IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2026

DOI: 10.17148/IARJSET.2026.134116

Automated Cloud Security Drift Detection: A Risk-Aware Framework

Nishchay N. Sahoo[1], Kanak Trivedi[2], Megha Sharma[3], Aradhana Manekar[4]

Electronics and Telecommunication, Thakur College of Engineering and Technology,

Zagdu Singh Charitable Trust, Kandivali(E), Maharashtra.[1-4]

Abstract: Cloud environments are highly dynamic and continuously evolving, making them vulnerable to configuration drift, where resources deviate from their intended secure baseline settings. Such drift can occur due to manual changes, automated deployments, or misconfigured policies, leading to security risks such as excessive access permissions, exposed storage, and network vulnerabilities.

Most existing drift detection approaches focus on infrastructure consistency and lack key capabilities such as real-time monitoring, risk-based prioritization, and intent-aware analysis. Additionally, many solutions rely on periodic scanning, which is insufficient for modern cloud systems where changes occur rapidly.

To address these challenges, this paper proposes a Risk-Aware Automated Cloud Security Drift Detection Framework. The system uses event-driven audit logs to continuously monitor cloud environments, detect deviations from secure baselines, and classify them based on both risk level and intent. Based on this classification, high-risk misconfigurations are automatically remediated, while sensitive actions can be controlled through approval mechanisms.

The proposed framework is designed to be cloud-agnostic, enabling integration across major platforms such as AWS, Microsoft Azure, and Google Cloud Platform. This approach improves security visibility, reduces response time, and helps organizations maintain a stronger and more adaptive cloud security posture.

Keywords: Cloud Security, Configuration Drift, Identity and Access Management (IAM), Security Misconfigurations, Risk-Aware Detection, Automated Remediation, Event-Driven Monitoring, Multi-Cloud, Cybersecurity

I. INTRODUCTION

The rapid adoption of cloud computing has significantly transformed how organizations store, process, and manage data. Cloud platforms such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP) provide scalable, flexible, and cost-effective infrastructure. However, this flexibility introduces complex security challenges, particularly in maintaining consistent and secure configurations across large-scale and continuously evolving environments [4].

Cloud environments are inherently dynamic, where resources are frequently provisioned, modified, and deprovisioned through both manual operations and automated deployment pipelines. This dynamic nature increases the likelihood of unintended configuration changes, making it difficult to maintain a secure and compliant state over time.

One of the most critical challenges in cloud security is configuration drift. Configuration drift occurs when the actual runtime state of a cloud resource deviates from its intended or baseline configuration. For instance, a storage bucket initially configured as private may become publicly accessible, or an Identity and Access Management (IAM) role may accumulate excessive permissions beyond its original scope. Such deviations can expose sensitive data and create potential security vulnerabilities.

The impact of configuration drift can be severe. A well-known example is the Capital One data breach in 2019, where a misconfigured AWS Web Application Firewall enabled unauthorized access to over 100 million customer records.

Traditional cloud security approaches rely on periodic audits and manual reviews to detect misconfigurations. However, these methods are time-consuming, error-prone, and insufficient for modern cloud environments where changes occur rapidly. Although automated tools exist, most focus on infrastructure consistency rather than security-critical

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2026

DOI: 10.17148/IARJSET.2026.134116

misconfigurations. Cloud-native services such as AWS CloudTrail provide detailed audit logs but require additional intelligence to transform raw data into actionable security insights [5].

To address these limitations, this paper proposes a Risk-Aware Automated Cloud Security Drift Detection Framework. The proposed system leverages event-driven audit logs to enable real-time detection of configuration drift, compares runtime configurations with predefined secure baselines, and classifies detected drift events based on both risk level and intent. Additionally, the framework incorporates policy-controlled automated remediation to mitigate high-risk misconfigurations while supporting human-in-the-loop approval for sensitive actions.

By combining real-time monitoring, risk-aware analysis, and automated response, the proposed framework aims to improve cloud security posture, reduce detection and response time, and minimize the risk of critical security incidents in modern cloud infrastructures.

