



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

# IJIRSET

International Journal of Innovative Research in  
**SCIENCE | ENGINEERING | TECHNOLOGY**



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 8, August 2024

**ISSN** INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA

Impact Factor: 8.524

9940 572 462

6381 907 438

[ijirset@gmail.com](mailto:ijirset@gmail.com)

[www.ijirset.com](http://www.ijirset.com)

# Cloud DevOps and Generative AI: Revolutionizing Software Development and Operations

Karthikeyan Anbalagan

Tech Mahindra, USA



**ABSTRACT:** This article explores the transformative impact of integrating Generative AI (Gen AI) into Cloud DevOps practices. It examines how this fusion addresses longstanding challenges in software development and operations by automating repetitive tasks, optimizing resource allocation, and providing intelligent insights. The article delves into key components of this integration, including automated code and infrastructure-as-code generation, smart testing, dynamic configuration, security automation, and predictive monitoring. It also discusses implementation strategies, emphasizing the importance of seamless tool integration, continuous AI model training, and ethical considerations. While highlighting the potential benefits such as reduced time-to-market, improved code quality, and enhanced system reliability, the article also addresses the challenges organizations face in adopting Gen AI for Cloud DevOps. These challenges include technical and operational complexity, data quality dependencies, and the skill gap in the workforce. By providing a comprehensive overview of the opportunities and hurdles in leveraging Gen AI in Cloud DevOps, this article aims to equip IT professionals and decision-makers with the knowledge needed to navigate this new frontier in software development and operations.

**KEYWORDS:** Cloud DevOps, Generative AI, Software Development, Automation, Machine Learning

## I. INTRODUCTION

In the rapidly evolving landscape of software development and IT operations, Cloud DevOps has emerged as a game-changing approach to streamline processes and improve efficiency. This paradigm shift has revolutionized how organizations develop, deploy, and maintain software applications in cloud environments. Cloud DevOps combines the

principles of DevOps with cloud computing, enabling teams to leverage scalable infrastructure and automation tools to accelerate software delivery and enhance operational performance [1].

Now, with the integration of Generative AI (Gen AI), we're witnessing a new era of automation and optimization in cloud environments. Generative AI, a subset of artificial intelligence that can create new content, code, and solutions, is poised to transform the Cloud DevOps landscape dramatically. This cutting-edge technology utilizes machine learning algorithms, particularly deep learning and natural language processing, to generate human-like text, code, and even infrastructure configurations [2].

The fusion of Cloud DevOps and Generative AI promises to address longstanding challenges in software development and operations. By automating repetitive tasks, optimizing resource allocation, and providing intelligent insights, this combination has the potential to significantly reduce time-to-market, improve code quality, and enhance overall system reliability. For instance, AI-powered tools can now generate infrastructure-as-code scripts, automate testing processes, and even predict potential system failures before they occur.

This article delves into the transformative impact of Generative AI on Cloud DevOps practices. We will explore the key components of this integration, examining how AI is reshaping various aspects of the software development lifecycle in cloud environments. From automated code generation to intelligent monitoring and security enhancement, we'll uncover the myriad ways in which Generative AI is optimizing Cloud DevOps workflows.

Furthermore, we'll discuss practical implementation strategies for organizations looking to harness the power of Generative AI in their Cloud DevOps practices. This includes insights on integrating AI tools with existing DevOps pipelines, the importance of data quality in training AI models, and the need for continuous learning and adaptation in AI systems.

However, as with any technological advancement, the integration of Generative AI into Cloud DevOps is not without its challenges. We'll address the potential hurdles organizations may face, such as the complexity of implementing AI systems, the ongoing need for human oversight, and the ethical considerations surrounding AI-generated code and decisions.

By the end of this article, readers will gain a comprehensive understanding of how Generative AI is revolutionizing Cloud DevOps, the potential benefits it offers, and the considerations necessary for successful implementation. As we stand on the brink of this new frontier in software development and operations, IT professionals and decision-makers must understand the implications and opportunities presented by this powerful combination of Cloud DevOps and Generative AI.

