads/sites/10/2021/10/Confidential-Computing-Consortium-A-Technical-Analysis-of-the-Threat-Model.pdf].

[31] A. Torkura, M. Sukmana, F. Cheng, and R. A. S. B. A. M. Rao, "CloudStrike: Chaos Engineering for Security and Resiliency in Cloud Infrastructure," in 2020 IEEE International Conference on Cloud Computing (CLOUD), Oct. 2020, pp. 110-120. doi: 10.1109/CLOUD49709.2020.00024.