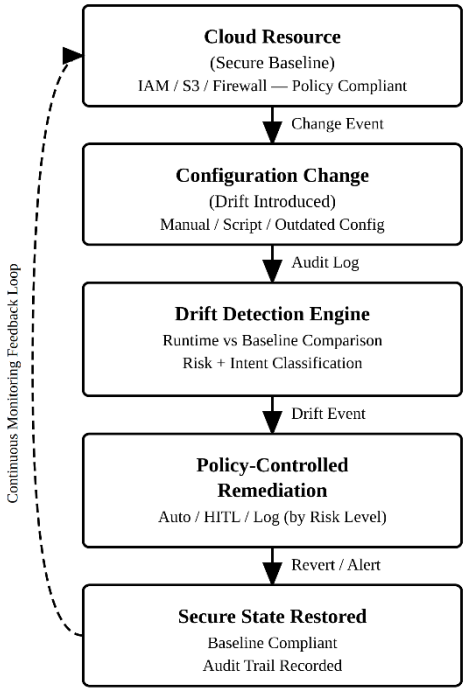

Fig. 1. Cloud Security Drift Lifecycle Overview

This figure shows the lifecycle of configuration drift in cloud environments, starting from baseline definition to drift occurrence, detection, classification, and remediation. It highlights how continuous monitoring helps maintain a secure configuration state.

II. LITERATURE REVIEW

A. AI-Driven Configuration Drift Detection (2024)

Recent research has explored the use of artificial intelligence for detecting configuration drift in cloud environments. Kumar and Mehta [1] proposed an AI-driven framework that compares Infrastructure-as-Code (IaC) baselines with runtime configurations using services such as AWS Config and Azure Config. The system employs a Random Forest classifier to categorize configurations as compliant or drifted based on extracted features. This approach significantly reduces manual monitoring effort and enhances compliance visibility by automating the detection process. Similar approaches have also been explored in recent studies [7].

The study demonstrates improved accuracy in identifying configuration inconsistencies and highlights the potential of machine learning in cloud security automation. However, the framework is limited to detection and does not provide automated remediation capabilities. Additionally, it lacks real-time event-driven analysis, relying instead on periodic evaluations. The system also does not incorporate intent-aware classification, making it difficult to distinguish between

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2026

DOI: 10.17148/IARJSET.2026.134116

legitimate administrative changes and potentially malicious activities. Furthermore, the evaluation is conducted on relatively small-scale and single-cloud environments, limiting its applicability in large-scale multi-cloud deployments.

B. Automated Drift Detection and Remediation in IaC Deployments (2024)

Patel and Singh [2] proposed an automated drift detection and remediation framework focused on Infrastructure-as-Code deployments within AWS environments. Their system utilizes AWS CloudFormation to define baseline configurations and compares them against the actual runtime state of resources. The framework integrates AWS Lambda functions with EventBridge and Simple Notification Service (SNS) to enable semi-automated detection and rollback mechanisms.

Experimental results indicate that drift detection can be achieved within 5–10 minutes, while remediation actions such as rollback can be executed within approximately 2 minutes. This significantly improves response time compared to manual approaches. However, the framework is restricted to AWS and lacks support for multi-cloud environments. It primarily relies on periodic scanning rather than continuous real-time monitoring, which limits its effectiveness in highly dynamic systems. Moreover, the approach does not incorporate AI/ML techniques, risk-based prioritization, or intent-aware analysis, reducing its ability to handle complex security scenarios. Additional implementations focusing on IaC-based drift handling are discussed in [8].

C. Automated Cloud IaC Reconciliation with AI Agents (2025)

Chen et al. [3] introduced NSync, a novel system that leverages Large Language Models (LLMs) and cloud audit logs to detect and reconcile configuration drift. The framework analyzes audit logs such as AWS CloudTrail to infer the intent behind configuration changes, consolidate drift events, and automatically generate Infrastructure-as-Code patches using Terraform. The system achieves a pass@3 accuracy of 0.97 across 372 real-world drift scenarios, demonstrating high effectiveness in automated reconciliation.

Despite its advanced capabilities, NSync primarily focuses on general infrastructure drift rather than security-specific misconfigurations. The system assumes that all detected drift should be reconciled, without considering the associated security risk or prioritizing critical issues. Additionally, while intent inference is performed, it is not explicitly categorized into actionable classes such as legitimate, accidental, or malicious.