## II. UNDERSTANDING CLOUD DEVOPS WITH GENERATIVE AI

Cloud DevOps represents a paradigm shift in the software development and IT operations landscape, aiming to unify traditionally siloed teams and processes. At its core, Cloud DevOps emphasizes the "CALMS" framework: Culture, Automation, Lean, Measurement, and Sharing [3]. This approach fosters a culture of collaboration, continuous improvement, and shared responsibility between development and operations teams, all within the context of cloud infrastructures.

In a Cloud DevOps environment, teams leverage cloud services and platforms to implement continuous integration and continuous delivery (CI/CD) pipelines, automate infrastructure provisioning, and streamline deployment processes. This integration of cloud technologies with DevOps practices enables organizations to achieve greater scalability, flexibility, and reliability in their software development and delivery processes.

Generative AI, a cutting-edge subset of artificial intelligence, is now taking Cloud DevOps to unprecedented levels of efficiency and innovation. By leveraging advanced machine learning techniques, particularly deep learning and natural language processing, Generative AI can create new content, code, and solutions that were previously the domain of human experts [4].

In the context of Cloud DevOps, Generative AI is revolutionizing several key areas:

- **Automated Code Generation:** Generative AI models can now produce code snippets, functions, and even entire modules based on natural language descriptions or high-level specifications. This capability significantly accelerates development time and reduces the likelihood of human error in coding.
- **Infrastructure Optimization:** AI algorithms can analyze usage patterns and performance metrics to optimize cloud infrastructure configurations automatically. This includes dynamically scaling resources, fine-tuning database queries, and suggesting architectural improvements.
- **Intelligent Testing:** Generative AI can create comprehensive test cases, generate synthetic test data, and even predict potential failure points in applications. This enhances test coverage and helps identify bugs earlier in the development cycle.
- **Enhanced Monitoring and Alerting:** AI-powered monitoring tools can learn from historical data to predict system anomalies and potential failures before they occur. This proactive approach allows DevOps teams to address issues before they impact end-users.
- **Security Automation:** Generative AI can assist in creating and updating security policies, identifying potential vulnerabilities in code or infrastructure, and even generating patches for known security issues.
- **Documentation and Knowledge Management:** AI can automatically generate and update documentation, create runbooks, and assist in knowledge sharing across teams, ensuring that information remains current and accessible.

The integration of Generative AI into Cloud DevOps practices is not just about automation; it's about augmenting human capabilities. By handling routine tasks and providing intelligent insights, AI frees up DevOps professionals to focus on more strategic, creative, and complex problem-solving activities.

Moreover, AI systems' self-improving nature means that as they process more data and receive feedback, they become increasingly accurate and efficient. This creates a virtuous cycle of continuous improvement in Cloud DevOps processes, leading to faster innovation, higher-quality software, and more reliable cloud infrastructures.

However, it's crucial to note that the successful implementation of Generative AI in Cloud DevOps requires careful consideration of ethical implications, data privacy concerns, and the need for human oversight. As AI systems become more integral to DevOps processes, organizations must establish clear guidelines and governance structures to ensure responsible and effective use of these powerful technologies.

As we continue to explore the synergies between Cloud DevOps and Generative AI, it's clear that this combination has the potential to redefine how we approach software development and operations in the cloud era. By embracing these technologies, organizations can unlock new levels of efficiency, innovation, and competitive advantage in an increasingly digital world.

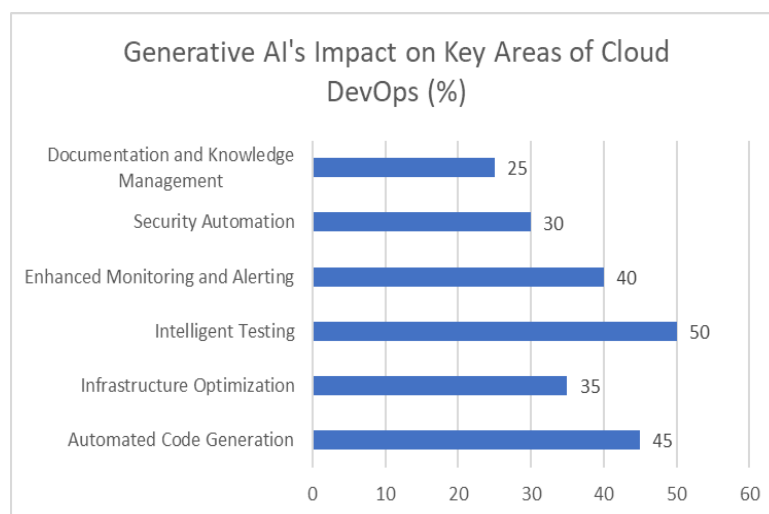


Fig. 1: Estimated Efficiency Improvements in Cloud DevOps with Generative AI [3, 4]

### III. KEY COMPONENTS OF GENERATIVE AI IN CLOUD DEVOPS

Integrating Generative AI into Cloud DevOps practices is revolutionizing the software development and operations landscape. By leveraging advanced machine learning techniques, organizations are able to automate complex tasks, optimize processes, and enhance overall efficiency. Here are the key components where Generative AI is making a significant impact:

#### 1. Infrastructure as Code (IaC) Generation

Generative AI models have emerged as powerful tools for creating and optimizing Infrastructure as Code scripts. These AI systems can analyze project requirements, best practices, and historical data to generate IaC templates that are both efficient and compliant with organizational standards [5].

For instance, AI-powered tools can:

- Generate Terraform, CloudFormation, or Ansible scripts based on high-level architectural descriptions
- Optimize existing IaC scripts for better performance and cost-efficiency
- Automatically update IaC templates to incorporate new cloud provider features or security patches

This automation significantly reduces the time and effort required to set up and manage cloud resources across development, testing, and production environments. It also minimizes configuration errors, ensuring more consistent and reliable infrastructure deployments.

#### 2. Automated Code Generation

AI-powered coding assistants, such as GitHub Copilot and OpenAI's Codex, are transforming the way developers write code. These tools can suggest or generate code snippets based on natural language descriptions or existing code contexts [6].

Key features of automated code generation include:

- Contextual code completion that understands the developer's intent
- Generation of boilerplate code, reducing repetitive tasks
- Creation of unit tests based on function specifications
- Translation of code between different programming languages

By automating routine coding tasks, these AI assistants significantly boost developer productivity, reduce the likelihood of errors, and help maintain consistency across large teams and projects.

#### 3. Smart Testing

Generative AI is revolutionizing software testing by automating the creation of test cases and test data. These AI systems leverage machine learning algorithms to analyze code structure, changes, and historical bug data to predict potential issues and generate comprehensive test suites.

Smart testing capabilities include:

- Automatic generation of unit, integration, and end-to-end tests
- Creation of realistic test data that covers edge cases
- Prioritization of test cases based on risk assessment
- Adaptive testing that evolves based on continuous integration results

This approach leads to more robust software, faster release cycles, and improved overall quality by ensuring thorough test coverage and early bug detection.

#### 4. Dynamic Configuration and Tuning

Self-optimizing systems powered by AI can dynamically adjust configurations in response to traffic patterns and system loads. These intelligent systems use machine learning algorithms to analyze performance metrics, user behavior, and resource utilization in real-time.

Key aspects of dynamic configuration and tuning include:

- Automatic scaling of cloud resources based on predicted demand
- Real-time optimization of database queries and caching strategies
- Dynamic adjustment of load balancing rules
- Intelligent resource allocation across microservices

By continuously optimizing system configurations, these AI-driven tools ensure optimal performance, improve resource efficiency, and help prevent downtime by predicting and mitigating potential issues before they impact users.

### 5. Security and Compliance Automation

AI tools are increasingly being employed to automatically generate and enforce security rules and policies within the development pipeline. These systems can analyze code, infrastructure configurations, and application behavior to identify potential vulnerabilities and ensure compliance with industry standards and regulations [7].