The framework also lacks integrated risk-aware remediation strategies, which are essential for practical cloud security operations. Recent advancements using AI-based reconciliation techniques are further explored in [9].

D. Summary of Research Gaps

Based on the analysis of existing literature, several key research gaps are identified:

I. Existing approaches primarily focus on infrastructure drift rather than security-critical misconfigurations such as IAM over-permission and publicly exposed storage resources.

II. There is a lack of risk-based prioritization, making it difficult to identify and address high-impact security issues effectively.

III. Most systems do not incorporate intent-aware classification to distinguish between legitimate administrative changes and potentially malicious actions.

IV. Many solutions rely on periodic scanning instead of real-time, event-driven monitoring, which limits their responsiveness in dynamic cloud environments.

V. Current frameworks are often limited to single-cloud platforms and lack a unified, extensible architecture for multi-cloud environments.

These limitations highlight the need for a comprehensive solution that integrates real-time detection, risk-aware prioritization, intent-based analysis, and automated remediation within a unified multi-cloud framework.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2026

DOI: 10.17148/IARJSET.2026.134116

TABLE I. COMPARISON OF EXISTING APPROACHES WITH PROPOSED SYSTEM

Paper	Approach	Key Contribution	Limitation
AI-Driven Config Drift Detection (2024)	Random Forest on AWS/Azure Config logs	Compliance classification; reduced manual monitoring effort	No remediation; no real-time or intent analysis
Automated Drift Detection in IaC (2024)	AWS CloudFormation + Lambda + EventBridge	Drift detected in 5-10 min; rollback in ~2 min	Single-cloud only; periodic scan; no AI/ML or risk classification
NSync: IaC Reconciliation with AI Agents (2025)	LLMs + CloudTrail for Terraform patch generation	0.97 pass@3 accuracy on 372 real-world drift scenarios	Not security-specific; no risk or intent-based prioritization

III. PROPOSED SYSTEM / METHODOLOGY

This section presents the design and working of the proposed Risk-Aware Automated Cloud Security Drift Detection Framework. The system is structured as a modular and scalable architecture consisting of five key components: Baseline Definition, Event-Driven Log Monitoring, Drift Detection Engine, Risk and Intent Classification, and Policy-Controlled Remediation. The overall workflow of the system is illustrated in Fig. 2.

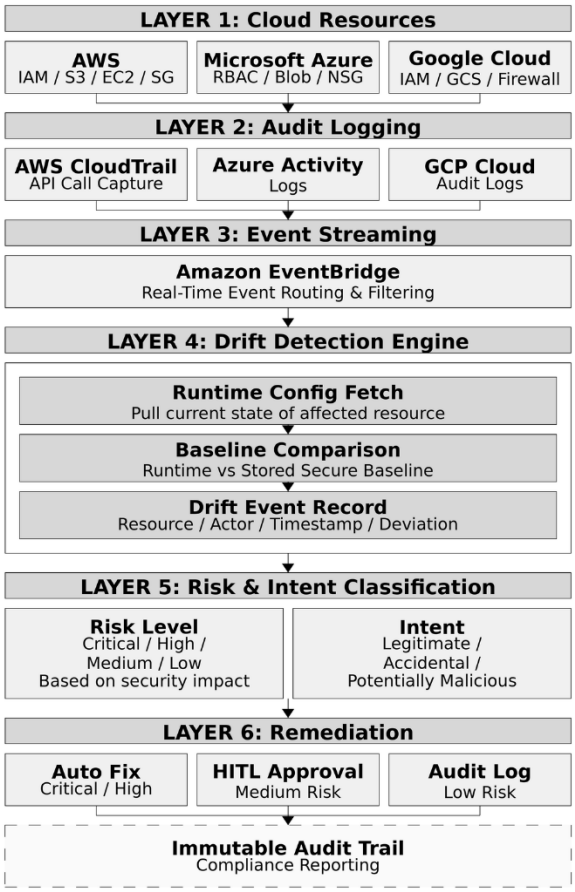

Fig. 2. System Architecture of the Proposed Framework

This figure illustrates the overall architecture of the proposed system, showing the interaction between baseline definition, event-driven monitoring, drift detection, classification, and remediation modules.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2026

DOI: 10.17148/IARJSET.2026.134116

A. Baseline Definition

The foundation of the proposed framework lies in defining secure configuration baselines for cloud resources. These baselines represent the intended and policy-compliant state of resources such as Identity and Access Management (IAM) roles, storage buckets, security groups, and firewall rules. Baselines are defined using structured formats such as JSON or YAML and are aligned with industry security standards including CIS Benchmarks and AWS Security Best Practices.