Security and compliance automation features include:

- Automatic generation of security policies based on best practices and regulatory requirements
- Real-time vulnerability scanning and risk assessment
- Intelligent intrusion detection and prevention
- Automated patch management and security updates

This automation enhances the overall security posture of cloud environments and ensures consistent application of security best practices across the entire DevOps lifecycle.

### 6. Predictive Monitoring and Anomaly Detection

AI-driven monitoring tools have transformed the way organizations approach system reliability and performance management. By analyzing patterns in historical data, these tools can predict potential failures and anomalies before they occur.

Key capabilities of predictive monitoring include:

- Real-time analysis of system logs and metrics to identify unusual patterns
- Prediction of potential system failures based on historical data and current trends
- Automatic correlation of events across complex, distributed systems
- Intelligent alerting that reduces false positives and prioritizes critical issues

This proactive approach significantly reduces downtime, improves system reliability, and allows DevOps teams to address potential issues before they impact end-users.

The integration of these Generative AI components into Cloud DevOps practices is not just enhancing individual processes; it's creating a new paradigm of intelligent, self-improving systems. As these AI technologies continue to evolve, we can expect even greater levels of automation, optimization, and innovation in cloud-based software development and operations.

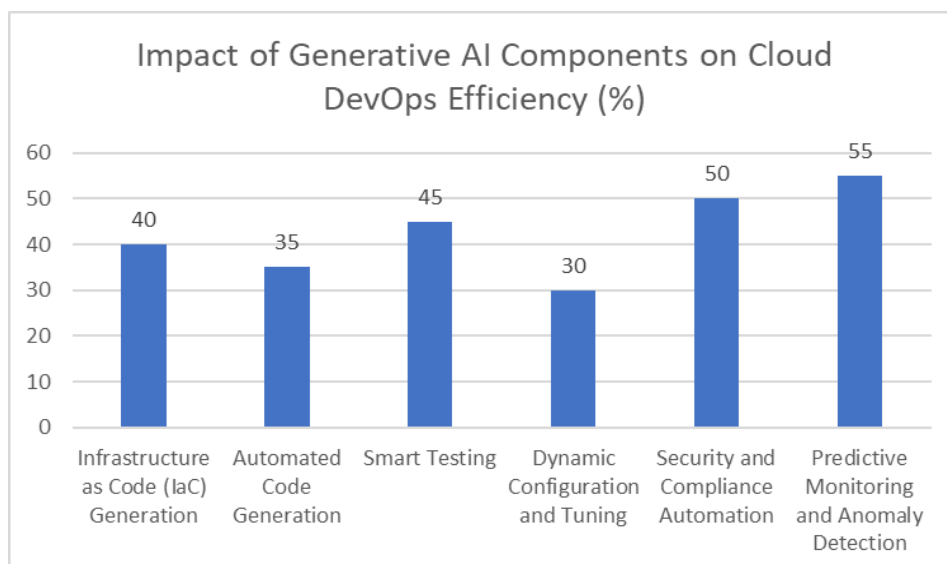


Fig. 2: Key Generative AI Components and Their Estimated Performance Improvement in Cloud DevOps [5-7]

#### **IV. IMPLEMENTATION STRATEGIES**

The successful integration of Generative AI into Cloud DevOps practices requires careful planning and execution. Organizations should consider the following key strategies to ensure a smooth adoption process and maximize the benefits of AI-driven DevOps:

##### **1. Integration with Existing Tools**

Ensuring seamless integration between Generative AI tools and current DevOps toolchains is crucial for successful implementation. This integration should encompass popular tools such as Jenkins for continuous integration and continuous delivery (CI/CD), Kubernetes for container orchestration, and Terraform for infrastructure as code (IaC) [8]. Key considerations for tool integration include:

- **API Compatibility:** Develop or select AI tools with well-documented APIs that can interact effectively with existing DevOps tools.
- **Workflow Adaptation:** Modify current workflows to incorporate AI-generated insights and suggestions without disrupting established processes.
- **Data Flow:** Establish clear data pipelines between AI systems and DevOps tools to facilitate informed decision-making and continuous improvement.
- **User Interface:** Integrate AI insights and suggestions into familiar interfaces to promote adoption among team members and reduce the learning curve.
- **Performance Monitoring:** Implement monitoring systems to track the impact of AI integration on overall DevOps performance and resource utilization.

Example integration scenarios:

- Integrating AI-powered code review suggestions into Git workflows to improve code quality and reduce review time.
- Using AI to optimize Kubernetes cluster configurations based on application performance data and resource utilization patterns.
- Leveraging AI to generate and refine Terraform scripts for infrastructure provisioning, ensuring best practices and cost optimization.

##### **2. Continuous Training and Adaptation**

Generative AI's effectiveness in Cloud DevOps relies heavily on the quality and relevance of its training data. Regularly training and fine-tuning AI models with new operational data is essential to improve accuracy and effectiveness over time [9].

Strategies for continuous training and adaptation include:

- **Data Collection Pipeline:** Implement robust systems to collect relevant operational data from various stages of the DevOps lifecycle, ensuring a comprehensive and up-to-date dataset for AI training.
- **Feedback Loops:** Establish mechanisms for DevOps teams to provide feedback on AI-generated suggestions and outputs, allowing for continuous refinement of the AI models.
- **Incremental Learning:** Utilize incremental learning techniques to update AI models without the need for complete retraining, reducing downtime and resource consumption.
- **Version Control for Models:** Implement version control for AI models to track changes, enable rollbacks if necessary, and maintain a history of model improvements.
- **Performance Metrics:** Define and monitor key performance indicators (KPIs) to assess the impact of AI on DevOps processes, such as deployment frequency, lead time for changes, and mean time to recovery (MTTR).
- **Automated Retraining Triggers:** Set up automated triggers for model retraining based on performance degradation or significant changes in the operational environment.

##### **3. Ethical Considerations and Human Oversight**

While Generative AI offers significant benefits, it's crucial to maintain human supervision for critical deployment and operational decisions. Organizations must carefully consider the ethical implications of using AI in development environments to ensure responsible and transparent use of the technology.

Key aspects to address include:

- **Bias Detection and Mitigation:** Regularly audit AI models for potential biases in their outputs or decision-making processes, implementing correction mechanisms where necessary.

- **Transparency and Explainability:** Ensure that AI-driven decisions can be explained and understood by human operators, promoting trust and accountability in the DevOps process.
- **Privacy and Security:** Implement strong data protection measures to safeguard sensitive information used in AI training and operations, complying with relevant regulations and industry standards.
- **Human-in-the-Loop Processes:** Design workflows that incorporate human judgment for critical decisions, using AI as a supportive tool rather than a replacement for human expertise.
- **Ethical Guidelines:** Develop and enforce clear ethical guidelines for the use of AI in DevOps practices, addressing issues such as data usage, decision-making autonomy, and potential societal impacts.
- **Continuous Education:** Provide ongoing training to team members on the capabilities, limitations, and ethical considerations of AI in DevOps, fostering a culture of responsible AI adoption.
- **Governance Framework:** Establish a governance framework that oversees the use of AI in DevOps, ensuring alignment with organizational values and industry best practices.

By implementing these strategies, organizations can harness the power of Generative AI to enhance their Cloud DevOps practices while maintaining control, ensuring ethical use, and continuously improving their AI-driven processes. The key lies in striking the right balance between automation and human oversight, and in creating a culture of responsible AI adoption within the DevOps team.

Implementation Strategy	Estimated Impact (1-10)	Adoption Difficulty (1-10)	Time to Implement (months)
Integration with Existing Tools	9	7	6
Continuous Training and Adaptation	8	8	9
Ethical Considerations and Human Oversight	7	6	3
API Compatibility	8	6	4
Workflow Adaptation	7	7	5
Data Flow Optimization	8	8	7
User Interface Integration	6	5	3
Performance Monitoring	7	6	4
Bias Detection and Mitigation	8	9	8
Governance Framework	7	7	6

Table 1: Key Strategies and Their Estimated Impact on Successful AI Integration in DevOps [8, 9]

### V. CHALLENGES IN ADOPTING GEN AI FOR CLOUD DEVOPS

While the integration of Generative AI into Cloud DevOps practices promises significant benefits, organizations face several notable challenges during implementation. These challenges require careful consideration and strategic planning to overcome effectively.