Each baseline acts as a reference model against which real-time configurations are evaluated. Any deviation from this predefined state is considered a potential security drift.

By formalizing baseline definitions, the system ensures consistency, compliance, and a clear benchmark for detecting misconfigurations.

B. Event-Driven Log Monitoring

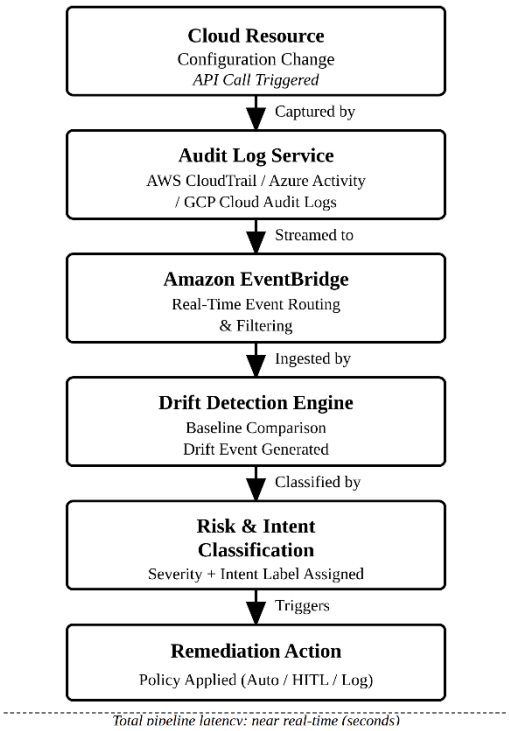

Fig. 3. Event-Driven Monitoring Pipeline

This figure represents the real-time monitoring pipeline where cloud audit logs are streamed through event-driven services and processed instantly for drift detection.

Unlike traditional approaches that rely on periodic scanning, the proposed system adopts an event-driven monitoring strategy. Cloud-native audit logging services such as AWS CloudTrail, Azure Activity Logs, and GCP Cloud Audit Logs continuously capture configuration changes and API activities across the cloud environment.

These logs are streamed in real time using event-driven services such as Amazon EventBridge or Azure Event Grid. As soon as a configuration change occurs, the corresponding event is immediately forwarded to the Drift Detection Engine. This approach ensures near real-time visibility into configuration changes, significantly reducing detection latency compared to batch-based monitoring systems.

As shown in Fig. 3, this event-driven pipeline enables continuous monitoring and rapid processing of configuration changes, making the system highly responsive to potential security threats.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2026

DOI: 10.17148/IARJSET.2026.134116

C. Drift Detection Engine

The Drift Detection Engine is the core analytical module responsible for identifying configuration drift. Upon receiving an event, the engine retrieves the current runtime configuration of the affected resource and compares it with the corresponding baseline configuration.

This comparison is performed using a structured policy evaluation algorithm that checks for deviations in permissions, access rules, and configuration parameters. If a mismatch is detected, the system generates a drift event record containing detailed information such as resource identifier, type of deviation, timestamp, actor identity, and change metadata.

The generated drift event is then passed to the Risk and Intent Classification module for further analysis. This modular approach ensures scalability and allows the detection engine to operate independently across multiple cloud services.

D. Risk and Intent Classification

A key contribution of the proposed framework is the dual classification of drift events based on both risk level and intent. Risk classification categorizes each drift event into four severity levels: Low, Medium, High, and Critical. This classification is based on the potential security impact of the misconfiguration. For example, enabling public access to a storage bucket or assigning wildcard IAM permissions is classified as Critical, whereas minor configuration mismatches such as tagging inconsistencies are categorized as Low risk.