### 1. Complexity and Cost

Implementing and maintaining advanced AI systems in Cloud DevOps environments can be both complex and costly [10]. This challenge manifests in several ways:

- **Technical Complexity:**
  - Integration of AI tools with existing DevOps pipelines and workflows
  - Management of AI model lifecycles, including versioning, deployment, and monitoring
  - Handling of large-scale data processing and storage requirements for AI training
  - Ensuring compatibility between AI frameworks and cloud infrastructure
- **Operational Complexity:**
  - Need for specialized DevOps practices tailored to AI workflows (MLOps)
  - Increased complexity in debugging and troubleshooting AI-enhanced systems
  - Requirement for continuous monitoring and adjustment of AI models in production
- **Financial Considerations:**
  - High initial investment in AI infrastructure, tools, and talent acquisition
  - Ongoing costs for cloud resources, especially for compute-intensive AI operations
  - Potential need for premium support services for AI platforms and tools

Strategies to mitigate complexity and cost:

- Start with small, well-defined AI projects to gain experience and demonstrate value
- Leverage cloud-based AI services to reduce infrastructure management overhead
- Implement cost monitoring and optimization tools specifically for AI workloads
- Develop a clear ROI model to justify AI investments in DevOps

### 2. Data Quality Dependence

The effectiveness of AI systems in Cloud DevOps relies heavily on the quality of training data. Poor or biased data can lead to ineffective or potentially harmful outcomes [11]. Key aspects of this challenge include:

- **Data Collection and Preparation:**
  - Ensuring comprehensive coverage of DevOps scenarios in training data
  - Cleaning and preprocessing data to remove noise and inconsistencies
  - Maintaining data privacy and security throughout the AI pipeline
- **Bias Mitigation:**
  - Identifying and addressing biases in historical DevOps data
  - Ensuring fair representation of different operational scenarios
  - Regular auditing of AI model outputs for potential biases
- **Data Drift and Model Decay:**
  - Detecting changes in data distributions over time
  - Implementing strategies to keep AI models up-to-date with evolving DevOps practices
  - Balancing model stability with the need for adaptation to new data

Strategies to address data quality challenges:

- Implement robust data validation and cleansing pipelines
- Utilize data augmentation techniques to enhance training datasets
- Develop a comprehensive data governance framework specific to AI in DevOps
- Employ continuous monitoring and retraining pipelines to address data drift

### 3. Skill Gap

Successfully leveraging Generative AI in Cloud DevOps requires a workforce skilled in both DevOps practices and AI technologies, which can be challenging to assemble. This skill gap presents several challenges:

- **Multidisciplinary Expertise:**
  - Need for professionals with knowledge spanning DevOps, cloud computing, and AI/ML
  - Scarcity of individuals with deep expertise in all required domains
- **Rapid Technological Evolution:**
  - Continuous learning requirements to keep up with advancements in AI and Cloud technologies
  - Potential obsolescence of skills due to fast-paced changes in the field
- **Team Dynamics:**
  - Integrating AI specialists into existing DevOps teams

- Fostering collaboration between AI experts and traditional IT operations staff

Strategies to bridge the skill gap:

- Invest in comprehensive training programs for existing staff
- Develop partnerships with educational institutions to create tailored AI-DevOps curricula
- Implement mentorship programs to facilitate knowledge transfer within the organization
- Consider outsourcing or partnering with AI specialists for initial implementation phases
- Foster a culture of continuous learning and experimentation

By addressing these challenges proactively, organizations can smooth the path to successful adoption of Generative AI in their Cloud DevOps practices, ultimately realizing the full potential of this powerful integration. It's crucial for organizations to approach these challenges with a long-term perspective, understanding that the journey to AI-enhanced DevOps is an ongoing process of adaptation and improvement.