In addition to risk assessment, the system performs intent classification by analyzing contextual information from audit logs. Factors such as user identity, time of change, frequency of similar actions, and authorization status are used to determine whether the change is Legitimate, Accidental, or Potentially Malicious.

This dual-layer classification helps reduce false positives and ensures that remediation actions are applied selectively to the most impactful and suspicious drift events.

TABLE II. COMPARISON OF EXISTING APPROACHES AND PROPOSED SYSTEM

Feature	Paper 1	Paper 2	Paper 3	Proposed System
Security-specific drift	Partial	No	No	Yes
Risk-aware prioritization	No	No	No	Yes
Intent-aware classification	No	No	Partial	Yes
Real-time event-driven	No	No	Yes	Yes
Automated remediation	No	Yes	Yes	Yes (policy-controlled)
Multi-cloud support	Partial	No	Partial	Yes (extensible)

E. Policy-Controlled Automated Remediation

Based on the classification results, the system initiates a policy-controlled remediation workflow. For High and Critical risk drift events, automated remediation is triggered immediately to restore the resource to its secure baseline configuration. This is achieved using cloud-native APIs such as AWS Systems Manager Automation or Azure Policy remediation services.

For Medium-risk events, alerts are generated and forwarded to the security team along with recommended remediation actions. Low-risk events are logged for auditing and compliance purposes without immediate intervention.

To handle sensitive operations, the framework incorporates a Human-in-the-Loop (HITL) mechanism, where security administrators can approve or reject remediation actions before execution. This ensures a balance between automation

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2026

DOI: 10.17148/IARJSET.2026.134116

and control. As illustrated in Fig. 4, the remediation workflow adapts dynamically based on the risk level of detected drift events.

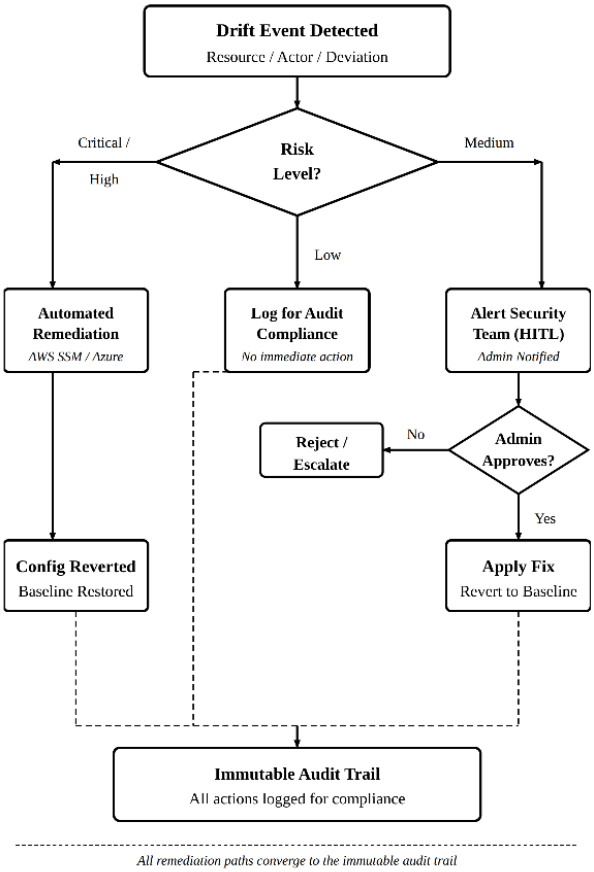

Fig. 4. Remediation Workflow Based on Risk Level

This figure shows the decision-making process for remediation, where actions are triggered based on risk severity and may involve automated execution or human approval.

F. Multi-Cloud Extensible Architecture

The proposed system is designed with a cloud-agnostic architecture, enabling seamless integration across multiple cloud platforms. While the initial implementation focuses on AWS, the framework includes an abstraction layer that allows extension to Microsoft Azure and Google Cloud Platform.

Cloud-specific adapters handle variations in API structures, audit log formats, and resource models, while the core detection and classification logic remains platform-independent. This design ensures scalability, flexibility, and applicability in hybrid and multi-cloud environments, making the system suitable for modern enterprise infrastructures.

IV. CONCLUSION AND FUTURE WORK

This paper presented a Risk-Aware Automated Cloud Security Drift Detection Framework designed to address critical limitations in existing cloud security solutions. Unlike traditional approaches that primarily focus on infrastructure consistency or periodic monitoring, the proposed system emphasizes real-time detection of security-critical configuration drift using an event-driven architecture.

The framework integrates multiple key components, including secure baseline definition, continuous log monitoring, drift detection, and dual-layer classification based on risk and intent. This approach enables the system to not only identify configuration deviations but also prioritize them based on their potential impact and underlying cause. By incorporating

DOI: 10.17148/IARJSET.2026.134116

policy-controlled automated remediation, the framework ensures that high-risk misconfigurations are resolved promptly while maintaining control through human-in-the-loop mechanisms for sensitive actions.

The proposed system offers several important contributions. First, it focuses specifically on security-related misconfigurations such as IAM over-permission, exposed storage, and network vulnerabilities. Second, it introduces a dual classification mechanism that combines risk severity and intent analysis, improving decision-making and reducing false positives. Third, it provides a cloud-agnostic and extensible architecture, allowing deployment across multiple cloud platforms including AWS, Microsoft Azure, and Google Cloud Platform.

Overall, the framework enhances cloud security posture by improving visibility, reducing detection and response time, and minimizing the risk of large-scale security incidents. It is particularly well-suited for modern enterprise environments that operate in dynamic and multi-cloud settings.

Future work will focus on implementing and validating the framework in real-world cloud environments to evaluate its performance under practical conditions. This includes measuring key metrics such as detection latency, false positive rate, and remediation effectiveness. Additionally, machine learning models can be integrated to enable predictive drift detection and anomaly-based analysis. The intent classification module can be further enhanced using behavioral analytics and threat intelligence feeds to improve accuracy in identifying malicious activities. Finally, the framework can be extended with advanced visualization dashboards and compliance reporting tools to support security operations and auditing requirements.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the Department of Electronics and Telecommunication Engineering at Thakur College of Engineering and Technology, Mumbai, for providing the necessary support and resources to carry out this research.

The authors also extend their appreciation to the faculty members for their valuable guidance, constructive feedback, and continuous encouragement throughout the course of this work. Their insights have played a crucial role in shaping the direction and quality of this research.

REFERENCES

[1]. S. Kumar and A. Mehta, "AI-Driven Configuration Drift Detection in Cloud Environments," International Journal of Cloud Computing and Security, vol. 12, no. 3, pp. 45–57, 2024.
[2]. R. Patel and J. Singh, "Automated Drift Detection and Remediation in Infrastructure-as-Code Deployments," in Proceedings of the IEEE International Conference on Cloud Engineering (IC2E), 2024, pp. 112–119.
[3]. L. Chen, M. Wang, and Y. Zhang, "NSync: Automated Cloud Infrastructure-as-Code Reconciliation with AI Agents," in Proceedings of the ACM Symposium on Cloud Computing (SoCC), 2025, pp. 234–247.
[4]. Cloud Security Alliance, "Security Guidance for Critical Areas of Focus in Cloud Computing v4.0," CSA, 2021.
[5]. Amazon Web Services, "AWS CloudTrail Documentation," 2024. [Online]. Available: https://docs.aws.amazon.com/cloudtrail
[6]. National Institute of Standards and Technology (NIST), "Security and Privacy Controls for Information Systems and Organizations," NIST SP 800-53 Rev. 5, 2020.
[7]. G. Thiyagarajan, V. Bist, and P. Nayak, "AI-Driven Configuration Drift Detection in Cloud Environments," International Journal of Communication Networks and Information Security (IJCNIS), vol. 16, no. 5, pp. 721–743, 2024.
[8]. A. M. Solanki, "Automated Drift Detection and Remediation in Infrastructure-as-Code Deployments," MSc Research Project, School of Computing, National College of Ireland, Apr. 2024.
[9]. Z. Yang, H. Guan, V. Nicolet, B. Paulsen, J. Dodds, D. Kroening, and A. Chen, "Automated Cloud Infrastructure-as-Code Reconciliation with AI Agents," arXiv preprint arXiv:2510.20211v1, Oct. 2025.