Challenge Category	Specific Challenge	Estimated Impact (1-10)	Difficulty to Address (1-10)
Complexity and Cost	Integration with existing DevOps pipelines	8	7
Complexity and Cost	AI model lifecycle management	7	8
Complexity and Cost	Large-scale data processing and storage	9	8
Complexity and Cost	High initial investment	8	6
Data Quality	Comprehensive coverage of DevOps scenarios	9	7
Data Quality	Data cleaning and preprocessing	7	6
Data Quality	Bias mitigation in historical data	8	9
Data Quality	Data drift and model decay	8	8
Skill Gap	Multidisciplinary expertise requirement	9	9
Skill Gap	Rapid technological evolution	8	7
Skill Gap	Integration of AI specialists with DevOps teams	7	6

Table 2: Key Challenges and Their Estimated Impact on Generative AI Integration in Cloud DevOps [10, 11]

## VI. CONCLUSION

Generative AI has the potential to revolutionize Cloud DevOps by automating complex processes, enhancing efficiency, and reducing human error. As organizations continue to adopt and refine these technologies, we can expect

to see even greater innovations in software development and operations. However, successful implementation requires careful integration, ongoing management, and alignment with business objectives and ethical standards. The fusion of Cloud DevOps and Generative AI represents a significant step forward in the evolution of software development and IT operations. As these technologies continue to mature, they promise to deliver faster, more reliable, and more secure software solutions in cloud environments.

## REFERENCES

- [1] L. Bass, I. Weber, and L. Zhu, "DevOps: A Software Architect's Perspective," Addison-Wesley Professional, 2015. [Online]. Available: <https://www.informit.com/store/devops-a-software-architects-perspective-9780134049847>
- [2] T. Brown et al., "Language Models are Few-Shot Learners," arXiv:2005.14165 [cs.CL], 2020. [Online]. Available: <https://arxiv.org/abs/2005.14165>
- [3] J. Willis, "What DevOps Means to Me," Chef Blog, 2010. [Online]. Available: <https://web.archive.org/web/20210429060955/https://blog.chef.io/what-devops-means-to-me>
- [4] A. Vaswani et al., "Attention Is All You Need," in Advances in Neural Information Processing Systems, 2017, pp. 5998-6008. [Online]. Available: <https://papers.nips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>
- [5] L. E. Lwakatere et al., "DevOps in practice: A multiple case study of five companies," Information and Software Technology, vol. 114, pp. 217-230, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/abs/pii/S0950584917302793>
- [6] M. Chen et al., "Evaluating Large Language Models Trained on Code," arXiv:2107.03374 [cs.LG], 2021. [Online]. Available: <https://arxiv.org/abs/2107.03374>
- [7] T. Lyu et al., "DeepOps: Deep Learning Based Proactive Operation and Maintenance of IoT," in IEEE Internet of Things Journal, vol. 8, no. 12, pp. 9663-9675, 15 June 15, 2021, doi: 10.1109/JIOT.2021.3051764
- [8] D. Sculley et al., "Hidden Technical Debt in Machine Learning Systems," in Advances in Neural Information Processing Systems 28, 2015, pp. 2503-2511. [Online]. Available: <https://papers.nips.cc/paper/2015/file/86df7dcfd896fcdf2674f757a2463eba-Paper.pdf>
- [9] T. Dingsøyr and C. Lassenius, "Emerging themes in agile software development: Introduction to the special section on continuous value delivery," Information and Software Technology, vol. 77, pp. 56-60, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0950584916300829>
- [10] M. Hüttermann, "DevOps for Developers," Apress, 2012. [Online]. Available: <https://link.springer.com/book/10.1007/978-1-4302-4570-4>
- [11] T. M. Mitchell, "Machine Learning," McGraw Hill, 1997. [Online]. Available: <http://www.cs.cmu.edu/~tom/mlbook.html>



INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  [ijirset@gmail.com](mailto:ijirset@gmail.com)



[www.ijirset.com](http://www.ijirset.com)

Scan to save the contact details